

```
In [1]: #real estate price prediction
      ### created by : siva kumar
```

```
In [2]: #import pandas Library helps us work with table like data
import pandas as pd#from now onwards i will call pandas as pd in my code its Lik

data = {
    'Location': ['palasa','vizag' , 'srikakulam'],
    'Bedrooms': [3, 2, 4],
    'Area_sqft': [1200, 900, 1500],
    'Age_Years': [5, 3, 10],
    'Amenities_Score': [8, 7, 9],
    'Price_Lakhs': [75, 60, 85]
}

df = pd.DataFrame(data)

df

X = df[['Bedrooms', 'Area_sqft', 'Age_Years', 'Amenities_Score']]
y = df['Price_Lakhs']

from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X, y)

predicted_prices = model.predict(X)

comparison_df = pd.DataFrame({
    'Location': df['Location'],
    'Actual Price (Lakhs)': y,
    'Predicted Price (Lakhs)': predicted_prices
})

print(comparison_df)
#creating some date examples about houses
# -----
#   Creating a Data Dictionary
# -----

# WHAT ARE WE DOING?
# We are storing details of houses in Python using a DICTIONARY.
# A dictionary is a type of variable that stores data in the format:
#
#     KEY : VALUE
#
# Each KEY is like a column name (e.g., 'Location')
# Each VALUE is a list of data (e.g., ['palasa,vizag,srikakulam'])

# SYNTAX RULES for Dictionary:
# 1. Use curly brackets { }
# 2. Write keys in quotes: "KeyName"
# 3. Put a colon : after each key
# 4. Add a list of values [ ] after the colon
# 5. Separate each pair with commas
```

```

# Example:
#     data = {
#         'Key1': [value1, value2],
#         'Key2': [value3, value4]
#     }

#data={ 'location':[palasa,vizag,srikakulam],#city names
#       'bedrooms':['3','2','4'],#no.of bedroom
#       'area_sqft':[1200,900,1500],#size of the house
#       'age_years':[5,3,10],#how old the house is?
#       'rating_score':[8,7,9]#rating according to facilities they provide
#       'price_lakhs':[75,60,85]#price of house in lakhs }
#we created a variable called data that stores value like which holds 3 rows of
#it like small table that stored in memory

#Turn this data into a proper table using pandas
#the dot(.)means : hey python inside the big library(pd) i want to use some spec
#in this case from the pd library ,use the dataframe function
#imagine pd is a tool box ,and dataframe is like hammer in toolbox (pd) to take
#pd is shortcut name for pandas
#dataframe() is a function from pandas
#data is the dictionary we created
#df is the variable name
# 'pd' is our nickname for pandas, and 'DataFrame' is a function to create a table
# This new table will be saved in a variable called 'df'
# Think of 'df' like a mini Excel sheet you can use in your code
#-----
#yes x is a variable even though x is just variable name but in machine learning
#x and y are variable and its names
#x is input features ---meaning ---->clues or data to make prediction
#y is output/target----meaning----->the value we want to predict like example
# why df came before the square bracket we created a data frame and named it as
#when we write df['bedrooms']--->meaning : hey python look inside the table (df
#if we want more than one column like bedroom ,area,years,etc.. same like df[['b
#df is where to look our main data table
#[ ] tells what to pick which column
#x=df[['bedrooms','area_sqft','rating_score']]
#y means the answer what we want the computer to learn to predict like this is t
#y=['price_lakhs']
#created the table df and selected the input x and output y.
#-----
#from =>we are taking something from a library/toolbox
#sklearn=>a popular tool box in python used for machine learning means like a sc
#linear_model=>its like small section(drawer) inside the sklearn that has tools
#import=>bring this item into my program so i can use it
#linearregression=>a tool that helps us guess the numbers based on the past exa
#model=> this is a name we are giving to our robot brain
#linear regression() => build a new brain using linear regression tool like make
#what is linear regression() the brackets .these brackets creates new object lik
#why do we use () => because we are calling or creating something just like in py
#print is the function
#() is how you use it
#same here linear regression() means: make a new model

#What's going on here?

# We are writing: model = LinearRegression()
# This means: "Make a new brain (model) to learn and predict house prices."

```

```

# Why are there empty brackets () after LinearRegression?

# In Python, brackets () are used when calling:
# - a function like print() or input()
# - or creating an object from a class like LinearRegression()

# Sometimes, we put something inside the brackets (example: print("Hello"))
# Because print needs something to display

# But here, LinearRegression() doesn't need any special instructions.
# So we leave the brackets empty – it will use default settings.
# It's like asking Python: "Hey, give me a basic model, I'll customize later if
#-----
#model=fit(x,y)
# model = this is the computer's brain we created using linear regression()
#fit() = this means learn or study
#x=clues,y=answers
# telling like hey computer please learn these examples x and their answers y
#-----
#predicted_prices=model.predict(x)
# Ask the model to guess (predict) prices

# predicted_prices = model.predict(X)

# What is 'predict'?
# 'predict' is a function that uses the model's brain (trained earlier) to guess

# What are we predicting?
# We're predicting house prices using the same features (X).
# The model already learned the relationship between house features and price us

# So now we ask:
# "Hey model, based on what you learned, what should be the price for each house

# The result (predicted_prices) will be a list of guessed prices like:
# [74.8, 60.1, 85.3] – close to actual prices [75, 60, 85]

# These predictions help us understand if the model learned correctly.
#-----

#comparison_df = pd.DataFrame({
#    'Location': df['Location'], # city names
#    'Actual Price (Lakhs)': y, # real prices of houses
#    'Predicted Price (Lakhs)': predicted_prices # prices predicted by machine
#})

# What are we doing here?
# We're asking the machine (our trained model) to predict or "guess" the house p
# based on the details we gave earlier like Bedrooms, Area, Age, Amenities.

# model.predict(X)
# 'model' is like a student who learned from examples in Step 9 using model.fit(
# Now we say, "Okay student, based on what you learned, tell me what the price s
# X contains the input data (like Bedrooms, Area, Age, etc.) – no prices, only f
# This line does the guessing part – the brainwork!

# predicted_prices
# This is a new variable where we're storing the machine's guesses.
# Example: If there are 3 houses, it will return 3 price guesses like [74.2, 61.

```

```
# Note: These are not exact values from before – they are based on Learning patt  
  
# print(predicted_prices)  
# This line shows the guessed prices on the screen.  
# It helps us see what the machine predicted so we can check if it's close to th  
  
# We are NOT giving the price.  
# We ARE letting the machine learn from old data and guess for new data.  
# That's what Machine Learning really means – Learning and predicting!  
  
# -----
```

	Location	Actual Price (Lakhs)	Predicted Price (Lakhs)
0	palasa	75	75.0
1	vizag	60	60.0
2	srikakulam	85	85.0

In []:

In []: