Introduction à l'Apprentissage Artificiel

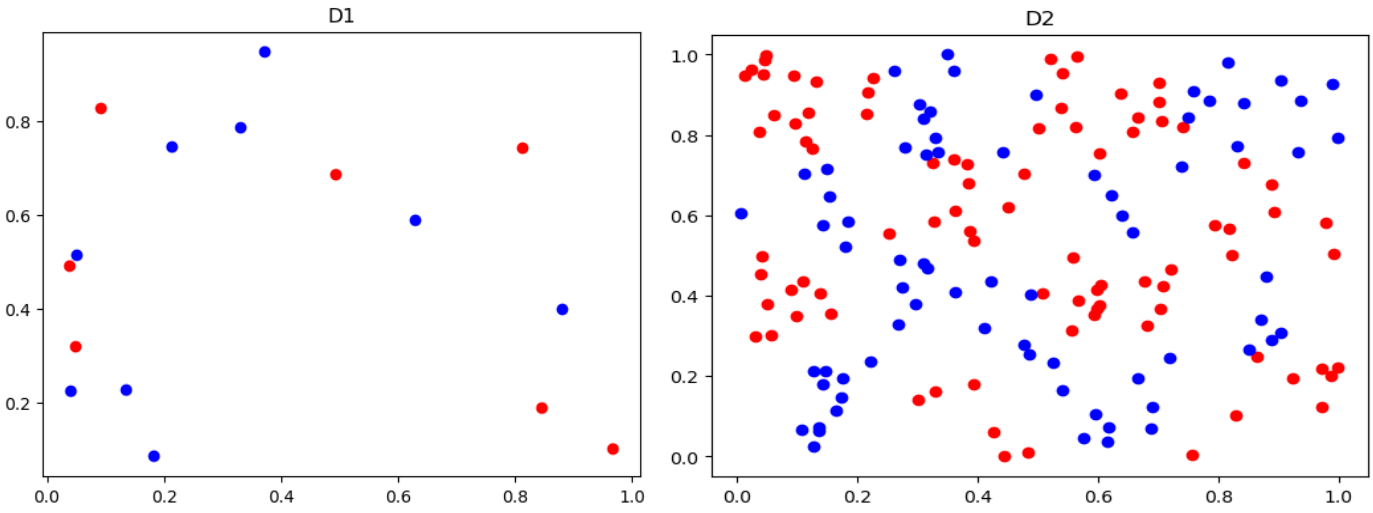
TDM1

- **Etude k-ppv sur des données en damier :**
  1. **Quand la dimension augmente avec un nombre constant d'exemples :**
     - **1)**
     - **2)**

Le programme génère des échantillons aléatoires de données à **d** dimensions, où **d** varie de **1** à **20**. Pour chaque valeur de **d**, il génère **10** ensembles de **100** échantillons de données aléatoires à **d** dimensions et calcule la distance moyenne des points de données par rapport au centre de l'espace de **d** dimensions et la distance moyenne de voisins les plus proches du centre.

Le résultat de l'exécution du programme est une série de paires de valeurs, représentant la distance moyenne et la distance moyenne de voisins les plus proches pour chaque valeur de **d**. En général, la distance moyenne diminue à mesure que **d** augmente, car les points de données ont plus d'espace pour se répartir.

  2. **Données en damier :**
     - **1)**



     - **2)**

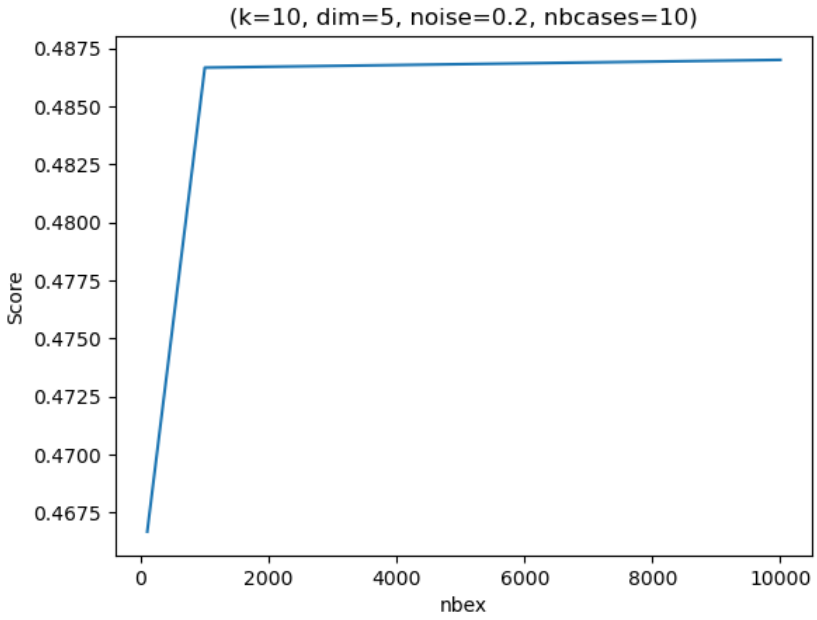
Score du KNN sur D1 : 0.68
Score du KNN sur D2 : 0.79

Le score obtenu pour le damier **D1** est plus faible que celui pour le damier **D2** car le nombre de points dans **D1** est très faible par rapport à celui de **D2**, ce qui peut

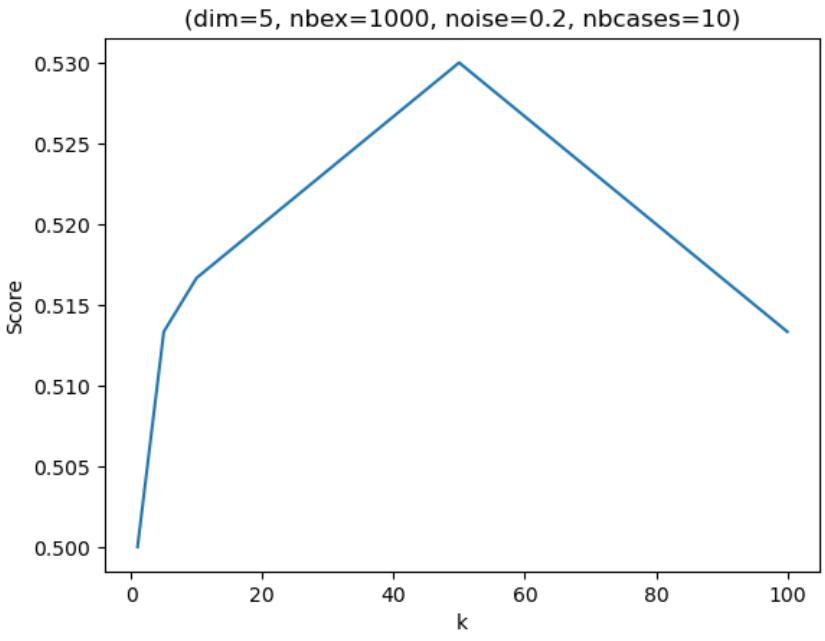
affecter la capacité du classifieur **KNN** à apprendre des modèles complexes et à généraliser sur de nouvelles données.

Dans l'affichage des damiers, la séparation entre les points des deux classes est plus claire et nette dans **D2** par rapport à **D1**, où les points semblent être plus dispersés et moins séparés.

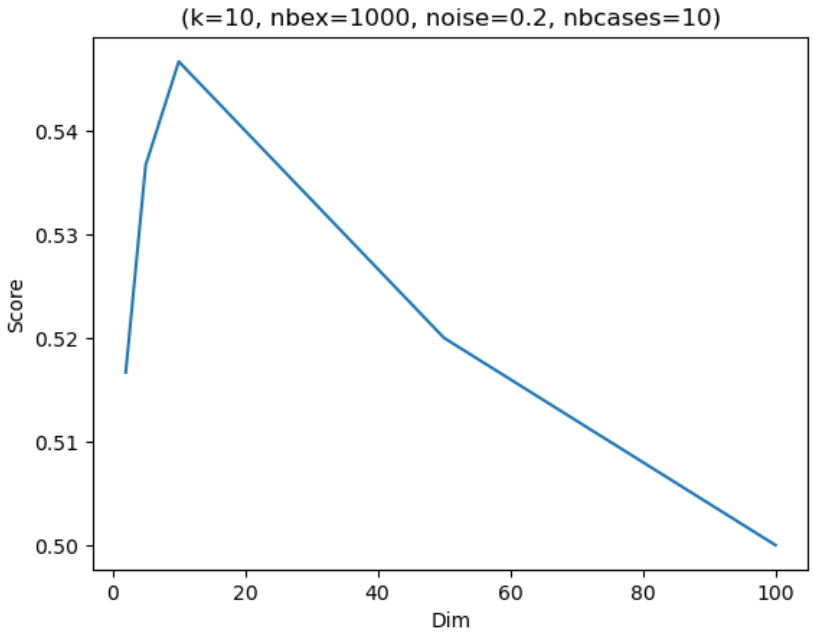
3)



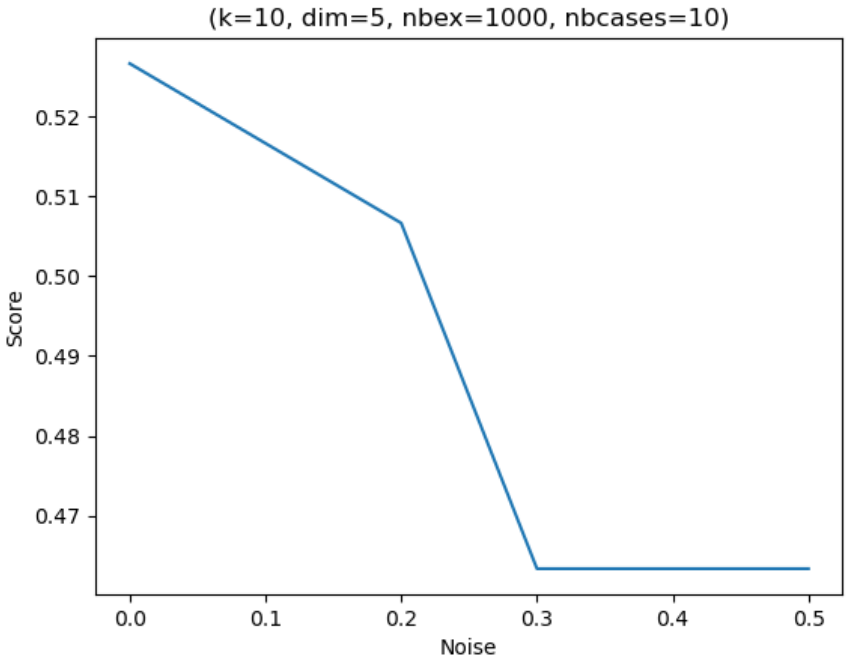
En variant le nombre d'exemples, on constate que si le nombre d'exemples augmente, la performance du modèle s'améliore jusqu'à un certain point il stagne, car avec plus d'exemples, le classificateur **KNN** a plus d'informations à tirer et peut faire de meilleures prédictions.



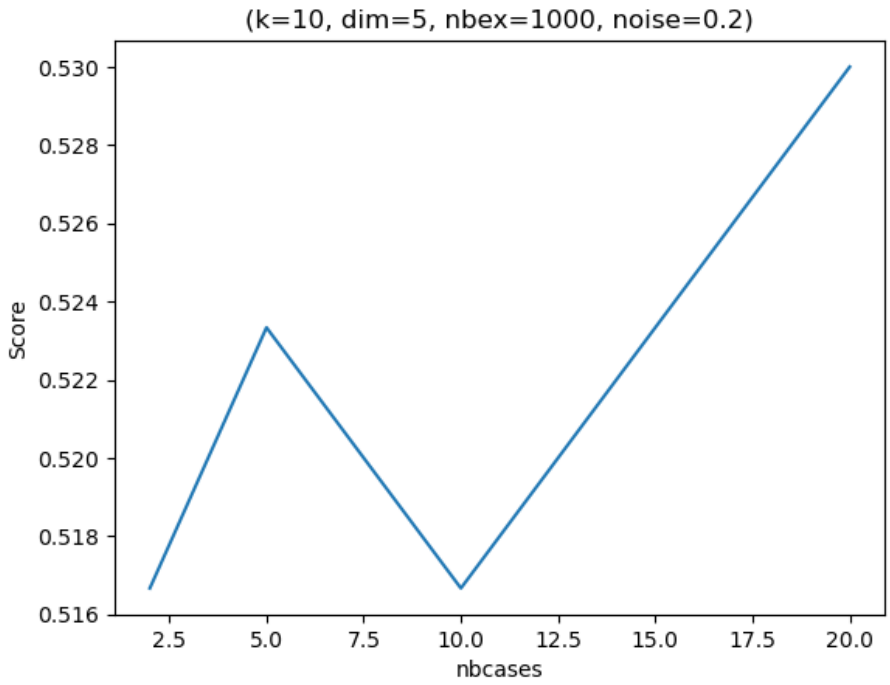
En variant le k, on constate que les valeurs plus grandes de k ont tendance à lisser la limite de décision et à réduire la précision, tandis que des valeurs plus petites de k ont tendance à augmenter et à donner une plus grande précision. Cependant, la valeur optimale de k dépend de la complexité du problème, de la quantité de bruit et du nombre d'exemples.



En variant la dimension, on constate si la dimension augmente, la performance de notre modèle se diminue, car avec l'augmentation de la dimension les exemples d'apprentissage seront plus éloignés les uns des autres, ce qui rend plus difficile de trouver des voisins proches et fiables pour effectuer une classification précise.



En variant le bruit, on constate que la performance de notre modèle se diminue avec l'augmentation du bruit, car avec l'augmentation du bruit peut rendre les frontières de décision plus floues et moins distinctes.



En variant le nombre de cases, on observe que la performance du modèle varie d'un domaine à l'autre, avec une amélioration suivie d'une détérioration. Ceci suggère une interdépendance entre le choix du nombre de cases et un autre hyperparamètre.

4)

dim	nb_cases	noise	nb_ex	best_k	score
2	2	0	100	5	0.9333333333333333
2	2	0	1000	1	0.9666666666666666
2	2	0	10000	1	0.9883333333333333
2	2	0.2	100	10	0.6
2	2	0.2	1000	50	0.7166666666666667
2	10	0	100	5	0.4333333333333333
2	10	0	1000	1	0.6799999999999999
2	10	0	10000	1	0.8939999999999999
2	10	0.2	1000	1	0.5533333333333333
2	10	0.2	10000	10	0.6813333333333335
2	10	0.5	100	1	0.6333333333333333
2	10	0.5	10000	1	0.4993333333333333
5	2	0	1000	1	0.65
5	2	0	10000	50	0.784
5	2	0.2	100	1	0.5333333333333333
5	2	0.2	1000	5	0.6033333333333333
5	2	0.2	10000	100	0.6116666666666667
5	2	0.5	100	5	0.33333333333333337
5	2	0.5	1000	10	0.4066666666666666
5	2	0.5	10000	100	0.509
5	5	0	100	1	0.5000000000000001
5	10	0.2	10000	500	0.4953333333333333
5	10	0.5	100	10	0.5666666666666667

```
5          10          0.5          10000          10          0.5196666666666667
10          2          0          100          5          0.7
10          2          0          1000          5          0.48
10          2          0          10000          1          0.5353333333333333
10          2          0.2          100          1          0.6333333333333333
10          2          0.2          1000          1          0.4333333333333334
10          2          0.2          10000          1          0.5163333333333334
10          2          0.5          1000          1          0.4333333333333334
10          2          0.5          10000          100          0.49366666666666664
10          10          0          1000          5          0.5599999999999999
10          10          0          10000          100          0.49866666666666665
10          10          0.2          100          1          0.5
10          10          0.2          10000          5          0.5109999999999999
10          10          0.5          100          1          0.6
10          10          0.5          10000          1          0.506
```

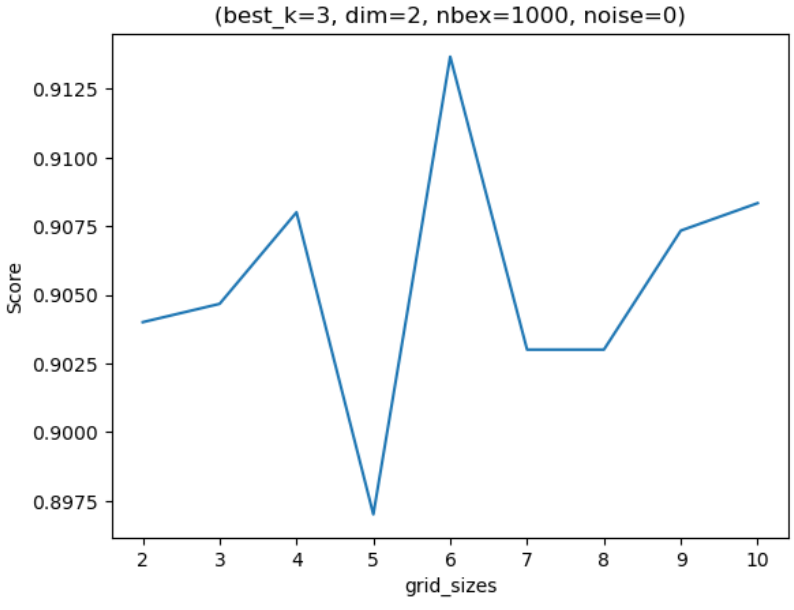
On constate que le score de classification tend à augmenter à mesure que le nombre d'exemples et le nombre de cases du damier augmentent. Cela peut être dû au fait que plus il y a d'exemples et de cases, plus il y a de chances d'avoir des points qui sont bien séparés entre les cases blanches et noires du damier, ce qui facilite la tâche de classification.

En revanche, le score diminue à mesure que le bruit augmente. Cela est dû au fait que plus il y a de bruit dans les étiquettes des points, plus la tâche de classification devient difficile, car les points qui auraient dû être classés dans une certaine catégorie peuvent être étiquetés à tort dans l'autre catégorie.

5)

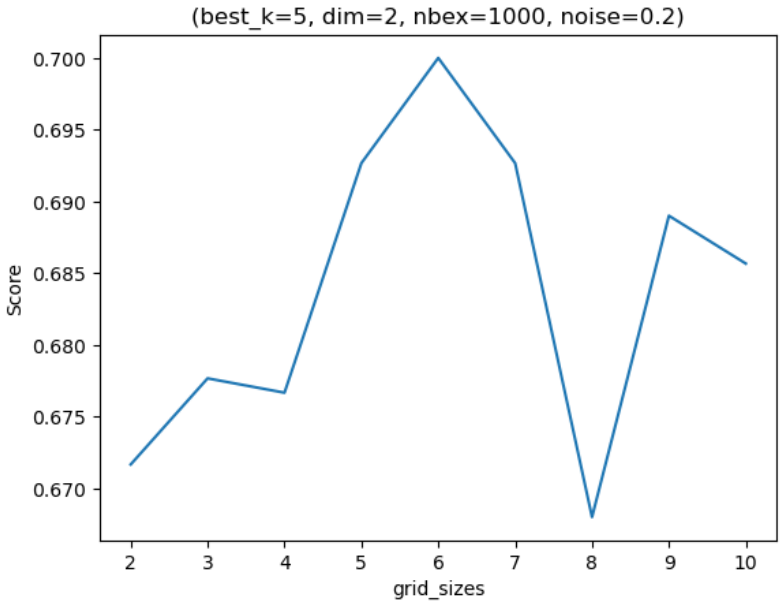
Noise = 0.0 :

```
Grid size: 2, Score: 0.9040
Grid size: 3, Score: 0.9047
Grid size: 4, Score: 0.9080
Grid size: 5, Score: 0.8970
Grid size: 6, Score: 0.9137
Grid size: 7, Score: 0.9030
Grid size: 8, Score: 0.9030
Grid size: 9, Score: 0.9073
Grid size: 10, Score: 0.9083
```



Noise = 0.2:

```
Grid size: 2, Score: 0.6717
Grid size: 3, Score: 0.6777
Grid size: 4, Score: 0.6767
Grid size: 5, Score: 0.6927
Grid size: 6, Score: 0.7000
Grid size: 7, Score: 0.6927
Grid size: 8, Score: 0.6680
Grid size: 9, Score: 0.6890
Grid size: 10, Score: 0.6857
```



On observe que le score de classification augmente avec la taille de la grille de recherche, jusqu'à atteindre un plateau à partir de **grid_size = 6**. Pour cette valeur de **grid_size**, on obtient généralement une meilleure performance en termes de score de classification sur l'échantillon de test. Cela peut s'expliquer par le fait que pour des valeurs de **grid_size** plus faibles, les classes peuvent être plus difficiles à distinguer et cela peut causer des erreurs de classification.

Lorsque l'on introduit du bruit dans les données, on observe que la tendance générale reste la même. Cependant, la performance globale de l'algorithme est généralement plus faible en présence de bruit, ce qui s'explique par le fait que le bruit peut rendre les classes plus difficiles à distinguer et donc causer des erreurs de classification.

6)

3. Impact de la distance :

On observe des phénomènes similaires avec la distance euclidienne car les principes généraux de **KNN** s'appliquent indépendamment de la distance utilisée.

4. Conclusion :

- Plus le nombre d'exemples augmente, plus la performance de **KNN** peut potentiellement augmenter, car il y a plus de données disponibles pour la classification.

- Lorsque la dimension de augmente, la performance de **KNN** diminue.
- L'hyperparamètre **k** de **KNN** doit être choisi avec soin, car il peut avoir un impact significatif sur les performances du modèle.
- Plus le bruit est élevé, plus la performance du **KNN** diminue. Cela est dû au fait que le bruit ajoute des exemples qui sont mal étiquetés, ce qui rend la tâche de classification plus difficile pour le **KNN**.