Aprendizaje no supervisado

Lección 5: Estrategias para seleccionar el número adecuado de grupos

Introducción

Una compañía trasnacional de telefonía móvil desea renovar los enlaces de comunicación entre los clientes que viajan por jornadas amplias a otros países, por lo que te solicitan que hagas un análisis de las redes comunicación para identificar el tipo de conexión que requieren este tipo de clientes.

¿Cómo preparar los datos, de manera que funcionen algunos de los algoritmos de agrupamiento de datos más importantes?

Todo ese proceso es no supervisado, por lo que es difícil conocer con anterioridad cuáles son los resultados esperados o qué conclusiones pueden ayudar a tus líderes a mejorar la toma de decisiones.

Los investigadores usan medidas de calidad para medir numéricamente la importancia de los resultados. En este contexto, a esas medidas se les conoce como índices de validación de agrupamientos. En este tema estudiaremos esas medidas.

Los índices de validación de agrupamiento son un área muy activa de estudio y anualmente se publican muchos trabajos al respecto. Incluso, nuevos índices son publicados y analizados según su utilidad.

La razón por la que son muy importantes es porque a través de esos índices puedes elegir los mejores parámetros para tus algoritmos, seleccionar entre distintos algoritmos y ayudar a crear una historia para la toma de decisiones.

Índices de validación de los resultados de agrupamiento

Es primordial entender los conceptos fundamentales de este tema y para ello nos apoyaremos en una situación relacionada con el mercado inmobiliario.

¿Estás listo para analizar cómo agrupar datos de viviendas en Estados Unidos y cómo seleccionar la mejor opción entre varias?

La base de datos seleccionada se encuentra disponible en el sitio Kaggle a través de la url: https://www.kaggle.com/datasets/shree1992/housedata

¡Es momento de continuar en este camino de aprendizaje!

Las estrategias para validar los resultados de agrupamiento son las siguientes:

Distinguiendo los datos

En las situaciones anteriores, cargaste los datos desde *Scikit-learn*, pero en esta situación los datos están en un archivo con extensión ***.csv***, por lo que debes cargarlos usando **pandas**. Una manera de hacerlo es como muestra el siguiente código:

```
import pandas as pd
house = pd.read_csv("housedata.csv")
house.head()
```

Como resultado, puedes ver en la imagen un resumen de los primeros cinco datos, o filas, de la base de datos:

	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4

Figura 2. Resumen de la base de datos housedata.

Aunque en la imagen anterior solo aparecen las primeras variables, las últimas variables de estos datos tienen valores categóricos, esas variables son: ***street, city, statezip y country***.

Como ya sabes de temas anteriores, cuando hay datos categóricos y necesitas agruparlos, lo común es que transformes esos datos a numéricos. Seguramente escogerías ***OneHotEncoder*** para esa transformación, pero ese método de transformación tiene una característica importante.

Cuando una variable tiene muchos valores distintos, se generan o crean muchas variables, una para cada valor distinto, y eso puedo ser impracticable en ocasiones. Por lo tanto, por esta vez, y en situaciones similares, se prefiere transformar los datos con el método ***OrdinalEncoder***.

Al analizar los datos, ¿te has preguntado cuál es la variable más importante de todas? Todo parece indicar que el precio es precisamente la variable más importante, y sobre la cual se esperan algunas conclusiones. Es tu decisión como científico de datos si seleccionas algunas variables para el estudio, o todas, y si realizas algún otro preprocesamiento, en particular.

La siguiente imagen muestra el histograma de las primeras 16 variables contra el precio:

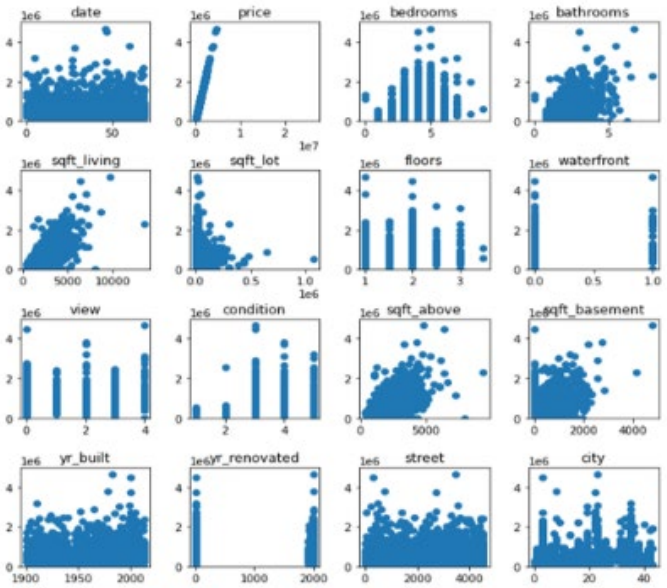


Figura 3. Histograma de las variables.

El código para transformar los datos y plotear los histogramas aparece a continuación:

```
house2 = house.copy()
for col in house2.columns:
  if house2[col].dtype == 'object':
    house2[col] = OrdinalEncoder().fit_transform(house2[[col]])
X_train = house2.to_numpy()
v = 0
fig, axs = plt.subplots(4, 4, figsize=(10, 10))
for i in range(4):
 for j in range(4):
  axs[i,j].scatter(X_train[:,v], X_train[:,1])
  axs[i,j].set_title(house.columns[v])
  axs[i,j].set_ylim([0,0.5e7])
  v+=1
plt.subplots_adjust(hspace=0.5, wspace=0.3)
plt.show()
```

¿Cuáles conclusiones obtienes al analizar los histogramas? La verdad, hay pocas conclusiones evidentes. Quizás una de ellas es que las variables relacionadas con el tamaño de las casas, las que tienen sqft en el nombre, están relacionadas con el precio. Esto es algo un tanto evidente, ya que es normal que casas más grandes cuesten más caro. Pero **¿cómo se pueden agrupar los datos y en cuántos grupos?** Seguramente puedes responder a la pregunta de cómo agrupar los datos, pero estudiaremos a continuación cómo seleccionar el número adecuado de grupos.

Identificando los índices de validación de agrupamientos

Los índices de validación de agrupamientos (Lee et al., 2018) son medidas de calidad que evalúan qué tan buenos son los resultados del agrupamiento, como ya se había mencionado. Por lo general se dividen en dos grandes grupos:

- **Índices externos.** Comparan los resultados contra un ***ground truth***, por ejemplo, las etiquetas. Como en la práctica casi nunca esto es posible, no estudiaremos esta categoría.
- **Índices internos.** Miden propiedades del agrupamiento como compacidad y separación de los grupos usando distancias.
    - ¿Cómo se miden?

Prestaremos especial atención a los índices internos. Por lo general, las propiedades que estos índices evalúan son propiedades relacionadas con la noción que normalmente tenemos de buenos agrupamientos. Por ejemplo, si un agrupamiento tiene grupos compactos significa que los datos dentro de un grupo son similares entre sí, y si hay separación entre los grupos significa que los objetos de cada grupo son diferentes a los de grupos vecinos.

Un índice de validación interno que mide estas propiedades y ha obtenido buenos resultados es el índice *Silhouette* (Mamat et al., 2018).

Este índice se define para cada dato y se compone de dos medidas:

Para un solo dato, la ecuación del cálculo de ***Silhouette*** se define como:

$$s = \frac{b - a}{\max(a,b)}$$

Para todo el agrupamiento, el índice calcula el promedio del cálculo para cada dato. Mayores valores del índice se relacionan con mejores resultados de agrupamiento. La imagen siguiente te muestra una representación gráfica de la definición del índice:

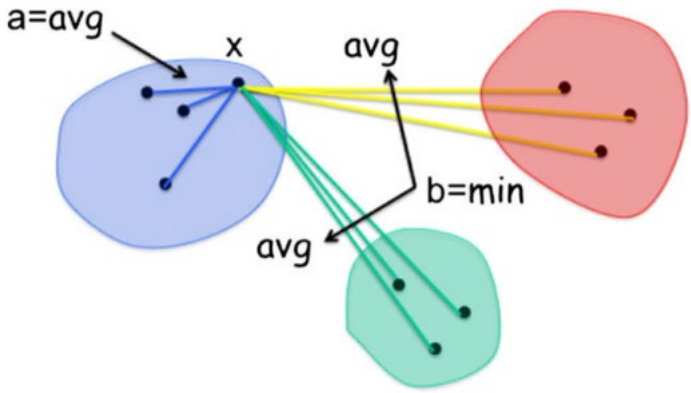


Figura 4. Representación gráfica del índice Silhouette.

1. Código para el agrupamiento de datos

No te preocupes si la definición del índice no la comprendes en su totalidad, lo importante es que aprenderás a usarlo a partir de una implementación del índice que aparece en *Scikit-learn*. Claro que, para poder utilizar el índice, los datos ya deben estar agrupados, y eso seguro ya sabes hacerlo con el siguiente código.

```
from sklearn import cluster
X = StandardScaler().fit_transform(X_train)
k_means = cluster.KMeans(n_clusters=2).fit(X)
y2 = k_means.predict(X)
```

Una vez que se han agrupado los datos, en la variable y^2 aparecen los grupos asignados, y el índice evaluará si la distribución de los datos, con estas nuevas etiquetas asignadas, cumplen las propiedades de compacidad y separación que mide.

2. Código para la implementación del índice

¿Cómo usamos entonces la implementación del índice? Primero hay que importar la implementación y luego 'llamarlo' pasando como parámetros los datos originales y las nuevas etiquetas, para que devuelva un valor numérico que representa la calidad del agrupamiento. Para ello, analiza el código siguiente:

```
from sklearn.metrics import silhouette_score
idx_2 = silhouette_score(X, y2)
```

3. Código para comparar resultado de datos

El resultado del índice se almacena en una variable, idx_2 en este caso, que toma el valor 0.172 como resultado. Valores cercanos a 1.0 indican los mejores resultados. Entonces, ¿es este un mal resultado? No necesariamente. Para poder decir que este resultado es bueno o no hay que compararlo con otros resultados sobre los mismos datos. Por ejemplo, ¿es mejor agrupar en dos grupos que en tres? ¿Crees que puedes hacer el código para ello? La respuesta se muestra a continuación:

```
k_means = cluster.KMeans(n_clusters=3).fit(X)
y3 = k_means.predict(X)
idx_3 = silhouette_score(X, y3)
```

Si analizas el resultado de la variable idx_3, notarás que el valor es menor: 0.161. Por lo tanto, puedes decir que es mejor agrupar en dos grupos que en tres grupos.

¡Consideración muy importante! Las propiedades que miden los índices de validación por lo general se encuentran mejor representadas mientras menos grupos se generen. Es decir, casi siempre agrupar en dos grupos producirá particiones más compactas y separadas y esta relación es inversamente proporcional al número de grupos. Por lo tanto, siguiendo la estrategia anterior, siempre saldrá dos como el mejor número de grupos. ¿Qué hacer entonces? Lo veremos en el siguiente subtema. Antes, estudiemos otro índice de validación.

Analizando el índice Calinski-Harabasz

Ya habíamos comentado que hay una gran cantidad de índices en la literatura, por lo que una buena estrategia es no quedarse con una sola opinión, sino revisar los resultados de varios índices. Aunque estudiarás el índice Calinski-Harabasz además del Silhouette, no quiere decir que solo recomendamos estos dos índices, solo que no se pueden presentar todos. Lo que sí recomendamos es incluir al índice Silhouette junto a cualquier otro conjunto de índices que se seleccionen para el análisis.

- **Diferencia entre los índices Silhouette y Calinski-Harabasz**

Realmente, los índices ***Silhouette*** y ***Calinski-Harabasz*** no son muy diferentes en su definición, ya que ambos miden las propiedades de compacidad y separación. Pero Calinski-Harabasz mide las propiedades de manera ligeramente diferente, y esa pequeña diferencia hace que los resultados de ambos sean en ocasiones muy distintos. Es por eso por lo que podemos sacar conclusiones analizando ambos índices.

- **Ecuación del índice Calinski-Harabasz**

Para un conjunto de datos de tamaño ***nE***, que se ha agrupado en ***k*** grupos, el índice ***Calinski-Harabasz*** se define como la relación entre el promedio de dispersión entre grupos, y la dispersión dentro del grupo, medida con la ecuación:

$$s = \frac{tr(B_k)}{tr(W_k)} \times \frac{n_E - k}{k - 1}$$

- **Índice con *Scikit-learn***

En la ecuación, ***tr(Bk)*** es la mínima distancia entre dos grupos, mientras que ***tr(Wk)*** es la máxima distancia en un mismo grupo. Otra vez, es posible que todos los detalles de la ecuación no estén claros, pero lo importante por ahora es aprender a usar el índice con ***Scikit-learn***.

Analiza y ejecuta el código siguiente:

```
from sklearn.metrics import calinski_harabasz_score
k_means_2 = cluster.KMeans(n_clusters=2).fit(X)
k_means_3 = cluster.KMeans(n_clusters=3).fit(X)
y_2 = k_means_2.predict(X)
y_3 = k_means_3.predict(X)
idx_2 = calinski_harabasz_score(X, y_2)
idx_3 = calinski_harabasz_score(X, y_3)
```

- **Valores del índice Calinski-Harabasz**

En las variables ***idx_2*** e ***idx_3*** aparecen los valores del índice ***Calinski-Harabasz*** para dos y tres grupos. Es probable que sus valores sean cercanos a 837 y 661 respectivamente. Estos valores son mayores en magnitud a ***Silhouette***, pero de la misma forma valores grandes indican mejores resultados. Por lo tanto, este índice tiene esa misma característica de favorecer los agrupamientos con menos grupos. En el siguiente subtema analizaremos este punto a detalle.

Gráficas *K-Elbow* y Gráfica *Silhouette-Plot*

Como recordarás en el ejemplo que revisaste al principio del tema relacionado con el mercado inmobiliario, con la agrupación de datos de viviendas, para seleccionar la mejor opción debes seleccionar el número de grupos antes de segmentar, lo cual es necesario para la mayoría de los algoritmos. Recuerda que los índices de validación son una buena herramienta para determinar ese número de grupos.

¿Estás listo para identificar adecuadamente las características de cada índice para poder usarlos de manera correcta?

¡Es momento de continuar en este camino de aprendizaje!

Recordando los índices *Silhouette* y *Calinski-Harabasz*
Los índices ***Silhouette*** y ***Calinski-Harabasz*** son muy usados para este propósito, pero ambos favorecen agrupaciones con pocos grupos. Es decir, si solo revisamos los valores de estos índices, el número de grupos donde aparentemente se obtienen los mejores resultados será un número de grupos pequeño. Esto se conoce como tendencia decreciente de los índices, o creciente en otros casos. El problema de esta tendencia es que recomienda todo el tiempo agrupar con dos o tres grupos incluso en situaciones donde un número mayor de grupos es recomendable.

¡Ahora, activa la reflexión con esta pregunta!, ¿qué puedes hacer entonces para que la decisión de en cuántos grupos segmentar no esté sesgada o influenciada por esta característica? La respuesta no es simple, pero puedes detenerte a pensar en ella, ya que la decisión de qué hacer en ocasiones puede depender de tu experiencia e intuición como científico de datos. No obstante, hay algunas estrategias ya establecidas en la literatura y bastante usadas en la práctica, que verás a continuación.

Analizando la gráfica *K-Elbow*
La gráfica *K-Elbow* plotea diferentes valores de un índice de validación, de manera que en el eje *x* aparecen los distintos números de grupos y en el eje *y* los valores del índice. En esta gráfica se puede ver claramente la

tendencia creciente o decreciente del índice. Se conoce también como gráfica de codo porque en ocasiones se rompe esa tendencia haciendo una especie de pico o codo en varios lugares de la gráfica. Esos puntos que rompen la tendencia se pueden asociar con valores adecuados para el número de grupos.

El código siguiente muestra cómo podemos hacer esta gráfica en *Python*:

```
y_preds = []
results = {}
for k in range(2, 11):
  k_means = cluster.KMeans(n_clusters=k).fit(X)
  y_pred = k_means.predict(X)
  y_preds.append(y_pred)
  results[k] = metrics.calinski_harabasz_score(X, y_pred)
clusters_counts = list(results.keys())
index_values = list(results.values())
```

- **Código de valores**

Al analizar el código seguramente notaste algunos detalles interesantes. Por ejemplo, el índice que se está usando es el Calinski-Harabasz. Además, se está agrupando para varios grupos y se almacena en un diccionario los valores del índice para el número de grupos correspondiente.

Ese diccionario tiene los valores necesarios para realizar la gráfica, a través del siguiente Código:

```
plt.plot(clusters_counts, index_values, 'o-')
plt.grid(True)
plt.title('"Maximize"')
plt.xlabel('Num Clusters')
plt.xticks(clusters_counts)
plt.ylabel('Calinski-Harabasz')
plt.show()
```

En la siguiente imagen podrás observar la gráfica *K-Elbow*. En esencia, es una 'línea' con tendencia decreciente que en algunos puntos cambia bruscamente formando picos o codos, como ya se mencionó.

En este ejemplo, ese pico aparece para seis grupos, por lo que podemos decir que seis grupos puede ser una buena partición de los datos, a pesar de que el valor del índice sea bastante menor que para dos grupos. En la literatura seguramente encontrarás método más avanzados para detectar los puntos donde cambia la pendiente, pero la mayoría de las veces esos picos o codos que detectas visualmente ya son de interés.

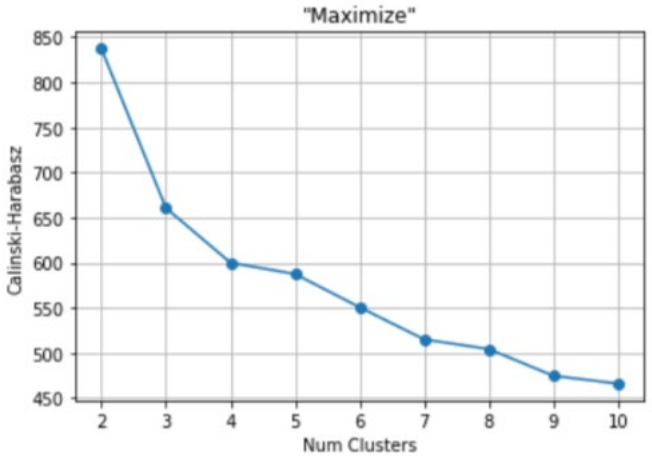


Figura 5. Gráfica *K-Elbow* con *Calinski-Harabasz* en la *BD housedata*.

En este ejemplo se ploteó la gráfica de 2 a 10 grupos. Una pregunta obligada es hasta cuál número de grupos debes construir la gráfica. La respuesta otra vez depende de cada situación. Una 'regla de dedo' es construir la gráfica hasta la raíz cuadrada de la cantidad de datos. No obstante, este número puede ser grande en ocasiones y bastaría hacerlo hasta un número de grupos que permita luego interpretar los resultados. A fin de cuentas, analizar más de una treintena de grupos ya puede ser impracticable.

Distinguiendo la Gráfica *Silhouette-Plot*

La mayoría de los índices de validación hacen una evaluación global del agrupamiento. Recuerda que no debemos decir que eso sea bueno o malo, simplemente es una característica de los índices.

1. Importancia del valor global

En ocasiones, de esa media global no se pueden extraer detalles sobre qué tan bien agrupado quedó cada grupo. Conocerlo es importante, por ejemplo, si un grupo reúne a los datos ruidosos, seguramente ese grupo no será compacto y afectará el valor final del índice. La consecuencia es que, a pesar de ese grupo, podemos estar en presencia de un excelente agrupamiento que no sea recomendado por el índice.

2. Índices para calcular el valor global

Algunos índices, para calcular el valor global, promedian los valores de cada grupo, por lo que pueden ser usados para separar ese análisis sobre qué tan bien agrupado quedó cada grupo. Es posible que hayas pensado cómo extraer esa información, por ejemplo, del índice *Calinski-Harabasz*, o de manera concreta, como extraerla de la implementación del índice en *Scikit-learn*. También te lo puedes preguntar sobre el índice *Silhouette*.

3. Características del índice silhouette

Una de las características que hace al índice *Silhouette* muy conocido es que no solo evalúa la calidad de cada grupo, sino que evalúa la calidad de… ¡cada dato! Esto es muy relevante porque nos puede dar información acerca de cuáles datos el índice no considera que deben estar en un grupo, aún cuando el grupo sea bastante bueno. Pero ¿cómo puedes obtener esa información? Precisamente a partir de la gráfica *Silhouette-Plot*.

4. ¿Cómo generar la gráfica Silhouette-Plot?

Si te fijas en la definición del índice *Silhouette*, cada dato aporta una pequeña cantidad al valor final del índice. Esto hace que sea costoso computacionalmente pero muy útil en la práctica. Afortunadamente, *Scikit-learn* nos deja obtener los valores de *Silhouette* para cada dato, lo que ayuda a conocer qué tan bien quedó el agrupamiento. El código para crear la gráfica *Silhouette-Plot* es un tanto complicado, pero lo explicaremos a continuación. Recuerda que lo importante primero es saber usar estas herramientas y ya luego puedes profundizar en cada tema.

Un ejemplo del código que se puede escribir para generar la gráfica *Silhouette-Plot* aparece a continuación:

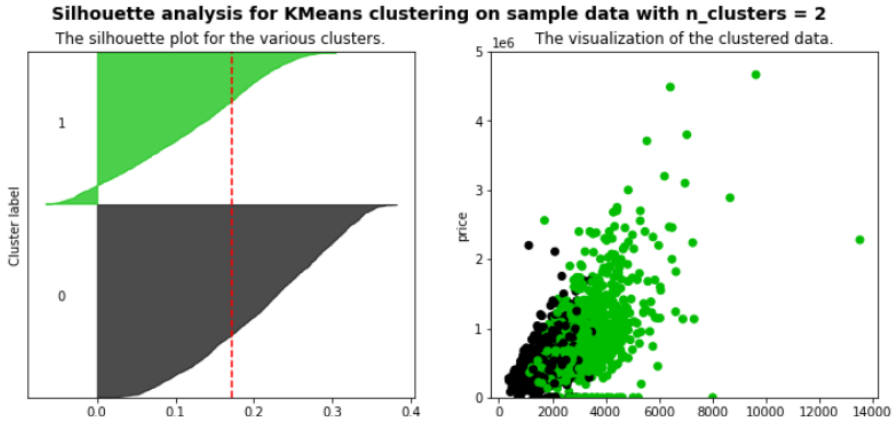
```
Import matplotlib.cm as cm
for k in [2, 3, 6]:
  fig, (ax0, ax1) = plt.subplots(1, 2)
  fig.set_size_inches(12, 5)
  # The (k+1)*10 is for inserting blank space between silhouette
  # plots of individual clusters, to demarcate them clearly.
  ax0.set_ylim([0, len(X) + (k + 1) * 10])
  k_means = cluster.KMeans(n_clusters=k)
  y_pred = k_means.fit_predict(X)
  silhouette_avg = metrics.silhouette_score(X, y_pred)
  sample_silhouette_values = metrics.silhouette_samples(X, y_pred)
  y_lower = 10
for i in range(k):
    # Aggregate the silhouette scores for samples belonging to
    # cluster i, and sort them
    ith_cluster_silhouette_values = \
       sample_silhouette_values[y_pred == i]
    ith_cluster_silhouette_values.sort()
    size_cluster_i = ith_cluster_silhouette_values.shape[0]
    y_upper = y_lower + size_cluster_i
    color = cm.nipy_spectral(float(i) / k)
    ax0.fill_betweenx(np.arange(y_lower, y_upper),
                      0, ith_cluster_silhouette_values,
                      facecolor=color, edgecolor=color, alpha=0.7)
# 2nd Plot showing the actual clusters formed
  colors = cm.nipy_spectral(y_pred.astype(float) / k)
  ax1.scatter(X_train[:, f], X_train[:, 1], c=colors)
  ax1.set_ylim(0, 0.5e7)
  ax1.set_title("The visualization of the clustered data.")
  ax1.set_xlabel(t)
  ax1.set_ylabel('price')
  plt.suptitle(("Silhouette analysis for KMeans clustering on sample data with n_clusters
= %d" % k), fontsize=14, fontweight='bold')
plt.show()
```

Al revisar el código y la imagen es probable que te preguntes, ¿por qué se grafica solo para 2, 3 y 6 grupos? Bueno, lo cierto es que en la práctica

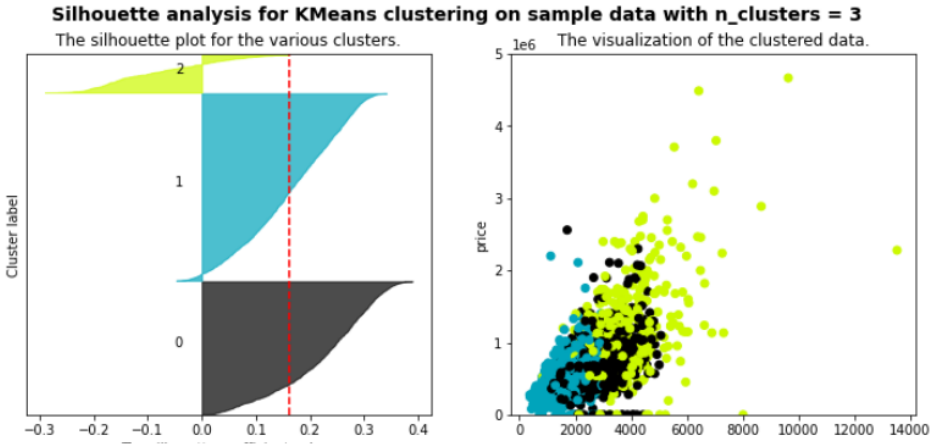
puedes construir la gráfica para todos los números de grupos que desees, pero aquí solo mostramos un resumen.

Analizar la información que se muestra en la imagen tiene también un carácter subjetivo. Quizás te parezca que no es tan simple en esta gráfica determinar cuál es el número de grupos adecuados. Veamos algunos detalles.

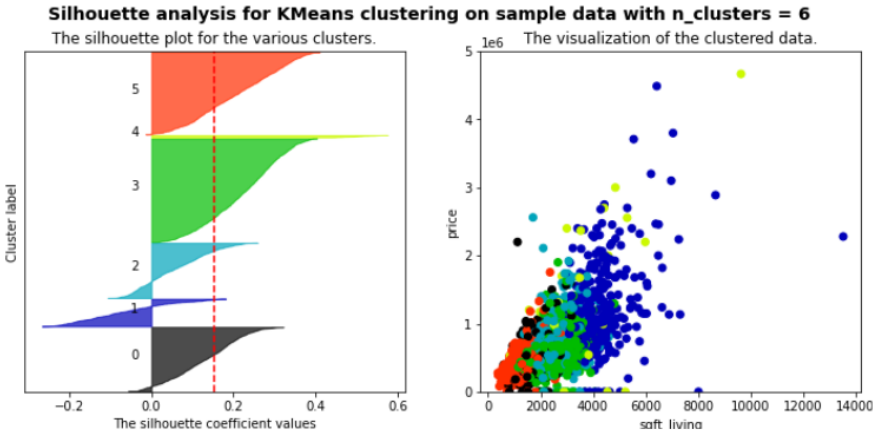
En la siguiente imagen se presenta la gráfica:



Empecemos con la gráfica para dos grupos. En ella, un grupo está coloreado de verde y el otro de gris. Se puede notar que el grupo gris es mayor que el grupo verde, eso significa que hay más datos en ese grupo. La forma que hace el color gris está creada uniendo las líneas de cada dato en ese grupo. Cada línea corresponde a cada dato, y las líneas empiezan en 0 hasta el valor que a ese dato le asigna el índice *Silhouette*. Como seguro notaste, hay líneas que empiezan en 0 y se dibujan hacia la izquierda, hacia los valores negativos; eso significa que el índice no considera que esos datos quedaron bien agrupados.



Ahora bien, comparando la gráfica para dos grupos contra la gráfica para tres grupos ¿cuál indica un mejor agrupamiento? Puedes notar que en la gráfica para tres grupos hay muchos datos con valores negativos, seguramente estos influyeron en el hecho de que el valor del índice es menor para tres grupos. Solo recuerda que a medida que aumentan los grupos el valor del índice será menor, por lo que tampoco este puede ser un criterio fijo para comparar los resultados. Una ventaja que se nota en tres grupos es que hay dos grupos muy balanceados, el azul y el gris; por lo que este puede ser un detalle que ayude en la interpretación.



Revisa entonces la gráfica para seis grupos. En ella, hay una línea amarilla delgada. Como tiene un color diferente, responde a un grupo del agrupamiento, pero con pocos datos. ¿Es esto incorrecto? No se sabe realmente, pero una característica importante es que esa línea amarilla

llega a un valor cercano a 0.6, lo que indica que, para el índice, estos datos quedaron muy bien agrupados con relación a los demás. Así que, probablemente, este sea un grupo donde esos pocos datos son muy parecidos entre sí. Si esto tiene sentido en el problema, entonces es un indicativo de buen agrupamiento, ya que se detectó ese grupo relevante. Más detalles de esta situación particular la veremos en el próximo tema.

Como ves, tampoco esta gráfica nos da una idea exacta de cuál es el número de grupos adecuados, pero lo cierto es que no hay una respuesta a esa pregunta. ¿Por qué debes entonces construir esta gráfica? Porque da información que puede ayudar a tomar una decisión de manera diferente a las demás, ya que se basa en mostrar qué tan bien agrupado quedó cada dato de la base de datos.

Diferenciado *K-Elbow* para varios algoritmos

Observa el siguiente video para comprender mejor el resultado que se puede obtener generando las gráficas cuando se agrupan con varios algoritmos.

Finalmente, veamos qué información se puede tener generando estas gráficas, en particular *K-Elbow*, cuando agrupamos con varios algoritmos. Para hacerlo, puedes replicar el código para construir *K-Elbow* usando *K-means*, ahora con los otros algoritmos *Single-Linkage* y *Spectral Clustering*.

Las tres gráficas pueden quedar como muestra la siguiente imagen:

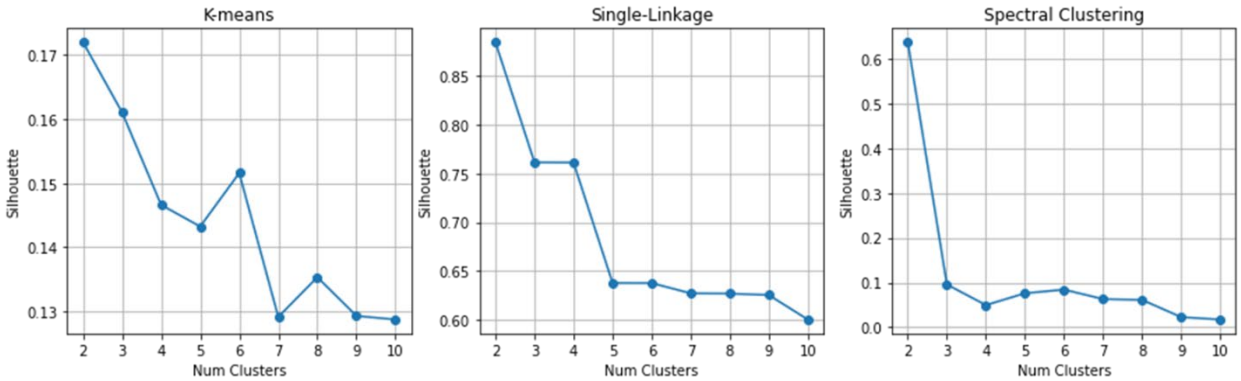


Figura 7. Gráficas K-Elbow para distintos algoritmos de agrupamiento.

Analizando la imagen te puedes dar cuente que al agrupar con *K-means* 6 y 8 grupos pueden ser buenas soluciones. Agrupando con *Single-Linkage*, quizás 4 grupos sea una buena solución. Mientras que con *Spectral Clustering*, 6 grupos tendría sentido a pesar del valor del índice es mucho menor que con 2 grupos.

Nota que cada algoritmo puede obtener grupos muy diferentes, y que es probable que no tengan nada que ver los 4 grupos que se obtienen con *Single-Linkage* comparado con los 4 grupos que obtiene *K-means* por ejemplo. En este tipo de análisis puede ayudar la gráfica *Silhouette-Plot,* para revisar como están distribuidos los datos dentro de cada grupo, o simplemente se puede revisar cómo quedaría la base de datos ya segmentada. Esencialmente veremos estos detalles también en el próximo tema.

Ideas para llevar

Validar los resultados del agrupamiento es fundamental antes de la toma de decisiones, sobre todo por el carácter subjetivo del proceso de agrupamiento.

A continuación, se enuncian algunas recomendaciones sobre los índices de validación que te pueden ser muy útiles:

- Aunque el índice de validación *Silhouette* es posiblemente uno de los más populares, no te quedes solo con este índice para validar los resultados, ten siempre a mano otras opciones.
- Revisa en la literatura situaciones parecidas a las que te corresponde resolver y presta atención si se han reportado resultados usando algún algoritmo o índice de validación. Eso puede ayudarte en la decisión de cuáles herramientas usar.
- Si comparas diversos resultados de agrupamiento, ya sea con un solo índice de validación o con varios, recuerda que este proceso

de validación también es subjetivo y no debes confiar ciegamente en las recomendaciones de los índices. La utilidad de las conclusiones a las que llegues es lo que indica que el proceso de agrupamiento ayudó o no a la toma de decisiones.

Material de apoyo

Material adicional

- ***Scikit-learn:*** https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation
- ***housedata.csv:*** https://drive.google.com/file/d/1HpeaXiwcnN2znGFLGilAdcc4OJ1vNziW/view?usp=sharing
- ***kaggle:*** https://www.kaggle.com/code/pasqua1/housedata

Bibliografía

Los contenidos de esta lección están basados en la siguiente bibliografía:

- Lee, S. H., Jeong, Y. S., Kim, J. Y., & Jeong, M. K. (2018). A new clustering validity index for arbitrary shape of clusters. *Pattern Recognition Letters*, *112*, 263-269.
- Mamat, A. R., Mohamed, F. S., Mohamed, M. A., Rawi, N. M., & Awang, M. I. (2018). Silhouette index for determining optimal k-means clustering on images in different color models. *International Journal of Engineering and Technology*, *7*(2.14), 105-109.
- Wang, J., Zhang, W., Hua, T., & Wei, T. C. (2021). Unsupervised learning of topological phase transitions using the Calinski-Harabaz index. *Physical Review Research*, *3*(1), 013074.