QUIZ AVEC CODE

1. Quelle affirmation est correcte concernant DBSCAN ? (10 points)

- A. DBSCAN nécessite de spécifier le nombre de clusters à l'avance.

- B. DBSCAN peut identifier des clusters de forme arbitraire.

- C. DBSCAN est sensible à l'initialisation des centroids.

- D. DBSCAN ne peut pas détecter les points bruyants.

2. Quel code permet d'importer et d'initialiser correctement l'algorithme DBSCAN en utilisant scikit-learn avec une distance epsilon de 0.5 et un nombre minimal de points de 5 ? (10 points)

- A.

from sklearn.cluster import KMeans

kmeans = DBSCAN(n\_clusters=3)

- B.

from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=0.3, min\_samples=3)

- C.

from sklearn.cluster import DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

- D.

from sklearn.cluster import DBSCAN

dbscan = DBSCAN(0.5, 0.5)

3. Quel code applique l'algorithme DBSCAN sur un jeu de données 'X' pour identifier les clusters ? (10 points)

- A.

clusters = dbscan.predict(X)

- B.

clusters = dbscan.fit\_predict(X)

- C.

clusters = dbscan.fit\_transform(X)

- D.

clusters = dbscan.transform(X)

4. Quel code permet de visualiser les résultats du clustering DBSCAN en utilisant matplotlib ? (10 points)

- A.

import matplotlib.pyplot as plt

plt.scatter(X[:, 0], X[:, 1], c=clusters)

plt.show()

- B.

import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))

labels = set(clusters)

for label in labels:

cluster\_points = X[clusters == label]

color = 'k' if label == -1 else None

plt.scatter(cluster\_points[:, 0], cluster\_points[:, 1], label=f'Cluster {label}', c=color)

plt.legend()

plt.show()

- C.

import matplotlib.pyplot as plt

plt.scatter(X[:, 0], X[:, 1])

plt.show()

5. Quel code permet d'effectuer une recherche par grille pour trouver les meilleurs paramètres pour DBSCAN ? (10 points)

- A.

from sklearn.model\_selection import GridSearchCV

param\_grid = {

'eps': [0.3, 0.4, 0.5, 0.6],

'min\_samples': [3, 5, 7, 10]

}

dbscan = DBSCAN()

grid\_search = GridSearchCV(dbscan, param\_grid, cv=3)

grid\_search.fit(X)

print('Meilleurs paramètres:', grid\_search.best\_params\_)

- B.

from sklearn.model\_selection import GridSearchCV

param\_grid = {

'eps': [0.1, 0.2, 0.3, 0.4],

'min\_samples': [5, 10, 15, 20]

}

dbscan = DBSCAN()

grid\_search = GridSearchCV(dbscan, param\_grid, cv=3)

grid\_search.fit(X)

print('Meilleurs paramètres:', grid\_search.best\_params\_)

- C.

from sklearn.model\_selection import RandomizedSearchCV

param\_grid = {

'eps': [0.3, 0.4, 0.5, 0.6],

'min\_samples': [3, 5, 7, 10]

}

6. Quel code permet d'extraire et d'afficher le nombre de clusters (hors bruit) détectés par DBSCAN ? (10 points)

- A.

num\_clusters = len(set(clusters)) - (1 if -1 in clusters else 0)

print('Nombre de clusters détectés:', num\_clusters)

- B.

num\_clusters = len(set(clusters))

print('Nombre de clusters détectés:', num\_clusters)

- C.

num\_clusters = max(clusters) + 1

print('Nombre de clusters détectés:', num\_clusters)

- D.

num\_clusters = clusters.max() + 1

print('Nombre de clusters détectés:', num\_clusters)

7. Quel code permet d'identifier et de compter les points considérés comme du bruit par DBSCAN ? (10 points)

- A.

noise\_points = sum(clusters == -1)

print("Nombre de points de bruit:", noise\_points)

- B.

noise\_points = sum(clusters == 0)

print("Nombre de points de bruit:", noise\_points)

- C.

noise\_points = len(clusters[clusters == -1])

print("Nombre de points de bruit:", noise\_points)

- D.

noise\_points = len(clusters[clusters == 0])

print("Nombre de points de bruit:", noise\_points)

8. Quel code permet de charger des données depuis un fichier CSV et d'appliquer DBSCAN pour identifier les clusters ? (10 points)

- A.

import pandas as pd

from sklearn.cluster import DBSCAN

data = pd.read\_csv('data.csv')

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

- B.

import numpy as np

from sklearn.cluster import DBSCAN

data = np.genfromtxt('data.csv', delimiter=',')

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

- C.

import pandas as pd

from sklearn.cluster import KMeans

data = pd.read\_csv('data.csv')

dbscan = KMeans(n\_clusters=3)

clusters = dbscan.fit\_predict(data)

- D.

import numpy as np

from sklearn.cluster import KMeans

data = np.genfromtxt('data.csv', delimiter=',')

9. Quel est l'effet de l'augmentation de la valeur de 'eps' dans DBSCAN ? (10 points)

- A. Réduire la taille des clusters détectés.

- B. Augmenter la taille des clusters détectés.

- C. Augmenter le nombre de points de bruit.

- D. Réduire le nombre de points de bruit.

10. Quel code permet de calculer l'indice de silhouette pour évaluer les clusters détectés par DBSCAN ? (10 points)

- A.

from sklearn.metrics import silhouette\_score

sil\_score = silhouette\_score(X, clusters)

print("Indice de silhouette:", sil\_score)

- B.

from sklearn.metrics import calinski\_harabasz\_score

sil\_score = calinski\_harabasz\_score(X, clusters)

print("Indice de silhouette:", sil\_score)

- C.

from sklearn.metrics import davies\_bouldin\_score

sil\_score = davies\_bouldin\_score(X, clusters)

print("Indice de silhouette:", sil\_score)

- D.

from sklearn.metrics import silhouette\_samples

sil\_score = silhouette\_samples(X, clusters)

print("Indice de silhouette:", sil\_score)

11. Quel code permet de charger un jeu de données, appliquer DBSCAN et visualiser les clusters détectés en utilisant matplotlib ? (10 points)

- A.

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import DBSCAN

# Charger les données

data = pd.read\_csv('data.csv')

X = data.values

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(X)

# Visualiser les clusters

plt.scatter(X[:, 0], X[:, 1], c=clusters)

plt.show()

- B.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import DBSCAN

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

# Visualiser les clusters

plt.scatter(data[:, 0], data[:, 1], c=clusters)

plt.show()

- C.

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

# Charger les données

data = pd.read\_csv('data.csv')

X = data.values

# Appliquer KMeans

kmeans = KMeans(n\_clusters=3)

clusters = kmeans.fit\_predict(X)

# Visualiser les clusters

plt.scatter(X[:, 0], X[:, 1], c=clusters)

plt.show()

- D.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer KMeans

kmeans = KMeans(n\_clusters=3)

clusters = kmeans.fit\_predict(data)

# Visualiser les clusters

plt.scatter(data[:, 0], data[:, 1], c=clusters)

plt.show()

12. Quel code permet d'évaluer la qualité des clusters détectés par DBSCAN en utilisant l'indice de Calinski-Harabasz ? (10 points)

- A.

from sklearn.metrics import calinski\_harabasz\_score

# Calculer l'indice de Calinski-Harabasz

ch\_score = calinski\_harabasz\_score(X, clusters)

print("Indice de Calinski-Harabasz:", ch\_score)

- B.

from sklearn.metrics import silhouette\_score

# Calculer l'indice de silhouette

sil\_score = silhouette\_score(X, clusters)

print("Indice de silhouette:", sil\_score)

- C.

from sklearn.metrics import davies\_bouldin\_score

# Calculer l'indice de Davies-Bouldin

db\_score = davies\_bouldin\_score(X, clusters)

print("Indice de Davies-Bouldin:", db\_score)

- D.

from sklearn.metrics import silhouette\_samples

# Calculer l'indice de silhouette

sil\_samples = silhouette\_samples(X, clusters)

print("Indices de silhouette:", sil\_samples)

13. Quel code permet d'identifier et de compter les points considérés comme du bruit par DBSCAN ? (10 points)

- A.

noise\_points = sum(clusters == -1)

print("Nombre de points de bruit:", noise\_points)

- B.

noise\_points = sum(clusters == 0)

print("Nombre de points de bruit:", noise\_points)

- C.

noise\_points = len(clusters[clusters == -1])

print("Nombre de points de bruit:", noise\_points)

- D.

noise\_points = len(clusters[clusters == 0])

print("Nombre de points de bruit:", noise\_points)

14. Quel code permet de charger des données depuis un fichier CSV et d'appliquer DBSCAN pour identifier les clusters ? (10 points)

- A.

import pandas as pd

from sklearn.cluster import DBSCAN

# Charger les données

data = pd.read\_csv('data.csv')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

- B.

import numpy as np

from sklearn.cluster import DBSCAN

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

- C.

import pandas as pd

from sklearn.cluster import KMeans

# Charger les données

data = pd.read\_csv('data.csv')

# Appliquer DBSCAN

dbscan = KMeans(n\_clusters=3)

clusters = dbscan.fit\_predict(data)

- D.

import numpy as np

from sklearn.cluster import KMeans

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

15. Quel code permet de calculer l'indice de silhouette pour évaluer les clusters détectés par DBSCAN ? (10 points)

- A.

from sklearn.metrics import silhouette\_score

# Calculer l'indice de silhouette

sil\_score = silhouette\_score(X, clusters)

print("Indice de silhouette:", sil\_score)

- B.

from sklearn.metrics import calinski\_harabasz\_score

# Calculer l'indice de Calinski-Harabasz

ch\_score = calinski\_harabasz\_score(X, clusters)

print("Indice de Calinski-Harabasz:", ch\_score)

- C.

from sklearn.metrics import davies\_bouldin\_score

# Calculer l'indice de Davies-Bouldin

db\_score = davies\_bouldin\_score(X, clusters)

print("Indice de Davies-Bouldin:", db\_score)

- D.

from sklearn.metrics import silhouette\_samples

# Calculer l'indice de silhouette

sil\_samples = silhouette\_samples(X, clusters)

print("Indices de silhouette:", sil\_samples)

16. Quel code permet de charger un jeu de données, appliquer DBSCAN et visualiser les clusters détectés en utilisant matplotlib ? (10 points)

- A.

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import DBSCAN

# Charger les données

data = pd.read\_csv('data.csv')

X = data.values

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(X)

# Visualiser les clusters

plt.scatter(X[:, 0], X[:, 1], c=clusters)

plt.show()

- B.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import DBSCAN

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

# Visualiser les clusters

plt.scatter(data[:, 0], data[:, 1], c=clusters)

plt.show()

- C.

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

# Charger les données

data = pd.read\_csv('data.csv')

X = data.values

# Appliquer KMeans

kmeans = KMeans(n\_clusters=3)

clusters = kmeans.fit\_predict(X)

# Visualiser les clusters

plt.scatter(X[:, 0], X[:, 1], c=clusters)

plt.show()

- D.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer KMeans

kmeans = KMeans(n\_clusters=3)

clusters = kmeans.fit\_predict(data)

# Visualiser les clusters

plt.scatter(data[:, 0], data[:, 1], c=clusters)

plt.show()

17. Quel est l'effet de l'augmentation de la valeur de 'eps' dans DBSCAN ? (10 points)

- A. Réduire la taille des clusters détectés.

- B. Augmenter la taille des clusters détectés.

- C. Augmenter le nombre de points de bruit.

- D. Réduire le nombre de points de bruit.

18. Quel code permet de calculer l'indice de silhouette pour évaluer les clusters détectés par DBSCAN ? (10 points)

- A.

from sklearn.metrics import silhouette\_score

# Calculer l'indice de silhouette

sil\_score = silhouette\_score(X, clusters)

print("Indice de silhouette:", sil\_score)

- B.

from sklearn.metrics import calinski\_harabasz\_score

# Calculer l'indice de Calinski-Harabasz

ch\_score = calinski\_harabasz\_score(X, clusters)

print("Indice de Calinski-Harabasz:", ch\_score)

- C.

from sklearn.metrics import davies\_bouldin\_score

# Calculer l'indice de Davies-Bouldin

db\_score = davies\_bouldin\_score(X, clusters)

print("Indice de Davies-Bouldin:", db\_score)

- D.

from sklearn.metrics import silhouette\_samples

# Calculer l'indice de silhouette

sil\_samples = silhouette\_samples(X, clusters)

print("Indices de silhouette:", sil\_samples)

19. Quel code permet de charger des données depuis un fichier CSV et d'appliquer DBSCAN pour identifier les clusters ? (10 points)

- A.

import pandas as pd

from sklearn.cluster import DBSCAN

# Charger les données

data = pd.read\_csv('data.csv')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

- B.

import numpy as np

from sklearn.cluster import DBSCAN

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

- C.

import pandas as pd

from sklearn.cluster import KMeans

# Charger les données

data = pd.read\_csv('data.csv')

# Appliquer KMeans

kmeans = KMeans(n\_clusters=3)

clusters = kmeans.fit\_predict(data)

- D.

import numpy as np

from sklearn.cluster import KMeans

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

20. Quel code permet de calculer l'indice de Davies-Bouldin pour évaluer les clusters détectés par DBSCAN ? (10 points)

- A.

from sklearn.metrics import davies\_bouldin\_score

# Calculer l'indice de Davies-Bouldin

db\_score = davies\_bouldin\_score(X, clusters)

print("Indice de Davies-Bouldin:", db\_score)

- B.

from sklearn.metrics import silhouette\_score

# Calculer l'indice de silhouette

sil\_score = silhouette\_score(X, clusters)

print("Indice de silhouette:", sil\_score)

- C.

from sklearn.metrics import calinski\_harabasz\_score

# Calculer l'indice de Calinski-Harabasz

ch\_score = calinski\_harabasz\_score(X, clusters)

print("Indice de Calinski-Harabasz:", ch\_score)

- D.

from sklearn.metrics import silhouette\_samples

# Calculer l'indice de silhouette

sil\_samples = silhouette\_samples(X, clusters)

print("Indices de silhouette:", sil\_samples)

21. Quel code permet d'extraire et d'afficher le nombre de clusters (hors bruit) détectés par DBSCAN ? (10 points)

- A.

num\_clusters = len(set(clusters)) - (1 if -1 in clusters else 0)

print('Nombre de clusters détectés:', num\_clusters)

- B.

num\_clusters = len(set(clusters))

print('Nombre de clusters détectés:', num\_clusters)

- C.

num\_clusters = max(clusters) + 1

print('Nombre de clusters détectés:', num\_clusters)

- D.

num\_clusters = clusters.max() + 1

print('Nombre de clusters détectés:', num\_clusters)

22. Quel code permet d'identifier et de compter les points considérés comme du bruit par DBSCAN ? (10 points)

- A.

noise\_points = sum(clusters == -1)

print("Nombre de points de bruit:", noise\_points)

- B.

noise\_points = sum(clusters == 0)

print("Nombre de points de bruit:", noise\_points)

- C.

noise\_points = len(clusters[clusters == -1])

print("Nombre de points de bruit:", noise\_points)

- D.

noise\_points = len(clusters[clusters == 0])

print("Nombre de points de bruit:", noise\_points)

23. Quel code permet de charger un jeu de données, appliquer DBSCAN et visualiser les clusters détectés en utilisant matplotlib ? (10 points)

- A.

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import DBSCAN

# Charger les données

data = pd.read\_csv('data.csv')

X = data.values

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(X)

# Visualiser les clusters

plt.scatter(X[:, 0], X[:, 1], c=clusters)

plt.show()

- B.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import DBSCAN

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

# Visualiser les clusters

plt.scatter(data[:, 0], data[:, 1], c=clusters)

plt.show()

- C.

import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

# Charger les données

data = pd.read\_csv('data.csv')

X = data.values

# Appliquer KMeans

kmeans = KMeans(n\_clusters=3)

clusters = kmeans.fit\_predict(X)

# Visualiser les clusters

plt.scatter(X[:, 0], X[:, 1], c=clusters)

plt.show()

- D.

import numpy as np

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer KMeans

kmeans = KMeans(n\_clusters=3)

clusters = kmeans.fit\_predict(data)

# Visualiser les clusters

plt.scatter(data[:, 0], data[:, 1], c=clusters)

plt.show()

24. Quel est l'effet de l'augmentation de la valeur de 'eps' dans DBSCAN ? (10 points)

- A. Réduire la taille des clusters détectés.

- B. Augmenter la taille des clusters détectés.

- C. Augmenter le nombre de points de bruit.

- D. Réduire le nombre de points de bruit.

25. Quel code permet de charger des données depuis un fichier CSV et d'appliquer DBSCAN pour identifier les clusters ? (10 points)

- A.

import pandas as pd

from sklearn.cluster import DBSCAN

# Charger les données

data = pd.read\_csv('data.csv')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

- B.

import numpy as np

from sklearn.cluster import DBSCAN

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')

# Appliquer DBSCAN

dbscan = DBSCAN(eps=0.5, min\_samples=5)

clusters = dbscan.fit\_predict(data)

- C.

import pandas as pd

from sklearn.cluster import KMeans

# Charger les données

data = pd.read\_csv('data.csv')

# Appliquer KMeans

kmeans = KMeans(n\_clusters=3)

clusters = kmeans.fit\_predict(data)

- D.

import numpy as np

from sklearn.cluster import KMeans

# Charger les données

data = np.genfromtxt('data.csv', delimiter=',')