PHASE 4

HOUSE PRICE PREDICTION
USING MACHINE LEARNING

PHASE 4

HOUSE PRICE PREDICTION USING MACHINE LEARNING

Introduction:

●Housepricepredictioncanhelpthedeveloperdeterminethesellingprice ofahouseand.canhelpthecus
To merto arrange the right time to purchase a house.
●Therearethreefactorsthatinfluencethepriceofahousewhichincludeph ysicalconditions,conceptand
location.
● Aproperty'svalueisimportantinrealestatetransactions.
●Housingpricetrendsarenotonlytheconcernofbuyersandsellers,butital soindicatesthecurrentecono
Mic situation.
●Therefore,itisimportanttopredicthousingpriceswithoutbiastohelpboth thebuyersand
Sellers make their decisions.
● This project development may help to dict the house price.

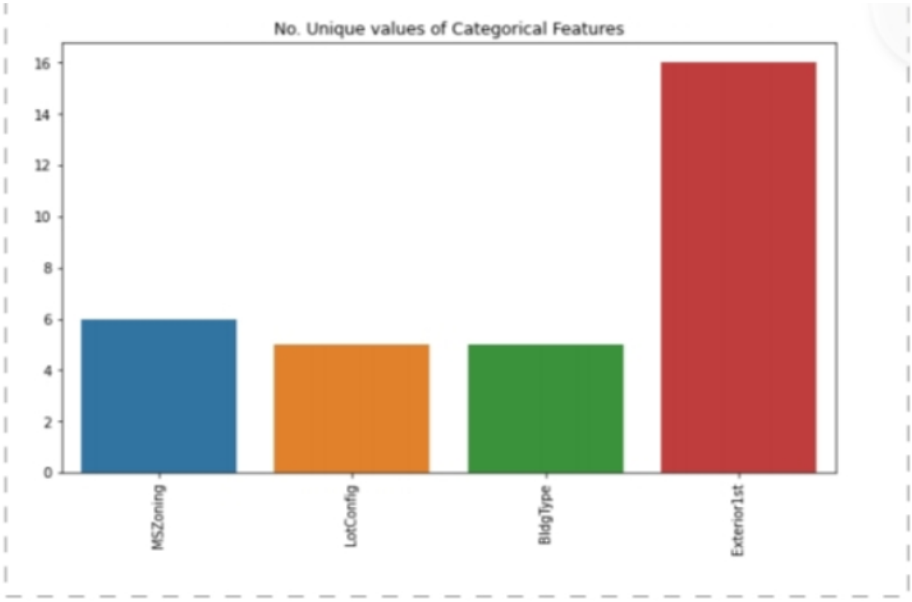
Code :

python
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Load your dataset (assuming it's in a CSV file)
# Replace 'dataset.csv' with your actual dataset file
data = pd.read_csv('dataset.csv')

Assume 'features' contains the columns you want to

```
use for prediction
features = [&#39;bedrooms&#39;, &#39;bathrooms&#39;,
&#39;sqft_living&#39;, &#39;sqft_lot&#39;, &#39;floors&#39;]
X = data[features]
y = data[&#39;price&#39;]

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

# Create a linear regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
predictions = model.predict(X_test)

# Calculate and print metrics
mse = mean_squared_error(y_test, predictions)
r2 = r2_score(y_test, predictions)
print(f&#39;Mean Squared Error: {mse}&#39;)
print(f&#39;R-squared: {r2}&#39;)
# Now you can use the trained model to make predictions for new
data
# For example:
# new_data = pd.DataFrame([[3, 2, 2000, 5000, 2
```



Regression analysis:

Problem Statement:

Consider a real estate company that has a dataset containing the prices of properties in the Delhi region. It wishes to use the data to optimise the sale prices of the properties based on important factors such as area, bedrooms, parking, etc.

Essentially, the company wants —

- **To identify the variables affecting house prices, e.g. area, number of rooms, bathrooms, etc.**
- **To create a linear model that quantitatively relates house prices with variables such as number of rooms, area, number of bathrooms, etc.**
- **To know the accuracy of the model, i.e. how well these variables can predict house prices.**

Reading and Understanding the Dat

```
# Import the numpy and pandas package

import numpy as np
import pandas as pd

Data Visualisation

import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
housing = pd.DataFrame(pd.read_csv("../input/Housing.csv"))
```

In [3]:

```
housing.shape
```

Out[3]: (545, 13)

In [4]:

```
housing.info()
```

```
<class 'pandas.core.frame.DataFrame'>

RangeIndex: 545 entries, 0 to 544

Data columns (total 13 columns):

price               545 non-null int64

area                545 non-null int64

bedrooms            545 non-null int64

bathrooms           545 non-null int64

stories             545 non-null int64

mainroad            545 non-null object
```

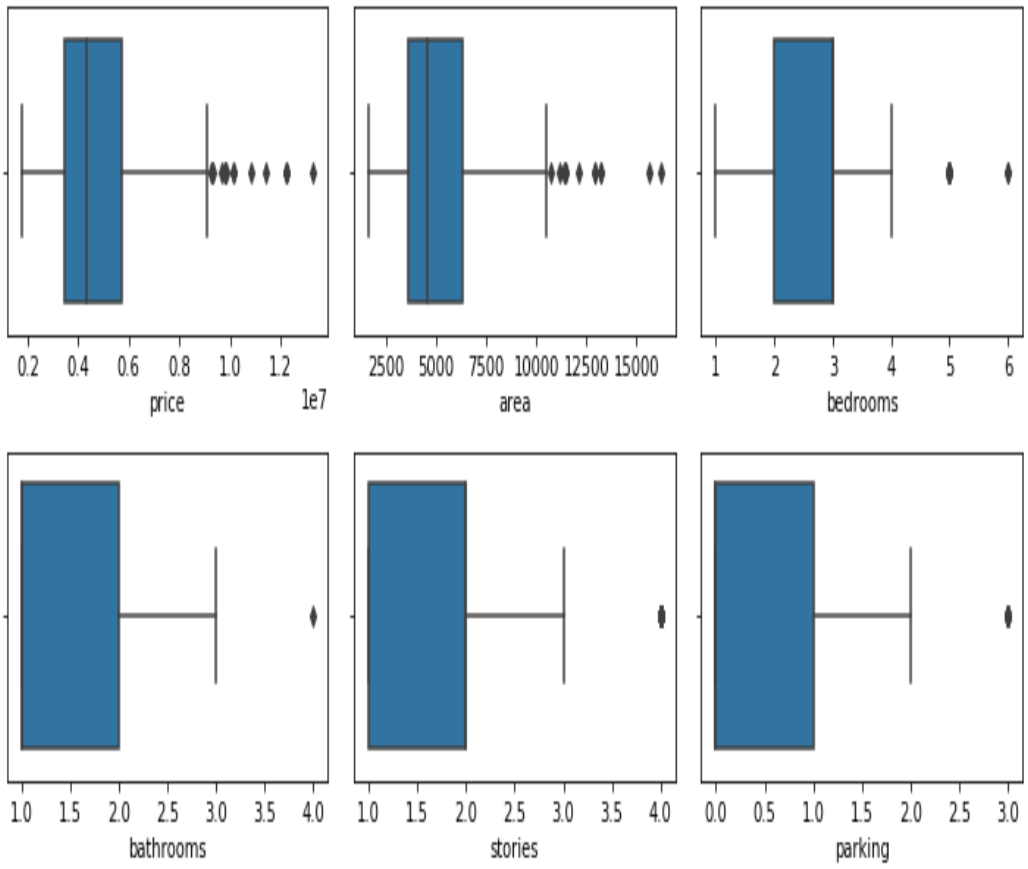
```
guestroom           545 non-null object
basement            545 non-null object
hotwaterheating     545 non-null object
airconditioning     545 non-null object
parking             545 non-null int64
prefarea            545 non-null object
furnishingstatus    545 non-null object
dtypes: int64(6), object(7)
memory usage: 55.4+ KB
```

```
housing.head()
```

Outlier Analysis

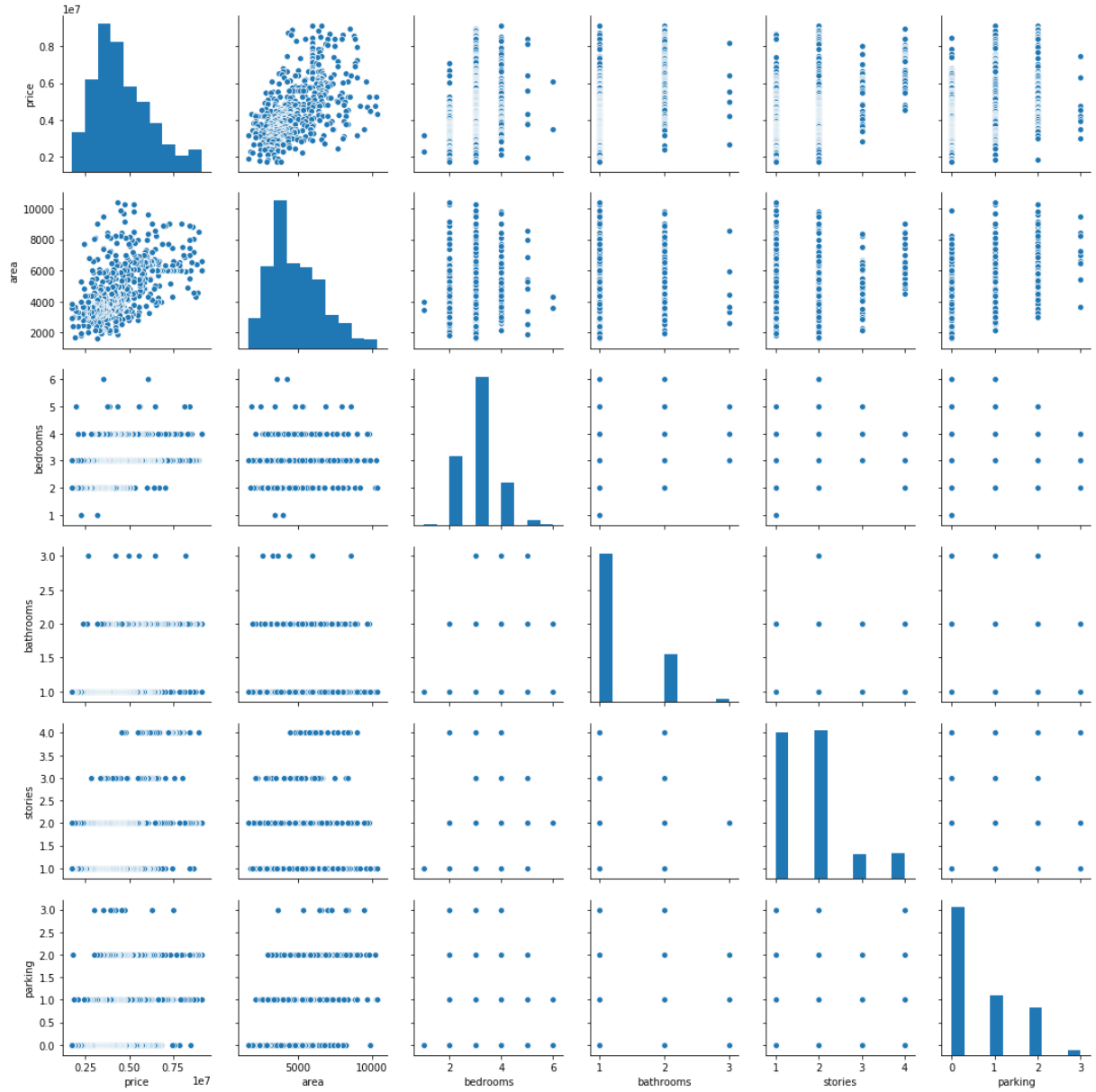
```
fig, axs = plt.subplots(2,3, figsize = (10,5))
plt1 = sns.boxplot(housing['price'], ax = axs[0,0])
plt2 = sns.boxplot(housing['area'], ax = axs[0,1])
plt3 = sns.boxplot(housing['bedrooms'], ax = axs[0,2])
plt1 = sns.boxplot(housing['bathrooms'], ax = axs[1,0])
plt2 = sns.boxplot(housing['stories'], ax = axs[1,1])
```

```
plt3 = sns.boxplot(housing['parking'], ax = axs[1,2])


plt.tight_layout()
```



Exploratory data analytics

```
sns.pairplot(housing)
plt.show()
```



Linear Regression:

• Import the required libraries and modules, including pandas for data manipulation, scikitlearn for machine learning algorithms, and LinearRegression for the linear regression model.

• Loading the required dataset with pd.read_csv and select the features we want to use for prediction (e.g., bedrooms, bathrooms, sqft_living, sqft_lot, floors, and zip code), as well as the target variable (price).
unction fit() with the training data.

• Once the model is trained, we make predictions for the test data set using predict and store the results in y_pred. To evaluate the performance of the model, we calculate the R^2 score using the score for the test set.

• Demonstrate how to predict the price of a new house by creating a new dataframe new_house with the features of the house. We pass this dataframe to the model's prediction function to obtain the predicted price.

We will use this section of the report to explain some of the data preparation and cleaning processes that we followed to have standardized data. We start off with showing the top few rows of the data for unders

Conclusion:

In conclusion using machine learning in Python is a powerful tool for predicting house prices. By gathering and cleaning data, visualizing patterns, and training and evaluating our models, we can make informed decisions in the dynamic world of real estate.
By leveraging advanced algorithms and data analysis, we can make accurate predictions and inform decision-making processes. This approach empowers buyers, sellers, and investors to make informed choices in a dynamic and competitive market, ultimate

THANK YOU!!..