A Study to Determine the Accurate Algorithm for House Price Prediction

Noreen Chihora

Oklahoma State University

noreen.chihora@okstate.edu

Graduate Student in Business Analytics and Data Science at Oklahoma State University.

Sushma Reddy Mandhadapu

Oklahoma State University

Sushma_reddy.mandhadapu@okstate.edu

Graduate Student in Business Analytics and Data Science at Oklahoma State University. An experienced engineering professional in Information Technology and Finance.

ABSTRACT

The housing market can be overwhelming for both buyers and sellers. The process of finding a dream home can involve hours or months of research. When it comes to housing prices, different factors that affect the prices of houses. It is hard to fully predict the price of a house with 100% accuracy, or even 90% accuracy. There are factors like the economy, number or bedroom and bathroom or location that come into effect. This paper compares different machine learning algorithms that include Linear Regression, K-Nearest Neighbor, Decision Trees, XGBoost and Random Forests to predict the prices of houses. This project uses a Kaggle dataset for King County, USA which has 21,613 records and 21 variables including the ID variable. The study will determincd32dce which variables are important in predicting the prices for houses. The accuracy of each model will be measured using root mean square error. The paper also reviews performance of the models using ROC Index and the confusion matrix. The analysis will be performed using python programming, SAS Enterprise Miner and Tableau for predictive modeling and visualizations respectively.

INTRODUCTION

Everywhere across the United States and all over the world, houses are being bought and sold. According to document prepared by the Congressional Research Service (CSR), the housing market and real estate plays an important role in the economy of the United States. An article by the Bank of England talks about how the housing market is closely related to consumer spending. It says that when house prices go up, homeowners tend to spend more, because they are confident in the value of their property but when the prices go down, they become wary of the resale price and how their house might be worth less than their mortgage and in turn spend less. CSR reported that "consumer spending makes up roughly 70% of the economy". This is a huge portion of the economy. When the economy is doing well, the demand for houses increases and so do the prices. Although buying or selling existing homes does not count towards a country's GDP, costs associated with those homes still contribute to the GDP. These costs can include remodeling costs, furniture buying, closing costs etc. In 2020, CSR reported that spending on new homes, remodeling and broker's fees accounted for 4.2% of the GDP and if we also include rental properties, it goes up to 17.5% of the GDP. This short overview shows how important the housing market is to the economy and it is only right for consumers to get fair prices for their homes as this is a big financial move. House prices are hard to predict because we cannot control the factors that affect the prices of houses it goes up to but with the help of machine learning algorithms, we can somewhat try to get close and help home buyers, and sellers.

In this paper, we will use Multiple Linear Regression, XGBoost, Random Forest, Decision Trees, and K- Nearest Neighbor machine learning models to predict the prices of homes.

LITERATURE REVIEW

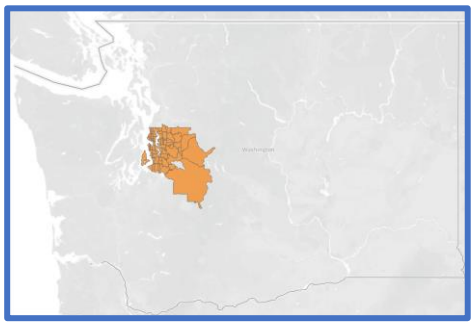
Many studies have been done on predicting housing prices using different machine learning algorithms over the years. A paper by Alyousfi(2018) compared seven different algorithms, Linear Regression, Nearest Neighbors, Support Vector Machine Learning, Decision Tree, Neural Networks, Random Forest and Gradient boosting and uses MAE to decide on the best algorithm. Alyousfi used Kaggle data for Ames, Iowa with dates ranging between 2006 and 2010. The data contains 82 variables and 2,930 records. The author cleaned the data and dealt with missing values and outliers before proceeding to explore relationships between the variables and the target which was the sales price of a property. Alyousfi used feature engineering to decide on the

variables that would be used in the models, dropping one of the variables that were highly correlated to avoid multicollinearity. The categorical variables were given numbers so that they could be used in the predictions as well. The data was split with 75% of the data belonging to the training data and 25% to the validation data. The best model was determined to be XGBoost with the worst being KNN.

Wu(2017) used Support Vector Regression to predict housing prices. The writer used Kings County Data. Wu started by exploring the data and looking at the correlation of the variables before moving on to the predictions. The paper focused on using different feature selection methods to figure out the best method to select important features and gave the best results in the prediction model. R square score, MAE, MSE and RMSE were used to evaluate the best model. Wu performed SVR without feature selection, with feature extraction using PCA, and with feature selection. Wu then did log transformations on the models and compared the results. The results concluded that there is no difference in performance for both feature extraction and feature selection after the log transformation was done.

DATA CLEANING

Before we can work with our data, we must get to know the data. The Kings County, WA data was obtained from kaggle.com. The dataset has 21,613 records and 21 variables including the ID variable. It includes homes sold between May 2014 and May 2015. The data involves some categorical variables and continuous variables.



Zip code distribution

This image gives a general idea of where the houses in this data frame are located in the state of Washington. Below is an overview of the data in the each of the columns in the data frame:

id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade
7129300520	20141013T000000	221900	3	1.00	1180	5650	1.00	0	0	3	7
6414100192	20141209T000000	538000	3	2.25	2570	7242	2.00	0	0	3	7
5631500400	20150225T000000	180000	2	1.00	770	10000	1.00	0	0	3	6
2487200875	20141209T000000	604000	4	3.00	1960	5000	1.00	0	0	5	7
1954400510	20150218T000000	510000	3	2.00	1680	8080	1.00	0	0	3	8

sqft_above	sqft_basement	yr_built	yr_renovated	zipcode	lat	long	sqft_living15	sqft_lot15
1180	0	1955	0	98178	47.51	-122.26	1340	5650
2170	400	1951	1991	98125	47.72	-122.32	1690	7639
770	0	1933	0	98028	47.74	-122.23	2720	8062
1050	910	1965	0	98136	47.52	-122.39	1360	5000
1680	0	1987	0	98074	47.62	-122.05	1800	7503

Overview of the Data

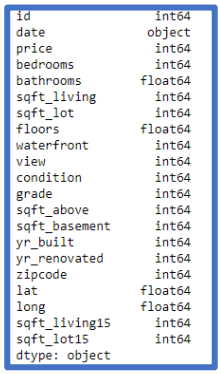
Data Dictionary

To be able to understand the data, a data dictionary is shown below:

Variable	Description
id	Unique identifier for a house
date	Date the house was sold
price	Price is the prediction target
bedrooms	Number of bedrooms/houses
bathrooms	Number of bathrooms/bedrooms
sqft_living	Square footage of home
sqft_lot	Square footage of the lot
floors	Total floors/levels in the house
waterfront	House with waterfront view
view	Has been viewed
condition	Overall condition of the house
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	Year built
yr_renovated	Year when the house was renovated
zipcode	Zipcode
lat	Latitude coordinates
long	Longitude coordinates
sqft_living15	The square footage of interior housing living space for the nearest 15 neighbors
sqft_lot15	The square footage of the land lots of the nearest 15 neighbors

The target / independent variable is **price**, and the rest of the variables are predictor/independent variables. Next, we looked at the different datatypes that present initially in the data.



Data Types

Looking at the data types, zipcode and waterfront needed to be changed to be non-numerical because we do not use both variables for any meaningful calculations, but we can derive meaningful information from these variables if they are strings. Date is also showing as an object which is the wrong data type.

Statistical Information

	mean	std	min	50%	max
id	4580301520.86	2876565571.31	1000102.00	3904930410.00	9900000190.00
price	540088.14	367127.20	75000.00	450000.00	7700000.00
bedrooms	3.37	0.93	0.00	3.00	33.00
bathrooms	2.11	0.77	0.00	2.25	8.00
sqft_living	2079.90	918.44	290.00	1910.00	13540.00
sqft_lot	15106.97	41420.51	520.00	7618.00	1651359.00
floors	1.49	0.54	1.00	1.50	3.50
waterfront	0.01	0.09	0.00	0.00	1.00
view	0.23	0.77	0.00	0.00	4.00
condition	3.41	0.65	1.00	3.00	5.00
grade	7.66	1.18	1.00	7.00	13.00
sqft_above	1788.39	828.09	290.00	1560.00	9410.00
sqft_basement	291.51	442.58	0.00	0.00	4820.00
yr_built	1971.01	29.37	1900.00	1975.00	2015.00
yr_renovated	84.40	401.68	0.00	0.00	2015.00
zipcode	98077.94	53.51	98001.00	98065.00	98199.00
lat	47.56	0.14	47.16	47.57	47.78
long	-122.21	0.14	-122.52	-122.23	-121.31
sqft_living15	1986.55	685.39	399.00	1840.00	6210.00
sqft_lot15	12768.46	27304.18	651.00	7620.00	871200.00

Statistical Summary

The statistical information is giving us a glimpse into whether the data is balanced or not. More analysis will be performed later with visualizations. From our data we can see that the prices of the houses in Kings County range from $75,000 to $7.7mil with a median of $450,000. The size of the homes ranges from 290sqft to 13,540sqt with a median of 1,910sqft. The condition of the house is scored on a scale of 1 to 5 with the median as 3 and a mean of 3.41. The grade goes from 1 to 13 with a median of 7 and a mean of 7.66. The houses were built between 1900 and 2015 with 1975 being the median year. There is a lot of information that can be seen and learned about the data from the statistics. We can see that there is also data that does not make sense like the min year that a house was renovated is 0, or the maximum number of bedrooms is 33. This could be true for number of bedrooms, or it could be information we might have to look out for whilst doing the analysis.

	unique	top	freq
date	372	20140623T000000	142
waterfront	2	0	21450
zipcode	70	98103	602

Data Uniqueness

We can see that waterfront has 2 unique values which are 0 for no waterfront view and 1 for waterfront view. 21,450 properties in this data do not have waterfront view. It can also be noted that there are 70 unique zipcodes and most of the properties listed are in zipcode 98103. ID is not an important variable in the prediction, so it is dropped from the table

Missing Data

Missing data can be problematic especially when creating predictive models. Our data does not have any missing values.

```
#missing values
"There are " + str(kings_county_data.isna().sum().sum()) + " missing values in the data"
```

```
'There are 0 missing values in the data'
```

Missing Data

Duplicated Data

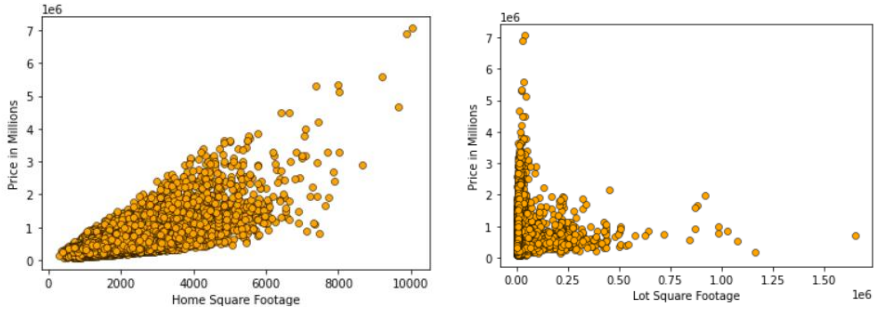
It is also important to check for duplications in the data as this could result in the wrong prediction models that would not work well with other test data. We use a combination of ID and date to check for duplication. The are no duplicated rows in the data as seen by the results below:

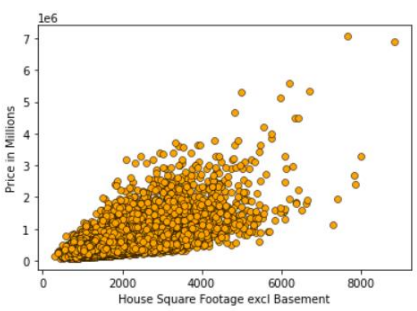
```
#check for duplicated variables
kings_county_data.duplicated(subset = ["id", "date"]).sum() == 0
```

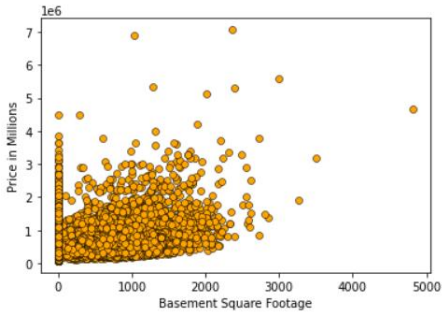
```
True
```

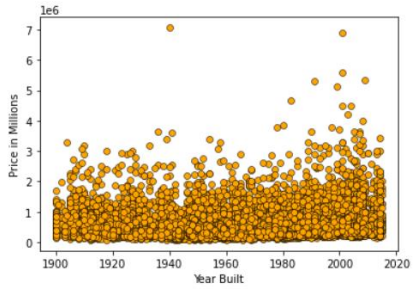
Outlier Analysis

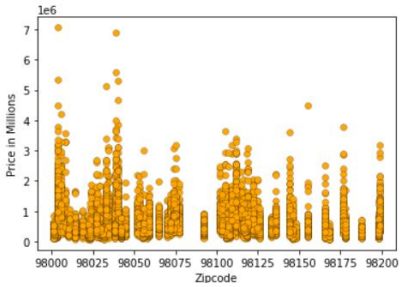
We used scatter plots with the independent variables plotted against the target variable for easier visibility of any outliers in the data.

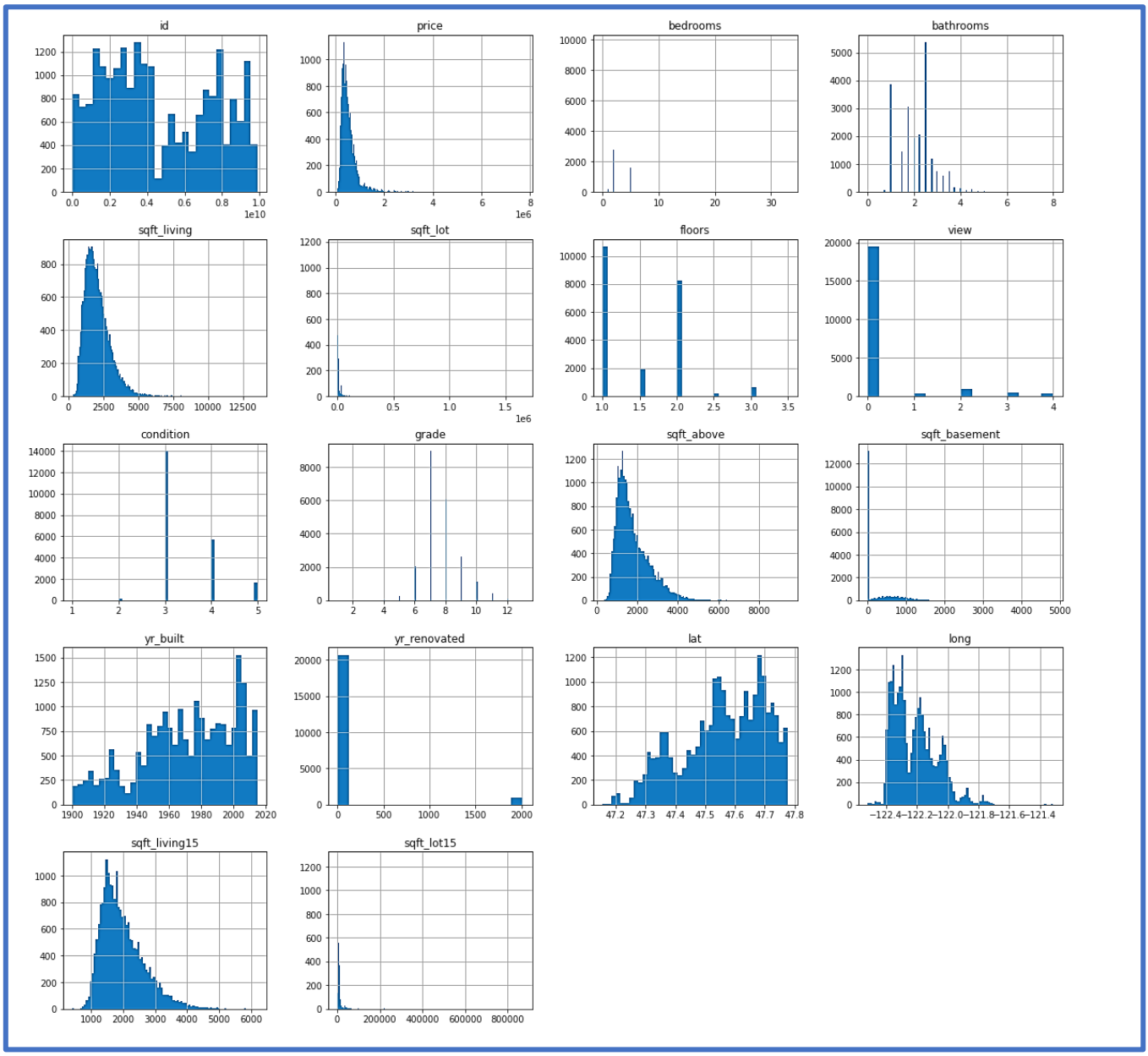



1e6
Price in Millions
House Square Footage excl Basement


1e6
Price in Millions
Basement Square Footage

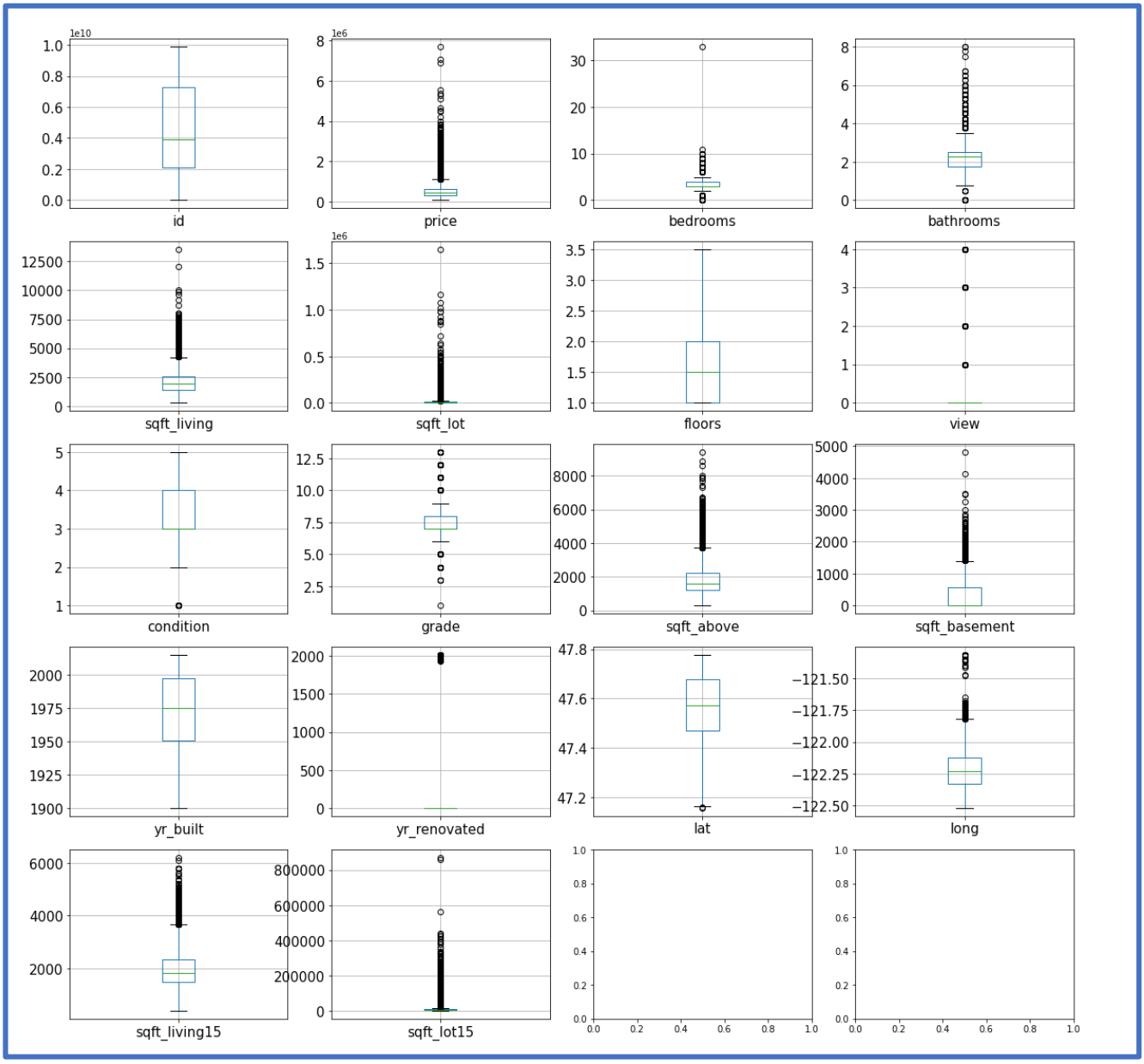

1e6
Price in Millions
Year Built


1e6
Price in Millions
Zipcode



Histogram

Histograms were plotted to to look for outliers either on the lower end or upper end of the data. We are also able to see that the categorical variables and continuous variables. Some of the categorical variables are floors, view and condition. Examples of continuous variables are price, sqft_living and sqft_above. We can also see that the variables are unimodal like sqft_living15, price, sqft_above and yr_built. Histograms also allow us to see skewedness of the continuous variables. The variables like price, sqft_living, sqft_above etc are all right skewed. The histograms give insight into the data but it is hard to see the outliers so we plotted box plots for the numerical data.

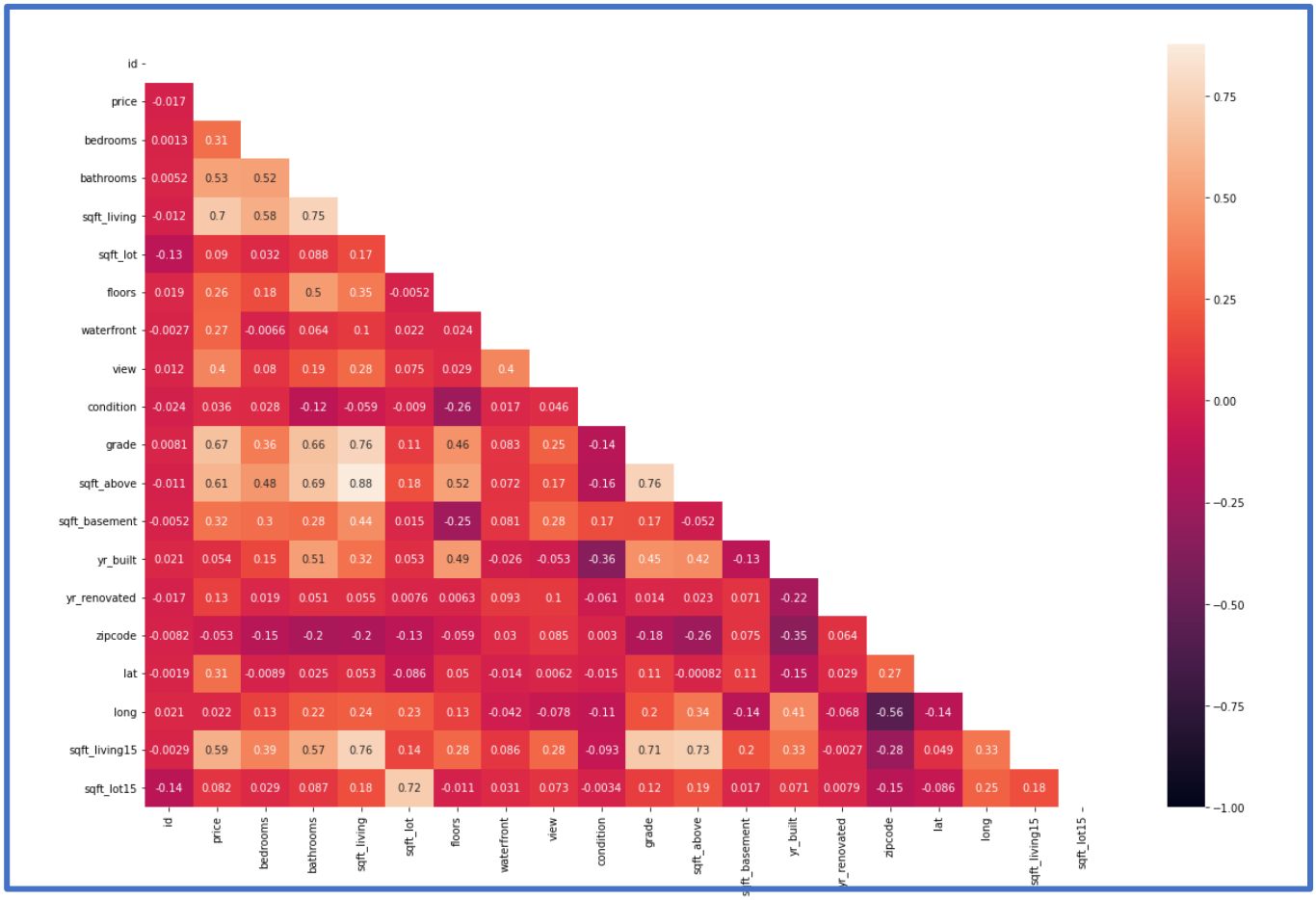


Outlier Analysis

The data shows some outliers but having knowledge about the outliers, they do not seem out of place except for the number of bedroom where 33 seems erroneous and probably wrong.

DATA PREPARATION

We did not have any duplicated or missing values so no preparation is to be made in that regard. However we do need to change date from object to date data type.



Heatmap

Looking at the heatmap, and considering a correlation greater than 0.7 as showing high multicollinearity, sqft_living and sqft_above are highly correlated. Other variables considered were sqft_lot, grade, sqft_lot15 and sqft_living15. Multicollinearity is not good in a model because it can reduce precision, which weakens the model. To decide on which of the variables to remove from the dataframe, VIF values were used. Below are the VIF results:

Variable	VIF
sqft_living	5.177023
sqft_above	4.812970
grade	2.828265
sqft_living15	2.680391
sqft_lot15	2.100967
sqft_lot	2.080651

VIF Results

From these results only sqft_living has a VIF greater than 5. We removed it and checked the results to see if there is any improvement.

Variable	VIF
sqft_lot	2.079515
grade	2.690580
sqft_above	2.880500
sqft_lot15	2.100966
sqft_living15	2.491762

sqft_living removed

It can be noted that there was great improvement of VIF for sqft_above and slight improvement in the other variables when sqft_living is removed. Now all the VIF is withing acceptable ranges and the derived model can yield more statistically significant results.

DATA TRANSFROMATIONS

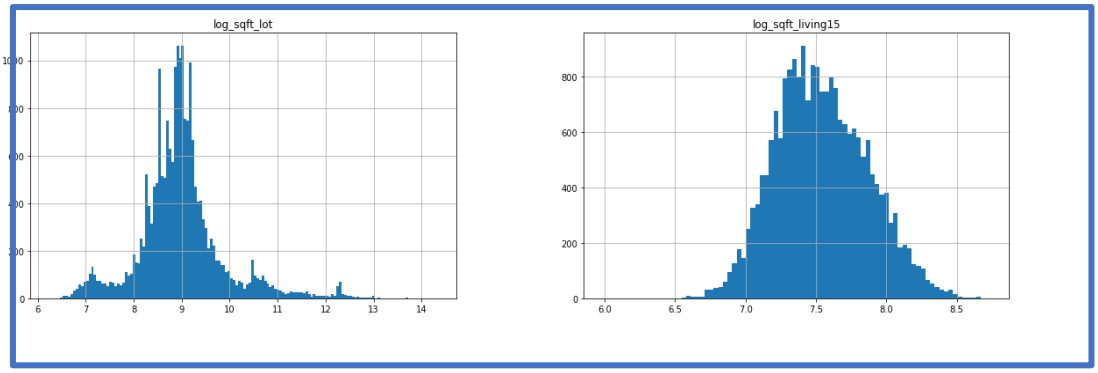
Earlier, we noted that the continuous variables were right skewed. To improve the perfomance of our model, we transform the variables so they look like a normal distribution. The transformed variables will be used for Multiple Linear Regression. We found the skewness of the variables and transformed variables which have high skewness.

The variables with high skewness were sqft_log and sqft_living_15

```
print (inputs.sqft_above.skew())
print(inputs.sqft_lot.skew())
print(inputs.sqft_lot15.skew())
print(inputs.sqft_living15.skew())
print(inputs.sqft_basement.skew())
print(target.skew())
```

```
1.4466644733818372
13.060018959031755
9.506743246764398
1.1081812758966967
1.5779650555996247
4.024069144684712
```

The transformation is log-transgormation and the results are shown below:



Normalized Variables

PREDICTION TYPES

In this paper we will be using 5 machine learning models.

Multiple Linear Regression

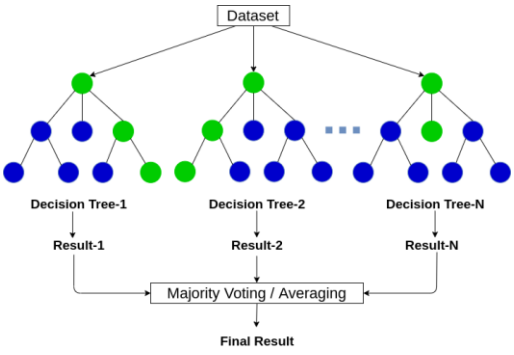
Multiple linear regression is used to analyze the relationship between one dependent variable and multiple independent variables.

XG Boost

XG Boost which stands for Extreme Gradient Boosting is a library that focuses on computational speed and model performance. It supports Gradient boosting, Stochastic Gradient Boosting and Regularized Gradient Boosting.

Random Forest

Random Forest is a machine learning algorithm that combines the output of multiple decision trees and comes up with a single result.



Decision Tree

Decision Tree is a model that predicts the value of a target variable. It makes a prediction based on the previous node.

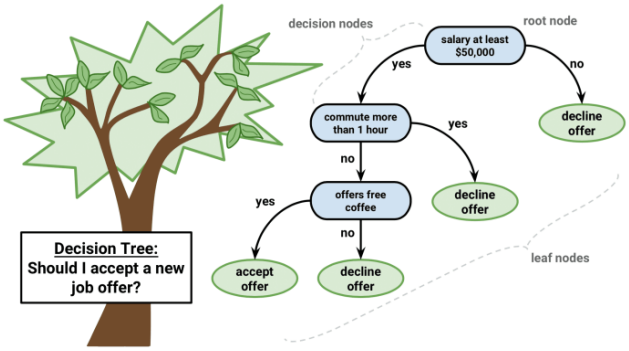


image source: towardsdatascience.com

K-Nearest Neighbor

K-Nearest Neighbor is an algorithm that estimates a data point's likelihood of belonging to one or more groups depending on which data points are closest to it.

MODELING

Now that we explained the model that we will be using, we will start the modeling process. We started by separating the target variables and the independent variables.

```
from sklearn.model_selection import train_test_split
```

```
X_train, X_test, y_train, y_test = train_test_split(inputs, target, test_size=0.3, random_state=123)
```

We split our data into two, a training sample consisting of 70% of the data and a validation set with the remaining 30% of the data. We now have four subsets, X_train, X_test, y_train, and y_test. X_train and y_train are used to train the data. X_test and y_test is used to validate the data.

Multiple Linear Regression:

To build the multiple linear regression model we used sklearn package that is available in python. Using the LinearRegression() function we have built the Multiple regression model with Price as target variable and 'bedrooms', 'bathrooms', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_lot15',

'log_sqft_lot', 'log_sqft_living15' as independent variables. We were able to predict the price with an accuracy of 71 percent. We estimated the accuracy based on the R-Square value.

```
#Model Evaluation
r2_linear = m.r2_score(y_test,y_test_pred)
mae_linear =m.mean_absolute_error(y_test,y_test_pred)
print('R^2: ', m.r2_score(y_test,y_test_pred) )
print('Adjusted R^2: ', 1-(1-m.r2_score(y_test, y_test_pred))*(len(y_test)-1)/(len(y_test)-X_test.shape[1]-1))
print('MAE: ', m.mean_absolute_error(y_test,y_test_pred))
print('MSE: ', m.mean_squared_error(y_test,y_test_pred))
print('RMSE: ', np.sqrt(m.mean_squared_error(y_test,y_test_pred)))
```

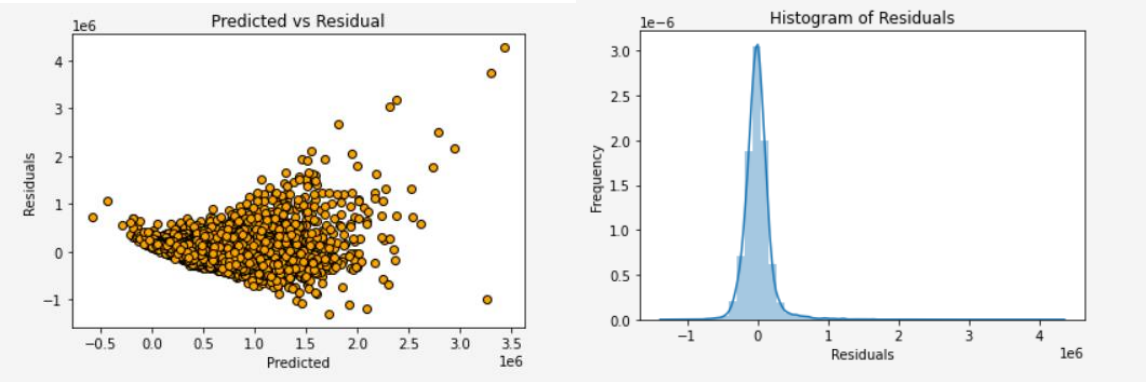
```
R^2:  0.7116014759686246
Adjusted R^2:  0.7108432367313011
MAE:  126090.30821597051
MSE:  39164408127.59936
RMSE:  197899.99526932626
```

We checked the following Multiple Linear Regression Assumptions while building the model:

- No multicollinearity
- Linear relationship between explanatory and response variables
- Homoscedasticity of error terms
- Normal distribution of model residuals

The following picture show the difference of price between the actual price and the predicted price.

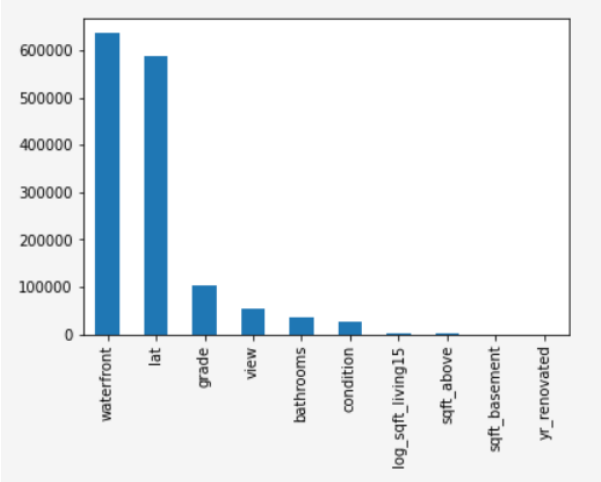
The below pictures show the residual distribution:



From the graphs we could see that the error distribution was almost normal.

Feature Selection:

According to Multiple Linear regression algorithm, the top five features that having high impact on the price were waterfront, lat, grade, view and bathrooms.



Decision Tree:

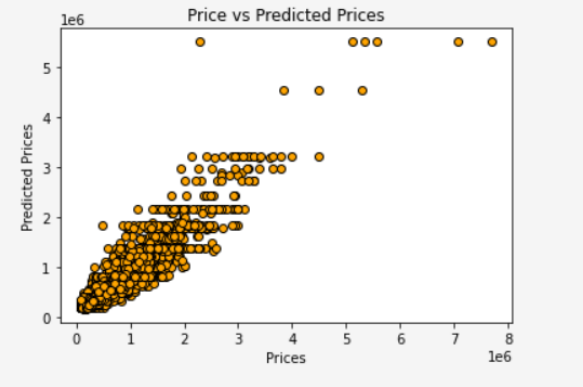
We used Decision Tree Regressor to predict the price with Decision Tree Regression algorithm. We used it, as it is a regression problem and we are trying to predict price, a continuous variable. The target variable is price and the input variables were: 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated' , 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15'. For building the model we used

the sklearn library available in python. DecisionTreeRegressor() function was used to build the model. We were able to predict the house price of validation data with an accuracy of 77.6 percent. We estimated the accuracy based on the R-Square error value.

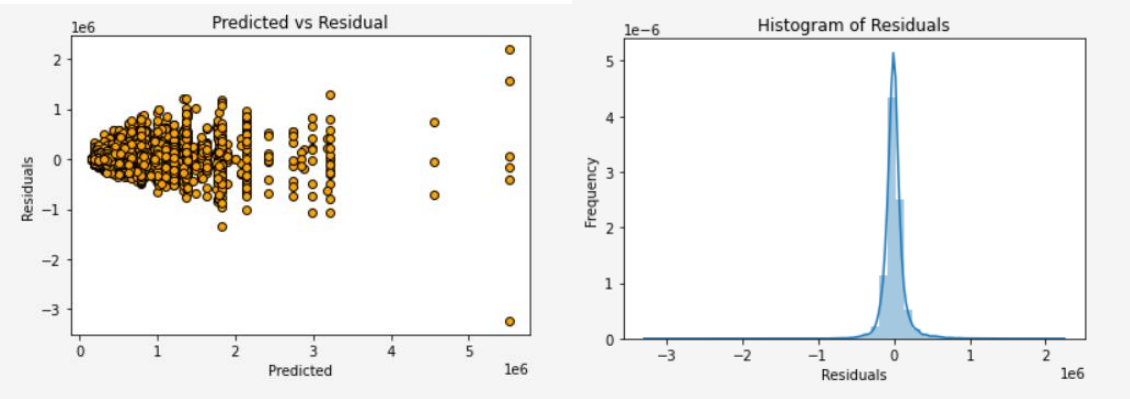
```
#Model Evaluation
r2_decision = m.r2_score(y_test,y_test_pred)
mae_decision =m.mean_absolute_error(y_test,y_test_pred)
print('R^2: ', m.r2_score(y_test,y_test_pred) )
print('Adjusted R^2: ', 1-(1-m.r2_score(y_test, y_test_pred))*(len(y_test)-1)/(len(y_test)-X_test.shape[1]-1))
print('MAE: ', m.mean_absolute_error(y_test,y_test_pred))
print('MSE: ', m.mean_squared_error(y_test,y_test_pred))
print('RMSE: ', np.sqrt(m.mean_squared_error(y_test,y_test_pred)))
```

```
R^2:  0.7766128495917911
Adjusted R^2:  0.775990890008288
MAE:  96878.45933017887
MSE:  30335888709.668694
RMSE:  174172.00897293657
```

The following picture show the difference of price between the actual price and the predicted price.



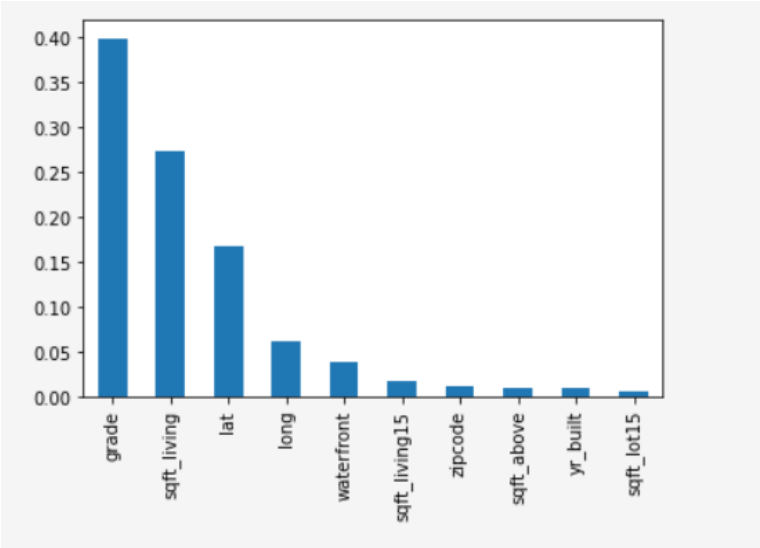
The below pictures show the residual distribution:



From the graphs we could see that the error distribution was almost normal.

Feature Selection:

According to decision tree regressor algorithm the top five features that having high impact on the price were grade, sqft_living, lat, long and waterfront.



Random Forest:

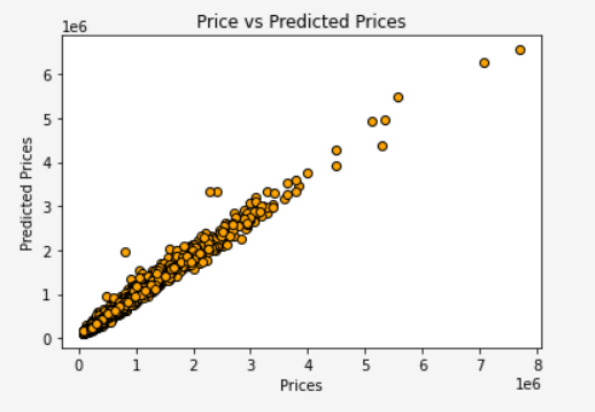
We used Random Forest Regressor to predict the price with Random Forest algorithm. We used it, as it is a regression problem and we are trying to predict price, a continuous variable. The target variable is price and the input variables were: 'bedrooms', 'bathrooms', 'sqft_living',

'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated' , 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15'. For building the model we used the sklearn library available in python. RandomForestRegressor() function was used to build the model. We were able to predict the house price of validation data with an accuracy of 87.8 percent. We estimated the accuracy based on the R-Square error value.

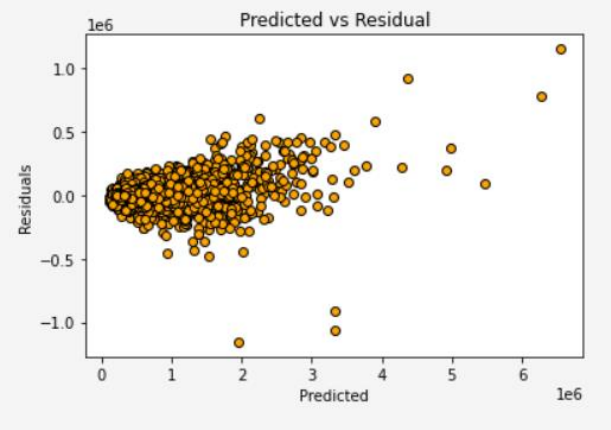
```
#Model Evaluation
r2_random = m.r2_score(y_test,y_test_pred)
mae_random =m.mean_absolute_error(y_test,y_test_pred)
print('R^2: ', m.r2_score(y_test,y_test_pred) )
print('Adjusted R^2: ', 1-(1-m.r2_score(y_test, y_test_pred))*(len(y_test)-1)/(len(y_test)-X_test.shape[1]-1))
print('MAE: ', m.mean_absolute_error(y_test,y_test_pred))
print('MSE: ', m.mean_squared_error(y_test,y_test_pred))
print('RMSE: ', np.sqrt(m.mean_squared_error(y_test,y_test_pred)))
```

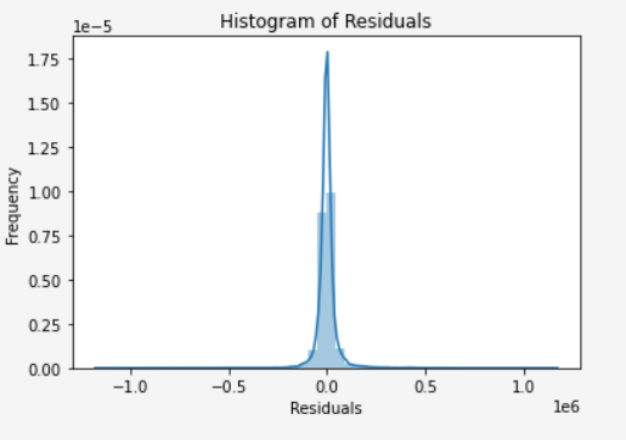
```
R^2:  0.8787153381658427
Adjusted R^2:  0.8783776546526153
MAE:  69891.34926459637
MSE:  16470410213.243818
RMSE:  128337.09601375519
```

The following picture show the difference of price between the actual price and the predicted price.



The below pictures show the residual distribution:

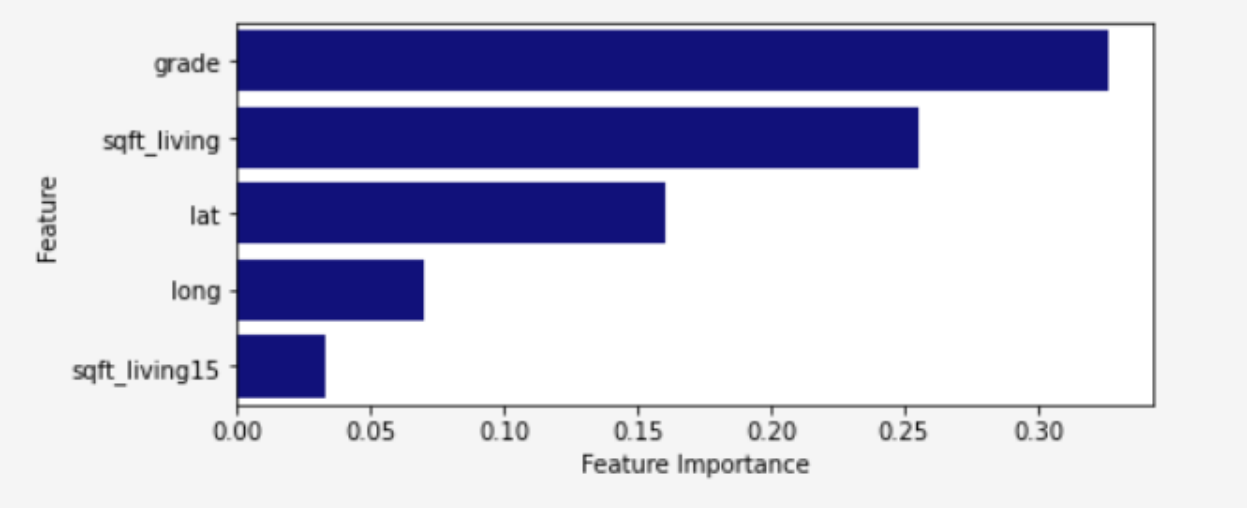




From the graphs we could see that the error distribution was almost normal.

Feature Selection:

According to random forest regressor algorithm the top five features that having high impact on the price were grade, sqft_living, lat, long and sqft_living15.



XGBOOST:

We used XGB Regressor to predict the price with xgboost Regression algorithm. We used it, as it is a regression problem and we are trying to predict price, a continuous variable. The target variable is price and the input variables were: 'bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot',

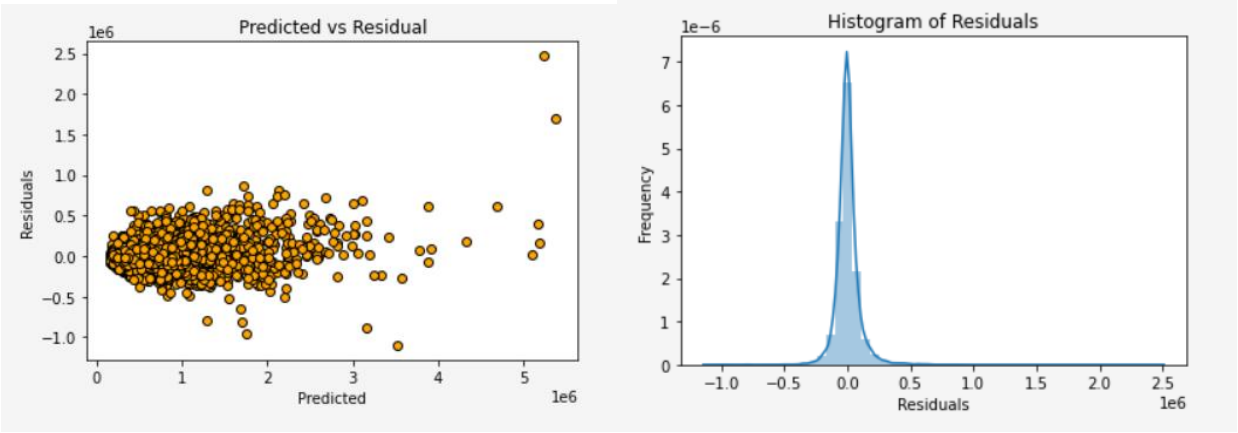
'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated' , 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15'. For building the model we used the sklearn library available in python. XGBRegressor() function was used to build the model. We were able to predict the house price of validation data with an accuracy of 89.9 percent. We estimated the accuracy based on the R-Square error value.

```
#Model Evaluation
r2_xgb = m.r2_score(y_test,y_test_pred)
mae_xgb =m.mean_absolute_error(y_test,y_test_pred)
print('R^2: ', m.r2_score(y_test,y_test_pred) )
print('Adjusted R^2: ', 1-(1-m.r2_score(y_test, y_test_pred))*(len(y_test)-1)/(len(y_test)-X_test.shape[1]-1))
print('MAE: ', m.mean_absolute_error(y_test,y_test_pred))
print('MSE: ', m.mean_squared_error(y_test,y_test_pred))
print('RMSE: ', np.sqrt(m.mean_squared_error(y_test,y_test_pred)))
```

```
R^2:  0.8993560310288841
Adjusted R^2:  0.8990758158020503
MAE:  68100.39168530228
MSE:  13667412097.91146
RMSE:  116907.70760694721
```

The following picture show the difference of price between the actual price and the predicted price.

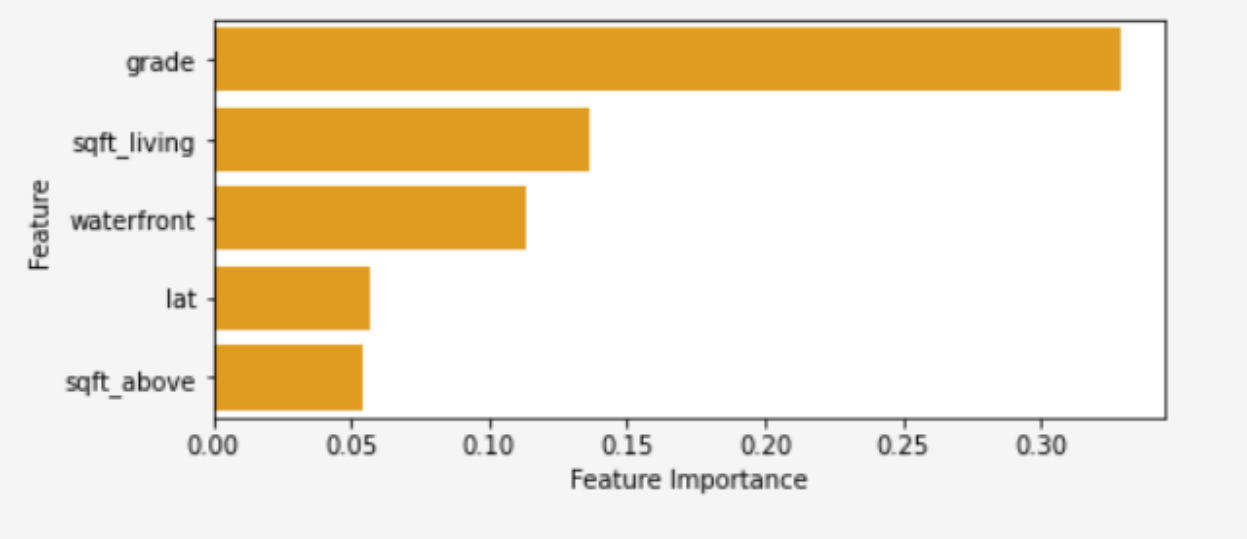
The below pictures show the residual distribution:



From the graphs we could see that the error distribution was almost normal.

Feature Selection:

According to xgboost regressor algorithm the top five features that having high impact on the price were grade, sqft_living, waterfront, lat and sqft_above.



K – Nearest Neighbor:

We used KNeighbors Regressor to predict the price with K-Nearest Neighbor (Knn) algorithm. We used it, as it is a regression problem and we are trying to predict price, a continuous variable. The target variable is price and the input variables were: 'bedrooms', 'bathrooms', 'sqft_living',

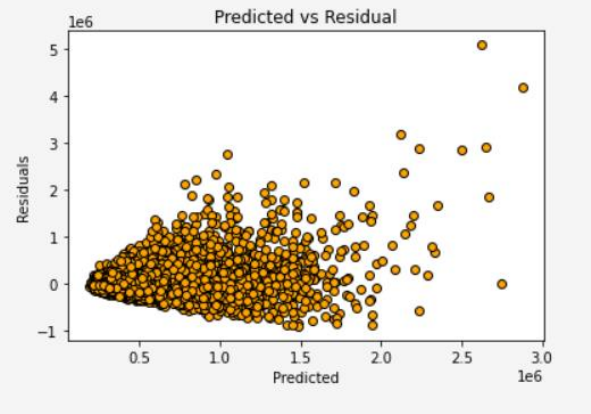
'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated' , 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lot15'. For building the model we used the sklearn library available in python. KNeighborsRegressor() function was used to build the model. We were able to predict the house price of validation data with an accuracy of 52 percent. We estimated the accuracy based on the R-Square error value.

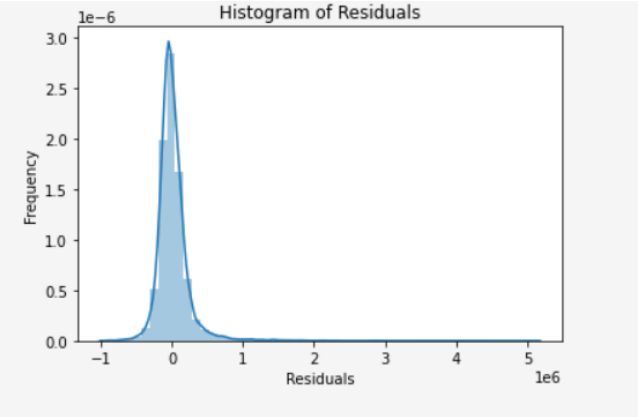
```
#Model Evaluation
r2_knn = m.r2_score(y_test,y_test_pred)
mae_knn =m.mean_absolute_error(y_test,y_test_pred)
print('R^2: ', m.r2_score(y_test,y_test_pred) )
print('Adjusted R^2: ', 1-(1-m.r2_score(y_test, y_test_pred))*(len(y_test)-1)/(len(y_test)-X_test.shape[1]-1))
print('MAE: ', m.mean_absolute_error(y_test,y_test_pred))
print('MSE: ', m.mean_squared_error(y_test,y_test_pred))
print('RMSE: ', np.sqrt(m.mean_squared_error(y_test,y_test_pred)))
```

```
R^2:  0.5201751495935494
Adjusted R^2:  0.5188392103348771
MAE:  155583.32168414557
MSE:  65160029283.08371
RMSE:  255264.6259924859
```

The following picture show the difference of price between the actual price and the predicted price.

The below pictures show the residual distribution:





From the graphs we could see that the error distribution was almost normal.

MODEL COMPARISON:

The performance of the five models were done based on the R-Square value of the model. Based on the R-Square value we were calculating the accuracy with which the model predicted the price. The model with high R-Square value is considered as the best model and the model with the least R- Square value is considered as the worst model for the given data.

Model	R-squared Score	MAE
XGBoost	89.935603	68100.391685
Random Forest	87.871534	69891.349265
Decision Tree	77.661285	96878.459330
Linear Regression	71.160148	126090.308216
KNN	52.017515	155583.321684

The above picture has the R-Squared Score and Mean Absolute Error for all the five models namely, XGBoost, Random Forest, Decision Tree, Linear Regression and KNN. The models were printed in the descending order of their performance.

Conclusion:

For the Kings County Housing dataset, XGBoost algorithm is performing significantly better than the rest of the algorithms namely Random Forest, Decision Tree, Linear Regression and

Knn. Random Forest was performing with an accuracy of 87 percent in predicting the house price, but we have observed that the model is overfitting. So, it is better to not go with the default Random Forest regressor available in python while predicting the house price for the given data. Therefore, the next best algorithm to predict the house price for king's county data is Decision Tree algorithm with an accuracy of 77 percent. Knn is performing worst with the accuracy of only 52 percent in predicting the house price.

The top five features impacting the house price according to the XGBoost algorithm were grade, sqft_living, waterfront, lat and sqft_above. So from this we can tell that the price of the house is highly dependent on the overall grade given to the housing unit, based on King County grading system square footage of home, waterfront view of home, latitude of the location and the Square footage of house apart from basement.

Future Analysis:

As the Random Forest algorithm is overfitting the data, we can analyze further to find the reasons for the overfitting and work on finding the accuracy of the model when it is not overfitting. As some of the variables were having outliers, we can check the different model's performance after dealing with outliers. We have not removed are modified any of the outlier because there might be changes that the values might not fit in the data but were true. Example, there might chances of a small house having high price because of the architecture. So left the outliers as they were.

References:

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

https://sgp.fas.org/crs/misc/IF11327.pdf

https://ammar-alyousfi.com/assets/documents/house-price-prediction.pdf

https://scholarworks.sjsu.edu/cgi/viewcontent.cgi?article=1540&context=etd_projects

https://www.bankofengland.co.uk/knowledgebank/how-does-the-housing-market-affect-the-economy

http://www.stat.yale.edu/Courses/1997-98/101/linmult.htm

https://www.ibm.com/cloud/learn/random-forest

https://www.ibm.com/cloud/learn/neural-networks

https://medium.com/swlh/rudimentary-data-cleaning-techniques-using-king-county-wa-housing-dataset-f7716bdf827e

https://towardsdatascience.com/targeting-multicollinearity-with-python-3bd3b4088d0b

https://www.youtube.com/watch?v=uI1wEw9AXwc