Mid-Semester Lab Test (25 marks)

A medical research team is investigating the effectiveness of using various physiological parameters to distinguish between diabetic (1) and non-diabetic (0) outcomes in patients. The datasets consist of several medical predictor (independent) variables and one target (dependent) variable, *Outcome*. The independent variables consist of '*Glucose*', '*BloodPressure'*, '*BMI'*, '*Insulin'* level, *and 'Age'* (total five variables), alongside one target variable labeled '*Outcome*' – (pl drop/remove other three independent variables). The objective is to develop a linear regression model capable of accurately classifying patients' outcomes into diabetic or non-diabetic categories based on these five independent variables.

Implement a Closed-form based as well as Gradient-descent based Linear Regression from scratch ["**linear_regression_closed**" and "**linear_regression_gradient**"]. [5+5 =10]

**In general, you may use libraries to process and handle data.

****DO NOT** perform feature scaling before feeding the data in your model.

Experiments: [2+2+4+5=13 Marks]

The dataset will be split into Train:Test with 80:20 ratio. Pl shuffle the data before splitting.

1. **Experiment 1**: Load the given dataset into a pandas dataframe. Delete the unwanted columns of independent variables. This altered dataset will serve as the base dataset for the subsequent experiments, let us call this dataset as "dataset_altered". Display the first 10 rows of dataset_altered.

2. **Experiment 2:** Tabulate the correlation coefficients for all the columns (including the target). Plot the corresponding correlation matrix heatmap. Report whether one can find significant correlation between any variables.

3. **Experiment 3:** Design a Closed form Linear Regression Algorithm **(linear_regression_closed).** Tabulate the Percentage Accuracy on training and testing dataset. Plot the confusion matrix.

4. **Experiment 4:** Design a Linear Regression Algorithm based on gradient descent **(linear_regression_gradient).** Tune the hyperparameter for η (learning rate) = [1e-5, 1e-3, 0.05, 0.1]. Train the model for the best η for 50 epochs. Find the optimal learning rate. Tabulate the Percentage Accuracy on training and testing dataset for the optimal learning rate. Plot the confusion matrix.

Submission:

A .zip file containing the python source code and a PDF report file. The final name should follow the template: <Assign-No>_<Your Roll No>.zip. For example, if your roll no is 15CE30021, the filename for LabTest-1 will be: `LabTest-1_15ce30021.zip`

1. A **single python code (.py)** containing the implementations of the models and experiments with comments at function level. The first two lines should **contain your name and roll no**.

2. A report [PDF] containing **[2 Marks]**
    a. Experiment 1: Tabulation of dataset
    b. Experiment 2: Correlation coefficients and your observation.
    c. Accuracy and Confusion Matrix
    d. Optimal learning rate. Accuracy and Confusion matrix (for the optimal learning rate).