Predict House Price

Capstone Project 2 Milestone Report

Introduction

The price of a house is dependent on various factors like size or area, how many bedrooms, location, the price of other houses, and many other factors. Real estate investors would like to find out the actual cost of the house in order to buy and sell real estate properties. They will lose money when they pay more than the current market cost of the house and when they sell for less than current market cost. The banks also want to find the current market price for the house, when they use someone's house as collateral for loans. Sometimes loan applicant overvalues their house to borrow the maximum loan from the bank. Banks and financial institutions also provide mortgage loan to home buyers. Local home buyers can also predict the price of the house to find out if a seller is asking for too much. The local seller can also predict their house price and find out how much is a fair market price.

Descriptive Data Analysis

The dataset for this project is downloaded as a csv file from the Kaggle website. This dataset contains house sale prices for King County, which includes Seattle. It includes homes sold between May 2014 and May 2015. The data set consists of 21613 observations and 19 features plus the house price and the id columns.

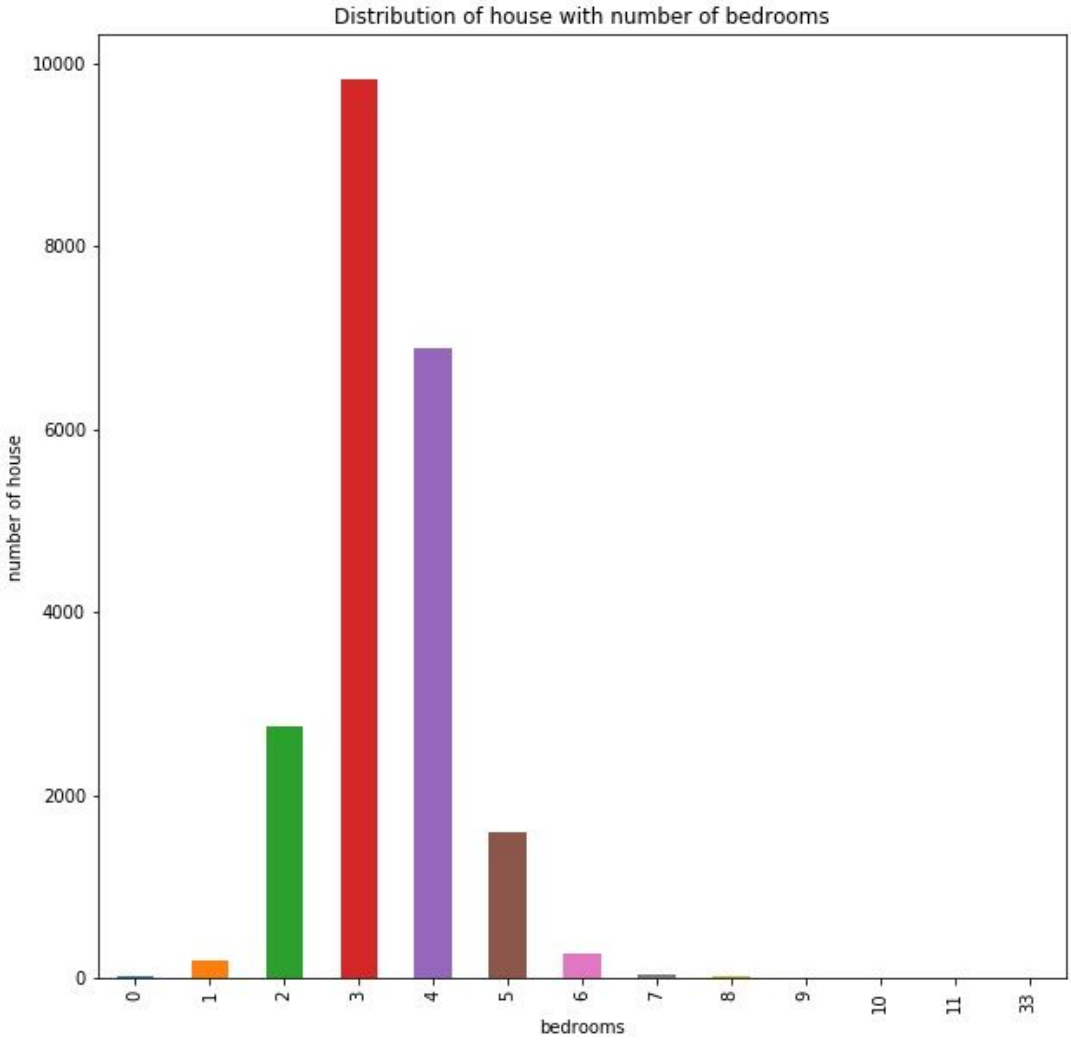
https://www.kaggle.com/harlfoxem/housesalesprediction

The columns are as follows:-
id: a notation for a house
date: Date house was sold
price: Price is prediction target
bedrooms: Number of Bedrooms
bathrooms: Number of Bathrooms
sqft_living: square footage of the home
sqft_lot: square footage of the lot
floors: total floors (levels) in house
waterfront: House which has a view of a waterfront
view: Has been viewed
condition: How good the condition is (Overall)
grade: overall grade given to the housing unit, based on King County grading system
sqft_above: square footage of house apart from basement
sqft_basement: square footage of the basement
yr_built: Built Year
yr_renovated: Year when the house was renovated

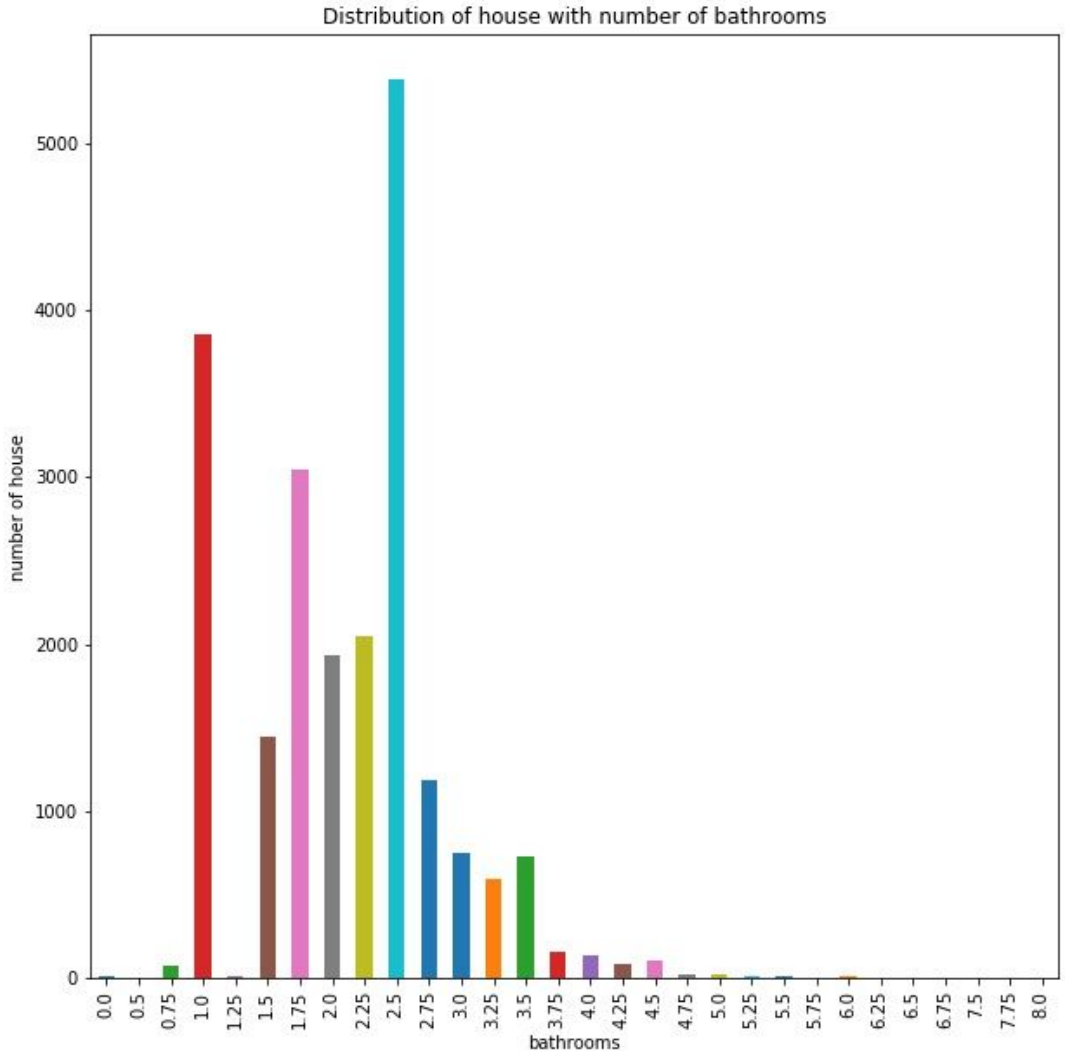
zipcode: zip
lat: Latitude coordinate
long: Longitude coordinate
sqft_living15: Living room area in 2015(implies -- some renovations) This might or might not have affected the lot size area
sqft_lot15: lot size area in 2015(implies--some renovations)

I imported the data from the csv file and converted them into pandas dataframe. From our dataset, I found that the most expensive house is priced 7,700,000.00 and the least expensive house is priced 75,000.00. The average or mean price of a house is 540,088.14 and the median price of a house is 450,000.00. I plot the histogram of price and found the dataset is right-skewed. This indicates that there might be some outliers. I also plotted a boxplot of price to check for outliers in the dataset and found that there are some outliers in our dataset. Then, I checked for outliers for other features like a number of bedrooms, a number of bathrooms, sqft_living, sqft_lot, floors, condition, grade, sqft_above, sqft_basement. Except for the column number of floors, all other columns show that there might be some outliers. Then, I converted the date column from string to datetime data type. Then, I checked how many houses were sold in 2014 and how many in 2015. There were 14633 houses sold in the year 2014, and 6980 houses sold in the year 2015.

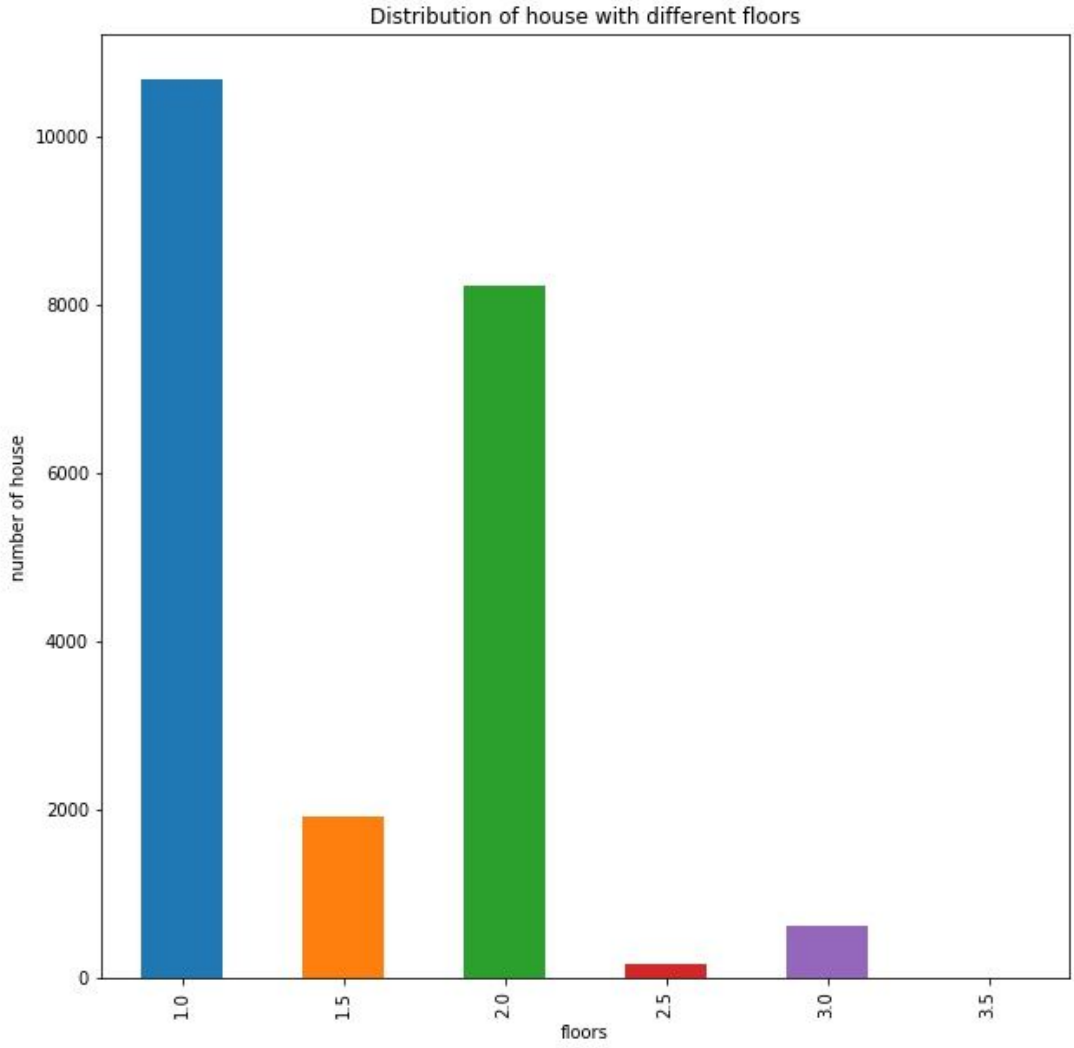
I plotted bar plots with grouped data to see the distributions and demographics of the data.



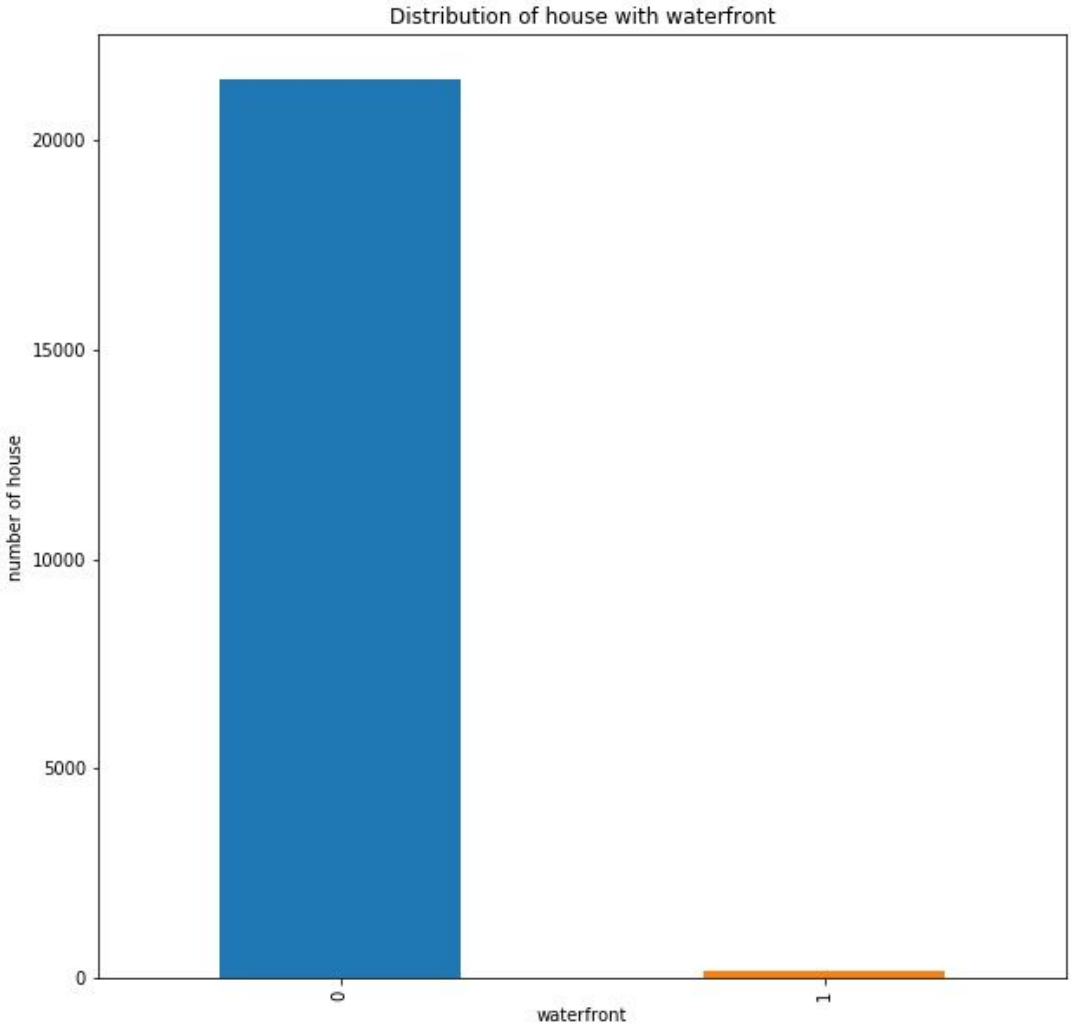
The distribution of the house with the number of bedrooms plot tells us that most of the houses have 3 bedrooms.



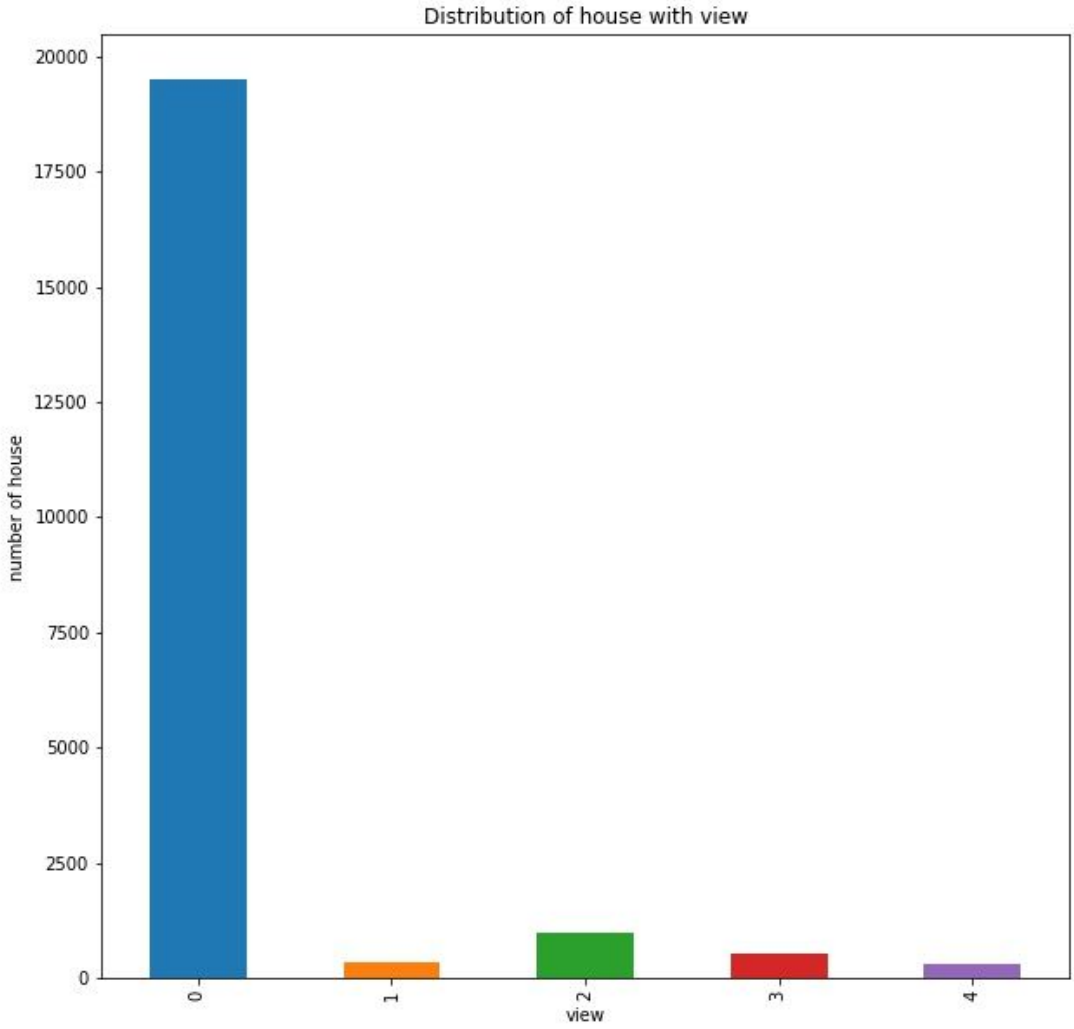
The distribution of the house with the number of bathrooms plot tells us that most of the houses have 2.5 bathrooms.



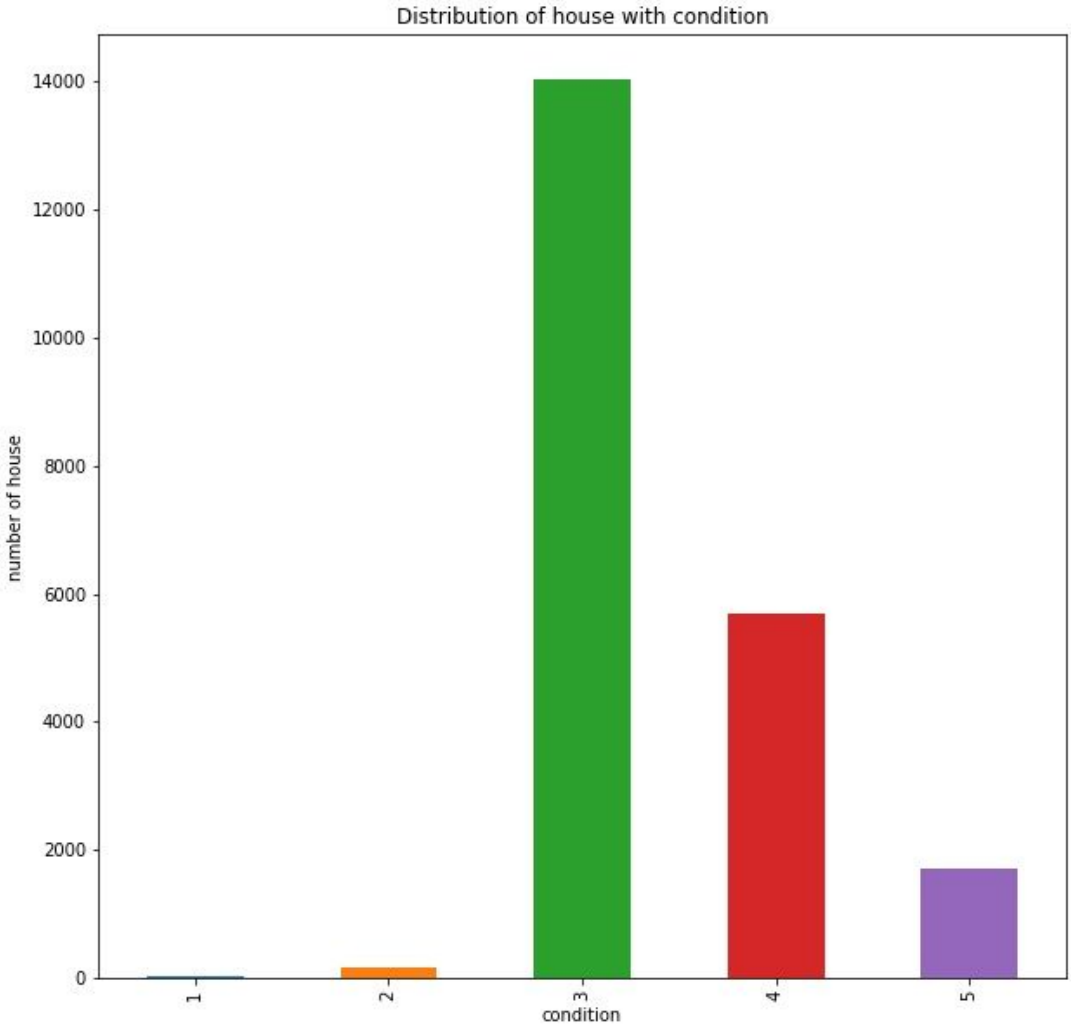
Grouping the house by floors tell us that most of the houses have 1 floor.



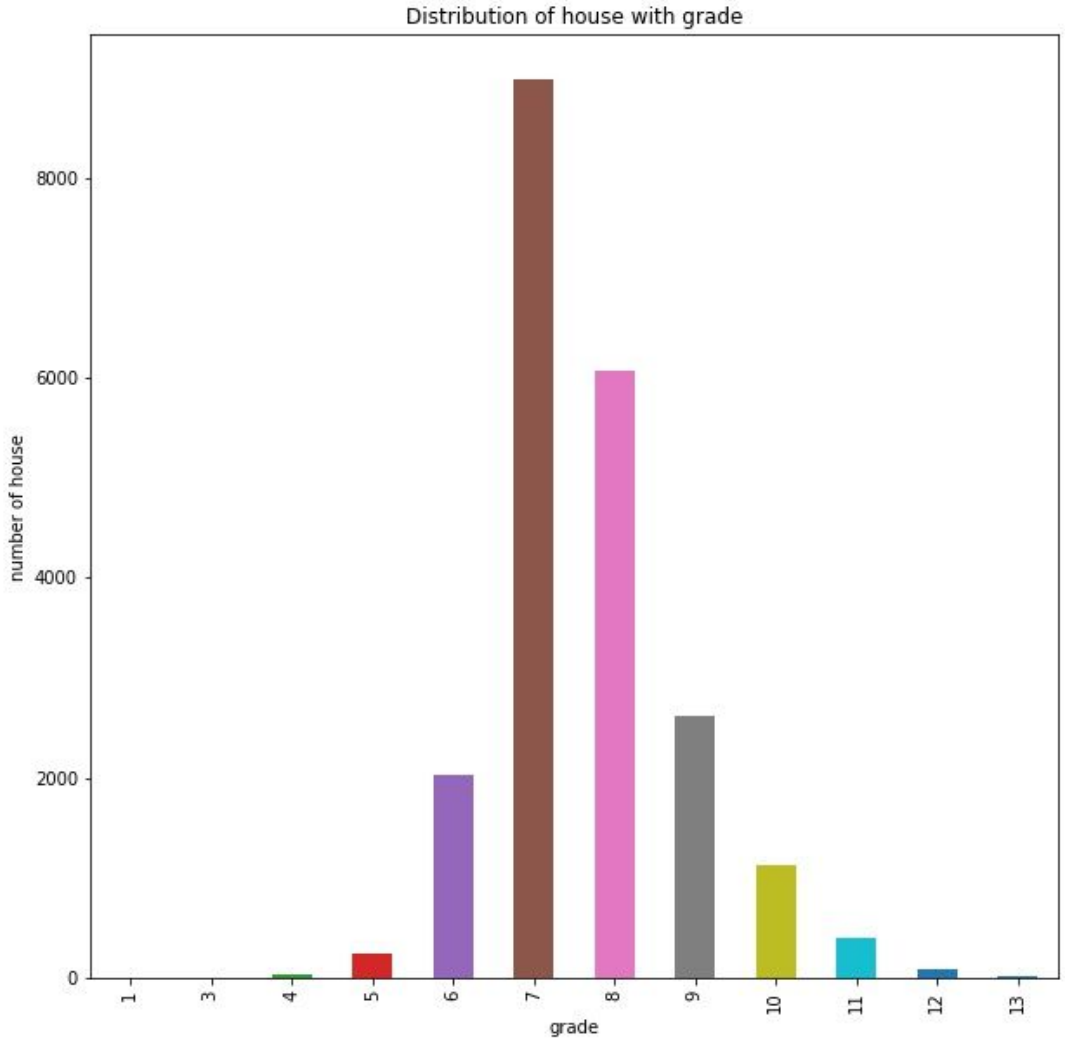
Grouping the house by waterfront tells us that most houses do not have waterfronts.



The distribution of the house with view bar plot tells us that most of the house has 0 views. It means that the houses are sold to the first viewer of the house.

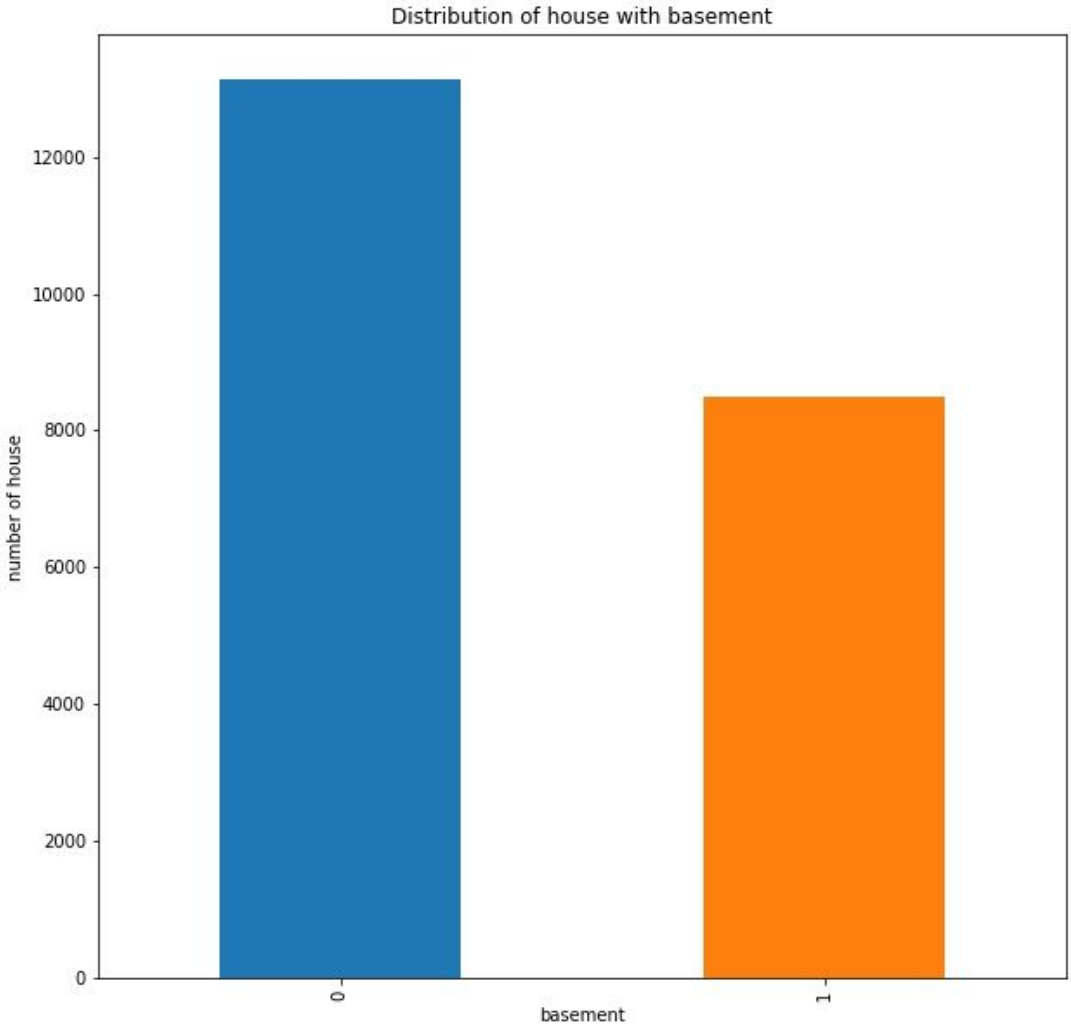


Grouping the house by condition tells us that most of the house has 3 points for the condition.

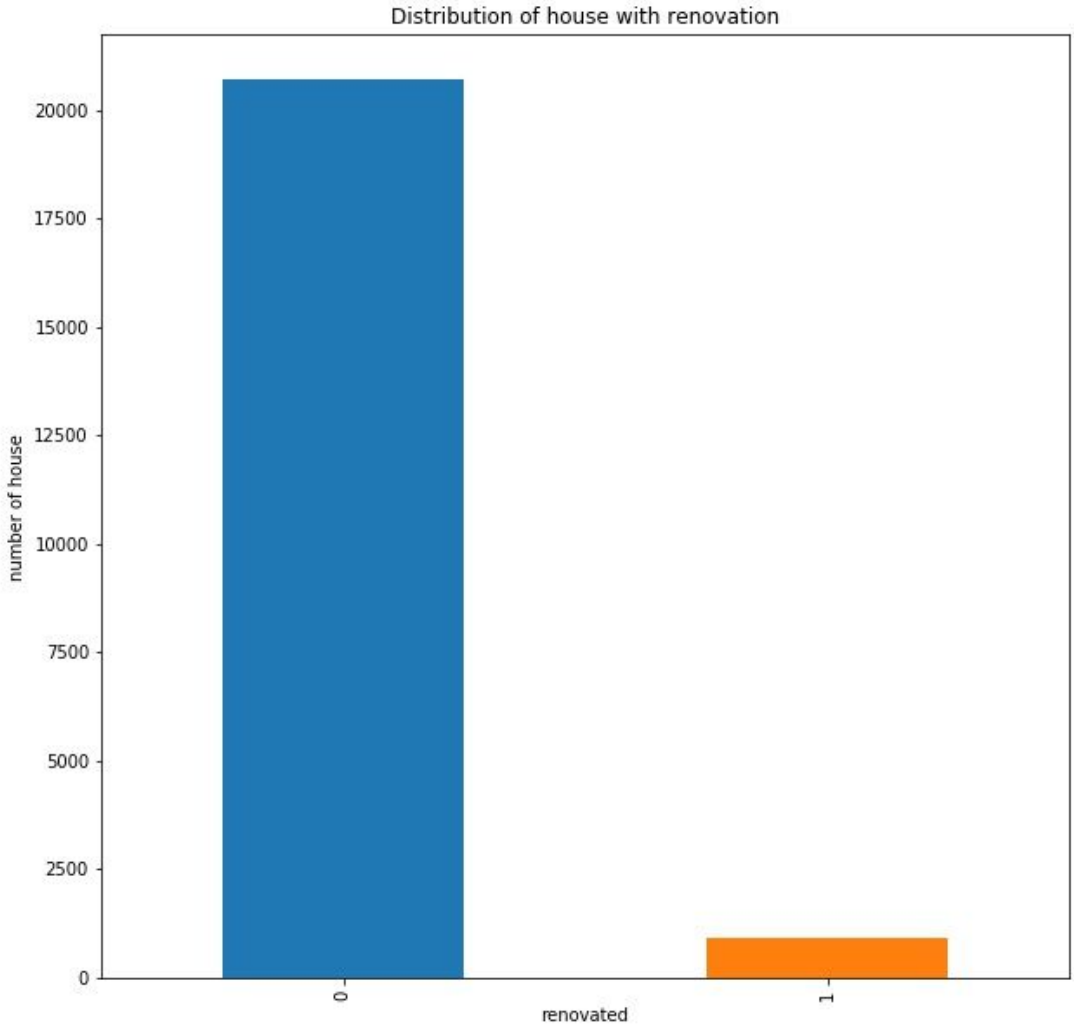


Grouping the house by grade tells us that most of the house has 7 points for a grade.

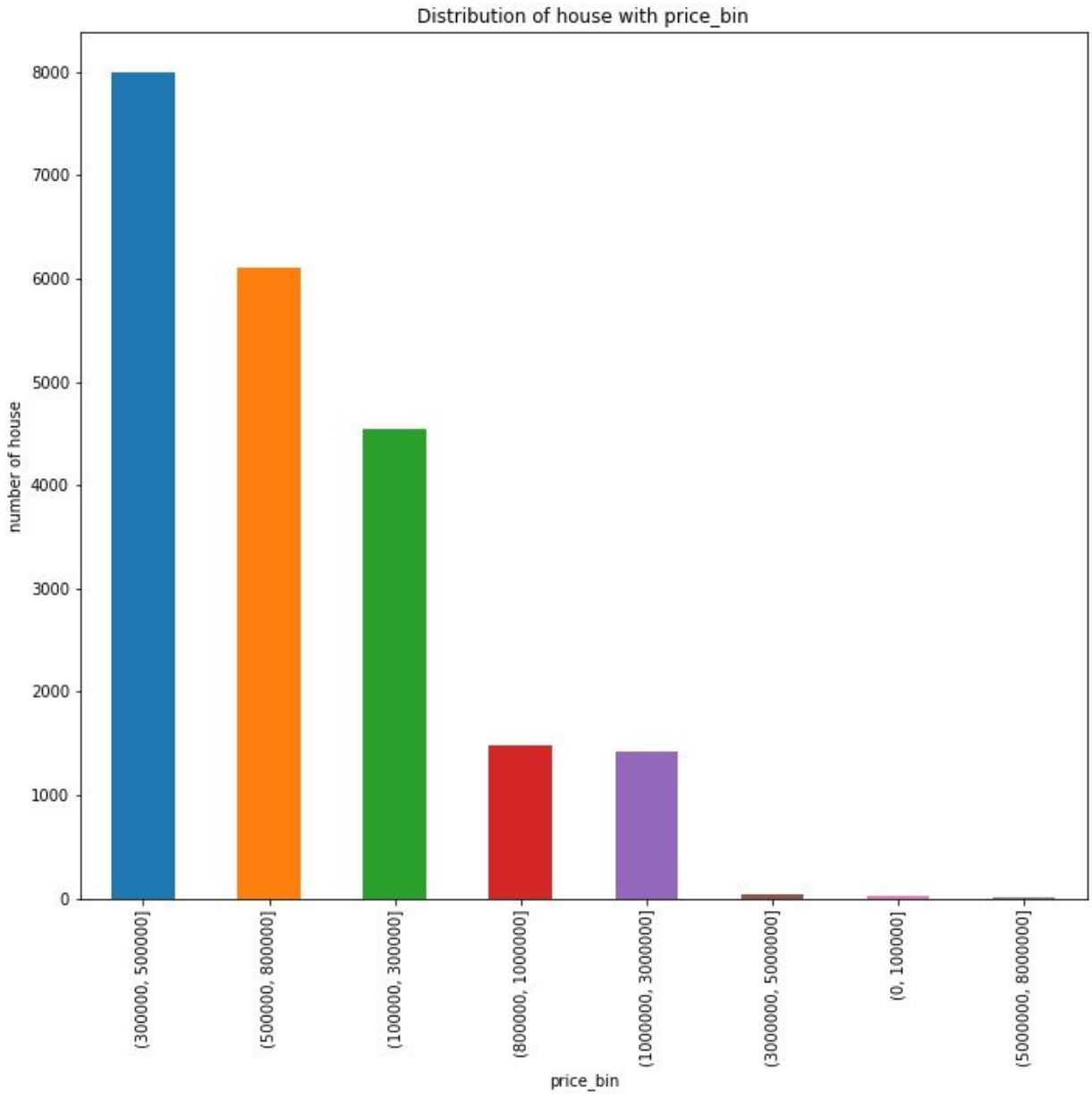
Then, I created a new column named basement to hold boolean data of house with or without the basement.



I created a bar plot and found most of the houses do not have a basement.

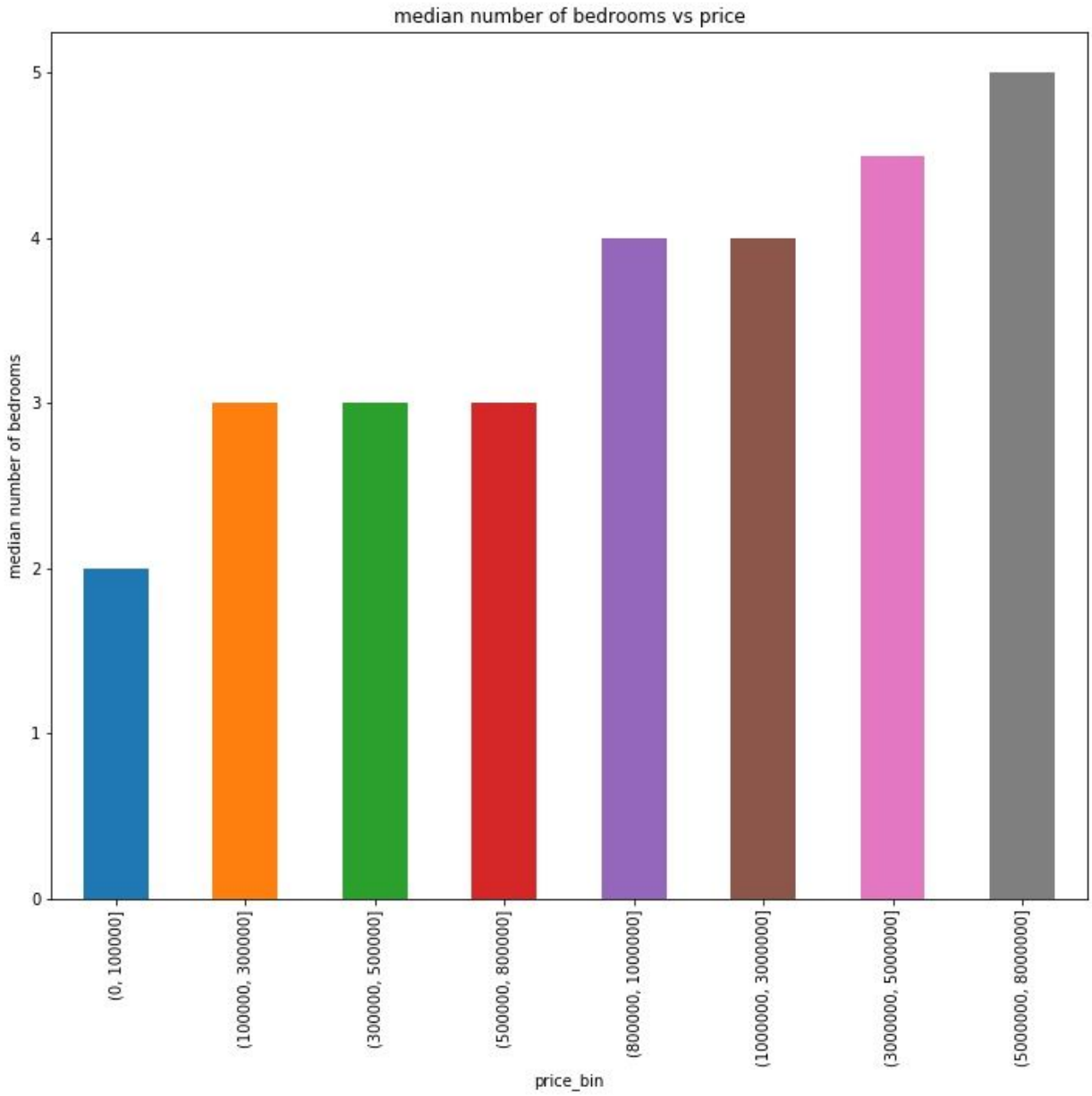


Grouping the house by renovated or not renovated, I found that most of the houses are not renovated.



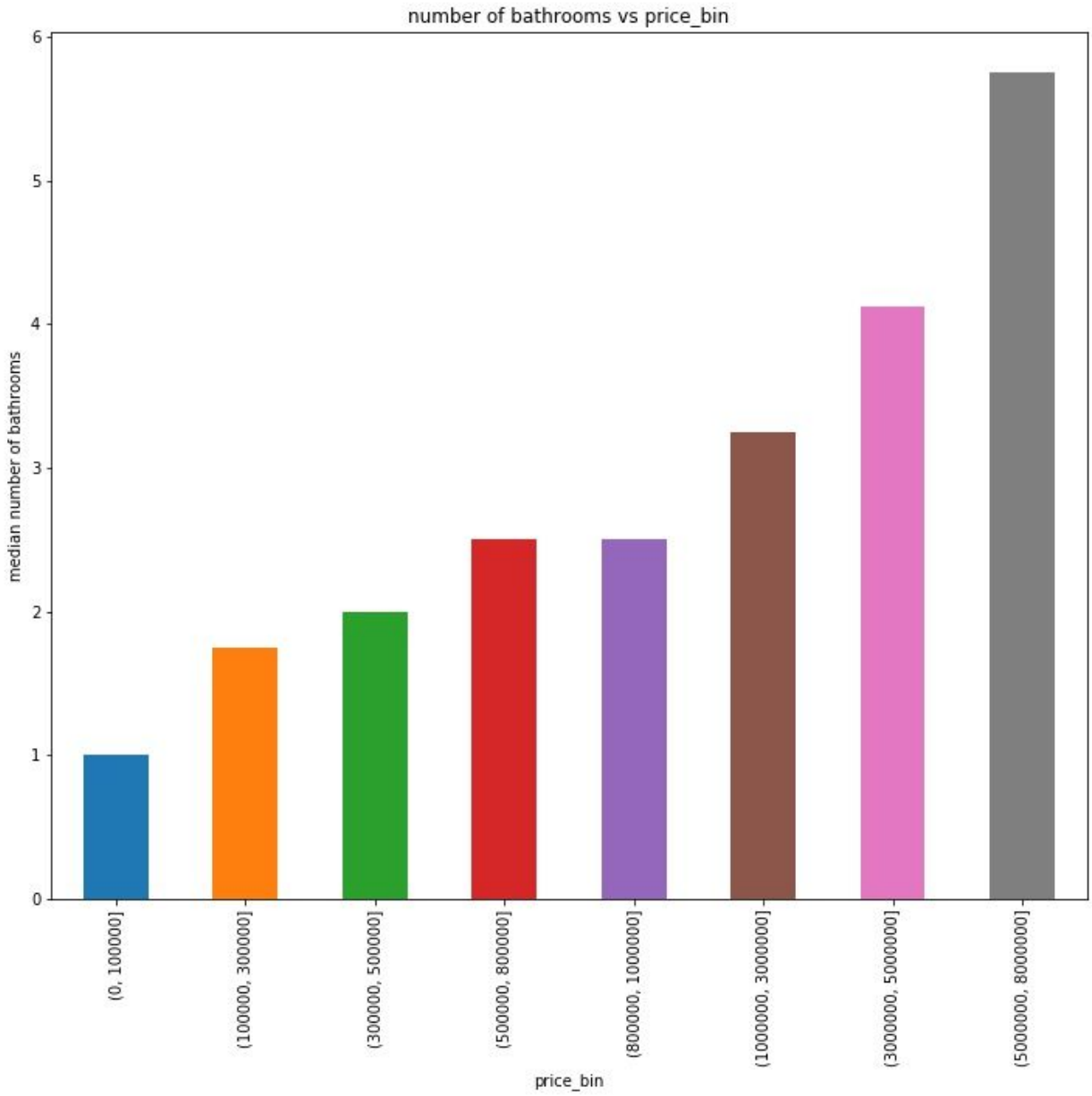
Grouping the house by price_bin, I found that most of the house is priced between 300,000 to 500,000.

Then, I plot a bar plot of the median number of bedrooms vs price_bin.

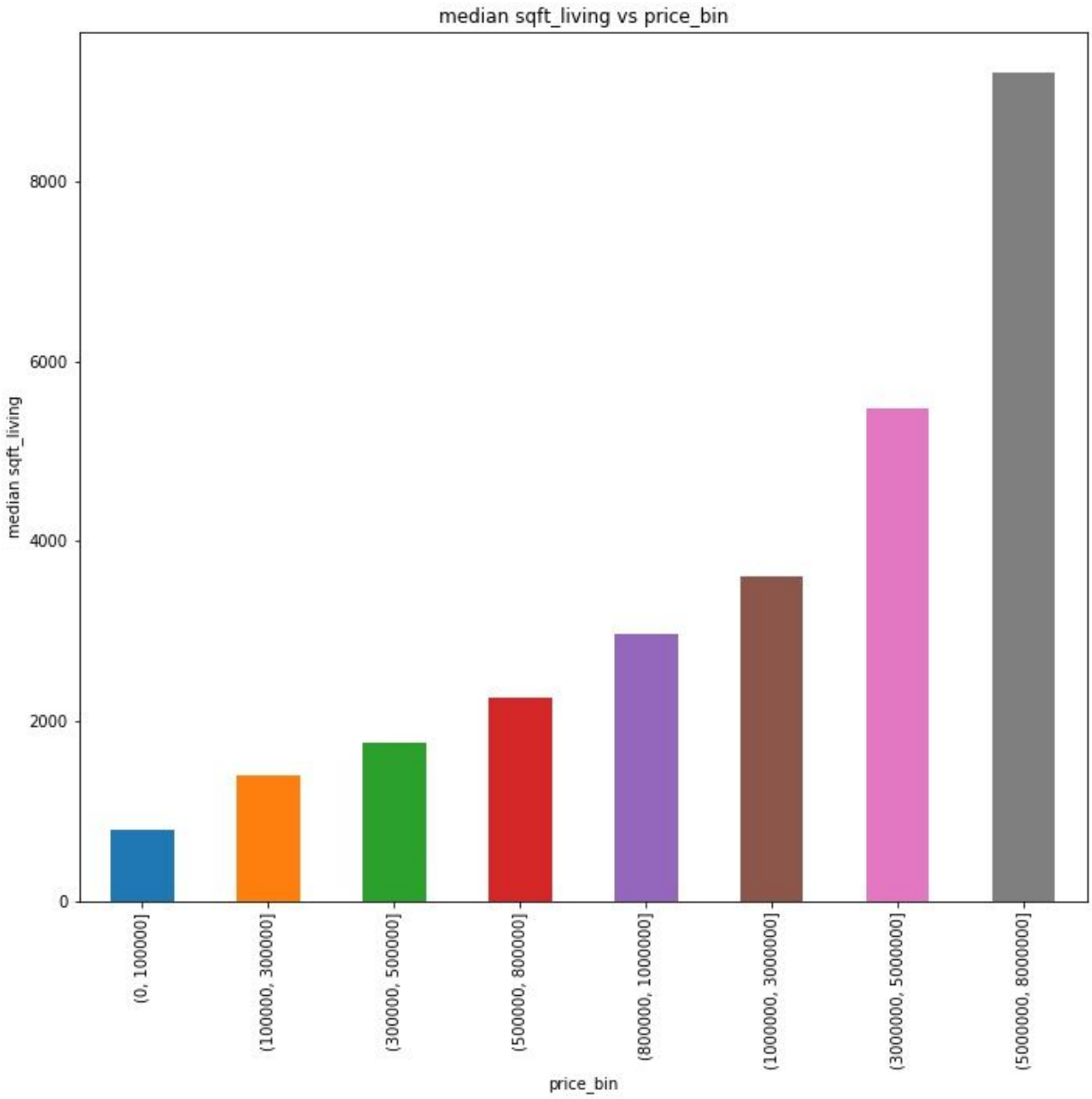


The plot shows the median number of bedrooms and price_bin is directly proportional to each other and tells us that more expensive houses have a maximum median number of bedrooms. Houses priced 5,000,000 to 8,000,000 has median 5 bedrooms.

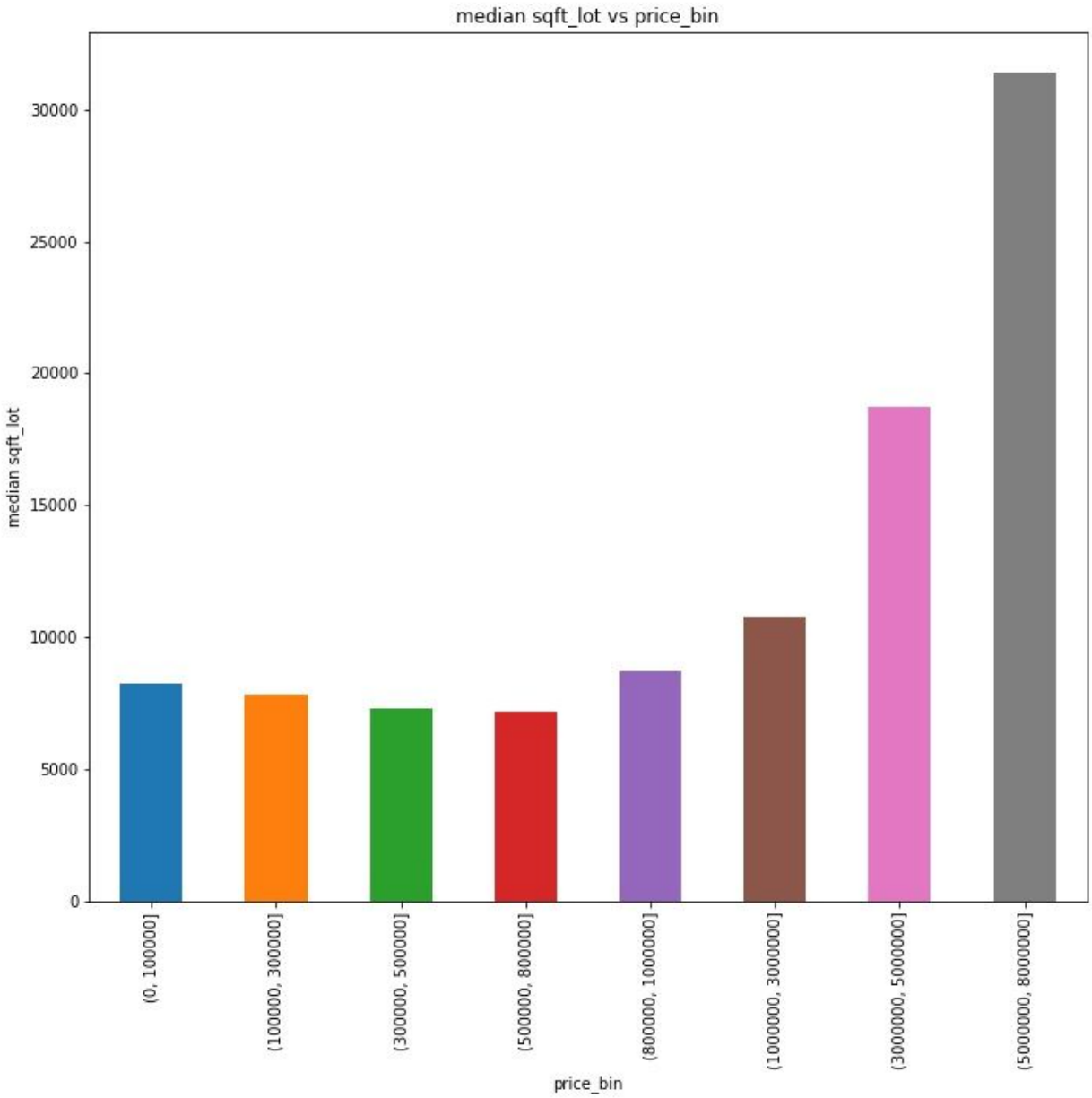
Then, I plot another bar plot of the median number of bathrooms vs price_bin.



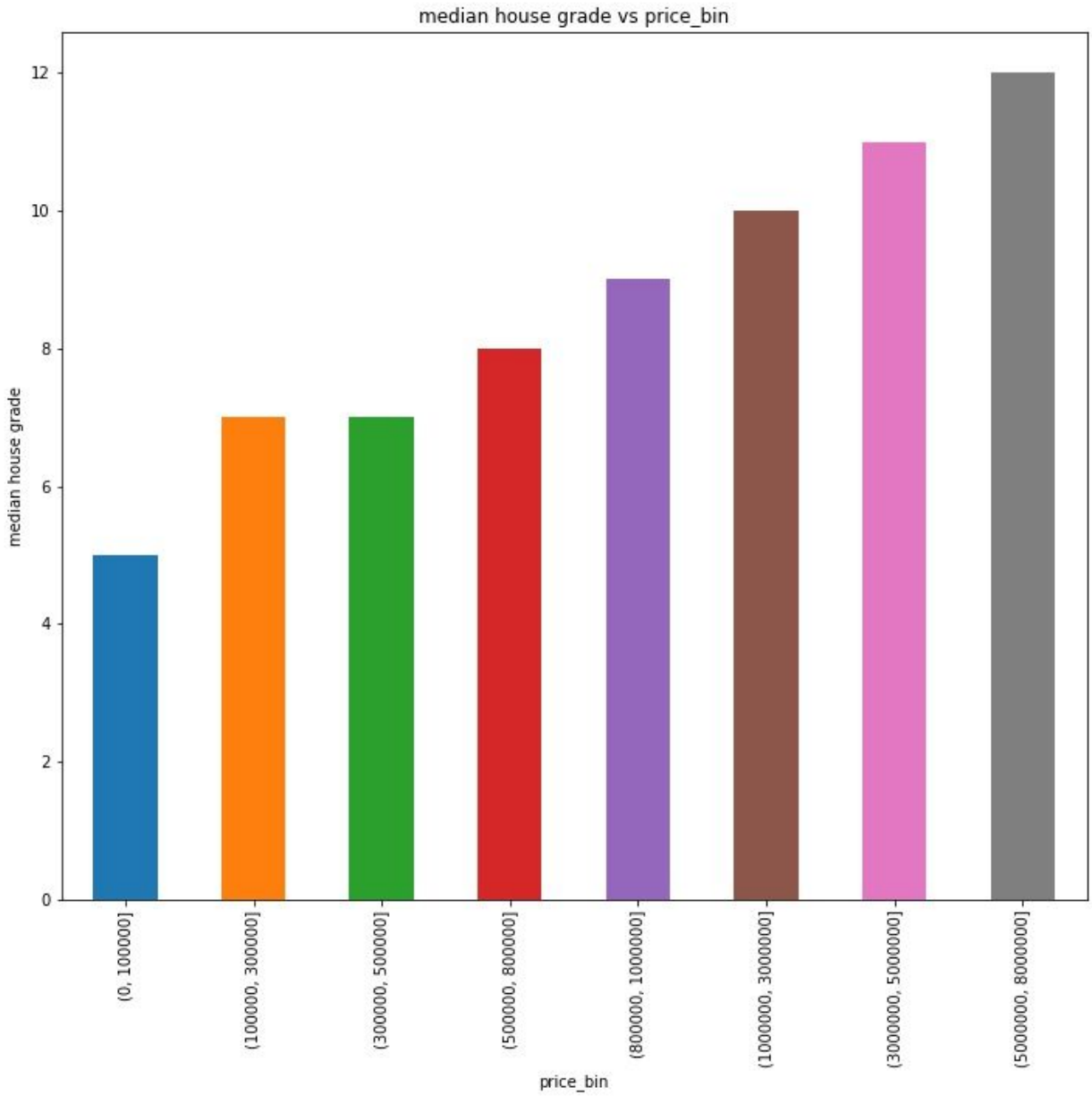
The plot shows the median number of bathrooms and price_bins are directly related to each other and tells us that more expensive houses have a maximum median number of bathrooms.



Sqft_living vs price_bin plot tells us that more expensive houses have larger sqft_living and a less expensive house has smaller sqft_living.



Sqft_lot vs price_bin plot shows that houses priced 0 to 500,000 have an inverse relation with median sqft_lot and houses priced 500,000 to 8,000,000 has a direct relationship with median sqft_lot.



Median house grade vs price_bin plot shows a direct relationship between median house grade and price_bin.

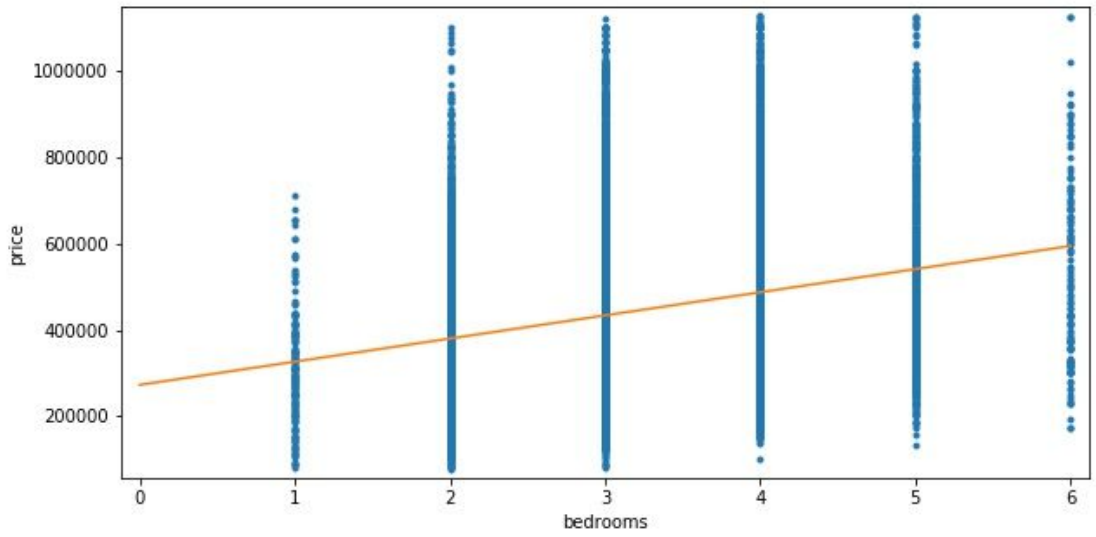
I also drew scatter plot to find if there is any correlation between price and other features. Price vs number of bedrooms shows a positive correlation between price and number of bedrooms. It also shows us some outliers data. Price vs number of bathrooms shows some positive correlation with some outliers. Price vs sqft_living also shows a positive correlation. Price vs sqft_lot shows some positive correlation but many outliers. Price vs number of floors, price vs waterfront, price vs view, and price vs condition shows very weak correlation. Price vs grade, price vs sqft_above, price vs sqft_basement, price vs sqft_living15, and price vs sqft_lot15 shows a positive correlation.

Data Wrangling and Cleaning

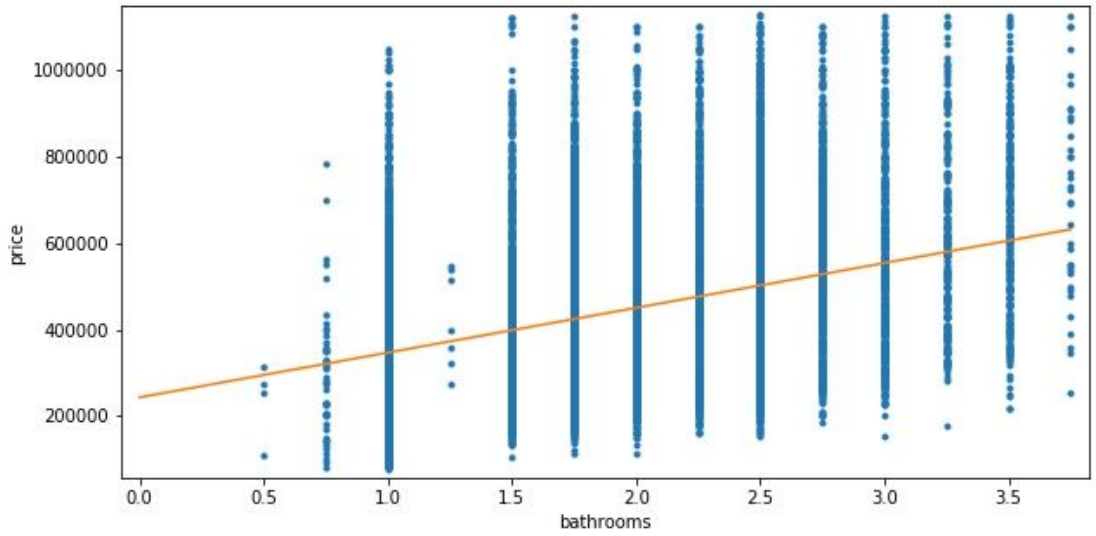
I created a copy of the dataframe df called new_df in order to perform data wrangling. First, I created a function named remove_outliers which takes two input as parameters. One is the string name of the column, and another is the name of the dataframe. Inside the function, interquartile range is calculated. Then, we will locate the outliers using the upper and lower limit values. Then, we will drop the observation which has outliers. I removed outliers from price, bathrooms, sqft_living, sqft_lot, sqft_above, sqft_basement, sqft_living15, and sqft_lot15. For the column bedrooms, there was a bad data 33 which looks like a mistakenly entered data. I checked to see what the features of the house with 33 bedrooms were. Comparing median sqft_living and median sqft_lot size, I found that the mistake data must be 3 instead of 33. I plotted a box plot for the column bedroom to see if there are any other outliers. The plot shows that there are some outliers in the data set. I decided to remove outliers with more than 6 bedrooms and less than 1 bedrooms because few houses have more than 6 bedrooms and less than 1 bedrooms. Majority of the houses have more than 1 bedroom and less than 6 bedrooms. After data wrangling, we have 16,496 observations in our dataset.

Data Story

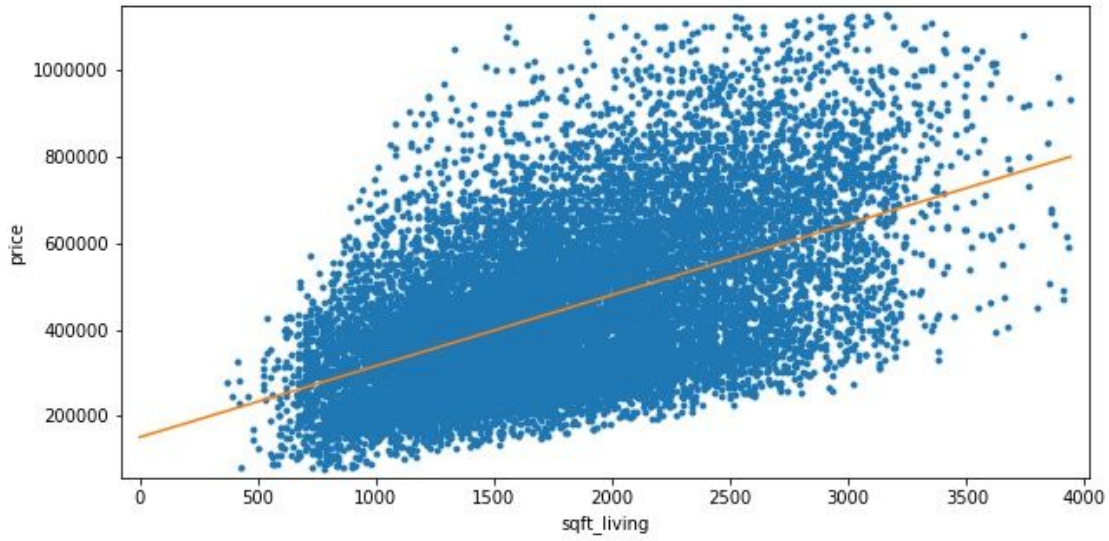
I defined a function which takes three parameters as input i.e. x_input, y_input, and data. This function will plot a scatterplot with a regression line. I drew a scatterplot of price vs most of the features from the dataset.



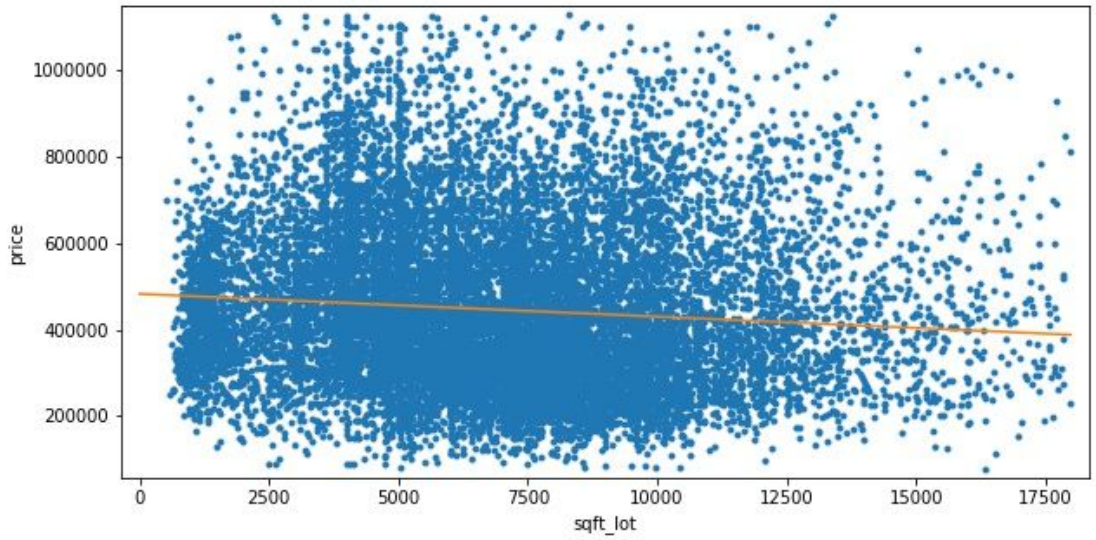
Price vs bedrooms scatterplot shows there is a positive correlation.



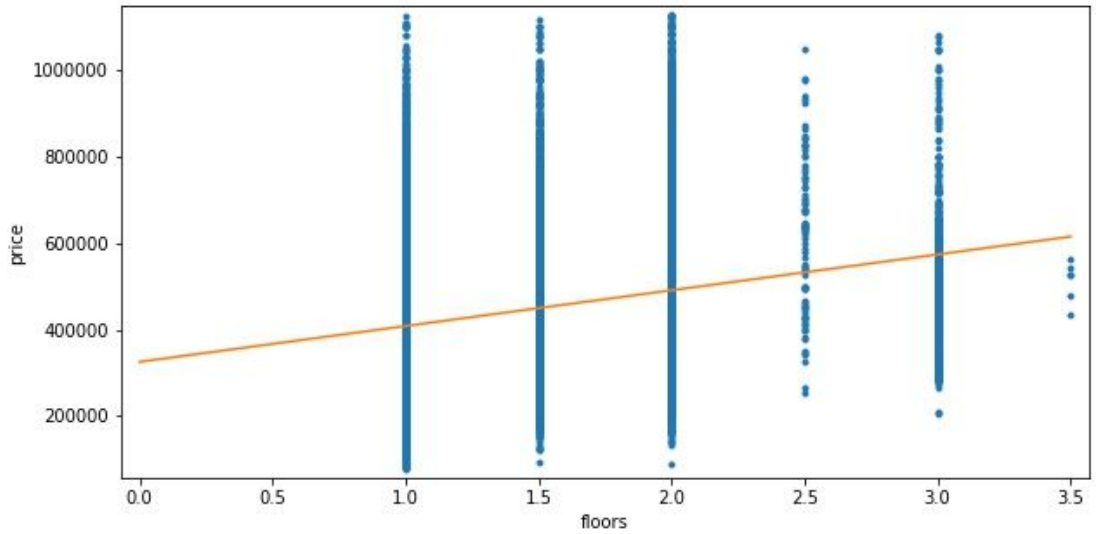
Price vs bathrooms scatterplot also shows there is a positive correlation.



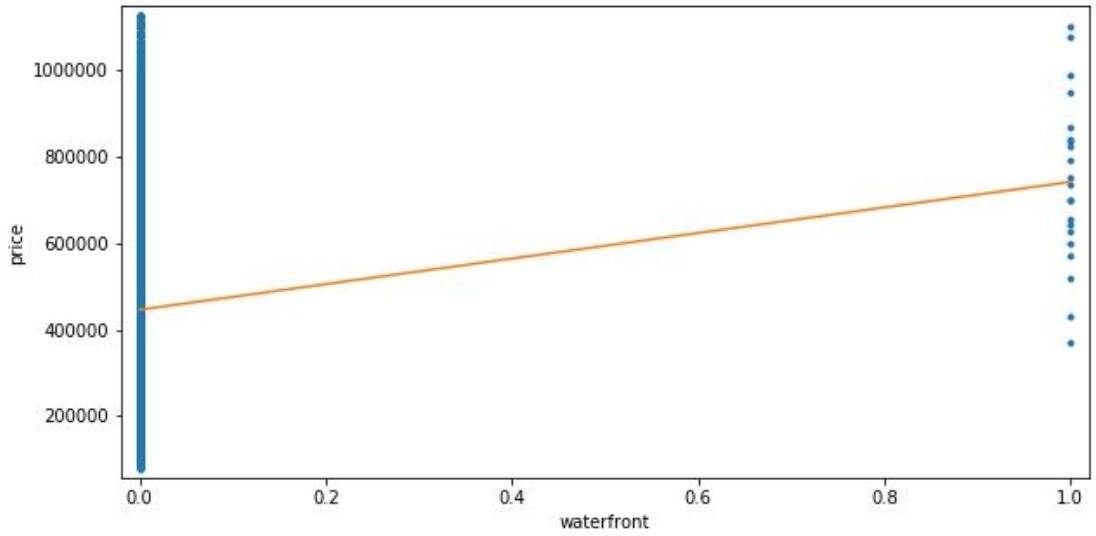
Price vs sqft_living plot also shows there is a positive correlation.



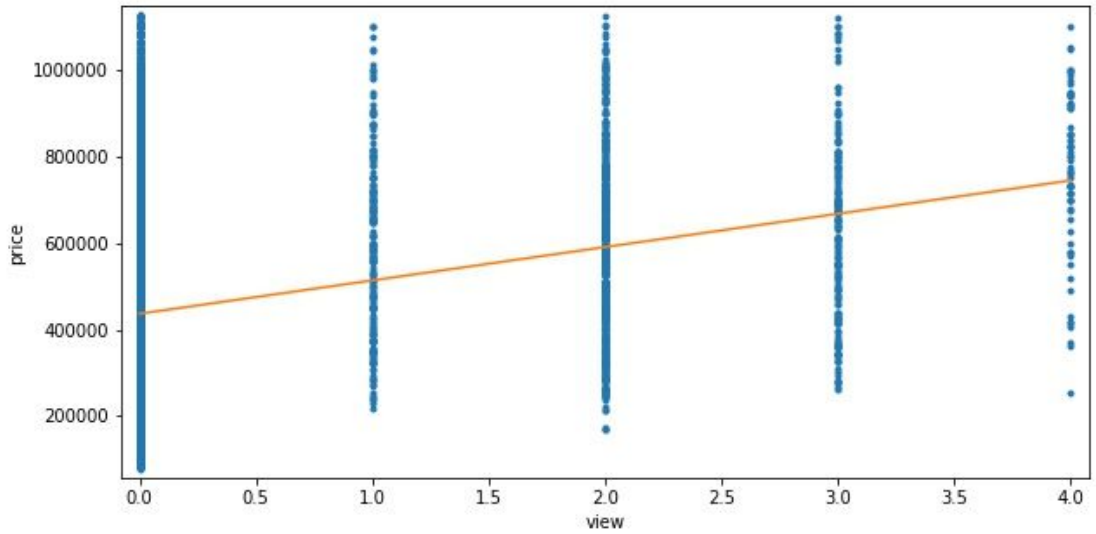
Price vs sqft_lot scatter plot shows a weak negative correlation.



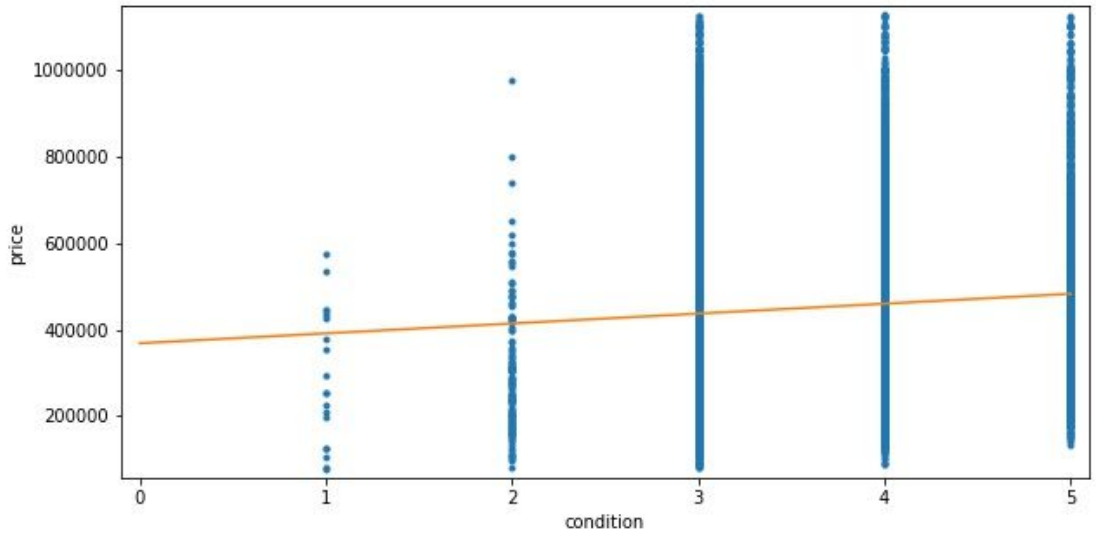
Price vs floors scatterplot shows a positive correlation.



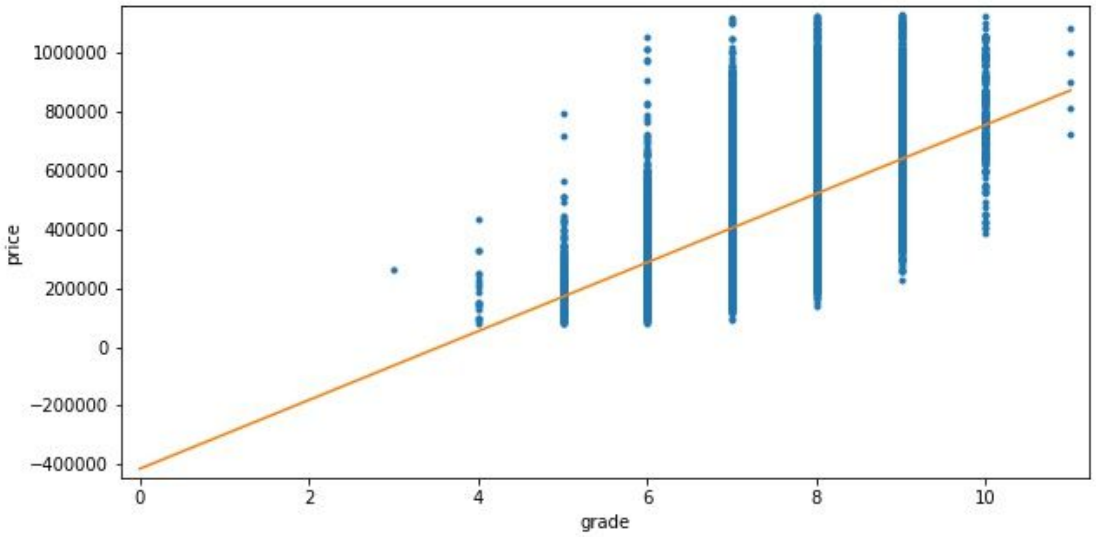
Price vs waterfront scatterplot shows a positive correlation.



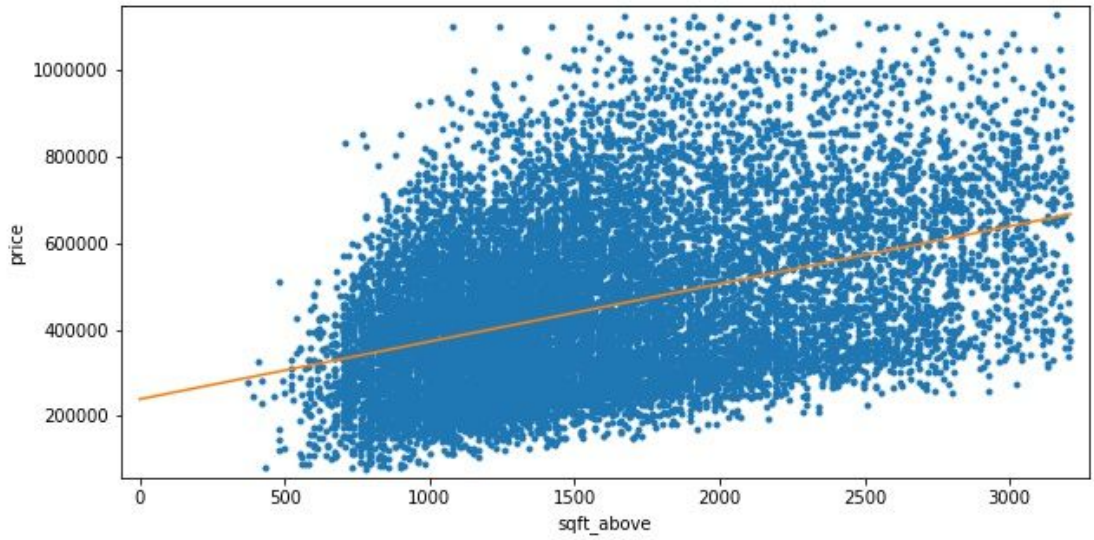
Price vs view scatterplot shows a positive correlation.



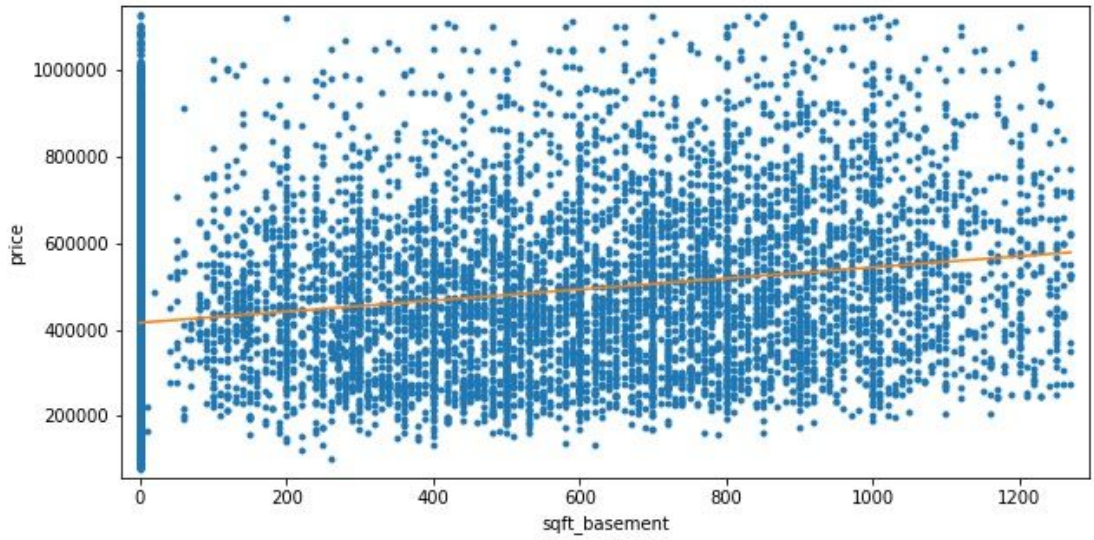
Price vs condition scatterplot shows a weak correlation.



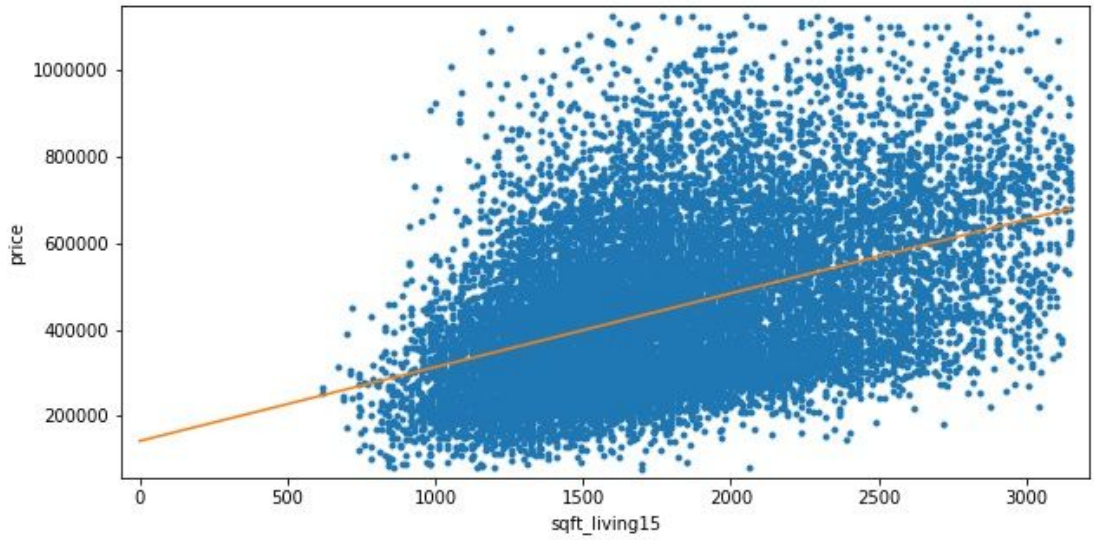
Price vs grade scatterplot shows a positive correlation.



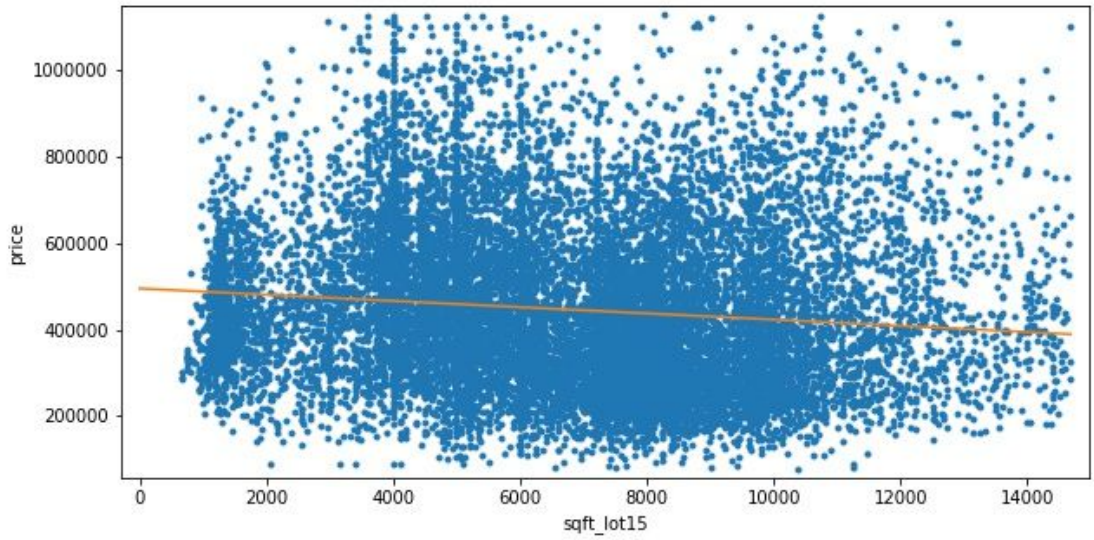
Price vs sqft_above scatterplot shows a positive correlation.



Price vs sqft_basement shows a weak correlation.

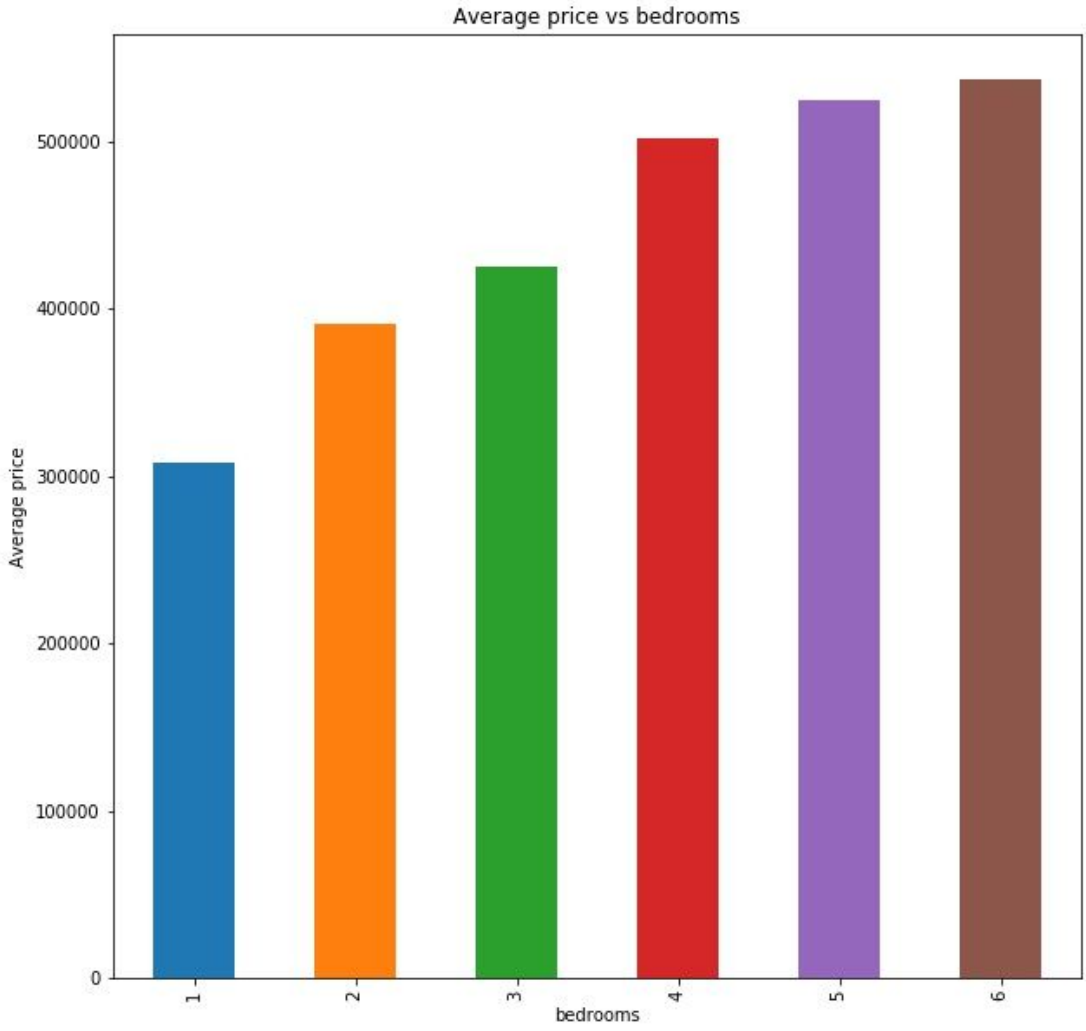


Price vs sqft_living15 scatterplot shows a positive correlation.

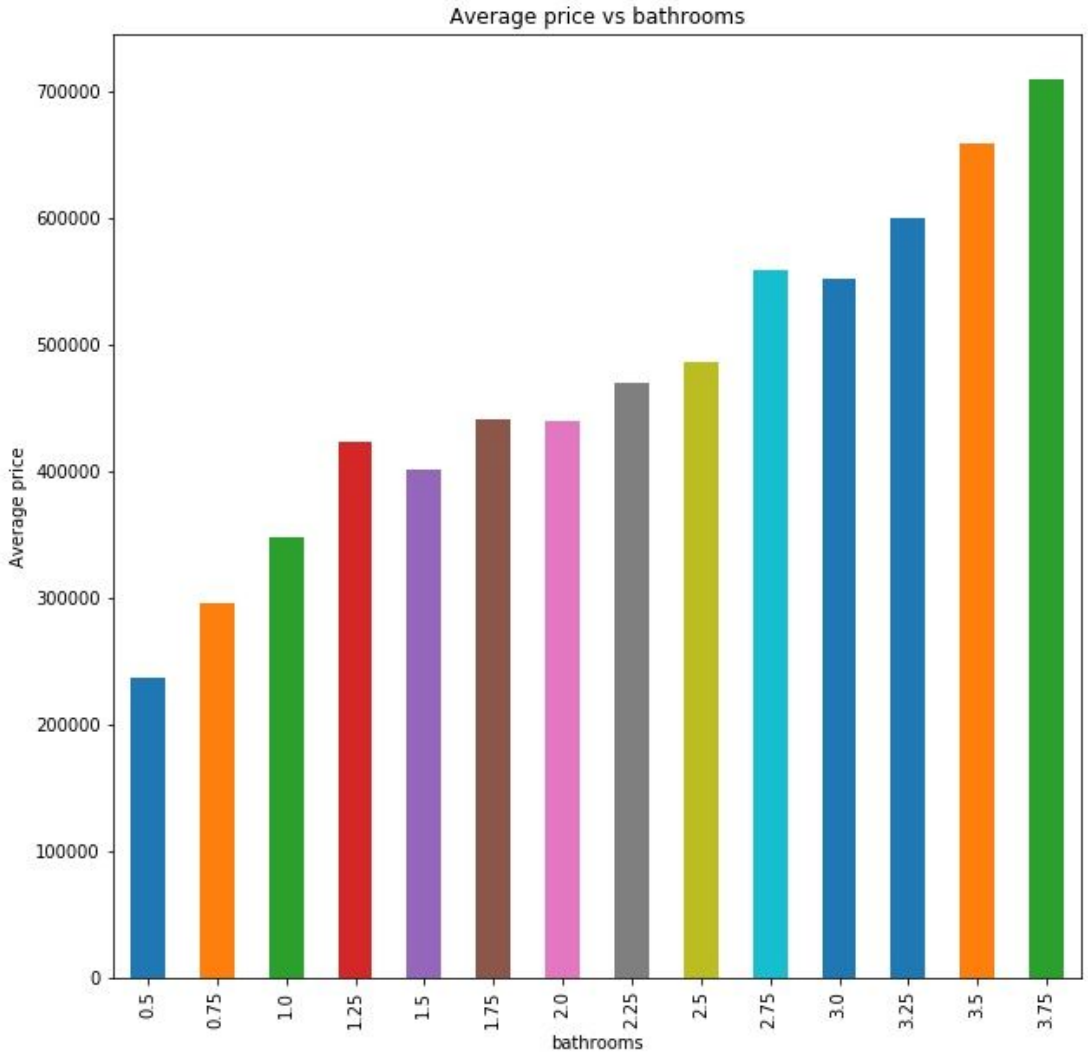


Price vs sqft_lot15 scatterplot shows a weak negative correlation.

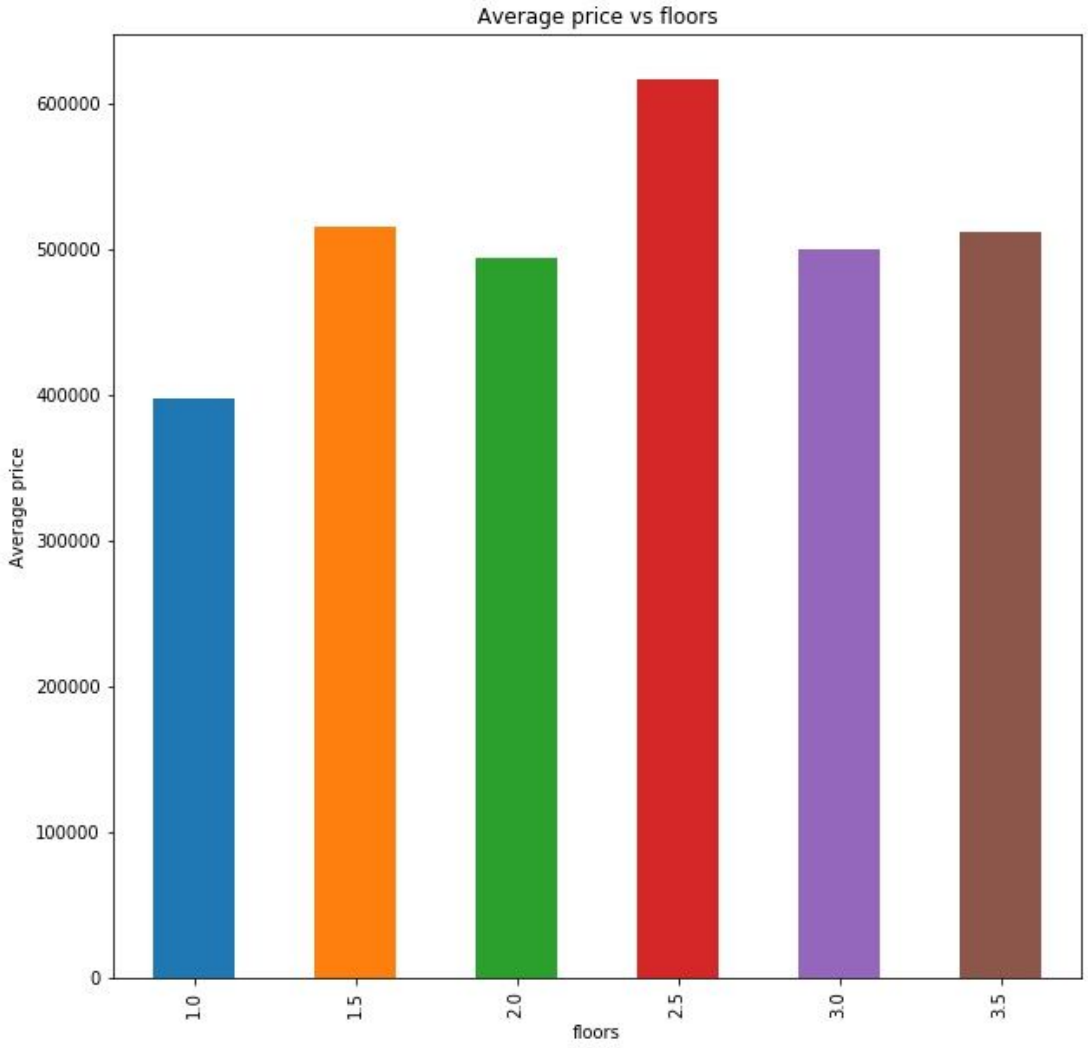
I defined another function named bar_chart which will draw a bar plot using a column name and dataframe as parameter input.



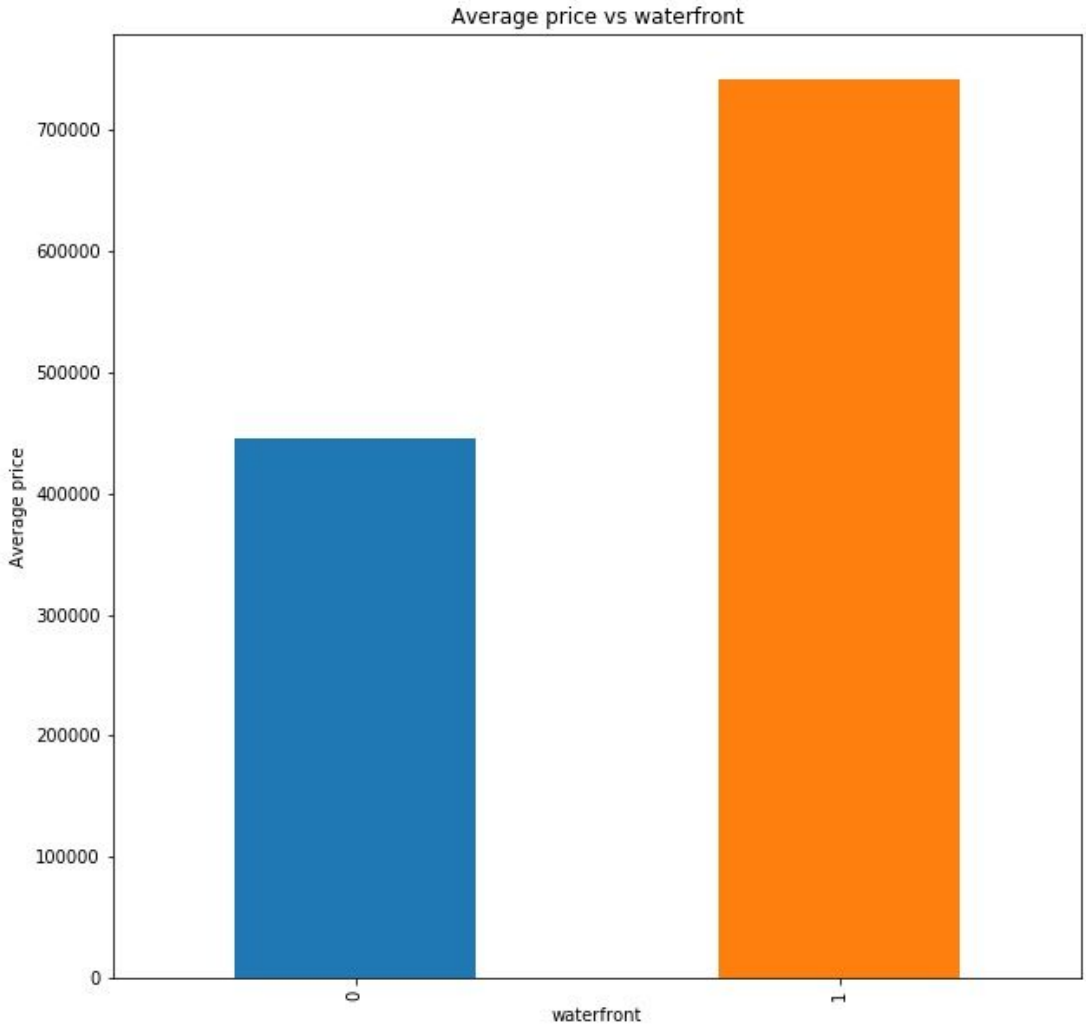
The bar plot average price vs bedrooms shows average price and bedrooms is directly proportional to each other. House with 6 bedrooms has the highest average price.



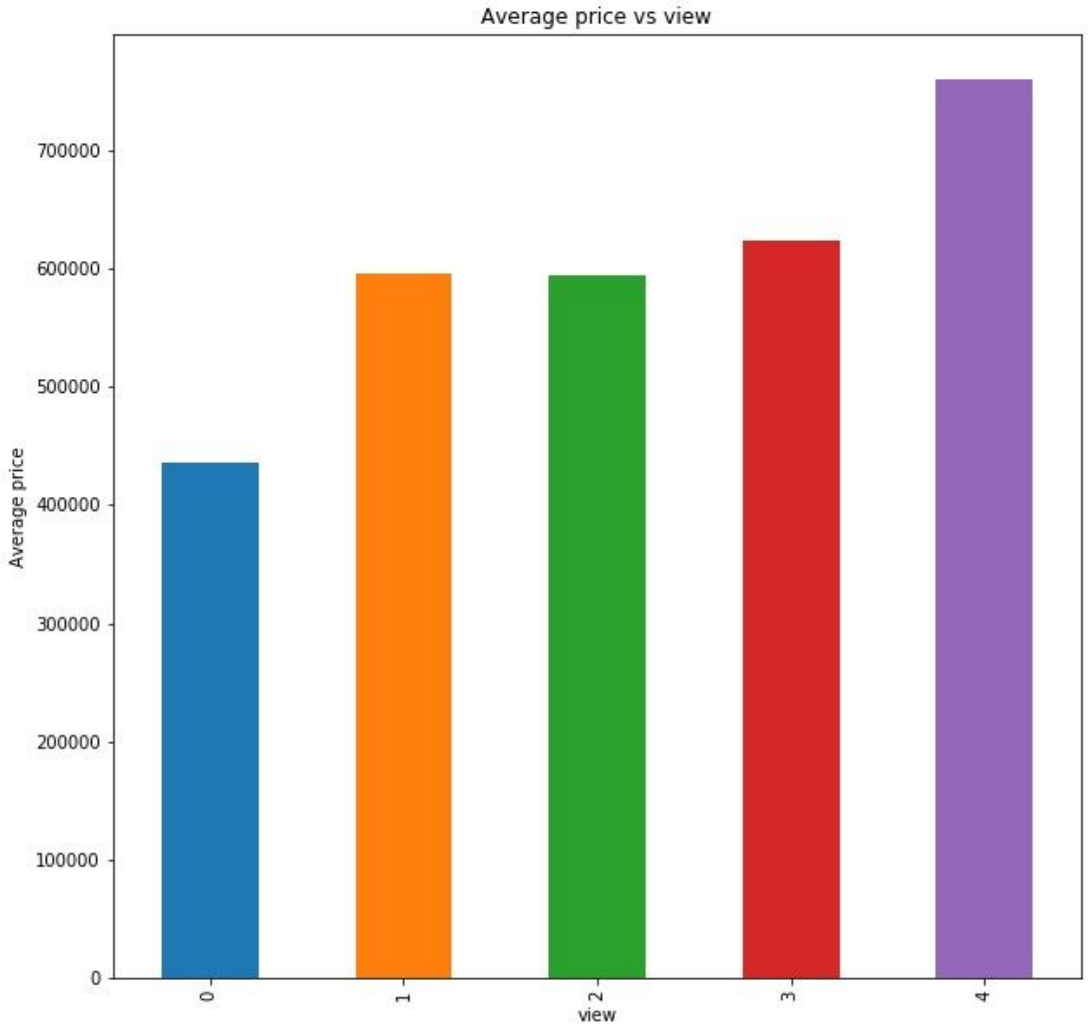
The bar plot average price vs bathrooms shows average price and bathrooms is also directly proportional to each other. House with 3.75 bathrooms has the highest average price.



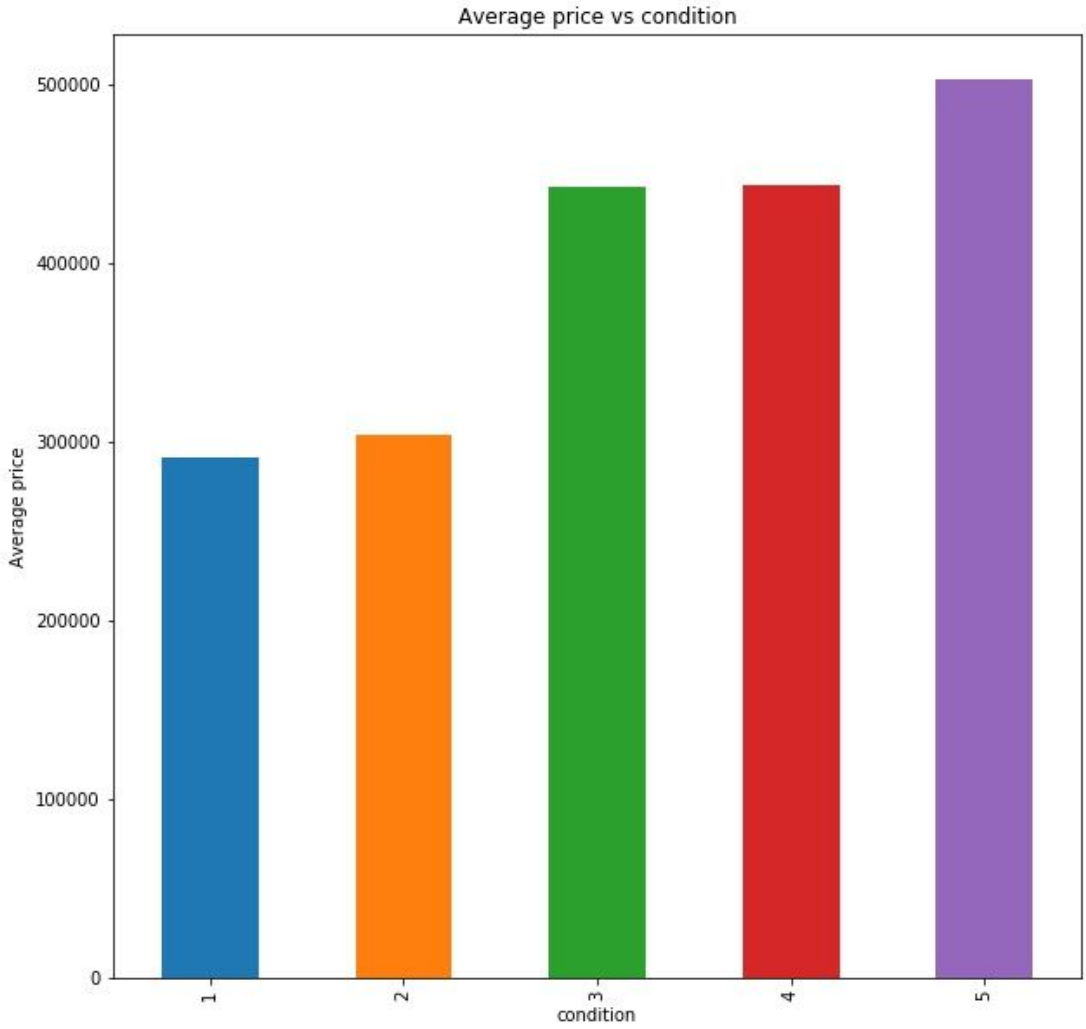
The bar plot average price vs floors shows house with 2.5 floors has the highest average price.



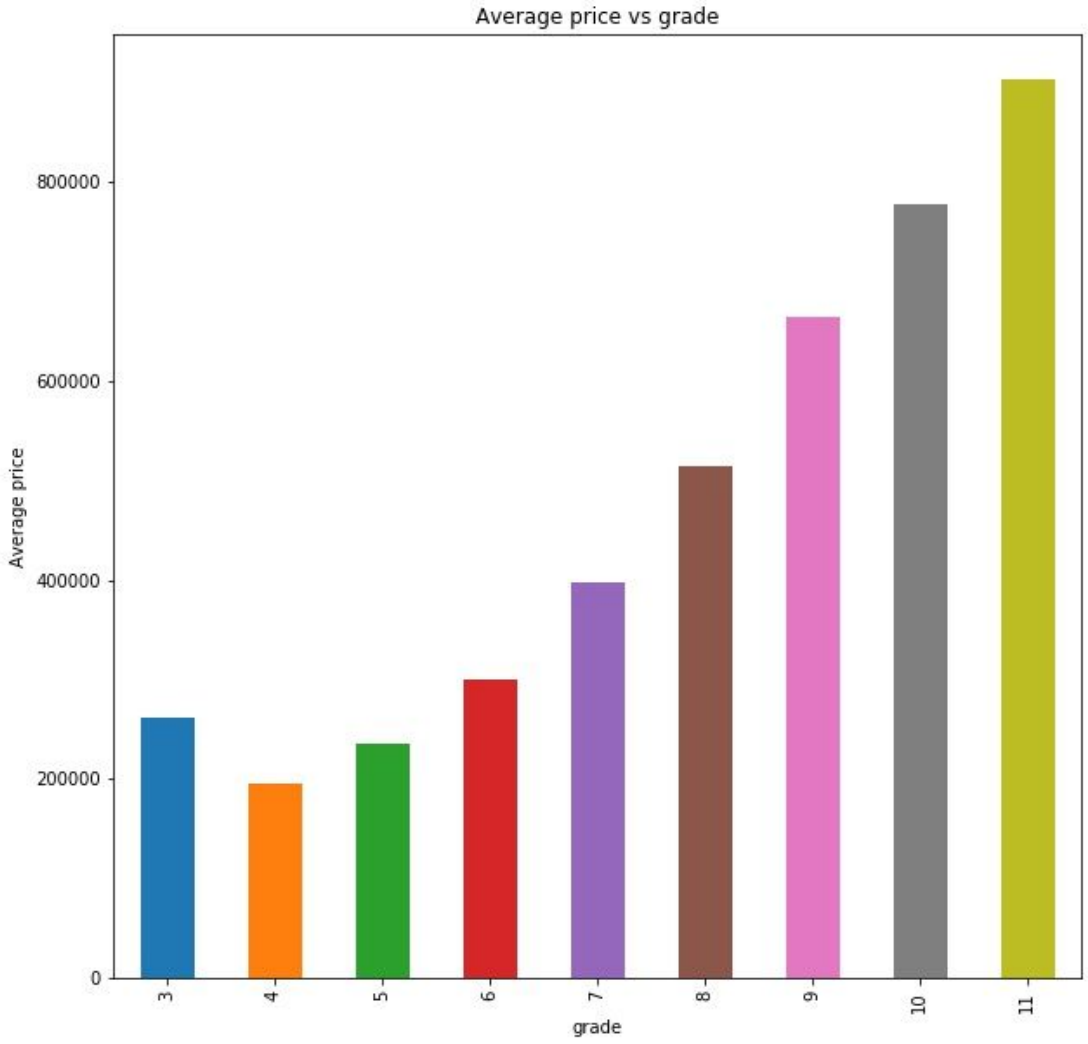
The bar plot average price vs waterfront shows house with waterfront has the highest average price.



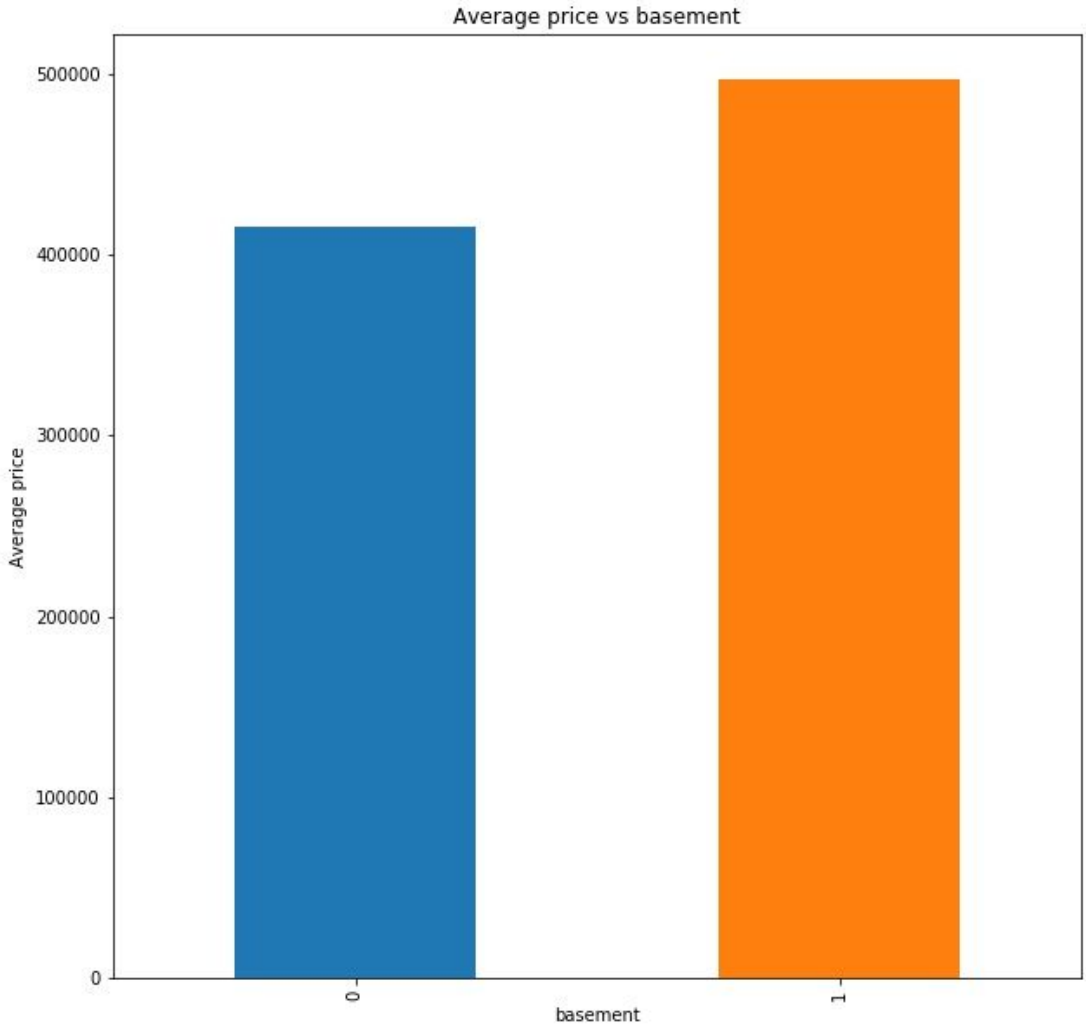
The bar plot average price vs view shows house with 4 view has the highest average price.



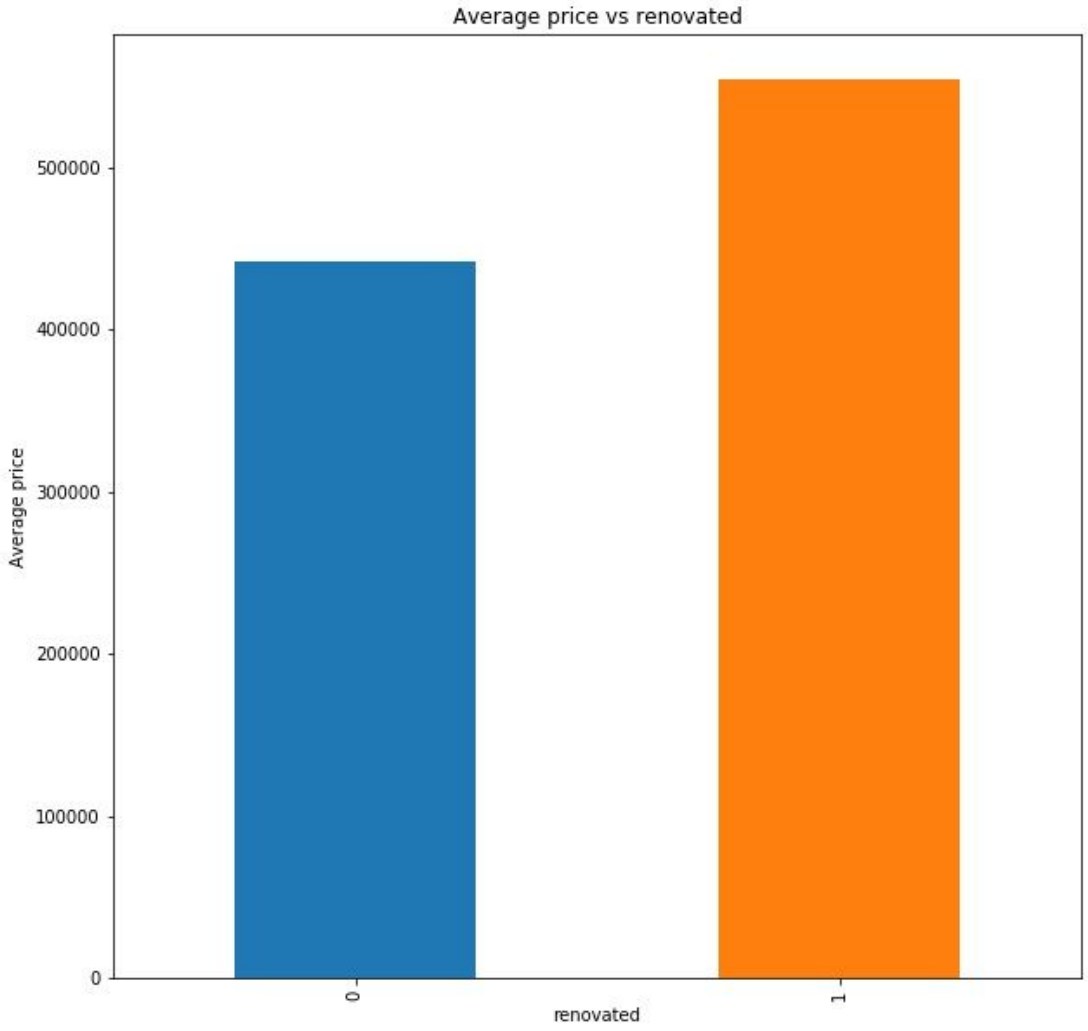
The bar plot average price vs condition shows house with condition 5 has the highest average price.



The bar plot average price vs grade shows house with grade 11 has the highest average price.



The bar plot average price vs basement shows house with a basement has the highest average price.



The bar plot average price vs renovated shows house which is renovated has the highest average price.

Inferential Statistics

I conducted a hypothesis test to check if there is no significant correlation between a number of bedroom and price. The p-value for the hypothesis test is less than the level of significance 0.05, so we reject the null hypothesis. So we support that there is a correlation between a number of bedrooms and price. I also conducted a hypothesis test to check the correlation between a number of bathrooms and price. The p-value for the hypothesis test is less than the level of significance 0.05, so we reject the null hypothesis and suggest that there is a correlation

between a number of bathrooms and price. Similarly, I conducted a hypothesis test to check the correlation between sqft_living and price. The p-value for the hypothesis test is less than the level of significance 0.05, so we reject the null hypothesis and suggest that there is a correlation between sqft_living and price. I also conducted a hypothesis test to check if there is a correlation between grade and price. The test suggests that there is a correlation between grade and price. I also conducted a hypothesis test to check if there is no statistical importance between mean house price and a number of bedrooms less than 3 and greater than 3. The p-value for the test was greater than the level of significance 0.05, so we fail to reject the null hypothesis. This suggests us that there is no statistical importance between mean house price and a number of bedrooms less than 3 and greater than 3.

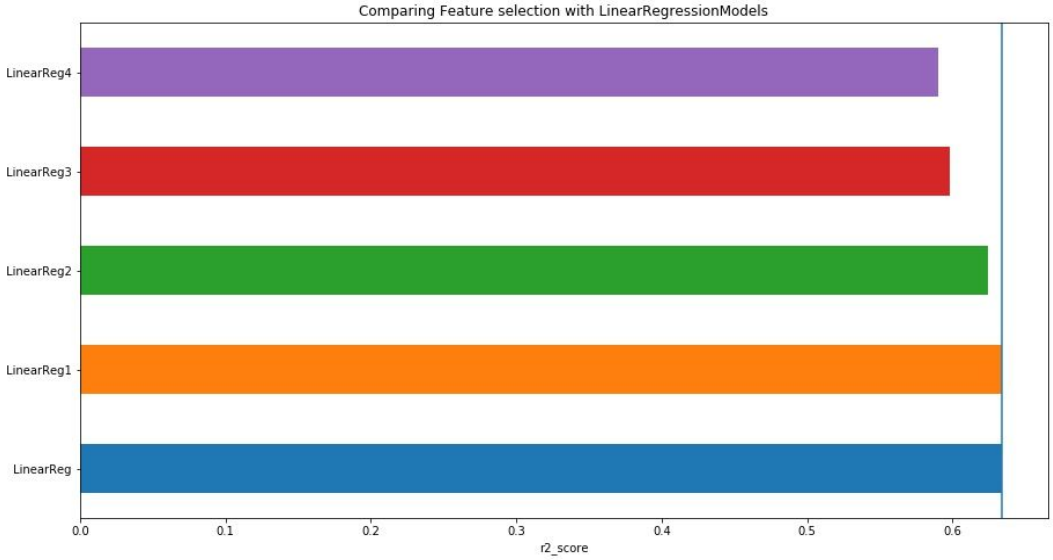
Machine Learning

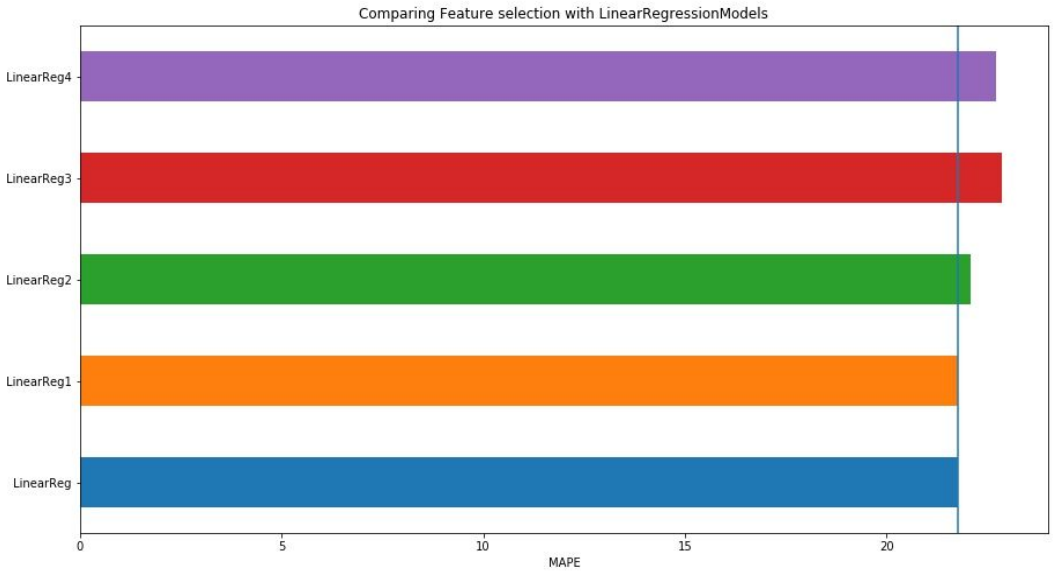
In this project, I use the dataset for supervised machine learning to predict house price. I build 4 different Regression models i.e. Linear Regression, Decision Tree Regression, Gradient Boosted Regression, and Random Forest Regression. I compared the performance of those models using R^2 because it is the ratio between how good our model is vs how good is the naive mean model. I finally recommend the better performing model to predict house price.

First, I divided the data into independent variable X and dependent variable y. Independent variable X is the features I am going to use to predict the target variable y. I dropped price, id and date column from the new_df dataframe to create the variable X. I used price column from the new_df dataframe to create the variable y. There are different metrics used to measures the performance of the Regression models such as Mean squared errors, Root mean squared errors, R-squared score, Mean absolute deviation, Mean absolute percent errors, etc. In this project, I use Root mean squared error and R-squared score to evaluate the performance of the regression model. In order to save the metrics of the model, I created a dataframe named metrics. Next, I split the data into training and testing set. I kept 80% of the randomly selected data as a training set and 20% of the randomly selected data as a testing set. The model will learn or fit using the 80% of the data, and the rest 20% testing data will be used as an unseen future dataset to predict the house price.

I build a Linear Regression Model using the default parameters, and I fit the model using the training dataset. I used X_test data to predict using the model. Then, I calculated Mean squared error (MSE), Root mean squared error (RMSE), R-squared score (r2_score), Mean absolute deviation (MAD), and Mean absolute percent error (MAPE).
Mean Squared Error (MSE): 13276038923.51
Root Mean Squared Error (RMSE): 115221.6947
r2_score: 0.6338
Mean Absolute Error (MAE): 88288.95
Mean Absolute Percent Error (MAPE): 21.78

I used a backward elimination method of feature selection. Feature selection is the process of selecting a subset of relevant features that may improve the performance of the model. First, I removed the one worst attribute from the feature. I removed waterfront because it has the very weak correlation with the price of the house. Then, I removed condition and sqft_lot from the feature set. Then, I removed sqft_lot15, view, and bedroom from the feature set. I also tried a univariate feature selection package called SelectKbest from the sklearn library.





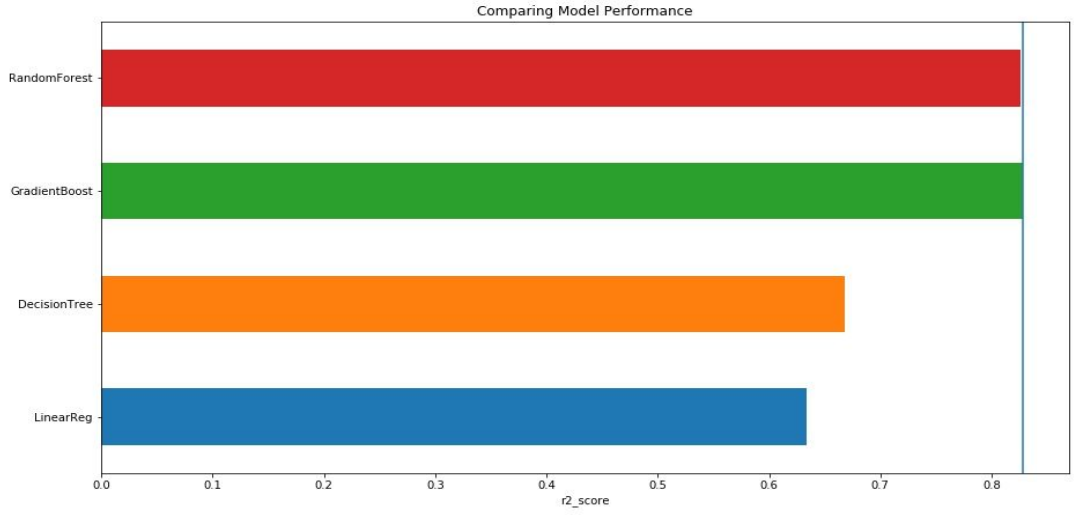
Comparing all different Linear Regression model with feature selection, both barplot suggests us that we should keep all the features to better predict the house price.

I also build a Decision Tree Regressor Model using the default parameters.
Mean Squared Error: 12025421297.16
Root Mean Squared Error: 109660.48
r-squared score : 0.6682893942359164
Mean Absolute Deviation (MAE): 76605.84
Mean Absolute Percent Error (MAPE): 18.45

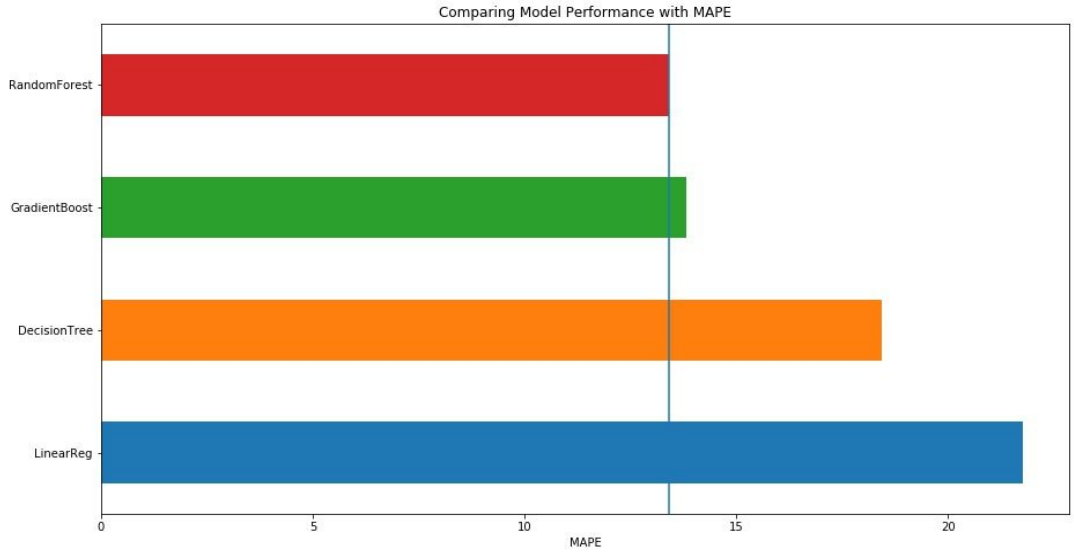
Then, I build a Gradient Boosting Regressor Model using the default parameters.
Mean Squared Error: 6228482868.33
Root Mean Squared Error: 78920.74
r-squared score : 0.8281928113627098
Mean Absolute Deviation (MAE): 56373.65
Mean Absolute Percent Error (MAPE): 13.83

Then, I build a Random Forest Regressor Model using the default parameters.
Mean Squared Error (MSE): 6302252436.45
Root Mean Squared Error (RMSE): 79386.73
r-squared score : 0.8261579431012475
Mean Absolute Deviation (MAE): 54840.62
Mean Absolute Percent Error (MAPE): 13.43

I drew a horizontal bar plot to compare r2_score and Mape of different regressor.

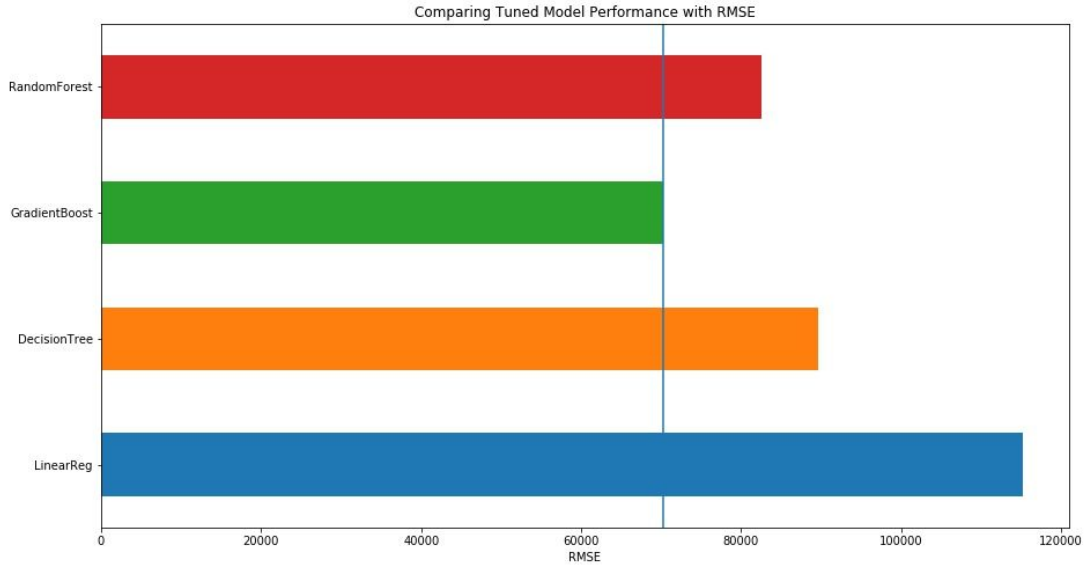


According to the r2_score, Gradient Boosted Regression model is the best performing model.
The random Forest Regression model is the second better performing model for this dataset.

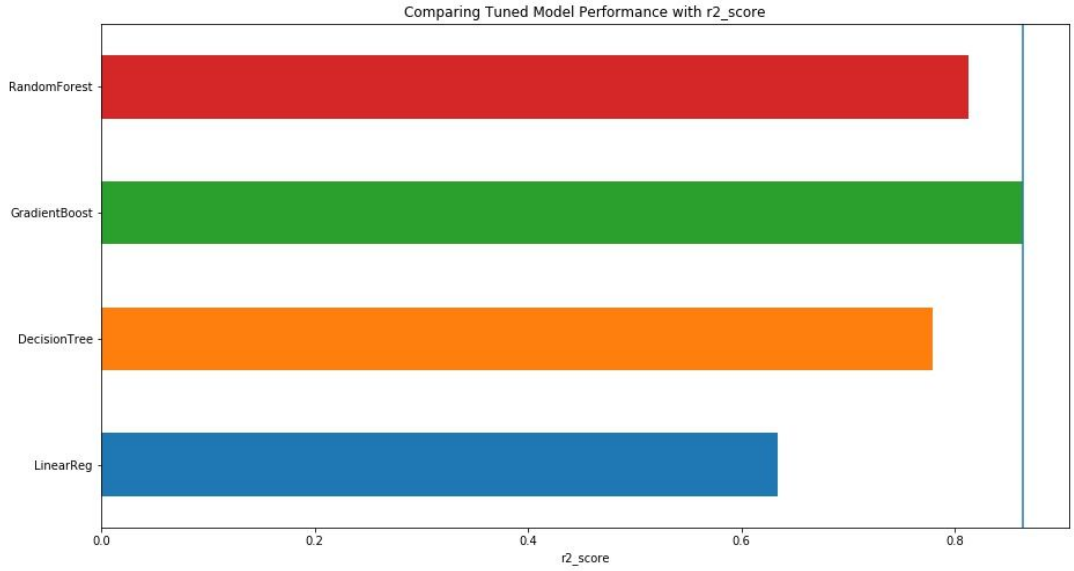


According to the Mean absolute percentage error (Mape), Random Forest Regressor model is the better performing model. Gradient Boosting Regressor model is the second better performing model.

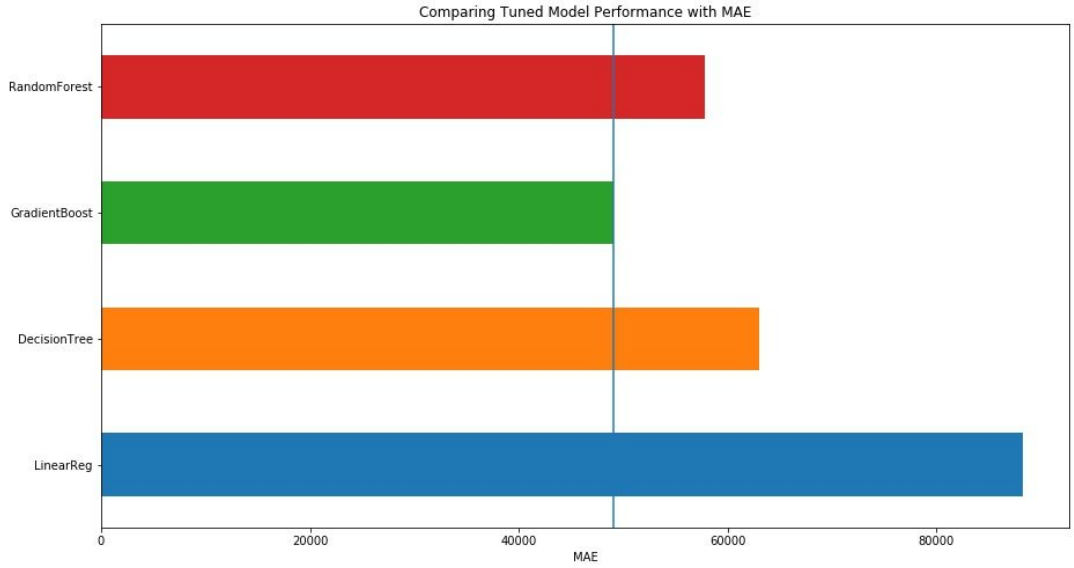
Next, I used GridSearchCV to tune the Hyperparameters of the model to improve the performance of the model. GridSearchCV is a cross-validation method which allows us to use a set of parameters that we want to try with a given model. Each of those parameters is used to perform cross-validation and finally, the best parameters for the model is saved. I created a new dataframe named tuned_metrics to save the metrics of the tuned models. I used the method .get_params() to find all the parameters of the model. For Linear Regression model, I used 'copy_X', 'fit_intercept', and 'normalize' parameters inside the param_grid. The param_grid is a dictionary with parameters names as keys and lists of parameter settings to try as values. I used cv = 5, which is the number of folds used. We can get the best parameter for any Regression model for this data set using the .best_params_ attribute of GridSearchCV. I tuned the Linear Regression model and Decision Tree Regressor model using GridSearchCV. Then, I tuned the Gradient Boosting Regressor model and Random Forest Regressor model using RandomizedSearchCV. RandomizedSearchCV helps us to minimize the computation time because GridSearchCV can take very long computational time to for both Gradient Boosting Regressor and Random Forest Regressor.



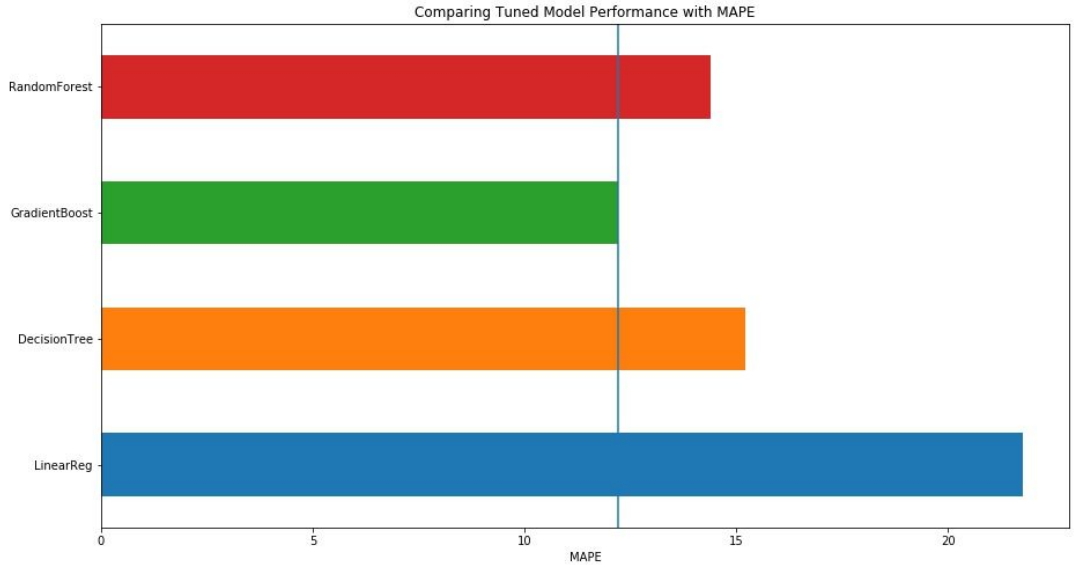
According to the Root Mean Squared Error, Gradient Boost Regression model is the best performing model with the lowest error 115222.



According to the r2_score, Gradient Boost Regression model is the best performing model with the highest score of 0.86.



According to the Mean Absolute Error (MAE), Gradient Boost Regression model is the best performing model with lowest error 49098.4.



According to the Mean absolute percentage error (MAPE), Gradient Boost Regression model is the best performing model with lowest percentage error 12%. All of the metrics suggest that Gradient Boosted Regression model is the better performing model for this dataset.

Gradient Boosting Regression model is a good model to predict house price because it is better than a random guess and it outperforms the other three Regression Model. The model may be improved in the future with more data collection. Many other Regression models which are not

included in this project can also be built and tried. I would recommend this Gradient Boosting Regression model to predict house price.