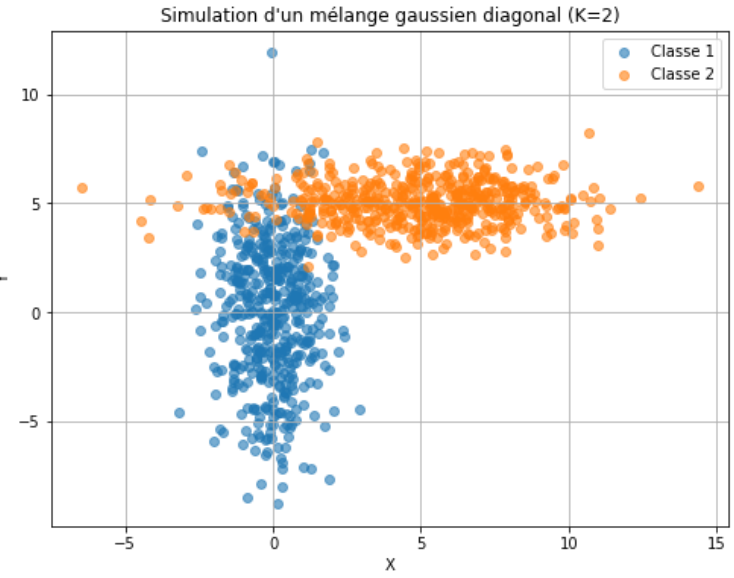
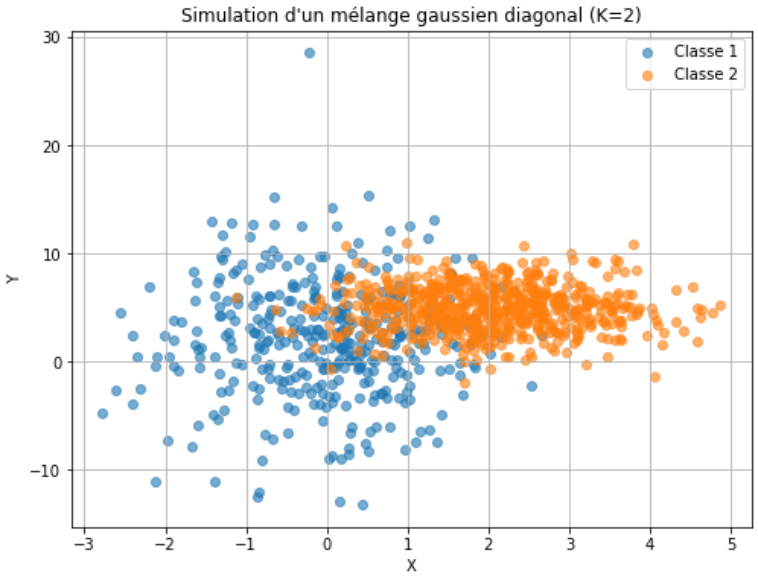
TP 3 Machine Learning – Logan RENAUD, William ROCHE, Ethan TRENTIN

1. Simulation d'un modèle de mélange gaussien diagonal

n = 1000
proportions = [0.5, 0.5]
means = [[0, 0], [5, 5]]
stds = [[1, 3], [3, 1]]



n = 1000
proportions = [0.4, 0.6]
means = [[0, 2], [2, 5]]
sds = [[1, 5], [1, 2]]



2. Algorithme des centres-mobiles

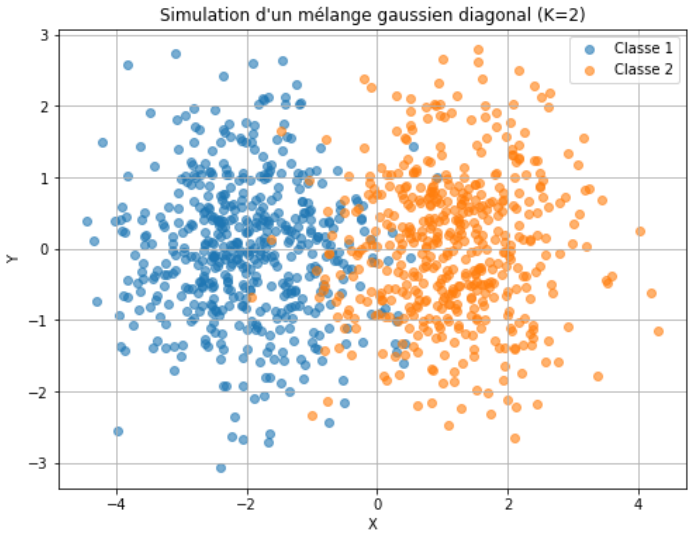
```python
def k_means(data, k, max_iters=100, tol=1e-4):
    if not isinstance(data, np.ndarray):
        data = data.to_numpy() if hasattr(data, "to_numpy") else np.array(data)
    n_samples= data.shape[0]
    centroids = data[np.random.choice(n_samples, k, replace=False)]
    iteration = 0
    norm = math.inf
    while iteration < max_iters and norm > tol:
        distances = np.linalg.norm(data[:, np.newaxis] - centroids, axis=-1)
        labels = np.argmin(distances, axis=1)
        new_centroids = np.array([data[labels == i].mean(axis=0) for i in range(k)])
        norm = np.linalg.norm(new_centroids - centroids)
        centroids = new_centroids
        iteration+=1

    return centroids, labels
```

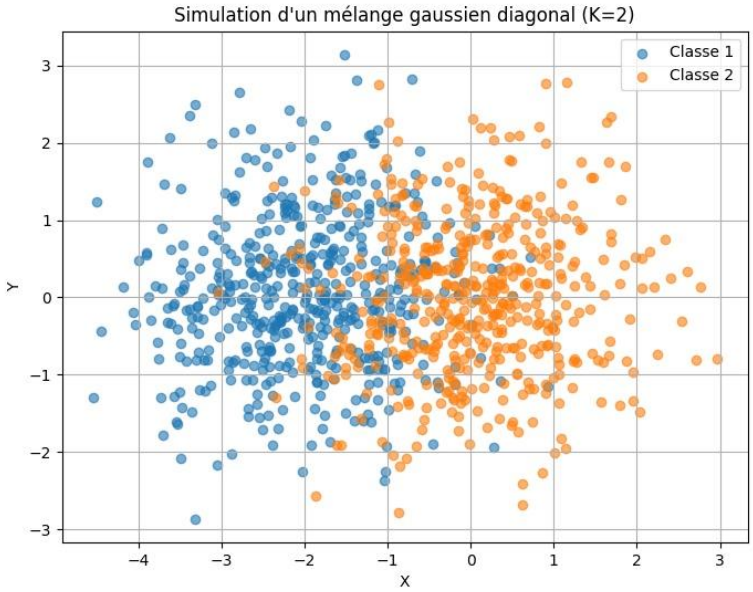
TP 3 Machine Learning – Logan RENAUD, William ROCHE, Ethan TRENTIN

3. Génération des jeux de données

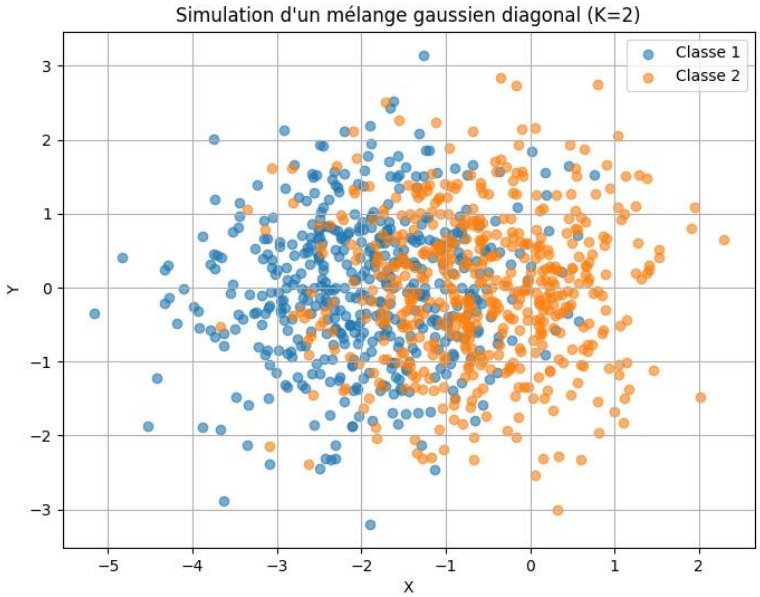
Degré d'erreur de 6% :



Degré d'erreur de 16%

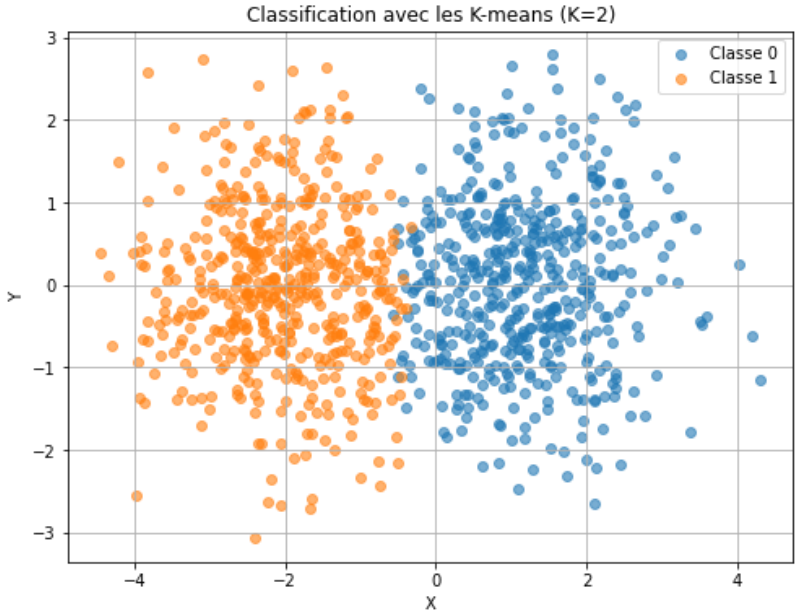


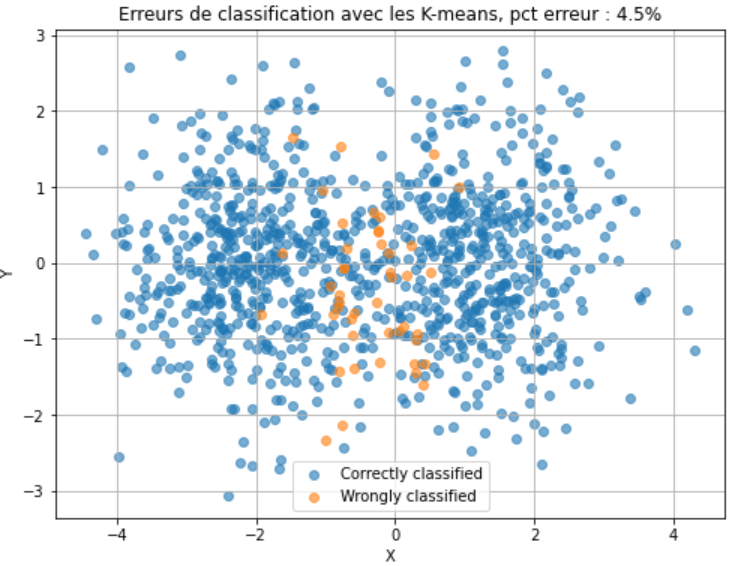
Degré d'erreur de 26%

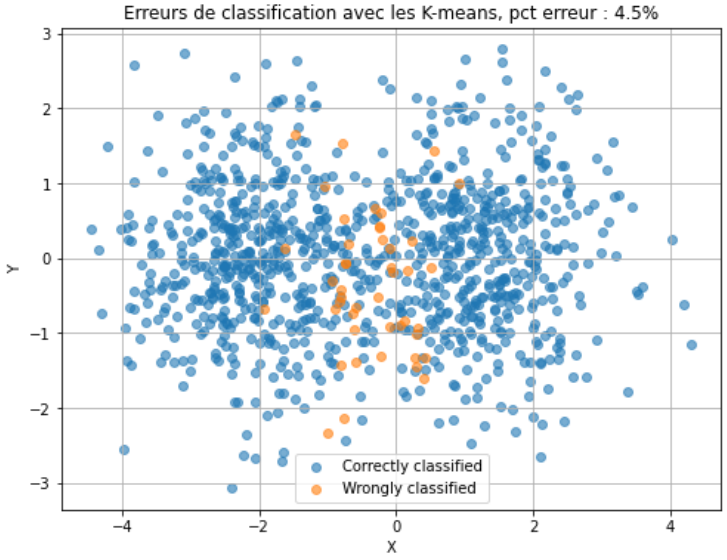


TP 3 Machine Learning – Logan RENAUD, William ROCHE, Ethan TRENTIN

4. Résultats sur le premier dataset





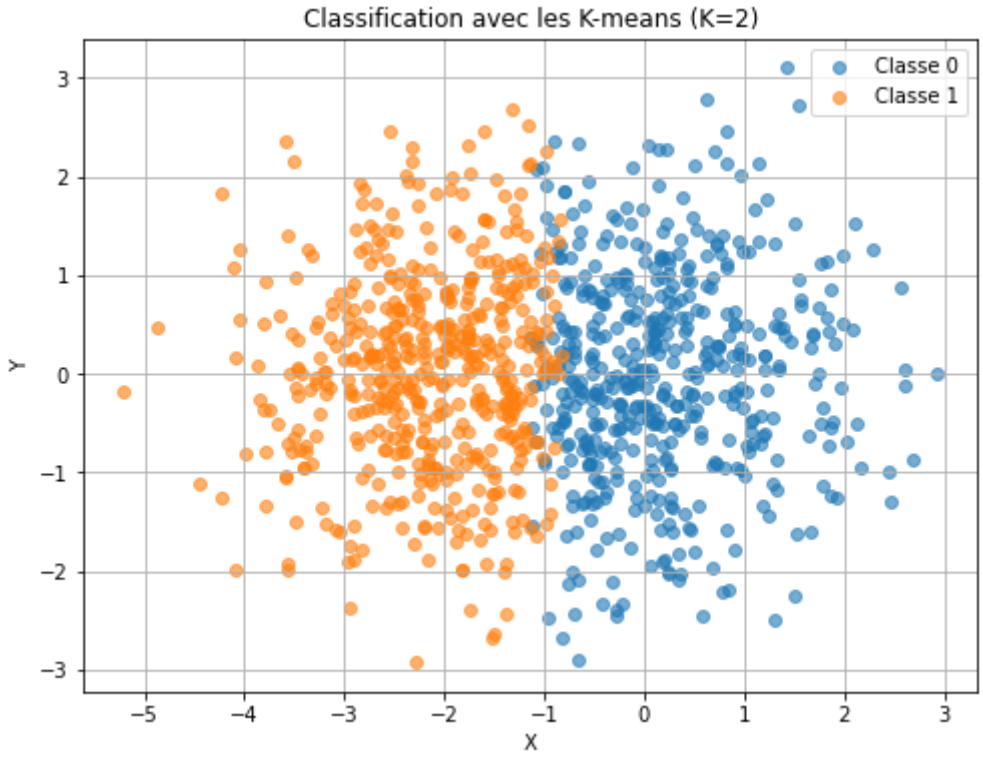


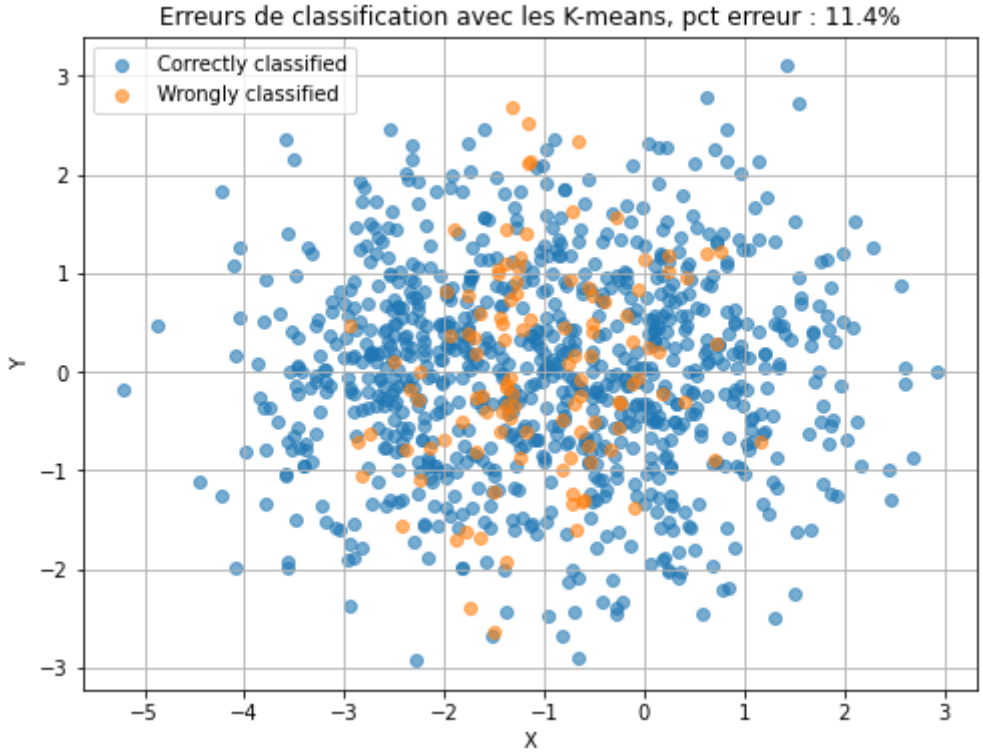
On peut voir dans les résultats pour 2 applications de notre algorithme qu'il n'y a pas besoin d'améliorer l'initialisation puisque le résultat est correct et est toujours le même. On pourrait améliorer l'initialisation des centroïdes dans le cas K =2 en initialisant avec deux points éloignés, pour améliorer la vitesse de calcul on pourrait simplement prendre les 2 points extrêmes dans une coordonnée (aléatoirement x ou y). Faire cela permettrait d'accélérer la convergence des centroïdes.

TP 3 Machine Learning – Logan RENAUD, William ROCHE, Ethan TRENTIN

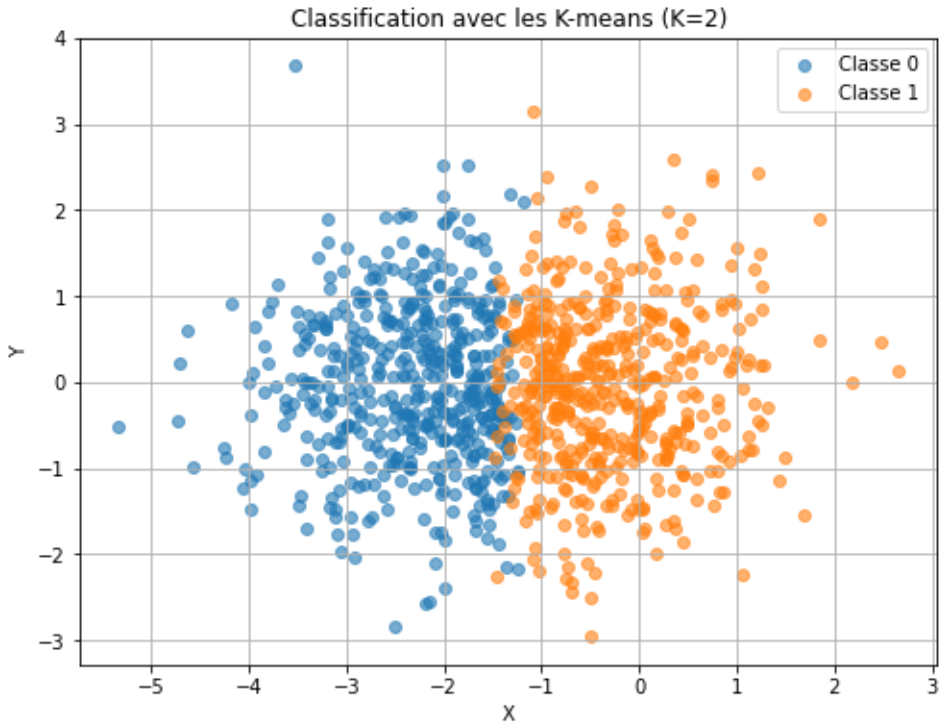
5. Résultats sur les 2 autres datasets

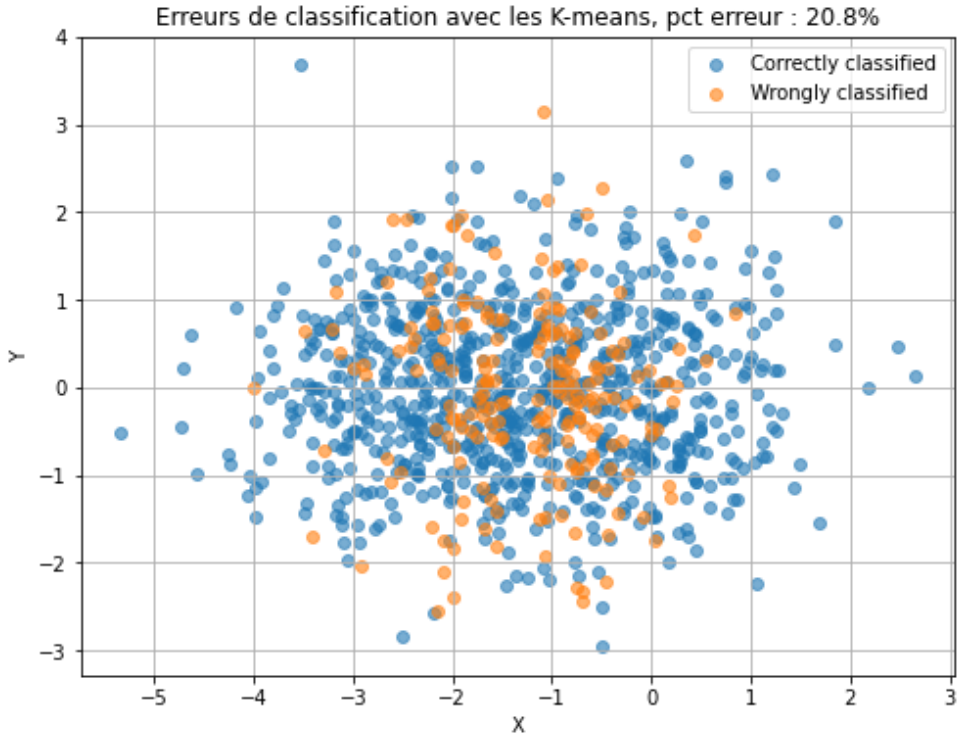
Résultats dataset 2 :





Résultat dataset 3 :





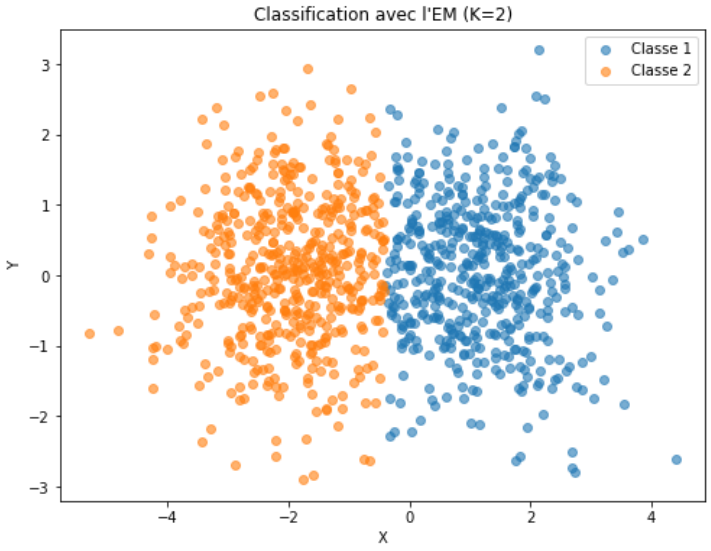
Le pourcentage de points mal classés augmente avec le degré de mélange, ce à quoi on s'attendait.

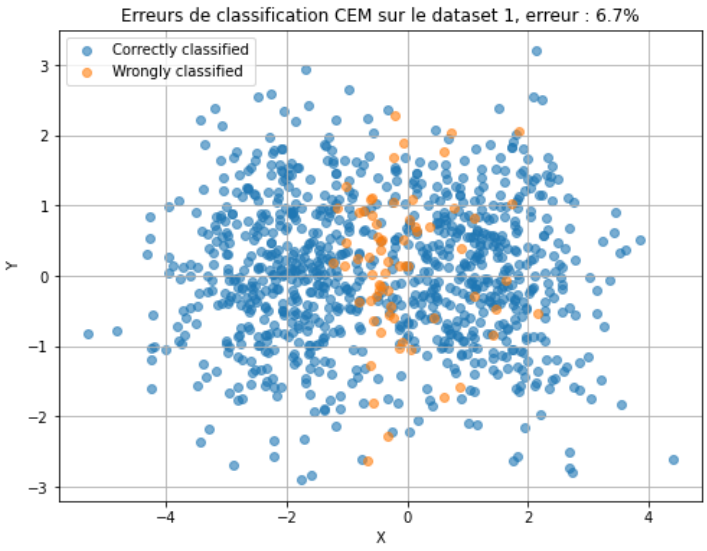
6. <u>Algorithme CEM adapté aux trois jeux de données</u>

On choisit un modèle gaussien parcimonieux diagonal (car chaque cluster a une matrice de covariance diagonale).

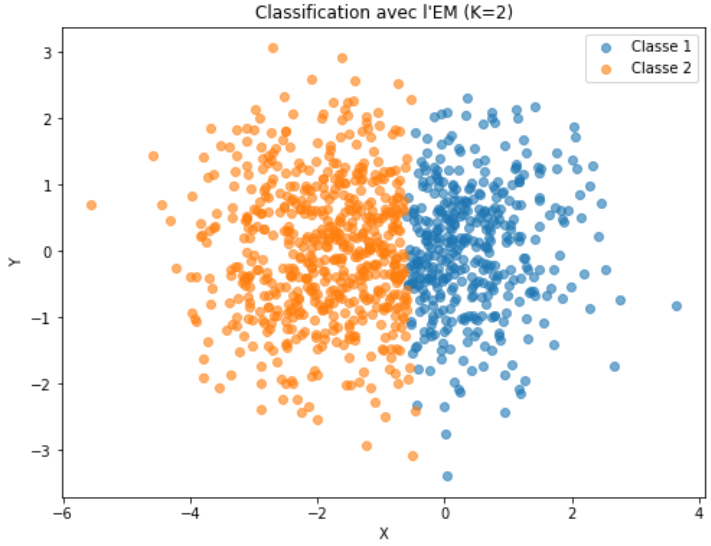
TP 3 Machine Learning – Logan RENAUD, William ROCHE, Ethan TRENTIN

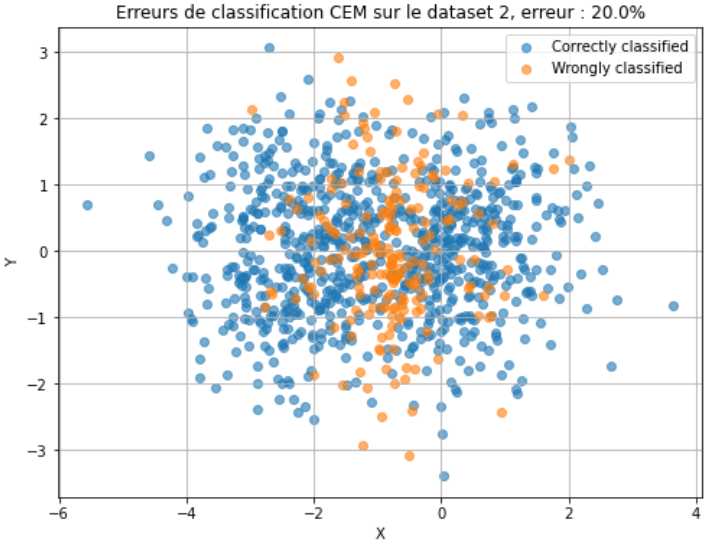
Premier dataset :



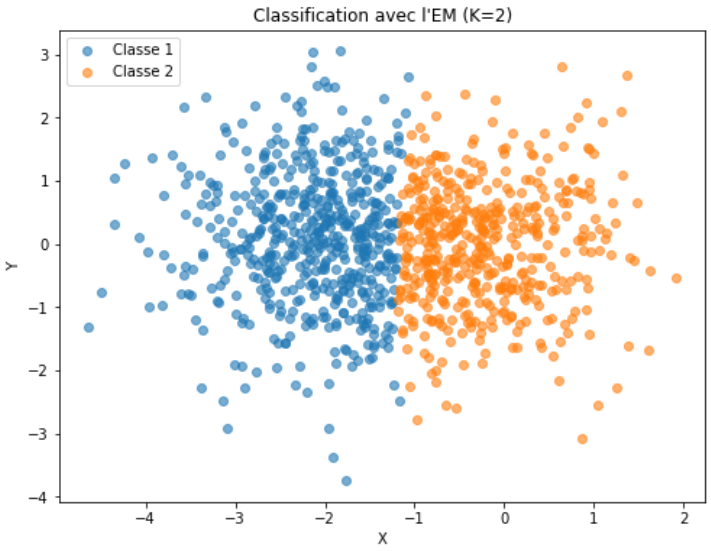


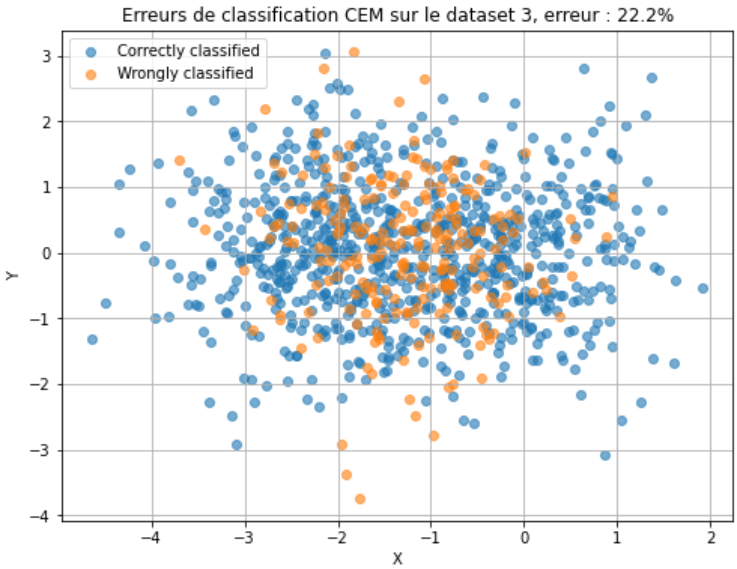
Second dataset :





Troisième dataset :





TP 3 Machine Learning – Logan RENAUD, William ROCHE, Ethan TRENTIN

La valeur de la vraisemblance maximisée est :

$$L_C(P, \boldsymbol{\theta}) = -\frac{1}{2}\left(np + np\log\left(\frac{\text{trace}(S_W)}{p}\right)\right) - n\log(g) + \frac{np}{2}\log(2\pi)$$

Le modèle CEM ne parvient pas à de meilleurs résultats que l'algorithme des K-means dans notre cas car ici, comme présenté dans le cours, l'algorithme fonctionne comme les K-means.