/*Practical No : 03,
Name: Dhiraj Vijay Barwal ,
Roll No : 08,
Class: TECSD,
Batch: T1 */

Prediction of dependent values using linear regression

```
# Import necessary 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

# Sample data (replace this with your dataset)
data = {
    'X': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],  # Independent variable
    'Y': [3, 6, 7, 8, 11, 14, 15, 18, 19, 22]  # Dependent variable
}

# Convert the data into a pandas DataFrame
df = pd.DataFrame(data)

# Define independent and dependent variables
X = df[['X']]  # Independent variable
Y = df['Y']    # Dependent variable

# Split the data into training and testing sets (80% train, 20% test)
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size=0.2, random_state=42)

# Initialize the Linear Regression model
model = LinearRegression()

# Train the model with the training data
model.fit(X_train, Y_train)

# Predict the dependent variable (Y) for the test data
Y_pred = model.predict(X_test)

# Print the predicted values
print("Predicted values: ", Y_pred)

# Evaluate the model's performance
mse = mean_squared_error(Y_test, Y_pred)
r2 = r2_score(Y_test, Y_pred)

print("Mean Squared Error (MSE):", mse)
print("R-squared (R2) score:", r2)
# Optionally, show the coefficients of the linear regression model
print("Slope (Coefficient) of the line:", model.coef_)
print("Intercept (Y-intercept) of the line:", model.intercept_)
```

OUTPUT:

```
Predicted values:  [ 7.16666667 16.33333333]
Mean Squared Error (MSE): 1.4375000000000004
R-squared (R2) score: 0.9836734693877551
Slope (Coefficient) of the line: [2.05]
Intercept (Y-intercept) of the line: 0.7499999999999991
```