ML Assignment 1

- Rohit Arrunachalam 21110433 IOT B

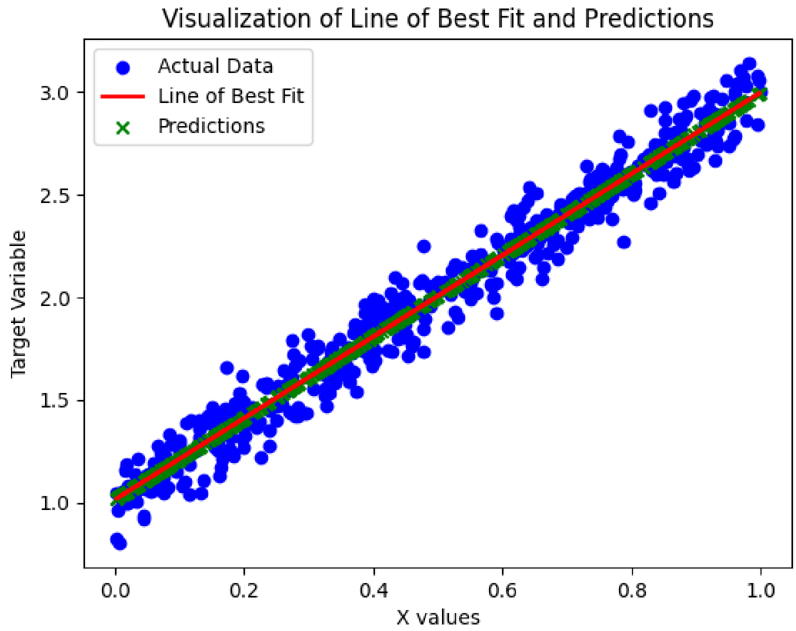
Question 1

Create a random 2-D numpy array with 1500 values. Simulate different lines of fit using 1000 values from the array and find the errors for each of these lines. Find the line with the least error among these lines and store it as the line of best fit. Using this line of best fit, predict the target variable for the other 500 values.

Soln

```
import numpy as np

import matplotlib.pyplot as plt

from sklearn.linear_model import LinearRegression


random_array = np.random.rand(1500)

X_train = random_array[:1000]

y_train = 2 * X_train + 1 + 0.1 * np.random.randn(1000)

X_train = X_train.reshape(-1, 1)

model = LinearRegression()

model.fit(X_train, y_train)

X_test = random_array[1000:]

y_test = 2 * X_test + 1 + 0.1 * np.random.randn(500)

X_test = X_test.reshape(-1, 1)

y_pred = model.predict(X_test)

errors = np.mean((y_test - y_pred) ** 2)

best_model = model

y_final_pred = best_model.predict(X_test)

plt.scatter(X_test, y_test, label='Actual Data', color='blue')

plt.plot(X_test, y_pred, label='Line of Best Fit', color='red',
linewidth=2)

plt.scatter(X_test, y_final_pred, label='Predictions', color='green',
marker='x')

plt.xlabel('X values')
plt.ylabel('Target Variable')
plt.title('Visualization of Line of Best Fit and Predictions')
plt.legend()
plt.show()
print(f"Mean Squared Error: {errors}")
```

Output



```
Mean Squared Error: 0.010186046952211432
```

Question 2

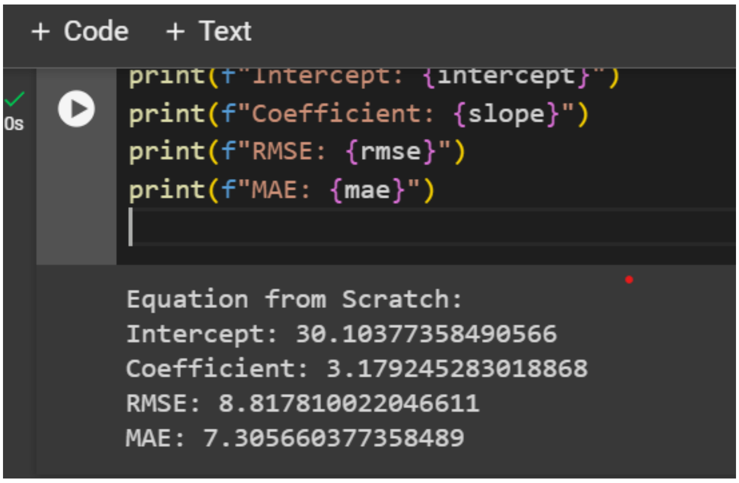
Use the data1.csv to build a simple linear regression from scratch without using sklearn libraries and print the RMSE and mean absolute error values. Use both the equations available in the slides (in theory page) to build the model and compare the intercept and coefficient values.

Soln

```
import pandas as pd
import numpy as np

data = pd.read_csv('/content/data1.csv')
x_values = data['x']
```

```
y_values = data['y']

mean_x = np.mean(x_values)
mean_y = np.mean(y_values)

numerator = np.sum((x_values - mean_x) * (y_values - mean_y))
denominator = np.sum((x_values - mean_x) ** 2)

slope = numerator / denominator
intercept = mean_y - slope * mean_x

y_predicted = slope * x_values + intercept

rmse = np.sqrt(np.mean((y_values - y_predicted) ** 2))
mae = np.mean(np.abs(y_values - y_predicted))

print("Equation from Scratch:")
print(f"Intercept: {intercept}")
print(f"Coefficient: {slope}")
print(f"RMSE: {rmse}")
print(f"MAE: {mae}")
```

Output



```
print(f"Intercept: {intercept}")
print(f"Coefficient: {slope}")
print(f"RMSE: {rmse}")
print(f"MAE: {mae}")
```

```
Equation from Scratch:
Intercept: 30.10377358490566
Coefficient: 3.179245283018868
RMSE: 8.817810022046611
MAE: 7.305660377358489
```