Lab Assignment 1 – Logistic Regression

1 Requirements

- Implement **logistic regression** to predict whether a microchip from a factory meets the market standards for sale or not.
- Raw data consists of 3 columns: the first and second columns are features, and the third column is the label.
- Training data is generated by mapping the raw data into a new feature space with 28 dimensions. The function `map_feature` provided in the file `map_feature.py` will perform this mapping.
- Implement the following auxiliary functions to support training and prediction:
  - `compute_cost`: compute the model's cost function on the dataset (the cost function formula is provided in Section 3).
  - `compute_gradient`: compute the gradient vector of the cost function (gradient vector formula is provided in Section 3).
  - `gradient_descent`: implement gradient descent.
  - `predict`: predict whether a set of microchips meet the market standards for sale (to predict for a single microchip, pass an array containing one element).
  - `evaluate`: evaluate the prediction results of the model using metrics: **accuracy, precision, recall, and F1-score** (similar to scikit-learn's `classification_report`, but needs to be implemented manually).
- Next, students should use these auxiliary functions to implement the main program, which includes the following tasks:
  - Read training configuration from the file `config.json`.
  - Train with data provided from the file `training_data.txt`.
  - Save the trained model into the file `model.json`.
  - Predict and evaluate training results on the training dataset, then save the evaluation results into the file `classification_report.json`.

2 Submission Guidelines

- Put all source code and related files into a folder named `<StudentID>`.
- Compress the folder `<StudentID>` and submit it through Moodle.

3 Formulas

- The cost function is calculated using the following formula:

$$J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left[-y^{(i)}\log(h_\theta(x^{(i)})) - (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

- Gradient vector formulas:

$$\frac{\partial J(\theta)}{\partial \theta_0} = \frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)} \quad \text{for } j = 0$$

$$\frac{\partial J(\theta)}{\partial \theta_j} = \left(\frac{1}{m}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})x_j^{(i)}\right) + \frac{\lambda}{m}\theta_j \quad \text{for } j \geq 1$$