



La grande école du numérique pour tous

Perceptron Multicouches: implémentation et application à un dataset

Présenté par:

Yann SASSE

Alexandre TEISSIER

Xavier TRENTIN

Hedi ZARROUK

MASTER OF SCIENCE IA & DATA

2024-2025



Sommaire

3

Présentation du MLP

5

**Simulations et Visualisation
du playground**

5

MLP en POO

6

**Application avec la base de données
Predict Students' Dropout and Academic
Success**

9

Perspectives

11

Conclusion

1. Présentation du MLP

Définition:

- réseau de neurones avec plusieurs couches cachées entre la couche d'entrée et la couche de sortie;
- permet de modéliser des relations non linéaires complexes entre des features et une variable d'intérêt (avec ReLU, tanh, sigmoïde)

Architecture classique:

- une couche d'entrée: données brutes, features/attributs..
- des couches intermédiaires/cachées: avec un nombre de neurones qui dépend de la complexité du problème.
- une couche de sortie.

Termes associés

Fonction d'activation

Propagation

Rétropropagation

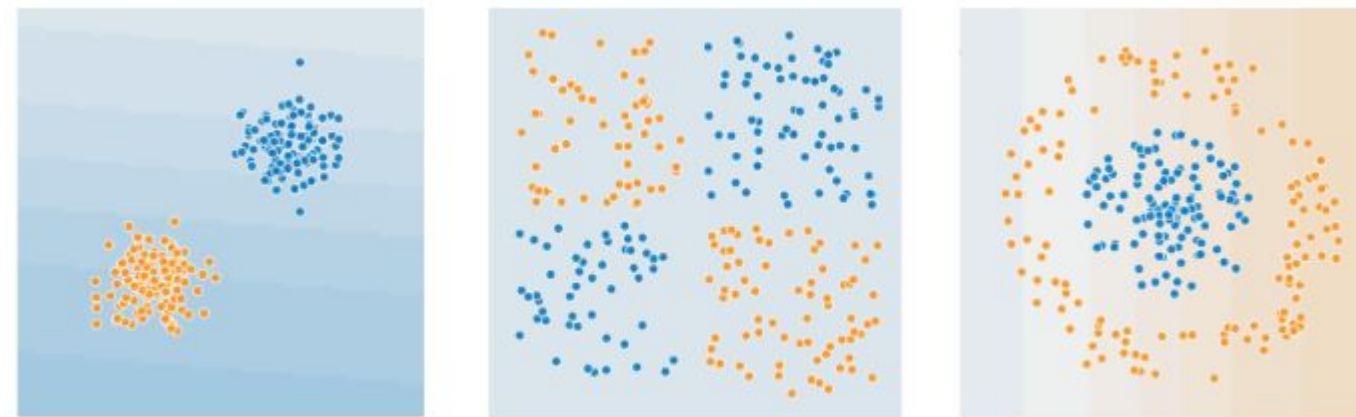
Loss function

Descente de gradients

Vanishing gradients

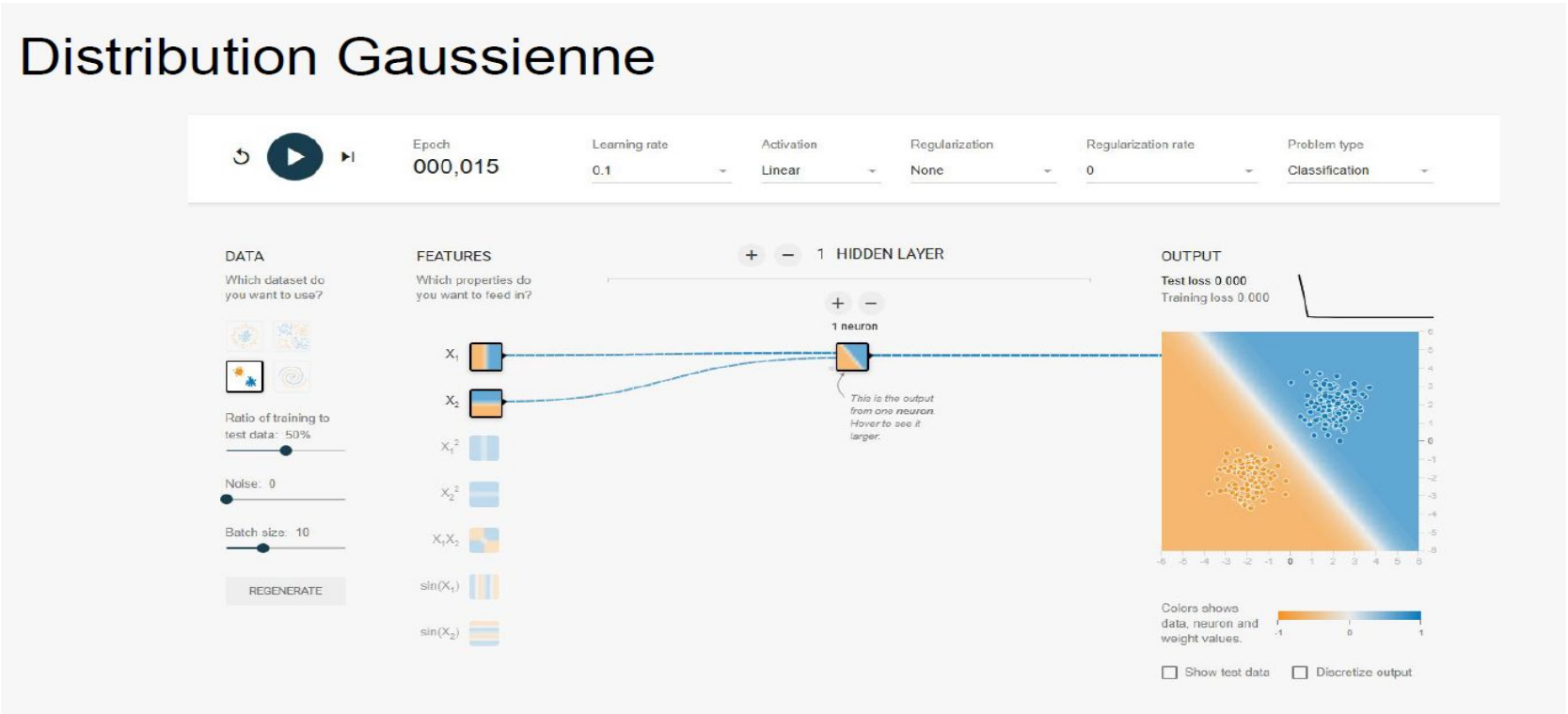
ANN playground

- ingénierie de caractéristiques en fonction de la distribution des classes
- construction du MLP (couches cachées et rôle des hyperparamètres)

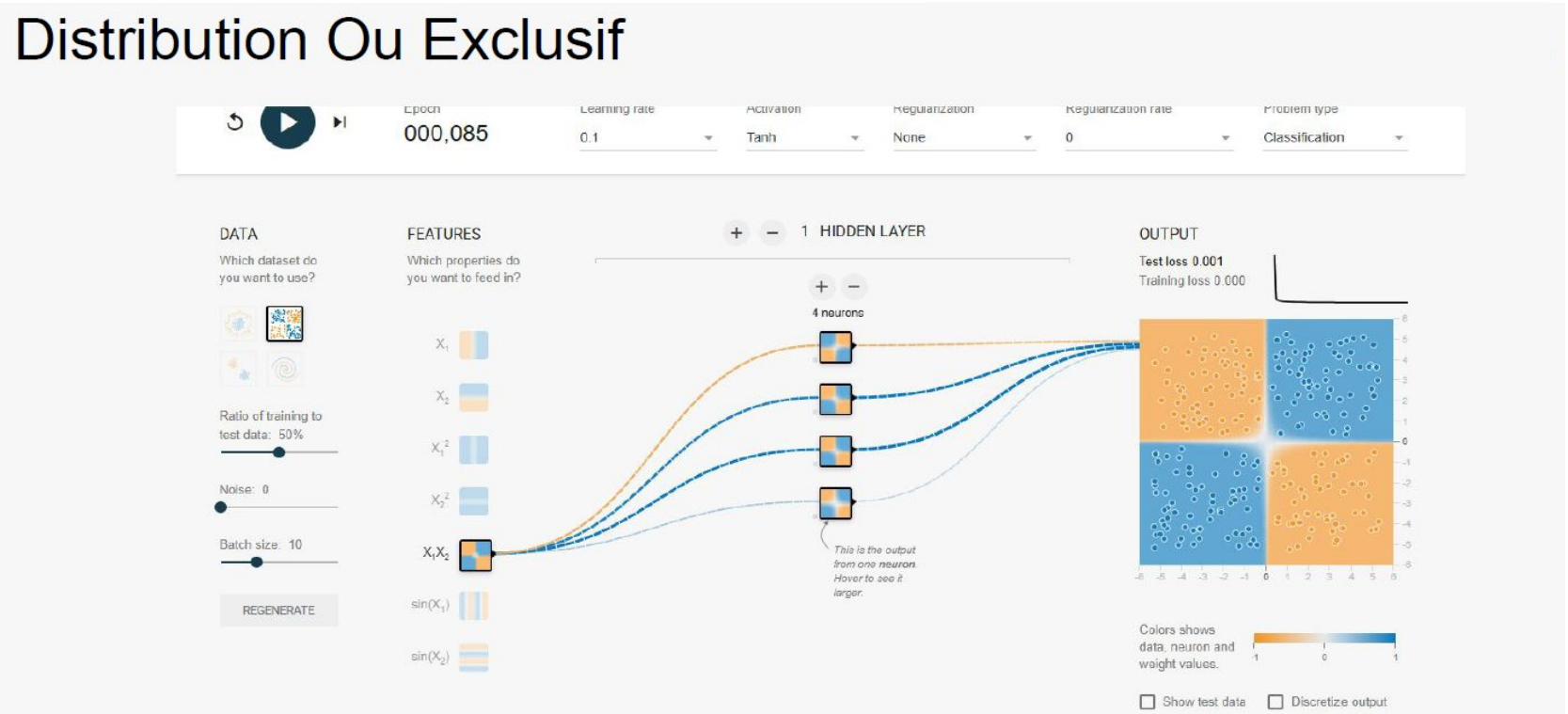


ANN playground

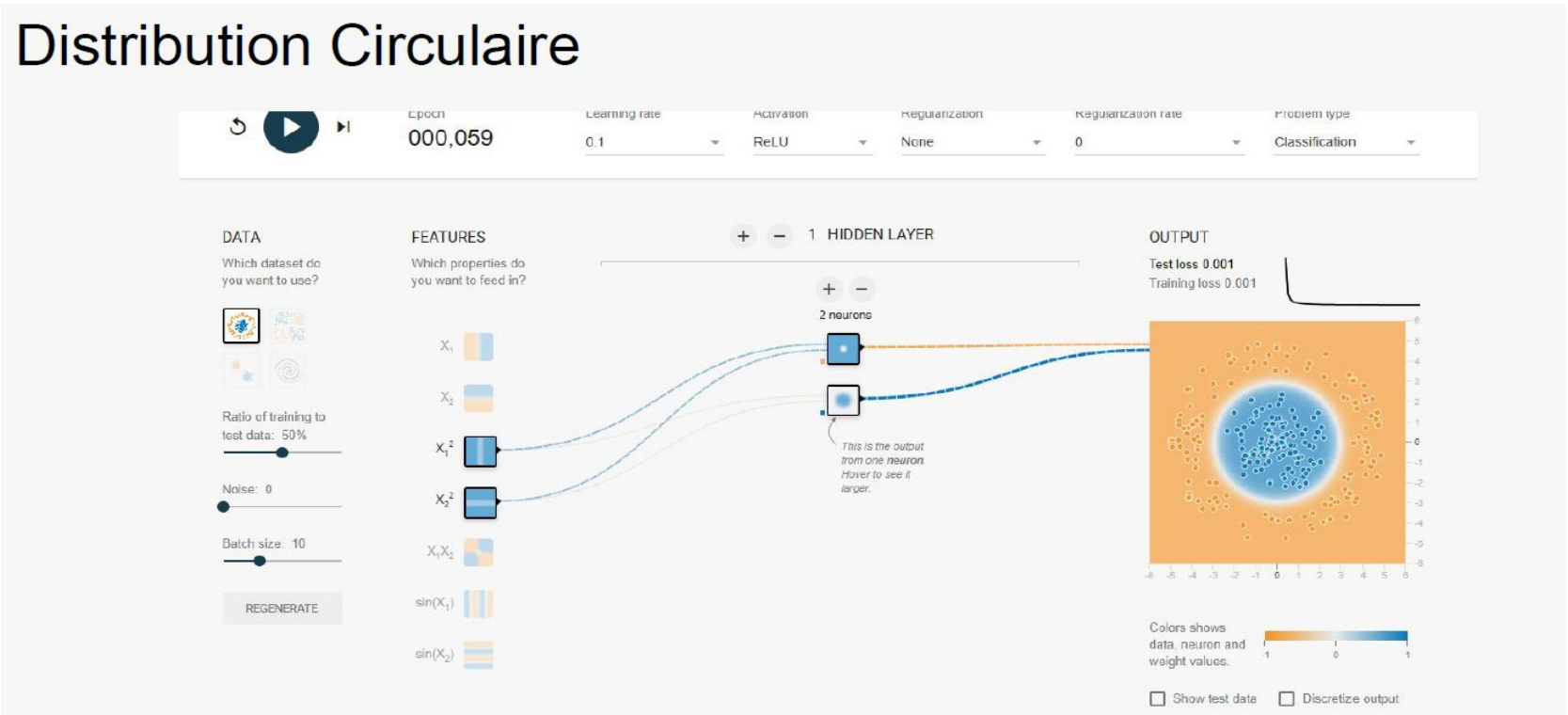
Distribution Gaussienne



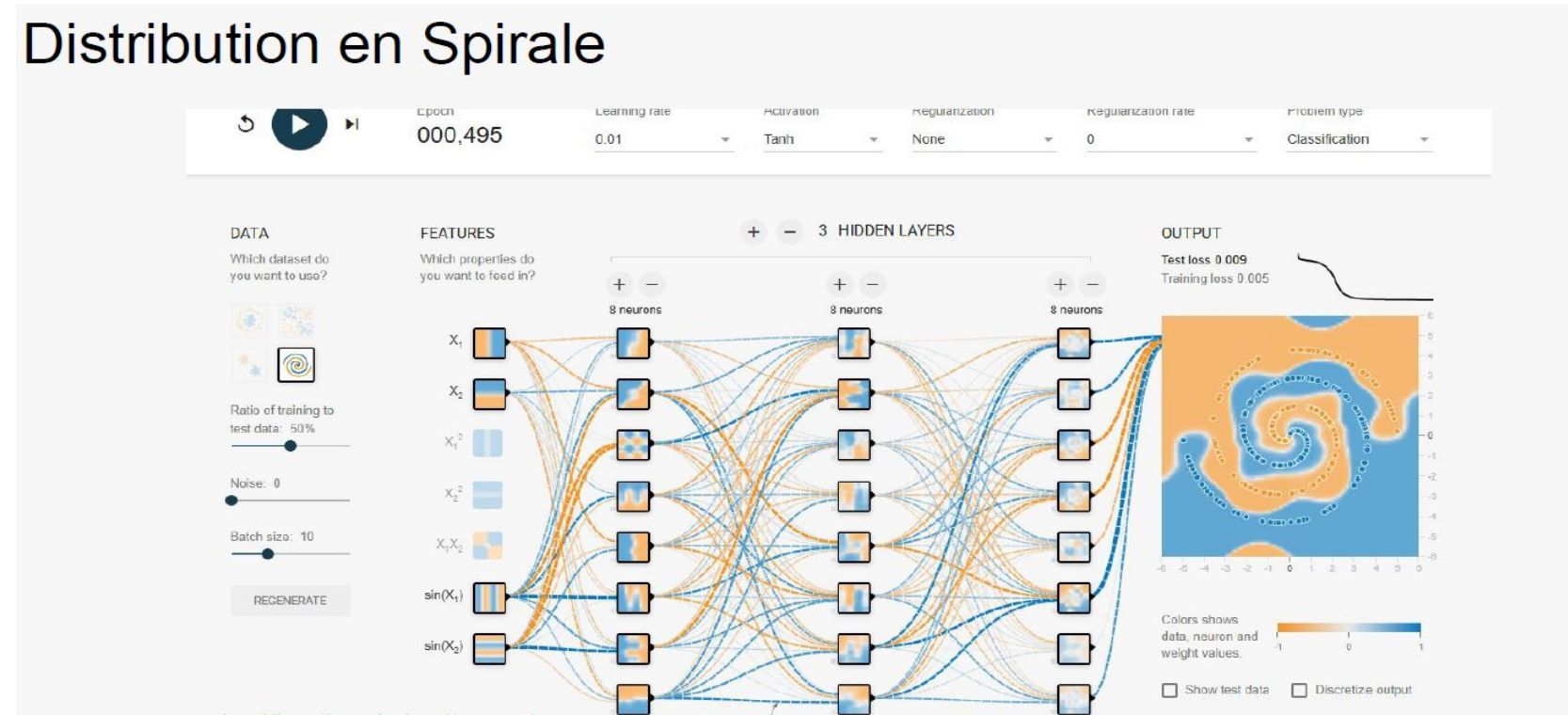
Distribution Ou Exclusif



Distribution Circulaire



Distribution en Spirale



Implémentations du MLP en POO

1. Simple descente de gradient, valeur par valeur
2. Gradient moyen, calcul donc plus rapide
3. Méthode de gradient adaptative

On note $\Theta = (\Theta_j) = \begin{pmatrix} w_{ij} & \text{pour tous les } i, j \\ b_j & \text{pour tous les } j \end{pmatrix}$

On définit $G_{j,k}$ par :

$$G_{j,k} = G_{j,k-1} + \left(\frac{\partial E}{\partial \Theta_j}(x, y, \Theta^k) \right)^2$$

$G_{j,k-1}$ contient donc la somme de tous les gradients au carré des précédentes itérations.

Puis on met à jour Θ_j par la formule :

$$\Theta_j^{k+1} = \Theta_j^k - \frac{\eta}{\sqrt{G_{j,k}} + \varepsilon} \frac{\partial E}{\partial \Theta_j}(x, y, \Theta^k)$$

Implémentations du MLP en POO

4. Méthode RMSPro (amélioration du gradient adaptatif)

Dans ce cas, $G_{j,k}$ est défini par :

$$G_{j,k} = (1 - \gamma) G_{j,k-1} + \gamma \left(\frac{\partial E}{\partial \Theta_j}(x, y, \Theta^k) \right)^2$$

où le paramètre γ est tel que $0 \leq \gamma < 1$

Avec cette formule l'importance des gradients des itérations précédentes décroît exponentiellement avec le nombre d'itérations. Ainsi, le terme $\eta / \sqrt{G_{j,k}} + \varepsilon$ ne va pas diminuer trop et le processus d'apprentissage ne s'arrêtera pas prématurément.

Θ_j est toujours mis à jour par la formule :

$$\Theta_j^{k+1} = \Theta_j^k - \frac{\eta}{\sqrt{G_{j,k}} + \varepsilon} \frac{\partial E}{\partial \Theta_j}(x, y, \Theta^k)$$

Implémentations du MLP en POO

5. Méthode AdaDelta (élimine la nécessité de définir un pas d'apprentissage)

$G_{j,k}$ est toujours défini par :

$$G_{j,k} = (1 - \gamma)G_{j,k-1} + \gamma \left(\frac{\partial E}{\partial \theta_j}(x, y, \theta^k) \right)^2$$

où le paramètre γ est tel que $0 \leq \gamma < 1$

Mais on rajoute le calcul d'un paramètre Δ_j défini par :

$$\Delta_{j,k} = (1 - \gamma)\Delta_{j,k-1} + \gamma \left(\frac{\sqrt{\Delta_{j,k-1}} + \varepsilon}{\sqrt{G_{j,k-1}} + \varepsilon} \frac{\partial E}{\partial \theta_j}(x, y, \theta^k) \right)^2$$

Ce nouveau paramètre va intervenir dans la mise à jour de θ_j suivant la formule :

$$\theta_j^{k+1} = \theta_j^k - \frac{\sqrt{\Delta_{j,k}} + \varepsilon}{\sqrt{G_{j,k}} + \varepsilon} \frac{\partial E}{\partial \theta_j}(x, y, \theta^k)$$

Application avec la base de données

Predict Students' Dropout and Academic Success

Dataset

- 4424 observations; 36 features (marital status, father's qualification, previous qualification grade, etc);
- 01 target: graduate, dropout ou enrolled;
- pas de valeurs manquantes, ni de doublons;
- split train test
- normalisation des données
- dataset non équilibré: $\frac{2}{3}$ - $\frac{1}{3}$ pour les classes graduate et dropout → rééquilibrage avec SMOTE

Analyse exploratoire

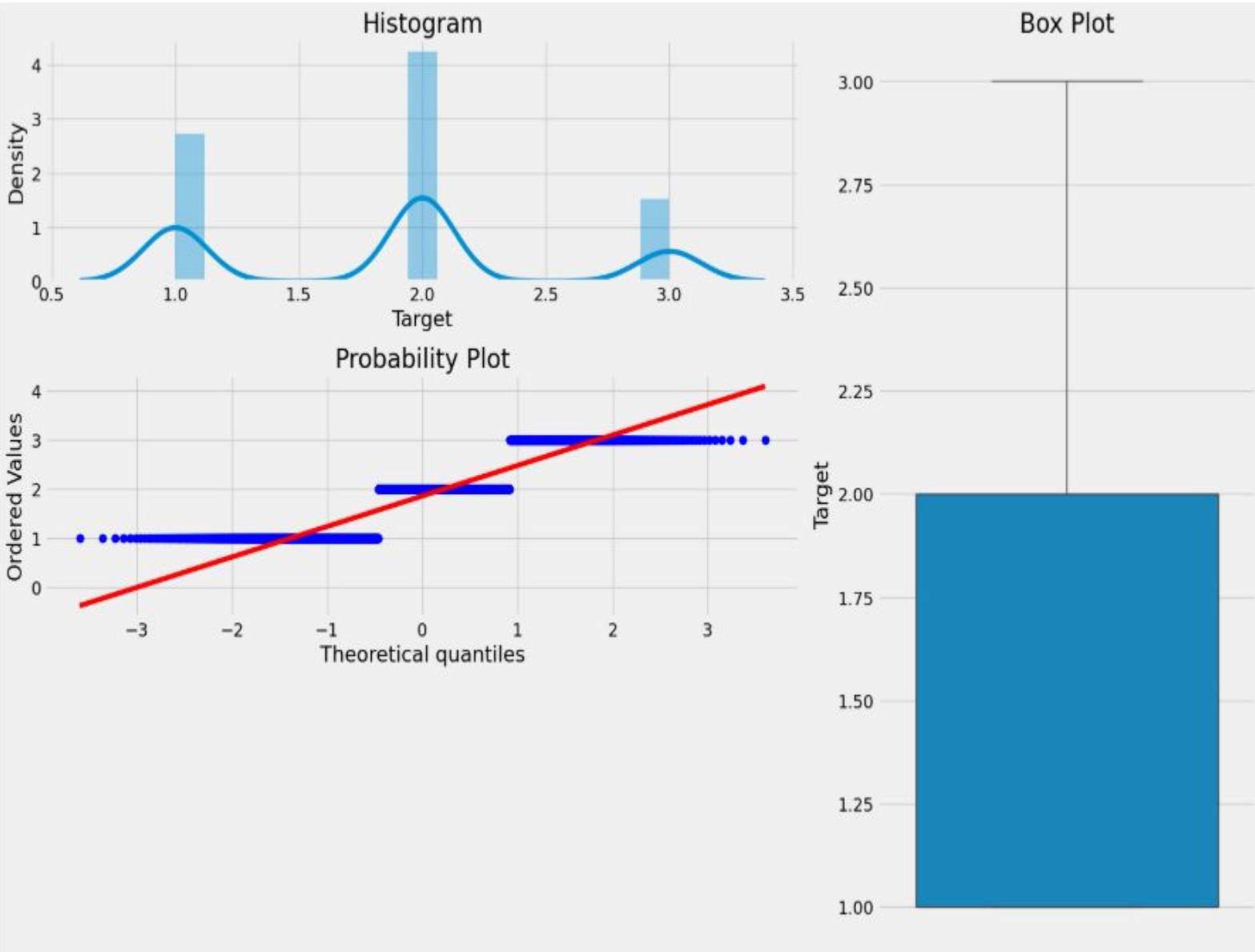
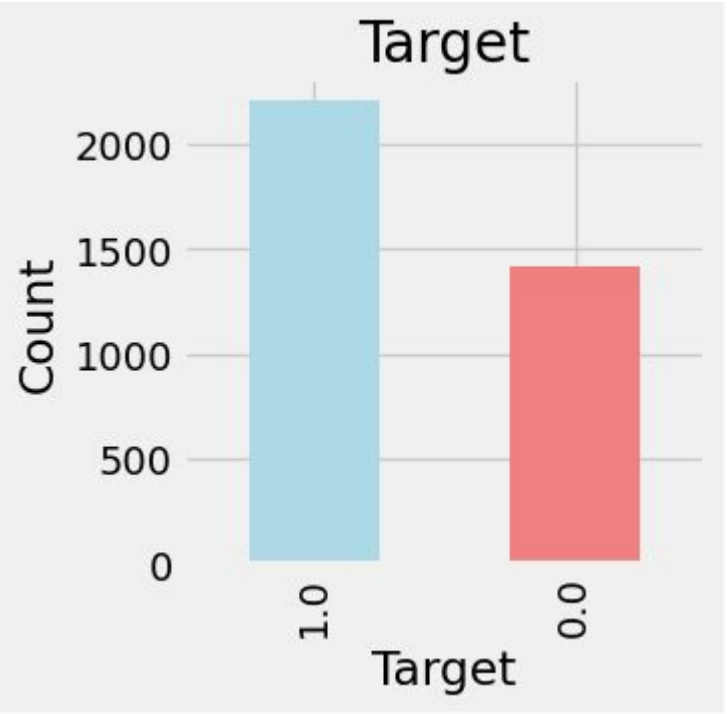
RangeIndex: 4424 entries, 0 to 4423

Data columns (total 37 columns):

#	Column	Non-Null Count	Dtype
0	Marital status	4424 non-null	int64
1	Application mode	4424 non-null	int64
2	Application order	4424 non-null	int64
3	Course	4424 non-null	int64
4	Daytime/evening attendance	4424 non-null	int64
5	Previous qualification	4424 non-null	int64
6	Previous qualification (grade)	4424 non-null	float64
7	Nacionality	4424 non-null	int64
8	Mother's qualification	4424 non-null	int64
9	Father's qualification	4424 non-null	int64
10	Mother's occupation	4424 non-null	int64
11	Father's occupation	4424 non-null	int64
12	Admission grade	4424 non-null	float64
13	Displaced	4424 non-null	int64
14	Educational special needs	4424 non-null	int64
15	Debtor	4424 non-null	int64
16	Tuition fees up to date	4424 non-null	int64
17	Gender	4424 non-null	int64
18	Scholarship holder	4424 non-null	int64
19	Age at enrollment	4424 non-null	int64
20	International	4424 non-null	int64
21	Curricular units 1st sem (credited)	4424 non-null	int64
22	Curricular units 1st sem (enrolled)	4424 non-null	int64
23	Curricular units 1st sem (evaluations)	4424 non-null	int64
24	Curricular units 1st sem (approved)	4424 non-null	int64
25	Curricular units 1st sem (grade)	4424 non-null	float64
26	Curricular units 1st sem (without evaluations)	4424 non-null	int64
27	Curricular units 2nd sem (credited)	4424 non-null	int64
28	Curricular units 2nd sem (enrolled)	4424 non-null	int64
29	Curricular units 2nd sem (evaluations)	4424 non-null	int64
30	Curricular units 2nd sem (approved)	4424 non-null	int64
31	Curricular units 2nd sem (grade)	4424 non-null	float64
32	Curricular units 2nd sem (without evaluations)	4424 non-null	int64
33	Unemployment rate	4424 non-null	float64
34	Inflation rate	4424 non-null	float64
35	GDP	4424 non-null	float64
36	Target	4424 non-null	object

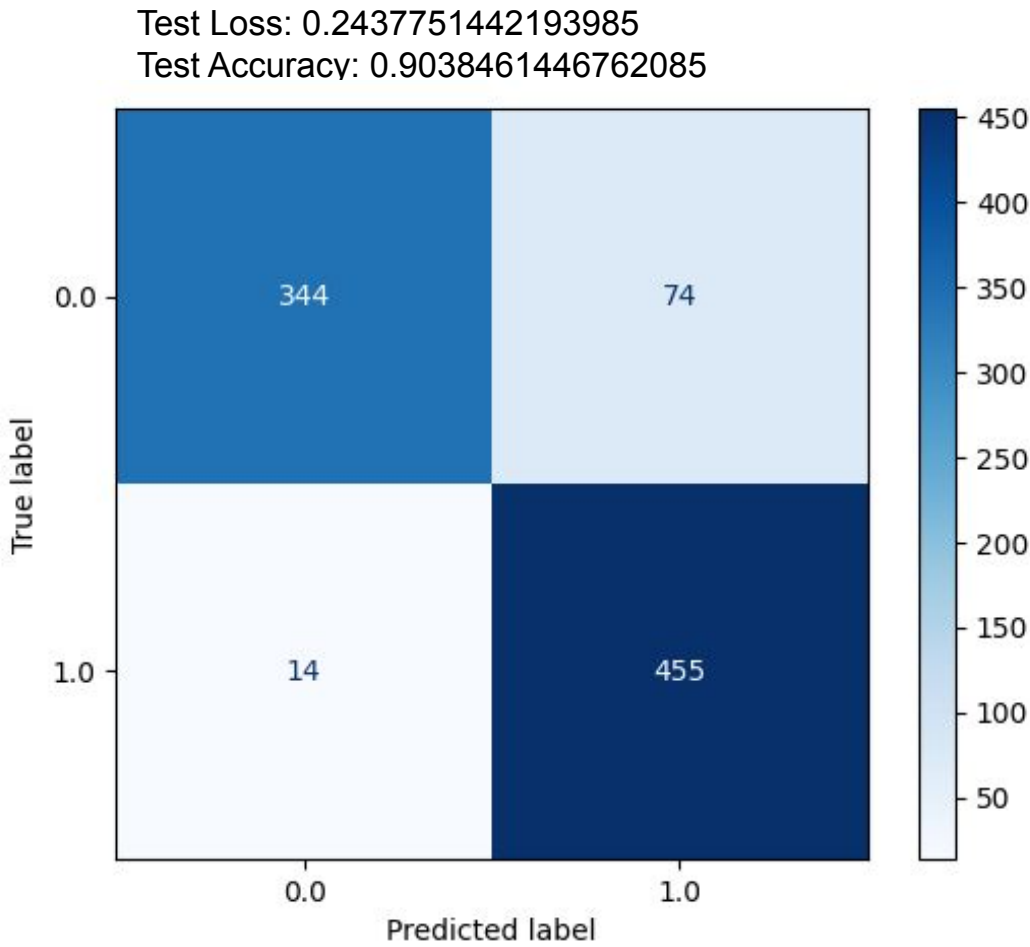
dtypes: float64(7), int64(29), object(1)

memory usage: 1.2+ MB



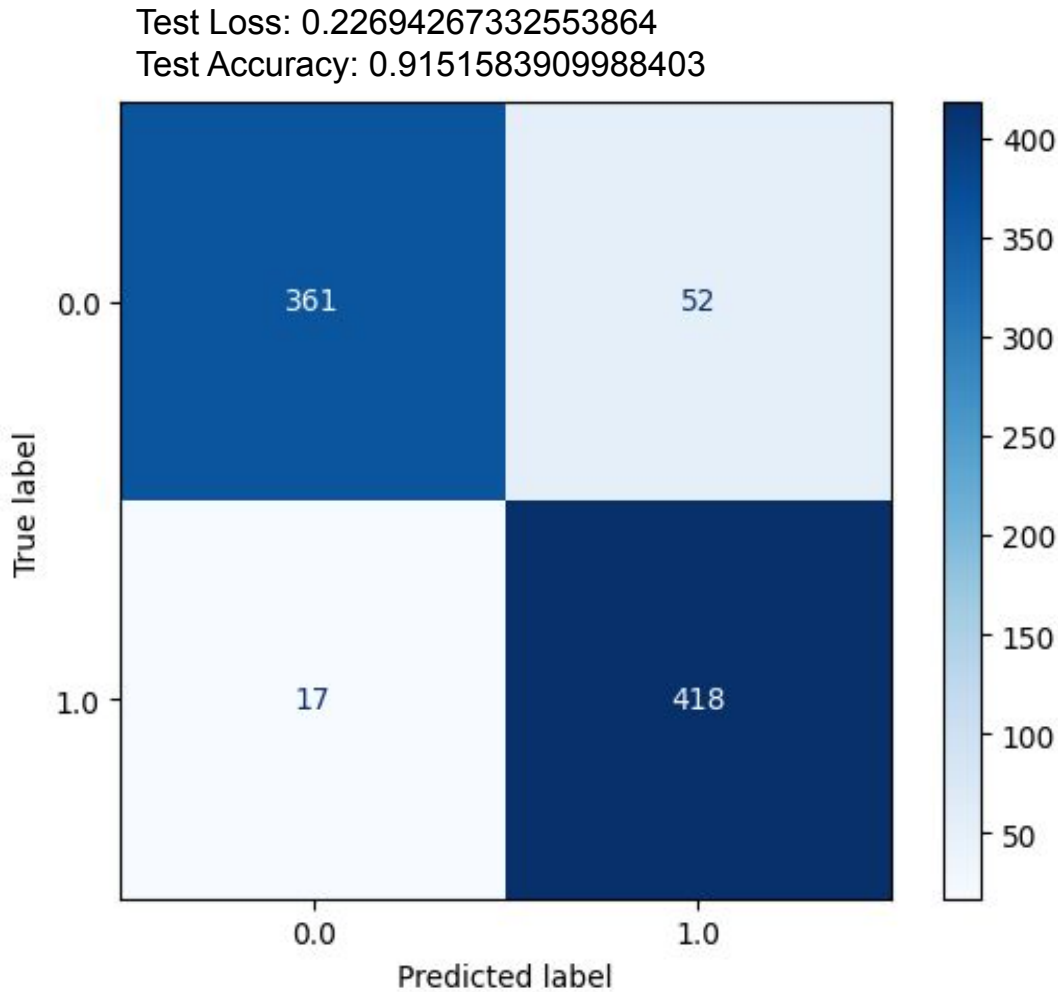
Tests de classification

DNN:



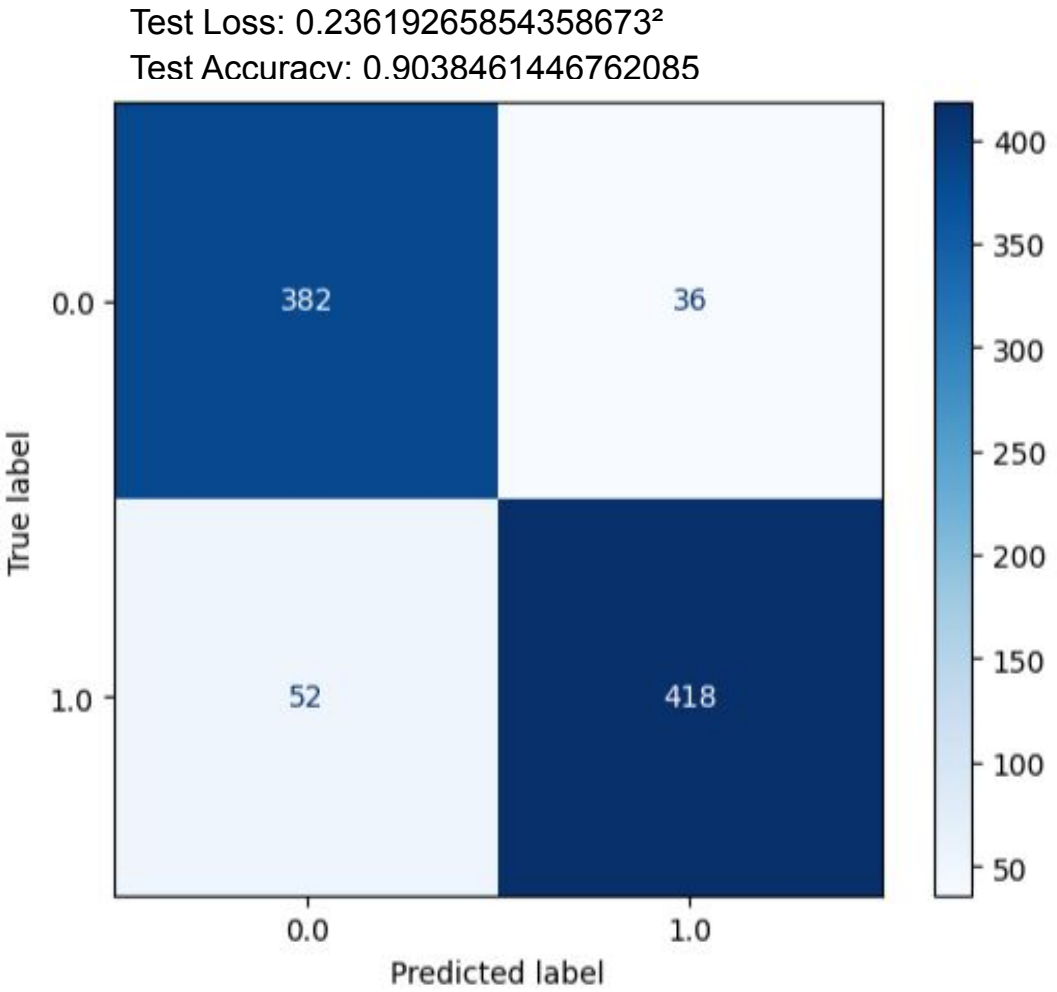
	precision	recall	f1-score	support
0.0	0.97	0.82	0.89	420
1.0	0.86	0.98	0.91	464
accuracy			0.90	884
macro avg	0.91	0.90	0.90	884
weighted avg	0.91	0.90	0.90	884

CNN:



	precision	recall	f1-score	support
0.0	0.96	0.86	0.91	420
1.0	0.89	0.96	0.92	464
accuracy			0.92	884
macro avg	0.92	0.91	0.91	884
weighted avg	0.92	0.92	0.91	884

LSTM:

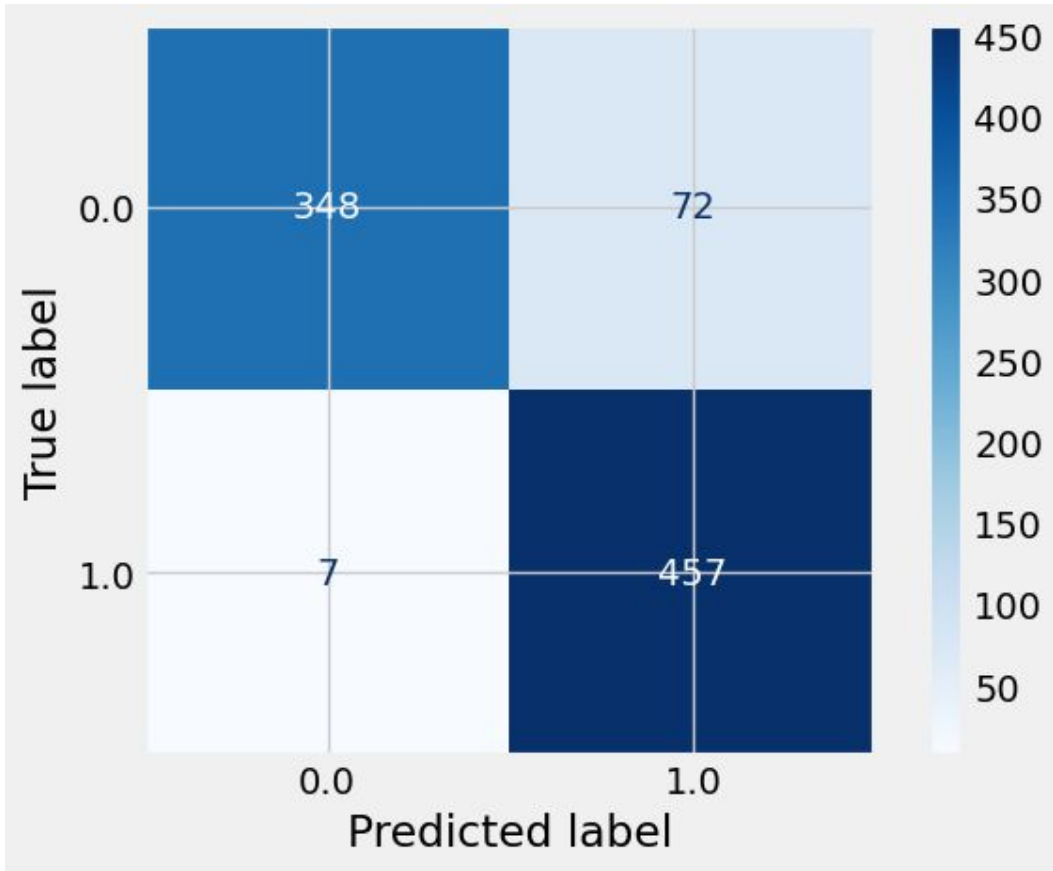


	precision	recall	f1-score	support
0.0	0.89	0.91	0.90	420
1.0	0.92	0.90	0.91	464
accuracy			0.90	884
macro avg	0.90	0.90	0.90	884
weighted avg	0.90	0.90	0.90	884

Tests de classification

Sequential:

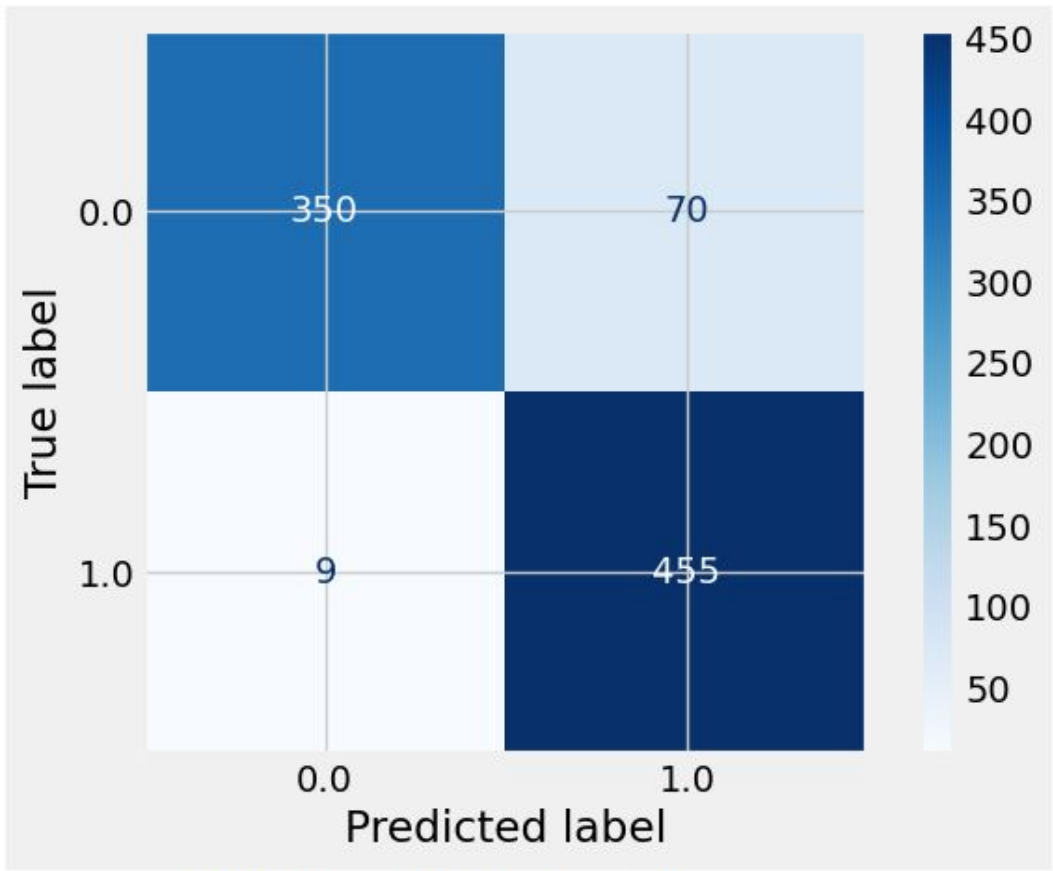
Test Loss: 0.23132942616939545
Test Accuracy: 0.9106335043907166



	precision	recall	f1-score	support
0.0	0.98	0.83	0.90	420
1.0	0.86	0.98	0.92	464
accuracy			0.91	884
macro avg	0.92	0.91	0.91	884
weighted avg	0.92	0.91	0.91	884

Sequential dropout:

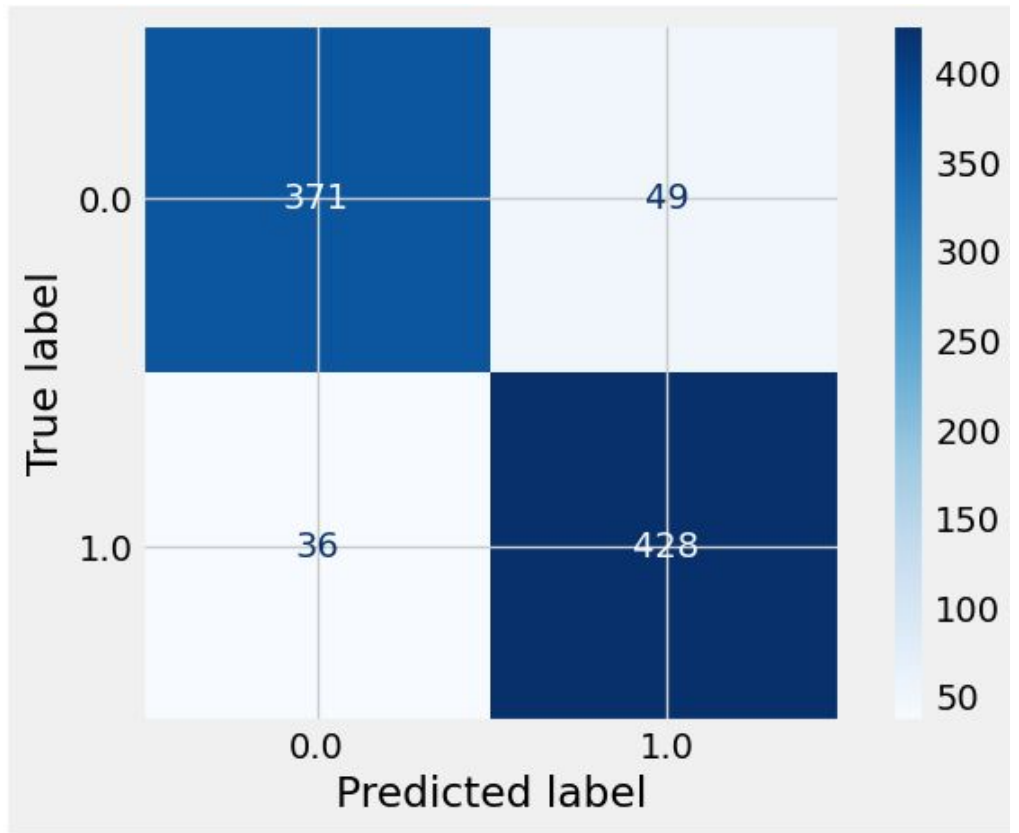
Test Loss: 0.24920935928821564
Test Accuracy: 0.9106335043907166



	precision	recall	f1-score	support
0.0	0.97	0.83	0.90	420
1.0	0.87	0.98	0.92	464
accuracy			0.91	884
macro avg	0.92	0.91	0.91	884
weighted avg	0.92	0.91	0.91	884

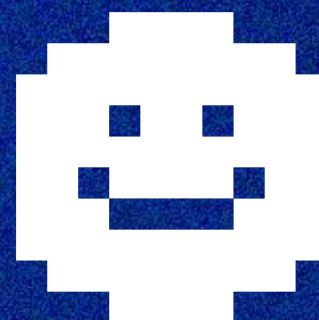
Sequential batch normalisation:

Test Loss: 0.23058392107486725
Test Accuracy: 0.9038461446762085



	precision	recall	f1-score	support
0.0	0.91	0.88	0.90	420
1.0	0.90	0.92	0.91	464
accuracy			0.90	884
macro avg	0.90	0.90	0.90	884
weighted avg	0.90	0.90	0.90	884

Conclusion



 **La Plateforme**

contact@laplateforme.io

8 rue d'Hozier 13002 Marseille

04.84.89.43.69