Victor Motogna
Advanced Methods in Data Analysis

Logistic Regression

Specification, analysis and design documentation

The software project covered in this paper represents an implementation for logistic regression, from scratch, in Python. The project does not use an already existing implementation of the data analysis method (from libraries like sci-kit learn, for example). In this documentation, we will cover the aspects of the implementation, the used dataset and analyze the results.

The specification of the project was to build a program that can use a certain dataset and classify prediction results into binary values using logistic regression. The chosen technique actually means creating a relationship between one binary dependent value and one or more independent variables, that we will get from the dataset. For this exact project, we had two key features: viewing the dataset as a visual representation and calculating the prediction (dependent variable), time of run and manipulate the number of iterations in order to view how this influences the results.

The dataset we have used is the well-known Iris dataset, which we have . It contains 150 records, containing 3 variations of Iris flower species - Iris Setosa, Iris Versicolour, Iris Virginica. Each data entry is complete (the dataset has no missing values) and has 4 attributes: sepal length , sepal width, petal length and petal width (all measured in centimeters). As mentioned above, the dependent variable for logistic regression is always binary, which led us to considering only two species (in our case, Setosa & Virginica). Also, because of the reduced dataset (only 50 entries for each species) and the need to prove performance and accuracy differences between the 2 approaches on a powerful machine, we initially took into consideration all 4 attributes when computing the predictor value, but also only two - the sepal length and width.

The implementation we have approached was to write a logistic regression class for the prediction algorithm and a main program for a console menu and plot the data/results. The main tools used are Python 3 (or above), pip (a tool for installing Python dependencies), matplotlib (a Python library designed for plots and visual representations) and numpy (a Python library designed for array, matrix and scientific computing). These two external libraries were used only as helpers for our implementation, as the algorithm was still implemented from scratch, without an instance of a logistic regression implementation. In our approach, the general idea was to take a set of inputs X from the dataset and assign them a value: 0 or 1. In order to do this, we have created a class by our own for the logistic regression algorithm, that contains a sigmoid method: the function that outputs values between 0 and 1, assigning the dependent value. Besides this method, we also needed a loss function (that will assign weights, starting from random values), a function to measure the gradient descent (that will compute the derivative of the loss function with respect to each weight, in order to minimize the loss function) and two prediction functions that will assign a value between 0 and 1 given by the sigmoid to the value needed (0 or 1 - this will differ based on the problem and the binary values chosen). We have also considered a variable to handle the number of iterations to view how different values influence the results. For the main program, we have created a simple console menu to access the possible features and also have the main methods for them.

To run the program, we only need a few setup steps: installing Python 3.x on the machine, installing pip and the necessary dependencies - matplotlib, scikit learn, numpy - and run the program - we are using python3 main.py. The code has already implemented a way to get the dataset from the scikit learn datasets module and the necessary import logic to be ready to go.