Gaumart Siméon
0845209

Machine Learning HW3
2020.4.30

Problem 1:

We have to prove that $H(p(x),q(x))=D_{KL}(p(x)||q(x))+H(p(x))$.

We know that $D_{KL}(p(x)||q(x))=\sum_x p(x)\log\frac{p(x)}{q(x)}$.

Furthermore, H(p(x)) is the Entropy of x (with probability p(x)), so

$$H(p(x))=-E_p[\log p(x)]=-\sum_x p(x)\log p(x)=\sum_x p(x)\log\frac{1}{p(x)}$$

So

$$D_{KL}(p(x)||q(x))+H(p(x))=\sum_x p(x)\log\frac{p(x)}{q(x)}+\sum_x p(x)\log\frac{1}{p(x)}$$
$$\Leftrightarrow D_{KL}(p(x)||q(x))+H(p(x))=\sum_x p(x)[\log\frac{p(x)}{q(x)}+\log\frac{1}{p(x)}]$$
$$\Leftrightarrow D_{KL}(p(x)||q(x))+H(p(x))=\sum_x p(x)[\log p(x)-\log q(x)+\log 1-\log p(x)]$$
$$\Leftrightarrow D_{KL}(p(x)||q(x))+H(p(x))=\sum_x p(x)[-\log q(x)+\log 1]$$
$$\Leftrightarrow D_{KL}(p(x)||q(x))+H(p(x))=\sum_x p(x)\log\frac{1}{q(x)}$$

And we know that $H(p(x),q(x))=-E_p[\log(q(x))]=\sum_x p(x)\log\frac{1}{q(x)}$

So we have $H(p(x),q(x))=D_{KL}(p(x)||q(x))+H(p(x))$

Problem 2:

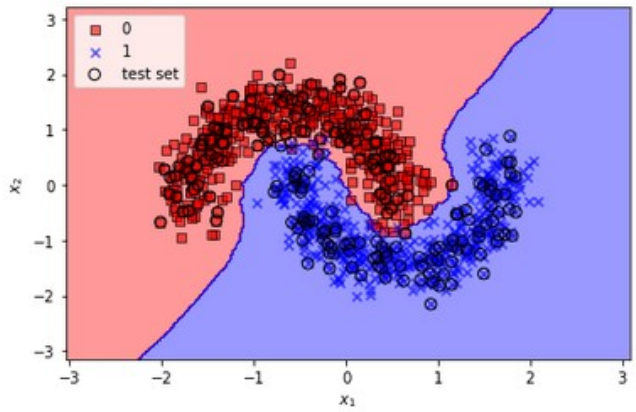
For this problem, I chose to show the results for a total of 1000 number of data (so a training set of 800 data and a testing set of 200 data after splitting).

a)
We implement a KNN classifier using a Euclidian distance metric and K=11.

We find a number of misclassified samples of 4 (over 200) and an accuracy of 98%.

Gaumart Siméon
0845209

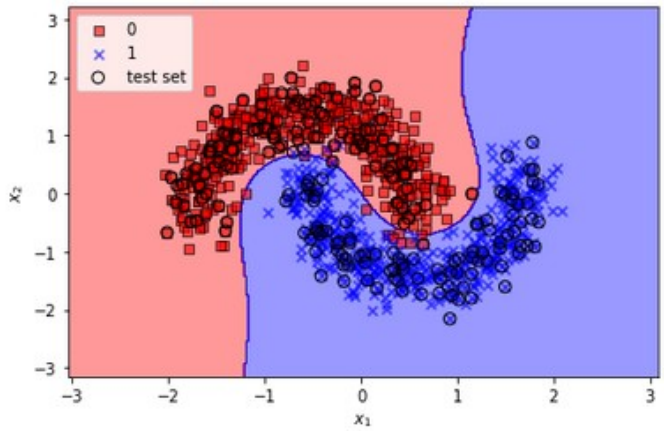
We can see the decision regions on the following graph :



b)
We implement a SVM classifier using a rbf kernel where random_state=0, γ=0.2 and C=10.0.

We find a number of misclassified samples of 4 (over 200) and an accuracy of 98%.

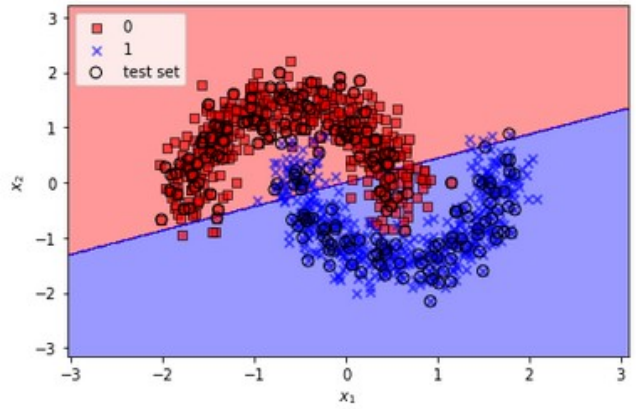
We can see the decision regions on the following graph :



c)
We implement a SVM classifier using a linear kernel where random_state=0 and C=1000.0.

We find a number of misclassified samples of 25 (over 200) and an accuracy of 88%.

We can see the decision regions on the following graph :

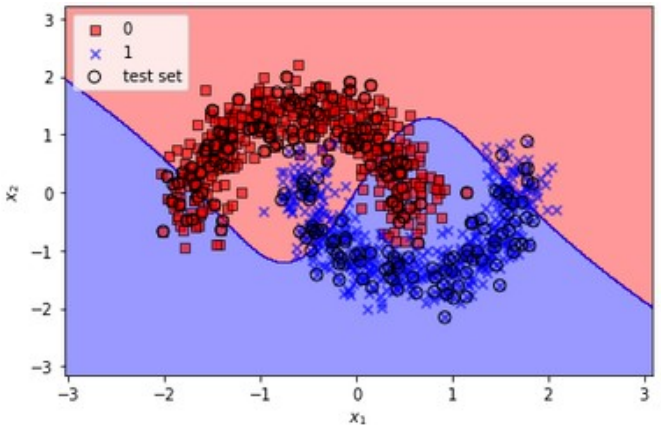


Gaumart Siméon
0845209

d)
We implement a SVM classifier using a sigmoid kernel.

We find a number of misclassified samples of 58 (over 200) and an accuracy of 71%.

We can see the decision regions on the following graph :

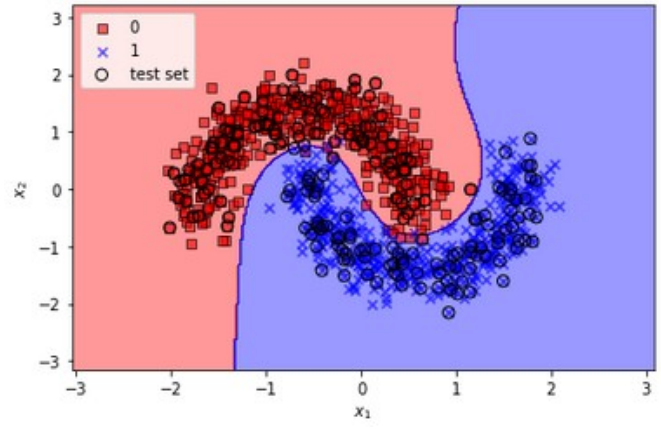


e)
In this sub-problem, we try to find the best pair of C and γ for a SVM classifier using random_state=0.
So we try each pair and keep those where we have the lower number of misclassified samples (and the higher accuracy).

With the previous set of data, we find that the best pairs of C and γ are :
- C=1000.0 γ=0.1
- C=1000.0 γ=1.0
- C=10000.0 γ=0.1

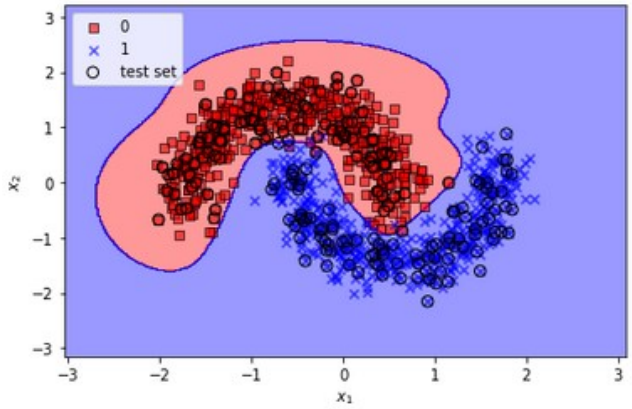
For each of these pairs, we find a number of misclassified samples of 2 (over 200) and an accuracy of 99%.

We can see the decision regions for C=1000.0 and γ=0.1 on the following graph :

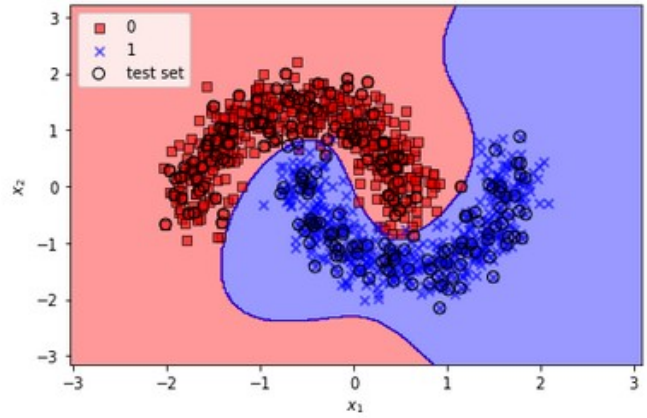


Gaumart Siméon
0845209

For C=1000.0 and γ=1.0 :



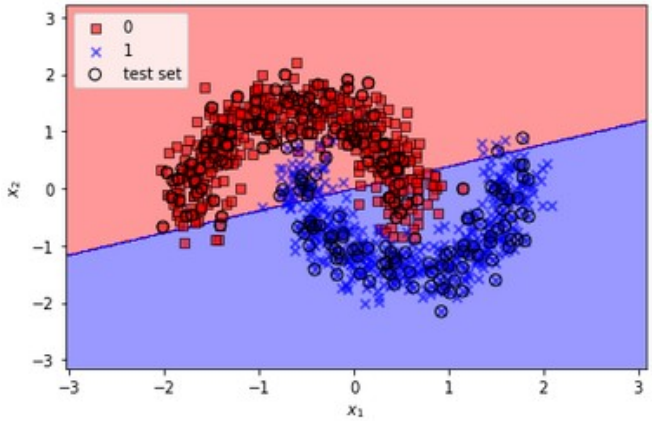
For C=10000.0 and γ=0.1 :



f)
We implement a Logistic Regression using a liblinear solver, C=1000.0 and random_state=0.

We find a number of misclassified samples of 26 (over 200) and an accuracy of 87%.

We can see the decision regions on the following graph :



Gaumart Siméon
0845209

Problem 3:

a)
The MSE, because we take the square of the difference (so that it magnifies the error), allow us to get a model without big errors with no outlier predictions.
However, if only one prediction is bad, the quadratic part of the MSE magnifies the error even if we don't care of this value (it could be noise, error), and the model will be less balanced.
It fits only with regression problems.

b)
Against MSE, MAE weighted the big errors so the model is much more balanced and the loss function is uniform related to the errors (because it's linear so a little error have the same importance (proportionally) compared to a big error).
However, if the big errors are important, the model will weighted it with the others prediction so it will often give good predictions, but, sometimes, give really bad predictions.
It fits only with regression problem.

c)
This loss function is taking the maximum of all incorrect classes, when $y \neq t$, and return it at the loss (+1).
When a x is classed correctly, the loss is 0. However, we have a gap of 1 where, if the x is classed correctly but too close from the boundary, the $max_{y \neq t} \{ w_y x - w_t x \}$ will be between -1 and 0, and the loss will not be null (but between 0 and 1).

d)
The Cross Entropy Loss fits well for classification problems with 2 classes and where we work with probabilities. However, it fits only with classification and we may use an other loss function (or adapt it) for multi-classification problems.
Furthermore, the function $-\log(1-\hat{y}_i)$ tends to infinity when $\hat{y}_i$ tends to 1. So the loss function could tend to infinity for some probabilities, if the prediction is bad. But it's really a particular case.