**HuYuDataInsight LLC**

Zhaowei Cai Apr 24, 2024

**1. Data Preprocessing:**

**(a) Explain what the function train valid split does and why we need this step.**

The function is used to split the dataset into two parts, one is the training part and another is the validation part.

In this function, we have raw data samples (**‘raw\_data’**), target values/labels for data samples (**‘labels’**), the index for splitting the training and validation dataset (**‘split\_index’**).

We use training dataset to train models in machine learning. Learning from the training dataset, the models will minimize its prediction error on the training samples by adjusting its parameters.

We use validation dataset to evaluate the performance of models after training. According to models’ performance on validation dataset, we can know the models’ ability on generalizing to unknown data.

**The reason why we need this step:**(1) We need this step to assess the performance of our models.

(2) We need this step to help us identify overfitting. When models try to fit the training data too much, performing very well on training data but very poor on validation data, the overfitting may occur.

(3) This step can help us to perform experiments on different parameter settings and different machine learning models or algorithms. We can compare them by assessing their performance on the same dataset and choose the models with best results.

**(b) The following explains why not re-train the model before testing.**

No. It is not correct. We should avoid re-training the model on the whole training set before testing.

Since the model has already learned from the training dataset, the model memorizes some patterns and relationships within the training dataset. If you re-train the model on the training dataset, it means that you are giving the same dataset to the model, the model may give us some results from its memory. This can result in very perfect performance on training dataset but very poor generalization on unknown, new data, which is likely an overfitting. The model may be an over-optimized model on training dataset which performs not very well on validation dataset.

**(c) The following explains why we need a third feature which is always 1 in the function prepare X.**

The third feature which is always set to 1 serves as a bias/constant term, it’s a fundamental term in many machine learning models. For logistic regression models, we need to capture the constant term (Intercept). It is not necessary for models to go through the origin with this term, which makes the model more flexible. The third term that always set to 1 also represents the baseline prediction (output when all other features are zero).

**(d) We test our code in “code/main.py” and visualize the training data from class 1 and 2 by implementing the function visualize features. The visualization doesn’t include the third feature. Therefore, it’s a 2-D scatter plot.**

We visualized the training data from class 1 and class 2 by implementing the function ‘visualize features. Here is the result:

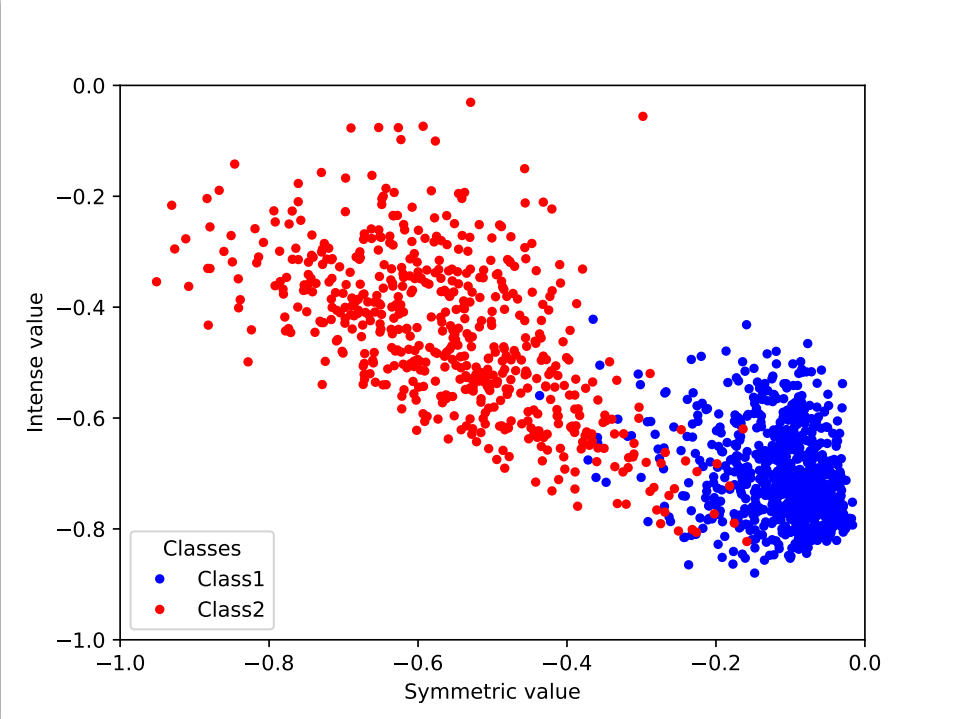


Figure 1 Training Data Visualization

We can easily find that the class 1 and class 2 are very different from each other.

**2. Sigmoid logistic regression:**

**(a) We test our code in “code/main.py” and visualize the results after training by using the function visualize results.**

We visualized the results after training by using the function visualize results. We set the learning rate to be 0.1, and the max iter to be 2000. Here is the result:

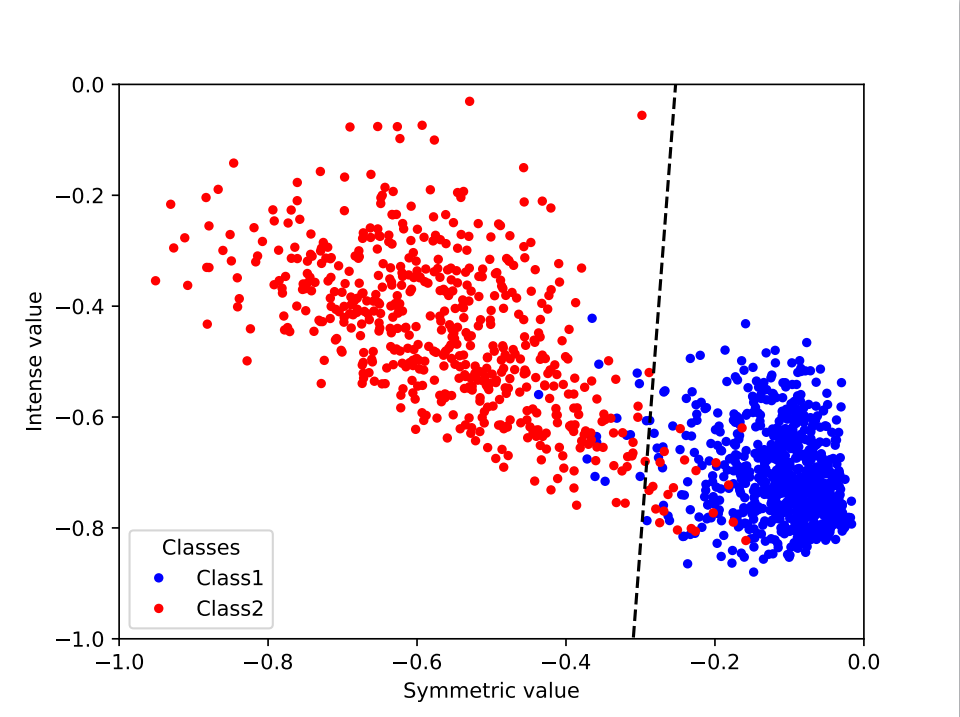


Figure 2 Visualized Results after Training

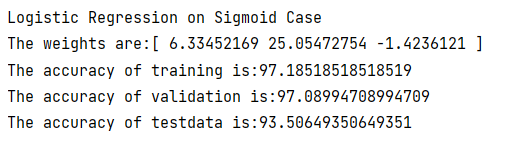


Figure 3 Logistic Regression on Sigmoid Case

The accuracy of training is 97.19%, the accuracy of validation is 97.09%. We can also see the decision boundary line in the figure.

**(b) We implement the testing process and get the test accuracy of our best logistic regression model.**

The accuracy of test data is 93.51%.