Indian Institute of Information Technology Surat

Lab Report on

Machine Learning (CS 601) Practical

Submitted by

[RAHUL KUMAR SINGH] (UI21CS44)

Course Faculty

Dr. Pradeep Kumar Roy

Dr. Rajesh K. Ahir

Department of Computer Science and Engineering
Indian Institute of Information Technology Surat
Gujarat-394190, India

Jan-2024

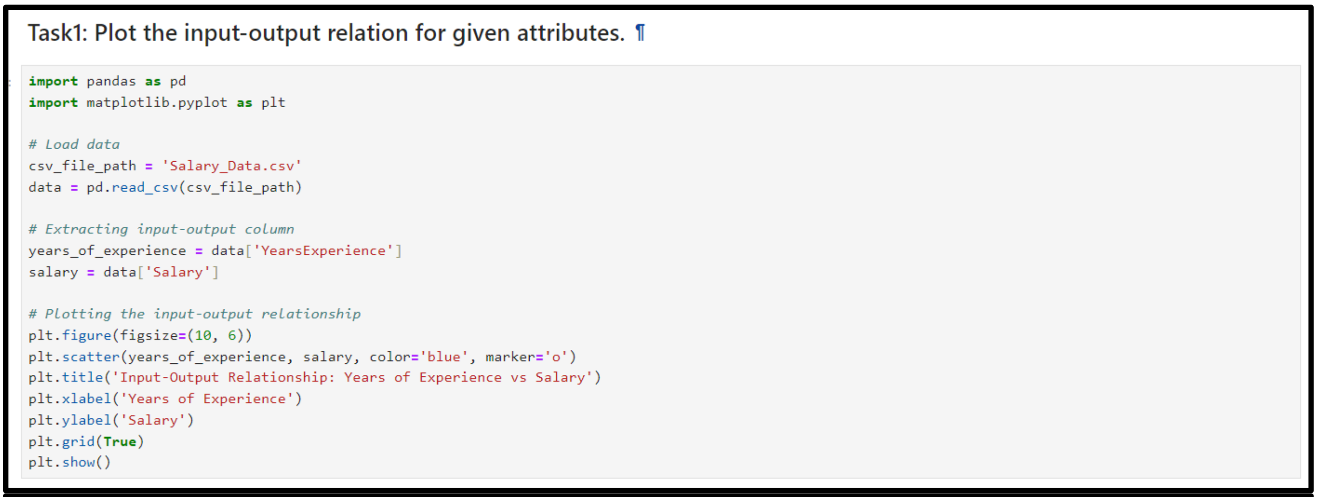
Lab No: 3

Aim: To understand and implement linear regression

Description: Perform the following task with using inbuilt Python Libraries.:

- Plot the input-output relation for given attributes.
- Design a mathematical function to find the best-fitted line for the given data (attached here).
- Plot Error vs. Slope graph and show the global minima for the sample data X={2, 4, 6, 8} and Y={3, 7, 5, 10} considering different learning rate values (alpha).

Source Code:



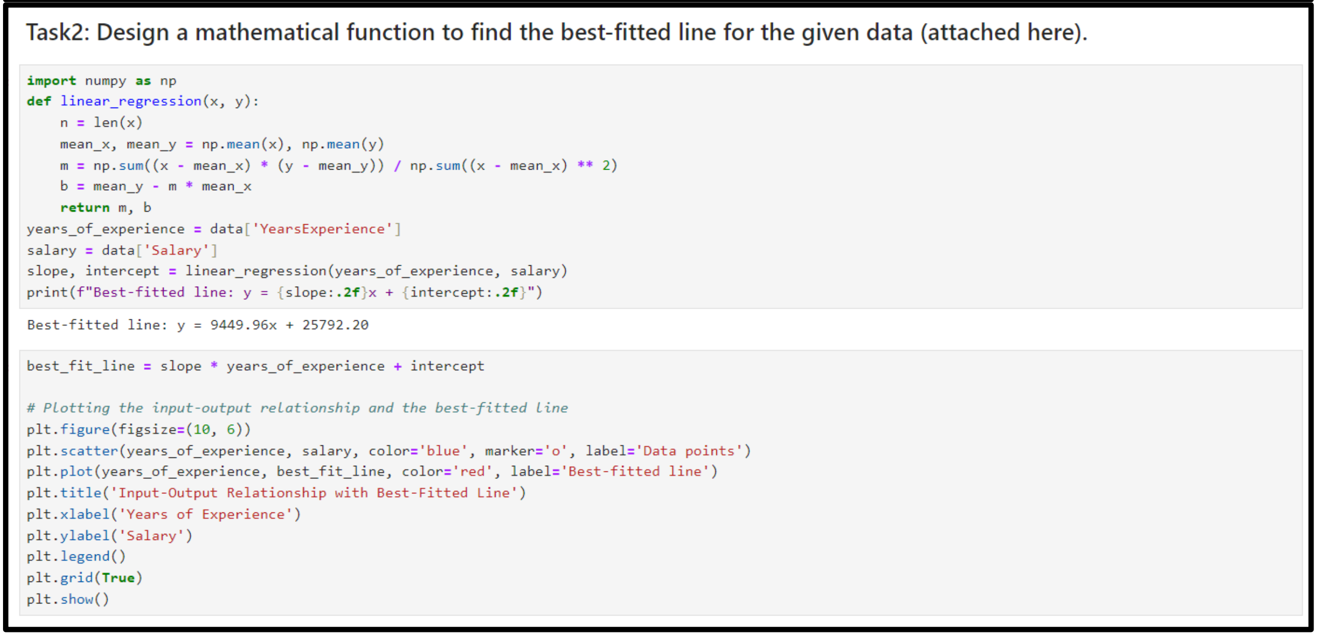
Task1: Plot the input-output relation for given attributes. ¶

```python
import pandas as pd
import matplotlib.pyplot as plt

# Load data
csv_file_path = 'Salary_Data.csv'
data = pd.read_csv(csv_file_path)

# Extracting input-output column
years_of_experience = data['YearsExperience']
salary = data['Salary']

# Plotting the input-output relationship
plt.figure(figsize=(10, 6))
plt.scatter(years_of_experience, salary, color='blue', marker='o')
plt.title('Input-Output Relationship: Years of Experience vs Salary')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.grid(True)
plt.show()
```



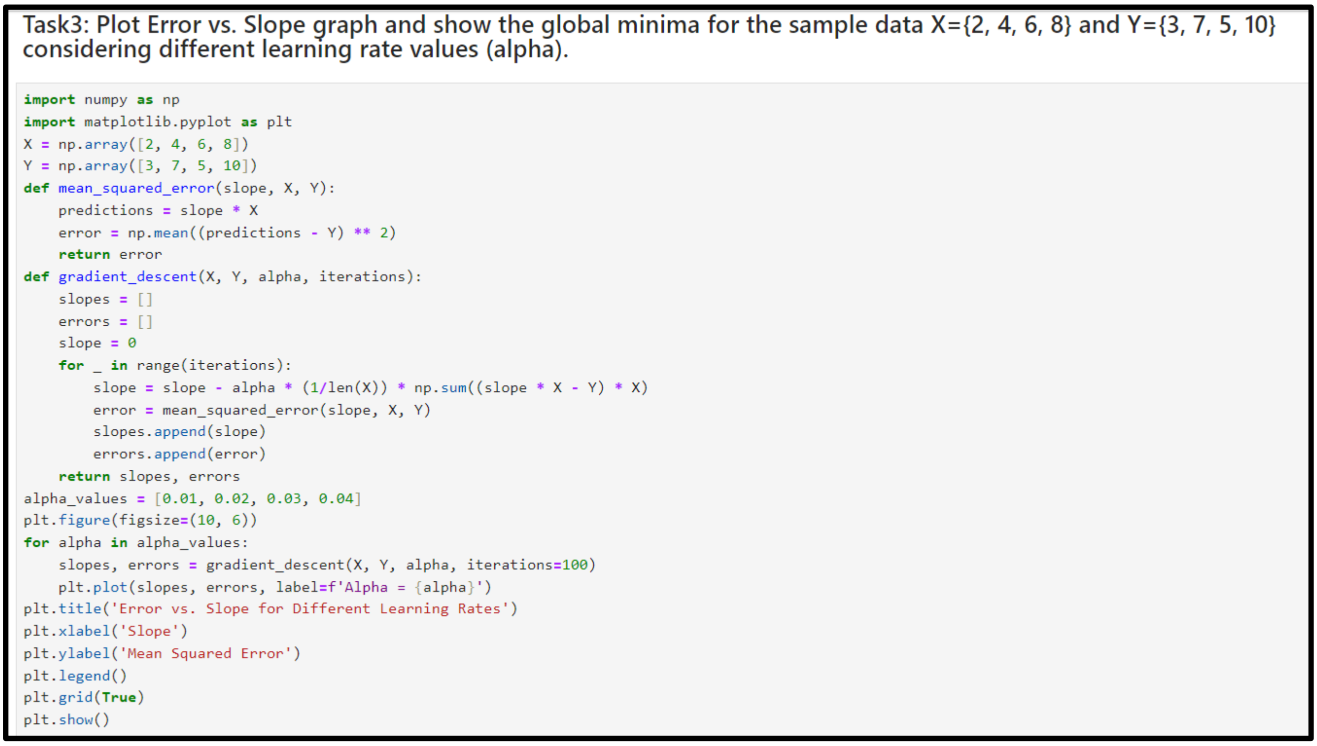
Task2: Design a mathematical function to find the best-fitted line for the given data (attached here).

```python
import numpy as np
def linear_regression(x, y):
    n = len(x)
    mean_x, mean_y = np.mean(x), np.mean(y)
    m = np.sum((x - mean_x) * (y - mean_y)) / np.sum((x - mean_x) ** 2)
    b = mean_y - m * mean_x
    return m, b
years_of_experience = data['YearsExperience']
salary = data['Salary']
slope, intercept = linear_regression(years_of_experience, salary)
print(f"Best-fitted line: y = {slope:.2f}x + {intercept:.2f}")
```

```
Best-fitted line: y = 9449.96x + 25792.20
```

```python
best_fit_line = slope * years_of_experience + intercept

# Plotting the input-output relationship and the best-fitted line
plt.figure(figsize=(10, 6))
plt.scatter(years_of_experience, salary, color='blue', marker='o', label='Data points')
plt.plot(years_of_experience, best_fit_line, color='red', label='Best-fitted line')
plt.title('Input-Output Relationship with Best-Fitted Line')
plt.xlabel('Years of Experience')
plt.ylabel('Salary')
plt.legend()
plt.grid(True)
plt.show()
```

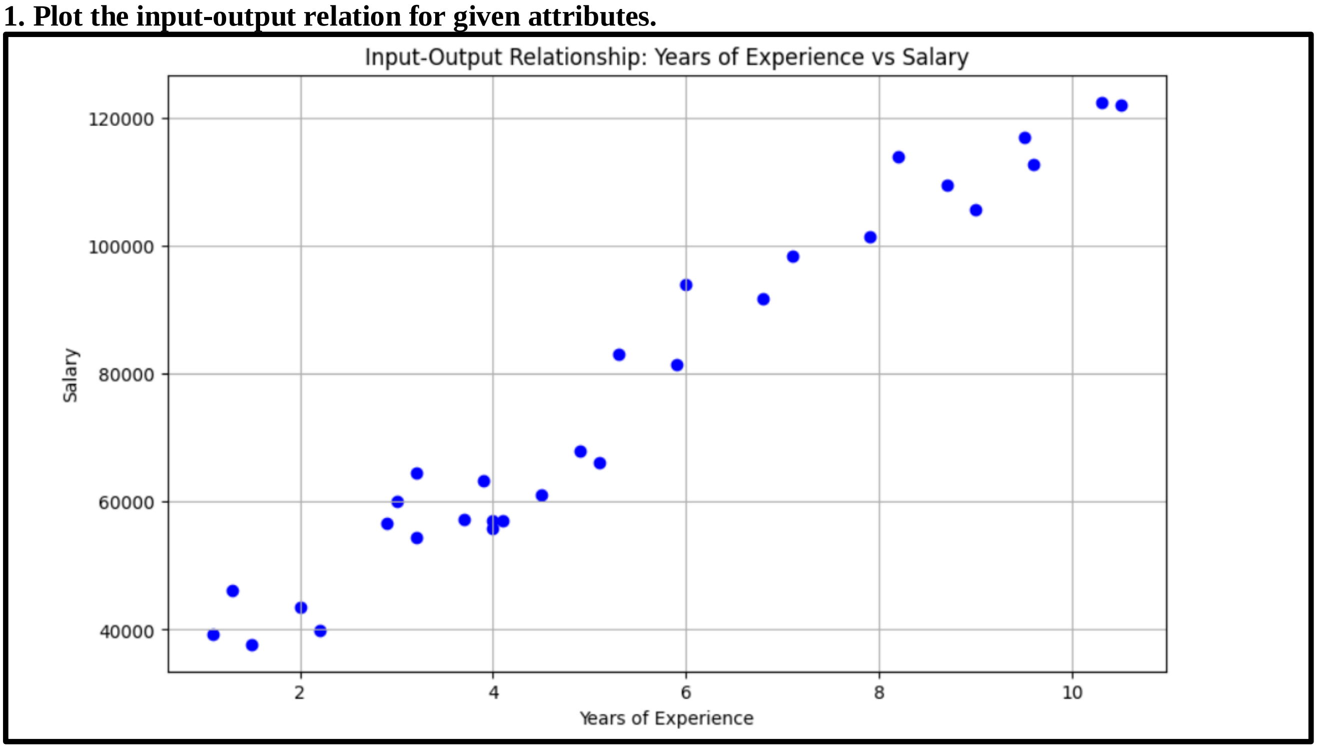


Task3: Plot Error vs. Slope graph and show the global minima for the sample data X={2, 4, 6, 8} and Y={3, 7, 5, 10} considering different learning rate values (alpha).

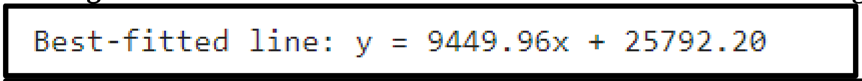
```
import numpy as np
import matplotlib.pyplot as plt
X = np.array([2, 4, 6, 8])
Y = np.array([3, 7, 5, 10])
def mean_squared_error(slope, X, Y):
    predictions = slope * X
    error = np.mean((predictions - Y) ** 2)
    return error
def gradient_descent(X, Y, alpha, iterations):
    slopes = []
    errors = []
    slope = 0
    for _ in range(iterations):
        slope = slope - alpha * (1/len(X)) * np.sum((slope * X - Y) * X)
        error = mean_squared_error(slope, X, Y)
        slopes.append(slope)
        errors.append(error)
    return slopes, errors
alpha_values = [0.01, 0.02, 0.03, 0.04]
plt.figure(figsize=(10, 6))
for alpha in alpha_values:
    slopes, errors = gradient_descent(X, Y, alpha, iterations=100)
    plt.plot(slopes, errors, label=f'Alpha = {alpha}')
plt.title('Error vs. Slope for Different Learning Rates')
plt.xlabel('Slope')
plt.ylabel('Mean Squared Error')
plt.legend()
plt.grid(True)
plt.show()
```

Output:

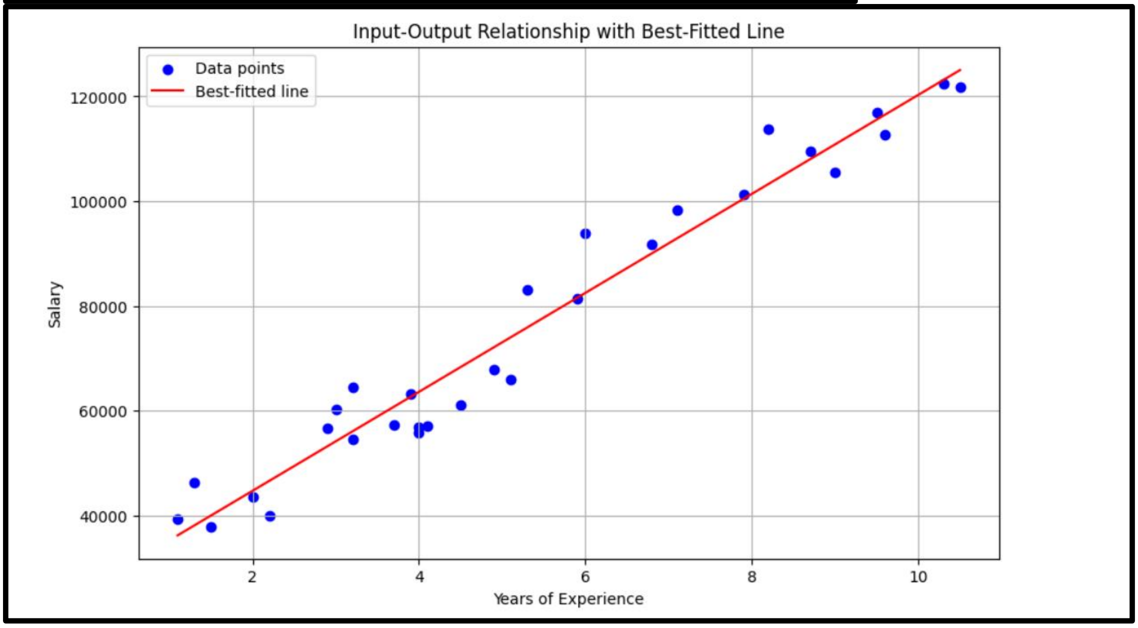
1. Plot the input-output relation for given attributes.



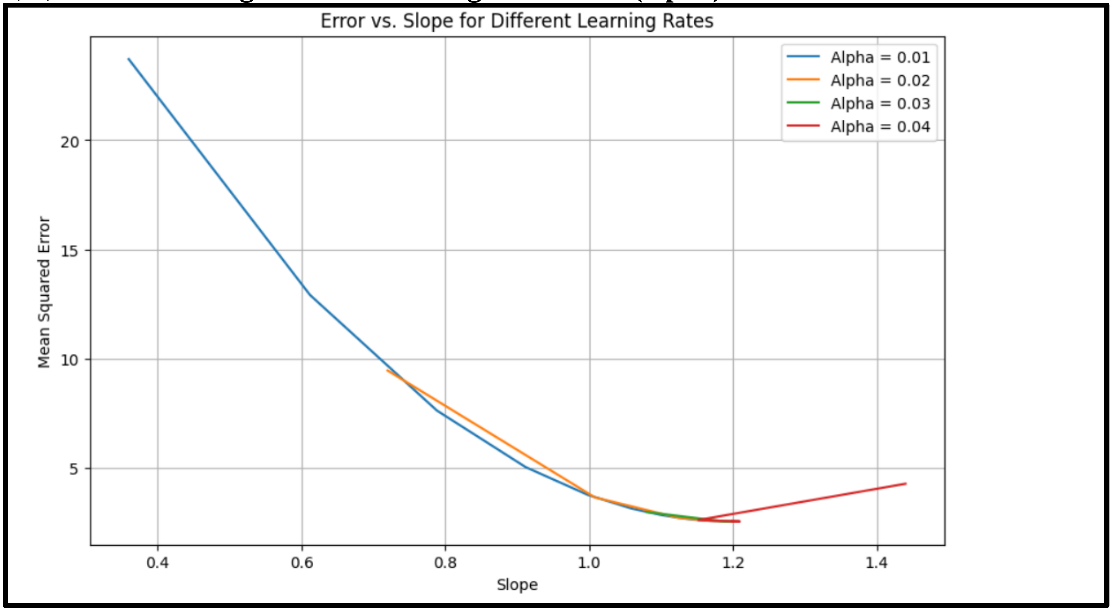
2. Design a mathematical function to find the best-fitted line for the given data



```
Best-fitted line: y = 9449.96x + 25792.20
```



3. Plot Error vs. Slope graph and show the global minima for the sample data X={2, 4, 6, 8} and Y={3, 7, 5, 10} considering different learning rate values (alpha).



Conclusion:

- A custom linear regression function was developed to find the best-fitted line for the given data.
- Utilized the gradient descent algorithm to minimize the cost function, aiming to find the optimal slope for the given linear regression problem.
- Plotted the Error vs. Slope graph for each learning rate, illustrating the convergence behavior over epochs.
- The impact of the learning rate on the convergence speed and the final error was observed through the plotted graphs.