Introduction to Data Science **Assignment 4**

Instructors: **Dr. Bahrak, Dr. Yaghoobzadeh** TA(s): Ali Karimi

Deadline: Tuesday,
Ordibehesht 4th, 11:59 PM

Introduction

In this assignment, you will explore various loss functions and apply gradient descent methods to optimize these functions. Your focus will be on the Diabetes dataset from the scikit-learn library, a well-regarded dataset in the machine learning community. This dataset consists of medical diagnostic measurements from numerous patients and is specifically designed to study diabetes progression. You will use these data points to predict the quantitative measure of disease progression one year after baseline, thus practicing the application of regression analysis in a medical context.

Important Notes

For this assignment, you are required to implement "Part 1: Functions' Implementation" functions from scratch. The use of pre-built libraries such as PyTorch, TensorFlow, or similar for these specific tasks is prohibited.

Dataset Description

The diabetes dataset consists of 442 instances with the following ten baseline variables:

- Age (age in years)
- Sex
- Body Mass Index (BMI)
- Average Blood Pressure (BP)
- Six blood serum measurements:
    - s1: tc, total serum cholesterol
    - s2: ldl, low-density lipoproteins
    - s3: hdl, high-density lipoproteins
    - s4: tch, total cholesterol / HDL
    - s5: ltg, possibly log of serum triglycerides level
    - s6: glu, blood sugar level

The target variable is a quantitative measure of disease progression one year after baseline.

Warm-Up!

- Load the diabetes dataset provided by scikit-learn (scaled=False) or directly via this link.
- Display the first ten rows of the dataset to understand its structure.
- Print the data types of each feature to ensure they are numeric.
- Check for any missing values in the dataset and handle them appropriately (if missing values exist).
- Normalize the features to ensure all are on a similar scale.
- Split the data into training and testing sets using a standard ratio (e.g., 95% training, 5% testing). Utilize sklearn's train_test_split function or an equivalent method ensuring a random split.
- Display the number of instances in both the training and the testing datasets to confirm the split.

Main Task

Part 1: Functions' Implementation

- Implement the following functions from scratch: Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), R^2 Score (Coefficient of Determination)

Part 2: Building and Training the Linear Regression Model

- Construct a regression model and train it using the diabetes dataset

Part 3: Model Evaluation

- Compare the predicted values with the actual progression measures using a scatter plot, where the x-axis represents the actual values, and the y-axis represents the predicted values.
- Evaluate the regression model on the training and testing data using the following functions:
  - MSE
  - MAE
  - RMSE
  - R^2 score

Fill in the table (1) with the calculated metrics.

Data Type/Optimizer	MSE	MAE	RMSE	R^2 score
Train Set				
Test Set				

Table (1) - Result of Linear Regression Model

Part 4: Ordinary Least Squares

- Use OLS from the scipy library, train the model, and finally display the statistics obtained from this process.

Questions

1. Analyze and evaluate the values in Table (1).
2. Review the R^2 and Adjusted R^2 values obtained in part 4. Explain what these values indicate and what the implications of high or low values might be. Also, discuss the differences between these two metrics.
3. Review the p-values obtained in part 4 for each column of data and explain what these values indicate. Discuss what an appropriate value for p-values is and which columns currently have suitable values.
4. Assess and analyze the importance of each feature in the dataset based on the results obtained in part 4 regarding an individual's diabetic condition.

Notes

- Upload your work as a zip file in this format on the website: DS_CA4_[Std number].zip. If the project is done in a group, include all of the group members' student numbers in the name.
- If the project is done in a group, only one member must upload the work.
- We will run your code during the project delivery, so make sure your results are reproducible.