APRENDIZAJE AUTOMÁTICO AVANZADO

INFORME TÉCNICO UNIDAD III – APRENDIZAJE NO SUPERVISADO

PRESENTADO POR

Daniel José Rueda Lobato

Diego Andrés Valderrama Laverde

Catalina Piedrahita Jaramillo

PROFESOR:

José Lisandro Aguilar Castro

UNIVERSIDAD EAFIT

MEDELLÍN

MAESTRÍA EN CIENCIAS DE LOS DATOS Y ANALÍTICA

SEPTIEMBRE 2020

1. OBJETIVO DE LA ITERACIÓN

El objetivo de este trabajo es aplicar a un conjunto de datos técnicas de aprendizaje no supervisado, para identificar si las agrupaciones con relación a la variable respuesta son claramente notorias cuando esta no está de manifiesto para el entrenamiento de modelos.

2. CONTEXTUALIZACIÓN DEL PROBLEMA

La base de datos utilizada proviene de Kaggle y es una extracción del Censo de 1994 de Estados Unidos, la cual trae una variable respuesta con la que se pretende identificar si un individuo gana más de 50 mil dólares al año a partir de características demográficas como el nivel de educación, la edad, el género, la ocupación, entre otras. En este ejercicio utilizando técnicas de aprendizaje no supervisado, se ha retirado de la base de datos la variable respuesta, con el objetivo de identificar si al momento de realizar agrupaciones sobre los individuos se presenta la segmentación en relación a la ganancia anual de 50 mil dólares, ya que se quería hacer un análisis de los resultados de aprendizaje supervisado vs no supervisado, además que se pretendía realizar una evaluación de la exactitud de los modelos basados en las variables respuesta que contiene la base de datos.

3. DISEÑO DEL MODELO

La base de datos es de corte transversal de individuos y cuenta con 15 variables descriptoras, de las cuales 6 son cuantitativas y 9 son cualitativas, y hay 48 842 registros. La variable objetivo es *income*, que tiene dos etiquetas: una si gana más de 50 mil dólares al año, y otra si no. Se comenta esta variable, aunque no se tiene en cuenta para el cálculo de los métodos no supervisados; solo para validar exactitud. La Tabla 1 muestra la caracterización de las variables y una descripción detallada de cada una.

Tabla 1. Caracterización y explicación de las variables

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 48842 entries, 0 to 48841
Data columns (total 15 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             48842 non-null  int64
 1   capital-gain    48842 non-null  int64
 2   capital-loss    48842 non-null  int64
 3   hours-per-week  48842 non-null  int64
 4   educational-num 48842 non-null  int64
 5   workclass       48842 non-null  object
 6   fnlwgt          48842 non-null  int64
 7   education       48842 non-null  object
 8   marital-status  48842 non-null  object
 9   occupation      48842 non-null  object
 10  relationship    48842 non-null  object
 11  race            48842 non-null  object
 12  gender          48842 non-null  object
 13  native-country  48842 non-null  object
 14  income          48842 non-null  object
dtypes: int64(6), object(9)
memory usage: 5.6+ MB
```

Explicación de cada una de las variables

- Age: continuous. (Edad)
- Workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked. (Tipo de Trabajo)
- fnlwgt: continuous. (Factor de Expansión)
- Education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool. (Nivel Educativo)
- Education-num: continuous (Número de años que ha estudiado)
- Marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse. (Estado civil)
- Occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces. (Ocupación)
- Relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried. (Posición en la familia)
- Race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black. (Raza)
- Sex: Female, Male. (Sexo)
- Capital-gain: continuous (Ganancia capital)
- Capital-loss: continuous. (Pérdida capital)
- hours-per-week: continuous. (Horas de trabajo por semana)
- Native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands. (Pais de Origen)

Para realizar la actividad de agrupación no supervisada primero es necesario identificar si existen datos faltantes o missing values. Como se pudo evidenciar visualmente los datos contaban con algunos valores desconocidos (con el símbolo ?) en tres variables. Dado que la cantidad de datos faltantes no es significativa, se optó por hacer una imputación. Específicamente, como los datos perdidos se encontraban en variables categóricas, se imputó la etiqueta más frecuente, es decir, la moda. La Gráfica 1 muestra cómo se realizó la identificación e imputación.

Gráfica 1. Identificación de Mising Values

```
#Reemplazamos el caracter "?" el cual previamente analizamos que es "Missing Value" por Null
df['workclass'].replace('?', np.NaN, inplace=True)
df['occupation'].replace('?', np.NaN, inplace=True)
df['native-country'].replace('?', np.NaN, inplace=True)
```

```
[15] #Imputación de datos en variable categoricas, usando la moda
     for df2 in [df]:
         df2['workclass'].fillna(df['workclass'].mode()[0], inplace=True)
         df2['occupation'].fillna(df['occupation'].mode()[0], inplace=True)
         df2['native-country'].fillna(df['native-country'].mode()[0], inplace=True)
```

Posteriormente, y antes de iniciar nuestro análisis de algoritmos no supervisados, se considera relevante buscar una técnica para describir nuestros datos en términos de nuevas variables, que permita reducir la dimensionalidad, y que además se puede aplicar a una combinación de variables numéricas y categóricas; encontramos entonces el método Análisis Factorial de Datos Mixtos (FAMD) que cumple con todas las características mencionadas.

Gráfica 2 Lógica de la Técnica FAMD

```
famd = prince.FAMD(
     n_components=104,
     n_iter=200,
     copy=True,
     check_input=True,
     engine='auto',
     random_state=42)
```

```
25] famd_1 = famd.fit(dff)
```

Algunos parámetros ingresados para la técnica de reducción de dimensionalidad fueron:

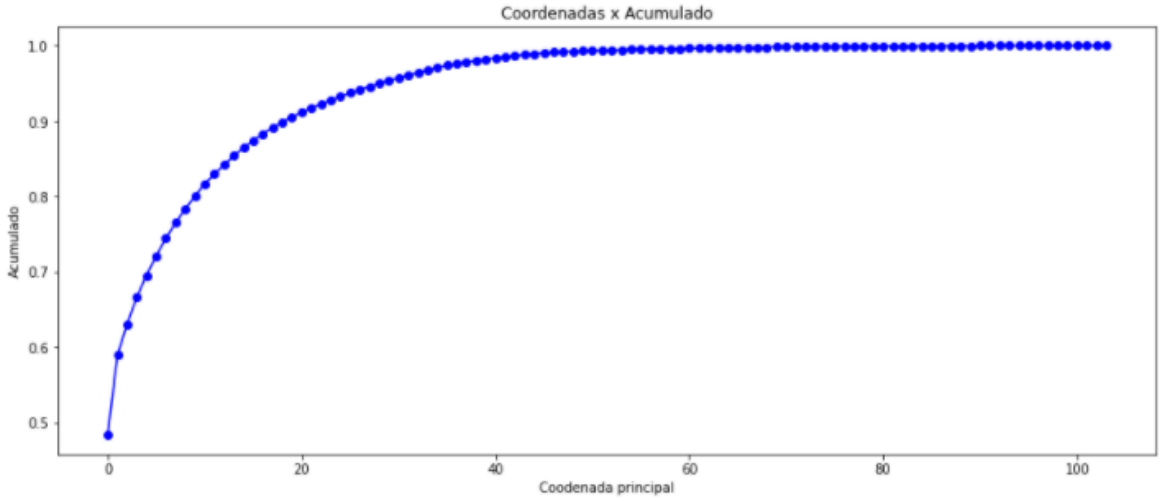
- Numero de componentes: 104. Es de aclarar que el método FAMD al permitir en su entrada variables categóricas, dentro de su lógica las convierte en dummies, por lo tanto, como ya habíamos estudiado nuestro base de datos en el apartado de ingeniería de características, sabíamos que la combinación máxima de variables en total era de 104.
- Numero de iteraciones: 200. Para la cantidad de datos y variables, consideramos un numero lógico y acorde.

Como resultado de la técnica, encontramos que las primeras 8 componentes ordenadas en importancia explican el 76% de la variabilidad de los datos, por tanto, se decide que nuestra nueva base de datos contendrá la información de las 8 primeras componentes como se puede ver en la Gráfica 3.

Gráfica 3 Componentes principales FAMD y grafico acumulado

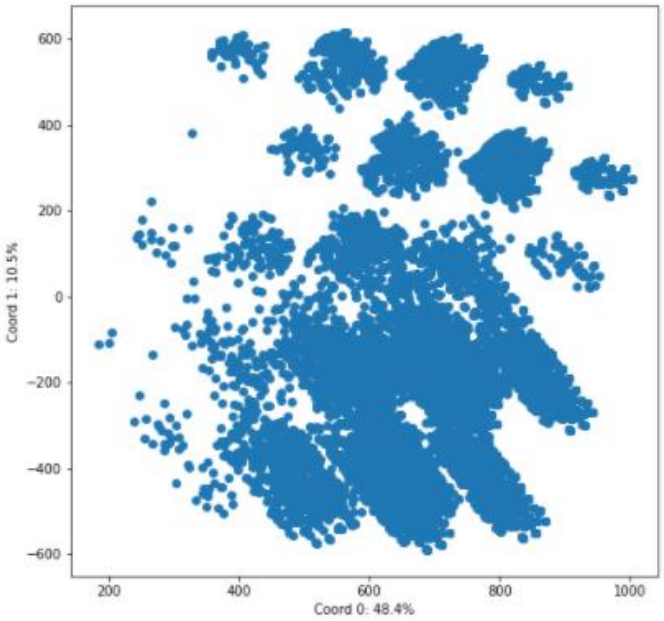
```
[27] #Se presenta la inercia explicada por cada coordenada (porcentaje de importancia de las variables)
     eigfamd_1=famd_1.explained_inertia_
     eigfamd_1[0:104]
```

```
[0.48416381803594255,
 0.10543965749470575,
 0.04116894251952647,
 0.035951869586370984,
 0.028757318843263154,
 0.025923672393749212,
 0.022976085354497824,
 0.0208318935429704 9,
```



Ahora bien, para darnos una idea gráfica de cómo se están comportando nuestras nuevas componentes, tomamos las dos con mayor relevancia, para ver su distribución en el plano. El resultado se aprecia en la Gráfica 4.

Gráfica 4. Grafica del conjunto de datos dos componentes principales



De manera empírica y a simple vista se pueden identificar grupos de puntos altamente sectorizados que serían posibles clústeres, pero teniendo en cuenta la problemática inicial que se planteaba en este conjunto de datos en la cual era necesario identificar los ingresos mayores y menores a 50 mil, si se observará la grafica sin contar con esta información inicial, no sería claro diferenciar 2 grupos

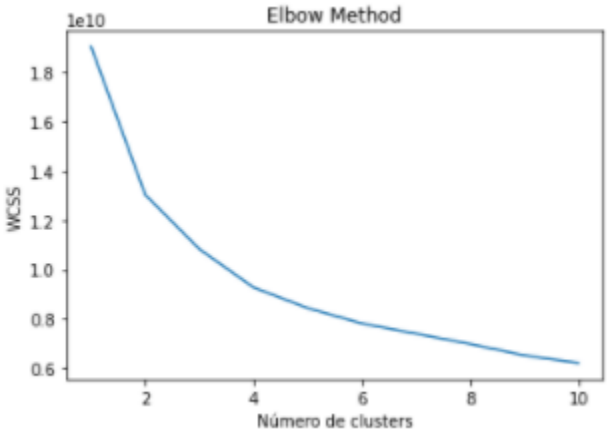
También como parte de nuestro análisis, en el notebook adjunto, para las tres componentes principales se generaron gráficos en 3D con diferentes vistas y un GIF animado (también en 3D y en movimiento) para que visualmente se pueda interpretar con un mayor nivel de detalle la dispersión de nuestros datos (ver adjuntos).

Algoritmos no supervisados empleados

3.1. K-Means:

Inicialmente para identificar el número óptimo de clústeres empleamos el método “Elbow”, el cual es la suma de la distancia cuadrada entre cada miembro del cluster y su centroide. De acuerdo con el resultado, el número de clústeres óptimo es de 2, lo que coincide con el conocimiento previo de que existen dos clases (o variables respuesta): ">50k" y "<=50k". El siguiente grafico muestra la gráfica de “codo” para la identificación del número de clústeres.

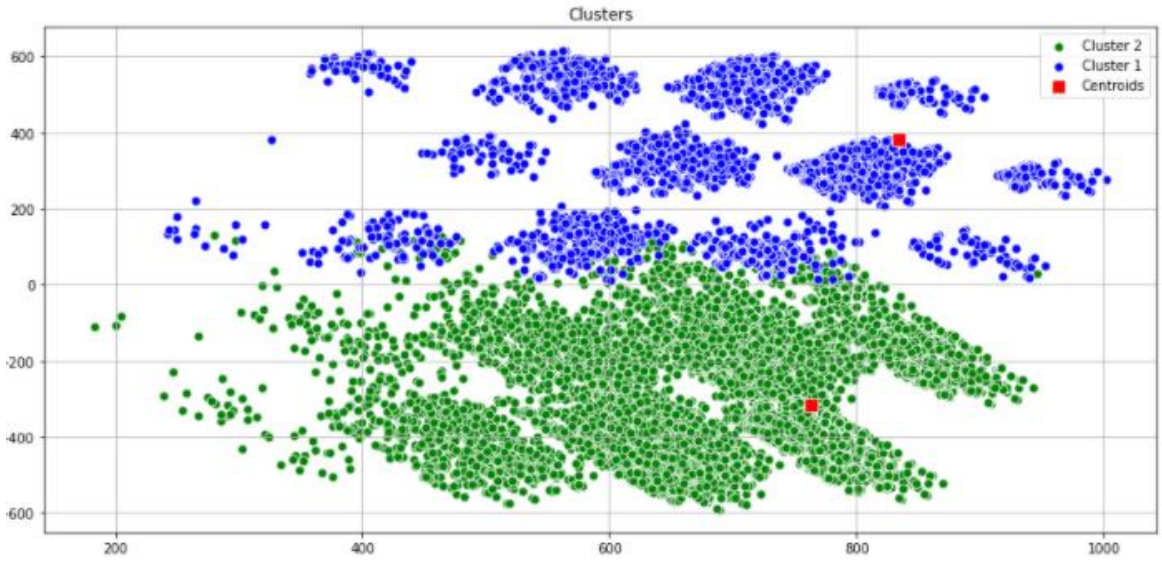
Gráfica 5 Grafica del método Elbow



Luego parametrizamos e instanciamos el modelo, utilizando como información de entrada la base de datos resultado del procedimiento FAMD, a continuación, mediante un gráfico y diferenciando los clústeres que el algoritmo identificó, se evidencian los resultados del K-Means. Cabe aclarar que como el conjunto de datos posee 8 variables, se están graficando las dos con más significancia.

Nótese también que, de color rojo, se identifican los centroides de cada uno de los clústeres que fueron generados producto de las 300 iteraciones que parametrizamos al ejecutar el proceso. A continuación, la gráfica.

Gráfica 6 Resultado grafico del KMeans



Una vez ejecutado el algoritmo de K-Means, es necesario evaluar que tan bien realizó la predicción de las agrupaciones. Para ello, existen métodos de evaluación **externos e internos**, y como en el conjunto de datos seleccionado poseemos la variable respuesta, se pueden emplear métodos de evaluación externo, el primer seleccionado es el "Índice de Rand", el cual calcula qué tan similares

son los clústeres (devueltos por el algoritmo de agrupamiento) a las clasificaciones de referencia. También se puede ver el índice Rand como una medida del porcentaje de decisiones correctas tomadas por el algoritmo. Esta nos arroja una exactitud del 74%. Lo cual consideramos una buena medida. El segundo método de evaluación externo seleccionado es la entropía, la cual es el grado de coincidencia de los clústeres a las clases ya definidas en los datos originales. Nótese que, para calcular la entropía total, es necesario inicialmente calcular la entropía de cada cluster. El resultado es de 0.60, que indica el nivel de coincidencia de los clústeres, es decir que los cluster conservan cierta superposición, la cual fue evidenciada empíricamente en las gráficas realizadas anteriormente.

También se evaluó un método interno, el "coeficiente de silueta", el cual contrasta la distancia media a elementos en el mismo grupo con la distancia media a elementos en otros grupos. Es una métrica limitada entre [-1,1]: -1 para la agrupación incorrecta y 1 para la agrupación muy densa (densa y bien separada), los valores alrededor de cero indican agrupaciones superpuestas. Para nuestro conjunto de datos el resultado del coeficiente de silueta es 0.30. Como se puede observar, no es una agrupación incorrecta (negativa), pero tampoco es una agrupación densa o separada. Este valor indica que es una agrupación con cierto grado de superposición como se pudo evidenciar empíricamente en las gráficas realizadas anteriormente.

A continuación, se muestra la tabla resumen con los resultados explicados.

Tabla 2. Evaluación del K-Means

Indicador	Resultado
Índice Rand	0.743
Entropía Total	0.6063
-Entropía clúster 0	0.3050
-Entropía clúster 1	0.9988
Coeficiente de silueta	0.3016

3.2 DBSCAN:

Al tratar con grupos espaciales de diferente densidad, tamaño y forma, podría ser un desafío detectar el grupo de puntos. La tarea puede ser aún más complicada si los datos contienen ruido y valores atípicos. Martin Ester propuso la agrupación espacial de aplicaciones con ruido basada en la densidad (DBSCAN). A continuación un ejemplo de puntos en un espacio con cierta forma sobre el cual DBSCAN podría funcionar muy bien.

Gráfica 7 Puntos DBSCAN

Si bien los datos de nuestro problema espacialmente no contienen una forma específica y densa, probemos el algoritmo para analizar desde otro punto de vista nuestra base de datos.

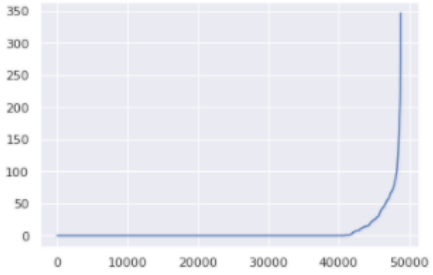
Gráfica 8. Parametrización DBSCAN

Inicialmente buscamos el epsilon mas indicado para nuestros datos https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc

```
[62] #Se calculan distancias para cada punto por medio del algoritmo de vecinos más cercanos.
     neigh = NearestNeighbors(n_neighbors=2)
     nbrs = neigh.fit(df_res_famd)
     distances, indices = nbrs.kneighbors(df_res_famd)
```

```
[63] #Se pintan las distancias y donde la curvatura sea mas extrema, se identifica el epsilon. En nuestro caso será 50
     distances = np.sort(distances, axis=0)
     distances = distances[:,1]
     plt.plot(distances)
```

```
[<matplotlib.lines.Line2D at 0x7f615a0326a0>]
```



Como se puede ver en la anterior imagen, inicialmente se calculan distancias para cada punto por medio del algoritmo de vecinos más cercanos. Posteriormente se pintan las distancias y donde la curvatura sea más extrema, se identifica el épsilon. En nuestro caso será 50.

Luego generamos el algoritmo con el épsilon recomendado y probamos dos cantidades de vecinos (100 y 700). Como puede observarse en los resultados de DBSCAN, los datos de ruido (marcados con -1) son muchísimos. Superan en ambos casos el 90% del total de datos. Por otro lado, si se disminuye aún más el límite de vecinos (por ejemplo 50), podríamos tener un número menor de ruido, pero se incrementa considerablemente el número de clústeres, y debido a la naturaleza de nuestro conjunto de datos y la variable resultado esperada no es conveniente. En conclusión, debido a la distribución y valor de las variables de nuestra base de datos, este algoritmo no es conveniente.

Tabla 3. Resultados DBSCAN

100 vecinos

cluster	
-1	25840
0	1431
1	213
2	269
3	154
...	...
83	111
84	124
85	108
86	106
87	155

700 vecinos

cluster	
-1	45805
0	801
1	1431
2	805

3.3. Agrupación Jerárquica Aglomerativa:

En esta técnica, inicialmente cada punto de datos se considera como un grupo individual. En cada iteración, los grupos similares se fusionan con otros grupos hasta que se forman uno o K grupos.

A continuación algunos pasos básicos para el algoritmo:

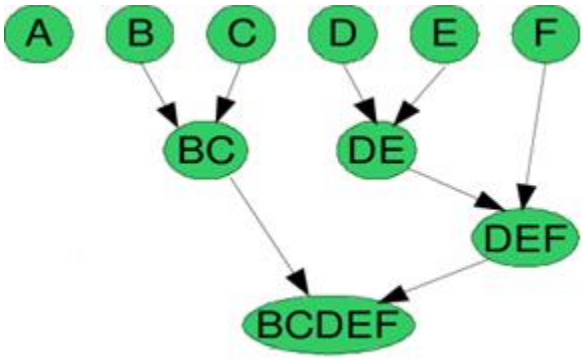
- Calcule la matriz de proximidad
- Deje que cada punto de datos sea un grupo
- Repetir: fusionar los dos grupos más cercanos y actualizar la matriz de proximidad
- Hasta que solo queden grupos altamente relacionados

La operación clave es el cálculo de la proximidad de dos grupos

Para entenderlo mejor, veamos una representación pictórica de la Técnica de agrupamiento jerárquico aglomerativo. Digamos que tenemos seis puntos de datos {A, B, C, D, E, F}.

- Paso 1: En el paso inicial, calculamos la proximidad de puntos individuales y consideramos los seis puntos de datos como grupos individuales como se muestra en la imagen a continuación.

Gráfica 9 Ejemplo agrupamiento jerárquico aglomerativo



- Paso 2: En el paso dos, los grupos similares se fusionan y se forman como un solo grupo. Consideremos que B, C y D, E son grupos similares que se fusionaron en el paso dos. Ahora, tenemos cuatro grupos que son A, BC, DE, F.
- Paso 3: Calculamos de nuevo la proximidad de nuevos grupos y fusionamos los grupos similares para formar nuevos grupos A, BC, DEF.
- Paso 4: Calcule la proximidad de los nuevos clústeres. Los grupos DEF y BC son similares y se fusionaron para formar un nuevo grupo. Ahora quedan dos grupos A, BCDEF.

Para nuestra base de datos, debido a que con toda la cantidad de datos (48842) Google Colab colapsa por temas de memoria RAM, se toman los primeros 30.000 registros para realizar el análisis.

Como se puede ver en la siguiente imagen, el algoritmo categorizó adecuadamente 16136 personas que tenían un salario inferior a 50 mil dólares, y 5421 que tenían un salario superior a este valor. Además, tiene una pureza de 76% y un coeficiente de silueta de 29%. Resultados muy similares e incluso un poco superiores al algoritmo de K-Means.

En las tablas a continuación se resumen los resultados

Tabla 4 Resultado. Cantidad de observaciones por clúster

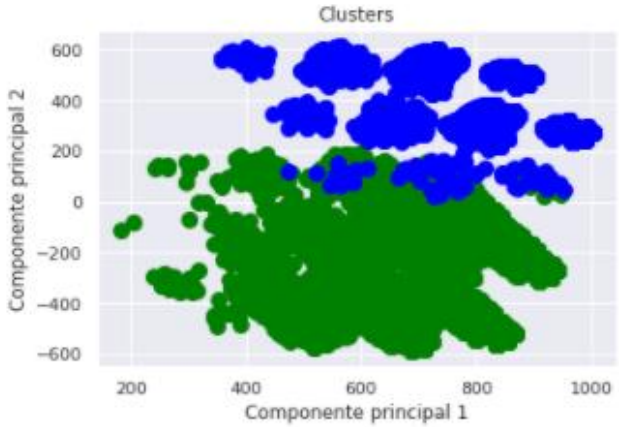
Clúster	*Income*	Conteo
0	0	16136
0	1	1709
1	0	6734
1	1	5421

Tabla 5 Evaluación del Agrupación Jerárquica Aglomerativa

Indicador	Resultado
Pureza	0.7623
Coeficiente de silueta	0.2984

En la Gráfica 10, utilizando las dos primeras componentes, se observa el resultado de los clústeres obtenidos mediante la técnica de Agrupación Jerárquica Aglomerativa

Gráfica 10 Clústeres con técnica de Agrupación Jerárquica Aglomerativa

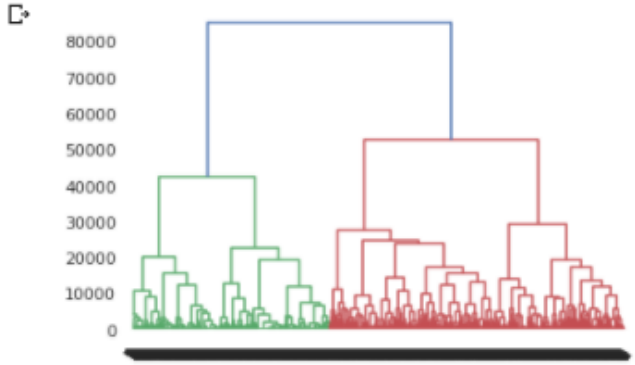


Como puede observarse en el gráfico, los resultados son concluyentes. Aunque se ven superpuestos, hay dos clústeres claramente definidos, muy similar a los resultados obtenidos con el algoritmo de K-Means.

Por último, graficamos el dendrograma que también muestra de otra forma visual el comportamiento de los clústeres.

Gráfica 11 Dendrograma con técnica de Agrupación Jerárquica Aglomerativa

```
[77] dendrogram(model2,
                labels=None,
                leaf_rotation=150,
                leaf_font_size=3,
     )
     plt.show()
```



4. ANÁLISIS DE RESULTADOS

Este trabajo buscó determinar si un individuo gana más de 50 mil dólares al año a partir de características demográficas como en el nivel de educación, la edad, el género, la ocupación, entre otras. Se utilizó una base de datos proveniente de Kaggle, que es una extracción del Censo 1994 de Estados Unidos. Se aplicaron métodos no supervisados para identificar estas agrupaciones obteniendo resultados satisfactorios en 2 de los 3 métodos empleados. También se realizaron validaciones internas y externas de los resultados basándonos en la variable respuesta la cual se contaba por el ser la misma base de datos que se empleó en los métodos supervisados.

En conclusión, por la distribución de los datos el algoritmo de K-Means y el método de agrupación jerárquica aglomerativa arrojaron buenos resultados con precisiones superiores al 70%. Validando además la existencia de los 2 grupos en el conjunto de datos.

REFERENCIAS

Bhattacharyya, S. (8 de 06 de 2019). *Towards Data Science.* Obtenido de DBSCAN Algorithm: Complete Guide and Application with Python Scikit-Learn: https://towardsdatascience.com/dbscan-algorithm-complete-guide-and-application-with-python-scikit-learn-d690cbae4c5d

Google. (s.f.). *Kaggle*. Obtenido de https://www.kaggle.com/wenruliu/adult-income-dataset

Maklin, C. (30 de 06 de 2019). *Towards Data Science.* Obtenido de DBSCAN Python Example: The Optimal Value For Epsilon (EPS): https://towardsdatascience.com/machine-learning-clustering-dbscan-determine-the-optimal-value-for-epsilon-eps-python-example-3100091cfbc

Patlolla, C. R. (8 de 12 de 2018). *Towards Data Science.* Obtenido de Understanding the concept of Hierarchical clustering Technique: https://towardsdatascience.com/understanding-the-concept-of-hierarchical-clustering-technique-c6e8243758ec