TD 2 Machine Learning : Logistic Regression

Adam Goux–Gateau

October 2023

1 Introduction

This work contains an implementation of a logistic regression for two different data sets. The purpose is to predict in which class a feature will belong to.

You can find all the codes here: GitHub TD2 Repository.

2 Logistic regression on one single class

In this section, we will determine whether a student is eligible for a university or not. The first step is to examine data to see if students are admitted or not based on their results in two exams. We first plot the data :

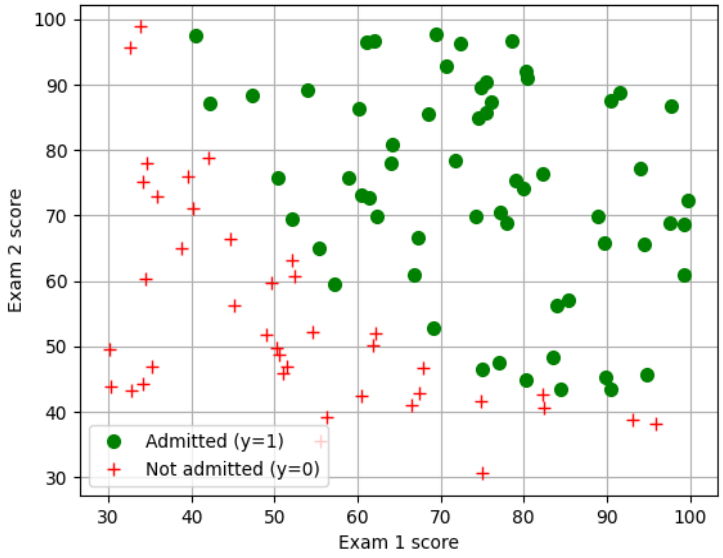

Figure 1: Admission according to the grade

Then, we introduce the cost function $J(\theta)$ to minimize. The formula for $J(\theta)$ is : $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(-y^{(i)}\log\left(h_\theta(x^{(i)})\right) - (1-y^{(i)})\log\left(1-h_\theta(x^{(i)})\right)\right)$

It indicates how far away the model is of the reality, the key to solve a machine learning problem is often to minimize this function. Then, we can perform a gradient descent by differentiating J with respect to θ :

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

By running the code, we obtain the following result, which is coherent with what we should get.

```
Cost at test theta: 0.218330
Expected cost (approx): 0.218
Gradient at test theta:[0.04290299 2.56623412 2.64679737]
Expected gradients (approx): 0.043 2.566 2.647
```

Figure 2: Cost and gradient

Then we needed to predict the result of students. We will use the sigmoid function ($f : z \mapsto \frac{1}{1+e^{-z}}$). If the results is ≥ 0.5, then the student is accepted, and rejected if < 0.5 . We obtain this graph :

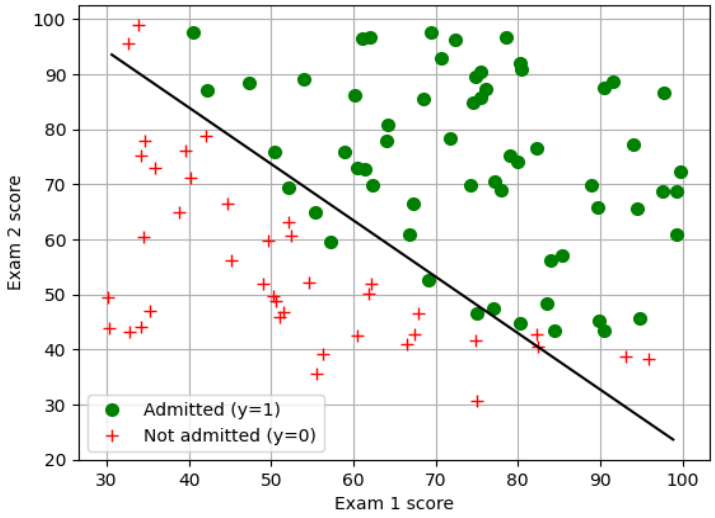

Figure 3: Decision boundary

For the prediction of the admission of a student with 45 and 85 score, we get this result :

```
For a student with scores 45 and 85, we predict an admission probability of 0.776291
Expected Proba (approx): 0.776
```

Figure 4: Expected and real results

3 Regularization

The goal is to deduce if an electronic microchip will pass through the quality test. The initial data set looks like this :

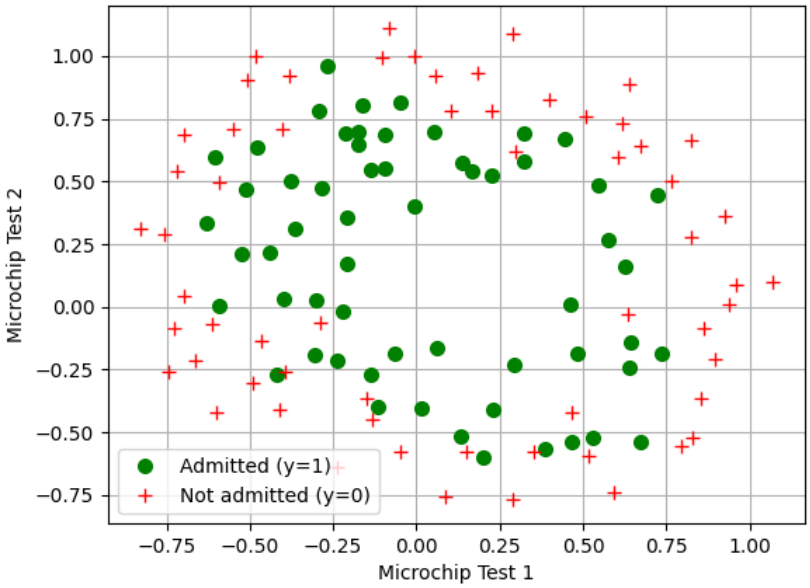

Figure 5: Initial features

We will use a polynomial approach to try to separate the two different classes. The cost function is almost the same, we just add : $\frac{\lambda}{2m}\sum_{j=1}^{n-1}\theta_j^2$. The purpose of λ is to prevent overfitting of the different models. Here are the plots I obtain with different values of λ :

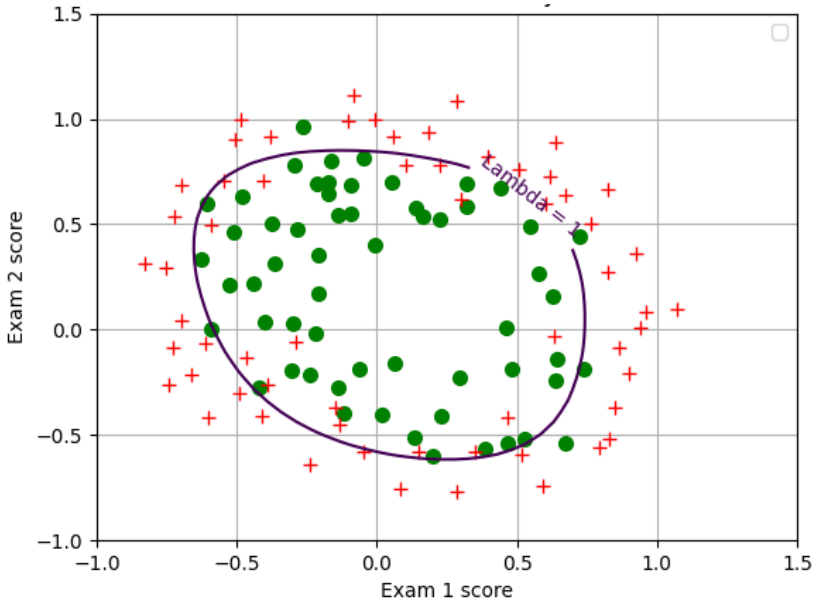

Figure 6: Decision boundary $\lambda = 1$

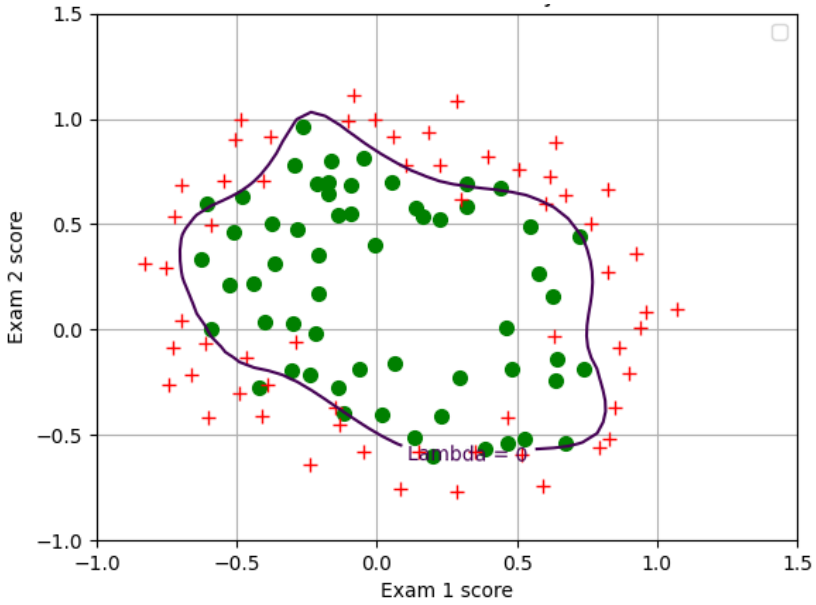

Figure 7: Decision boundary $\lambda = 0$

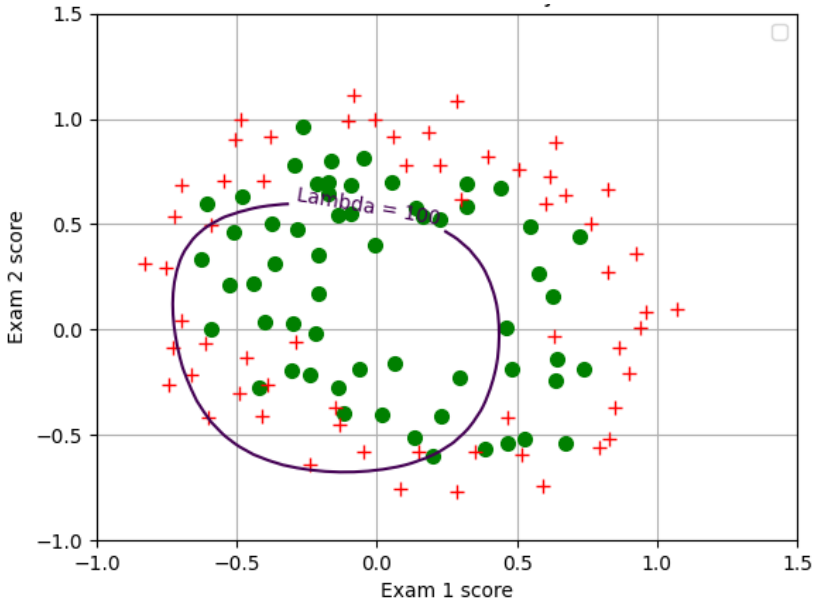

Figure 8: Decision boundary $\lambda = 100$

We can witness an overfitting if $\lambda = 0$, and an underfitting if $\lambda = 1$

4 Multiclass

We will here recognize hand-written figures. When zoomed in, the data look like this :

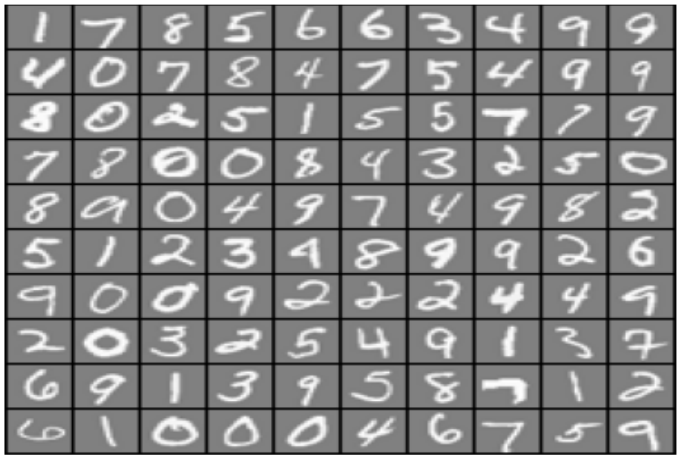

Figure 9: Hand written figures

I implemented the same cost and gradient functions than before. Here are the results :

```
Cost: 0.734819
Expected cost: 0.734819
Gradients:[0.14656137 0.05144159 0.12472227 0.19800296]
Expected gradients: 0.14656137  0.05144159  0.12472227  0.19800296
```

Figure 10: Result for multiclass gradient

As you can observe, it fits perfectly with what I should have obtained. Now, we perform a "one-vs-all" classification by training several regularized logistic regression classifiers, one for each of the K classes in our data set. We then create a classifier for each class. Here are the results :

```
Training Set Accuracy: 96.460000

Expected approx accuracy: 96.46%
```

Figure 11: Training set accuracy

5 More questions !

In the TD1, we used a linear regression to interpolate data. In this one, we did a classicication to identify the belonging class of data. The cost function is not the same : it is based on probabilities.

The regularisation principle prevents the overfitting and improve the generalization of the model outside of the data set we provide. We can avoid this overfitting thanks to the λ parameter. The smaller it is, the more accurate the model is with our data, which can cause overfitting. The bigger it is, the frontier is not precise at all. We need to decide which one to chose to improve our model.

The multiclass approach is used when the data can belong to more than 2 different classes. We can train a binary classificator for each class (what we did here), or for each possible pair.