SAIT

Final Capstone Project - Applied Machine Learning

Professional Data Science Certificate

India House Price Prediction

Written by:
Shubh Desai

Submitted to:
Professor Dr. Jaspreet Gill
Prof. Kanika Sehgal

List of Content

1. Introduction

The 14,620 records in the house price dataset utilized for the current research provide an extensive overview of Indian real estate data. Each entry represents a house and includes a variety of features that describe the characteristics and attributes of the house. These features range from basic information such as the number of bedrooms and bathrooms to more specific details like the presence of a waterfront, the condition and grade of the house, and its proximity to important amenities.

2. Business Understanding

Business Goal

Objective is to build a predictive model for house prices to assist stakeholders in making data-driven decisions.

Model Objective

Create a machine learning model that can accurately predict house prices based on input features such as the number of bedrooms, bathrooms, living area, lot area, and other attributes.

3. Problem Statement

Why is it worth solving?

Predicting house prices accurately is crucial for various stakeholders in the real estate market. Accurate predictions can lead to better investment decisions, good pricing, and improved customer satisfaction.

Importance of the Problem

The problem is significant because it directly impacts financial decisions in the real estate market, influencing buyers, sellers, and investors. A reliable prediction model can enhance market efficiency and reduce the risk associated with property investments.

Data Sourcing

The dataset was sourced from Kaggle and includes various features such as the number of bedrooms, bathrooms, living area, lot area, and other characteristics.

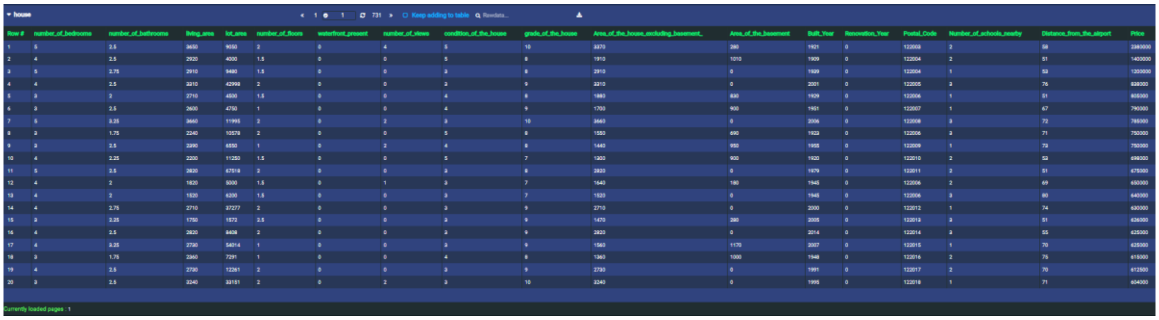
4. Data Collection

Data Understanding

The dataset consists of numerous features that describe the properties.

Key features include:

1. **id**: A unique identifier for each house.
2. **Date**: The date on which the house information was recorded.
3. **number of bedrooms**: The total number of bedrooms in the house.
4. **number of bathrooms**: The total number of bathrooms in the house, including half-bathrooms.
5. **living area**: The total living area (in square feet) of the house.
6. **lot area**: The total area (in square feet) of the lot on which the house is built.
7. **number of floors**: The total number of floors in the house.
8. **waterfront present**: A binary indicator (0 or 1) denoting whether the house has a waterfront view.
9. **number of views**: The number of times the house has been viewed.
10. **condition of the house**: A numerical rating (1 to 5) of the overall condition of the house.
11. **grade of the house**: A numerical rating (1 to 13) of the construction and design quality of the house.
12. **Area of the house (excluding basement)**: The total area (in square feet) of the house, excluding the basement.
13. **Area of the basement**: The total area (in square feet) of the basement.
14. **Built Year**: The year in which the house was originally built.
15. **Renovation Year**: The year in which the house was last renovated (0 if never renovated).
16. **Postal Code**: The postal code where the house is located.
17. **Lattitude**: The latitude coordinate of the house's location.
18. **Longitude**: The longitude coordinate of the house's location.
19. **living_area_renov**: The living area (in square feet) of the house after renovation.
20. **lot_area_renov**: The lot area (in square feet) after renovation.
21. **Number of schools nearby**: The number of schools located within a certain distance from the house.
22. **Distance from the airport**: The distance (in kilometers) from the house to the nearest airport.
23. **Price**: The price of the house in Indian Rupees.



Row #	number_of_bedrooms	number_of_bathrooms	living_area	lot_area	number_of_floors	waterfront_present	number_of_views	condition_of_the_house	grade_of_the_house	Area_of_the_house_excluding_basement_	Area_of_the_basement	Built_Year	Renovation_Year	Postal_Code	Number_of_schools_nearby	Distance_from_the_airport	Price
1	5	2.5	3650	9050	2	0	4	5	10	3370	280	1921	0	122003	2	58	2380000
2	4	2.5	2920	4000	1.5	0	0	5	8	1910	1010	1909	0	122004	2	51	1400000
3	5	2.75	2910	9480	1.5	0	0	3	8	2910	0	1939	0	122004	1	53	1200000
4	4	2.5	3310	42998	2	0	0	3	9	3310	0	2001	0	122005	3	76	838000
5	3	2	2710	4500	1.5	0	0	4	8	1880	830	1929	0	122006	1	51	805000
6	3	2.5	2600	4750	1	0	0	4	9	1700	900	1951	0	122007	1	67	790000
7	5	3.25	3660	11995	2	0	2	3	10	3660	0	2006	0	122008	3	72	785000
8	3	1.75	2240	10578	2	0	0	5	8	1550	690	1923	0	122006	3	71	750000
9	3	2.5	2390	6550	1	0	2	4	8	1440	950	1955	0	122009	1	73	750000
10	4	2.25	2200	11250	1.5	0	0	5	7	1300	900	1920	0	122010	2	53	698000
11	5	2.5	2820	67518	2	0	0	3	8	2820	0	1979	0	122011	2	51	675000
12	4	2	1820	5000	1.5	0	1	3	7	1640	180	1945	0	122006	2	69	650000
13	4	2	1520	6200	1.5	0	0	3	7	1520	0	1945	0	122006	3	80	640000
14	4	2.75	2710	37277	2	0	0	3	9	2710	0	2000	0	122012	1	74	630000
15	3	2.25	1750	1572	2.5	0	0	3	9	1470	280	2005	0	122013	3	51	626000
16	4	2.5	2820	8408	2	0	0	3	9	2820	0	2014	0	122014	3	55	625000
17	4	3.25	2730	54014	1	0	0	3	9	1560	1170	2007	0	122015	1	70	625000
18	3	1.75	2360	7291	1	0	0	4	8	1360	1000	1948	0	122016	2	75	615000
19	4	2.5	2730	12261	2	0	0	3	9	2730	0	1991	0	122017	2	70	612500
20	3	2.5	3240	33151	2	0	2	3	10	3240	0	1995	0	122018	1	71	604000

Currently loaded pages : 1

5. Exploratory Data Analysis (EDA)

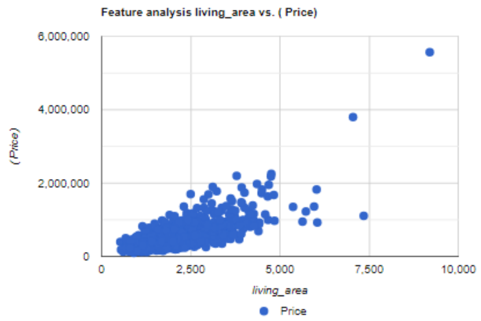
Initial data exploration reveals the following:

- The dataset contains 14620 rows and 23 columns.
- The target variable Outcome is numeric, with indicating house prices in INR.

Univariate Analysis

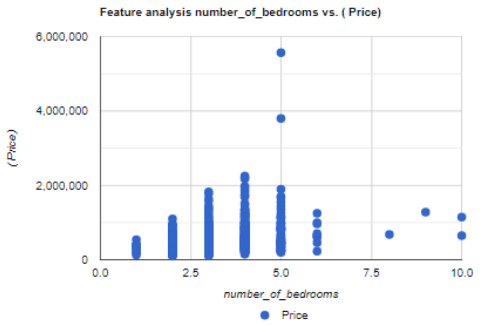
Univariate analysis was performed to understand the distribution of each feature and its potential influence on house prices.

Relationship between Living Area and Price (Moderate to strong positive correlation)



There is a visible positive correlation between living area and price; as the living area increases, the price tends to increase as well.

Number of Bedrooms vs. Price (Positive correlation)



There is a general trend where properties with more bedrooms tend to have higher prices.

Properties with 10 bedrooms are rare and show a wide price range.

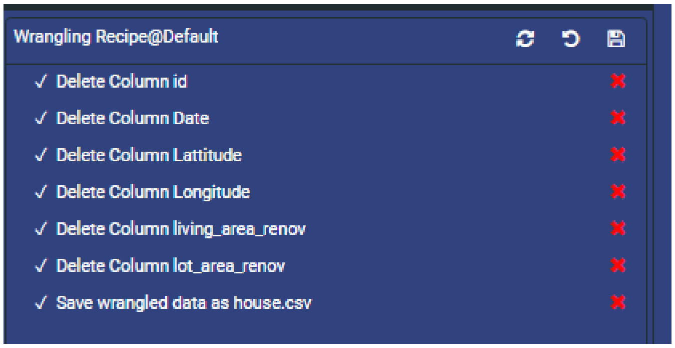
6. Data Preparation

Handling Missing Values

The dataset contains no missing values, but we did some wrangling and normalization to improve accuracy and reduce error rates.

Data Wrangling

Data wrangling entailed cleaning up the data, removing outliers, and altering variables to improve model performance.



Wrangling Recipe@Default

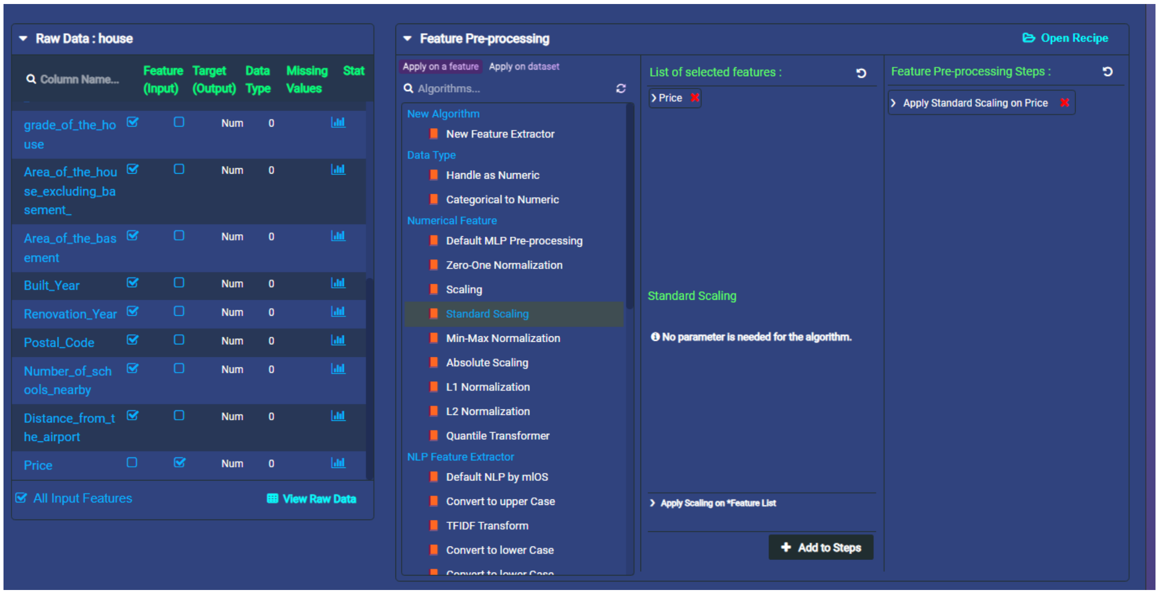
- ✓ Delete Column id
- ✓ Delete Column Date
- ✓ Delete Column Lattitude
- ✓ Delete Column Longitude
- ✓ Delete Column living_area_renov
- ✓ Delete Column lot_area_renov
- ✓ Save wrangled data as house.csv

Deleted columns

1. **Id:** The Id column is not used when building the model.
2. **Date:** The Date column is not used in the model building.
3. **Latitude:** We already have the postal code column, therefore we don't need the latitude.
4. **Longitude:** We already have the postal code column, therefore we don't need the latitude.
5. **Living_area_renovation:** There is no correct data; more than 90% of the data is 0 value.
6. **Lot_area_renovation:** There is no correct data; more than 90% of the data is 0 value.

Normalization

Features were standardized to maintain a consistent scale, which is critical for some machine learning techniques.



Raw Data : house

Column Name...	Feature (Input)	Target (Output)	Data Type	Missing Values	Stat
grade_of_the_house	☑	□	Num	0	📊
Area_of_the_house_excluding_basement_	☑	□	Num	0	📊
Area_of_the_basement	☑	□	Num	0	📊
Built_Year	☑	□	Num	0	📊
Renovation_Year	☑	□	Num	0	📊
Postal_Code	☑	□	Num	0	📊
Number_of_schools_nearby	☑	□	Num	0	📊
Distance_from_the_airport	☑	□	Num	0	📊
Price	□	☑	Num	0	📊

☑ All Input Features

Feature Pre-processing

List of selected features :

› Price ✖

Standard Scaling

No parameter is needed for the algorithm.

› Apply Scaling on *Feature List

Feature Pre-processing Steps :

› Apply Standard Scaling on Price ✖

Standard Scaling

Standard Scaling is a method used to adjust the values of data so that they have an average value of 0 and a spread (how much they differ from the average) of 1.

The formula looks like this:

$$\text{scaled value} = \frac{\text{original value} - \text{average of values}}{\text{standard deviation}}$$

House Prices:

- House 1: 1,400,000 INR
- House 2: 838,000 INR

Calculation:

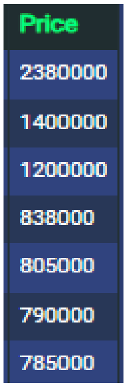
1. **Calculate the Mean (μ\muμ):**
   - From the dataset, let's assume the mean price is 791,000 INR.
2. **Calculate the Standard Deviation (σ\sigmaσ):**
   - From the dataset, let's assume the standard deviation is 316,000 INR.
3. **Apply the Standard Scaling Formula:**
   - For House 1 (1,400,000 INR): z=1,400,000−791,000316,000≈1.93z = \frac{1,400,000 - 791,000}{316,000} \approx 1.93z=316,0001,400,000−791,000≈1.93
   - For House 2 (838,000 INR): z=838,000−791,000316,000≈0.15z = \frac{838,000 - 791,000}{316,000} \approx 0.15z=316,000838,000−791,000≈0.15

Standard Scaled Prices:

- House 1: Approximately 1.93
- House 2: Approximately 0.15

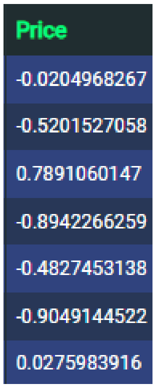
Because the cost of a house exceeds 5 or 6 digits, I used ordinary scaling normalization on price (the goal variable). To achieve a reduced error rate, use conventional scaling, which compresses between 0 and 1 or -1.

Before Normalization



Price
2380000
1400000
1200000
838000
805000
790000
785000

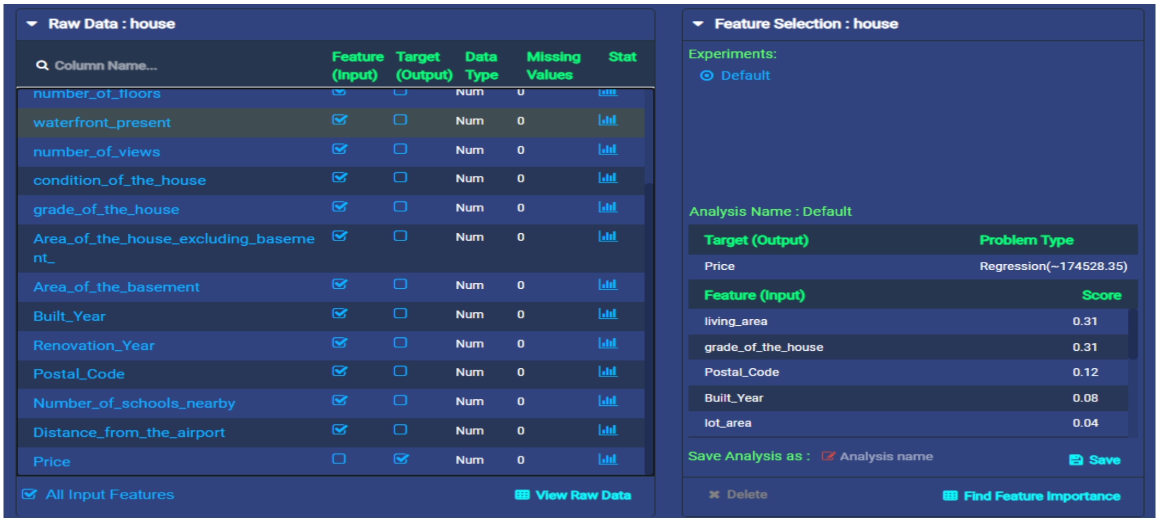
After Normalization



Price
-0.0204968267
-0.5201527058
0.7891060147
-0.8942266259
-0.4827453138
-0.9049144522
0.0275983916

Inputs and Target Features

All the columns excluding the Price are in Feature Input and the Price column is in Target Variable



Raw Data : house

Column Name...	Feature (Input)	Target (Output)	Data Type	Missing Values	Stat
number_of_floors	☑	□	Num	0	📊
waterfront_present	☑	□	Num	0	📊
number_of_views	☑	□	Num	0	📊
condition_of_the_house	☑	□	Num	0	📊
grade_of_the_house	☑	□	Num	0	📊
Area_of_the_house_excluding_basement_	☑	□	Num	0	📊
Area_of_the_basement	☑	□	Num	0	📊
Built_Year	☑	□	Num	0	📊
Renovation_Year	☑	□	Num	0	📊
Postal_Code	☑	□	Num	0	📊
Number_of_schools_nearby	☑	□	Num	0	📊
Distance_from_the_airport	☑	□	Num	0	📊
Price	□	☑	Num	0	📊

☑ All Input Features

Feature Selection : house

Experiments:
⦿ Default

Analysis Name : Default

Target (Output)	Problem Type
Price	Regression(~174528.35)

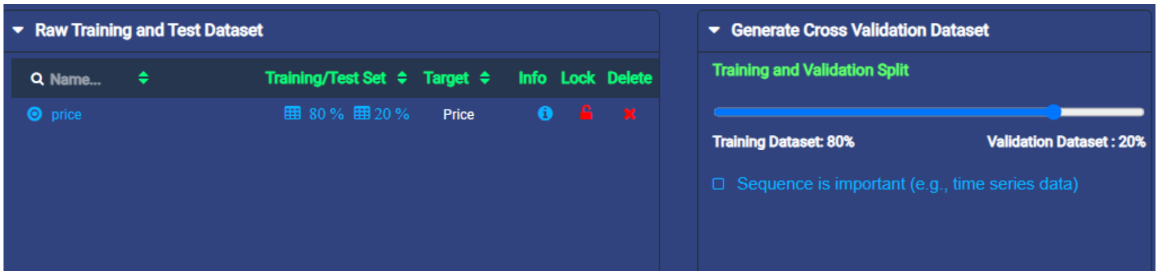
Feature (Input)	Score
living_area	0.31
grade_of_the_house	0.31
Postal_Code	0.12
Built_Year	0.08
lot_area	0.04

Save Analysis as : Analysis name

Data Splitting

The dataset was split into training and testing sets:

- 80% of the data was used for training.
- 20% of the data was reserved for testing.



Raw Training and Test Dataset

Name...	Training/Test Set	Target	Info	Lock	Delete
⦿ price	80 % 20 %	Price	ℹ	🔓	✖

Generate Cross Validation Dataset

Training and Validation Split

Training Dataset: 80%   Validation Dataset : 20%

□ Sequence is important (e.g., time series data)

7. Methodology

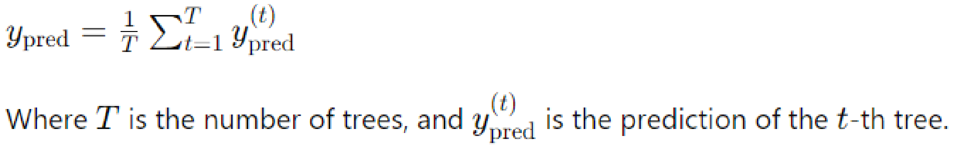
Model Selection

For this regression task, we used the following algorithms:

1. Random Forest

Definition: Random Forest Regression is an ensemble learning method that builds multiple decision trees and merges them together to get a more accurate and stable prediction.

Formula:



$$y_{\text{pred}} = \frac{1}{T} \sum_{t=1}^{T} y_{\text{pred}}^{(t)}$$

Where T is the number of trees, and $y_{\text{pred}}^{(t)}$ is the prediction of the t-th tree.

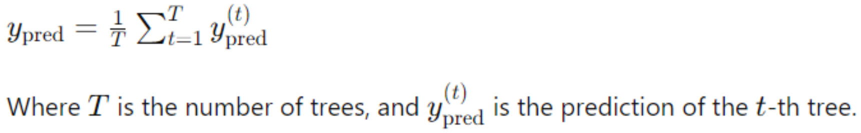
Uses:

- More accurate than individual decision trees.
- Reduces the risk of overfitting.
- Widely used in various domains like finance, healthcare, and marketing.

2. Extra Trees

Definition: Extra Trees Regression is similar to Random Forest, but it adds more randomness to the model. It builds multiple trees and splits the data at random points rather than the most optimal points.

Formula:



$$y_{\text{pred}} = \frac{1}{T} \sum_{t=1}^{T} y_{\text{pred}}^{(t)}$$

Where T is the number of trees, and $y_{\text{pred}}^{(t)}$ is the prediction of the t-th tree.

Uses:

- Similar benefits to Random Forest, but often faster to train.
- Good for large datasets with many features.

3. Decision Tree Regression

Definition: Decision Tree Regression is a type of model that predicts the target value by learning simple decision rules inferred from the data features. It splits the data into smaller and smaller subsets and then makes predictions based on the mean value of the target variable in these subsets.

Formula:

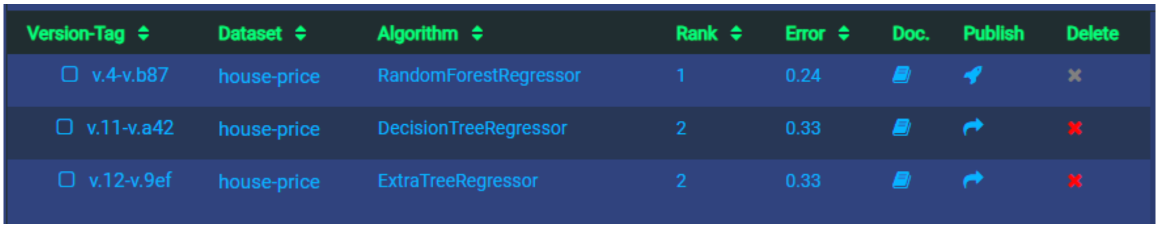
$$y_{\text{pred}} = \frac{1}{N} \sum_{i=1}^{N} y_i$$

Where N is the number of data points in the leaf, and y_i are the target values.

Uses:

- Simple and easy to interpret.
- Used in various applications where the relationships between features and the target variable are complex and non-linear.

Model Evaluation



Version-Tag ⬍	Dataset ⬍	Algorithm ⬍	Rank ⬍	Error ⬍	Doc.	Publish	Delete
□ v.4-v.b87	house-price	RandomForestRegressor	1	0.24	📘	🚀	✖
□ v.11-v.a42	house-price	DecisionTreeRegressor	2	0.33	📘	↪	✖
□ v.12-v.9ef	house-price	ExtraTreeRegressor	2	0.33	📘	↪	✖

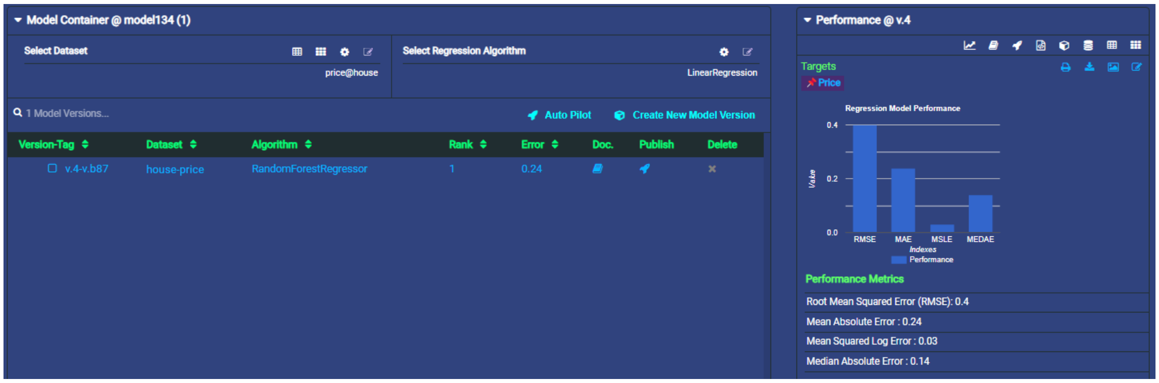
The performance of the models was evaluated using the following metrics:

Mean Squared Log Error (MSLE)
Mean Absolute Error (MAE)
Mean Squared Error (MSE)
Median Absolute Error

Regression	**Mean Squared Log Error**	**MAE**	**MSE**	**Median Absolute Error**
Random Forest	0.03	0.24	0.4	0.14
Extra Tree	0.06	0.33	0.55	0.19
Decision Tree	0.06	0.33	0.55	0.19

8. Best Model

The Random Forest Regressor performed the best with the highest accuracy and less accuracy.



Model Container @ model134 (1)

Select Dataset: price@house

Select Regression Algorithm: LinearRegression

1 Model Versions...

Auto Pilot | Create New Model Version

Version-Tag	Dataset	Algorithm	Rank	Error	Doc.	Publish	Delete
□ v.4-v.b87	house-price	RandomForestRegressor	1	0.24			×

Performance @ v.4

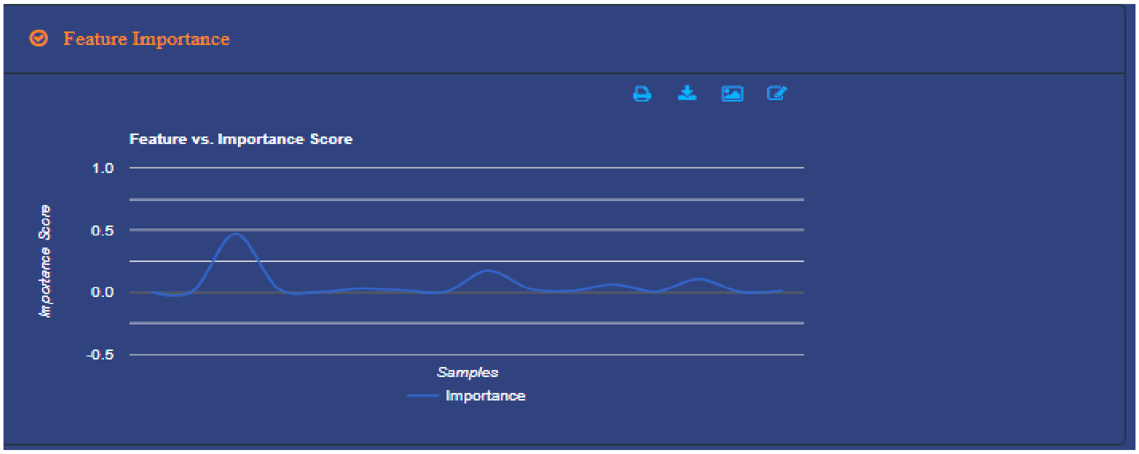
Targets: Price

Performance Metrics

Root Mean Squared Error (RMSE): 0.4
Mean Absolute Error : 0.24
Mean Squared Log Error : 0.03
Median Absolute Error : 0.14

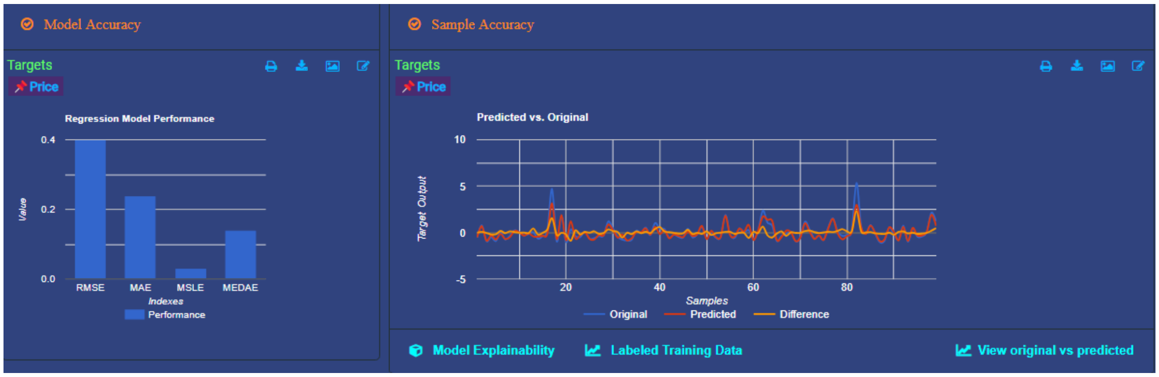
Feature Importance:

Feature Importance have higher significance ratings, indicating a stronger influence on price prediction. Understanding feature relevance improves in feature selection and model performance.



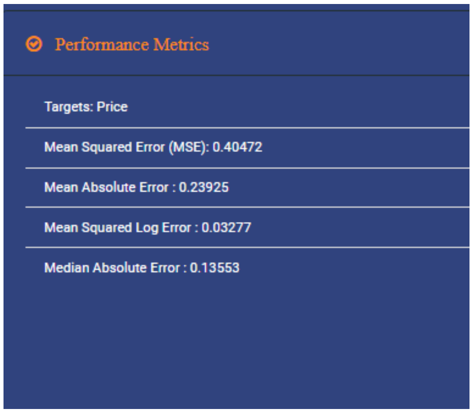
Model Accuracy:

Metrics like as RMSE, MAE, and MSLE provide a quantitative assessment of the model's accuracy.The predicted vs. original plot further confirms that the model's predictions are closely aligned with the actual values.



Performance Metrics:

Metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Mean Squared Log Error (MSLE), and Median Absolute Error (MAE) provide a comprehensive assessment of the model's performance. Lower values indicate better performance.



Performance Metrics

Targets: Price

Mean Squared Error (MSE): 0.40472

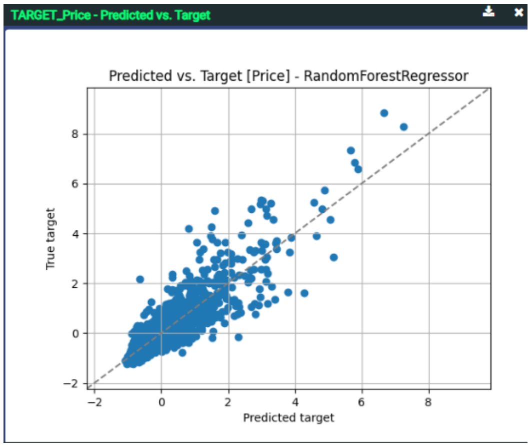
Mean Absolute Error : 0.23925

Mean Squared Log Error : 0.03277

Median Absolute Error : 0.13553

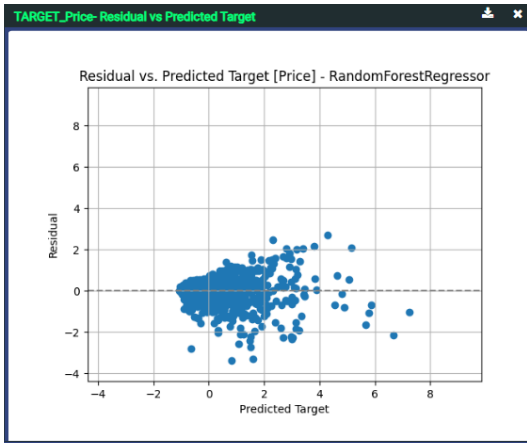
Predicted vs. Target - Random Forest Regressor

The points closely follow the diagonal line (y = x), showing that the predictions are very close to the actual values. This indicates a high level of accuracy in the model's predictions.



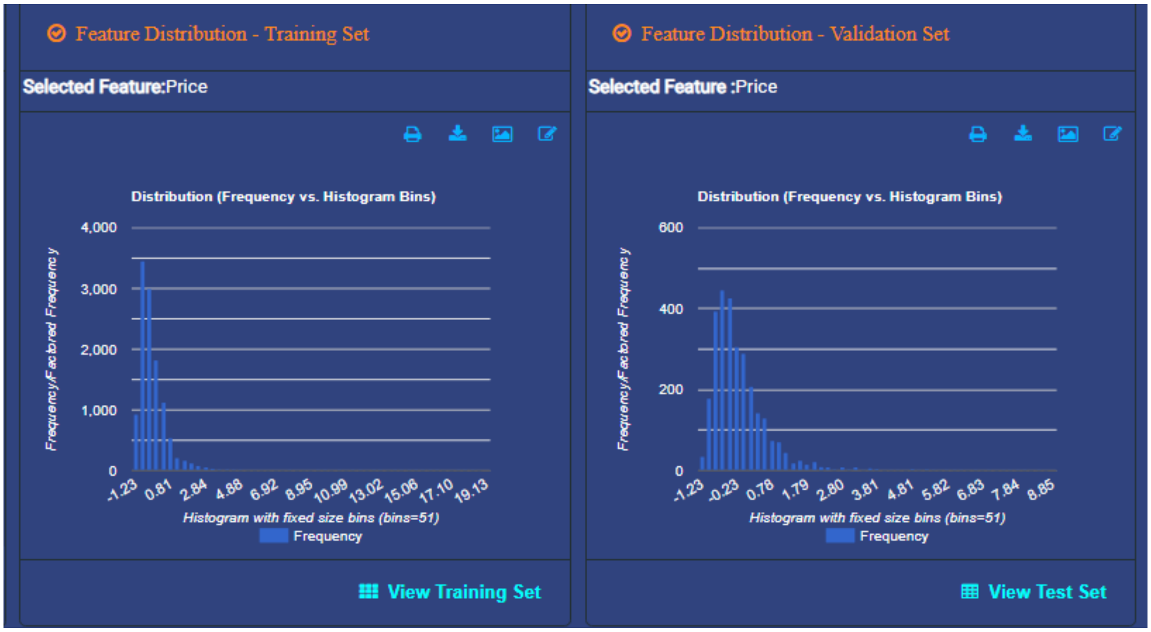
Residual vs. Predicted Target - Random Forest Regressor

Most residuals are clustered around the zero line, indicating that the model predictions are generally accurate. There are few outliers, suggesting the model performs well with minor exceptions.



Feature Distribution - Training and Validation Set (On Price)

Both distributions are centered on the same range, showing that the data is separated and representative of both sets. This ensures that the model can accurately generalize to previously unseen data.



Feature Distribution - Training Set

Selected Feature:Price

Feature Distribution - Validation Set

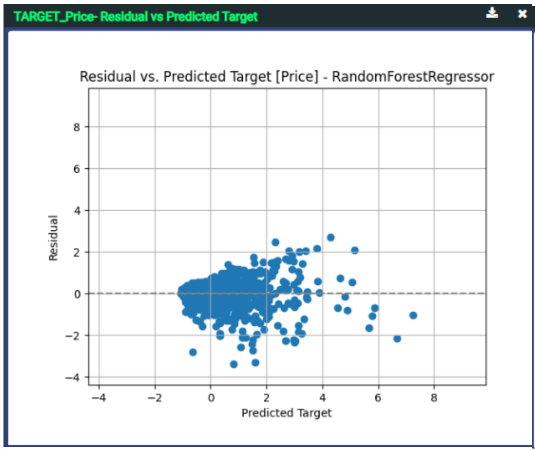
Selected Feature :Price

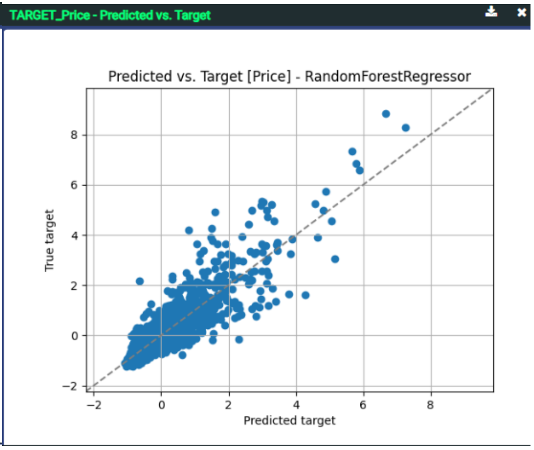
Distribution (Frequency vs. Histogram Bins)
Frequency/Factored Frequency
4,000
3,000
2,000
1,000
0
-1.23
0.81
2.84
4.88
6.92
8.95
10.99
13.02
15.06
17.10
19.13
Histogram with fixed size bins (bins=51)
Frequency
View Training Set
Distribution (Frequency vs. Histogram Bins)
Frequency/Factored Frequency
600
400
200
0
-1.23
-0.23
0.78
1.79
2.80
3.81
4.81
5.82
6.83
7.84
8.85
Histogram with fixed size bins (bins=51)
Frequency
View Test Set

9. Result (Accept/Reject)

The model was accepted by our professor based on its strong performance metrics, low error rate, and good accuracy.

Even as we can see below:





Most residuals cluster around the zero line, indicating that the model predictions are largely correct.

The dots closely follow the diagonal (y = x), indicating that the predictions are quite close to the actual values.

What were the results?

Random Forest had the highest accuracy with the lowest MSLE, MAE, and MSE. Feature importance analysis highlighted living area and number of bedrooms are crucial columns.

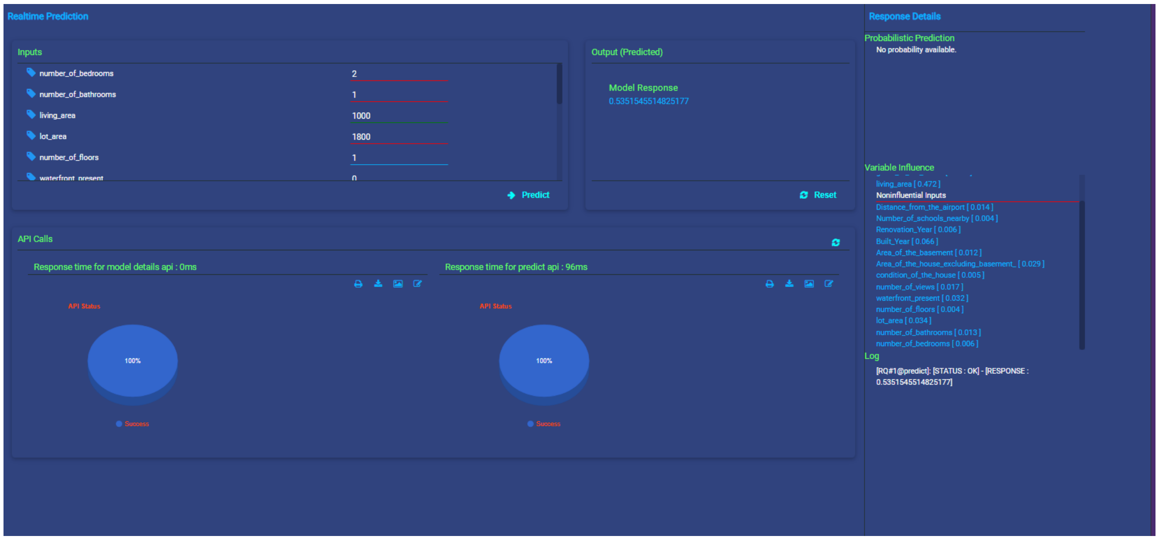
How did you evaluate the performance of your model? What metrics did you use?

Performance was evaluated using MSLE, MAE, MSE, and Median Absolute Error, which provided a comprehensive view of model accuracy and error distribution.

10. Real-time Prediction

Example 1:

Demonstrates the real-time prediction capabilities of the Random Forest Regressor for house prices based on various input features.



Inputs

number_of_bedrooms 2
number_of_bathrooms 1
living_area 1000
lot_area 1800
number_of_floors 1
waterfront_present 0

Output (Predicted)

Model Response
0.5351545514825177

Response time for model details api : 0ms

Response time for predict api : 96ms

Probabilistic Prediction
No probability available.

Variable Influence

living_area [0.472]
Noninfluential Inputs
Distance_from_the_airport [0.014]
Number_of_schools_nearby [0.004]
Renovation_Year [0.006]
Built_Year [0.066]
Area_of_the_basement [0.012]
Area_of_the_house_excluding_basement_ [0.029]
condition_of_the_house [0.005]
number_of_views [0.017]
waterfront_present [0.032]
number_of_floors [0.004]
lot_area [0.034]
number_of_bathrooms [0.013]
number_of_bedrooms [0.006]

Log

```
[RQ#1@predict]: [STATUS : OK] - [RESPONSE :
0.5351545514825177]
```

Inputs

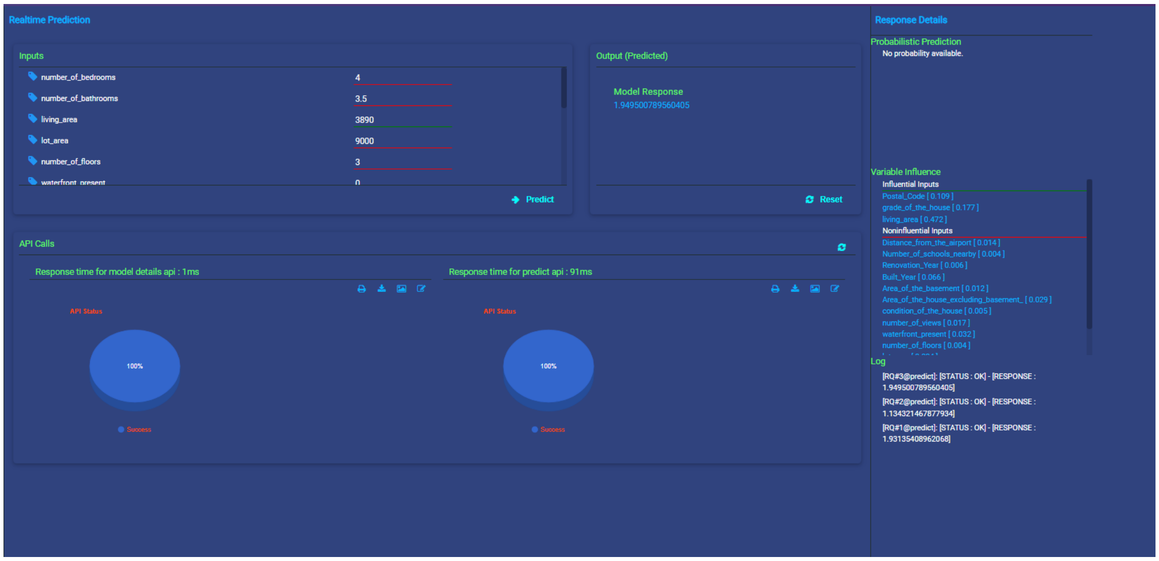
- **Number of Bedrooms**: 2
- **Number of Bathrooms**: 1
- **Living Area**: 1000 SQ FT
- **Lot Area**: 1800 SQ FT
- **Number of Floors**: 1
- **Waterfront Present**: 0 (No)
- **Number of Views**: 2
- **Condition of the House**: 5
- **Grade of the House**: 9
- **Area of the House Excluding Basement**: 600 SQ FT
- **Area of the Basement**: 0 (No basement)
- **Built Year**: 1956
- **Renovation Year**: 0 (There is no renovation)
- **Postal Code**: 122013
- **Number of Schools Nearby**: 1
- **Distance From The Airport**: 6.1 (in Kilometers)

Output (Predicted)

- **Model Response**: The predicted standardized house price is approximately 0.535 which can be around 835000 INR.

Example 2:

Demonstrates the real-time prediction capabilities of the Random Forest Regressor for house prices based on various input features.



number_of_bedrooms 4
number_of_bathrooms 3.5
living_area 3890
lot_area 9000
number_of_floors 3
waterfront_present 0

Model Response
1.949500789560405

Response time for model details api : 1ms

Response time for predict api : 91ms

No probability available.

Influential Inputs
Postal_Code [0.109]
grade_of_the_house [0.177]
living_area [0.472]

Noninfluential Inputs
Distance_from_the_airport [0.014]
Number_of_schools_nearby [0.004]
Renovation_Year [0.006]
Built_Year [0.066]
Area_of_the_basement [0.012]
Area_of_the_house_excluding_basement_ [0.029]
condition_of_the_house [0.005]
number_of_views [0.017]
waterfront_present [0.032]
number_of_floors [0.004]

```
[RQ#3@predict]: [STATUS : OK] - [RESPONSE :
1.949500789560405]
[RQ#2@predict]: [STATUS : OK] - [RESPONSE :
1.134321467877934]
[RQ#1@predict]: [STATUS : OK] - [RESPONSE :
1.93135408962068]
```

Inputs

- **Number of Bedrooms**: 4
- **Number of Bathrooms**: 3.5
- **Living Area**: 3890 SQ FT
- **Lot Area**: 9000 SQ FT
- **Number of Floors**: 3
- **Waterfront Present**: 0 (No)
- **Number of Views**: 0
- **Condition of the House**: 5
- **Grade of the House**: 13
- **Area of the House Excluding Basement**: 4210 SQ FT
- **Area of the Basement**: 980 SQ FT
- **Built Year**: 2001
- **Renovation Year**: 0 (There is no renovation)
- **Postal Code**: 122005
- **Number of Schools Nearby**: 3
- **Distance From The Airport**: 5.6 (in Kilometers)

Output (Predicted)

- **Model Response**: The predicted standardized house price is approximately 1.892 which can be around 1240000 INR.

11. Conclusion

Implementation and Usage of the Solution

How could an organization or institution implement/use the solution?

Organizations and institutions can leverage this house price prediction model in several impactful ways:

1. **Real Estate Agencies**: Can integrate the model into their platforms to provide realistic price estimates for homes, increasing transparency and confidence with clients.
2. **Financial Institutions**: Banks and mortgage brokers can use the model to determine the worth of properties before authorizing loans, lowering the risk of overvaluation.
3. **Government and Urban Planners**: Can use the model to better analyze housing market patterns and make data-driven decisions for urban development initiatives.
4. **Property Investors**: Investment firms can utilize the projections to find undervalued properties and make sound investment decisions.

Implementation Steps:

- **Data Integration**: Integrate the model into existing databases and ensure seamless data flow for real-time predictions.
- **API Deployment**: Develop APIs to allow various systems to access the model's predictions.

Improvements for the Future:

1. **Incorporate Additional Features**: Adding more features like economic indicators, crime rates, and school quality could improve model accuracy.
2. **Advanced Techniques**: Using advanced machine learning techniques like ensemble learning, deep learning, or hybrid models could enhance prediction capabilities.
3. **Regular Updates**: Continuously updating the model with new data will help maintain its accuracy and relevance.
4. **User Feedback**: Incorporating user feedback can help refine the model and the prediction process.

Lessons Learned and Improvement Areas

Lessons Learned:

1. **Good Data Matters**: The quality and accuracy of data are essential when developing reliable predictive models. Missing data, outliers, and irrelevant features have a major impact on model performance.
2. **Feature Engineering is important**: Creating new features and selecting the most relevant ones play a vital role in enhancing model accuracy. Understanding the domain and the data helps in identifying key features.
3. **Choosing the Right Model**: Different models have different strengths. Random Forest Regressor was chosen for its accuracy, but it's essential to compare multiple models.
4. **Interpreting Results**: It is not enough to just develop a model; actionable insights need analysing its output and comprehending the value of each feature.
5. **Practical Use**: Ensure that the model can be used in real-world applications by taking into account issues such as reaction speed, scalability, and simplicity of integration.