Task 3: Linear Regression

✅ Task 3: Linear Regression — Explanation

This task is about learning and implementing **Linear Regression**, which is one of the most important supervised machine learning algorithms.

🎯 Objective

To understand and implement:

✓ Simple Linear Regression

– Predicting a target variable using **one** input variable.
Example: Predicting house price using only square feet.

✓ Multiple Linear Regression

– Predicting the target using **two or more** input variables.
Example: Predicting house price using square feet, number of bedrooms, location score, etc.

You will learn **how the model works, how to train it, and how to evaluate it**.

🧰 Tools Used

The task requires using the following Python libraries:

1. Scikit-learn

For building the regression model (

sklearn.linear_model.LinearRegression
).

2. Pandas

For loading, manipulating, and preprocessing the dataset.

3. Matplotlib

For visualizing the regression line, predictions, and error metrics.

🔍 Hints / Mini Guide — Explained

1 Import and preprocess the dataset

Load the dataset using Pandas.
Check for missing values.
Clean/transform data (optional).

Select input (X) and output (y) variables.
Example:

```
df = pd.read_csv("data.csv")
X = df[['feature1']]
y = df['target']
```

2 Split data into train-test sets

Training set → Used to fit the model

Testing set → Used to check performance
Using scikit-learn:

```
from sklearn.model_selection
import train_test_split


X_train, X_test, y_train,
y_test = train_test_split(X,
y, test_size=0.2)
```

3 Fit the Linear Regression model

Use:

```
from sklearn.linear_model
import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

```
This step teaches the model
the relationship between
features and target.
```

4 Evaluate the model using MAE, MSE, R^2

Common evaluation metrics:

MAE (Mean Absolute Error) → Average error

MSE (Mean Squared Error) → Squared error (penalizes big mistakes)

R^2 Score → How well the model fits the data (0–1)

Example:

```
from sklearn.metrics import
mean_absolute_error,
mean_squared_error, r2_score


pred = model.predict(X_test)


mae =
mean_absolute_error(y_test,
pred)
mse =
mean_squared_error(y_test,
pred)
r2 = r2_score(y_test, pred)
```

5 Plot regression line and interpret coefficients

For simple linear regression:

Plot original data points

Add the regression line using Matplotlib

Example:

```
plt.scatter(X_test, y_test)
plt.plot(X_test, pred)
```

Interpret model coefficients

model.coef_

→ The slope (effect of X on y)

model.intercept_

→ Where the line starts

Example interpretation:

“If

coef_ = 0.8

, then for every 1 unit increase in X, y increases by 0.8."

📌 Summary

This task teaches you:

✓ How to load and clean data

✓ How to split into train/test

✓ How to apply linear regression

✓ How to evaluate using MAE, MSE, R^2

✓ How to visualize the regression line

✓ How to interpret slope & intercept

If you want, I can also provide:

📄 Full Python code

📊 Sample dataset

📈 Graph output

📝 Report/Explanation for submission

Just tell me!