Springboard Capstone Project 1

King County House price predictions

Final Report

Hye Joo Han

1. Problem Statement

Below (imaginary firm), a local real estate company serving King County in Washington, US, asked to build a house price prediction model. The company wants to utilize the model to provide their house price estimations to their customers, house sellers and buyers. The goal of this project is collecting data with house prices and finding the best model that can predict house sale prices with the least amount of errors.

2. Dataset and Data Wrangling

Collecting data

The main dataset, House Sales in King County, USA, was from Kaggle (link: https://www.kaggle.com/harlfoxem/housesalesprediction). The data contains records of house sales for King County included in the Seattle–Tacoma–Bellevue metropolitan area in Washington, US. The dataset has one year of records for houses sold between May 2014 and May 2015, but this project assumed that it is recent data. It consists of 19 house features, house id numbers, dates sold and sale prices for 21,613 house sales. The house features include the number of bedrooms and bathrooms, square footages of home and lot, year built, zip code, latitude and longitude of a house associated with each sale.

Zip codes can give some useful information about a house location, which is an important factor people consider when they buy a house. I looked for where to obtain such information using zip codes and found the website www.niche.com which provides grades from A+ to D- for public school, safety, cost of living, jobs, commute, etc. for each zip code. I collected those grades for relevant zip codes by web scraping.

Cleaning data

3 major cleaning steps were performed.

(1) Clean house data

In this step, I cleaned the main dataset which has house price and features for each sale. First, I checked overall information, description and plots of the dataset. I found some columns have unideal data types and suspicious values that can be missing values or outliers. The overall info did not show any typical missing values, but I found that zeros in the columns for number of bedrooms, number of bathrooms, and year renovated are actually missing values. Since there were only 16 observations with zero bathroom or zero bedroom (0.07% of all rows), those observations were removed. The column for renovated years has zeros 96% of times possibly because most houses had no major renovations recorded and only had minor updates. I filled the zeros with yr_built values to make a hybrid column, but in the machine learning part I made a new column named *renovated* that has 1 for renovated houses and 0 for not renovated.

I found some outliers in the main dataset using the describe() function and box plots. Most of them were plausible outliers except for one with 33 bedrooms. Using other features of the house, I guessed that 33 was a typo made by pressing 3 twice instead of once. Thus, I replaced 33 with 3. Another extreme outlier showed 1,651,359 sqft (38 acres) for lot size, but I did not do anything since it is an old (1920) small house with a huge lot by a cemetery according to Google Map.

I changed the data types of columns for zip code and house id into string, and made the column for dates sold as datetime type.

(2) Clean zip code data

In this step, I gathered grades (from A+ to D-) for 12 categories for each zip code by scraping Niche.com with a Python package, Beautiful Soup, for parsing HTML and XML documents. Niche.com had information about every zip code (70 in total) in the main dataset. The 12 categories are public school, crime, cost of living, jobs, commute, nightlife, housing, good for families, diversity, weather, outdoor activities, and commute. I removed one column (for crime) which has the same value for all zip codes. Next, I transformed the alphabet grades to score grades (from 4.3 to 0.7) and kept both alphabet and score grades for later analysis.

I found the diversity grade for 4 zip codes is NG meaning ungraded. When I transformed alphabet grades into score grades, I mapped NG to None representing missing values. Then, the missing values were filled with the median diversity score 3.7. Finally, the missing alphabet grade NG was replaced with A- which corresponds to 3.7.

(3) Merge zip code data into house data

After cleaning those dataframes in step (1) and (2), the zip code dataset was merged into the main dataset by left join on zip codes. In other words, the additional 11 columns derived from zip codes were added to the main house dataset by matching zip codes.

3. Exploratory Data Analysis

The cleaned and merged dataset contained 21,597 observations (house sales) each with 44 features. I did univariate, bivariate and multivariate analyses along with visualizations. I also performed some hypothesis testing for more statistically rigorous statements. The followings are the summary of what I found through exploratory data analysis (EDA) and inferential statistics.

Univariate analysis

Figure. Histograms of some numerical variables
(y-axis: frequency and x-axis: labeled on top of each histogram)

The above figure shows there are many skewed features including price, bedrooms, bathrooms, square footage related features starting with 'sqft_'. There are also many ordinal (not interval) variables such as conditions and grade. All features from zip code (not shown here) are also ordinal. For this reason, in bivariate analysis, the Spearman (or Kendal's tau) correlation is more appropriate than the Pearson correlation if significance tests are required. Spearman correlation, unlike the Pearson correlation, does not assume that both variables have

normal distributions. I used Spearman correlation (with p-values) in the inferential statistics part while I used Pearson correlation (without p-values) in the data storytelling part. Here I report only the Pearson correlation values since they are more consistent to the result from the machine learning part.

Bivariate analysis

The first row of the scatter plots and correlation heat map in the below figures shows that many features are correlated with house prices. The scatter plots also show some heteroscedastic relationships which violates one of the assumptions of linear regression, homoscedasticity. Thus, a linear regression is used as a model, results of significance tests should be carefully interpreted since they might not be valid.

Bright squares in the heat map show many independent variables are highly correlated with each other (multicollinearity). I further looked into highly correlated features with correlation coefficient values and interpreted below..

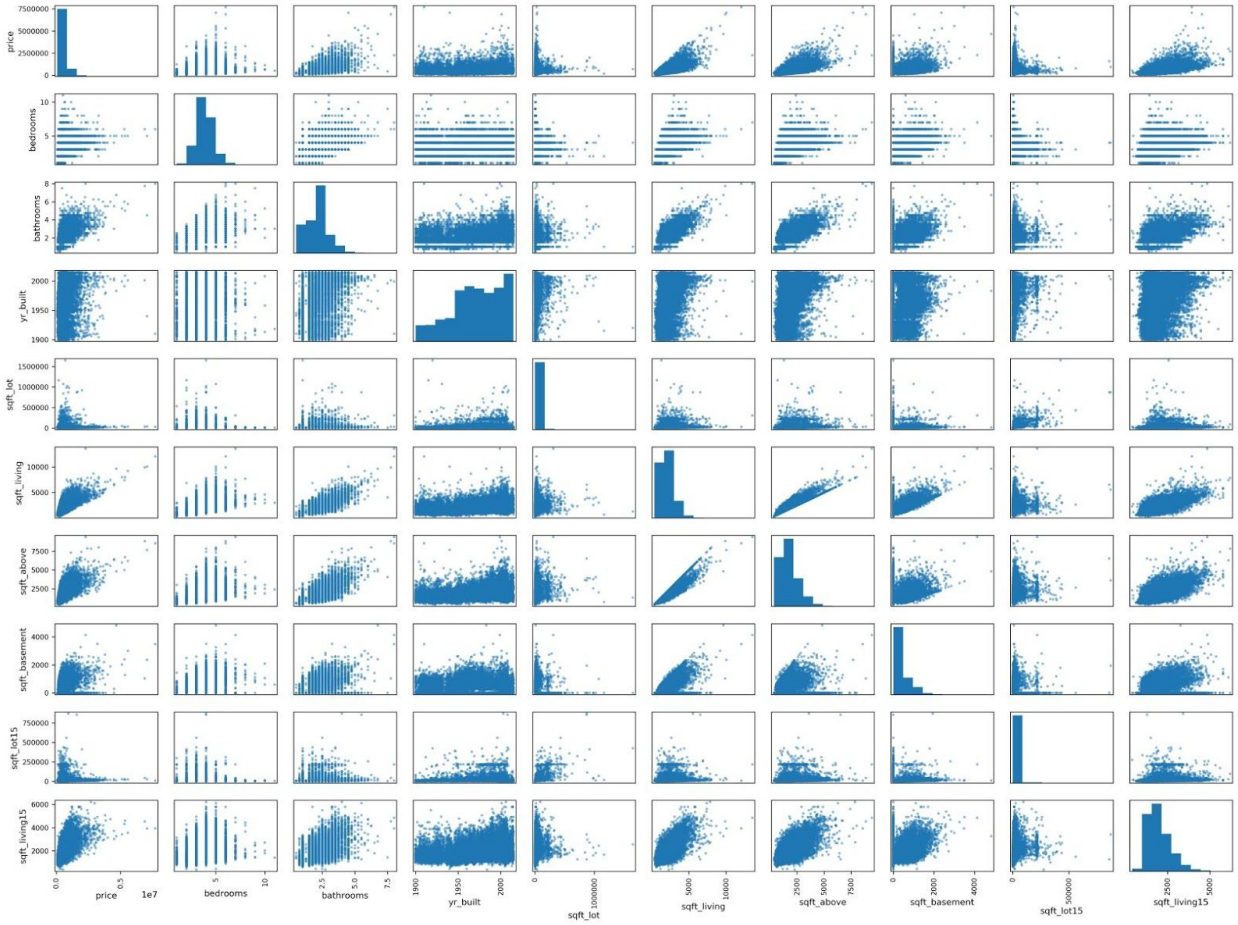


Figure. Scatter plots of numerical variables

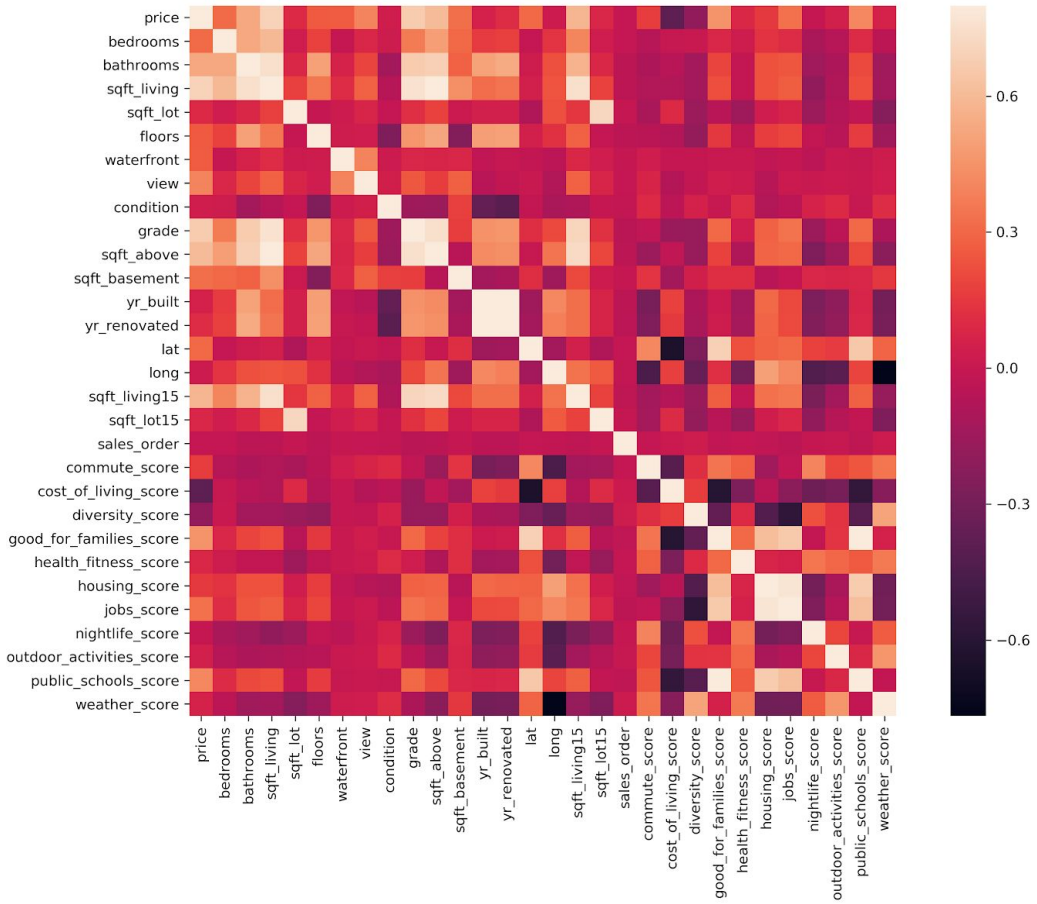


Figure.Correlation heat map of numerical variables

Features most correlated with house prices:

	correlation to price
sqft_living	0 .7020
grade	0.6679
sqft_above	0.6054
sqft_living15	0.5852
bathrooms	0.5259
good_for_families_score	0.4544
public_schools_score	0.4115
view	0.3973
jobs_score	0.3316
sqft_basement	0.3238
bedrooms	0.3160
lat	0.3068
cost_of_living_score	-0.3845

House size related		Zip code related		Others	
sqft_living	.70	good_for_families_score	.45	grade	.67
sqft_above	.61	public_schools_score	.41	view	.40
sqft_living15	.59	jobs_score	.33	Lat	.31
bathrooms	.53	cost_of_living_score	-.38		
sqft_basement	.32				
bedrooms	.32				

The above table shows Pearson correlation coefficients between price and the features highly correlated with price. I only listed the features whose correlation with price is over 0.3 or less than -0.3 to show moderate or strong correlations . It shows house sizes (square footages), house grade and the number of bathrooms are the most correlated features to house prices. The scores for families, public schools, and jobs, number of bedrooms and latitudes are also moderately correlated with house prices. The score for cost of living is the only feature that has moderate (at least) negative correlation with price.These features can be good predictors for a house price prediction model. I will see if there features have high importances in the best machine learning models to be found.

Most correlated independent variables:

	correlation
Good_for_families_score vs public_schools_score	0.9133
yr_built vs yr_renovated	0.9099
sqft_living vs sqft_above	0.8764
housing_score vs jobs_score	0.7734
sqft_living vs grade	0.7628
sqft_living vs sqft_living15	0.7564
grade vs sqft_above	0.7561
bathrooms vs sqft_living	0.7558
sqft_above vs sqft_living15	0.7318
sqft_lot vs sqft_lot15	0.7182
grade vs sqft_living15	0.7139
long vs weather_score	-0.7673

The above table shows Pearson correlation coefficients for highly correlated features (independent variables). I only listed the features whose correlation with price is over 0.7 or less than -0.7. Many independent variables are highly and significantly correlated; 12 pairs of independent variables have correlation above 0.7 and below -0.7. The top two pairs, who have over 0.9 of correlation, are good_for_families_score vs. public_schools_score and yr_built vs. yr_renovated. The yr_built and yr_renovated columns have high correlations because of too many overlapping values and the first pair shows good locations for families have good public

schools. The highly correlated independent variables should be carefully treated in a model since multicollinearity can cause many problems (e.g., unreliable feature importances). It can be avoided using some feature selection or feature projection methods.

Relationship between house price and categorical independent variables:

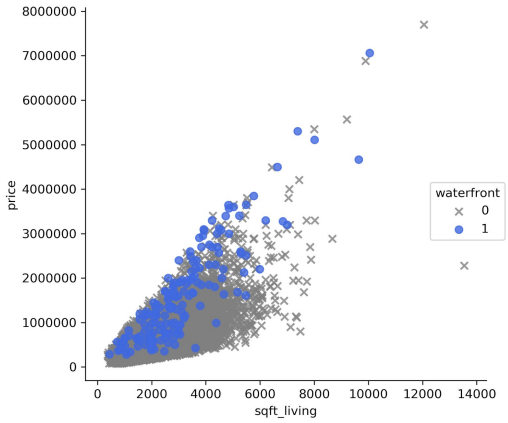
A two-sample independent *t*-test for comparing the population means was used to find a significant categorical variable in a house price prediction. I used Welch's *t*-test which does not assume equal variance or equal sample size in two populations. I had to use it because the two groups in my categorical variables have very unbalanced sample sizes.

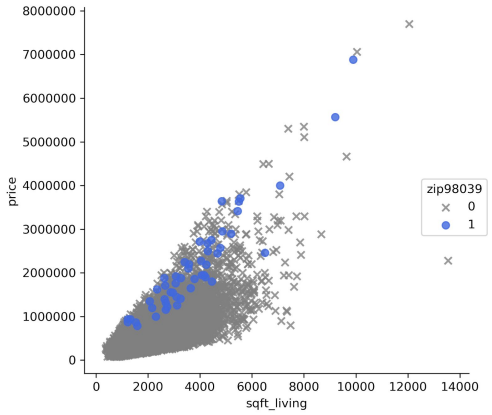
Waterfront vs Not waterfront

```
two-sample independent t-test
test statistic :  -12.8746
p-value :  0.0
```

98039 vs not 98039

```
two-sample independent t-test
test statistic :  -9.8445
p-value :  0.0
```





I found waterfront houses are significantly more expensive than not waterfront houses. I also found houses in zip code 98039 are significantly more expensive than houses not in the zip code. The price differences can be made just because the houses in waterfront or zip code 98039 have better qualities (e.g. bigger sqft_living), but the above two scatter plots show waterfront and 98039 houses are more expensive comparing to the other houses with similar square footages. Although there still can be some other features making the higher prices in those areas, I will add a feature indicating if a house is in 98039 to my models in the machine learning part.

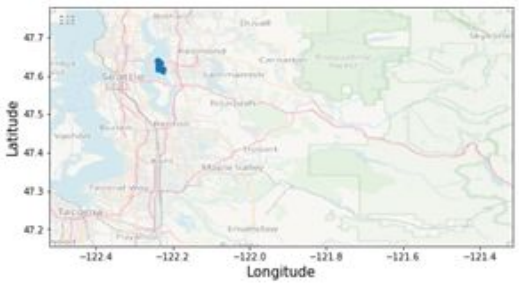


Figure. Medina, Washington. I found the place with zip code 98039 is a city called Medina on a peninsula in Lake Washington. This city is where many billionaires like Bill Gates and Jeff Bezos live.

Since the dataset contains geographical coordinates of a house, longitude and latitude, it was possible to draw houses on the King county map using colors for another feature. The figure below shows houses on the map colored with price. This map also shows waterfront houses and houses in 98039 are those with extremely expensive prices (orange and red dots). The map also shows inexpensive houses (blue) are mostly in the south and east of King county.

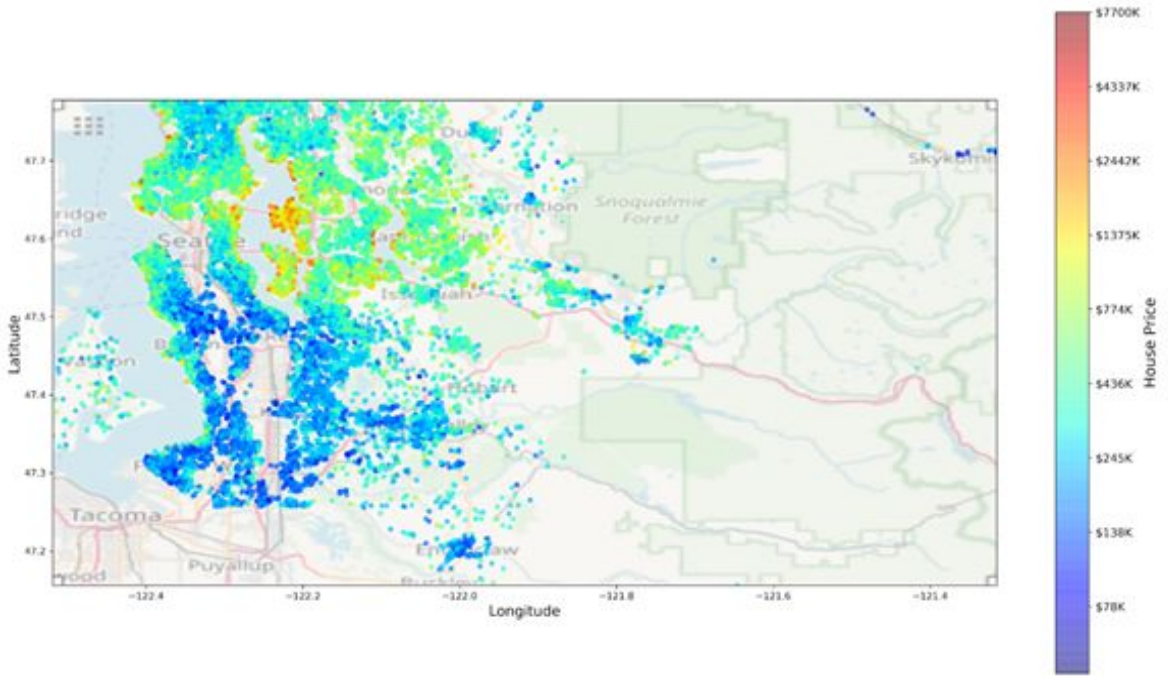
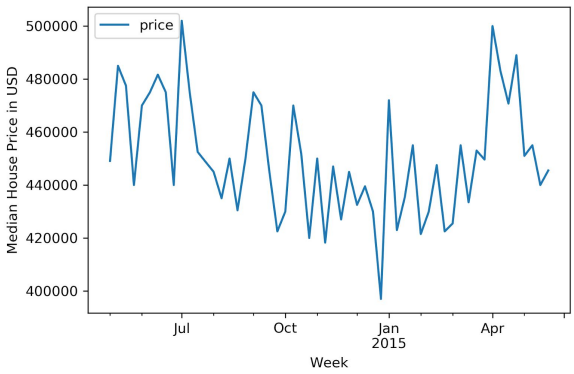
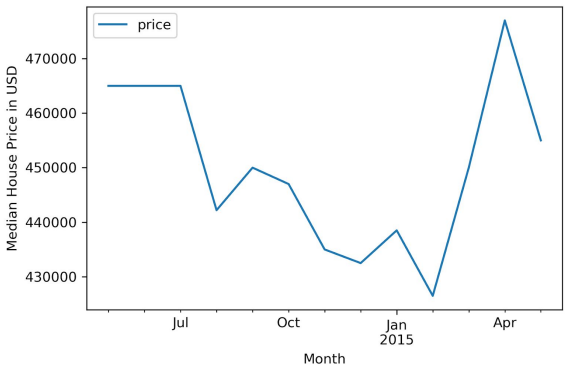
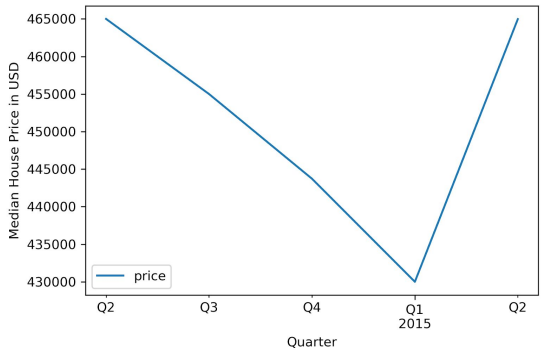


Figure. Houses on the King county map colored with house prices

Time-series Analysis



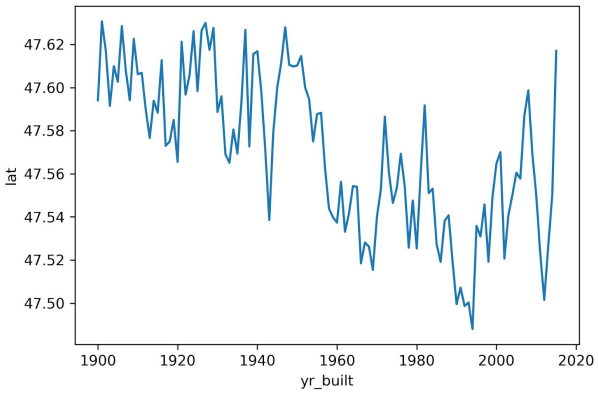


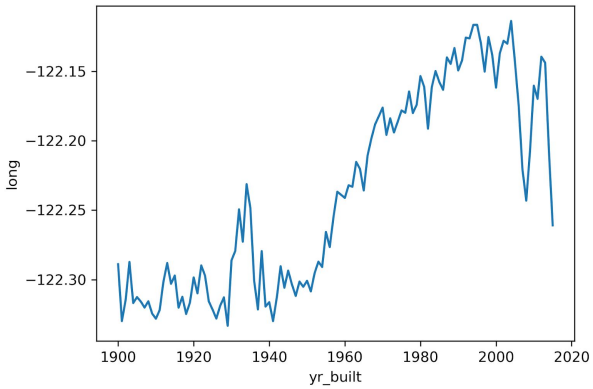


How times of the year impact housing prices? Some time series analysis was done using house prices and dates sold. The top left graph shows weekly median house prices. Although it shows a lot of fluctuations, there seems to be an overall v-shaped pattern. To see the pattern more clearly with less fluctuations, median house prices were aggregated for each month (top right). Now I can see the median prices going down from May to early months in 2015 and then going back up although they still fluctuate a little. Finally, the bottom left graph showing the quarterly median house prices, shows an exact v-shape pattern. The time series analysis showed house prices are higher between April and July and lower around January or February. This pattern suggests it is better to buy a house in winter and sell a house around late spring or early summer. To use this pattern in machine learning, a new categorical variable for sold months will be extracted from the variable 'date' for sold dates and used a feature in house price prediction models.

Dispersion of Seattle

It was found that the mean latitude tends to decrease and the mean longitude tends to increase as year built increases (see the below figures). The pattern seems to show how the Seattle metropolitan area has expanded from Seattle to the east and south suburb cities since 1900 (within King county). The map figure colored with built years showed a consistent result. Very old houses (blue) are mostly concentrated in Seattle while locations of newer houses are getting dispersed to the east and south of Seattle.





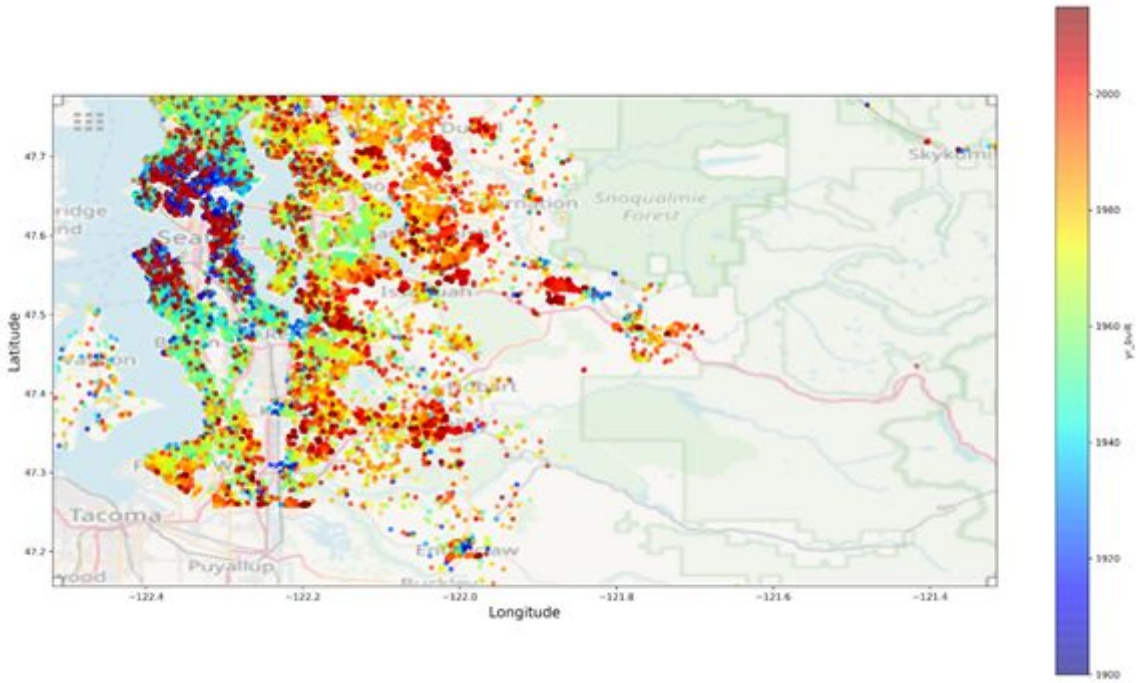
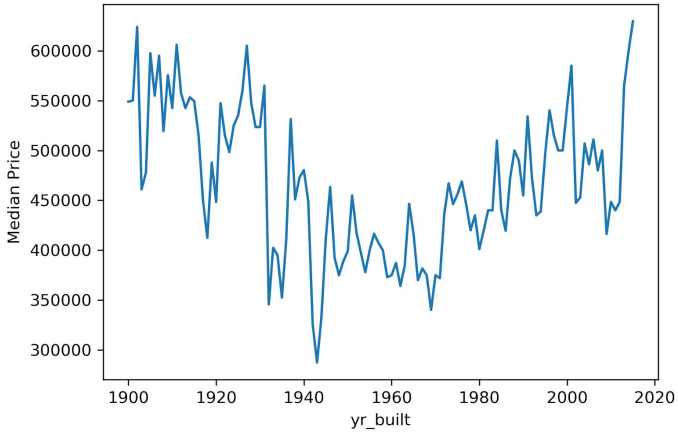


Figure. Houses on the King county map colored with built years

The old houses built before 1940 concentrated in Seattle might be one factor causing the interesting pattern I found in the mean and median house prices as a function of built years; newer houses built after 1960 tend to be more expensive, but for houses built before 1960, older houses tend to be more expensive (see the figure below). The below graph and low correlation (0.05) found between house prices and years built show the house prices are not linearly related with built years. However, there is still some pattern between them and this pattern could be captured by non-linear models like tree-based ones.



4. Machine Learning

The goal of this part of project was finding the best models that can predict house sale prices using various machine learning techniques. The followings are the summary of what I have done and found from the In-depth machine learning analysis.

Data Preparation

In the data wrangling part, I took care of missing values and invalid outliers and created some new features using existing columns and web scraping. Here the dataset was further prepared to build machine learning models using the following 4 steps. There were price (target) and 41 features (predictors) left at the end of these steps.

(1) Three new features were created from the existing columns.
- sold_month: month names were extracted from ‘date’ column
- renovated: 1 for houses with known renovated years and 0 for others
- zip98039: 1 for houses in zip code 98039 and 0 for others

(2) 15 columns were removed
- 'date' and 'yr_renovated' are removed since they were transformed to new columns.
- 'zipcode' is removed since it has too many levels for one-hot encoding and was used to make 12 new columns ending with 'grade'.
- The columns ending with 'grade' were were dropped since they were transformed to numerical columns ending with 'score' in the data wrangling part.

(3) One-hot encoding was applied to the non-numerical columns ‘sold_month’, which became 11 columns.
(4) The dataset was splitted into training (80%) and test (20%) sets after being shuffled.

Machine Learning Algorithms

The followings are the 5 machine learning algorithms used to build models and compared.

Ridge regression (Linear regression)
Random Forest
XGBoost
LightGBM
Neural Network

Ridge regression, linear regression with L2 regularization, was chosen as a base model to be compared with other more complex models of interest since I found many variables are strongly or moderately correlated with house prices in the EDA part. Decision tree-based

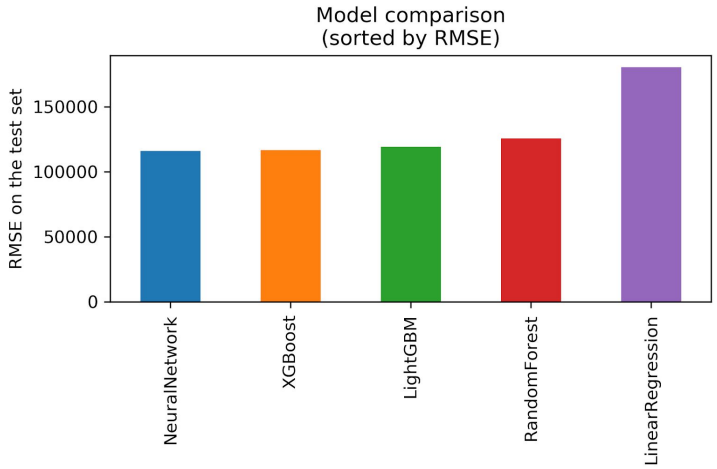
ensembles have been winning algorithms in many machine learning competitions like Kaggle, so I decided to try Random Forest, XGBoost, and LightGBM. Random Forest is an ensemble learning method that operate multiple Decision Trees via bagging. XGBoost and LightGBM are also ensemble methods, but with gradient boosted decision trees enhanced for speed and performance. XGboost uses level-wise tree growth while LightGBM uses leaf-wise tree growth. Neural Networks were chosen to add some fun variation in the choice of algorithms.

Model building summary

Feature scaling: If necessary or helping predictions, the feature scaling step was performed before fitting data into models.
Cross Validation and Hyperparameter tuning: For cross validation and hyperparameter tuning, GridSearchCV with 5 folds was used except for Neural network models. I used coarse to finer hyperparameter tuning to find the best model for each algorithm. I added early stopping for Neural Network models to avoid overfitting.
Metric: I selected the root mean squared error (RMSE) for the evaluation metric and the mean squared error for a regression loss function. The R^2 value on the test set was calculated just to feel how much variability in house prices is explained by a model

Performance Summary

	RMSE_test	RMSE_val	R_squared	Time	Scaling
NeuralNetwork	116252	118819	0.8961	22min 37s	True
XGBoost	116750	114288	0.8942	43.4 s	True
LightGBM	119439	119955	0.8903	18.5 s	True
RandomForest	125872	131775	0.8782	2min 44s	False
LinearRegression	180464	178080	0.7497	507 ms	True



RMSE: The table and graph shows model performances ordered by RMSE. The top three models are Neural network, XGBoost, and LightGBM. They had very similar RMSE and r-squared scores (between .89 and .90), so the ranking can be easily changed by a different random state or test-training set split. Since k-fold cross validation was not done with Neural networks due to early stopping, its average performance on multiple folds can be lower than XGBoost and LightGBM. The next best model is Random Forest and its RMSE is still pretty low and r-squared value is around .88. Linear regression was the absolute worst method.

Speed: LightGBM was the fastest among its competitors and the next best was XGBoost (I'm not considering linear regression due to its poor performance in RMSE). Neural network would be the worst choice timewise.

Tuning and building models: I found that a Neural network model is the most difficult to build. There are so many choices to build a new model and each model takes a long time to run. I stopped trying different structures when I found a Neural Network model that outperformed all the other models. A better Neural Network model could have been found if I had spent more time on testing new models. Tuning LightGBM and XGBoost was also somewhat challenging since there are many hyper-parameters, but I feel more confident with my current results.

Feature scaling: Scaling features was necessary for Ridge regression and Neural Network models, but also helpful for XGBoost and LightGBM models. It only did not help predictions of Random Forest models.

Feature Importances

I did some analysis on feature importances using the tree-based models, Random Forest, XGBoost, and LightBBM, whose feature importances can be conveniently obtained using feature_importances_ attribute.

XGBoost and LightGBM agreed the order of feature importances very well, but Random Forest had very different orders. Moreover, all features in the XGBoost and LightGBM models contributed the predictions to some extent while Random Forest had a few features that dominate the predictions and over a half of the features had almost no importance. It seems the better algorithms can learn how to utilize every features even with small importances rather than ignoring them. Moderately important features for XGBoost and LightGBM were those related to environment scores (derived from zipcode). The less important features for all three models agree and they are those related to sold months.High importance features for the XGBoost and LightGBM are described below with more details.

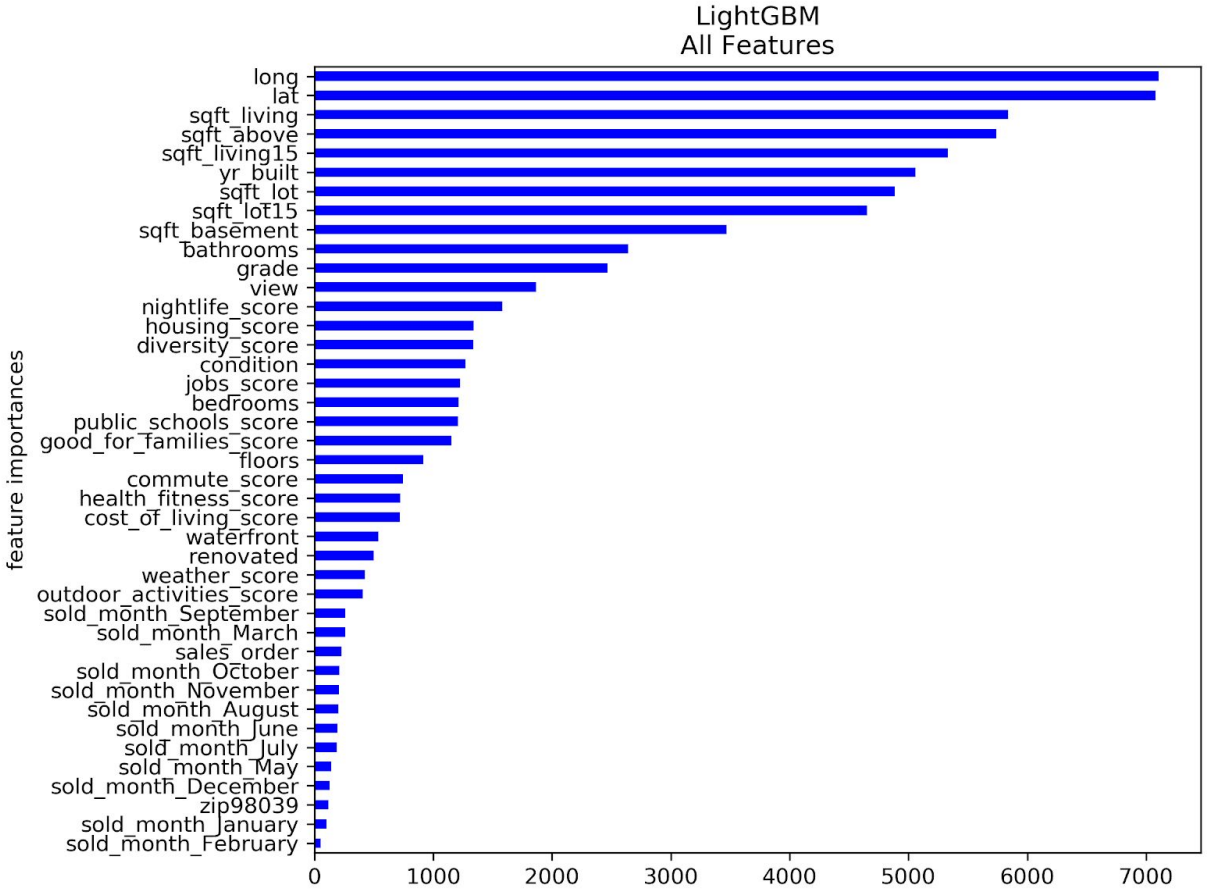


Figure. All feature importance for LightGBM

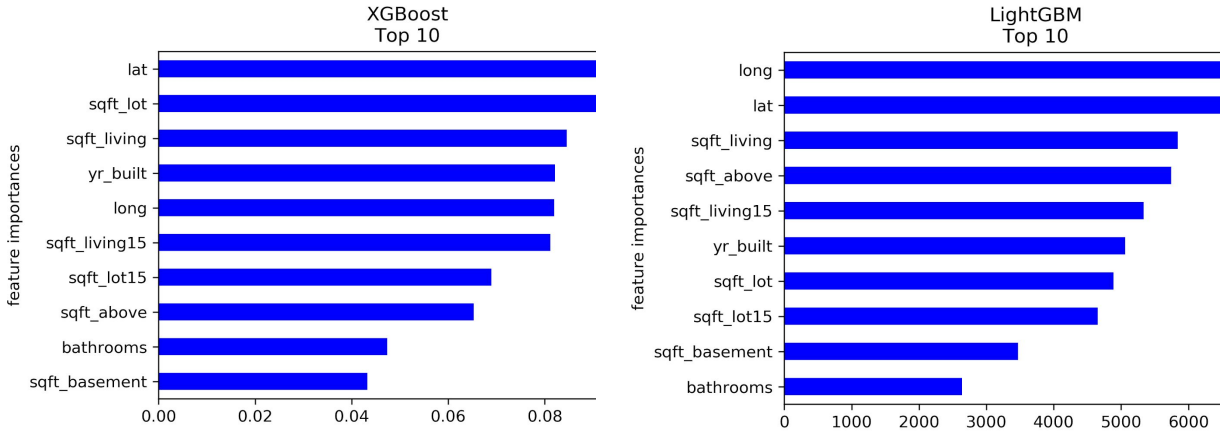


Figure. Top 10 features of XGBoost and LightGBM

The order of feature importances are different, but the top ten important features in the XGBoost and LightGBM models are exactly the same!!! Both models have latitude and longitude within the top 5. This is consistent to my finding in the price graph on the King County map; there are some regions where houses are more expensive. The expensive houses are mostly in the northwest region of King county and the inexpensive houses are in the south, east and also very top of King county. All features related to square footages of house and lot area are included in the top 10. The other two top 10 features are built years and the number of bathrooms. Since tree-based ensemble models are not linear models, these features might not have correlational directions. However, the EDA part showed square footages of living area, number of bathrooms and latitudes have moderate positive correlation with house prices. The EDA also showed a house grade has strong positive correlation with a house prices and grade is actually the 11th most important feature in the LightGBM model.

Discussion and future directions

Note that the order of feature importances should be carefully interpreted because there were multicollinearity which can make feature importances less reliable. For example, both sqft_living vs. sqft_living15 and sqft_lot vs. sqft_lot15 have correlation over .7. I have seen that the features sqft_lot and sqft_living have higher importances than latitude if sqft_lot15, sqft_living15, and sqft_basement are removed from the XGBoost model. I found removing them do not help my predictions in general, so I decided to keep them. More delicate feature selection methods might find removing them or some other features beneficial.

A dimension reduction method could have been applied to improve my predictions further. I tried Principal component analysis (PCA) with the best XGBoost model, but any number of components for PCA could not reduce the RMSE. However, tuning the number of components along with hyperparameters for XGBoost might be able to reduce errors. I also would like to try other types of feature projection method such as Non-negative matrix factorization. It would also be helpful for XGBoost and LightGBM to apply early stopping or graphically show a loss as a function of number of estimators to avoid overfitting.

Mean absolute error (MAE) could have been a better evaluation metric than RMSE because the dataset has many outliers which I do not want to penalize too much. However, I was puzzled a lot while using RMSE and learned a lot while trying to answer my questions.

Error analysis can be done to find out which features make those extreme outliers in the residual plots (below). The plots show XGBoost makes more outliers with residuals over 1 million or less than -1 million than LightGBM (6 vs. 2). However, the LightGBM model has one extremely big residual over 3 millions (this house turns out to be a waterfront house). The plots also show predictions of these models get worse if predicted prices become more extreme.

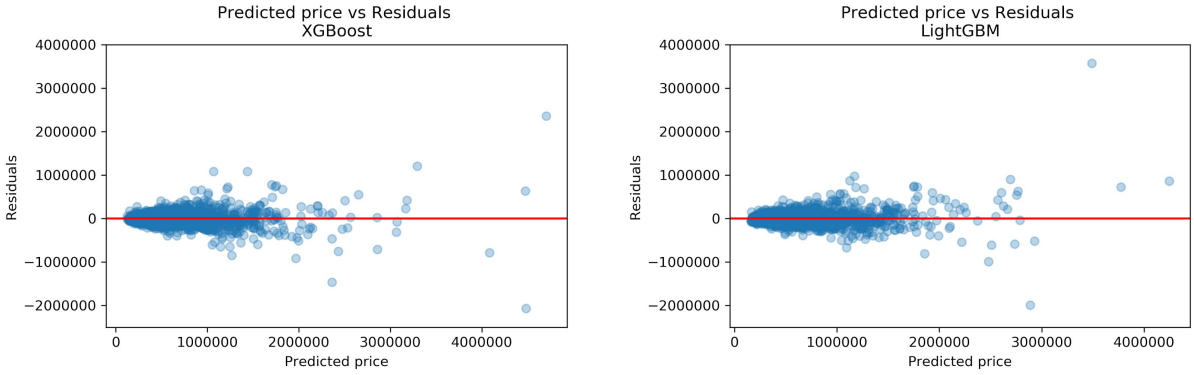


Figure. Predicted price vs. Residuals

5. Conclusion

Summary

The goal of this project was to collect data with house prices in King County, USA, and build a house price prediction model. The main dataset contained 21,613 records of house sales in King County. Each sale record consisted of 19 house features, house id numbers, dates sold. The house features include zip codes, which can give some useful information about a house location. The scores for public school, safety, cost of living, jobs, commute, and etc. were collected for each zip code from the website www.niche.com by web scraping. The collected data made another dataset and the two datasets were cleaned separately for missing values and outliers and then merged together.

In EDA, the scatter plots and correlation heat map showed that many independent features are correlated with house prices and other independent variables. The correlation coefficients found the most correlated features to house prices are house sizes (square footages), house grade and the number of bathrooms. The scores for families, public schools, and jobs, number of bedrooms and latitudes are also found to be moderately correlated with house prices. The houses colored with prices on the King county map revealed that expensive houses are mostly in the northwest region of King county. The price map also showed the extremely expensive houses are in waterfront or Medina,WA. The time series analysis showed house prices are higher between april and July and lower around January or February. I found an interesting v-shape pattern in the mean (median) house prices as a function of built years; newer houses tend to be more expensive for houses built after 1960, but for houses built before 1960 older houses tend to be more expensive. I found one factor causing the pattern could be those old houses built before 1940 concentrated in Seattle, which has higher house prices.

In the machine learning part, the data set was further prepared for machine learning by creating and removing some features, one-hot encoding, and splitting training-test sets. I found the top three models with high performance are Neural network, XGBoost, and LightGBM, which gave similar RMSE and r-squared around .89. The next best model was Random Forest with r-squared around .88 and Linear regression was the worst. Both XGBoost and LightGBM

models showed the most important features deciding house prices are the square footages of living area and lot, geographical locations (latitude and longitude), number of bathrooms and year built. Most of these features were expected to be important in a good house price prediction model from EDA.

Final Recommendation

Both EDA and machine learning showed that square footages of living area, number of bathrooms and latitudes are important features in predicting house prices and they have positive correlation with house prices. Some important features, but not linearly related to house prices, are sold months and year built. Time series analysis suggested it is better to buy a house in winter and sell a house around late spring or early summer price-wise. The plot for house prices vs. year built showed some v-shape pattern and it turns out built years are within the top 10 features for the best XGBoost and LightGBM models.

I have found the best house prediction models are the XGBoost and LightGBM models which showed high speed and best performance in RMSE. If one model should be selected I would recommend to use the LightGBM model since it is faster and it makes fewer outliers. Predicted price vs. Residuals plots (the last figure above) show XGBoost makes 6 outliers with residuals over 1 million or less than -1 million while LightGBM makes only 2. However, the LightGBM model has one extremely big residual over 3 millions. Thus, if avoiding such an extreme outlier is more important, I would recommend to use the XGBoost model. The plots also show predictions of these models get worse if predicted prices become more extreme. New house prices can be predicted using the already trained models. However, if a house price predicted by one of the models is extremely high (say, over 2 millions), one should be careful to use the price because the above residual graphs showed that errors get extreme for some of those houses.

More detailed comments and python codes are in the Jupyter notebooks in this depositories https://github.com/math470/Springboard_Capstone_Project_1.