GUÍA DE ACTIVIDADES Y RÚBRICA DE EVALUACIÓN - UNIDAD 3 - TAREA 4 - ALGORITMOS DE APRENDIZAJE NO SUPERVISADO

GRUPO: 202016899A_1144

JUAN CAMILO CHAVES HERNANDEZ

JOSE LUIS SIERRA

CÓDIGO: 1019028631 - 202016908A_1701

PRESENTADO A:

BREYNER ALEXANDER PARRA

TUTOR DE ANÁLISIS DE DATOS

UNIVERSIDAD NACIONAL ABIERTA Y A DISTANCIA- UNAD

ESCUELA DE CIENCIAS BÁSICAS, TECNOLOGÍA E INGENIERÍA (ECBTI)

ADMINISTRACIÓN DE BASES DE DATOS - (202016902A_1391)

SEDE PRINCIPAL BOGOTA

11 DE MAYO DEL 2024

Tabla de contenido

INTRODUCCIÓN

El presente trabajo contiene por objeto la entrega de unas actividades específicas, por medio de las cuales se logró tener un conocimiento más el análisis de datos anivel de inteligencia artificial.

OBJETIVOS

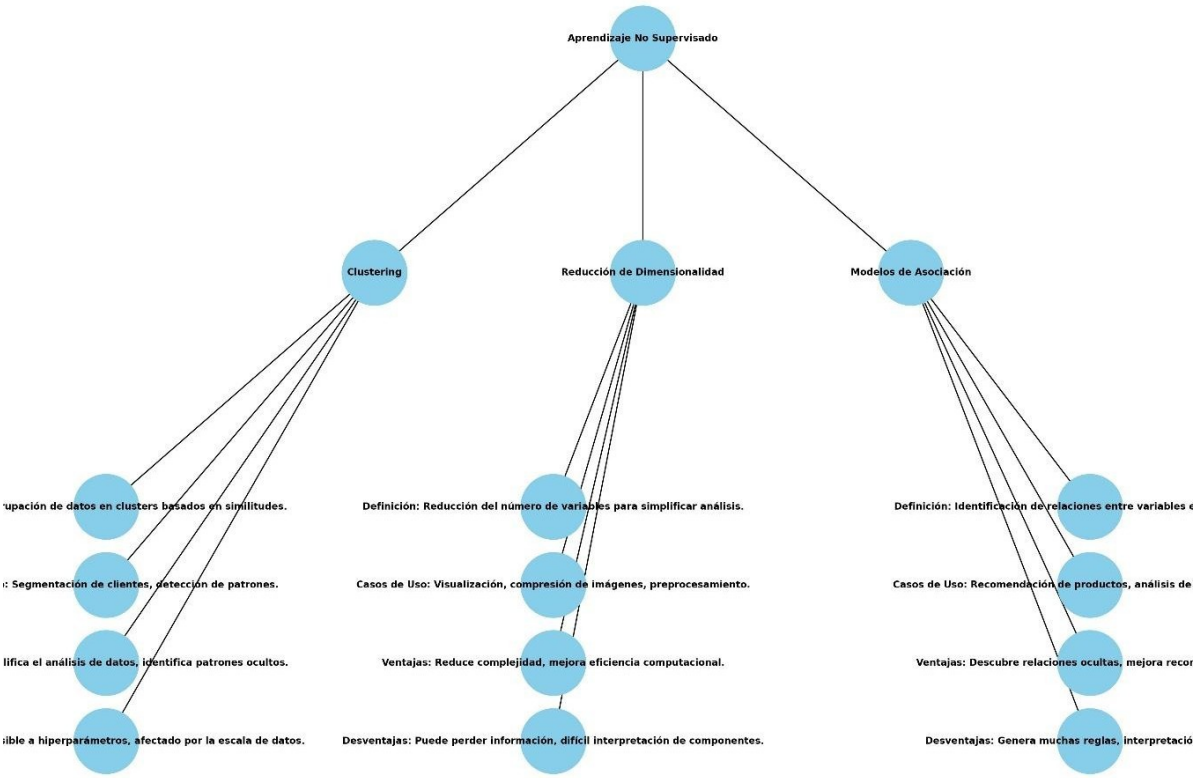
- Resolver en grupo los diferentes conceptos y comprender las principales características y relaciones entre la Inteligencia Artificial, el Machine Learning y el Deep Learning a través de la elaboración de un cuadro sinóptico

ACTIVIDAD

✓ PUNTO 1

Elaborar un cuadro sinóptico sobre los diferentes modelos de Aprendizaje no supervisado que incluya definición, casos de uso,

Modelo de Aprendizaje No Supervisado	Definición	Casos de Uso	Ventajas	Desventajas
K-Means	Algoritmo que agrupa datos en k grupos, minimizando la varianza dentro de cada grupo	Segmentación de clientes según sus preferencias de compra. Análisis de mercado para identificar grupos de productos similares.	Fácil de implementar y entender. Escalable a grandes conjuntos de datos.	Sensible a la inicialización de los centroides. Necesita especificar el número de clusters k.
DBSCAN	Algoritmo que identifica regiones densas de puntos en el espacio de características	Detección de anomalías en sistemas de seguridad. Agrupamiento de puntos en mapas según su densidad de población.	No requiere especificar el número de clusters. Robusto ante ruido y outliers.	Sensible a la elección de los parámetros epsilon y minPts. Puede tener dificultades con clusters de diferente densidad.
Redes Neuronales Auto-Organizativas	Redes neuronales que aprenden estructuras de datos sin supervisión	Reducción de dimensionalidad para visualización de datos. Agrupamiento de características similares en datos no estructurados.	Capacidad de aprender estructuras complejas y patrones no lineales.	Requiere más recursos computacionales que otros métodos. Puede ser difícil de interpretar y sintonizar correctamente.



MAPA MENTAL

```
import matplotlib.pyplot as plt
import networkx as nx

G = nx.DiGraph()
G.add_node("Aprendizaje No Supervisado", pos=(0, 4))
G.add_node("Clustering", pos=(-3, 2.5))
G.add_node("Reducción de Dimensionalidad", pos=(0, 2.5))
G.add_node("Modelos de Asociación", pos=(3, 2.5))
G.add_node("Def: Agrupación de datos en clusters basados en similitudes.", pos=(-6, 1))
G.add_node("Casos de Uso: Segmentación de clientes, detección de patrones.", pos=(-6, 0.5))
G.add_node("Ventajas: Simplifica el análisis de datos, identifica patrones ocultos.", pos=(-6, 0))
G.add_node("Desventajas: Sensible a hiperparámetros, afectado por la escala de datos.", pos=(-6, -0.5))
G.add_node("Def: Reducción del número de variables para simplificar análisis.", pos=(-1, 1))
G.add_node("Casos de Uso: Visualización, compresión de imágenes, preprocesamiento.", pos=(-1, 0.5))
G.add_node("Ventajas: Reduce complejidad, mejora eficiencia computacional.", pos=(-1, 0))
G.add_node("Desventajas: Puede perder información, difícil interpretación de componentes.", pos=(-1, -0.5))
G.add_node("Def: Identificación de relaciones entre variables en grandes datos.", pos=(5, 1))
G.add_node("Casos de Uso: Recomendación de productos, análisis de cesta de compra.", pos=(5, 0.5))
G.add_node("Ventajas: Descubre relaciones ocultas, mejora recomendaciones.", pos=(5, 0))
G.add_node("Desventajas: Genera muchas reglas, interpretación compleja.", pos=(5, -0.5))
G.add_edges_from([
   ("Aprendizaje No Supervisado", "Clustering"),
   ("Aprendizaje No Supervisado", "Reducción de Dimensionalidad"),
   ("Aprendizaje No Supervisado", "Modelos de Asociación"),
   ("Clustering", "Def: Agrupación de datos en clusters basados en similitudes."),
```

```
            ("Clustering", "Ejemplos: Segmentación de clientes, detección de patrones."),
            ("Clustering", "Ventajas: Simplifica el análisis de datos, identifica patrones ocultos."),
            ("Clustering", "Desventajas: Sensible a hiperparámetros, afectado por la escala de datos."),
            ("Reducción de Dimensionalidad", "Def: Reducción del número de variables para simplificar análisis."),
            ("Reducción de Dimensionalidad", "Casos de Uso: Visualización, compresión de imágenes, preprocesamiento."),
            ("Reducción de Dimensionalidad", "Ventajas: Reduce complejidad, mejora eficiencia computacional."),
            ("Reducción de Dimensionalidad", "Desventajas: Puede perder información, difícil interpretación de componentes."),
            ("Modelos de Asociación", "Def: Identificación de relaciones entre variables en grandes datos."),
            ("Modelos de Asociación", "Casos de Uso: Recomendación de productos, análisis de cesta de compra."),
            ("Modelos de Asociación", "Ventajas: Descubre relaciones ocultas, mejora recomendaciones."),
            ("Modelos de Asociación", "Desventajas: Genera muchas reglas, interpretación compleja.")
        ])

        pos = nx.get_node_attributes(G, 'pos')
        plt.figure(figsize=(14, 10))
        nx.draw(G, pos, with_labels=True, node_size=3000, node_color='skyblue', font_size=6, font_color='black',
        font_weight='bold', arrows=False)
        plt.title("Cuadro Sinóptico: Modelos de Aprendizaje No Supervisado", fontsize=8)
        plt.savefig('2.png', format='png', dpi=800)
plt.show()
```

✓ PUNTO 2

Elaborar un listado con las siguientes definiciones: Clustering, Centroide, Dendograma, Distancia euclidean, Dispersión intracluster, Dispersión inter-cluster, Coeficiente de Silhouette, Índice de Calinski-Harabasz, Índice de Davies-Bouldin, Coeficiente de correlación cofenética, Inertia.

- Clustering: Es un método de análisis de datos que agrupa puntos de datos con características similares en grupos o "clusters". El objetivo del clustering es identificar la estructura subyacente en un conjunto de datos sin información de clase predefinida.

- Centroide: Es el punto central de un cluster. Se calcula como la media de las posiciones de todos los puntos del cluster. Los centroides se utilizan para representar visualmente los clusters y para calcular la distancia entre clusters.

- Dendograma: Es un diagrama en forma de árbol que representa la jerarquía de clusters producida por un algoritmo de clustering. Los nodos del dendrograma representan clusters, y la distancia entre dos nodos representa la distancia entre los clusters correspondientes.

- Distancia euclidiana: Es una medida de distancia entre dos puntos en el espacio euclidiano. Se calcula como la raíz cuadrada de la suma de las diferencias al cuadrado de las coordenadas de los dos puntos. La distancia euclidiana se utiliza comúnmente en clustering para medir la similitud entre puntos de datos.

- Dispersión intra-cluster: Es la medida de la variabilidad dentro de un cluster. Se calcula como la media de las distancias entre cada punto del cluster y el centroide del cluster. Una dispersión intra-cluster baja indica que los puntos del cluster están muy cerca entre sí, mientras que una dispersión intra-cluster alta indica que los puntos del cluster están más dispersos.

- Dispersión inter-cluster: Es la medida de la distancia entre clusters. Se calcula como la media de las distancias entre los centroides de todos los pares de clusters. Una dispersión inter-cluster alta indica que los clusters están bien separados, mientras que una dispersión inter-cluster baja indica que los clusters están más cerca entre sí.

- Coeficiente de Silhouette: Es una medida de la calidad de un clustering. Se calcula como la media de la diferencia entre la distancia de un punto a su propio centroide y la distancia mínima a otro centroide. Un coeficiente de Silhouette alto indica que el clustering es bueno, mientras que un coeficiente de Silhouette bajo indica que el clustering es malo.

- Índice de Calinski-Harabasz: Es otra medida de la calidad de un clustering. Se calcula como la relación entre la dispersión inter-cluster y la dispersión intra-cluster promedio. Un índice de Calinski-Harabasz alto indica que el clustering es bueno, mientras que un índice de Calinski-Harabasz bajo indica que el clustering es malo.

- Índice de Davies-Bouldin: Es una medida de la calidad de un clustering. Se calcula como la relación entre la suma de las distancias promedio entre los puntos de un cluster y su centroide y la distancia mínima entre los centroides de dos clusters. Un índice de Davies-Bouldin bajo indica que el clustering es bueno, mientras que un índice de Davies-Bouldin alto indica que el clustering es malo.

- Coeficiente de correlación cofenética: Es una medida de la similitud entre la estructura jerárquica de un dendrograma y las distancias reales entre los puntos de datos. Un coeficiente de correlación cofenética alto indica que el dendrograma representa bien la estructura subyacente de los datos, mientras que un coeficiente de correlación cofenética bajo indica que el dendrograma no representa bien la estructura subyacente de los datos.

- Inertia: Es la medida de la dispersión total de los puntos de datos alrededor de los centroides de los clusters. Se calcula como la suma de las distancias al cuadrado entre cada punto de datos y el centroide del cluster al que pertenece. La inertia se utiliza para evaluar la calidad de un clustering y para comparar diferentes clusterings.

PUNTO 3

El lenguaje a utilizar es Python, el cual se trabajará mediante Jupyter notebooks, utilizando Anaconda.
• Descargar el siguiente dataset el cual se utilizará para el desarrollo de los 2 modelos 2 Dataset k-means e hierarchical clustering - Mall Customer Segmentation Data:

https://www.kaggle.com/vjchoudhary7/customer-segmentationtutorial-in-python

Este dataset contiene información de clientes de un mall. Las variables incluyen género, ingreso, puntaje de gasto, etc. o descárguelo del entorno de aprendizaje junto a la guía de actividades con el nombre Anexo 5 - Dataset Mall Customer Segmentation.zip

• Con el dataset anterior diseñar los modelos de Clustering (agrupación): K-means e hierarchical clustering. Para cada algoritmo realizar los siguientes pasos:

1. Realizar un análisis exploratorio de los datos para identificar relaciones entre variables, valores atípicos, tendencias, etc.
2. Preprocesar los datos limpiándolos, tratando valores faltantes y transformándolos según sea necesario.
3. Seleccionar las características más relevantes para entrenar el modelo utilizando selección de características.
4. Entrenar el modelo configurando los diferentes hiperparámetros.
5. Evaluar el desempeño del modelo con métricas como Coeficiente de Silhouette, Índice de Calinski-Harabasz, etc.

6. Realizar las diferentes gráficas que permitan visualizar los resultados del modelo

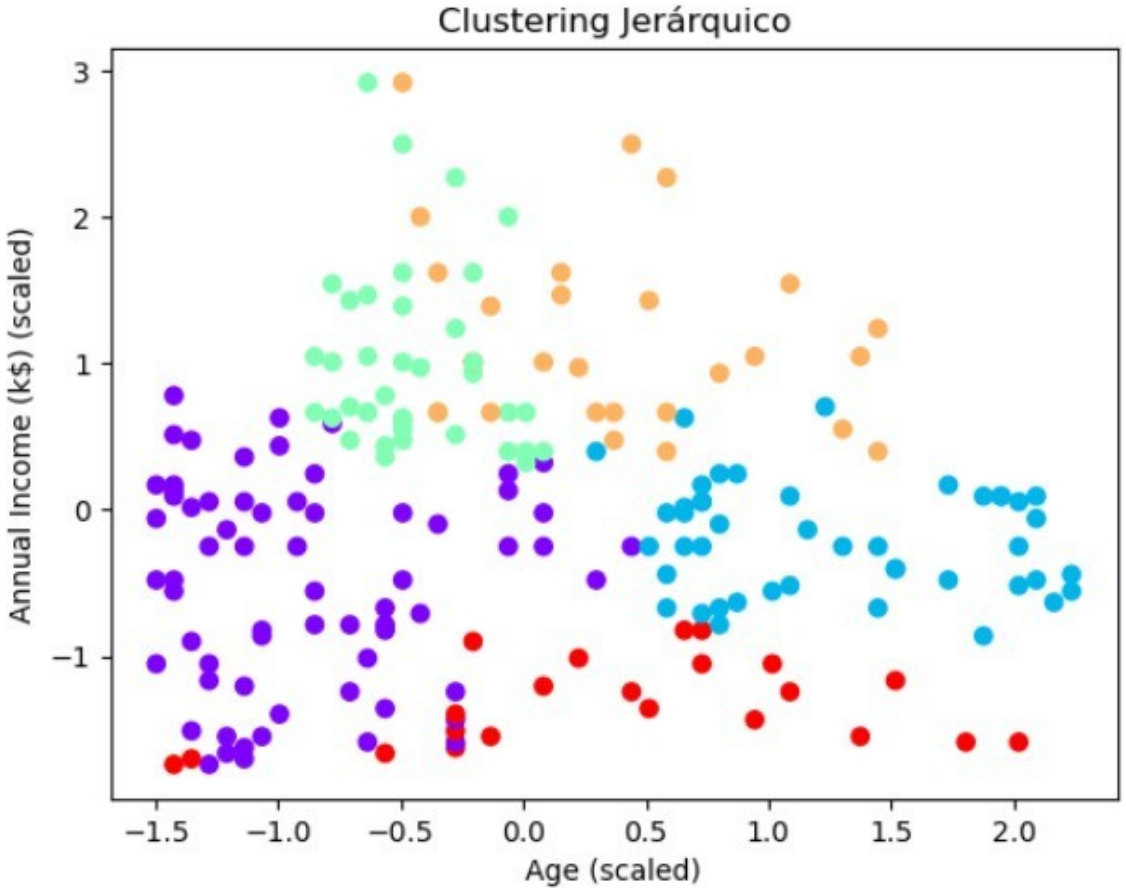
Ejercicio 1 K-means e hierarchical clustering

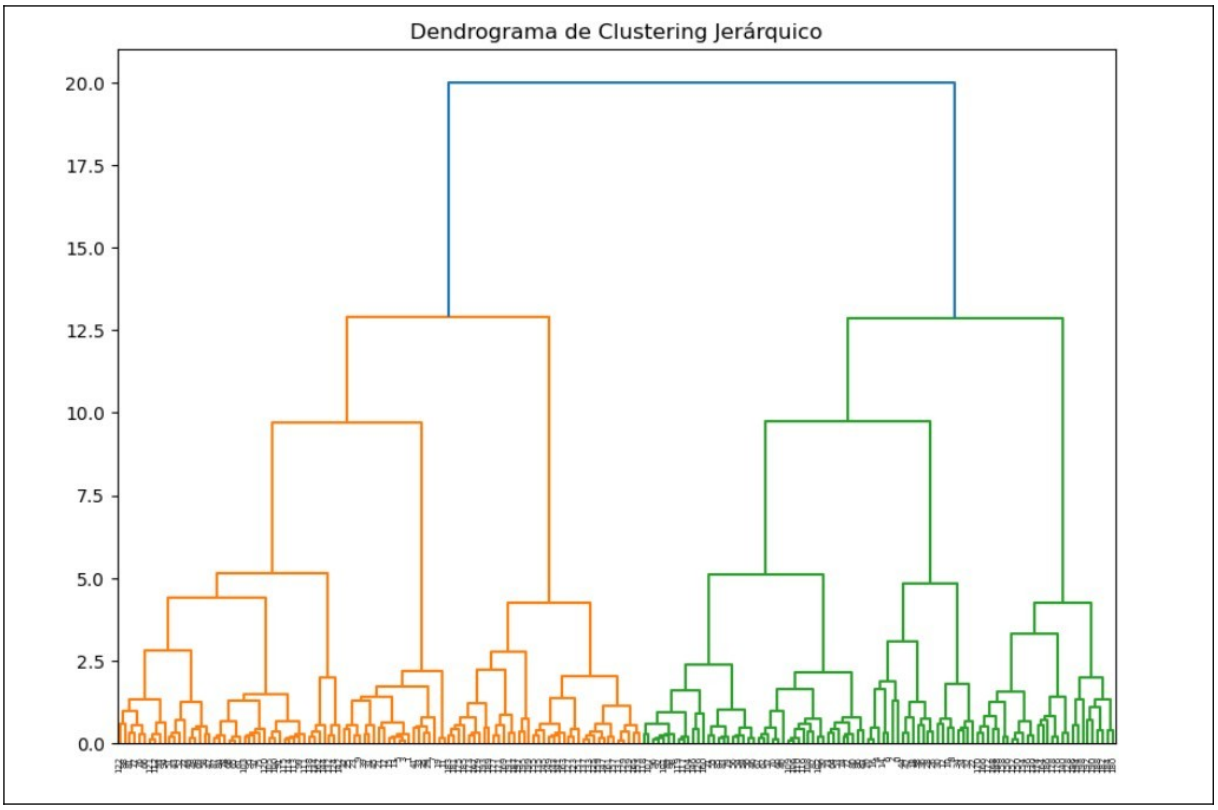
CODIGO JUAN CAMILO CHAVES HERNANDEZ

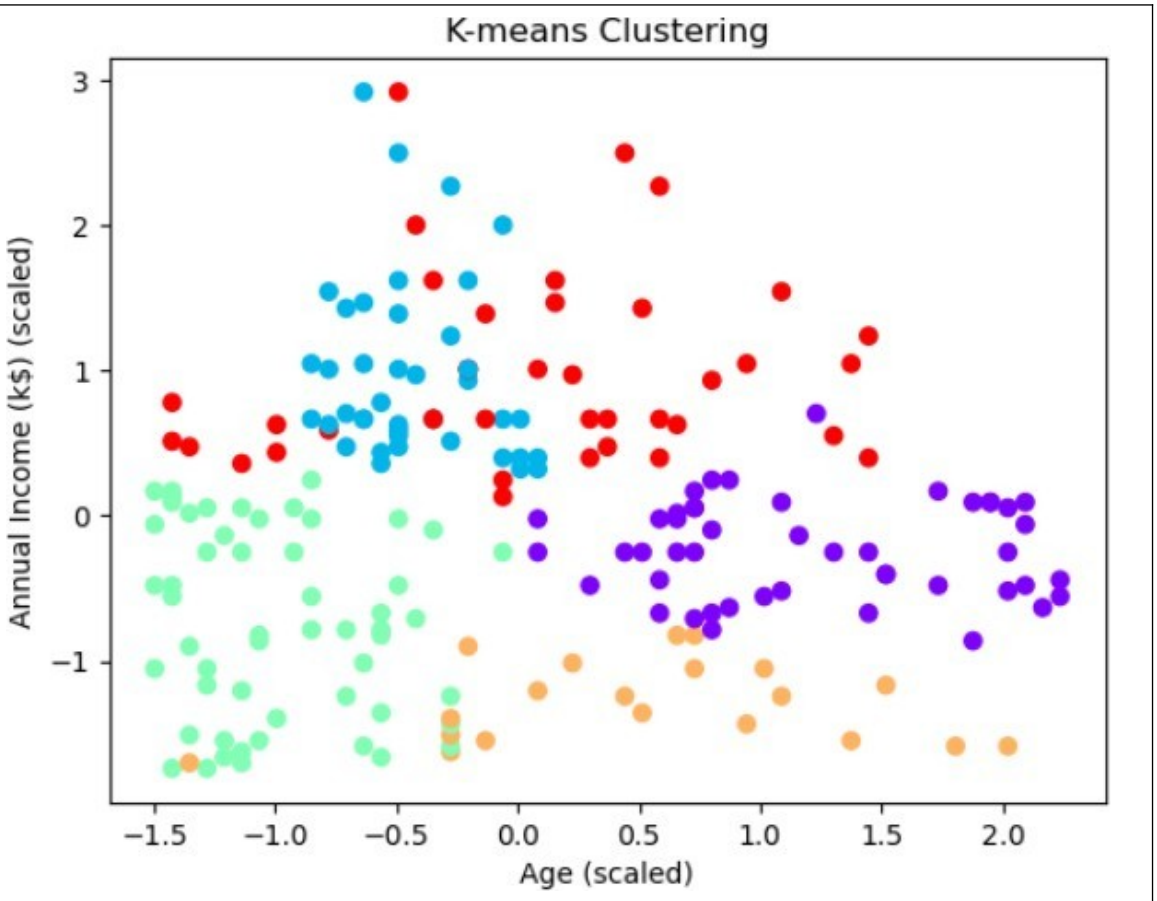
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.metrics import silhouette_score, calinski_harabasz_score, davies_bouldin_score
from sklearn.preprocessing import StandardScaler
from scipy.cluster.hierarchy import dendrogram, linkage
data = pd.read_csv('./1.csv')
print(data.head())
print(data.describe())
sns.pairplot(data)
plt.show()
plt.figure(figsize=(10, 5))
plt.subplot(1, 2, 1)
sns.histplot(data['Annual Income (k$)'], kde=True)
plt.title('Distribución del Ingreso Anual')
plt.subplot(1, 2, 2)
sns.histplot(data['Spending Score (1-100)'], kde=True)
plt.title('Distribución del Puntaje de Gasto')
plt.show()
data.dropna(inplace=True)
data = pd.get_dummies(data, columns=['Gender'], drop_first=True)
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data[['Age', 'Annual Income (k$)', 'Spending Score (1-100)']])
X = data_scaled
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit(X)
hierarchical = AgglomerativeClustering(n_clusters=5)
hierarchical.fit(X)
silhouette_kmeans = silhouette_score(X, kmeans.labels_)
ch_kmeans = calinski_harabasz_score(X, kmeans.labels_)
db_kmeans = davies_bouldin_score(X, kmeans.labels_)
silhouette_hierarchical = silhouette_score(X, hierarchical.labels_)
ch_hierarchical = calinski_harabasz_score(X, hierarchical.labels_)
db_hierarchical = davies_bouldin_score(X, hierarchical.labels_)
print(f'K-means Silhouette Score: {silhouette_kmeans}')
print(f'K-means Calinski-Harabasz Index: {ch_kmeans}')
print(f'K-means Davies-Bouldin Index: {db_kmeans}')
print(f'Hierarchical Clustering Silhouette Score: {silhouette_hierarchical}')
print(f'Hierarchical Clustering Calinski-Harabasz Index: {ch_hierarchical}')
print(f'Hierarchical Clustering Davies-Bouldin Index: {db_hierarchical}')
plt.scatter(X[:, 0], X[:, 1], c=kmeans.labels_, cmap='rainbow')
plt.title('K-means Clustering')
plt.xlabel('Age (scaled)')
plt.ylabel('Annual Income (k$) (scaled)')
plt.show()
linked = linkage(X, method='ward')
plt.figure(figsize=(10, 7))
dendrogram(linked, orientation='top', distance_sort='descending', show_leaf_counts=True)
plt.title('Dendrograma de Clustering Jerárquico')
plt.show()
plt.scatter(X[:, 0], X[:, 1], c=hierarchical.labels_, cmap='rainbow')
plt.title('Clustering Jerárquico')
plt.xlabel('Age (scaled)')
plt.ylabel('Annual Income (k$) (scaled)')
plt.show()
print("Resultados de K-means:")
```

```
print(f"Coeficiente de Silhouette: {silhouette_kmeans}")
print(f"Índice de Calinski-Harabasz: {ch_kmeans}")
print(f"Índice de Davies-Bouldin: {db_kmeans}")
print("Resultados de Clustering Jerárquico:")
print(f"Coeficiente de Silhouette: {silhouette_hierarchical}")
print(f"Índice de Calinski-Harabasz: {ch_hierarchical}")
print(f"Índice de Davies-Bouldin: {db_hierarchical}")
```

BANNERGRAVING







EVIDENCIAS

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.metrics import silhouette_score, calinski_harabasz_score, davies_bouldin_score
from sklearn.preprocessing import StandardScaler
from scipy.cluster.hierarchy import dendrogram, linkage
```

```
data = pd.read_csv('./1.csv')
```

```
print(data.head())
```

```
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
```

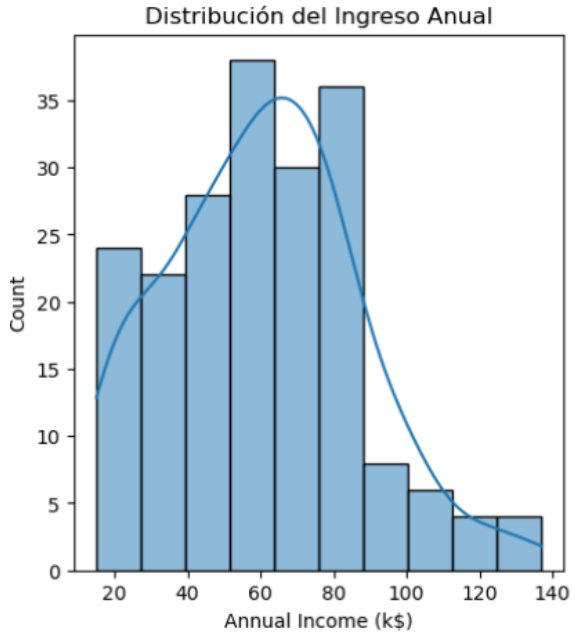
```
print(data.describe())
```

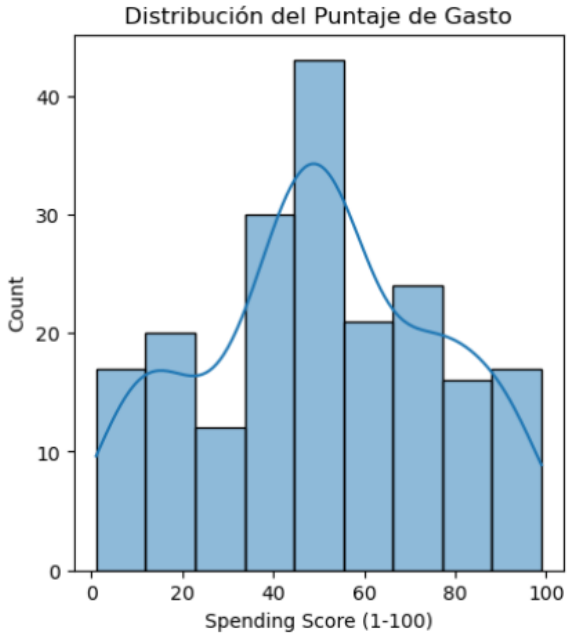
```
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
count  200.000000  200.000000          200.000000              200.000000
mean   100.500000   38.850000           60.560000               50.200000
std     57.879185   13.969007           26.264721               25.823522
min      1.000000   18.000000           15.000000                1.000000
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
max    200.000000   70.000000          137.000000               99.000000
```

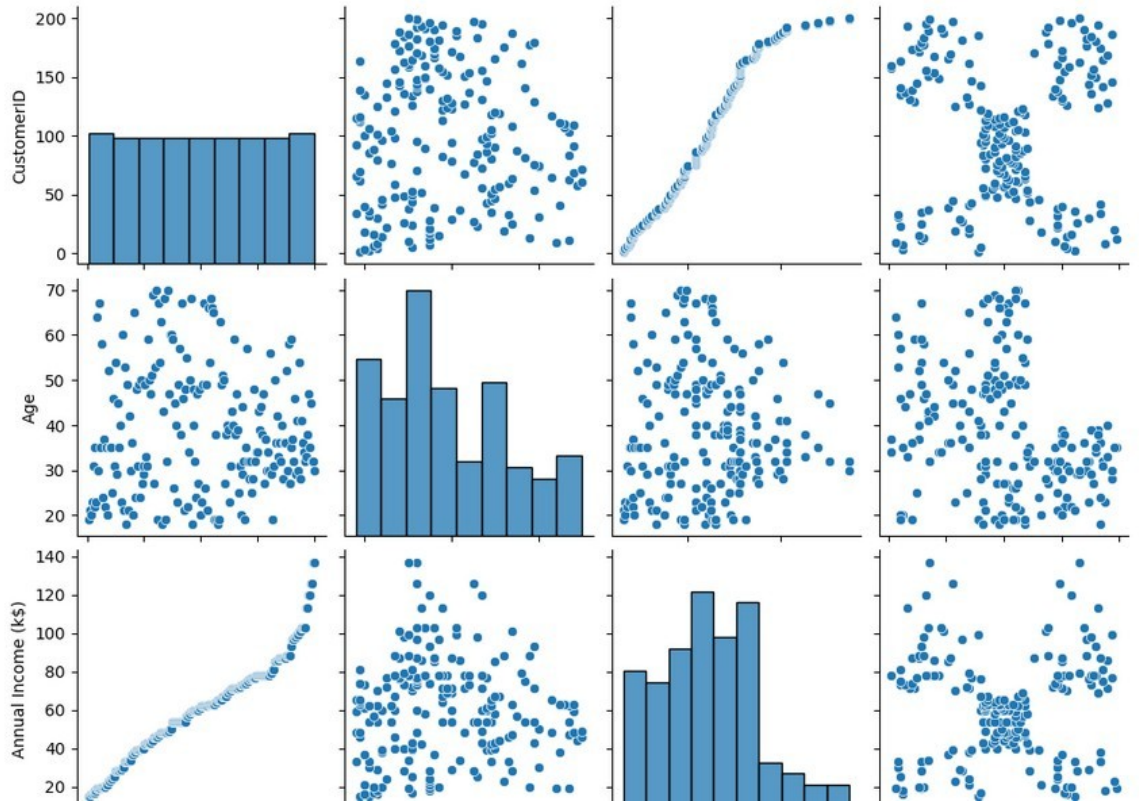
```
sns.pairplot(data)
plt.show()
```

```
plt.subplot(1, 2, 2)
sns.histplot(data['Spending Score (1-100)'], kde=True)
plt.title('Distribución del Puntaje de Gasto')
plt.show()
```

```
C:\Users\proyectos.ingenieria\AppData\Local\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\proyectos.ingenieria\AppData\Local\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```







```
C:\Users\proyectos.ingenieria\AppData\Local\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: Future
Warning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values
to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\proyectos.ingenieria\AppData\Local\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: Future
Warning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values
to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
C:\Users\proyectos.ingenieria\AppData\Local\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: Future
Warning: use_inf_as_na option is deprecated and will be removed in a future version. Convert inf values
to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

```
print("Resultados de K-means:")
print(f"Coeficiente de Silhouette: {silhouette_kmeans}")
print(f"Índice de Calinski-Harabasz: {ch_kmeans}")
print(f"Índice de Davies-Bouldin: {db_kmeans}")

print("Resultados de Clustering Jerárquico:")
print(f"Coeficiente de Silhouette: {silhouette_hierarchical}")
print(f"Índice de Calinski-Harabasz: {ch_hierarchical}")
print(f"Índice de Davies-Bouldin: {db_hierarchical}")
```

```
Resultados de K-means:
Coeficiente de Silhouette: 0.41664341513732767
Índice de Calinski-Harabasz: 125.10094020060956
Índice de Davies-Bouldin: 0.874551051002418
Resultados de Clustering Jerárquico:
Coeficiente de Silhouette: 0.39002826186267214
Índice de Calinski-Harabasz: 107.82656032570374
Índice de Davies-Bouldin: 0.9162886109753661
```

LINK GITHUB :

EVIDENCIAS JOSE LUIS SIERRA RAMIREZ

Dataset k-means e hierarchical clustering

Jose Luis Sierra Ramirez

Analisis de Datos

In [3]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.metrics import silhouette_score, calinski_harabasz_score
from sklearn.decomposition import PCA
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
/tmp/ipykernel_71016/1534492961.py in <module>
      2 import numpy as np
      3 import matplotlib.pyplot as plt
----> 4 import seaborn as sns
      5 from sklearn.preprocessing import StandardScaler
      6 from sklearn.cluster import KMeans, AgglomerativeClustering

ModuleNotFoundError: No module named 'seaborn'
```

In [4]:
```
data = pd.read_csv('./Mall_Customers.csv')
```

1. Análisis Exploratorio de Datos (EDA)

In [7]:
```
print(data.head())
print(data.info())
print(data.describe())
sns.pairplot(data)
plt.show()
```

```
   CustomerID  Gender  Age  Annual Income (k$)  Spending Score (1-100)
0           1    Male   19                  15                      39
1           2    Male   21                  15                      81
2           3  Female   20                  16                       6
3           4  Female   23                  16                      77
4           5  Female   31                  17                      40
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
None
       CustomerID         Age  Annual Income (k$)  Spending Score (1-100)
count  200.000000  200.000000          200.000000              200.000000
mean   100.500000   38.850000           60.560000               50.200000
std     57.879185   13.969007           26.264721               25.823522
min      1.000000   18.000000           15.000000                1.000000
25%     50.750000   28.750000           41.500000               34.750000
50%    100.500000   36.000000           61.500000               50.000000
75%    150.250000   49.000000           78.000000               73.000000
max    200.000000   70.000000          137.000000               99.000000
```

2. Preprocesamiento de Datos

Manejo de valores faltantes

In [8]:
```
data.dropna(inplace=True)
```

Transformación de variables categóricas

In [9]:
```
data = pd.get_dummies(data, columns=['Age'])
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_69283/1296570928.py in <module>
      1 data = pd.get_dummies(data, columns=['Age'])
----> 2 scaler = StandardScaler()
      3 data_scaled = scaler.fit_transform(data)

NameError: name 'StandardScaler' is not defined
```

Escalado de variables numéricas

In [ ]:
```
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

3. Selección de Características (opcional)

4. Entrenamiento del Modelo

K-Means

```
In [2]: kmeans = KMeans(n_clusters=5, random_state=42)
        kmeans.fit(data_scaled)

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_71016/1494699066.py in <module>
----> 1 kmeans = KMeans(n_clusters=5, random_state=42)
      2 kmeans.fit(data_scaled)

NameError: name 'KMeans' is not defined
```

Hierarchical Clustering

Puedes elegir diferentes métodos de enlace y métricas de distancia

```
In [11]: agg_clustering = AgglomerativeClustering(n_clusters=5, linkage='ward', affinity='euclidean')
         agg_clusters = agg_clustering.fit_predict(data_scaled)

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_69283/2809627540.py in <module>
----> 1 agg_clustering = AgglomerativeClustering(n_clusters=5, linkage='ward', affinity='euclidean')
      2 agg_clusters = agg_clustering.fit_predict(data_scaled)

NameError: name 'AgglomerativeClustering' is not defined
```

5. Evaluación del Modelo

K-Means

```
In [5]: silhouette_kmeans = silhouette_score(data_scaled, kmeans.labels_)
        calinski_kmeans = calinski_harabasz_score(data_scaled, kmeans.labels_)
        print("K-Means Silhouette Score:", silhouette_kmeans)
        print("K-Means Calinski-Harabasz Score:", calinski_kmeans)

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_71016/1444819522.py in <module>
----> 1 silhouette_kmeans = silhouette_score(data_scaled, kmeans.labels_)
      2 calinski_kmeans = calinski_harabasz_score(data_scaled, kmeans.labels_)
      3 print("K-Means Silhouette Score:", silhouette_kmeans)
      4 print("K-Means Calinski-Harabasz Score:", calinski_kmeans)

NameError: name 'silhouette_score' is not defined
```

Hierarchical Clustering

```
In [13]: silhouette_agg = silhouette_score(data_scaled, agg_clusters)
         calinski_agg = calinski_harabasz_score(data_scaled, agg_clusters)
         print("Hierarchical Clustering Silhouette Score:", silhouette_agg)
         print("Hierarchical Clustering Calinski-Harabasz Score:", calinski_agg)

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_69283/591503456.py in <module>
```

Visualización en 2D utilizando PCA

```
In [14]: pca = PCA(n_components=2)
         data_pca = pca.fit_transform(data_scaled)

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_69283/1859534815.py in <module>
----> 1 pca = PCA(n_components=2)
      2 data_pca = pca.fit_transform(data_scaled)

NameError: name 'PCA' is not defined
```

Scatter plot K-Means

```
In [15]: plt.figure(figsize=(10, 6))
         sns.scatterplot(x=data_pca[:,0], y=data_pca[:,1], hue=kmeans.labels_, palette='viridis', legend='full')
         plt.title('K-Means Clustering')
         plt.xlabel('Componente Principal 1')
         plt.ylabel('Componente Principal 2')
         plt.show()

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_69283/2733729864.py in <module>
      1 plt.figure(figsize=(10, 6))
----> 2 sns.scatterplot(x=data_pca[:,0], y=data_pca[:,1], hue=kmeans.labels_, palette='viridis', legend='full')
      4 plt.xlabel('Componente Principal 1')
      5 plt.ylabel('Componente Principal 2')

NameError: name 'sns' is not defined

<Figure size 720x432 with 0 Axes>
```

Scatter plot Hierarchical Clustering

```
In [16]: plt.figure(figsize=(10, 6))
         sns.scatterplot(x=data_pca[:,0], y=data_pca[:,1], hue=agg_clusters, palette='viridis', legend='full')
         plt.title('Hierarchical Clustering')
         plt.xlabel('Componente Principal 1')
         plt.ylabel('Componente Principal 2')
         plt.show()

---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_69283/1286674820.py in <module>
      1 plt.figure(figsize=(10, 6))
----> 2 sns.scatterplot(x=data_pca[:,0], y=data_pca[:,1], hue=agg_clusters, palette='viridis', legend='full')
      3 plt.title('Hierarchical Clustering')
      4 plt.xlabel('Componente Principal 1')
      5 plt.ylabel('Componente Principal 2')

NameError: name 'sns' is not defined

<Figure size 720x432 with 0 Axes>
```

CODIGO JOSE LUIS SIERRA RAMIREZ

```
# Dataset k-means e hierarchical clustering
# Jose Luis Sierra Ramirez
# Analisis de Datos
# Importar bibliotecas
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, AgglomerativeClustering
from sklearn.metrics import silhouette_score, calinski_harabasz_score
from sklearn.decomposition import PCA

# Cargar datos
data = pd.read_csv('./Mall_Customers.csv')

# 1. Análisis Exploratorio de Datos (EDA)
print(data.head())
print(data.info())
print(data.describe())

# 2. Preprocesamiento de Datos
# Manejo de valores faltantes
data.dropna(inplace=True)
# Transformación de variables categóricas
data = pd.get_dummies(data, columns=['columna_categorica'])
# Escalado de variables numéricas
scaler = StandardScaler()
data_scaled = scaler.fit_transform(data)
```

```
# 3. Selección de Características (opcional)
# Si se desea seleccionar características, hacerlo aquí

# 4. Entrenamiento del Modelo
# K-Means
kmeans = KMeans(n_clusters=5, random_state=42)
kmeans.fit(data_scaled)

# Hierarchical Clustering
# Puedes elegir diferentes métodos de enlace y métricas de distancia
agg_clustering = AgglomerativeClustering(n_clusters=5, linkage='ward', affinity='euclidean')
agg_clusters = agg_clustering.fit_predict(data_scaled)

# 5. Evaluación del Modelo
# K-Means
silhouette_kmeans = silhouette_score(data_scaled, kmeans.labels_)
calinski_kmeans = calinski_harabasz_score(data_scaled, kmeans.labels_)
print("K-Means Silhouette Score:", silhouette_kmeans)
print("K-Means Calinski-Harabasz Score:", calinski_kmeans)

# Hierarchical Clustering
silhouette_agg = silhouette_score(data_scaled, agg_clusters)
calinski_agg = calinski_harabasz_score(data_scaled, agg_clusters)
print("Hierarchical Clustering Silhouette Score:",
```

```
silhouette_agg)
print("Hierarchical
Clustering Calinski-
Harabasz Score:",
calinski_agg)

# 6. Visualización
de Resultados
# Visualización en
2D utilizando PCA
pca =
PCA(n_components
=2)
data_pca =
pca.fit_transform(da
ta_scaled)

# Scatter plot K-
Means
plt.figure(figsize=(1
0, 6))
sns.scatterplot(x=da
ta_pca[:,0],
y=data_pca[:,1],
hue=kmeans.labels_
, palette='viridis',
legend='full')
plt.title('K-Means
Clustering')
plt.xlabel('Compone
nte Principal 1')
plt.ylabel('Compone
nte Principal 2')
plt.show()

# Scatter plot
Hierarchical
Clustering
plt.figure(figsize=(1
0, 6))
sns.scatterplot(x=da
ta_pca[:,0],
y=data_pca[:,1],
hue=agg_clusters,
palette='viridis',
legend='full')
plt.title('Hierarchica
l Clustering')
plt.xlabel('Compone
nte Principal 1')
plt.ylabel('Compone
nte Principal 2')
plt.show()
```

- **TRABAJO EN GRUPO**

Link https://github.com/Josedeath1989/Tarea4--JUAN_CAMILO_CHAVES_HERNANDEZ.git

Link https://github.com/Josedeath1989/Tarea4--JOSE_LUIS_SIERRA_RAMIREZ--.git

CONCLUSIONES

Conclusion1:Por mi parte el desarrollo de esta actividad fue algo confusa debido a que la explicación fue buena pero el acercamiento con la herramienta fue complejo de comprender, y el análisis de datos no se acercó al resultado, entiendo que carezco de algún error en el proceso del análisis del ejercicio pero estuvo bastante complejo la interpretación debido a que la presentación tiene varios errores que no fueron suministrados.

Conclusion2: Conclusión compleja ya que se me ha ocurrido a este trabajo, es la formación académica y de los completos de refuerzo fueron buenos, lo cual no me pareció para la parte práctica baja, ya que debido a muchos problemas con la herramienta se debería profundizar a detalle.

BIBLIOGRAFIA

- **Holmes, D. E. (2018). *Big Data: una breve introducción*. (P 69-125). Antoni Bosch editor. https://elibro- net.bibliotecavirtual.unad.edu.co/es/ereader/unad/122682?page=69**
- **Casas Roma, J. Nin Guerrero, J. & Julbe López, F. (2019). *Big data: análisis de datos en entornos masivos.* (P 81-116). Editorial UOC. https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/117744?page=81**
- **López Murphy, J. J. & Zarza, G. (2017). *La ingeniería del big data: cómo trabajar con datos.* (P 127-146). Editorial UOC. https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unad/59093?page=127**
- **deRoos, D. (2014). *Hadoop For Dummies.*(P 53-68). Wiley. https://elibro-net.bibliotecavirtual.unad.edu.co/es/ereader/unadenglish/185161?page=67**