STAT596: Regression & Time Series Analysis

Final Project Report: House Price Prediction

Team: 8

Name:	Harshini Bonam	Usha Kiran Bellam	Venkata Datla	Yaniv Bronshtein
NetID:	sdb202	ub39	vkd20	yb262

Abstract

Having shelter is a necessity to human living, and owning a house is a privilege to many. House prices depend on various factors and current market price is a major contributor that affects house sales. This project aims to build a machine learning model that can predict house prices so that people buying and selling can make an informed decision with minimal to no loss during the purchase and sale.

Introduction

If a person needs to purchase/sell the house they should be aware of the market prices to make a decision on when to buy or sell it, and they should never incur in loss while purchasing or selling the home in accordance with the market rate. So, using regression algorithm we are attributing the price to different components of the house, and we are using this importance of the components back to predict the price of new houses.

Following are the features we have in the dataset which we will be using to predict the house price:

1. id: a notation for a house
2. date: Date house was sold
3. price: Price is prediction target
4. bedrooms: Number of Bedrooms/House
5. bathrooms: Number of bathrooms/bedrooms
6. sqft_living: square footage of the home
7. sqft_lot: square footage of the lot
8. floors: Total floors (levels) in house
9. waterfront: House which has a view to a waterfront
10. view: Has been viewed
11. condition: How good the condition is (Overall)
12. grade: grade given to the housing unit, based on grading system
13. sqft_above: square footage of house apart from basement
14. sqft_basement: square footage of the basement
15. yr_built: Built Year
16. yr_renovated: Year when house was renovated
17. zipcode: zip
18. lat: Latitude coordinate
19. long: Longitude coordinate
20. sqft_living15: Living room area in 2015(implies-- some renovations) This might or might not have affected the lot size area
21. sqft_lot15: lot size area in 2015(implies-- some renovations)

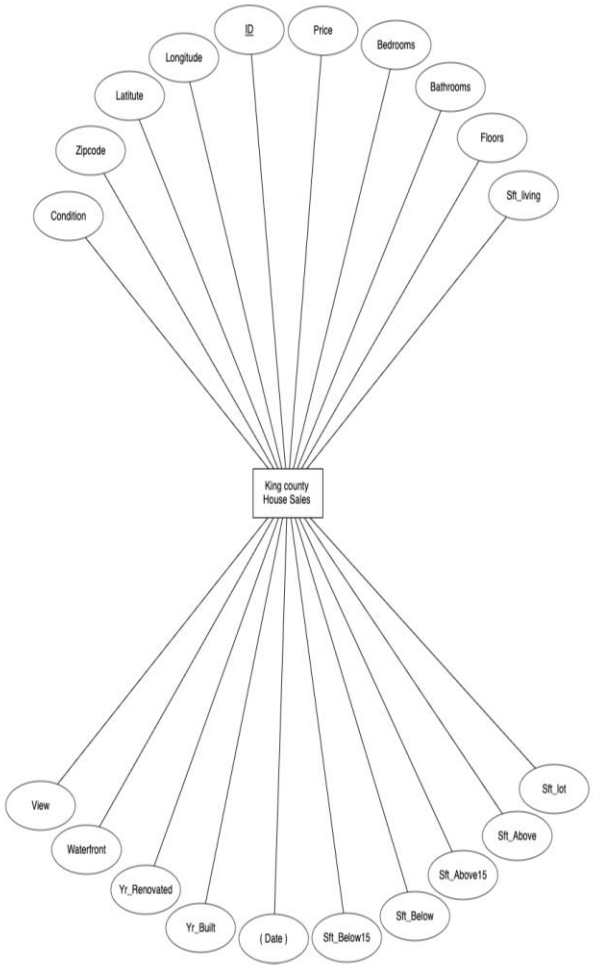
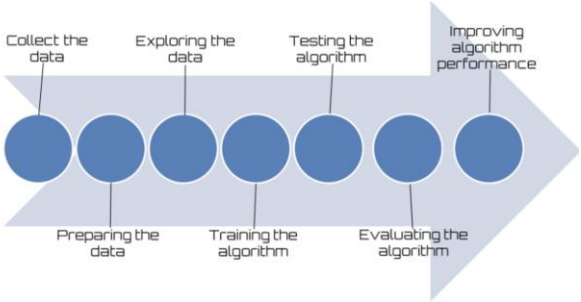


Fig. 1: The Entity Relationship model diagram for the King County House sale data.

Methodology:

As a part of Machine learning process, followed below methodology:



After exploring the dataset, it was observed in total we found 21613 rows and 21 columns then performed following analysis:

1. No null data was found.

2. The maximum number of bedrooms in a house are 33. So, we might want to look at that record and check if it is an outlier.

3. one floor houses are the most common type of houses sold

4. Very few houses have view to a waterfront and these houses might be costly

5. House condition is rated from 1 to 5 and the most common rating is 3

6. House grade is rated from 1 to 13 and the most common rating is 7.

Univariate analysis:

1. **Number of houses sold based on number of bedrooms**

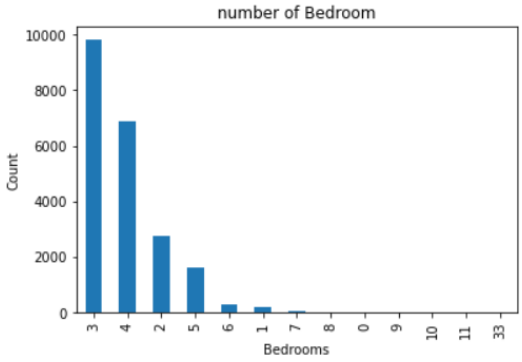


Fig. 2: Frequency bar chart on number of bedrooms to identify outliers.

The most common type of house sold are the ones with three bedrooms as shown in Fig. 2. So, this is helpful as we can understand 3-bedroom houses have more demand followed by 4 bedrooms. So, the number of bedrooms

seem to be an important factor while fixating the house price.

2. **To check if location latitude and longitude have an impact on number of houses sold.**

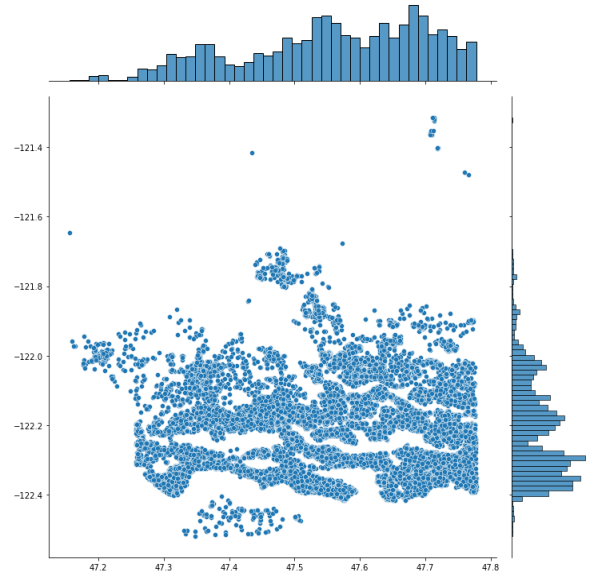


Fig. 3: Plot of area view with Latitude-Longitude over x-y axes.

We see in Fig. 3 a large number of houses between latitudes 47.5 and 47.8 and in terms of longitude there are large number of houses between -122.2 and -122.4. This location might be the ideal location for people to live and house prices might vary based up on this latitude and longitude.

3.Bivariate analysis:

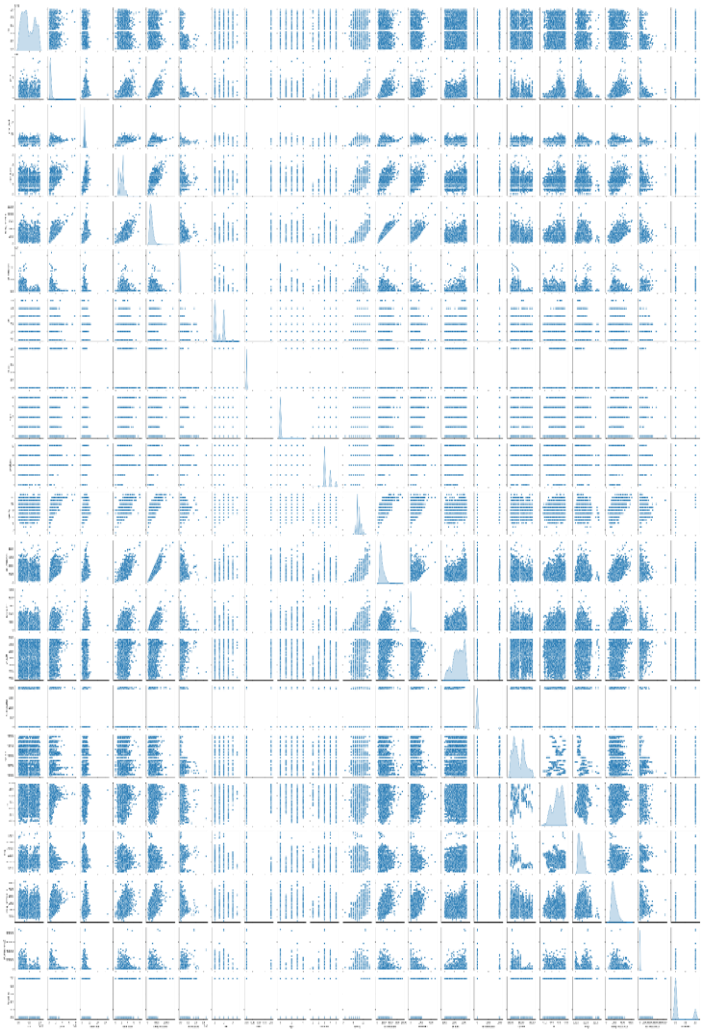


Fig. 4: Pair plot to see the distributions of variables and their correlation with other variables

4.Heat Map:

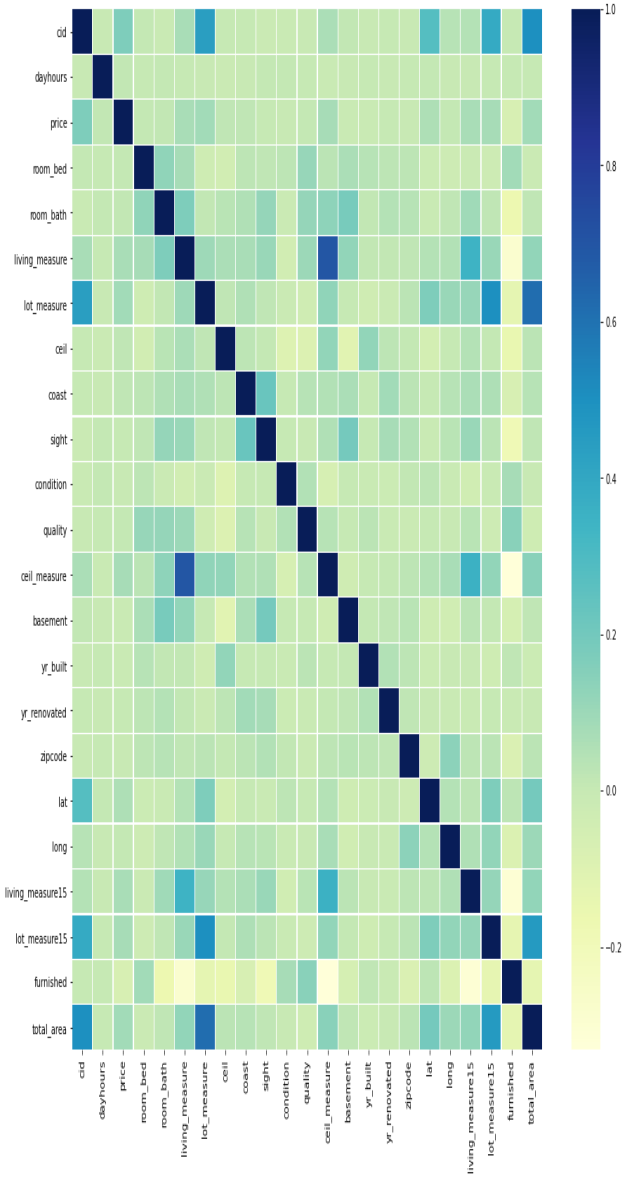


Fig. 5: Heat map to identify highly correlated variables

5. Observations from bivariate analysis:

1. sqft_living, bathrooms, grade, sqft_above, sqft_living15 are the metrics which have high correlation to the target variable price. So, these metrics might pop up as important metrics from the regression exercise

2. Also few of the above highlighted variables are also correlated with each other because of which only few of them might have significant impact on price because of multi-collinearity

6.Train and test split

Considered 70% of data as train and 30% of data as test data

7. Modeling

Since this is a price estimation project, we would like to use different regression techniques for estimation the price. The following are the regression techniques we tried.

a. Linear regression
b. Ridge regression
c. Lasso regression
d. Decision tree regressor & Decision tree regressor with tuning using Grid-Search
e. Random forest regressor & Random forest regressor with tuning using Grid-Search
f. Gradient boosting regressor

a. Linear regression:

Linear regression is the simplest and most widely used statistical technique for predictive modeling. It basically gives us an equation, where we have our features as independent variables, on which our target variable [price in our case] is dependent upon.

So, what does the equation look like? Linear regression equation looks like this:

$$Y = \theta_1 X_1 + \theta_2 X_2 + \dots \theta_n X_n$$

Here, we have Y as our dependent variable (price), X's are the independent variables and all b's are the coefficients. Coefficients are basically the weights assigned to the features, based on their importance.

R square:

It determines how much of the total variation in Y (dependent variable) is explained by the variation in X (independent variable). Mathematically, it can be written as:

$$R - Square = 1 - \frac{\sum(Y_{actual} - Y_{predicted})^2}{\sum(Y_{actual} - Y_{mean})^2}$$

R-square value obtained:

The R-square value that we achieved with linear regression is 70%

b. Ridge regression:

Till now our idea was to basically minimize the cost function, such that values predicted are much closer to the desired result.

Now, looking back again at the cost function for ridge regression.

$$\min\left(\left|\left|Y - X(\theta)\right|\right|_2^2 + \lambda||\theta||_2^2\right)$$

Here if you notice, we come across an extra term, which is known as the penalty term. λ given here, is actually denoted by alpha parameter in the ridge function. So, by changing the values of alpha, we are basically controlling the penalty term. Higher the values of alpha, bigger is the penalty and therefore the magnitude of coefficients are reduced.

R-square value obtained:

The R-square value that we achieved with Ridge regression is 70%.

c. Lasso regression:

In lasso, we can see that even at small values of alpha, the magnitude of coefficients has reduced a lot.

We can see that as we increased the value of alpha, coefficients were approaching towards zero, but if you see in case of lasso, even at smaller alpha's, our coefficients are reducing to absolute zeroes. Therefore, lasso selects the only some feature while reduces the coefficients of others to zero. This property is known as feature selection and which is absent in case of ridge.

Mathematics behind lasso regression is quiet similar to that of ridge only difference being instead of adding squares of theta, we will add absolute value of Θ.

$$\min\left(\left|\left|Y - X\theta\right|\right|_2^2 + \lambda||\theta||_1\right)$$

Here too, λ is the hypermeter, whose value is equal to the alpha in the Lasso function.

R-square value obtained:

The R-square value that we achieved with Lasso regression is 70%. In Lasso not many coefficients are going to zero implying that there is predictive power in many different

variables and not concentrated in few variables.

d. Decision tree regressor and decision tree regressor with tuning using Grid-Search:

Decision tree regression observes features of an object and trains a model in the structure of a tree to predict data in the future to produce meaningful continuous output. Continuous output means that the output/result is not discrete, i.e., it is not represented just by a discrete, known set of numbers or values.

R-square value obtained:

Using decision tree we are able to improve the Rsquare value to 72%. Then we tried decision tree with hyperparameter tuning by using Gridsearch and were able to improve the Rsquare value to 75%.

e. Random Forest regressor and random forest regressor with tuning using grid search:

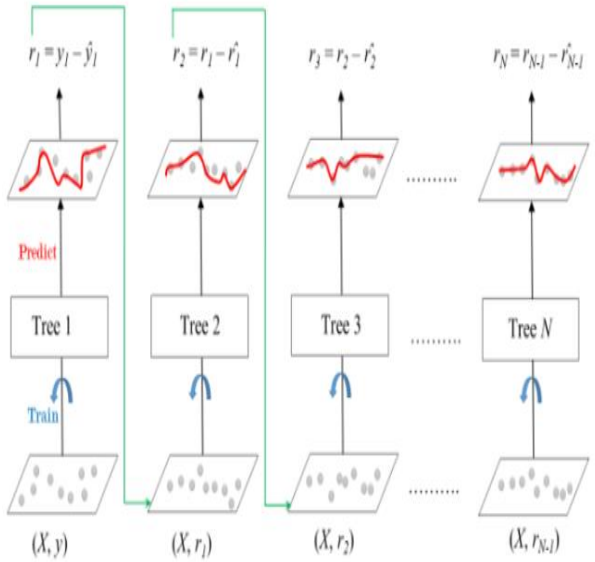
Random Forest has multiple decision trees as base learning models. We randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap.

R-square value obtained:

Using random forest regressor we are able to improve the Rsquare value to 88%. Then we tried random forest regressor with hyperparameter tuning by using Gridsearch and found no significant improvement from 88%.

f. Gradient boosting regressor:

Gradient Boosting is a popular boosting algorithm. In gradient boosting, each predictor corrects its predecessor's error. In contrast to Adaboost, the weights of the training instances are not tweaked, instead, each predictor is trained using the residual errors of predecessor as labels.
There is a technique called the Gradient Boosted Trees whose base learner is CART (Classification and Regression Trees).
The below diagram explains how gradient boosted trees are trained for regression problems.



R-square value obtained:

Finally, we tried gradient boosting regressor and were able to achieve R square value of 90%

8. Evaluation of different models (Comparison of different models and performance tuning)

1. Linear regression, Lasso regression, Ridge regression, Decision tree regressor, Hyperparameter tuned decision tree regressor, Random Forest regressor, Hyperparameter tuned random forest regressor and Gradient boosting regressor are the

different models we tried for this problem.

2. The R-square value that we achieved with linear regression is 70%
3. To improve this, we tried Lasso and Ridge regression, however in both these cases the R-square value obtained was 70%. No improvement in accuracy with respect to linear regression.
4. In Lasso not many coefficients are going to zero implying that there is predictive power in many different variables and not concentrated in few variables.
5. Then we tried decision tree and we were able to improve the R-square value to 72%.
6. Then we tried decision tree with hyperparameter tuning by using Grid-search and were able to improve the R-square value to 75%.
7. Then we tried random forest regressor and was able to increase the R-square to 88%
8. Then we tried random forest with hyperparameter tuning by using Grid-search and found no significant improvement from 88%.
9. Finally, we tried gradient boosting regressor and were able to achieve R square value of 90%.
10. Using grid search in our models we varied different hyperparameters and obtained the optimal hyperparameters**.**

Method	**Test - R^2**
Multiple Linear Regression	70%
Ridge Regression	70%
Lasso Regression	70%
Decision Tree Regression	72%
Decision Tree Regression with Hyperparameter tuning (max_depth: 10)	75%
Random Forest Regression	88%
Random Forest Regression with Hyperparameter tuning (max_depth: 20, n_estimators: 220)	88%
Gradient Boost	90%

9. Conclusion

1. To predict the prices of houses we first analyzed different independent variables by seeing the correlation of these variables with the target variable price
2. We also looked at the multi-collinearity of this independent variables
3. We then applied multi variate linear regression and achieved a test accuracy of 70%
4. We then applied a large group of models and the best accuracy obtained from those models is from gradient boosting regressor which is 90%
5. It is once again observed that ensemble techniques help us in achieving high accuracy either it is regression or classification as it involves a large group model and best of these models is taken.

10. References

- Dataset: https://www.kaggle.com/harlfoxem/housesalesprediction
- MLR python library https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
- Lasso python library https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html
- Ridge python library https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
- Decision Tree regression python library https://scikit-learn.org/stable/auto_examples/tree/plot_tree_regression.html
- Random Forest regression python library https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
- Gradient Boosting regression python library https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html