Predicting house prices in King County, Washington, USA

By Panagiotis Petsas

This report is part of the course “Applied Data Science Capstone” from “IBM Data Science Professional Certificate” on Coursera.

Table of contents

1 Introduction

1.1 Background

King County is one of the most populated counties in USA, with an estimated population of more than 2.25 million people **[1]**. It is located in Washington, where Seattle is the county seat. It has an area of almost 6000 km^2, where 490 of them being covered by water. This county consists from more than 30 cities and a big variety of historical places **[2]**. Considering the importance of King County, it is of great interest to identify how house prices vary in this area. In addition, it would be important to identify the factors that drive these prices.

1.2 Problem

Houses are one of the most important type of tangible assets. They are the place where people can settle, or start their business. If someone owns multiple houses, it is possible that they will maintain one for settlement and use the remaining ones to produce economic value, either by renting them or by selling them. House prices might vary depending on many different factors. The house features (e.g. total area, amount of bedrooms or bathrooms) play a major role in pricing. Other important factors are the city/area where the house is located or its proximity from different venues, such as shops, restaurants, utilities or services. Social and economic factors might also take place, which means that a house's value might change at different time intervals.

Stakeholders might be aware of many factors that determine the house prices from their experience, but they might not be able to quantify the magnitude of each factor. For example, houses with more bathrooms and bedrooms might have a higher price, but one would not know which of the two increases the price more. It is therefore important to utilize house data in order to identify patterns between house features and house prices.

1.3 Interest

Real estate agents would be the most interested individuals in this subject. Gaining knowledge about the factors that drive the house prices would be beneficial, as they could identify spatial and temporal trends, evaluate houses in a more informed way and help their clients to sell their property, satisfying both parties. Even in the case where an individual wants to sell its house without an agent, this could provide useful information in order to set the right selling price. While this study will be held in King County, interested stakeholders can utilize the results for similar places or use the same methods to repeat the analysis for other places that might not be similar to this one.

2 Data acquisition and manipulation

2.1 Data sources

We obtained data on houses in King County from Kaggle **[3]**. This dataset contains more than 21.000 house records, providing information on features like price, year build, number of bedrooms, longitude, latitude, zip code etc. The list of features is provided in **Table 1**.

In order to reduce computation time, we decided to work only with a specific area. After some initial data cleaning (see below), we scraped the area phone code of each house by utilizing its zip code. We maintained records with area phone code 253. These houses are located in an area near Tacoma, a medium sized city (**Figure 1**). We used 'www.getzips.com' site in order to

acquire the area phone codes. This procedure reduced the house records to 2766, a more manageable size for modeling.

Figure 1 Houses in King County, Washington with area phone code as 253, marked with blue dots.

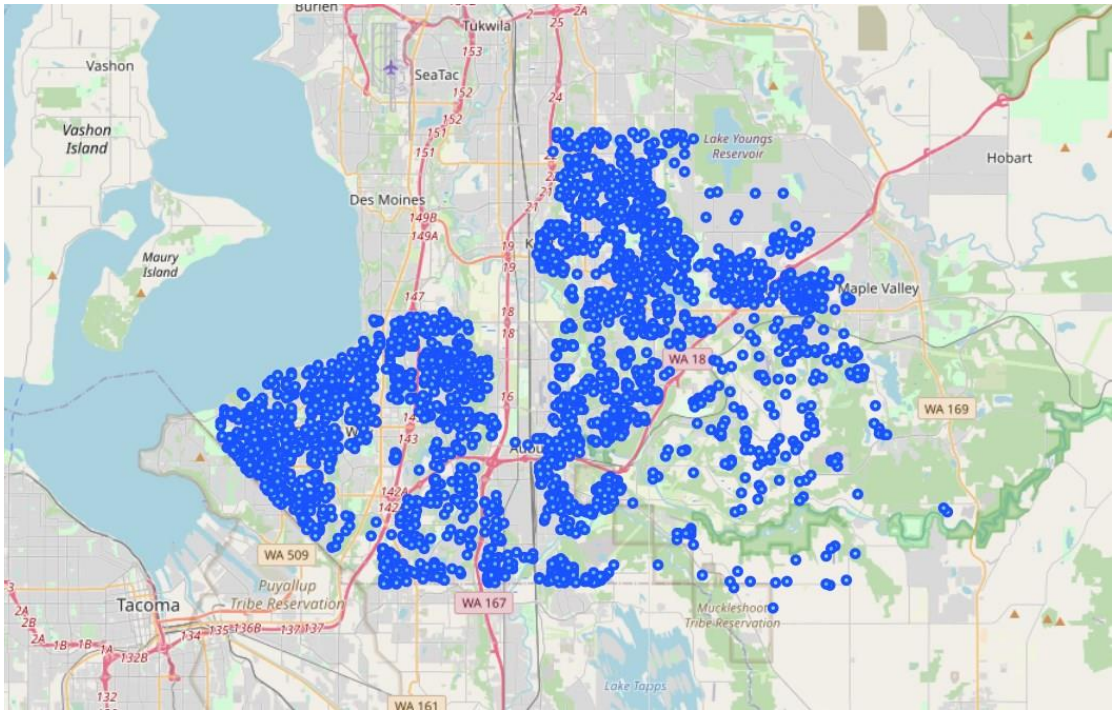


Table 1 Features in the dataset of houses in King County, Washington, along with their explanation.

Feature name	Explanation
ID	Unique id for each house record
Date	Date of home sale
Price	House price
Bedrooms	Number of Bedrooms
Bathrooms	Number of Bathrooms
Sqft_living	Apartment inter living space (in sq feet)
Sqft_lot	Lot space (in sq feet)
Floors	Number of floors
Waterfront	Binary (1 if the apartment overlooks the waterfront, 0 otherwise)
View	From 0-4 integer, a grade for house's view
Condition	From 1-5 integer, the condition of the house
Grade	From 1-13 integer, the construction and design level
Sqft_above	Apartment inter living space above ground (in sq feet)
Sqft_basement	Apartment inter living space below ground (in sq feet)
Yr_built	The year the house was built
Yr_renovated	Year of last renovation (0 if never renovated)
Zipcode	The zipcode of the area
Lat	Latitude
Long	Longitude
Sqft_living15	Apartment inter living space (in sq feet) for the 15 nearest neighboring houses
Sqft_lot15	Lot space (in sq feet) for the 15 nearest neighboring houses

Having the longitude and latitude of each house, we used Foursquare API **[4]** in order to extract the most popular venues in a proximity of 500 meters from it. We limited the API to the top 30 venues per house, in order to reduce computation time. This process derived a total of 8995 venues near the houses, providing information about their name, category and coordinates.

2.2 Data cleaning

The dataset did not contain any missing records. Some initial data cleaning was conducted by removing strange records (houses with zero bedrooms or bathrooms). The dataset contained a feature that indicated the year of renovation. Most houses were not renovated, thus they received a value of 0. For this reason, we created a new feature that indicated whether a house was renovated (1 if yes and 0 if not). After these adjustments, the next part of the cleaning process was to maintain the records with area phone code of 253.

The next part was to group the venue records based on their category. While Foursquare provides their own general categories, we preferred to classify the venues in our custom groups. We created 13 distinct groups (**Table 2**) to classify a total of 254 unique venue categories. The grouping was performed by searching on keywords within the venue category names. For example, venue category names including the word 'restaurant' would be put in the group of 'places to eat'. We ensured that categories that were grouped were removed from the grouping process, in order to avoid grouping them twice. There were some venue categories that could not be grouped based on keywords, which we grouped manually to the correct group.

In the next session, we summarized the amount of venues per group for each house. This process created 13 new features, one for each group, indicating the amount of venues of this group in the near proximity of each house. For example, a house with 2 places to eat and 3 venues related to services would get a value of 2 in the 'places to eat' feature, 3 in the 'venues related to services' features and 0 to the remaining 11 features related to venue groups.

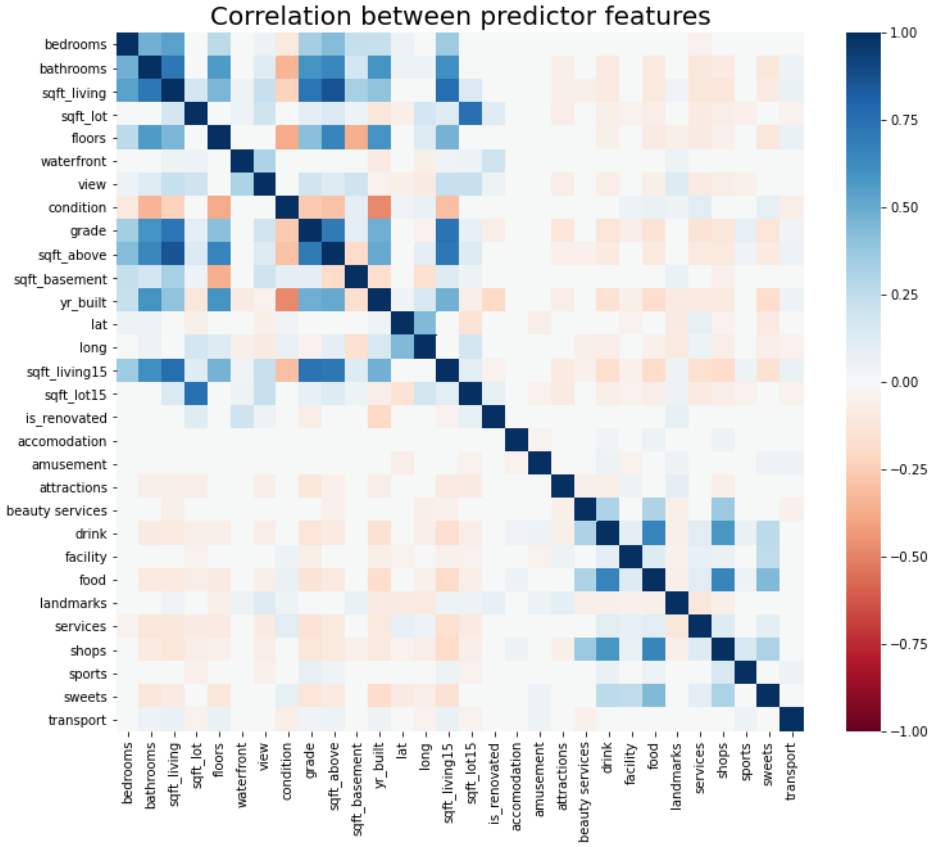
2.3 Feature selection

Since our purpose is to use features to predict the house price, we must maintain features that could provide such information. We initially discarded 'id' and 'date' (of recoding the house), as they are used for reference only. Similarly we discarded the 'zipcode' and 'area phone code' as they provide no further information. The year of renovation feature was replaced by the binary feature 'is renovated', as stated above.

Table 2 Groups of venue categories, along with some keywords used to identify the right venues.

Group	Some keywords used
Places to eat	Restaurant, food, pizza, burger, chicken, fish, breakfast, ...
Places to drink	Bar, wine, beer, pub, coffee, café, ...
Places with sweets	Candy, donut, dessert, pie, ice cream, ...
Amusement	Playground, arcade, casino, roller, night, ...
Facilities	Facility
Sports	Soccer, basketball, baseball, sport, gym, fit, ...
Transport	Airport, boat, ferry, bus, train, ...
Shops	Shop, market, store
Services	Service, assist, bank, doctor, police, ...
Beauty services	Spa, massage, salon, beauty, tattoo
Attractions	Art, history, attract, museum, ...
Landmarks	Mountain, lake, tree, ...
Accommodation	Hotel, motel, inn, rest, ...

Figure 2 Pearson correlation between the predictor features. The features 'sqft_above', 'grade', 'sqft_living15' and 'sqft_lot15' were removed from the analysis, as they caused high correlations with other features.



We performed Pearson correlation analysis (**Figure 2**) to identify linear patterns among the remaining features. We decided to discard four more features that caused high correlation, being 'above area in sqft' (after all, we maintained the 'total area in sqft', which is exactly equal to the summary of the 'above' and 'basement area in sqft'), 'grade' (which is a subjective value anyway), 'sqft_living15' and sqft_lot15' which indicate the total living and lot area of the 15 neighboring houses, and have high correlation with their respective features. With that procedure, we maintained 26 features for predicting the house prices.

3 Models for price prediction

3.1 Machine learning implementation

Machine learning algorithms share many similarities with statistics. Many statistical models, such as linear regression, logistic regression or decision trees are applied in machine learning algorithms. The main difference between machine learning and statistics is that in machine learning we want to utilize these models for prediction, while in statistics we want these models to describe the relationship between the predictor features and the target feature. That being said, a statistical model that can successfully describe these relationships might fail to predict correct values for new data that were not used in the model development.

This is where the concept of machine learning comes in. The data records are split into two sets; the training set and the test set. The training test will be used to develop the models. The test set will be utilized to determine how well the model performs on data that were not used in model development. We will use the test set predictor features to predict the target variable (in

our case, the house price), and then compare the predicted values to the actual values. The closer the two values are, the better the model can estimate the target variable (i.e. house price).

For our study purposes, we will use the 75% of the data records as the training set, and the remaining ones for the test set.

3.2 Machine learning algorithms

Since our problem requires to predict a value in a continuous scale, the initial thought is to create linear models. We can use these models to identify a linear relationship between the predictor features and the target feature. In order to capture the non-linear relationships between the predictor features and the house prices, we will develop a polynomial linear regression model, with maximum degree of two. Since we have 26 features, this process will create 377 features (i.e. 26 features for the initial features, 26 features for the squares of the initial features, and 325 features for the products of pairs of the initial features). We will not try higher degrees as it will create a big amount of features that would demand a lot of computation time and could probably cause overfitting, making the model successful to training data but insufficient to newer data. We will also try a ridge regression model.

In addition to these models, we will try two tree based algorithms; random forest algorithm and gradient boosting algorithm. Decision tree algorithms create a tree which splits the data depending on a feature. After several steps of branching, they derive a result. On the contrary, random forest and gradient boosting algorithms work with multiple decision trees instead of incorporating just one. They use the results derived by these trees and provide an average result from all of them. Random forests create decision trees independently and derive a result at the end of the process, while gradient boosting algorithms build one decision tree at the time, and combine the results along the way.

3.3 Permutation importance

Initially, we will test the models with all the available features. But in order to highlight the most important features, we will perform permutation importance. This method helps us identify how important each feature is for predicting the house prices. After we have fit the model, we use the test set to predict the target feature. For the selected feature we want to examine, we shuffle the records in its column, changing it entirely. Then we predict the target feature again. We repeat this process multiple times (e.g. 1000 times) and we note the mean difference between the actual prediction and the predictions with the shuffled records. If the difference is small, that means that this feature is not that important, as its alteration doesn't change the predictions that much. But if the difference is big, that means that this feature is really important, as its alteration changes the predictions. We will identify the most important features for each model and re-run the algorithm with it.

3.4 Evaluating the models

In order to evaluate how well our model performs on new data, we will compare the predicted values $\widehat{y_i}$ with the actual house price values y_i. For this purpose, we will use the next indices:

> Mean squared error: We will summarize the square of the differences between the predicted value and the actual value for each record, then get a mean value of them. The lower the value, the better the model performance. This index is derived by the type:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\widehat{y_i} - y_i)^2$$

> Root of mean squared error: It is simple the square root of mean squared error, which helps to lower the value and aid interpretation:

$$RMSE = \sqrt{MSE}$$

> Mean absolute error: Similar to mean squared error, but instead of squaring the values, we extract the absolute value of them:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |\widehat{y_i} - y_i|$$

> R squared: This index is based on the sum of the squared residuals e_i^2 (i.e. the difference between actual and predicted values), as well as the sum of the squared differences between the real values y_i and their mean value $\bar{y}$. This index will be used in both training and test sets. High values indicate that the models performs well, but extremely high values for the train test might suggest overfitting:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} e_i^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

> Cross validation: In addition of using the initial train-test splitting, we will split the dataset into four parts, fit and test the model four times, with each part being used for testing each time. In that way, we will use the entire dataset for testing. From this process, we will extract the mean R squared value from all four iterations.

The aforementioned analysis was performed in Python 3.6, using the 'pandas', 'numpy', 'scipy', 'matplotlib', 'seaborn', 'sklearn' and 'xgboost' modules.

4 Results

4.1 An inspection on the indices

The linear model that included all features had a relatively good performance with R^2 around 0.76 (**Table 3**). From permutation performance, inspecting the top 15 important features, we found that the majority of them referred to the house itself and not its surrounding venues (**Table 4**). The linear model with the top 15 features performed better, but the one with only the top 10 performed worse than the original.

The polynomial feature model performed worse than the initial linear model. All the errors have higher values compared to the linear model. Its R^2 has a lower value for the test set, but a higher value for the training set, something that implies that this model might be prone to overfitting (**Table 3**). Even when inspecting the top 15 features from permutation importance (**Table 4**), we can see that their impact is much bigger compared to the ones in linear model, and that their coefficients are much higher, meaning that some features might need to negate the impact of other features. The models with the top 350 and 300 top features out of the 377 possible did not perform any better. On the contrary, a polynomial model that was developed by the top 15 features of the linear model had significantly better results, but it was still outclassed from the linear models.

Ridge regression analysis had similar results to the linear models (**Table 3**). Even the top 15 features are identical to the ones from the linear model, with little changes to the importance score and the coefficients (**Table 4**).

Table 3 Report for the model performance. The models are linear regression (LR), polynomial regression with degree of 2 (PR-2), Ridge linear regression (RLR), random forest (RF) and gradient boosting (XGB). In details, the amount of features used is described (either all or the top portion of the model), along with other parameters (alpha for RLR, number of trees (n_est) for RF and XGB, learning rate (lr) for XGB). The indices provided are root mean squared error (RMSE), mean absolute error (MAE), R^2 for test and train set, along with the mean and standard deviation of R^2 from cross validation.

ID	Model	Details	RMSE	MAE	R^2(test)	R^2 (train)	CV mean	CV std
000	LR	All	50303,4651	34409,7484	0,7629	0,7519	0,7159	0,0000
001	LR	Top 15	50181,8816	34180,5789	0,7640	0,7498	0,7154	0,0000
002	LR	Top 10	50812,3709	34575,3377	0,7581	0,7468	0,7114	0,0000
003	PR-2	All	54054,9112	36326,2647	0,7262	0,8590	0,5545	0,0000
004	PR-2	Top 350	54158,2087	35644,8392	0,7251	0,8498	0,6259	0,0000
005	PR-2	Top 300	95705,1103	38884,7937	0,1417	0,8476	0,6495	0,0000
006	PR-2	Top 15 of LR (All)	51537,7138	33437,2147	0,7511	0,8249	0,7060	0,0000
007	RLR	All, a=0.1	50323,4252	34407,0399	0,7627	0,7519	0,7210	0,0558
008	RLR	Top 15, a=0.1	50196,3413	34176,7613	0,7639	0,7498	0,7205	0,0000
009	RLR	Top 10, a=0.1	50826,9021	34571,7113	0,7579	0,7468	0,7167	0,0000
010	RF	All, n_est = 50	49032,3400	32053,3813	0,7747	0,9590	0,7320	0,0293
011	RF	All, n_est = 100	48816,3847	31896,0149	0,7767	0,9596	0,7397	0,0324
012	RF	All, n_est = 200	48661,5101	31676,3119	0,7781	0,9606	0,7408	0,0316
013	RF	All, n_est = 300	48494,3229	31503,6009	0,7796	0,9611	0,7392	0,0321
014	RF	All, n_est = 400	48414,1023	31516,2945	0,7804	0,9613	0,7386	0,0311
015	RF	All, n_est = 500	48315,5904	31484,8109	0,7812	0,9614	0,7384	0,0314
016	XGB-1	All, n_est = 300, lr=0.1	44935,7156	30271,5383	0,8108	0,9137	0,7356	0,0517
017	XGB-2	All, n_est = 400, lr=0.1	44818,8982	30322,5732	0,8118	0,9268	0,7349	0,0513
018	XGB-3	All, n_est = 500, lr=0.1	44791,3097	30332,7105	0,8120	0,9352	0,7328	0,0510
019	XGB-4	All, n_est = 300, lr=0.05	45057,3886	30526,5902	0,8098	0,8809	0,7337	0,0480
020	XGB-5	All, n_est = 400, lr=0.05	44651,0750	30314,8434	0,8132	0,8933	0,7352	0,0502
021	XGB-6	All, n_est = 500, lr=0.05	44583,7157	30240,5342	0,8137	0,9035	0,7354	0,0524
022	XGB-6’	Top 21 from XGB 6	44928,8451	30238,2901	0,8108	0,9035	0,7410	0,0543

Tree-based models provided a significant improvement in the predictions. The RMSE decreased to 48000-49000 units, while the MAE decreased to 31000-32000 (**Table 3**). In addition, the R^2 on test data and the CV mean improved. The R^2 on the training test is significantly high (~0.96). A slightly better performance is observed as the number of trees increases. The gradient boosting algorithms provided even better results compared to the random forest models, with errors decreasing as trees increased. The model with a learning rate of 0.05 performed better compared to the one with 0.1. When comparing to the random forest models, R^2 value is increased on the test set, as well as it is lowered on the training set, something that implies better performance and less danger of overfitting. Since the gradient boosting models happened to perform better than the random forest models, we performed permutation importance on the best gradient boosting model (500 trees and learning rate of 0.05). The top 15 features are similar to the ones in linear models, but the venue groups that appear to be important are the shops and the landmarks, instead of places with food, sweet and services (**Table 5**). We created one last gradient boosting model with the top 21 features derived from permutation importance, keeping the number of trees and learning rate the same.

Table 4 Permutation importance value of the top 15 features per linear model that was fit with all the features (linear regression and ridge linear regression share the same features with the same ranking), along with their respective coefficients.

	Linear regression		Ridge linear regression			Polynomial reg. (d=2)	
	Importance	*Coefficient*	*Importance*	*Coefficient*		*Importance*	*Coefficient*
sqft_living	1,1283	111,55	1,1277	111,52	**view**	107492,6	-59055946,0
sqft_lot	0,0826	0,75	0,0822	0,75	**services**	44007,8	-15812379,9
sqft_basement	0,0336	-36,85	0,0334	-36,76	**lat**	38856,7	-346994991,9
view	0,0328	28801,77	0,0330	28839,34	**view*long**	28939,2	-250559,7
yr_built	0,0276	541,68	0,0274	538,82	**long**	25792,1	130105453,5
bedrooms	0,0159	-10765,37	0,0158	-10763,91	**view*lat**	21275,3	555215,7
lat	0,0152	220268,54	0,0145	212186,80	**amusement**	16143,2	22496184,5
waterfront	0,0078	99111,38	0,0076	95868,34	**lat*long**	15611,1	-1805800,1
condition	0,0060	9787,40	0,0060	9759,64	**sqft_living*long**	12937,4	98,6
bathrooms	0,0047	7747,38	0,0047	7810,90	**long*services**	11678,8	-66661,8
floors	0,0040	-9659,77	0,0040	-9638,05	**sqft_living**	9879,0	10526,5
is_renovated	0,0032	32812,63	0,0032	32869,56	**lat*services**	8256,2	144594,3
food	0,0020	-1148,73	0,0020	-1147,04	**shops**	7933,5	-4579649,8
sweets	0,0009	-3361,89	0,0009	-3413,97	**long*amusement**	7810,0	127992,1
services	0,0008	-2173,13	0,0008	-2155,89	**transport**	7671,0	-15427342,0

Table 5 Permutation importance value of the top 15 features of the gradient boosting model (500 trees, learning rate of 0.05).

Features 1-5		Features 6-10		Features 11-15	
sqft_living	0,8218	**long**	0,0196	**shops**	0,0030
sqft_lot	0,1932	**bathrooms**	0,0163	**landmarks**	0,0023
yr_built	0,0910	**sqft_basement**	0,0141	**is_renovated**	0,0022
lat	0,0481	**condition**	0,0082	**floors**	0,0022
view	0,0279	**bedrooms**	0,0067	**waterfront**	0,0005

4.2 A visual inspection on the predictions

Another way to identify how well our models perform is to visually inspect the actual values versus the predicted values. In **Figure 3** we can inspect the distribution plots between the actual house price values and the predicted ones from four different models. We projected this difference with four models; one linear, one polynomial, one ridge linear and one of gradient boosting. The first three models are created with the top 15 features of linear and ridge linear models, while the last was created with the top 21 features of the better performing gradient boosting model (500 trees, learning rate of 0.05).

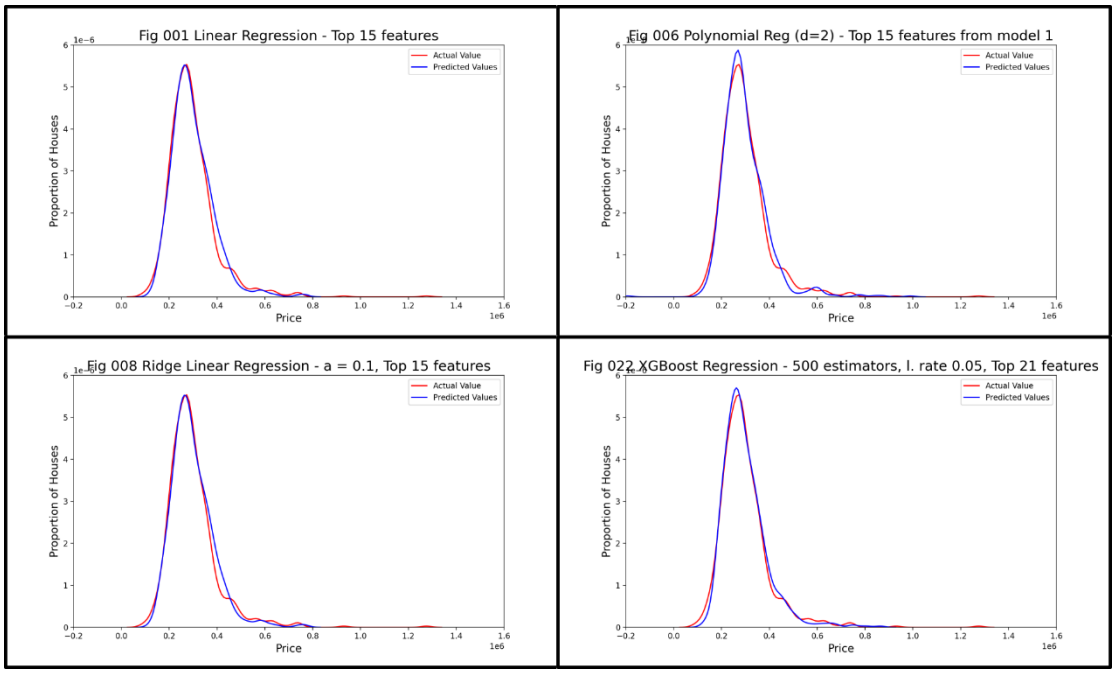
As shown by the red line, the majority of the house prices are around 300.000 units. This peak is better described by the linear and ridge linear model (top left and bottom left), as the fitted blue line almost falls on the red one. On the contrary, these models do not provide a good estimation for the values of 400.000 units and onwards. As we can inspect in the figure, when the distribution starts to descent, the blue line overestimates the houses with such price, and underestimates them after the values of 500.000 units. In addition, these models fail to capture house prices that are above 800.000 units.

The polynomial model does not seem to sufficient predict the house price values (top right). Initially, the peak of the distribution graph shows that it predicts more values around 400.000 units than it should do. Another red flag is that it is possible to predict negative values for the house prices, something that does not make sense in real world. Last but not least, it does not

fit the red line well in the region of 500.000 units, underestimating the amount of houses with that value.

The gradient boosting model seems to provide the better fit of all four models (bottom right). While it does not fit the peak of the curve as well as the linear and ridge linear models do, it provides a better overall fit in all regions, as well as it is able to detect higher house price values, something that the linear and ridge linear models failed to do.

Figure 3 Distribution plot between the actual house price values and the predicted house price values derived from the models (linear model, id: 001, top left; polynomial, id: 006, top right; ridge model, id: 008, bottom left; gradient boosting, id: 022, bottom right).



5 Discussion

This analysis provided the performance of different models in terms of predicting house prices based on house and neighborhood features. It is shown that in the present problem, tree-based models performed better than linear based models. This outcome should not come as a surprise, as the majority of the predictor features had low Pearson correlation with price, indicating poor linear relationship (**Table 6**). Polynomial transformation did not provide any better results, especially when all features where transformed and used for the model fit. They provided poor predictions and since they had a high R^2 on the training data records, they seem to be prone to overfitting. It is worth noting that this procedure provides better results when less features are used, as we did by providing only the top 15 features from the linear model based on permutation importance.

Tree-based algorithms were extremely useful to identify the non-linear relationships between the predictor features and the house prices. They have their own parameters to adjust and derive different results. This parameter tuning provided multiple models, with continuous improvement in each model. In the present study, random forest models were outclassed by the gradient boosting models. While they provided low error values and high R^2 value on the test data, their high R^2 value on the training data suggest that they might be prone to overfitting.

On the contrary, gradient boosting models derived better predictions with less R^2 value on the training set.

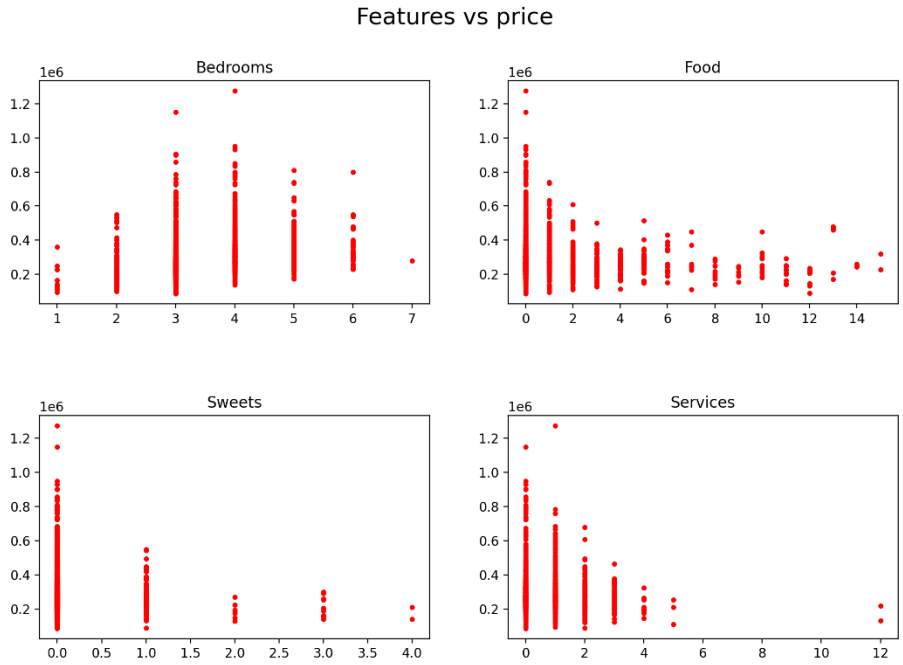
One of the most critical outcomes of this study is that we were able to identify the most important features that affect the house prices. The linear, ridge linear and the gradient boosting models share a lot of common features in their top 15 ranking. It comes as no surprise that the total area is the most important factor. As we can see in **Table 4**, each square feet unit increases the price by about 111.55 price units. Features such as lot area and basement area also play a major role in house pricing. The linear models suggest that for each square feet unit in the basement, the house price drops by around 36 units, which is something that one would not expect. On the contrary, 2035 out of 2766 (~73.6%) of the houses have no basement, therefore their zero value on this feature might have drove the price down.

In these two models, we will inspect some other features with a negative coefficient. That is, the bedrooms or the three venue groups; food, sweets and services. These features have a really low Pearson correlation with the price and this negative coefficient does not exactly describe the relationship between the features and the price (**Figure 4**).

Table 6 Summary of absolute Pearson correlation values.

	Pearson cor. (abs. value)
Mean	0.187
St. Dev.	0.192
Minimum	0.011
Quantile 25%	0.057
Median	0.121
Quantile 75%	0.293
Maximum	0.802

Figure 4 Comparison between four features (number of bedrooms, number of food venues, number of sweet shops and number of services venues) and the respective house price.



The gradient boosting model found similar features as important. The only difference is that the venue groups that were identified as more important were the shops and landmarks. Landmarks refer mostly to nature places such as mountain, lake or river, meaning that house positioning closer to nature affects the price. Lastly, the polynomial model had eight single degree features and seven product features. Common features appearing are latitude, longitude, total living area (sqft), as well as some venue groups, such as shops, services transport and amusement. Since the model has 377 features, feature importance derives a higher magnitude in importance values.

One must keep in mind that the modeling is directly affected by the predictor features, meaning that the addition of other features could improve the model predictions. Given the necessary hardware, one could scrape more venues per house, providing a more informed value about the corresponding features. Despite this fact, acquiring the 30 most important venues per house was sufficient in providing insights about house prices. Other useful features that one could use are features related to the house itself, such as type of heating (e.g. gas, oil or electric power) and available garage, or other spatial - temporal information, such as mean temperature per year or elevation. Nevertheless, the opinion of the experts on this subject, such as real estate agents, can be very helpful in order to identify the correct features and avoid spending precious time with the ones that do not provide any information.

6 Conclusion

In this capstone project, we constructed machine learning models in order to predict house prices, as well as we identified the most critical features that affect the house pricing. We utilized a dataset of house records, applied methods in order to refine it, and implemented new features that correspond to the venues in the house proximity. We tested various machine learning models, concluding that tree-based models performed better, as the data had many non-linear relationships with the house prices. The findings of this project can be utilized from real estate agents, individuals who would like to sell their house, as well as individual who would like to buy a new house.

While this study was conducted in King County, Washington, the results may be applied to other regions with similar characteristics. Given the necessary hardware and data, this project can be further developed for houses in a broader spatial scale, such as an entire state or country. As house transactions happen every day, the stakeholders should gather continuously data on these transactions in order to identify patterns between house features and prices, getting the upper hand in the real estate market.

References

[1] Most populated USA cities

(https://en.wikipedia.org/wiki/List_of_the_most_populous_counties_in_the_United_States)

[2] King County, Washington – Wikipedia

(https://en.wikipedia.org/wiki/King_County,_Washington)

[3] House prices in KC, Washington – Kaggle

(https://www.kaggle.com/harlfoxem/housesalesprediction)

[4] Foursquare API (https://developer.foursquare.com/)