How to replicate the results

This project is divided into three parts: *data preprocessing*, *logistic regression* (fit, predict, and accu_eval), and *testing*.
To understand this project properly, please refer to the complete report while performing the experiment.

Team Members

1. Yi Zhu, 260716006
2. Fei Peng, 260712440
3. Yukai Zhang, 260710915

Python Version

1. **Writing**
    Google Colab `Python 3.6`
2. **Testing**
    Google Colab `Python 3.6`

Imports

1. numpy
2. pandas
3. matplotlib.pyplot
4. time
5. seaborn
6. from google.colab import files
7. from google.colab import drive

Data Preprocessing

Please run the following file:

Data_Preprocessing.ipynb
This file is for *data preprocessing*. Important characteristics of features and the distribution of classes could be found in the result of this file.

Instructions on how to read the output plots:

> The first two plots are the distribution of the two classes of the two provided datasets: *Hepatitis* and *Bankruptcy*.

> The following histogram plots are the distribution of all the features of the two provided datasets.

Logistic Regression

Please run the following file:

Logistic_Regression.ipynb
This file contains the required functions: *fit*, *perdict*, and *accu_eval*, which are contained in a Python class called LogisticRegression, as well as the *k-fold validation* class (KFoldValidation). The hyperparameters and models used in this file are chosen based on the findings in the testing file.

Instructions on how to read the output plots:

> The first block of output plots are: `accuracy vs. iteration number plots` and `gradient vs. iteration number plots` of *Hepatitis* dataset.

> The second block of output plots are: `accuracy vs. iteration number plots` and `gradient vs. iteration number plots` of *Bankruptcy* dataset.

Testing

The testing part is divided into two subparts:

1. To test the hyperparameters of gradient descent algorithm - `hyperparameter testing`
2. To test the effect of z-score normalization and adding/removing features - `normalization and feature testing`

Hyperparameter_Testing.ipynb
This file aims to find the best hyperparameters (of gradient descent algorithm) for the model.

Instructions on how to read the output plots:

The following testing are performed for both datasets.

1. Learning rate testing
2. Stopping criteria testing
    * *Maximum Iteration testing*
    * *Epsilon (threshold for gradient) testing*
3. Beta (Momentum Gradient Descent Constant) testing

> For every hyperparameter test, the first plot is the `mean validation accuracy vs. hyperparameter`, and the second plot is `processing time vs. hyperparameter`.
> You could uncomment some lines of code to explore the result more comprehensively, detailed instructions could be found in the code file.

Normalization_Feature_Testing.ipynb
This file aims to find the effect of normalization, increasing feacures on the model.

Instructions on how to read the output plots:

The following testing are performed for both datasets.

1. *z*-score `normalization` testing
2. Adding more features testing
3. Removing features testing

> In the output plot, the difference between with and without normalization could be seen. Moreover, `the mean validation accuracy vs. added feature order` is also plotted.
> You could uncomment some lines of code to explore the result more comprehensively, detailed instructions could be found in the code file.