EJEMPLO DE EXAMEN

1. ¿Qué diferencia la evaluación extrínseca de la intrínseca?
2. la extrínseca considera sólo los centros de los clústeres mientras que la intrínseca considera todos los ejemplos
3. la intrínseca considera que se conocen los clústeres reales mientras que la extrínseca considera que son desconocidos
4. la extrínseca considera que se conocen los clústeres reales mientras que la intrínseca considera que son desconocidos
5. La dispersión intraclúster mide...
6. la distancia entre instancias de un mismo clúster
7. la distancia entre instancias de los dos clústeres más próximos
8. la distancia entre instancias de dos clústeres cualquiera
9. Con respecto al agrupamiento jerárquico cuál de las siguientes afirmaciones es falsa:
10. La longitud de las líneas verticales indica la distancia entre los clústeres que se unen y se separan
11. No es necesario indicar el número de clústeres para inicializar el algoritmo
12. Todas las anteriores son falsas
13. Con respecto al agrupamiento espectral, seleccionamos los K vectores propios:
14. Cuyos valores propios sean mayores, pues serán más importantes
15. Cuyos valores propios sean menores
16. No importa cuáles elijamos mientras que el número de vectores propios sea igual al número de clusters que deseamos
17. En el algoritmo DBSCAN, un punto nuclear es...
18. un centroide
19. un ejemplo que tiene un número mínimo de ejemplos en su vecindario
20. un ejemplo para el que todos sus vecinos tienen alta densidad
21. El clasificador naive Bayes asume...
22. ~~independencia entre las variables predictoras~~
23. independencia entre las variables predictoras dada la clase
24. independencia entre las variables predictoras y la variable clase

Nota: Ambas son válidas pero es más correcta la b.

1. Las variables originales suelen estandarizarse antes de realizar el análisis de componente principales para...
2. evitar que aquellas de menor rango dominen a las de mayor rango
3. evitar que aquellas de mayor rango dominen a las de menor rango
4. que aquellas de mayor rango dominen de manera estándar a aquellas de menor rango
5. En cuanto al algoritmo ICA:
6. Tendrá un buen funcionamiento para fuentes que siguen una distribución gaussiana
7. Está basado en el estimador máximo de verosimilud
8. Supone que las fuentes originales son dependientes
9. El término in-degree de un nodo se refiere al número de...
10. arcos que se dirigen hacia el nodo
11. arcos que se salen del nodo
12. las dos respuestas son válidas
13. En cuanto a los algoritmos basados en reglas de asociación, que valor de lift indica que hay complementariedad de los datos cuando:
14. 1
15. >1 (coocurrencia)
16. <1 (complementarios, implica que si ocurre uno no suele ocurrir el otro)
17. El objetivo del deep clustering es:
18. Obtener una representación con menor dimensión de los datos
19. Minimizar la distancia intracluster
20. Optimizar el espacio latente obtenido mientras que se mejora el agrupamiento

VIDEO EVALUABLE

1. ¿Cuáles de los siguientes son modelos generativos?
2. Regresión logística
3. Support Vector Machines
4. Naive Bayes
5. ¿Cuál es el principal problema de los autoencoders?
6. Realizan un ajuste casi perfecto de los datos, lo que produce un gran over-fitting y hace que el espacio latente no sea suave, por lo que no se pueden utilizar para generar nuevos datos diferentes de los ya vistos durante el entrenamiento.
7. Realizan un ajuste casi perfecto de los datos, lo que produce un gran over-fitting y hace que el espacio latente no sea suave, por lo que se pueden utilizar para generar nuevos datos diferentes de los ya vistos durante el entrenamiento.
8. Realizan un ajuste casi perfecto de los datos, lo que produce un gran under-fitting y hace que el espacio latente no sea suave, por lo que no se pueden utilizar para generar nuevos datos diferentes de los ya vistos durante el entrenamiento.
9. ¿Cómo se entrena el generador de una GAN?
   1. El generador toma como entrada N vectores de ruido aleatorio, genera N datos sintéticos y los introduce al discriminador. Así, el discriminador, que tiene sus pesos congelados, le indica al generador si ha conseguido engañarle o no. En caso de no haber conseguido engañarle, el generador actualizará sus pesos.
   2. El generador toma como entrada N vectores de ruido aleatorio, genera N datos sintéticos y los introduce al discriminador. Así, el discriminador, que tiene sus pesos congelados, le indica al generador si ha conseguido engañarle o no. En caso de haber conseguido engañarle, el generador actualizará sus pesos.
   3. El generador toma como entrada N vectores de ruido aleatorio, genera N datos sintéticos y los introduce al discriminador. Así, el discriminador le indica al generador si ha conseguido engañarle o no. En caso de no haber conseguido engañarle, el generador actualizará sus pesos.

EXAMEN DE CÓDIGO

P1. Tenemos un dataset de 1000 imágenes con 128 atributos (variables) realizamos un K-means con un número de cluster igual a 5, kmeans= KMeans(n\_clusters=5, random\_state=0). Obtenemos los clusters de la siguiente forma: clusters= kmeans.fit\_predict(data), siendo data los datos de entrada y target su etiqueta. ¿Cuál será la dimensión de kmeans.cluster\_centers?

a) 5x128

b) 128x5

c) 1000x5

* 1. Ninguna de las anteriores

P2. Queremos hacer una compresión en el número de colores de una imagen pasando de 16 millones a 16 y para ello utilizaremos el algoritmo de K-means, siendo data los datos disponibles.

¿Cómo deberíamos hacerlo?

1. kmeans = MiniBatchKMeans(16) new\_colors = kmeans.fit(data)
2. kmeans = MiniBatchKMeans(16) kmeans.fit(data)

new\_colors = kmeans.cluster\_centers\_[kmeans.predict(data)]

1. kmeans = MiniBatchKMeans(16) kmeans.fit(data)

new\_colors = kmeans.predict(data)

1. Ninguna de las anteriores

P3. Siendo D la matriz de grado, W\_ad la matriz de adyacencia, W\_dist la matriz de distancias y L la matriz laplaciana ¿Cómo obtenemos la eigenvectores para clustering espectral?

1. Import numpy as np e, v = np.linalg.eig(L)
2. Import numpy as np

e, v = np.linalg.eig(W\_dist)

1. Import numpy as np

e, v = np.linalg.eig(W\_ad)

1. Import numpy as np e, v = np.linalg.eig(D)

P4. ¿Cuáles son los pasos para aplicar el algoritmo DBSCAN?

1. dbscan = DBSCAN(eps=0.123, min\_samples=2) clusters = dbscan.fit\_predict(X)
2. X\_scaled = scaler.fit\_transform(X)

dbscan = DBSCAN(eps=0.123, min\_samples=2) clusters = dbscan.fit\_predict(X)

1. scaler = StandardScaler() X\_scaled = scaler.fit\_transform(X)

dbscan = DBSCAN(eps=0.123, min\_samples=2) clusters = dbscan.fit\_predict(X\_scaled)

1. Ninguna de las anteriores

P5. Referente a los métodos de clustering probabilísticos, ¿cómo obtenemos las etiquetas a las que pertenece cada uno de los datos?

1. gmm= GaussianMixture(n\_components=3).fit(X) labels= gmm.predict(X)
2. gmm= GaussianMixture(n\_components=3).fit(X) labels = gmm.predict\_proba(X)
3. a y b son ciertas
4. Ninguna de las anteriores es correcta

P6. ¿Cuáles son los pasos correctos para aplicar PCA a los datos (X) (con una componente) y obtener los datos reproyectados en el espacio original?

1. pca\_reduce = PCA(1)

X\_rebuild = pca\_reduce.inverse\_transform(X)

1. pca\_reduce = PCA(1)

X\_rebuild = pca\_reduce.inverse\_transform(X)

1. pca\_reduce = PCA(1)

X\_proj = pca\_reduce.fit\_transform(X)

X\_rebuild = pca\_reduce.inverse\_transform(X\_proj)

1. Ninguna de la anterior es correcta

