2021 Introduction to Machine Learning Program Assignment #4 - Linear Regression & Logistic Regression

TA: Ian Lee ian0908221ian@yahoo.com.tw (mailto:ian0908221ian@yahoo.com.tw)

This programming assignment aims to help you understand ***Linear Regression*** and ***Logistic Regression***.

Before we start

> Join the discord server for TA support (https://discord.gg/XJkvmNrcjp)
>
> - Ask questions on it, and we shall reply.
> - Try not to ask for obvious answers or bug fixes.
> - Memes and chit-chat welcome.

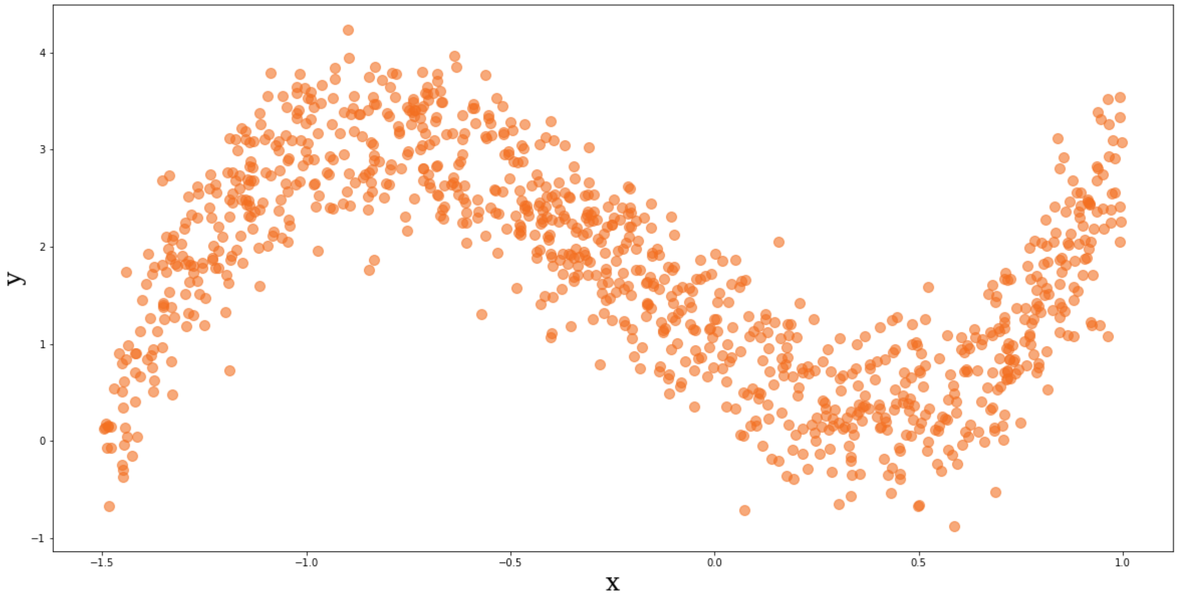
Objective

- Linear Regression - *55% + (10%)*

  1. Data Generation - *15%*
     - Randomly generate 1000 (x_i, y_i) pairs which follow the equation (1)

$$y_i = 3x_i^3 + 2x_i^2 - 3x_i + 1 + \epsilon_i \tag{1}$$

where $-1.5 < x_i < 1.0$, $\epsilon_i \sim N(0, 0.25)$ and N represents Normal distribution



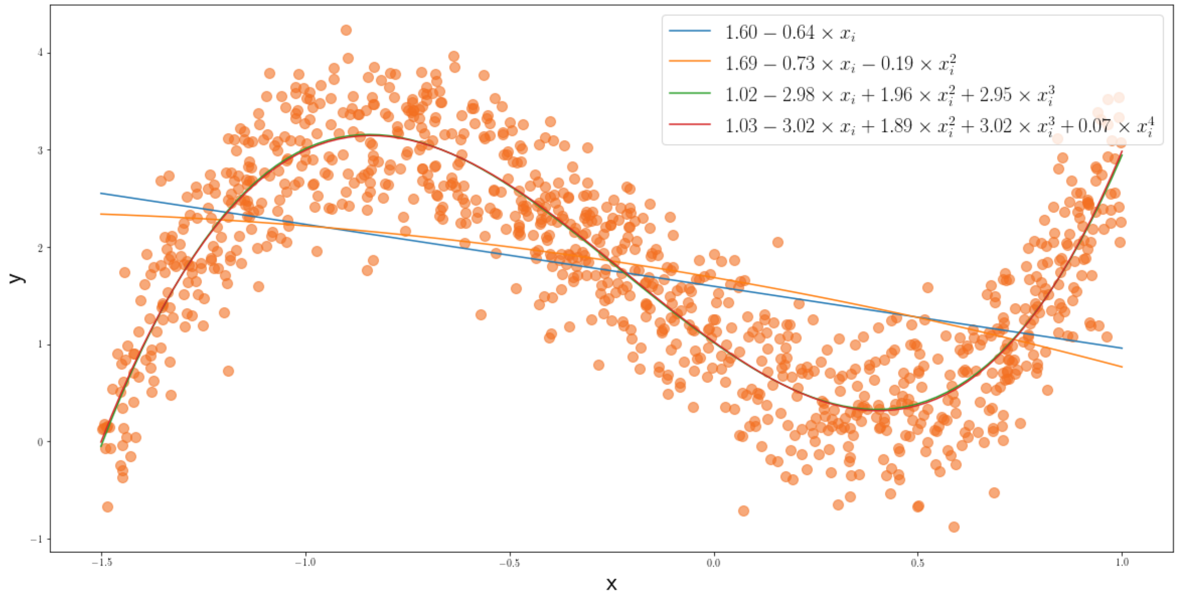
2. Data Preprocessing - *10%*
    - Generate degree-K polynomial features $\hat{x}$ from x

$$\hat{x}_i = \begin{bmatrix} 1 \\ x_i \\ x_i^2 \\ \\ x_i^K \end{bmatrix}$$

    - You must experiments 4 different K settings, $K = 1, 2, 3, 4$
    - hint (https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html)
3. Model Construction - *20%*
    - **Linear Regression**
        - Which makes predictions $\hat{y} = w\hat{x}$, s.t.

$$w = \underset{w'}{\operatorname{argmin}} ||y - w'\hat{x}||^2$$

    - You must construct Linear Regression models to fit and predict data generated by (1)
4. Validation - *0%*
    - Due to the simplicity of Linear Regression, you are not required to implement validation methods.
5. Results - *10%* + *(10%)*
    - Show the fitted weights and the equations
    - Show the predicted $\hat{y}$ for $-1.5 < x < 1.0$
    - Bonus - show the results in a single figure - *(10%)*



        - Legend equations must be written in $\LaTeX$
        - Use $\times$ instead of $*$ to represent multiplication operations
        - Use x_i instead of x
        - Limit the floating-point numeric weights to be 2 decimal places
            - i.e. no 1.54323423456 but 1.54

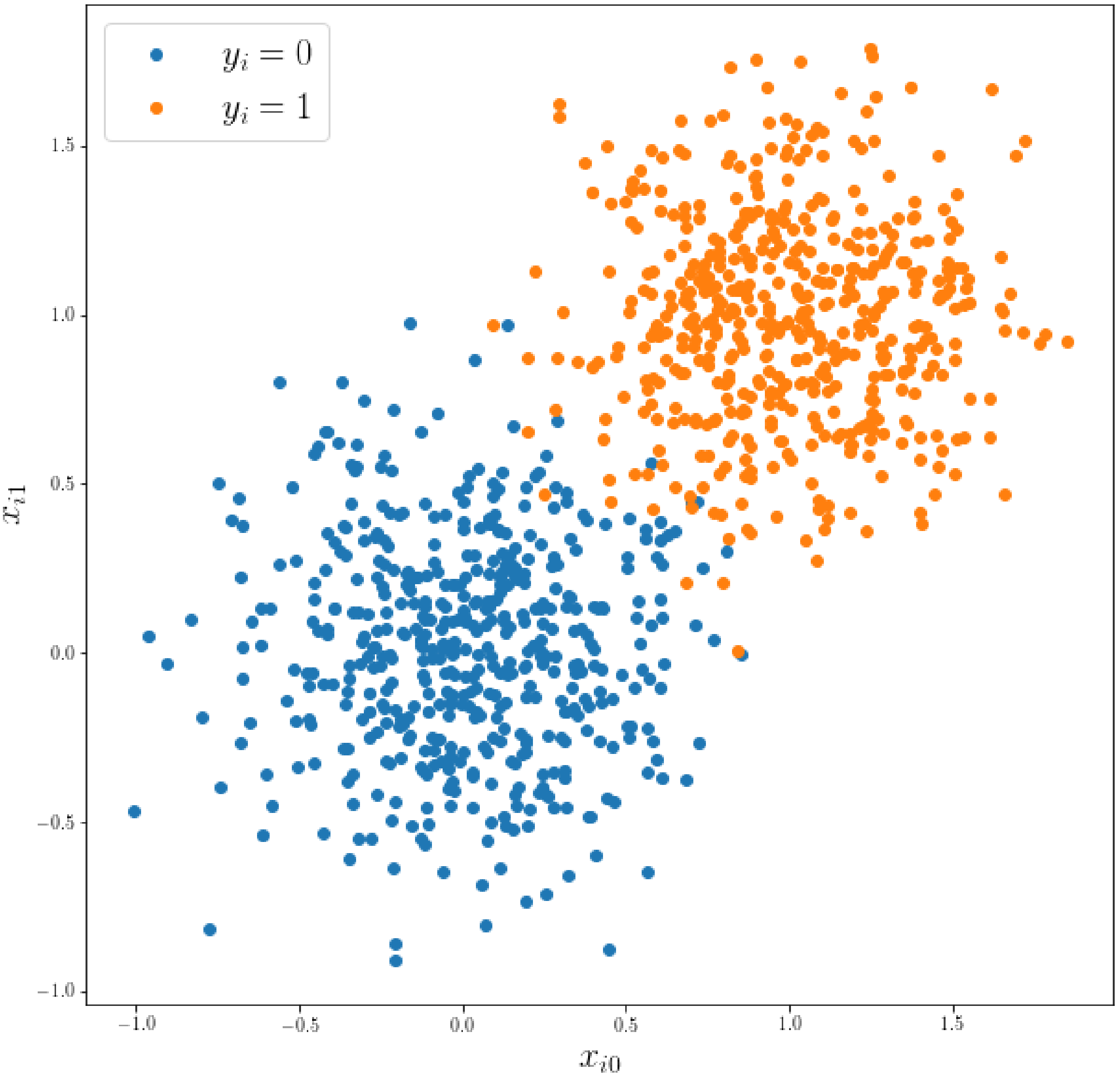
        - There should be no redundant signs before weights, i.e no $1 + -3.36 \times x_i$

- Logistic Regression - *45% + (10%)*

    1. Data Generation - *15%*
        - Randomly generate 1000 (x_{i0}, x_{i1}, y_i) triplets which follows (2)

$$\begin{bmatrix} x_{i0} \\ x_{i1} \end{bmatrix} \sim N\left(\begin{bmatrix} y_i \\ y_i \end{bmatrix}, \begin{bmatrix} 0.1 & 0 \\ 0 & 0.1 \end{bmatrix}\right) \tag{2}$$

where y_i is randomly assigned as 0 or 1.



    2. Model Construction - *20%*
        - **Logistic Regression**
            - Whose divider M_w uses Logistic function L to perform classification

$$\begin{aligned} M_w(x_i) &= L(w \cdot x) \\ &= \frac{1}{1 + e^{-w \cdot x}} \end{aligned}$$

            - Takes L2-norm as the objective function to optimize weight w

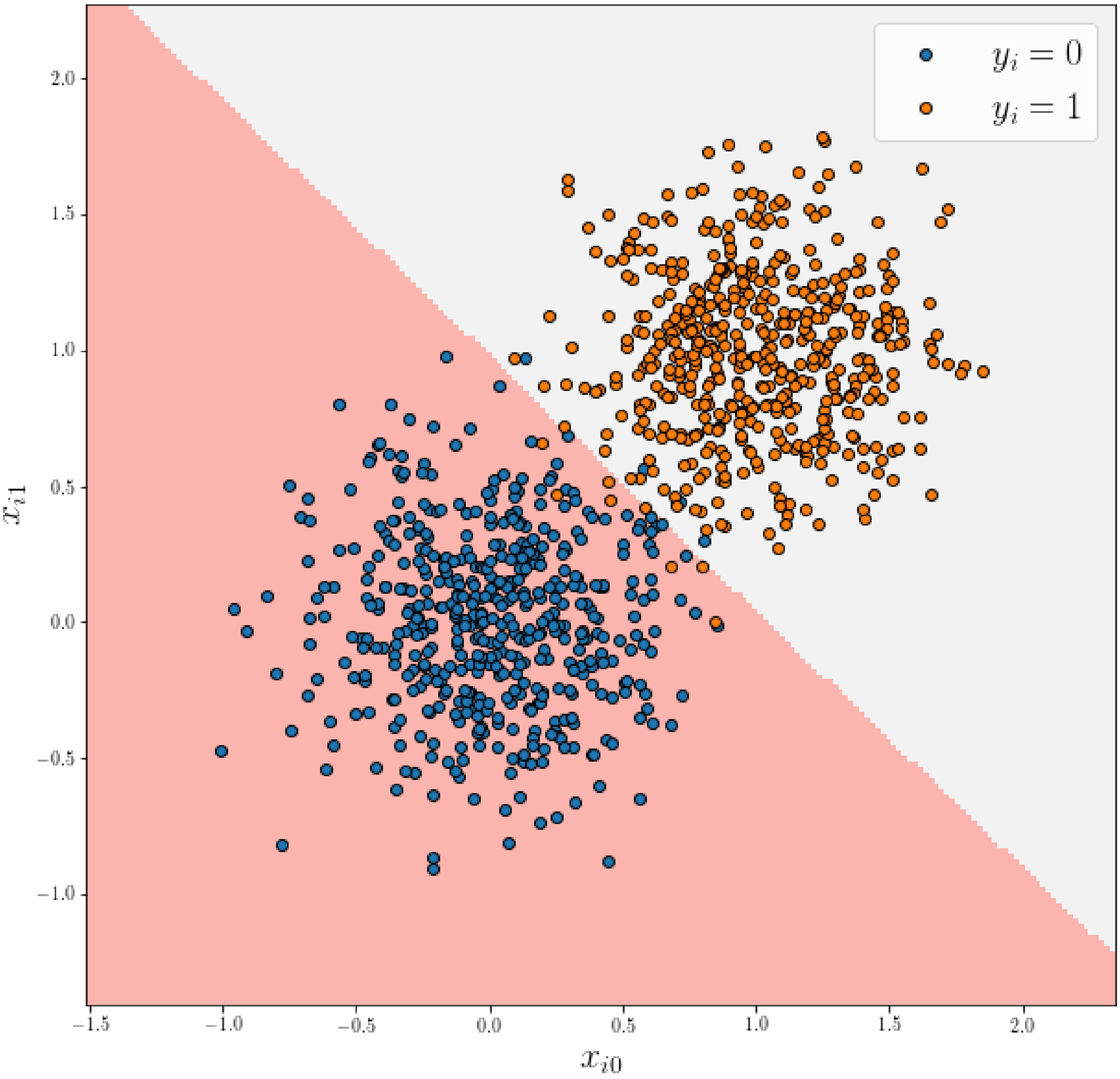
$$w = \underset{w'}{\operatorname{argmin}} ||y - M_{w'(x)}||^2$$

- Construct a **Logistic Regression** model to predict y_i from $[x_{i0} \quad x_{i1}]^T$ generated from equation (2)

3. Validation - *0%*
    - Validation methods are not required in this assignment either.
4. Results - *10%* + *(10%)*
    - Show the model accuracy - *5%*
    - Show the model weights and the corresponded terms - *5%*
        - e.g.

$$y_i = L(4.2 + 7.7 \times x_{i0} + 6.9 \times x_{i1})$$

    - Bonus - show the decision boundary with a figure - *(10%)*



Submission & Scoring Policy

- Please submit a **zip** file, which contains the following, to the newE3 system.
    1. Report
        - Explanation of how your code works.
        - All the content mentioned above.
        - Your name and student ID at the very beginning - 10%

        - Accept formats: **HTML**
            - From markdowns (https://hackmd.io/?nav=overview) or jupyter notebooks.
    2. Source codes
        - Accept languages: **python3**
        - Accept formats: **.ipynb** (https://jupyter.org/)
        - Package-provided models are allowed
- Your score will be determined mainly by the submitted report.
    - if there’s any problem with your code, TA might ask you (through email) to demo it. Otherwise, no demo is needed.
- Scores will be adjusted at the end of the semester for them to fit the school regulations.
- **Plagiarizing is not allowed.**
    - Plagiarizing is checked by MOSS (https://theory.stanford.edu/~aiken/moss/) and manually afterward.
    - You will get **ZERO** on that homework if you get caught the first time.
    - The second time, you’ll **FAIL** this class.
    - 抄襲第一次作業零分，第二次當掉

Tools that might be useful

- Jupyter Lab (https://jupyter.org/) - Better data science experience
- numpy (https://numpy.org/) - Math thingy
- matplotlib (https://matplotlib.org/tutorials/introductory/pyplot.html) - Plot thingy
- pandas (https://pandas.pydata.org/) - Data thingy
- scipy (https://www.scipy.org/) - Science thingy
- scikit-learn (https://scikit-learn.org/stable/) - Machine Learning and stuff