Predictive Classification of Breast Cancer Diagnosis

Quyen Linh TA
MIDO Department, University Paris Dauphine, PSL
Pl. du Maréchal de Lattre de Tassigny, 75016 Paris
https://github.com/qlinhta
quyen-linh.ta@dauphine.eu

Ha Anh TRAN
MIDO Department, University Paris Dauphine, PSL
Pl. du Maréchal de Lattre de Tassigny, 75016 Paris
https://github.com/haanh0811
ha-anh.tran@dauphine.eu

Abstract

In this project, we implemented Logistic Regression and Linear Discriminant Analysis (LDA) algorithms to classify breast cancer cases as benign or malignant. The dataset used for this study consisted of 569 samples and 30 features, which were collected from fine needle aspirates of breast mass. The performance of the two algorithms was evaluated using various metrics such as accuracy, precision, recall, and F1-score. Our results show that LDA and Logistic Regression have very good predictive results with overall accuracy of 98.12% and 98.80%. Furthermore, LDA achieved a higher precision, recall and F1-score for both benign and malignant cases. This study demonstrates the effectiveness of various machine learning algorithms in classifying breast cancer cases and highlights the potential of these algorithms for use in clinical decision-making. In addition to Logistic Regression and LDA, this project also implemented several other machine learning algorithms such as Support Vector Machine (SVM), XGBoost, AdaBoost, and CatBoost to compare the results.

2023 Machine Learning MIAGE APP Authors: Ha Anh Tran & Quyen Linh TA

Keywords and phrases Machine Learning — Classification — Logistic Regression — Linear Discriminant Analysis

GitHub Repository https://github.com/qlinhta/ML-PSL

1 Overview of algorithms

To classify breast cancer cases as benign or malignant, in this project, several machine learning algorithms were implemented. These algorithms include Logistic Regression with Gradient Descent, Linear Discriminant Analysis (LDA), Support Vector Machine (SVM) with `scikit-learn`, XGBoost, AdaBoost, and CatBoost.

1.1 Logistic Regression

Logistic Regression is a type of supervised learning algorithm used for classification tasks. It is a statistical method that uses a logistic function to model a binary dependent variable.

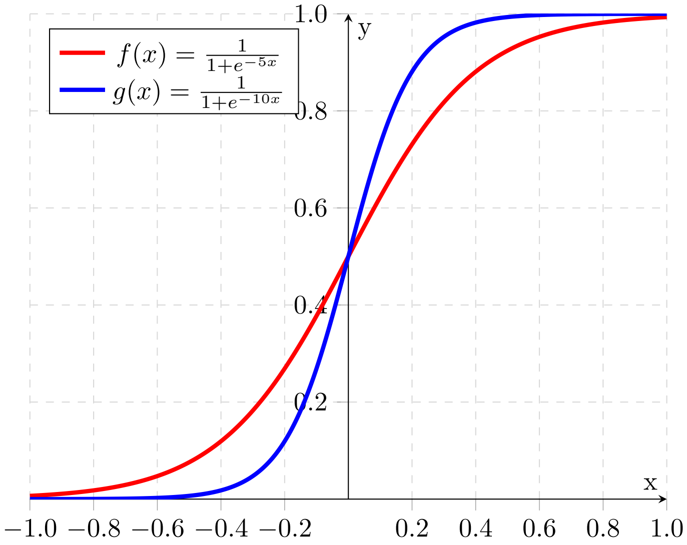

Figure 1 Sigmoid function

The logistic function, also known as the `sigmoid` function, maps any input value to a value between 0 and 1, which can be interpreted as the probability of the input belonging to a certain class.

The logistic regression model is represented by the following equation:

$$p(y = 1|x) = \frac{1}{1+e^{-w^Tx}}$$

- $p(y = 1|x)$ is the probability of the input x belonging to class 1
- e is the base of the natural logarithm
- $-w^Tx$ is the dot product of the input vector x and the model's weight vector w
- w and x are the model's parameters.

The goal of training a logistic regression model is to find the optimal values of the parameters w that maximize the likelihood of the observed data, given the input features. This is usually done using an optimization algorithm such as gradient descent. The loss function used in logistic regression is the `log loss`, also known as the `cross-entropy loss`. The log loss measures the dissimilarity between the predicted probability and the true label.

The log loss for a single example is defined as:

$$L(y, p) = -(y \log(p) + (1 - y) \log(1 - p))$$

- y is the true label (0 or 1)
- p is the predicted probability of the positive class

To compute the overall loss for all the examples in the dataset, we average the loss for each example.

$$J(w) = -\frac{1}{n}\sum_{i=1}^{n}(y^{(i)}\log(p^{(i)}) + (1 - y^{(i)})\log(1 - p^{(i)}))$$

- n is the number of examples in the dataset
- w are the parameters of the model
- $y^{(i)}$ is the true label of the i-th example
- $p^{(i)}$ is the predicted probability of the positive class for the i-th example

Algorithm 1 Gradient Descent for Logistic Regression

```
Initialize the model parameters $w$ with random values
repeat
    for i = 1 to n do
        Compute the predicted probability $p^{(i)} = \frac{1}{1+e^{-w^T x^{(i)}}}$
        Compute the gradient of the loss function with respect to the parameters $w$: $\nabla_w J(w) = -\frac{1}{n}\sum_{i=1}^{n} x^{(i)}(y^{(i)} - p^{(i)})$
    end for
    Update the parameters $w$: $w = w - \alpha \nabla_w J(w)$
until convergence
return the trained parameters $w$
```

Once the model is trained, it can be used to make predictions for new inputs by evaluating the logistic function for the input features and the learned parameters. The output of the logistic function can be interpreted as the probability of the input belonging to a certain class, and a threshold value can be used to make binary predictions (e.g., if the probability is greater than 0.5, predict class 1; otherwise, predict class 0).

1.2 Linear Discriminant Analysis (LDA)

1.3 Linear SVM

1.4 XGBoost

1.5 AdaBoost

1.6 CatBoost