Objective: Build a simple Linear Regression model to predict a numerical value based on a single feature.

1. Install the necessary libraries:

```
pip install numpy pandas scikit-learn
```

```
Requirement already satisfied: numpy in /usr/local/lib/python3.10/dist-packages (1.26.4)
Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.1.4)
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.10/dist-packages (1.3.2)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.2)
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)
Requirement already satisfied: scipy>=1.5.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.13.1)
Requirement already satisfied: joblib>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (1.4.2)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn) (3.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)
```

2. Import the required libraries:

```
import numpy as np
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
```

3. Load your dataset into a Pandas DataFrame:

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
```

```
df = pd.read_csv("/content/drive/MyDrive/DataSets/Customer Purchasing Behaviors.csv")
df.head()
```

	user_id	age	annual_income	purchase_amount	loyalty_score	region	purchase_frequency
0	1	25	45000	200	4.5	North	12
1	2	34	55000	350	7.0	South	18
2	3	45	65000	500	8.0	West	22
3	4	22	30000	150	3.0	East	10
4	5	29	47000	220	4.8	North	13

Next steps: View recommended plots New interactive sheet

```
#from google.colab import sheets
#sheet = sheets.InteractiveSheet(df=df)
```

4. Split the data into features (X) and target variable (y):

```
# Define the feature (Floor_area) and target (Price_in_taka)
X = df[['annual_income']]  # Feature
y = df['purchase_amount']  # Target

# Check the feature and target data
print(X.head())
print(y.head())
```

```
   annual_income
0          45000
1          55000
2          65000
3          30000
4          47000
0    200
1    350
2    500
3    150
4    220
Name: purchase_amount, dtype: int64
```

```
X = df['annual_income'].values.reshape(-1, 1)  # Reshape for single feature input
y = df['purchase_amount'].values
```

5. 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)
```

6. Create an instance of the Linear Regression model and fit it on the training data:

```
model = LinearRegression()
model.fit(X_train, y_train)
```

```
▾ LinearRegression
LinearRegression()
```

7. Make predictions on the test set:

```
predictions = model.predict(X_test)
```

8. Evaluate the model using metrics like Mean Squared Error and R-squared score:

```
mse = mean_squared_error(y_test,predictions)
r2 = r2_score(y_test,predictions)

print(f"Mean Squared Error: {mse}")
print(f"R-squared Score: {r2}")
```

```
Mean Squared Error: 581.4415125533451
R-squared Score: 0.9639912361054181
```

This is just a basic outline to get you started with building a simple Linear Regression model in Python using scikit-learn library.