REAL ESTATE PRICE PREDICTION

Across the Cities In the USA

MOHAMED SAHAD M
23375041

INTRODUCTION

The real estate market is one of the most dynamic sectors of the economy, significantly influenced by various factors such as location, property size, amenities, and market demand. With the increasing digitization of property listings and user engagement in online platforms like Zillow.com, there is a growing need for accurate, data-driven tools to predict real estate prices. These predictions not only assist buyers and sellers in making informed decisions but also support developers, investors, and policymakers. This project aims to leverage machine learning techniques to build a predictive model for estimating property prices across various cities in the United States using publicly available data.

OBJECTIVE OF THE STUDY

The primary goal of this study is to develop a machine learning model capable of predicting real estate prices across different cities in the USA. The model uses key property features such as:

- Number of bedrooms and bathrooms
- Square footage
- Availability status
- Property location
- Society and balcony information

This model will serve as a prototype similar to Zillow's "Zestimate" feature and demonstrate how machine learning can be used to provide accurate and meaningful real estate price predictions.

DATASET OVERVIEW

availability	location	size	society	total_sqft	bath	balcony	price
46010	New Bedford	2 BHK	Coomee	1056	2	1	39.
Ready To Move	Fort Dodge	4 Bedroom	Theanmp	2600	5	3	1
Ready To Move	Stowell	3 BHK		1440	2	3	[illegible]
Ready To Move	Woodburn	3 BHK	Soiewre	1521	3	1	[illegible]
Ready To Move	Cuyahoga Falls	2 BHK		1200	2	1	[illegible]
Ready To Move	Beetham	2 BHK	DuenaTa	1170	2	1	[illegible]
45795	South Jordan	4 BHK	Jaades	2732	4		2
Ready To Move	Texas	4 BHK	Brway G	3300	4		6
Ready To Move	North Providence	3 BHK		1310	3	1	63.
Ready To Move	Weymouth	6 Bedroom		1020	6		3
45706	Beetham	3 BHK		1800	2	2	[illegible]
Ready To Move	Beetham	4 Bedroom	Prrry M	2785	5	3	2
Ready To Move	Little Rock	2 BHK	Shncyes	1000	2	1	[illegible]
Ready To Move	Eden Prairie	2 BHK		1100	2	2	[illegible]
Ready To Move	Petersburg	3 Bedroom	Skityer	2250	3	2	1
Ready To Move	Sisseton	2 BHK	PrntaEn	1175	2	2	73
Ready To Move	Elgin	3 BHK	Prityel	1180	3	2	[illegible]
Ready To Move	Lynnwood	3 BHK	GrrvaGr	1540	3	3	[illegible]
Ready To Move	Vienna	3 BHK	PeBayle	2770	4	2	2
Ready To Move	Meadville	2 BHK		1100	2	2	[illegible]
Ready To Move	Bismarck	1 BHK		600	1	1	[illegible]
46010	Rockford	3 BHK	She 2rk	1755	3	1	1
Ready To Move	Egton Bridge	4 Bedroom	Soitya	2800	5	2	3
Ready To Move	Lahaina	3 BHK		1767	3	1	1
45979	Opelika	1 RK	Bhe 2ko	510	1	0	25.

This dataset, sourced from Kaggle, contains 13,320 entries and 8 columns. It is designed to support a real estate price prediction model, focusing on various property features across cities in the USA. Each row represents an individual property listing with multiple attributes that influence property value.

COLUMNS DESCRIPTION

Column Name	Description
availability	Indicates whether the property is ready to move or a future availability date (some are numerical, likely to be corrected during cleaning).
location	Specifies the city or town where the property is located (e.g., "New Bedford", "Fort Dodge").
size	Describes the number of bedrooms, typically in the form "2 BHK", "4 Bedroom", etc.
society	Name of the residential society or complex (may contain missing values).
total_sqft	The total area of the property in square feet.
bath	Number of bathrooms in the property.
balcony	Number of balconies in the property.
price	**Target variable** — the property price (presumably in lakhs or thousands of dollars, depending on dataset source).

PROGRAMMING LANGUAGE AND LIBRARIES TO BE USED

In this project, I will use Python as the programming language for real estate price prediction. following libraries will be utilized:

<u>NUMPY</u> **- It will be used for numerical computing and array operations.**

<u>PANDAS</u> **- It will help in data manipulation, preprocessing, and analysis.**

<u>MATPLOTLIB</u> **- It will be used for visualizing data through graphs and plots.**

<u>SCIKIT-LEARN (SKLEARN)</u> **It will provide machine learning algorithms and tools for model building, evaluation, and feature transformation.**

METHODOLGY

KEY STEPS

1 - Data Cleaning
2 - Feature engineering
3 - Outlier Detection & Removal
4 - Model Building

DATA CLEANING METHODS

1 - REMOVING UNWANTED COLUMNS

We remove three columns from the dataset:

- **Availability**
- **Society**
- **Balcony**

Reason: These columns do not provide significant value for price prediction. Since our goal is to build a generic and effective model, removing irrelevant features simplifies the dataset and improves model performance.

	availability	location	size	society	total_sqft	bath	balcony	price
0	46010	New Bedford	2 BHK	Coomee	1056	2.0	1.0	39.07
1	Ready To Move	Fort Dodge	4 Bedroom	Theanmp	2600	5.0	3.0	20.00
2	Ready To Move	Stowell	3 BHK	NaN	1440	2.0	3.0	62.00
3	Ready To Move	Woodburn	3 BHK	Soiewre	1521	3.0	1.0	95.00
4	Ready To Move	Cuyahoga Falls	2 BHK	NaN	1200	2.0	1.0	51.00
5	Ready To Move	Beetham	2 BHK	DuenaTa	1170	2.0	1.0	38.00
6	45795	South Jordan	4 BHK	Jaades	2732	4.0	NaN	204.00
7	Ready To Move	Texas	4 BHK	Brway G	3300	4.0	NaN	00.00
8	Ready To Move	North Providence	3 BHK	NaN	1310	3.0	1.0	63.25
9	Ready To Move	Weymouth	6 Bedroom	NaN	1020	6.0	NaN	370.00

2 - HANDLING MISSING VALUES

- **We identified missing (null) values in the dataset.**
- **Since the dataset contains 13,000+ data points, and the number of missing values is very small, we chose to remove the rows with missing values rather than imputing them.**

	location	size	total_sqft	bath	price
56	Sulphur	4 Bedroom	3010 - 3410	NaN	192.000
81	Kirkwood	4 Bedroom	2957 - 3450	NaN	224.500
224	Sulphur	3 BHK	1520 - 1740	NaN	74.820
344	Freeport	1 BHK	525	NaN	21.530
579	Glendale	NaN	1200 - 2400	NaN	34.185
...	...	...	...	...	...
11496	Freeport	1 BHK	525	NaN	27.000
11569	Wolf Point	NaN	1350	NaN	8.440
12768	Idaho Falls	5 Bedroom	3210	NaN	53.000
12861	Mansfield	4 BHK	2204 - 2362	NaN	121.000
13240	Sulphur	1 BHK	1020 - 1130	NaN	52.570

3 - STANDARDIZING THE 'TOTAL_SQFT' COLUMN

The 'total_sqft' column had inconsistent formats, such as:

- **A single value (e.g., 2600)**
- **A range of values (e.g., 2100 - 2850)**
- **A different unit format (e.g., 34.46 Sq. Meter)**

To maintain consistency, we converted all values into a single numerical format (float) for uniformity.

	location	size	total_sqft	bath	price
30	Redmire	4 BHK	2100 - 2850	4.0	186.000
122	California	4 BHK	3067 - 8156	4.0	477.000
137	Jonesboro	2 BHK	1042 - 1105	2.0	54.005
165	Petersburg	2 BHK	1145 - 1340	2.0	43.490
188	Mansfield	2 BHK	1015 - 1540	2.0	56.800
410	Bismarck	1 BHK	34.46Sq. Meter	1.0	18.500
549	Kirkwood	2 BHK	1195 - 1440	2.0	63.770
648	Meriden	9 Bedroom	4125Perch	9.0	265.000
661	Redmire	2 BHK	1120 - 1145	2.0	48.130
672	Idaho Falls	4 Bedroom	3090 - 5002	4.0	445.000

4 - HANDLING MISSING VALUES

- **We identified missing (null) values in the dataset.**
- **Since the dataset contains 13,000+ data points, and the number of missing values is very small, we chose to remove the rows with missing values rather than imputing them.**

	location	size	total_sqft	bath	price
56	Sulphur	4 Bedroom	3010 - 3410	NaN	192.000
81	Kirkwood	4 Bedroom	2957 - 3450	NaN	224.500
224	Sulphur	3 BHK	1520 - 1740	NaN	74.820
344	Freeport	1 BHK	525	NaN	21.530
579	Glendale	NaN	1200 - 2400	NaN	34.185
...	...	...	...	...	...
11496	Freeport	1 BHK	525	NaN	27.000
11569	Wolf Point	NaN	1350	NaN	8.440
12768	Idaho Falls	5 Bedroom	3210	NaN	353.000
12861	Mansfield	4 BHK	2204 - 2362	NaN	121.000
13240	Sulphur	1 BHK	1020 - 1130	NaN	52.570

FEATURE ENGINEERING

1 - BHK (NUMBER OF BEDROOMS, HALL, AND KITCHEN)

Extract the Integer part from the column 'size'

	location	size	total_sqft	bath	price	BHK
0	New Bedford	2 BHK	1056.0	2.0	39.07	2
1	Fort Dodge	4 Bedroom	2600.0	5.0	120.00	4
2	Stowell	3 BHK	1440.0	2.0	62.00	3
3	Woodburn	3 BHK	1521.0	3.0	95.00	3
4	Cuyahoga Falls	2 BHK	1200.0	2.0	51.00	2
5	Beetham	2 BHK	1170.0	2.0	38.00	2
6	South Jordan	4 BHK	2732.0	4.0	204.00	4
7	Texas	4 BHK	3300.0	4.0	600.00	4
8	North Providence	3 BHK	1310.0	3.0	63.25	3
9	Weymouth	6 Bedroom	1020.0	6.0	370.00	6

2 - PRICE PER SQUARE FOOT

Calculated using the formula:

$$\text{price\_per\_sqrft} = \frac{\text{price} \times 100000}{\text{total\_sqft}}$$

	location	size	total_sqft	bath	price	BHK	price_per_sqrt
0	New Bedford	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Fort Dodge	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Stowell	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Woodburn	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Cuyahoga Falls	2 BHK	1200.0	2.0	51.00	2	4250.000000
5	Beetham	2 BHK	1170.0	2.0	38.00	2	3247.863248
6	South Jordan	4 BHK	2732.0	4.0	204.00	4	7467.057101
7	Texas	4 BHK	3300.0	4.0	600.00	4	18181.818182
8	North Providence	3 BHK	1310.0	3.0	63.25	3	4828.244275
9	Weymouth	6 Bedroom	1020.0	6.0	370.00	6	36274.509804

3 - REDUCING THE NUMBER OF CATEGORIES IN 'LOCATION'

The 'location' column originally contained 1,289 unique categories, which is too high compared to our total number of data points. Keeping too many categories can lead to:

- **High dimensionality issues**
- **Sparse data problems**
- **Poor model generalization**

To address this:

- **We calculated the frequency of each location in the dataset.**
- **Locations that appear fewer than 10 times are grouped into a new category called 'Other'.**

```
Green River     10
Olive Branch    10
Frederick       10
Wasilla         10
Gunnerside      10
                ..
Natchez          1
Nenthead         1
Cayce            1
New Bern         1
Lakeland         1
```

	location	size	total_sqft	bath	price	BHK	price_per_sqrt
0	New Bedford	2 BHK	1056.0	2.0	39.07	2	3699.810606
1	Fort Dodge	4 Bedroom	2600.0	5.0	120.00	4	4615.384615
2	Stowell	3 BHK	1440.0	2.0	62.00	3	4305.555556
3	Woodburn	3 BHK	1521.0	3.0	95.00	3	6245.890861
4	Cuyahoga Falls	2 BHK	1200.0	2.0	51.00	2	4250.000000
...	...	...	...	...	...	...	...
13315	Beetham	5 Bedroom	3453.0	4.0	231.00	5	6689.834926
13316	other	4 BHK	3600.0	5.0	400.00	4	11111.111111
13317	Lynnwood	2 BHK	1141.0	2.0	60.00	2	5258.545136
13318	Kaysville	4 BHK	4689.0	4.0	488.00	4	10407.336319
13319	Baltimore	1 BHK	550.0	1.0	17.00	1	3090.909091

OUTLIER DETECTION & REMOVAL

1 - UNUSUAL BEDROOM-TO-SQUARE-FOOT RATIO

Understanding the Issue in Real Estate Data:
In real estate, there is a logical relationship between the number of bedrooms (BHK) and the total square footage of a property. A larger number of bedrooms should typically correspond to a larger total area. However, in some cases, the dataset might contain illogical or unrealistic values, such as:

- **A 2 BHK apartment with only 500 sqft**
- **A 6 BHK property with just 600 sqft (which means each bedroom has only 100 sqft)**

These are outliers because such properties are highly unlikely to exist in the real world.

Our Approach: Setting a Minimum Square Footage per Bedroom
To identify and remove these anomalies, we introduce a "square feet per bedroom" threshold:

- **Compute:**

$$\text{sqft per BHK} = \frac{\text{total\_sqft}}{\text{BHK}}$$

- **Set a threshold of 300 sqft per bedroom**
- **Remove rows where sqft per BHK is less than 300**

📌 Example:

BHK	Total Sqft	Sqft per BHK	Outlier?
2	500	250	✅ Yes
3	1200	400	❌ No
6	600	100	✅ Yes
4	1400	350	❌ No

2 - FILTERING EXTREME PRICE PER SQUARE FOOT VALUES

Understanding the Issue in Real Estate Data:

- **When analyzing real estate pricing, we often find extreme variations in price per square foot (sqft). While some variation is expected (due to location, amenities, etc.), certain values may be too low or too high to be realistic for a generic model.**

```
count      12457.000000
mean        6308.427888
std         4167.968413
min          267.829813
25%         4210.526316
50%         5294.117647
75%         6916.666667
max       176470.588235
Name: price_per_sqrt, dtype: float64
```

Look at the statistics of price per sqft column,

- **Minimum price per sqft: ₹267.83 (which is unusually low)**
- **Maximum price per sqft: ₹176,470.58 (which is extremely high)**

- **Using Standard Deviation to Identify Outliers**
- **Since we assume the price per square foot follows a normal distribution, we can apply standard deviation (std) filtering to detect and remove outliers.**
-
- **Why Use Standard Deviation?**
- **In a normal distribution:**
- **68% of the data falls within ±1 standard deviation (σ) of the mean (μ).**
- **95% of the data falls within ±2σ.**
- **99.7% of the data falls within ±3σ.**

To remove extreme values, we set a threshold of 1 standard deviation from the mean:

$$\text{Lower Bound} = \mu - \sigma$$

$$\text{Upper Bound} = \mu + \sigma$$

Any price per sqft value outside this range is considered an outlier and removed.

3 - CHECKING LOGICAL PRICE RELATIONSHIPS FOR BHK AND SQUARE FOOT AREA

Understanding the Issue in Real Estate Pricing:

In real estate pricing, a logical expectation is:

- **For the same location and similar square footage, properties with more bedrooms (BHK) should not be cheaper than those with fewer bedrooms.**

However, in our dataset, we found anomalies:

Texas	3 BHK	Raesta	1210	2	2	81
Texas	3 BHK	Brway G	1800	3	2	240
Texas	3 BHK	PhestOn	2533	3	3	425
Texas	2 Bedroom		432	2	1	65
Texas	2 BHK		720	2	2	65
Texas	6 BHK		3000	6	3	250
Texas	2 BHK	ProdsWe	1268	2	1	127

This is illogical because:

- **The 3 BHK property (1210 sqft) is priced at ₹81 lakhs,**
- **While a 2 BHK property (1268 sqft) is priced at ₹127 lakhs,**
- **Even though the 2 BHK has a similar square footage, it is significantly more expensive, which is unrealistic in a normal market.**

These types of anomalies are likely outliers or mispriced entries that can mislead our machine learning model.

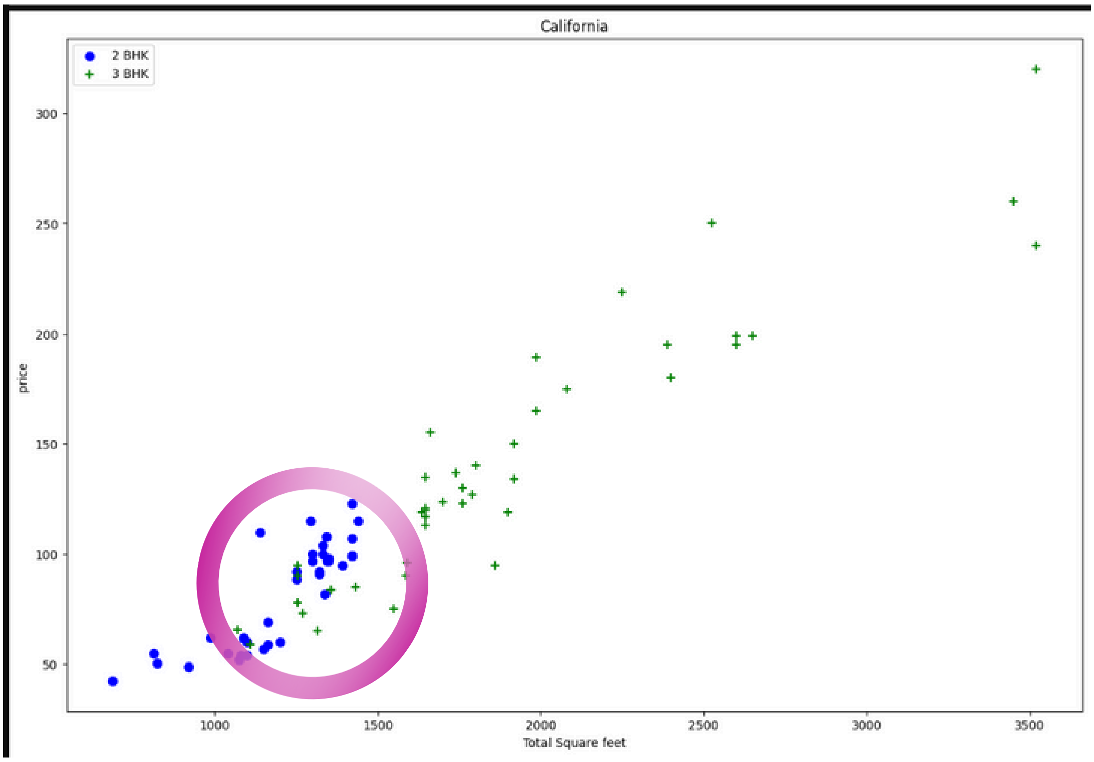
Approach to Detect & Remove These Outliers:
To systematically find and remove such data points, we will:

Step 1: Visualize Data Using a Scatter Plot

- **Plot a scatter diagram with:**
  - **X-axis → Total square footage (total_sqft)**
  - **Y-axis → Price (price)**
  - **Different colors for different BHK values**
- **This helps visually identify points where lower BHK properties are priced significantly higher than higher BHK properties for similar square footage in the same location.**

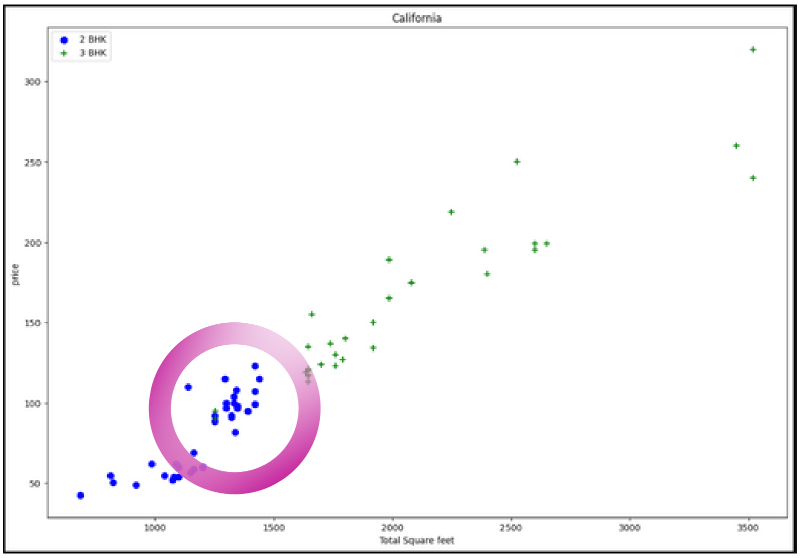
Step 2: Define a Function to Detect These Anomalies

- **For each location, group properties by square footage and BHK count.**
- **Compare prices of different BHKs with similar sqft values.**
- **Identify cases where lower BHK properties are priced higher than higher BHK properties.**
- **Remove these anomalies.**



Here, We can see that, In the same location (California) with same square foot, price for 3 BHK is less than price of 2 BHK

After the application of the function, most of the outliers are removed.



4 - IDENTIFYING UNUSUAL BATHROOM COUNTS

Understanding the Issue in Real Estate Pricing
In real-world real estate, the number of bathrooms is usually related to the number of bedrooms (BHK) in a property. While luxury properties may have more bathrooms than bedrooms, in a typical scenario, the number of bathrooms is close to or slightly higher than the number of bedrooms.
However, some listings in our dataset might have excessively high bathroom counts, which could indicate:

- **Data entry errors (e.g., mistakenly entering 10 bathrooms instead of 1)**
- **Unrealistic property configurations (e.g., a 3 BHK house with 8 bathrooms, which is rare in a normal setting)**
- **Outlier properties that do not fit a general model**

Our Approach: Setting a Threshold for Bathroom Counts
📌 **Rule for Detecting Outliers:**

- **If number of bathrooms > (number of bedrooms + 2) → Mark it as an outlier and remove it**

This ensures that properties with an excessive number of bathrooms relative to their bedrooms are filtered out.

Example Analysis

Property	BHK	Bathrooms	Condition	Outlier?
A	2	2	Normal	❌ No
B	3	4	Normal	❌ No
C	4	7	Too many bathrooms (4 + 2 = 6, but found 7)	✅ Yes
D	5	8	Too many bathrooms (5 + 2 = 7, but found 8)	✅ Yes

MODEL BUILDING

When building a machine learning model, all input features must be numerical because algorithms cannot process text directly. In our dataset, the "location" column is a categorical variable (text-based). To use it in a model, we need to convert it into a numerical format.

	location	total_sqft	bath	price	BHK
0	Alamogordo	1280.0	2.0	56.0	3
1	Alamogordo	1275.0	2.0	52.0	2
2	Alamogordo	2150.0	4.0	100.0	4
3	Alamogordo	1150.0	2.0	42.0	2
4	Alamogordo	1110.0	2.0	53.0	2
5	Alamogordo	1060.0	2.0	48.0	2
7	Alamogordo	1063.0	2.0	42.0	2
8	Alamogordo	1470.0	2.0	75.0	3
10	Alamogordo	1050.0	2.0	48.0	2
11	Alamogordo	1132.0	2.0	42.0	2

One of the most effective ways to do this is One-Hot Encoding (OHE), which creates separate binary (0/1) columns for each unique category in the "location" column.

That is,
When we apply one-hot encoding using the 'pd.get_dummies()' (pandas dummies) method in Pandas, it does the following:
1️⃣ **Creates new columns for each unique category in the "location" column.**
2️⃣ **Assigns 1 to the column corresponding to the location of the row.**
3️⃣ **Assigns 0 to all other location columns.**

For example, consider the original dataset:

Index	Location	Total Sqft	Bath	Price	BHK
0	Texas	1280	2	56	3
1	New York	1275	2	52	2
2	Texas	2150	4	100	4
3	Florida	1150	2	42	2

After applying **one-hot encoding**, the dataset transforms into:

Index	Total Sqft	Bath	Price	BHK	Location_Florida	Location_NewYork	Location_Texas
0	1280	2	56	3	0	0	1
1	1275	2	52	2	0	1	0
2	2150	4	100	4	0	0	1
3	1150	2	42	2	1	0	0

Each location is now represented as a **separate colu** **with binary values (0 or 1).**

One-hot encoding creates one column per category, but using all these columns can introduce a problem called the Dummy Variable Trap (DVT).

📌 **What is the Dummy Variable Trap?**

- **If we have N categories, one-hot encoding creates N new columns.**
- **However, using all N columns creates redundant information because knowing the values of (N-1) columns automatically determines the value of the last column.**
- **This causes multicollinearity, where one column is a linear combination of the others, leading to inaccurate predictions.**

🚀 **How to Fix It?**

- **To avoid this trap, we drop one column (any one of the new location columns).**
- **If all other columns are 0, the model can infer that the row belongs to the dropped category.**

For example, if we drop "**Location_Texas**", then:

Index	Total Sqft	Bath	Price	BHK	Location_Florida	Location_NewYork
0	1280	2	56	3	0	0
1	1275	2	52	2	0	1
2	2150	4	100	4	0	0
3	1150	2	42	2	1	0

Now, when all the columns have **0**, it means the location is **Texas**.

In your case, you are **dropping the "other" column** to avoid the dummy variable trap.

MODEL SELECTION USING GRID SEARCH CV

Once we have prepared our dataset, the next step is to select the best model for predicting real estate prices. Instead of manually testing different models and tuning parameters one by one, we use Grid Search CV, an efficient technique from Scikit-learn that automates this process.

What is Grid Search CV?

📌 **Grid Search CV (Cross-Validation) is a powerful technique that:**

✅ **Tests multiple models and hyperparameters to find the best combination.**

✅ **Performs cross-validation, ensuring the model generalizes well to unseen data.**

✅ **Reduces human effort, as it systematically tries all possible combinations.**

Models Used in This Project:
1️⃣ **Linear Regression**
2️⃣ **Lasso Regression (L1 Regularization)**
3️⃣ **Decision Tree Regressor**

SELECTING THE BEST MODEL BASED ON GRID SEARCH CV RESULTS

After applying Grid Search CV, we obtained the following model performance scores:

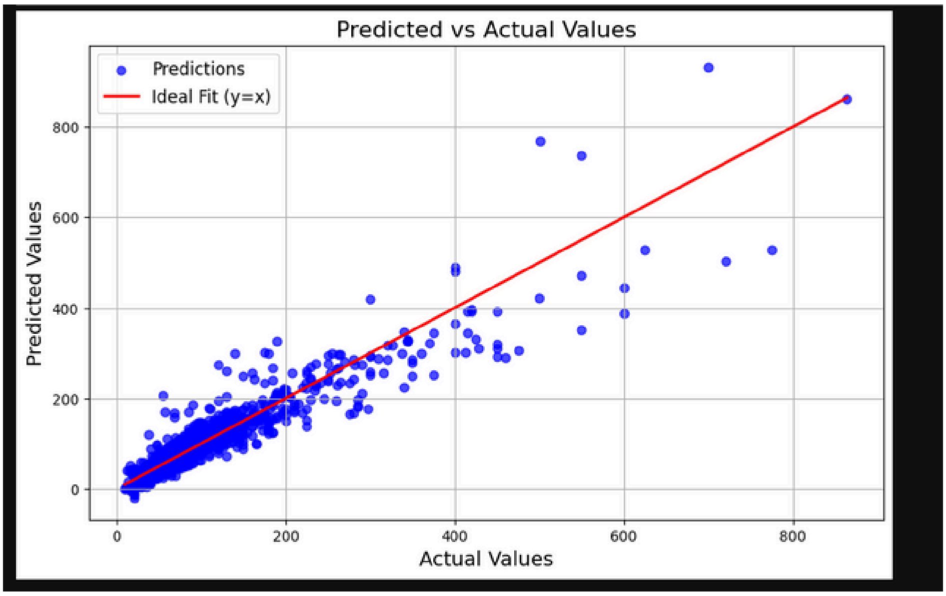
Model	Best Score (R^2 Value)	Best Parameters
Linear Regression	**0.840**	`{'regressor__fit_intercept': False, 'scaler': ...}`
Lasso Regression	0.692	`{'alpha': 1, 'selection': 'random'}`
Decision Tree	0.689	`{'criterion': 'friedman_mse', 'splitter': 'random'}`

Final Decision:
Since Linear Regression has the highest R^2 score (0.84), it is the most effective model for predicting real estate prices. We will proceed with Linear Regression for final model training and predictions.

RESULTS AND DISCUSSION

We have built a Linear Regression model for real estate price prediction and achieved an R^2 (coefficient of determination) value of 0.867.

- **What does this R^2 value mean?**
  - **The R^2 value of 0.867 indicates that 86.7% of the variance in house prices can be explained by the features in your dataset (such as location, square footage, number of bathrooms, etc.).**
  - **This is a strong indication that our model performs well in predicting house prices.**



✅ **Strong Correlation:**
- **Most of the blue points are clustered around the red line, indicating that the model's predictions are close to the actual values.**
- **This suggests that the model is making reasonably accurate predictions.**

✅ **Good Fit in Lower Price Range:**

- **For lower actual prices (near 0–200), predictions align very well with actual values.**
- **This means the model is very reliable for predicting mid-range properties.**

⚠ **Higher Prediction Errors for Expensive Properties:**

- **As property prices increase (above 400+), the spread of blue dots increases, meaning that the model struggles to predict extremely high-priced houses accurately.**
- **This could suggest that luxury properties have unique features that are not fully captured in the dataset.**

General Observations:

- **The model shows a strong positive correlation between features like square footage and number of bathrooms with the property price.**
- **Location also plays a significant role in price variability, reaffirming real-world real estate valuation principles.**

CONCLUSION

This project successfully demonstrates how machine learning, specifically linear regression, can be applied to predict real estate prices in the United States. The model achieves high accuracy in predicting prices for most properties, especially those in the low to mid-range market. However, improvements can be made for high-value properties, possibly by incorporating additional features like property age, nearby amenities, crime rates, and school ratings. Future work may also involve experimenting with more advanced algorithms (e.g., Random Forest, XGBoost) to enhance prediction accuracy and robustness.

BIBLIOGRAPHY AND REFERENCES

1 - Kaggle Dataset - Real Estate Price Prediction Dataset
https://www.kaggle.com/datasets/arslanali4343/real-estate-dataset

2-Model building in machine learning
https://www.techtarget.com/searchenterpriseai/feature/How-to-build-a-machine-learning-model-in-7-steps

3-Scikit-learn Documentation.
https://scikit-learn.org/stable/documentation.html

4-NumPy Documentation.
https://numpy.org/doc

5-Pandas Documentation
https://pandas.pydata.org/docs/

6-Matplotlib Documentation.
https://matplotlib.org/stable/contents.html