Housing Price Estimator

# Final Project Report

**Student Name**: Vishnu Girish Nujaralla Iyappan

**Mail ID**: [nujarallaiyappan.v@northeastern.edu](mailto:nujarallaiyappan.v@northeastern.edu)

**Signature of Student :**

**Submission Date: 04-21-2023**

Table of Contents

**Problem Setting5**

**Problem Definition5**

**Data Sources5**

**Problem Setting5**

**Data Description5**

Table 1: Description of Variables 5

Figure 1: Dataset Sample6

**Data Mining Tasks6**

Data Understanding6

Figure 2.1 Predictors6

Figure 2.2 Target Variable6

Data Pre-Processing7

Dimension Reduction7

Figure 3 Pie chart7

Figure 4 Nan Cells7

Label Encoding7

Figure 5 Categorical Boolean Values8

One Hot Encoding8

Figure 6 Categorical Values8

Feature Scaling8

Figure 7 Numerical Values8

**Exploratory Data Analysis9**

Categorical Value9

Figure 8 Box Plot of City vs Sale Price 9

Figure 10 Box Plot of Categorical Variables9

Chi Square Test9

Figure 9 Chi Square Test Result10

Numerical Variable11

Figure 11 Pair plot of all the numerical variables 11

Figure 12 Heat Map 12

Figure 13 Density Plot 12

**Data Partitioning13**

**Data Mining Models/Methods**13

Linear Regression13

Decision Tree Regressor14

Random Forest Regressor14

Gradient Boosting Regressor15

Neural Networks15

**Model Selection**16

**Performance Evaluation**16

Linear Regression16

Figure 14 Scatter Plot(Actual VS Predicted Values) 16

Decision Tree Regressor17

Figure 15 Scatter Plot(Actual VS Predicted Values) 17

Gradient Boosting Regressor17

Figure 16 Scatter Plot(Actual VS Predicted Values) 17

Neural Network17

Figure 17 Scatter Plot(Actual VS Predicted Values) 18

Random Forest Regressor19

Figure 18 Scatter Plot(Actual VS Predicted Values) 19

**Future Scope19**

**Project Results20**

Figure 19 Bar Chart for all RMSE Values20

Figure 20 Bar Chart for all MAE Values21

Figure 21 Bar Chart for all R2 Score22

*IE 7275: Data Mining in Engineering*

# Problem Setting:

The problem setting for a Housing Price Estimator is to develop a model that can accurately predict the sale price of a residential property based on a set of features such as square footage, number of bedrooms and bathrooms, etc. The model should be able to help real estate professionals and potential buyers and sellers with a valuable tool for determining the fair market value of a property, and it should be robust and accurate enough to make predictions even when faced with new and unseen data, while also accounting for local market variations in housing prices.

# Problem Definition:

The main objective of the model is to identify which algorithm best suits the dataset to predict the price of the property. In today’s world due to the ever-changing economy, it is difficult to estimate the fair price of a property. The model should be able to handle a wide range of input features, including categorical variables as well as numerical variables. The model should be able to handle missing or incomplete data for some input features to predict the sale price of a house within a certain range of error. Hence, this model helps in predicting the fair price of the property to potential buyers and sellers.

# Data Sources:

The Housing Price Estimator is sourced from [**Kaggle.**](https://github.com/pavankat/flask-ml)

# Data Description:

The dataset includes information about houses, including amenities, the area of the house, and the location of the house. It contains 42703 observations and 19 variables, with both numerical and categorical data.

|  |  |
| --- | --- |
| Stories |  |
| Num\_bedrooms | Number of bedrooms in the property |
| Full\_bathrooms | Number of full bathrooms on the property |
| Half\_bathrooms | Number of half bathrooms in the property |
| Livable\_sqft | Livable Area in sqft of the property |
| Total\_sqft | Total Area of the property |
| Garage\_type | Whether the property has garage attached or not |
| Carport\_sqft | Carpet Area of the property |
| Has\_fireplace | Whether the property has a fireplace or not |
| Has\_pool | Whether the property has a pool or not |
| Has\_central\_heating | Whether the property has central heating or not |
| Has\_central\_cooling | Whether the property has a central cooling or not |
| House\_number | States the property number |
| Street\_name | States the property street name |

|  |  |
| --- | --- |
| Unit\_number | States the property unit number of that apartment |
| City | States the city of the property located |
| Zip\_code | States the zip code of the property located |
| Sale\_price | States the sale price of the property |

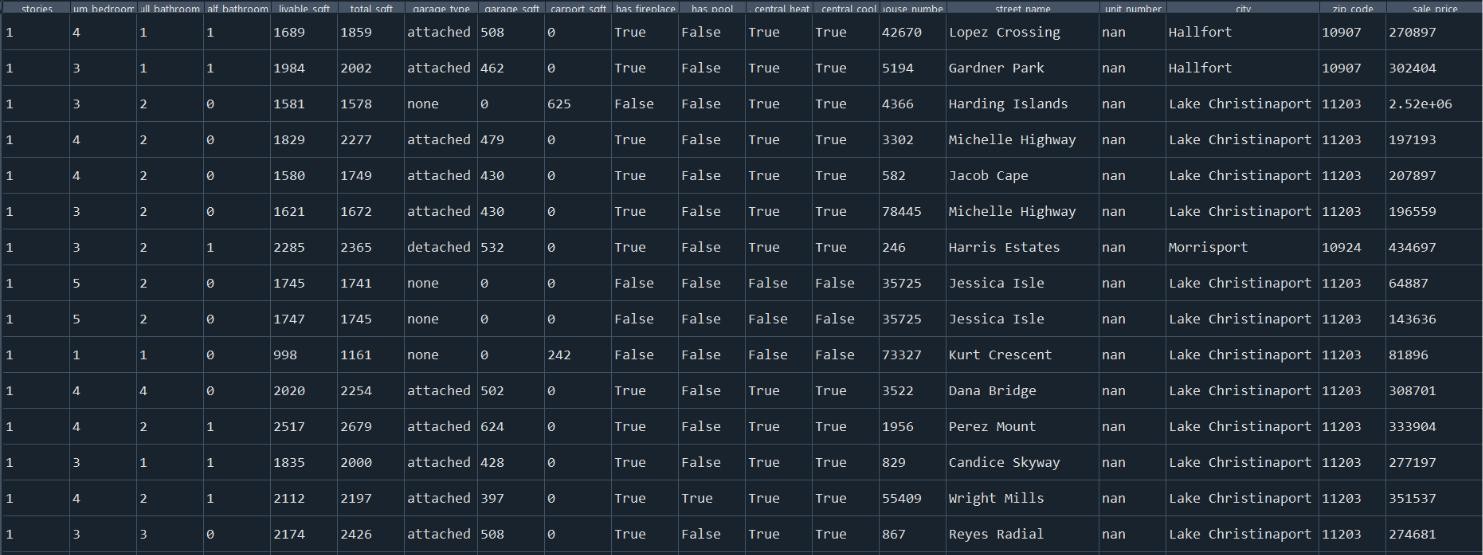


Fig 1. Dataset Sample

# Data Mining Tasks:

1. **Data Understanding:**

The initial dataset contains around 42703 records and 19 columns having both numerical and categorical values. After analysing dataset it was observed that there is 18 predictors and one column “Sale Price” that needs to be predicted. This column was numerical. The categorical columns were in many forms ranging from “True and

False” to “Attached, Detached and None”. There is a city column in the dataset which is categorical in nature and has 47 unique values in that column. Some data pre processing tasks is needed for the dataset to be ready for algorithm fit.

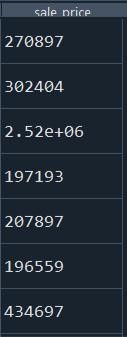
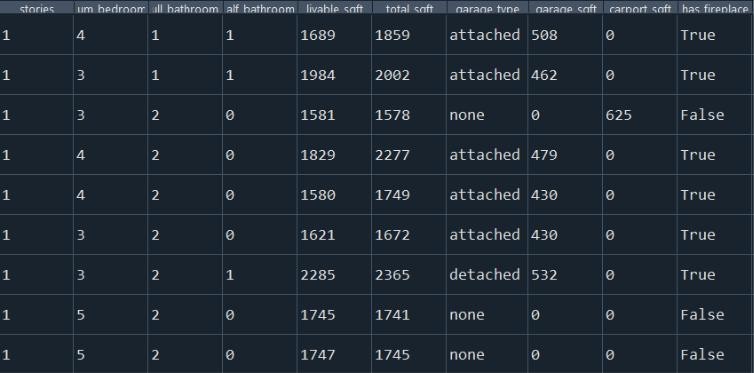


Fig 2.1 Predictors Fig 2.2 Target Variable

# Data Pre-Processing:

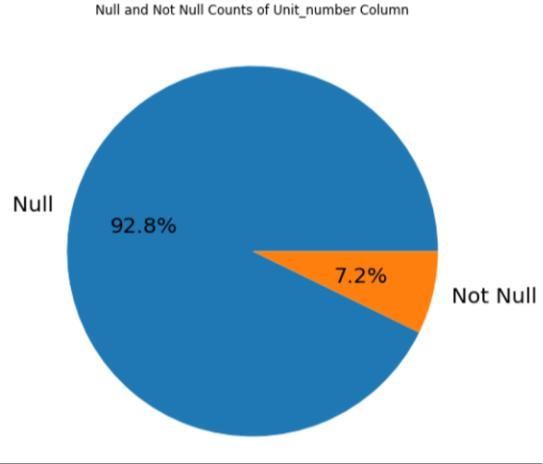
* + Dimension Reduction- In this dataset we have dropped some columns which had many null values. As you can see from the pie chart on the “Unit Number” column, 92.8% of the records are null. So, we must drop that column.

Fig 3. Pie Chart showing null and not null value.

* + Null Values- Unit Number column has around 90% of records as null values, so we need to drop that column. Other than that, no other null values are there in the dataset. So, we do not need to remove any further rows for which has null records.



Fig 4. Nan cells in unit number column

* + Label Encoding- Some columns like Central Cooling, Central Heat, has pool etc. have two categorical values like True, False which directly contributes to the sale price of the house. So, we need to convert those values into their corresponding numerical values.

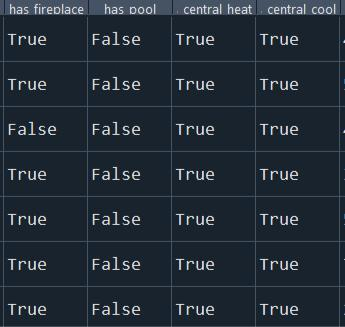


Fig 5. Categorical Boolean values in columns

* + One Hot Encoding- Columns like city has more than two types of categorical values and it is required for the prediction of the sale price of the property. So, we need to convert these categorical values to numerical to fit them to the model, for prediction of the sale price.



Fig 6. Categorical Values in City Column

* + Feature Scaling -The numerical columns in the dataset apart from the encoded columns need to be scaled to a range. The columns involving stories, bedrooms and liveable sqft. and total sqft. have significant scale difference. Since all these features are important for the prediction of sale price, we cannot drop them and there is a chance that the algorithm which will be used for prediction can give more weightage to the columns with higher magnitude. This will impact the performance of the ML model, so we need to scale these features.

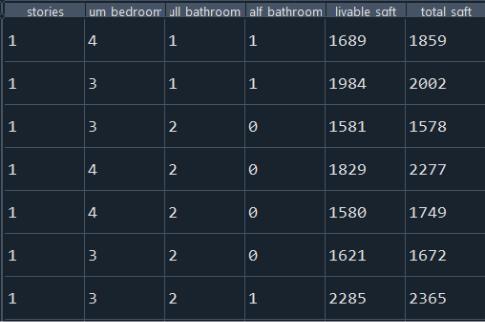


Fig 7. Numerical Values for feature scaling.

# Exploratory Data Analysis:

The data was analysed using exploratory data analysis, both the categorical and numerical data were examined during our analysis.

1. **Categorical Value-**



Fig 8. Box Plot of City vs Sale Price

Based on the provided box plot, it is evident that the city column has a significant impact on predicting the sale price of a house. This is because the mean, median, and maximum prices differ between the two cities shown in the plot. As the median values for each city vary, it is crucial to consider the city column in house price prediction. Therefore, we have decided to retain this column. To improve clarity, we have filtered the data to show only two cities in the plot.

# Chi Square Test Between City and Garage Type-

We conducted a chi square test between two categorical values named Garage\_type between and City. The result of test was a very small value of 0.0015213268. This indicates that the observed data is not significantly different from what would be expected under the null hypothesis of independence between the two variables.

Specifically, a small chi-square test value indicates that there is little or no association between the two categorical variables. In other words, the observed frequencies in the contingency table are very similar to what would be expected if the two variables were independent. This suggests that there is no evidence of a relationship or association between the two variables. Hence both the categorical values will have significance while predicting the sale price of houses.

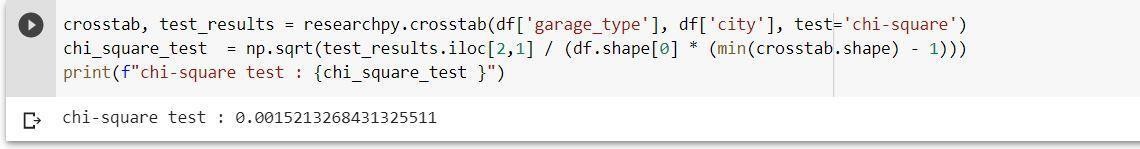


Fig 9. Chi Square Test Result

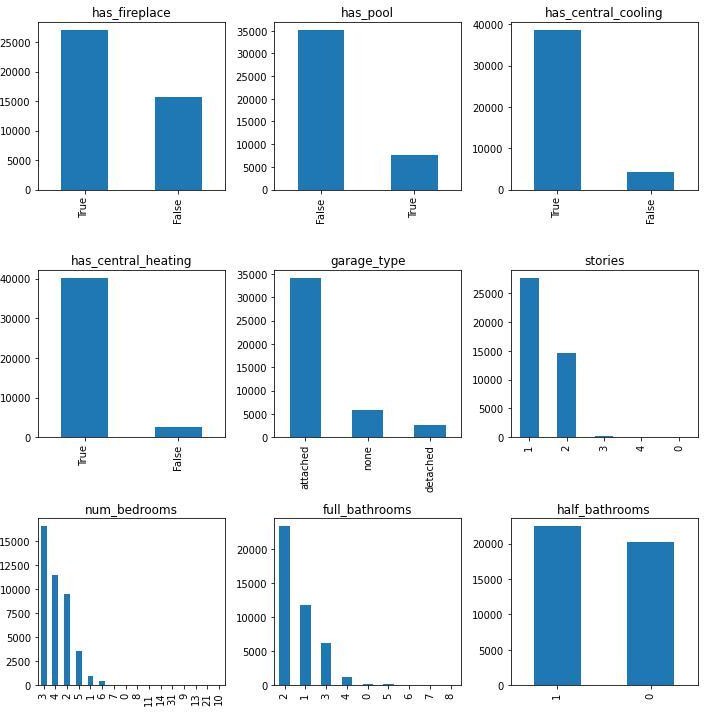


Fig. 10 Box plot of Categorical Variables

From the above bar chart these are the types of categories and its distribution for each variable is observed .

# Numerical Variable-



Fig 11. Pair plot of all the numerical values with target variable to see correlation between them using univariate analysis

Based on the pair plot, it appears that there is a correlation between liveable sqft and total sqft. This suggests that it may be redundant to include both variables in our analysis, and we could potentially eliminate one of them.

If we see the plot of zip code and target variable, we see no correlation between them and zip code is also not related to any other predictor in the dataset. This means that it is not worth using the zip code column as one of our predictors.

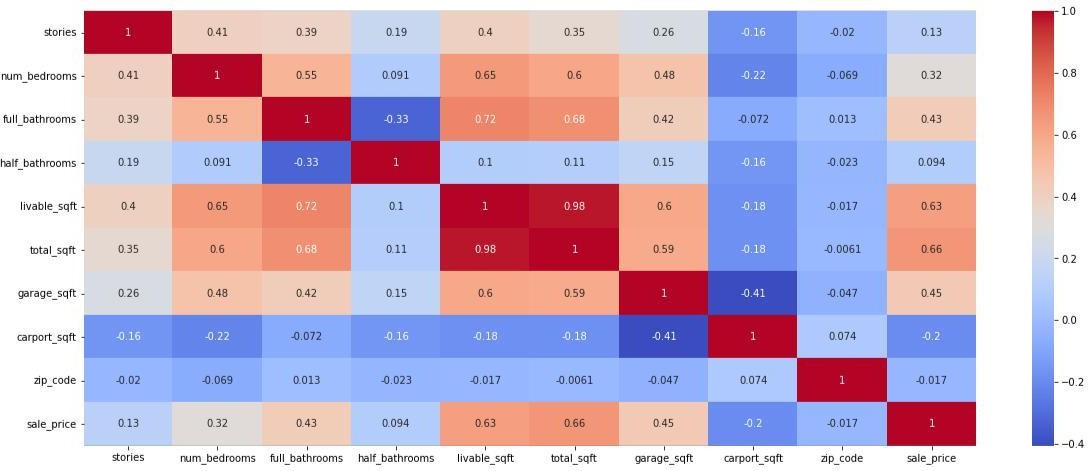


Fig. 12. Heat Map of all the numerical variables

After generating a heatmap to better understand the correlation values, we can support the use of the pair plot. Based on this analysis, we have found that there is a strong correlation between the variables of liveable sqft and total sqft.

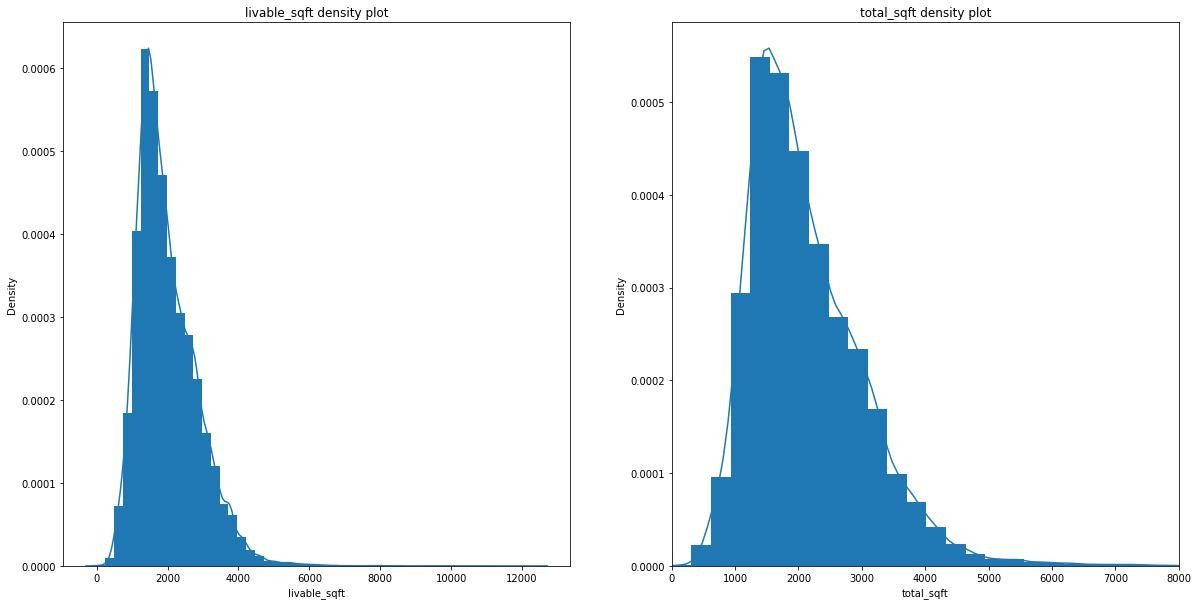


Fig. 13 Density Plot livalblre sqft and Total sqft

From the above, we can see that all the variables are right skewed. Hence, we handled the data by normalizing it and transforming the data to make it more amendable for statistical analysis.

Normalization involves rescaling the data to have values between 0 and 1. This is done by subtracting the minimum value from each data point and dividing by the range (i.e., the difference between the maximum and minimum values). This can be helpful when the data range is large or when you want to compare data across different scales.

# Data Partitioning:

The predictor variables are labelled as “x” and the target variable are labelled as “y”. The predictor variable has 60 columns and the target variable is “sale price”. The splitting of dataset was achieved by dividing the data into train, test and validation data. The ratio of all the three splits is 80:10:10. This implies that 80% of data was used for training and 10 % was used for both testing and validation. The training data has 80% data to avoid overfitting as our data became wide after pre-processing. The x training set has 42,703 records and testing and validation has 4271 records each. The y train and y test on the other hand has similar records as the x train and x test.

# Data Mining Models/Methods

Four regression models were constructed using the training data through data mining techniques. These models were developed to predict the outcome of sale price based on many predictors. The models were selected based on their ability to minimize the error between the predicted and actual values in the training data. The regression models can be used to make predictions on new, unseen data, and their accuracy can be evaluated using appropriate metrics.

1. **Linear Regression-** In this algorithm the idea was to make use of training data to make prediction regarding the sale price of a property from the test data. It works by estimating the coefficients of the regression equation based on the training data, and then using those coefficients to predict the values of the dependent variable for new, unseen data.

Advantages:

* + This model can be computed quickly and efficiently, even when dealing with large amounts of training data.
  + It is robust to noise and outliers, as long as it does not significantly affect the overall trend of the data.
  + Linear regression can easily handle multiple variables and can account for the interactions between them. For example, the relationship between the size of the house and the price may depend on the location, and linear regression can capture this complex relationship.

Disadvantages:

* + Linear regression assumes that the relationship between the predictors and the predicting variable is linear. If the relationship is not linear, the results obtained from linear regression can be misleading or incorrect.
  + It assumes that the predictor variables are independent of each other. If the predictor variables are correlated, the regression results can be biased and misleading.

Implementation:

* + The algorithm was fitted on the training data containing around 34 thousand records with 60 predictors and the accuracy of the model was about 62.7%.

1. **Decision Tree Regressor-** A decision tree regressor can be used as a part of a house price estimator to predict the value of a house based on its features. It works by recursively partitioning the data into subsets based on the values of input features, and then making predictions based on the values of the output variable in each subset. As a result a tree like structure is formed where each internal node is corresponds to decision based on input variables and each leaf node leads to a prediction.

Advantages:

* + It can handle nonlinear relationships, unlike linear regression, decision trees can handle nonlinear relationships between the independent variables and the dependent variable. This is important in the case of house price estimation, where factors such as location, size, and amenities can have complex, nonlinear relationships with the final sale price.
  + Once a decision tree has been trained, predicting the class label of a new instance is fast and efficient. The time complexity of prediction is O(log n), where n is the number of predictors in the dataset.

Disadvantages:

* + The algorithms are not good at handling continuous variables especially if there many possible values.
  + It is sensitive to small changes in data which can lead to incorrect predictions.

Implementation:

The training dataset was fitted with this algorithm resulting in an accuracy of 60.2%. Fine tuning the hyperparameters of the algorithm like maximum depth of the tree can result in an increase in accuracy.

1. **Random Forest Regressor**- Random Forest Regressor is a popular algorithm used in housing price estimation tasks. It is implemented because it can manage many features, efficiently handle missing data and outliers, and

capture non-linear correlations between the features and the target variable. It is an ensemble learning method that combines multiple decision trees to improve the accuracy of the model. The algorithm works by creating a set of decision trees from random subsets of the training data and randomly selected subsets of the features.

Advantages:

* + This typically provide high accuracy for both classification and regression tasks. They can handle large datasets with many features and can deal with missing data and outliers.
  + Random forests are less likely to overfit the training data than a single decision tree, making them more robust to noise and outliers in the data.

Disadvantages:

* + It requires a large amount of memory to store all the decision tree that it is making for the forest. This is a problem if the dataset is very large or if there are memory constraints.
  + It can struggle to handle datasets with highly correlated features. In these cases, the model may give too much weight to certain features and ignore others, leading to biased predictions.

Implementation:

After fitting the training data with this algorithm an accuracy of 79% was achieved. We used 300 trees in this forest prediction. Increasing the number trees or estimators can lead to better predictions to an extent, as this can also lead to overfitting.

1. **Gradient Boosting Regressor-** Gradient boosting regressor is an algorithm that works on boosting technique. This uses a collection of decision trees for prediction which are made sequentially as it rectifies the errors made by trees that were built previously. As a result, it can provide accurate predictions for wide range of regression tasks.

Advantages:

* + Gradient boosting produces highly accurate results compared to other machine learning algorithms
  + It has a low bias and variance, which means it is less likely to overfit or underfit the training data.
  + Gradient boosting can handle large datasets with millions of samples and features.

Disadvantages:

* + Gradient boosting has many hyperparameters that need to be tuned to achieve optimal performance, which can be time-consuming and challenging.
  + It requires a lot of computation power and can be slow to train, especially when using large datasets or complex models.

Implementation:

An accuracy of around 80% was achieved after implementation of this model to training data. We fined tune the hyperparameters which included 1000 estimators, 9 minimum sample leafs, and max depth of trees was set to 6. Increasing or decreasing these parameters may result in higher accuracy.

# Model Selection

After careful consideration of all the models that were deployed and reading about their advantages, disadvantages and keeping in mind the accuracy of each model on our dataset. We are going to choose Gradient Boosting Regressor as our predictive model for this project.

**Performance Evaluation**

1. **Linear Regressor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| models | R2\_Score | rmse\_arr | mse\_arr | sse\_arr | mae\_arr |
| linear regressor | 0.572607 | 208600.023125 | 4.351397e+10 | 1.858482e+14 | 97044.898572 |

Chart, scatter chart

Description automatically generated

Figure 4 Actual Values VS Predicted Values

The linear regressor model exhibits high average errors in predictions, as indicated by its RMSE and MAE values. The RMSE further represents the model's average weighted performance. Such high error values imply that exploring alternative models could potentially yield improved results. Moreover, a low R2 score is often used to assess the regression model's ability to explain the observed data.

1. **Decision Tree Regressor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| models | R2\_Score | rmse\_arr | mse\_arr | sse\_arr | mae\_arr |
| decision\_tree | 0.678605 | 180892.511887 | 3.272210e+10 | 1.397561e+14 | 81256.756102 |

**Chart, scatter chart

Description automatically generated**

Figure 5 Actual Values VS Predicted Values

Compared to the linear regression model, the decision tree regressor exhibits lower RMSE and MAE values, and a higher R2 score, indicating superior performance. However, despite the improved results, the RMSE is not the lowest, suggesting that exploring additional models may lead to further enhancements.

1. **Gradient Boosting Regressor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| models | R2\_Score | rmse\_arr | mse\_arr | sse\_arr | mae\_arr |
| gradient\_regression | 0.720227 | 168773.250085 | 2.848441e+10 | 1.216569e+14 | 63042.321596 |

Chart, scatter chart

Description automatically generated

Figure 6 Actual Values VS Predicted Values

In this scenario, we utilized Gradient Boosting Regressor to make predictions for the sale price of a house, which is the target variable. The root mean squared error (RMSE) and mean absolute error (MAE) obtained from this model were lower than those obtained from the Decision Tree Regressor model. Additionally, the R2 score for the Gradient Boosting Regressor model was superior to that of the Decision Tree Regressor model.

1. **Neural Network**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| models | R2\_Score | rmse\_arr | mse\_arr | sse\_arr | mae\_arr |
| Neural Network | 0.6111850 | 1.301098e+10 | 2.602196e+10 | 1.698482e+14 | 65726.370780 |

Chart, scatter chart

Description automatically generated

Figure 7 Actual Values VS Predicted Values

In this scenario, a neural network was employed to predict the sale price of a house. However, when compared to other models, the root mean square error (RMSE) and mean absolute error (MAE) obtained from this neural network were the highest. Additionally, the R2 score, a measure of how well the model fits the data, did not perform better than the gradient boosting regressor.

1. **Random Forest Regressor**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| models | R2\_Score | rmse\_arr | mse\_arr | sse\_arr | mae\_arr |
| random\_regressor | 0.746633 | 160611.342000 | 2.579600e+10 | 1.101747e+14 | 63321.996274 |

**Chart, scatter chart

Description automatically generated**

Figure 8 Actual Values VS Predicted Values

The evaluation metrics show that the random forest regressor model outperforms the Gradient Boosting Regressor model in terms of lower RMSE and MAE values, and it also has the highest R2 score among all other models, indicating the best fit. Therefore, it can be concluded that the random forest regressor model is the most appropriate predictor to use.

**Future Scope**

* Integration: Integration with other real estate tools: House price estimators can be integrated with other real estate tools to create a more comprehensive picture of the real estate market.
* Improved accuracy: With the availability of more data and the development of more sophisticated algorithms, house price estimators will become even more accurate in the future.
* Enhanced feature engineering: Feature engineering is a crucial aspect of ML-based house price estimation. In the future, there will likely be more advancements in feature engineering techniques, such as the inclusion of climate data, demographic data, and social media data.

**Project Results**

After evaluating multiple prediction models for our dataset, we have concluded that the Random Forest Regressor is the optimal choice. One of the key reasons for this decision is that the Random Forest Regressor achieved high R2 values, indicating that the model fits the data well. Additionally, this model yielded low error values, specifically a low root mean square error (RMSE), which is a measure of the average difference between predicted and actual values.

Furthermore, we verified our results by calculating error values and R2 score using proper methodology. This analysis confirmed that the Random Forest Regressor is indeed the best model for our project. This is a significant finding because it suggests that the Random Forest Regressor is highly accurate in predicting the sale price of a house based on the given features in our dataset. As a result, this model may be useful for decision-making in the real estate industry, such as for determining the fair market value of a property.

Graphical user interface, text, application

Description automatically generated

Chart, bar chart

Description automatically generated

Figure 9 Bar Chart For all RMSE Values

Chart, bar chart

Description automatically generated

Figure 20 MAE For all Models

Chart, bar chart

Description automatically generated

Figure 21 R2 Score Values For all Models