



# Data Science × Logistic Regression

Harry Potter and the Data Scientist

*Summary: Write a classifier and save Hogwarts!*

*Version: 5*

# Contents

I	Foreword	2
II	Introduction	3
III	Objectives	4
IV	General Instructions	5
V	Mandatory Part	6
V.1	Data Analysis . . . . .	6
V.2	Data Visualization . . . . .	7
V.2.1	Histogram . . . . .	7
V.2.2	Scatter plot . . . . .	7
V.2.3	Pair plot . . . . .	7
V.3	Logistic Regression . . . . .	8
VI	Bonus Part	9
VII	Submission and peer-evaluation	10
VIII	Annexes	11
VIII.1	Mathematics . . . . .	11
VIII.2	Data visualization examples . . . . .	12

# Chapter I

## Foreword

This is what Wikipedia says about Yann Le Cun, one of the founding fathers of AI :

Yann Le Cun was born near Paris, France, in 1960. He is a researcher in artificial intelligence and computer vision (robotics). He is considered one of the inventors of deep learning.

Obtaining a B.S. degree from the ESIEE Paris in 1983, he graduated from the Pierre and Marie Curie University and received a PhD in 1987, during which he proposed an early form of the back-propagation learning algorithm which is commonly used by the gradient descent optimization algorithms to adjust the weight of neurons by calculating the gradient of the loss function. He was a postdoctoral research associate in Geoffrey Hinton's lab at the University of Toronto from 1987 to 1988.

Since 1980s Yann Le Cun has been working on machine learning, computer vision and deep learning : the ability of a computer to recognize representations (images, texts, videos, sounds) by repeatedly exposing them to the training samples.

In 1987, Yann Le Cun joined the University of Toronto and in 1988 the AT&T research facilities, where he developed the methods of supervised learning.

Yann Le Cun is also one of the main creators of the DjVu image compression technology (together with Léon Bottou and Patrick Haffner).

Yann Le Cun is a professor at the University of New York where he has created the Center for Data Science. He works in particular on technological development of autonomous cars.

On December 9, 2013, Yann Le Cun was invited by Mark Zuckerberg to join Facebook to design and run the artificial intelligence lab FAIR (« Facebook Artificial Intelligence Research ») in New York, Menlo Park and since 2015 in Paris, to work on the image recognition. He had previously refused a similar propose from Google.

In 2016, he was the visiting professor of computer science on the "Chaire Annuelle Informatique et Sciences Numériques" at Collège de France in Paris.

# **Chapter II**

## **Introduction**

On no! Since its creation, the famous school of wizards, Hogwarts, had never known such an offense. The forces of evil have bewitched the Sorting Hat. It no longer responds, and is unable to fulfill his role of sorting the students to the houses.

The new academic year is approaching. Gladly, the Professor McGonagall was able to take action in such a stressful situation, since it is impossible for Hogwarts not to welcome new students. . . She decided to call on you, a muggle "datascientist" who is able to create miracles with the tool which all muggles know how to use: a "computer".

Despite the intrinsic reluctance of many wizards, the director of the school welcomes you to his office to explain the situation. You are here because his informant discovered that you are able to recreate a magic Sorting Hat using your muggle tools. You explain to him that in order for your "muggle" tools to work, you need students data. Hesitantly, Professor McGonagall gives you a dusty spellbook. Fortunately for you, a simple "Digitalis!" and the book turned into a USB stick.

# Chapter III

## Objectives

In this project *Data Science × Logistic Regression*, you will continue your exploration of *Machine Learning* by discovering different tools.

The use of the term *Data Science* in the title may be considered by some as excessive. That is true. We do not claim to give you all the basics of *Data Science* in this topic. The subject is vast. We will only cover some basics that we believe are useful for data exploration before sending it to the *machine learning* algorithm.

You will implement a linear classification model as a continuation of the *linear regression* subject: a *logistic regression*. We also encourage you to create a *machine learning* toolkit as you progress along the way.

Summarizing:

- You will learn how to read a data set, visualize it in different ways, and select and clean unnecessary information from your data.
- You will train a logistic regression model that will solve a classification problem.

# Chapter IV

## General Instructions

You can use any language you want. However, we recommend that you choose a language with a library that facilitates plotting and calculating statistical properties of a dataset.

Any method that performs all the heavy lifting for you (for example, the `describe()` method from a `pandas` DataFrame) will be considered cheating.

# Chapter V

## Mandatory Part



It is highly recommended to perform the steps in the following order.

### V.1 Data Analysis



We will see some basic steps of data exploration. Of course, these are not the only techniques available or the only steps to follow. Each dataset and problem has to be approached in a unique way. You will surely find other ways to analyze your data in the future.

First of all, take a look at the available data. Look at what format it is presented in, whether there are various types of data, the different ranges, and so on. It is important to make an idea of your raw material before starting. The more you work with data, the more you will develop an intuition about how you will be able to use it.

In this part, Professor McGonagall asks you to produce a program called `describe.[extension]`. This program will take a *dataset* as a parameter. All it has to do is display information for all numerical *features* like in the example:

	Feature 1	Feature 2	Feature 3	Feature 4
Count	149.000000	149.000000	149.000000	149.000000
Mean	5.848322	3.051007	3.774497	1.205369
Std	5.906338	3.081445	4.162021	1.424286
Min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.400000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
Max	7.900000	4.400000	6.900000	2.500000



It is forbidden to use any function that does the job for you, such as: `count`, `mean`, `std`, `min`, `max`, `percentile`, etc., no matter the language that you use. Of course, it is also forbidden to use the `describe` library or any function that looks (more or less) similar to it from another library.

## V.2 Data Visualization

Data visualization is a powerful tool for a data scientist. It allows you to gain insights and develop an intuition of what your data looks like. Visualizing your data also allows you to detect defects or anomalies.

In this section, you are asked to create a set of scripts, each using a particular visualization method to answer a question. There is not necessarily a single answer to the question.

### V.2.1 Histogram

Create a script called `histogram.[extension]` which displays a *histogram* answering the following question:

Which Hogwarts course has a homogeneous score distribution between all four houses?

### V.2.2 Scatter plot

Create a script called `scatter_plot.[extension]` which displays a *scatter plot* answering the following question:

What are the two *features* that are similar?

### V.2.3 Pair plot

Create a script called `pair_plot.[extension]` which displays a *pair plot* or *scatter plot matrix* (depending on the library that you are using).

From this visualization, which features are you going to use for your logistic regression?

## V.3 Logistic Regression

You arrive at the last part: code your Magic Hat. To do this, you have to perform a multi-classifier using a logistic regression *one-vs-all* (*one-vs-rest*).

You will have to create two programs:

- The first one will *train* your models, and it's called `logreg_train.[extension]`. It takes `dataset_train.csv` as a parameter. For the mandatory part, you must use the technique of *gradient descent* to minimize the error. The program generates a file containing the weights that will be used for the prediction.
- The second one must be named `logreg_predict.[extension]`. It takes `dataset_test.csv` as a parameter and a file containing the weights trained by the previous program.

In order to evaluate the performance of your classifier, this second program will have to generate a prediction file `houses.csv` formatted exactly as follows:

```
$> cat houses.csv
Index,Hogwarts House
0,Gryffindor
1,Hufflepuff
2,Ravenclaw
3,Hufflepuff
4,Slytherin
5,Ravenclaw
6,Hufflepuff
[...]
```

# Chapter VI

## Bonus Part

There are many interesting bonuses that can be added to this subject. Here are some suggestions:

- Add more fields to `describe`. [extension]
- Implement *stochastic gradient descent*
- Implement other optimization algorithms (Batch GD, mini-batch GD, or others)



The bonus part will only be assessed if the mandatory part is PERFECT. Perfect means that the mandatory part has been fully completed and works without any issues. If you have not met ALL the mandatory requirements, your bonus part will not be evaluated.

# Chapter VII

## Submission and peer-evaluation

Turn in your assignment in your `Git` repository as usual. Only the work inside your repository will be evaluated during the defense. Don't hesitate to double-check the names of your folders and files to ensure they are correct.

During the correction, you will be evaluated on your turn-in (no functions that do all the heavy-lifting for you), as well as your ability to present, explain, and justify your choices.

Your classifier will be evaluated on the data present in `dataset_test.csv`. Your answers will be evaluated using the [accuracy score](#) from the *Scikit-Learn* library. Professor McGonagall agrees that your algorithm is comparable to the Sorting Hat only if it has a minimum accuracy score of **98%**.

It will also be important to be able to explain the functioning of the *machine learning* algorithms used.

# Chapter VIII

## Annexes

### VIII.1 Mathematics

Logistic regression works almost like the linear regression. Here is a cost (*loss*) function:

$$J(\theta) = -\frac{1}{m} \sum_{i=1}^m y^i \log(h_\theta(x^i)) + (1 - y^i) \log(1 - h_\theta(x^i))$$

Where  $h_\theta(x)$  is defined in the following way :

$$h_\theta(x) = g(\theta^T x)$$

With :

$$g(z) = \frac{1}{1 + e^{-z}}$$

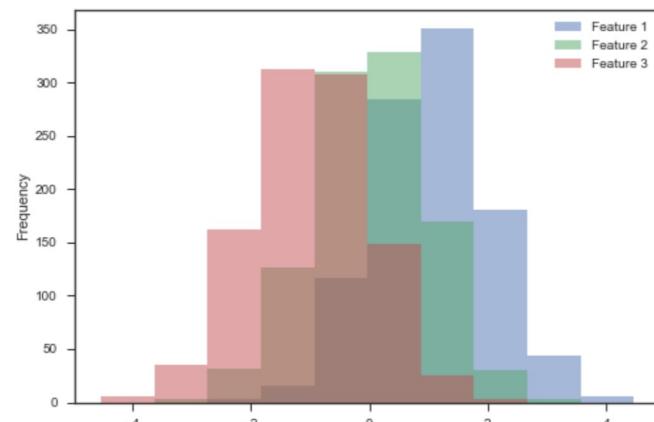
The loss function gives us the following partial derivative :

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{i=1}^m (h_\theta(x^i) - y^i) x_j^i$$

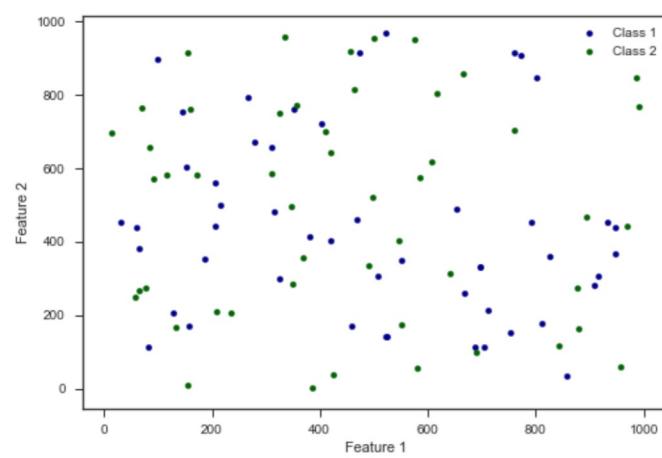
## VIII.2 Data visualization examples

Here are some examples of data visualization :

- Histogram



- Scatter plot



- Pair plot

