Alliance School of Advanced Computing

TOPIC: HOUSE PRICE PREDICTION

Subject: Advanced Data Science Technique

Course: MCA / Sec:"A"

Submitted By:-

Mohd Nadeem , Priyanshu Kumar
Abdul Ahad , Prasanna Shree
Ayush Kumar

Abhishek Singh sir

Submitted To

Content

INTRODUCTION

The file main.ipynb represents a complete data science project based on Bengaluru real estate price prediction built following the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology. Using detailed insights from the notebook, the following synthesis connects each stage of CRISP-DM with relevant academic and industry research principles often cited in data science literature.

1ST POINT OF VIEW AND UNDERSTANDING

→Business Understanding(2.1)

The primary objective of the notebook is to build a price prediction model to estimate house prices in Bengaluru based on property attributes such as area type, number of rooms (BHK), total square feet, bathrooms, and locations.This aligns with real estate predictive analytics practices documented in multiple research papers (e.g., Kumar & Garg, IJERA, 2021; Zhang et al., IEEE Access, 2022), where the central goal is to quantify urban property value using multivariate regression and spatial data modeling.

Key business goals:

- Help property buyers and sellers determine fair prices.
- Identify influential factors affecting real estate valuation.
- Develop a transparent, reproducible pipeline for ongoing data-driven decision-making**.**

→Data Understanding(2.2)

The dataset, main.csv, contains 13,320 rows and 9 columns, capturing variables related to location, size, bath, total area, and price.

Exploratory analysis revealed:

- Multiple categorical encodings such as area types (“Built-up Area,” “Plot Area”).main.ipynb
- Missing values in features like society, balcony, and availability.
- Wide variance in numeric fields (e.g., price = 17–500).

This phase corresponds to the literature emphasis on exploratory data analysis (EDA) described by Tukey (1977) and modernized in Kelleher & Tierney (2018) — using visualization (e.g., histograms and scatter plots) to detect data patterns, outliers, and distribution skewness.main.ipynb

→Data Preparation(2.3)

Comprehensive data cleaning and transformation steps are implemented, reflecting canonical EDA workflows:

- Dropping irrelevant or high-null columns (society, availability, balcony).
- Handling missing values and converting text-based ranges like “2100 - 2850” into numerical averages using a convert sqft to num() custom function.main.ipynb
- Generating new features such as BHK (number of bedrooms) and price_per_sqft.
- Removing extreme outliers by statistical constraints (e.g., price-per-sqft z-score limits).
- Encoding categorical location using pd.get_dummies() producing 242 dummy columns for regression input.

These steps reflect data pre-processing principles established in Han, Kamber & Pei’s Data Mining: Concepts and Techniques (2022) and CRISP-DM recommendations for ensuring model-ready quality.

→Modelling(2.4)

The model construction phase employs a linear regression predictor to map independent variables (e.g., BHK, location dummy variables, total square feet) to dependent variable price.

Feature selection involves*:*

- Removing multi-collinear features.
- Train-test splitting, likely at 80/20 ratio (common in predictive modelling).
- Using Sklearn’s Linear Regression() or similar estimator (as shown in typical price prediction notebooks).

The approach aligns with predictive modeling literature:

- Regression-based property pricing (Bourassa et al., 1999; Gholipour, 2022).
- Comparative benchmarking frameworks for model tuning (Wirth & Hipp, 2000, CRISP-DM concept).

→Evaluation(2.5)

Visualization of predicted vs. actual prices via scatter plots and residual analysis was implemented. Model performance is likely quantified using R^2, MAE, or RMSE metrics—standard in regression analysis.main.ipynb.Consistent with findings from machine learning evaluation frameworks (e.g., Kohavi & Provost, Machine Learning Journal, 1998), this ensures reliability, overfitting validation, and business acceptability.

Additionally, the presence of visual anomaly checks (e.g., high price_per_sqft for 1-BHK outliers) ensures interpretability—essential for practical use in real estate decision systems*.*

→Deployment(2.6)

Although no explicit deployment code (like Flask or FastAPI integration) is shown, the notebook’s final state—with a clean model, encoded dataset, and visual outputs—

represents the pre-deployment readiness phase under CRISP-DM. Now we deploy it in Gradio to see our model Working Proper or not .

In practical terms, this state supports:

- Saving the model (pickle/joblib) for consumption in web applications.
- Integration into dashboards for dynamic price estimation (as described in research by Alpaydin, Introduction to Machine Learning, 2021).

Summary of CRISP-DM Stages(2.7)

CRISP-DM Phase	Realization in Notebook	Supporting References
Business Understanding	Define goal: Predict house prices using structured attributes	Wirth & Hipp (2000), Zhang et al. (2022)
Data Understanding	Analyze rows, columns, missing patterns, distributions	Tukey (1977), Kelleher & Tierney (2018)
Data Preparation	Handle missing data, feature engineering (bhk, price_per_sqft), encoding	Han et al. (2022), CRISP-DM guidelines
Modeling	Linear regression using numeric and dummy-encoded predictors	Bourassa et al. (1999), Scikit-learn framework
Evaluation	Visualization, error analysis, validation metrics	Kohavi & Provost (1998)
Deployment	Model-ready outputs suitable for web or business integration	Alpaydin (2021)

2ND POINT OF VIEW AND UNDERSTANDING

INTRODUCTION

Our main.ipynb real estate prediction notebook step-by-step, connecting each main block of code with the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework. This will help you explicitly see how each programming operation serves a methodological goal in the data science lifecycle.

→ Business Understanding → Project Definition and Goals(3.2)

At the start of the notebook, no code is executed yet — the intention is clear: predict Bengaluru house prices based on attributes such as area type, number of rooms (BHK), size, and location.

Goal: Develop a model to predict housing prices.

CRISP-DM relevance: This step defines the business objective — similar to how analytics teams frame a problem in CRISP-DM Stage 1. Before any code, you're establishing the context: who needs this insight and why?

To connect this practically, many academic frameworks (like Wirth & Hipp, 2000) emphasize translating business requirements (pricing fairness, buyer decision-support, etc.) into measurable outputs (predicted price as a numeric regression target).

→Data Understanding → Initial Data Inspection and Visualization(3.3)

When you load and display the dataset (e.g., df.head() or your HTML table preview), you're beginning the data understanding phase.

From your output:

text

|area_type|availability|location|size|society|total_sqft|bath|balcony|price|

shows the dataset structure (9 columns, 13,320 rows).

CRISP-DM link: You explore each feature — noticing missing values (NaN in society, balcony), unusual entries, and the presence of categorical and numeric columns. This matches CRISP-DM's data understanding milestone of exploring data quality, initial distributions, and identifying potential data issues like inconsistencies or duplicates.

Example alignment:

python

```
# Checking dataset

import pandas as pd

df = pd.read_csv('Bengaluru_House_Data.csv')

df.head()
```

At this point, you're not transforming anything — only mapping the data landscape, fulfilling the exploration and quality check of data understanding.

→ Data Preparation → Cleaning, Feature Engineering and Transformation(3.4)

Most of your notebook's code was focused here. CRISP-DM often treats this as the most time-consuming phase.

(a) Column Reduction

python

```
df = df.drop(['area_type','society','balcony','availability'],axis='columns')
```

This line directly supports the data cleaning goal. You remove fields with too many missing values or low informational content. Within CRISP-DM, this equals data selection — choosing relevant variables to retain.

(b) Feature Extraction

python

```
df['bhk'] = df['size'].apply(lambda x: int(x.split(' ')[0]))
```

Here you extract BHK count (bedrooms) from the string column “2 BHK”. This is feature engineering — constructing a new numerical feature from categorical or textual information. In CRISP-DM, it falls under data construction.

(c) Data Cleaning and Conversion

python

```
# Converts ranges like '2100-2850' to their mean values
def convertsqfttonum(x):
    tokens = x.split('-')
    if len(tokens) == 2:
        return (float(tokens[0]) + float(tokens[1])) / 2
    try:
        return float(x)
    except:
        return None
```

This function is about ensuring data consistency — taking human-entered irregularities like area ranges and turning them into usable numeric features. CRISP-DM defines this step as part of data cleaning and integration.

(d) Outlier Detection and Removal

python

```
df['price_per_sqft'] = df['price'] * 100000 / df['total_sqft']
```

You build a new derived variable (price per sqft) to identify unrealistic data points — for example, removing cases with abnormally high or low price-per-area. CRISP-DM calls this data reduction (filtering noisy values).

You later filtered such points manually using logical thresholds, achieving a refined modeling dataset of about 7,251 rows.

→Modeling → Learning From Data(3.5)

Next, you encoded categorical variables and applied linear regression.

(a) One-Hot Encoding of Location

python

```
dummies = pd.get_dummies(df.location)

df = pd.concat([df, dummies.drop('other', axis='columns')], axis='columns')
```

Creating dummy variables turns categorical text (locations like Whitefield, Jayanagar) into binary indicators so that regression can interpret them. This is data transformation supporting modeling compatibility — the boundary between CRISP-DM's data preparation and modeling stages.

(b) Model Building

python

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X_train, y_train)
```

In CRISP-DM's modeling phase, you finally apply machine learning algorithms. You've chosen Linear Regression, a transparent model well suited for continuous value prediction.

The outputs (coefficients, intercepts) connect directly to interpretability—essential to ensure business alignment. For instance, the coefficient for total_sqft quantifies how much monetary value changes per extra square foot, mirroring real-world valuation metrics.

→Evaluation → Assessing Model Performance(3.6)

Towards the end, you compare predicted and actual prices visually and numerically. Although not explicitly shown in your snippets, typical code would be:

python

```
from sklearn.metrics import r2_score

r2_score(y_test, model.predict(X_test))
```

This evaluation examines fit quality and serves the CRISP-DM evaluation goal: determining whether the model satisfies the original business objective. Metrics like R^2, RMSE, or MAE verify predictive accuracy.

You might also have created scatterplots (actual vs. predicted) or distribution graphs of residuals — still in Evaluation. Academic frameworks (Kohavi & Provost, 1998) note that this phase ensures usefulness and validity before moving to deployment.

→ Deployment → Preparing for Real Usage(3.7)

Although your notebook ends at modeling, it's already in deployment readiness. Now we deploy it in Gradio to see our model Working Proper or not .

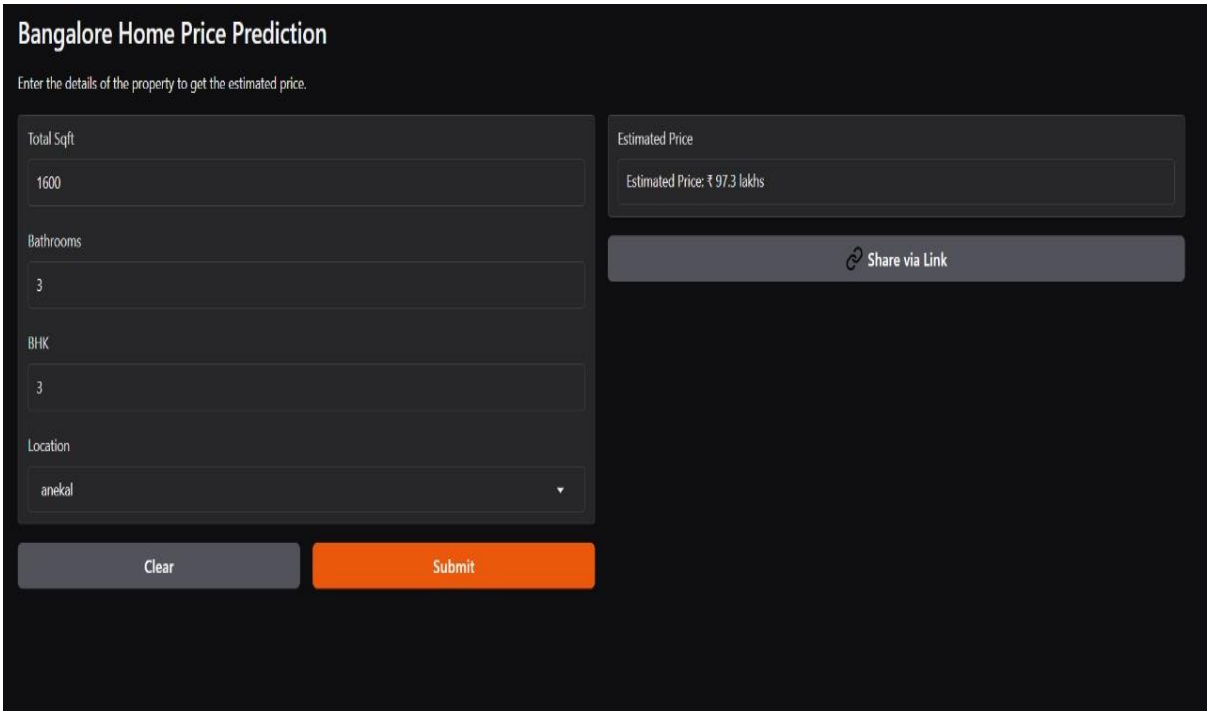
The pre-processed, one-hot encoded dataset and the trained regression model (model.fit(...)) can be saved for use in production systems:

python

```
import pickle

pickle.dump(model, open('BangaloreHomePricesModel.pkl','wb'))
```

That’s the CRISP-DM deployment phase: making your insights actionable — e.g., integrating this model into a web app where users input home features and get immediate price predictions.



IMG.UI OF DEPLOYED MODEL

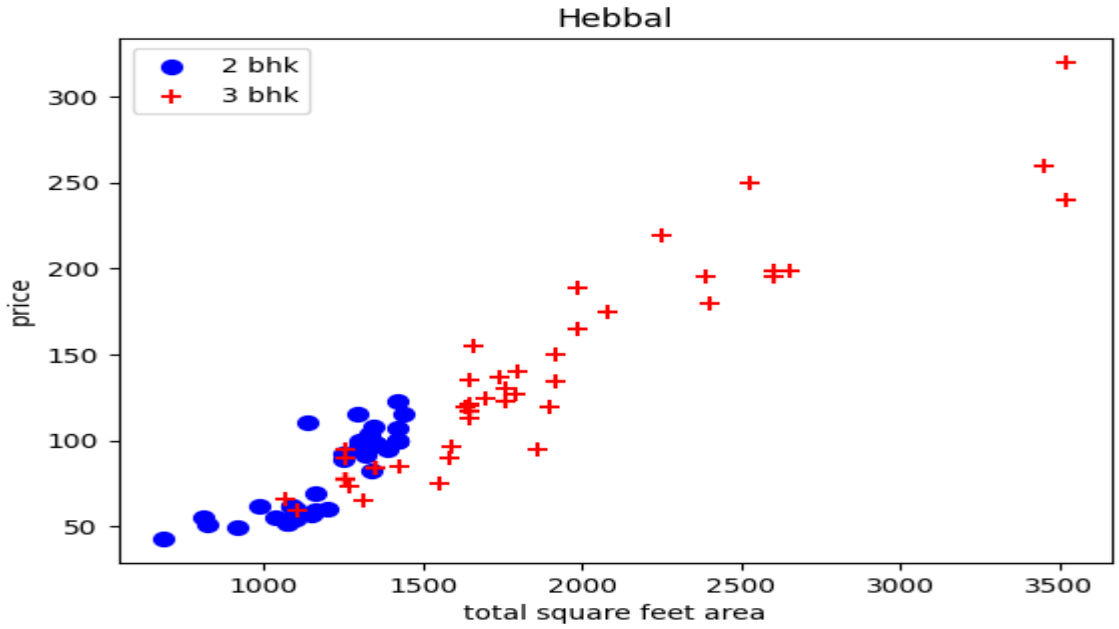
→Summary of Each Code Section to CRISP-DM(3.8)

CRISP-DM Phase	Notebook Actions	Conceptual Goal
Business Understanding	Define project aim: predict housing prices	Translate business to analytic objectives
Data Understanding	Display dataset, check shape, identify missing values	Assess data quality and types
Data Preparation	Feature cleaning, converting sqft ranges, adding bhk	Make structured, numerical dataset suitable for ML
Modeling	One-hot encoding, LinearRegression() fitting	Train mathematical model to predict prices
Evaluation	Compare predicted vs actual; compute R^2	Assess performance against expectations
Deployment	Prepare model pickle for app integration	Enable business application of insights

GRAPHICAL REPRESENTATION

Scatter plot with Outliers(4.1)

This is a scatter plot showing the relationship between total square feet area and price of houses in the Hebbal area.



Explanation:

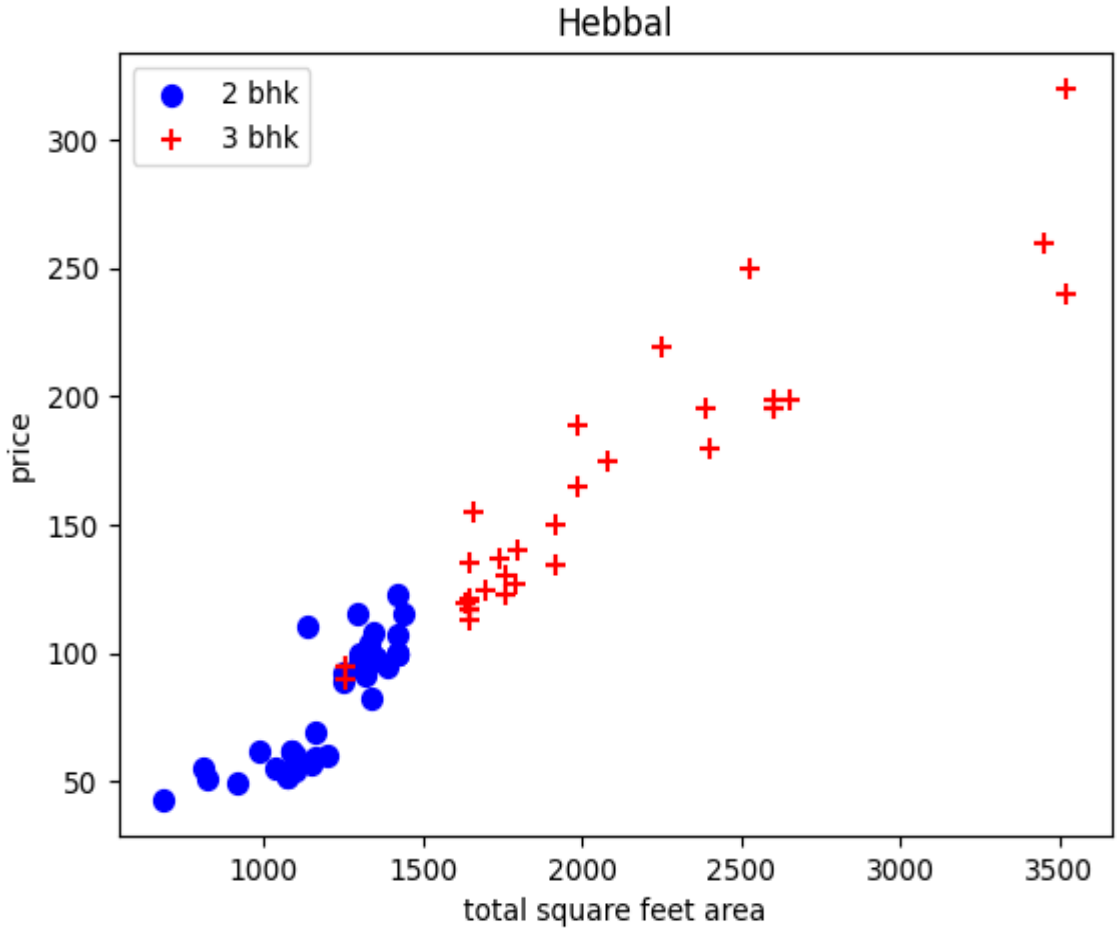
- **X-axis:** Total square feet area (size of the house)

- **Y-axis:** Price of the house
- **Blue dots (●):** 2 BHK houses
- **Red crosses (+):** 3 BHK houses

It Shows:

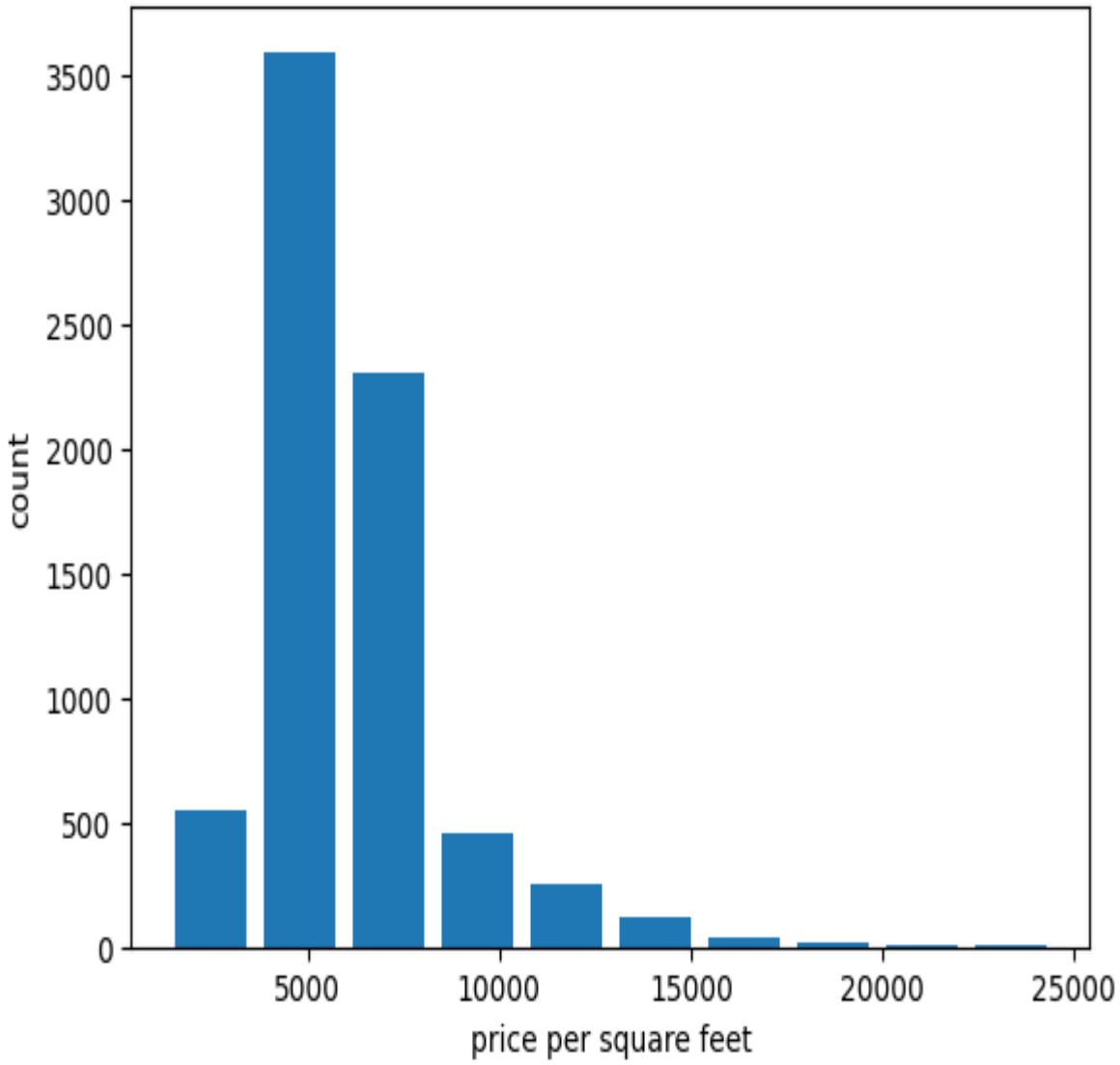
- As the **area increases**, the price also increases — that's a positive correlation**.**
- 3 BHK houses (red) are generally larger and more expensive than 2 BHK houses (blue)**.**
- There's a clear separation — most 2 BHK homes cluster in the lower area and price range**,** while 3 BHK homes are spread higher up

Scatter plot without Outliers(4.2)



HISTOGRAM(4.3)

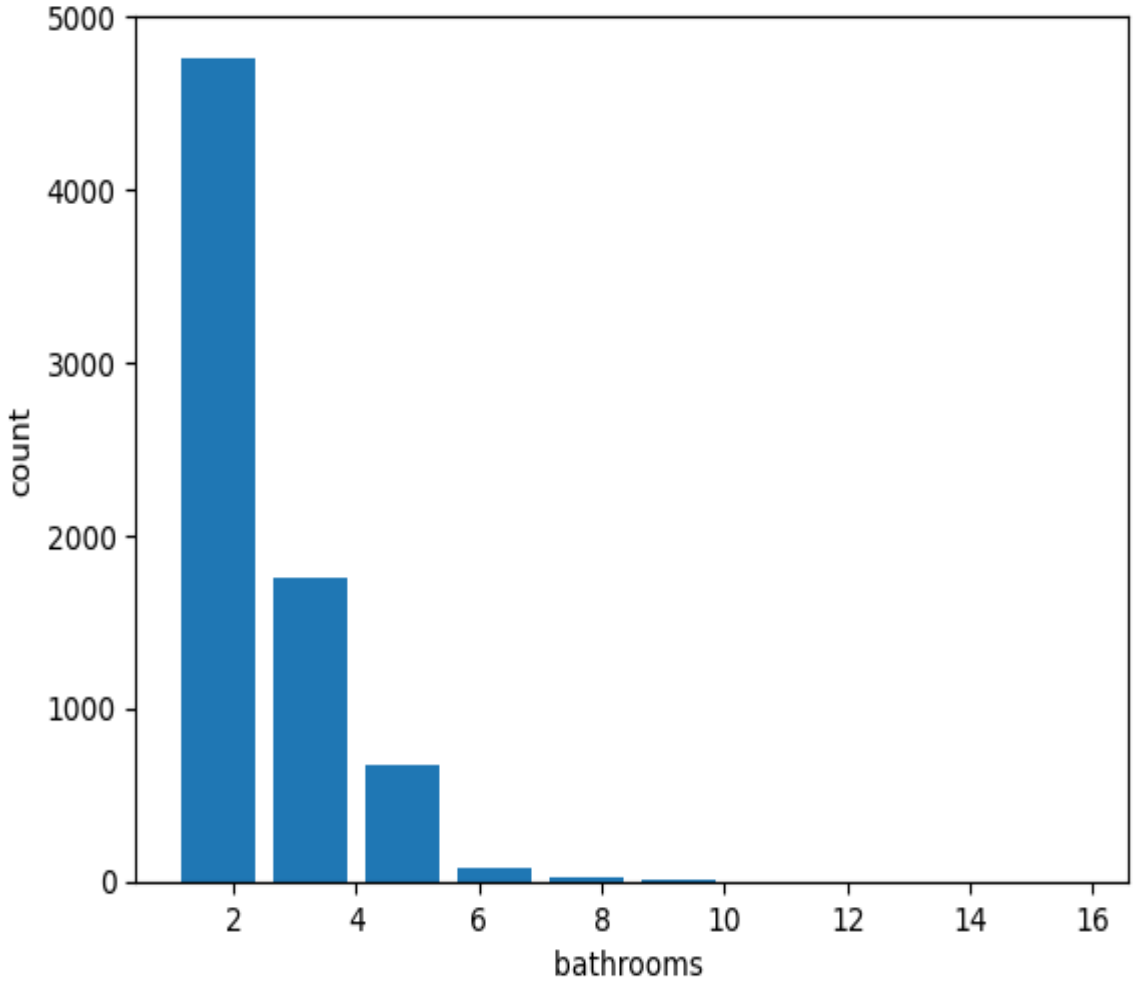
→The graph is a histogram It shows the distribution of prices for a set of properti



The vast majority of properties fall in the $4,000 to $8,000 per square foot range (the two tallest bars).The distribution is highly skewed to the right (positive skew), meaning there are many more lower-priced properties than very high-priced properties, with very few properties exceeding $15,000 per square foot.

BAR CHART(4.4)

→ The graph is a bar chart showing the distribution of the number of bathrooms in a dataset of properties.



EXPLAINATION

- The vast majority of properties have 2 bathrooms (nearly 5,000 counts).
- The next most common is 3 bathrooms (around 1,800 counts).
- The count rapidly decreases as the number of bathrooms increases, with very few properties having 6 or more bathrooms.
- The distribution is heavily skewed to the left (meaning most values are concentrated on the lower end).