ILLINOIS INSTITUTE OF TECHNOLOGY

# House Price Forecasting Using Machine Learning Models

**Karan Sathiayan**
**A20469856**
*ksathiayan@hawk.iit.edu*

**Deepshika Chandrasekar**
**A20471561**
*dchandrasekar@hawk.iit.edu*

December 2nd 2022

Prof. Yan Yan
Machine Learning (CS 584)

**Karan Sathiayan**
**A20469856**
*ksathiayan@hawk.iit.edu*

**Deepshika Chandrasekar**
**A20471561**
*dchandrasekar@hawk.iit.edu*

## ABSTRACT

Real estate is a dynamic industry with constantly fluctuating market conditions. Real estate agents use this to their advantage by increasing the prices exorbitantly. This affects the buyers who are looking to purchase a new home based on their budgets. The objective of this project is to implement a machine learning model that forecasts the actual price of the houses based on key attributes of the house. Many aspects must be considered when projecting house prices and attempting to estimate effective house pricing for buyers based on their budget and preferences.

Machine Learning is a scientific study of algorithms that computers use to perform a certain task without explicitly programmed to do so. This project involves predictions of house prices using various regression techniques like - ‘Multiple Linear Regression’, ‘Polynomial Regression’, ‘Random Forest Regression’. This system assists in determining a beginning price for a property based on neighborhood. Future costs will be predicted by breaking down past market patterns and value ranges as well as recent house renovations.

This strategy will assist people in putting money into a bequest without using a broker. According to the findings of this study, the Random Forest Regression model provides the highest level of accuracy and the least root mean square error (RMSE) value.

*Keywords: Machine Learning, Regression, Random Forest, RMSE*

## 1. INTRODUCTION

Every organization in today's real estate industry is working hard to gain a competitive advantage over their adversaries. Before the advent of machine learning, the price of any housing property was determined by the real estate agents. These agents have managed to delude a lot of buyers by offering them the house for extravagant prices. Furthermore, the market demand for houses was increasing exponentially due to the rise in population, which added on to the agent’s advantage. Hence, there arose a need to simplify the process that would help a fresh buyer to overcome the hurdle of paying unreasonable prices that is caused due to the involvement of the agents.

With the onset of new technology, all the industries are moving towards automation. House price forecasting can also be implemented using machine learning algorithms. Machine Learning develops algorithms and creates models from input data and then utilizes the model to predict values on unseen data. Supervised Learning is a type of Machine Learning model that uses labeled data, while unsupervised learning makes use of unlabeled data for prediction. In this project, the price of the house is forecasted which is a continuous target variable. Thus, a supervised learning **Regression model** is incorporated.

Many hypotheses have emerged because of the contributions of numerous researchers from around the world on the house prices forecasting data [2]. Some of these researchers believe that the physical location and culture of a particular area affect how the property values will rise or fall, whereas some others feel that socioeconomic factors are the driving force behind these price increases. Some researchers feel the average square feet area of surrounding houses play a role in predicting the house price. Thus, this is an emerging research area of interest.

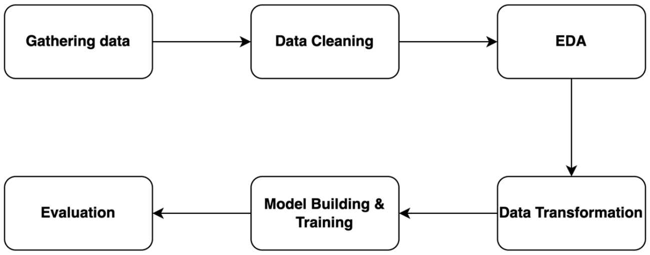
There are two major challenges that one faces in this project. The biggest challenge is to find which features and the optimal number of features that the model must use to fit and predict the price of the house accurately. The other challenge is to find the machine learning algorithm that will be the most effective when it comes to forecasting the price with minimal Root Mean Square Error (RMSE).

## 2. PROPOSED METHODOLOGY

The task of imparting intelligence to the system can be broken down into 6 major steps.

A. **Data gathering** - It is important to collect reliable and quality data so that the machine learning model can find accurate patterns.
B. **Data Cleaning** - Remove unwanted rows, columns, redundant data, missing values
C. **Exploratory Data Analysis** – Understand the descriptive detail of the columns, relation between two columns. Then detect and treat outliers.
D. **Data Transformation** – It includes feature scaling and feature/column type casting.
E. **Model Building** – Split the data into training and testing data sets and build the ML model.
F. **Evaluation** – After training, it is essential to see how this model is fairing on previously unseen data.

These steps are implemented using the Scikit-learn packages in Python language on Jupyter Notebook. In this project, a target variable is forecasted i.e., price of a house based on various features about the house. The ML algorithms used here are: Linear Regression model, Polynomial model, and Random Forest model which are categorized as supervised regression technique.



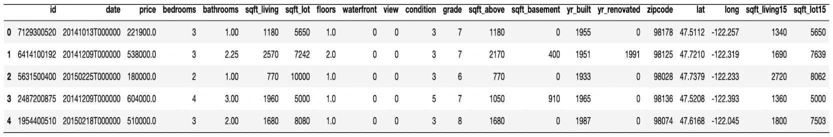
After finding a model that best suits the application, it can be deployed in the real world and favorable insights can be drawn and forecasted.

### 2.1 Dataset Description

The dataset used for this project is *‘House Sales in King County, USA’* obtained from the Kaggle official website [3]. This dataset contains house sale prices of homes sold between May 2014 and May 2015 in Seattle. This dataset has 20 features which help in predicting the price of the house.

**id** - Unique ID for each home sold
**date** - Date of the house sale
**price** - Price of each house sold
**bedrooms** - Number of bedrooms
**bathrooms** - Number of bathrooms (where .5 accounts for a room with a toilet but no shower)
**sqft_living** - Square footage of the apartments interior living space
**sqft_lot** - Square footage of the land space
**floors** - Number of floors
**waterfront** - Apartment overlooking the waterfront or not [0 or 1]
**view** - An index from [0 to 4] of how good the view of the property was
**condition** - An index from [1 to 5] on the condition of the apartment,
**grade** - An index from [1 to 13], explaining the quality level of construction and design
**sqft_above** - The square footage of the interior housing space that is above ground level
**sqft_basement** - The square footage of the interior housing space that is below ground
**yr_built** - The year the house was initially built
**yr_renovated** - The year of the house’s last renovation
**zipcode** - What zip code area the house is in
**lat** - Latitude
**long** - Longitude
**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

This dataset is in csv format. So, it can be loaded to the Python environment.
The preview of the top 5 rows of the dataset is as below -



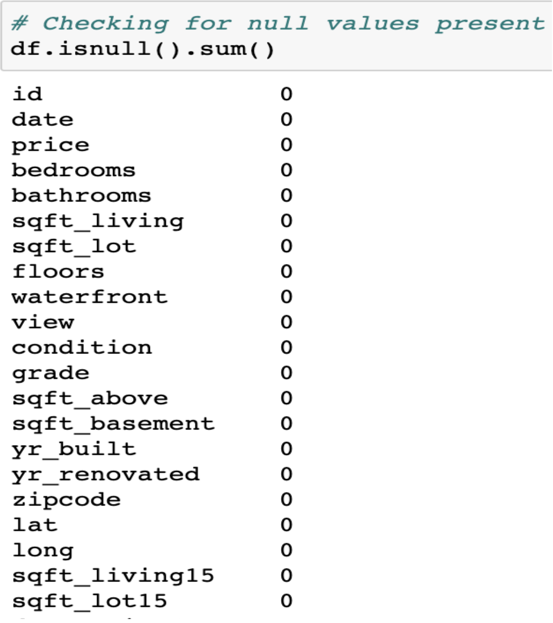
| | id | date | price | bedrooms | bathrooms | sqft_living | sqft_lot | floors | waterfront | view | condition | grade | sqft_above | sqft_basement | yr_built | yr_renovated | zipcode | lat | long | sqft_living15 | sqft_lot15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 7129300520 | 20141013T000000 | 221900.0 | 3 | 1.00 | 1180 | 5650 | 1.0 | 0 | 0 | 3 | 7 | 1180 | 0 | 1955 | 0 | 98178 | 47.5112 | -122.257 | 1340 | 5650 |
| **1** | 6414100192 | 20141209T000000 | 538000.0 | 3 | 2.25 | 2570 | 7242 | 2.0 | 0 | 0 | 3 | 7 | 2170 | 400 | 1951 | 1991 | 98125 | 47.7210 | -122.319 | 1690 | 7639 |
| **2** | 5631500400 | 20150225T000000 | 180000.0 | 2 | 1.00 | 770 | 10000 | 1.0 | 0 | 0 | 3 | 6 | 770 | 0 | 1933 | 0 | 98028 | 47.7379 | -122.233 | 2720 | 8062 |
| **3** | 2487200875 | 20141209T000000 | 604000.0 | 4 | 3.00 | 1960 | 5000 | 1.0 | 0 | 0 | 5 | 7 | 1050 | 910 | 1965 | 0 | 98136 | 47.5208 | -122.393 | 1360 | 5000 |
| **4** | 1954400510 | 20150218T000000 | 510000.0 | 3 | 2.00 | 1680 | 8080 | 1.0 | 0 | 0 | 3 | 8 | 1680 | 0 | 1987 | 0 | 98074 | 47.6168 | -122.045 | 1800 | 7503 |

In this dataset, the price is the target variable also known as dependent variable. The remaining columns are called the feature variable or independent variables. Thus, the price of a house is forecasted by building a ML model using the remaining columns.

### 2.2 Data Cleaning

Data preprocessing is a process of cleaning the raw data that is collected. This is an important step in machine learning to improve the accuracy of the model. The first step is to check for null values present in the dataset. If null values are encountered, the reason for these null values must be investigated and treated accordingly.

The two most common methods to treat null values present in the dataset are to delete the corresponding row, or to calculate the mean/median/mode of the corresponding column and replace the missing values with them. This dataset does not have any null values as seen below.



```
# Checking for null values present
df.isnull().sum()
```

```
id               0
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
grade            0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
zipcode          0
lat              0
long             0
sqft_living15    0
sqft_lot15       0
```

It is also important to remove any redundant rows from the data. The two columns – ‘yr_built’ and ‘yr_renovated’ were similar because the majority of the houses were not renovated. More than 95% of the values in yr_renovated was 0. Thus, to give more meaning to this feature, a new column named ‘new_age’ of the house was computed based on the year it was sold, and the former two columns were dropped from the dataset.

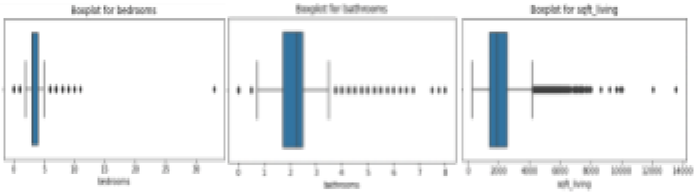
<u>Pseudocode for creating new_age column-</u>

```
if yr_renovated == 0:
        {new_age = yr - yr_built}
else:
        {new_age= yr - yr_renovated}
```

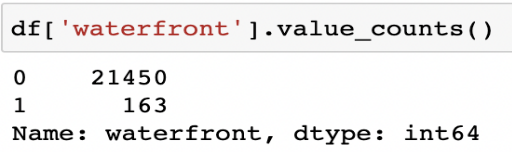
There are a few rows which contain zero bedrooms and zero bathrooms which is obsolete, so there is a necessity to drop these rows. There are a few features/columns like {‘id’, ‘date’} that don’t add much value to this model and thus can be dropped.

### 2.3 Exploratory Data Analysis (EDA)

The primary goal of EDA is to detect **outliers**/errors and to understand various patterns in the data. Outliers are data points that deviate drastically from the other data points and thus reduce the accuracy of the model. It sometimes can be a typo. Outliers can be best visualized using boxplots. A couple of boxplots have been represented below -

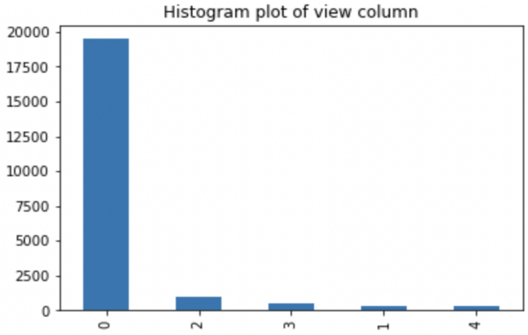


Boxplots are a way of displaying the distribution of data i.e., “minimum”, first quartile (Q1), median, third quartile (Q3), and “maximum”. The box contains the quartile values, and the whiskers are the minimum and maximum values. The data points outside the whiskers are deemed as outliers and need to be treated.



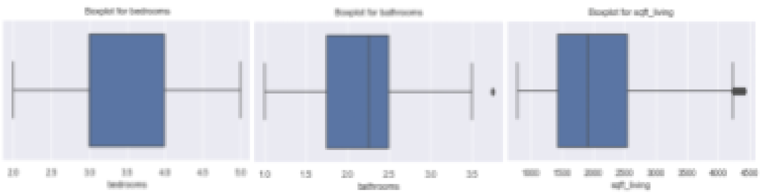
```
df['waterfront'].value_counts()

0     21450
1       163
Name: waterfront, dtype: int64
```



Looking at the distribution of two columns above – {‘wavefront’ & ‘view’} it is seen that the majority of the values are centered around 0 which implies the remaining values are like outliers. Thus, these two columns don’t contribute much to the data model and can be dropped from the dataset.

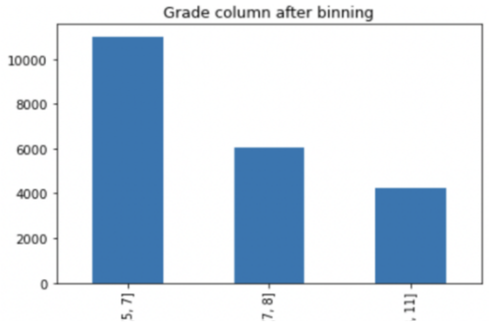
There are many ways to treat outliers. The method followed in this project is clipping. This means, for any selected column containing outliers, the values lying outside the boundary value are clipped to the boundary value. The boundary value in this case is 99th quantile for the higher values and 1st quantile for the lower values. Thus, after removing outliers the box plots look like below-

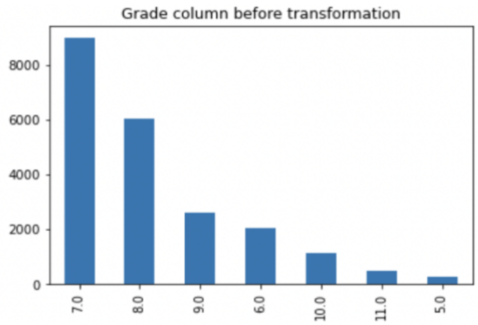


### 2.4 Data Transformation

Scaling in machine learning is a pre-processing technique to standardize the feature columns in the dataset. **Standardizing** involves scaling the distribution of values such that the mean of observed values is 0 and standard deviation is 1. This is done to all the numeric data feature columns. [4]

Machine Learning models can handle only numeric columns hence categorical and ordinal columns must be transformed to numeric columns. For ordinal columns like {‘grade’ & ‘condition’} a transformation concept called **binning** can be used. In this method instead of reading ordinal values as integers, the variable can be transformed to categorical columns by creating bins based on value distribution and then dummy



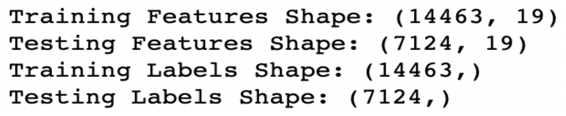


variables are created for this categorical value. [5].

In the above example, the grade column has values between [5-11]. Based on the value count distribution, 3 bins can be created namely – {‘grade_(5, 7]’, ‘grade_(7,8]’, ‘grade_(8,11]’}. These bins are of categorical data type and are converted into numerical type by using dummy variable indicators.

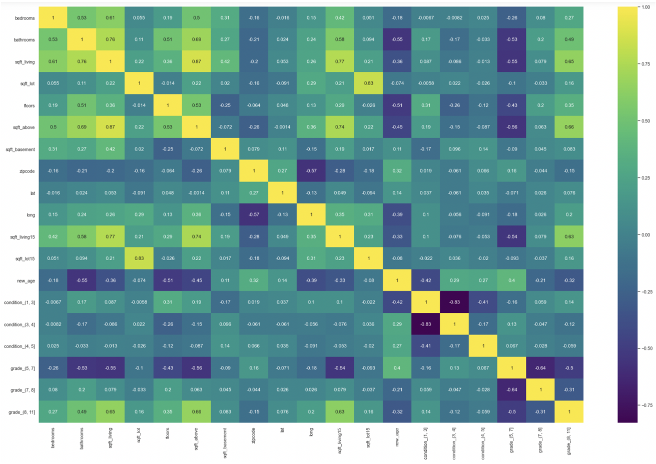
### 2.5 Model Building

To evaluate the predictive model on fresh data, the dataset is split randomly into training and testing sets with a test_size of 33%. The shape of training and testing dataframes are below.



```
Training Features Shape: (14463, 19)
Testing Features Shape: (7124, 19)
Training Labels Shape: (14463,)
Testing Labels Shape: (7124,)
```

The main step before machine learning model building is to plot the correlation matrix. This correlation matrix summarizes the large dataset and helps in understanding which two variables are strongly related to each other. The correlation matrix plotted on the training feature dataset is as follows -



Here, it is observed that there is a high correlation value of 0.87 between sqft_living and sqft_above variables. On investigating it is understood that –

sqft_living = sqft_basement + sqft_above

It is essential to have linearly independent rows, one variable must be dropped from the dataset. In this project ‘sqft_above’ is dropped from the training set.

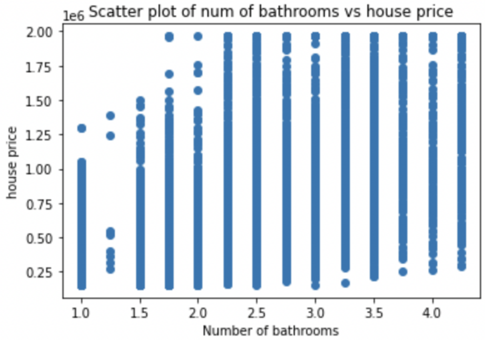
The training set is used for model building and then once the model is fitted, the testing set is used to evaluate this model.

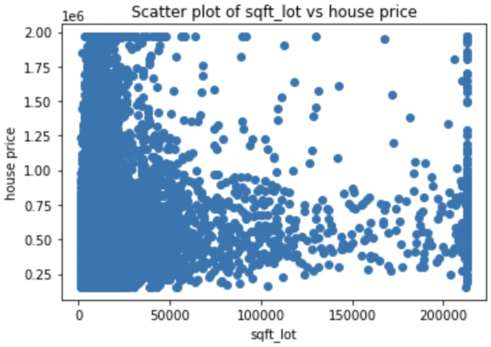
### 2.5.1 Linear Regression Model

The linear regression model is a supervised learning algorithm that predicts the target variable based on a set of input features. To create this model the LinearRegression() function is instantiated and assigned to a variable called regressor. Then the fit() function of this model is run on the training dataset which effectively trains the model. After the training is complete, the model is used to generate some predictions on the test data. This model will use the coefficients it calculated during training to examine the features and predict the house price value. [6]

### 2.5.2 Polynomial Regression Model

In cases where the target variable is not linearly related to the feature variable, the polynomial regression model is utilized. Few features in this dataset that cannot be represented in linear form are -

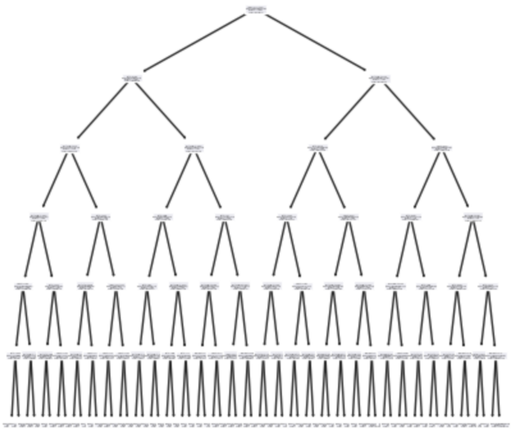




In this project, the degree of the polynomial is set to 2 and then the model is built. The pipeline to build this model is to first perform a polynomial transform on the pre-processed data, and then apply simple linear regression fit. Hence this process is streamlined. [7]

### 2.5.3 Decision Tree

Decision Tree is one of the most widely used approaches for supervised learning. Both Regression and Classification tasks can be solved with the use of this approach. Decision trees are tree-structured classifiers with three types of nodes. The initial node, known as the Root Node, represents the complete sample and may be divided into many nodes. A data set's features are represented by its interior nodes, while its decision-making processes are represented by its branches. The final representation of the result is provided by the Leaf Nodes. This algorithm is quite beneficial for tackling decision-making problems. Given below is a representation of a decision tree.



### 2.5.4 Random Forest Model

Random forest is a supervised learning algorithm that uses an ensemble learning method for classification & regression. Random forest operates by constructing multiple decision trees at the time of training and outputs a value which is the mean prediction of these individual trees. In this project, the n_estimators (number of decision trees) are chosen to be 50 and the model is fitted. This model performs efficiently even during the presence of non-linear features.

### 2.5.5 XGBoost Regressor

Extreme Gradient Boosting (XGBoost) is an open-source toolkit that implements the gradient boosting technique in an efficient and effective manner. Shortly after its inception, XGBoost quickly rose to prominence and was frequently the winning strategy for a variety of tasks in machine learning challenges. In this project, the optimal parameters were estimated using hyper parameter tuning thus resulting in an increase in accuracy.

## 3. EVALUATION

Once a model is built and target values are predicted, it is important to assess the model's performance and compare these predicted values with the actual values. Errors can be computed by the difference between the actual and predicted target values.

Accuracy of the model is defined as the ratio of the number of correctly predicted values to the total number of values.

R-squared is a statistical measure of how close the data are to the fitted regression line. It is also known as the coefficient of determination. It is a goodness-of-fit measure for regression models.

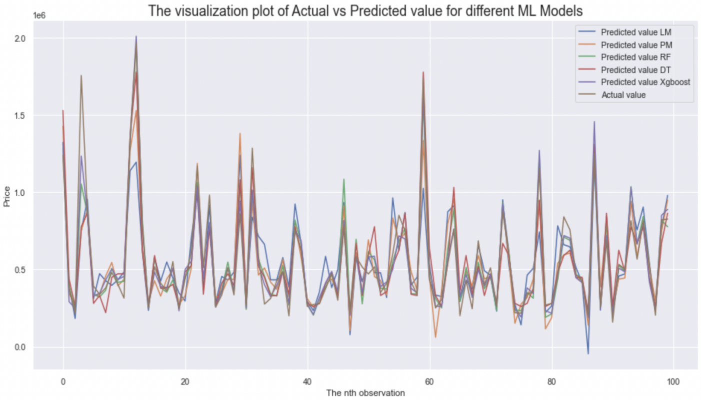
Another metric to evaluate the model is to use root mean square error (RMSE) which computes the average distance between the actual value and predicted value.

The summary of the three regression models used in this project -

| MODEL TYPE | ACCURACY | $R^2$ | RMSE |
|---|---|---|---|
| Linear Regression | 74.84% | 0.67 | 179863.80 |
| Polynomial Regression | 79.55% | 0.78 | 147111.04 |
| Decision Tree | 81.20% | 0.79 | 152732.0 |
| Random Forest | 86.78% | 0.86 | 116356.74 |
| XGBoost | 87.21% | 0.88 | 111066.12 |

It can be observed that the accuracy of the XG Boost Regressor algorithm is the highest among the 5 models. The polynomial Regression model is also performing better than the linear regression model which implies there are many features that aren't linearly related to the target variable 'price'. The Random Forest algorithm performs better than the other 3 models as expected.

The visualization plot of how close the predicted value is to the actual value for all the Machine Learning Models is as below:



## 4. CONCLUSION

In this project, the price of a house was forecasted using Machine Learning models like - Linear Regression, Polynomial Regression, Random Forest, Decision Tree and XGBoost Regressor. In conclusion, the XGBoost Regressor yielded the best results with an accuracy of 87.21%. The computation of new_age of the house from yr_built and yr_renovated added to the efficiency of this model, thereby increasing its accuracy. This simple novel idea can be considered a key feature that influence's people's decision while purchasing a house, rather than resorting to traditional methods of having two columns (that are: yr_built and yr_renovated) to serve the same purpose.

This prediction model once deployed can be useful to the real estate investors as well as the house owners looking to sell their houses, as it predicts the exact quotation of the house based on an accurate assessment of the house features.

## 5. FUTURE WORK

This dataset is limited to the location of Seattle. Thus, to increase the scope of this project, it is essential to obtain data from other regions of the US and evaluate the model prediction. An attempt to increase the model accuracy can be performed by further tweaking the parameters of the model. Furthermore, Deep Learning methods can be looked upon.

## 6. REFERENCES

[1] A. Varma, A. Sarma, S. Doshi and R. Nair, "House Price Prediction Using Machine Learning and Neural Networks," 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT), 2018, pp. 1936-1939, doi: 10.1109/ICICCT.2018.8473231.

[2] Kulkarni, Sushant, Shefin Shajit, Akshay Mohite and Dr. Swati Sinha. “House Price Prediction Using Ensemble Learning.” (2021).

[3]Dataset link – (https://www.kaggle.com/datasets/harlfoxem/housesalesprediction)

[4]https://machinelearningmastery.com/standardscaler-and-minmaxscaler-transforms-in-python/

[5]https://towardsdatascience.com/feature-transformation-for-multiple-linear-regression-in-python-8648ddf070b8

[6]https://practicaldatascience.co.uk/machine-learning/how-to-create-a-linear-regression-model-using-scikit-learn

[7]https://www.analyticsvidhya.com/blog/2021/07/all-you-need-to-know-about-polynomial-regression/