Tool for Logistic Regression

Team member: Xuanyi Fu, Qianqi Huang, Shaohua Shen, Shangshang Wang, Rong Yang

Overview

This is a tool built for performing regression analysis on the dataset given with utilization of logistic regression .

It supports automatic logistic regression model generated by scikit_learn library and manual model by user.

User can plot data points with our frontend. Besides, the result of analysis will be stored in file and be presented in frontend graph.

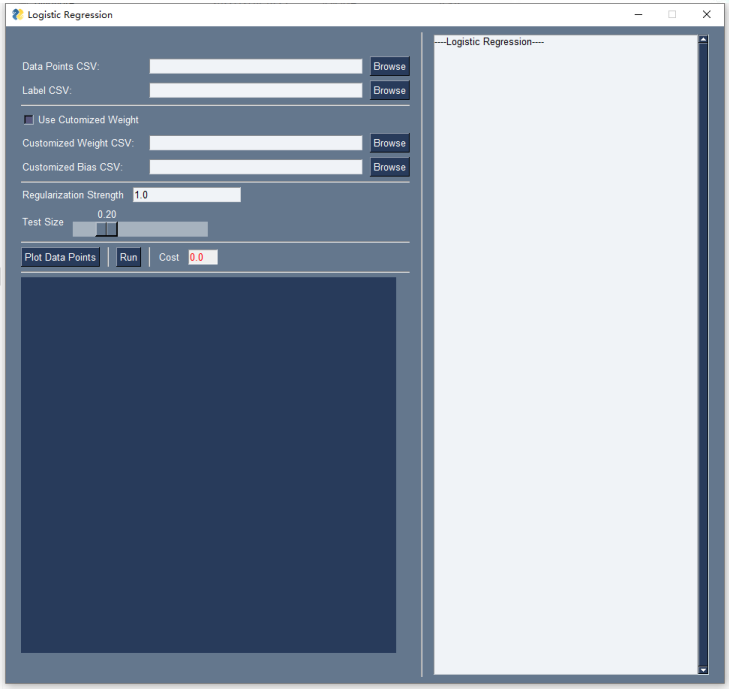
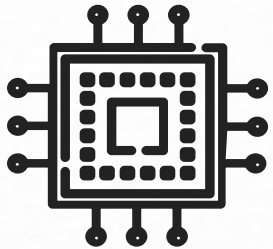


Table of Contents



01

Prerequisite

Prerequisite

First you need to have python 3 installed on your device, and then use **`pip install`** command to install the following libraries and packages.

For creating static, animated, and interactive visualizations

For comprehensive mathematical functions and more

For fast and simple-to-learn GUI programming

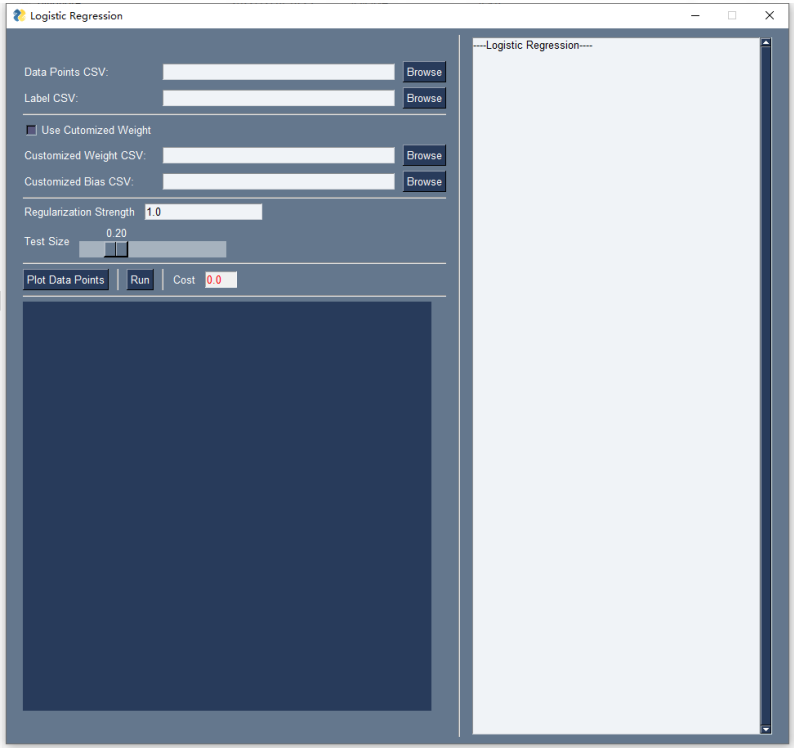
For built-in ready-to-use logistic regression model

02

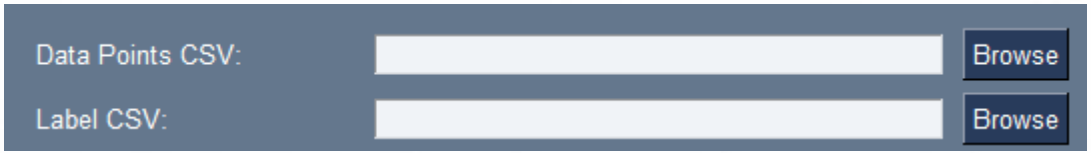
Manual

After Libraries Installed ...

Use python command run main.py in the root directory of the project. A frontend will show up as below



Import Data Set



Here, user can browse local files to import data points and labels for training and testing. In a standard data points csv file, a row is a piece of data and each column stands for a feature value. A standard label csv should contains rows of 0s and 1s, which represents the logistic label for the associated data.

0	19	19000
0	35	20000
1	26	43000
1	27	57000
0	19	76000
0	27	58000
1	27	84000

Data points csv example

1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	1
9	0

Label csv example

Set customized weight

Here, user can click to choose whether he/she would like to use manual logistic regression model. If yes, the user can browse local files to import customized weights and bias values. The dimension of weights and bias should be correct according to the dimension of data points. For example, if data points is a N × M (row × column) matrix, then weights should be M × 1 and bias should be N × 1.

1	0.01
2	0.005
3	0.0006

Weight csv example

1	0.1
2	0.1
3	0.1
4	0.1
5	0.1
6	0.1
7	0.1
8	0.1
9	0.1
10	0.1
11	0.1

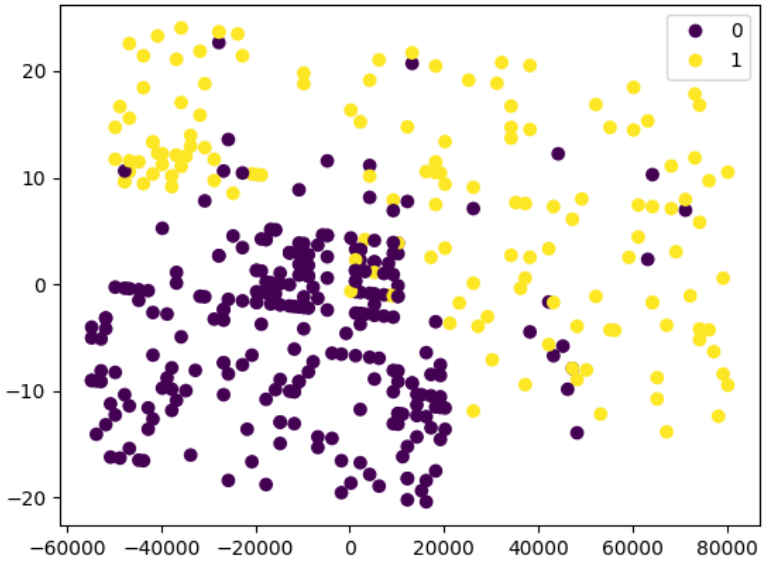
Bias csv example

Set other variables

Here, user can enter the regularization strength of the model and choose the test size, which sets how much data will be used for testing.

Plot data points

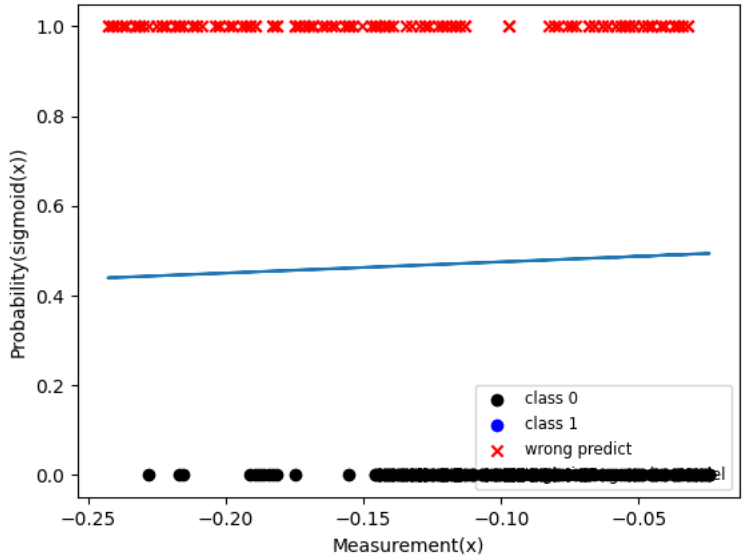
By clicking **Plot Data Points**, user can plot the data set in the frontend with labels represented by different colors.

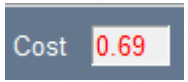


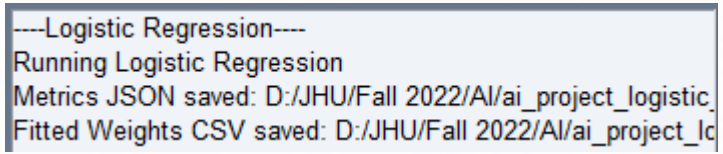
Data points plot example

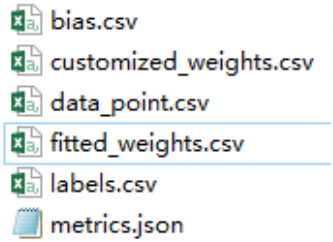
Get the model

By clicking **Run**, user can get the model generated with logistic regression and results will be stored in metrics.json. Cost value will be shown in the frontend just on the right of **Run** button. If the user chose to use automatic logistic regression, a file contains fitted weights will be generated and stored. The directory of saved files will be shown on right.









Result plot example

Contents of metrics.json


```
{
  "0.0": {
    "precision": 0.725,
    "recall": 1.0,
    "f1-score": 0.8405797101449275,
    "support": 58
  },
  "1.0": {
    "precision": 0.0,
    "recall": 0.0,
    "f1-score": 0.0,
    "support": 22
  },
  "accuracy": 0.725,
  "macro avg": {
    "precision": 0.3625,
    "recall": 0.5,
    "f1-score": 0.42028985507246375,
    "support": 80
  },
  "weighted avg": {
    "precision": 0.525625,
    "recall": 0.725,
    "f1-score": 0.6094202898550725,
    "support": 80
  }
}
```

03

Technology

Technical Skills

- Cost value calculation can be customized by using manual logistic regression model with user-specified weights and bias values through our frontend UI.
- We use log loss as our cost function. Each time when the user changes input parameters, our tool would recalculate the cost value and output to the frontend.

END