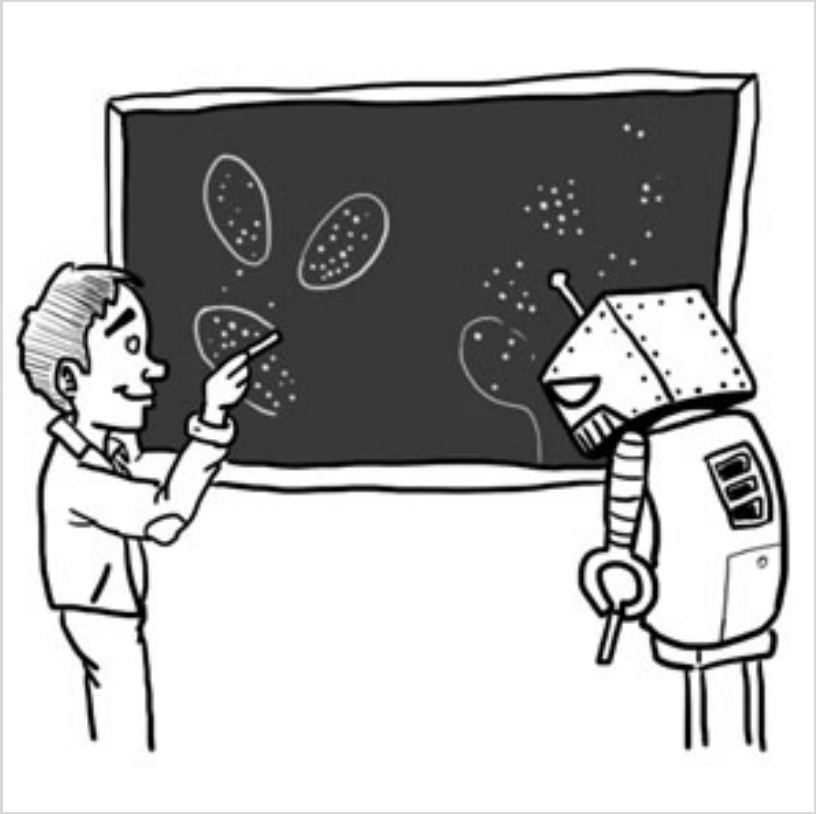


DigitalHouse > Coding School

DATA SCIENCE

Unidad: 4
Modulo: 7

Tuning de Clustering

Noviembre 2017
Buenos Aires

OBJETIVOS DE LA CLASE

1

SOBRE CLUSTERS

Entender la composición de los clusters y como fueron construidos

2

SOBRE EVALUACIÓN DEL ANÁLISIS

Evaluar el resultado del análisis realizado por K-Means

TUNING DE CLUSTERS (INTRODUCCIÓN)

Bien! Ya Corrimos un Proceso de Clusterización! ...Y ahora qué?

- Recordemos que anteriormente corrimos un K-Means y tuvimos la necesidad de decidir cómo elegir el número 'k' de clusters.
- Evaluar un modelo de clustering no es tan fácil como evaluar un modelo supervisado. ¿Por qué?
- En principio nosotros lo seleccionamos **visualmente**. Ahora vamos a aprender cómo evaluar esta elección y verificar si obtuvimos un buen resultado de nuestro análisis K-Means.
- **¿Qué constituye un buen clúster frente a un clúster malo?**
- Basándonos en medidas de exactitud y precisión del análisis, podemos explorar hasta qué punto hemos **caracterizado nuestros datos.**

TÉCNICAS PARA EVALUAR CLUSTERS

— Dos grandes tipos de formas de evaluación:

Internas: buscan evaluar qué tan parecidos entre sí son los miembros de un cluster (homogeneidad) y qué tan diferentes son con respecto al resto de los clusters. Es decir, se basan en la información intrínseca que posee el dataset del que disponemos.

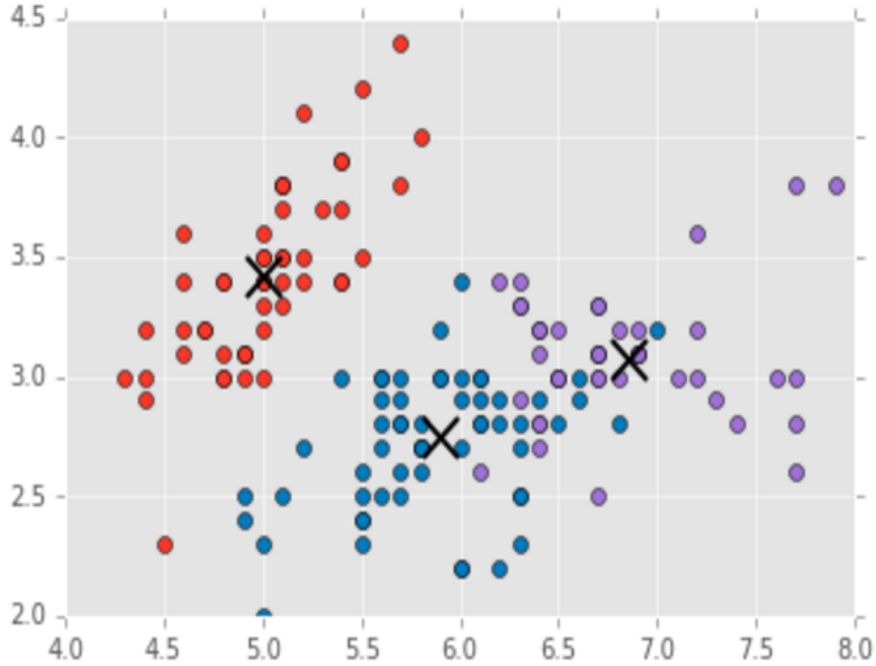
Externas: utilizan alguna variable que se asuma “correlacionada” a la clusterización para evaluar la performance de los clusters. La idea sería que el clustering separe “correctamente” estas clases. Es decir, se basan en buena medida en el conocimiento previo sobre el problema en cuestión.

VISUALIZACIÓN

- Corroborar Visualmente!

SENCILLO!!!

- Después de ejecutar el algoritmo y calcular los centroides como lo hicimos en la clase anterior, podemos trazar los clusters resultantes para ver dónde se posan los centroides y cómo se agrupan los clusters.



Medidas de Validación Internas

SILHOUTTE SCORE

- El coeficiente de silueta, es la medida de **cuán estrechamente relacionado** está un punto con miembros de su grupo en lugar de con miembros de otros grupos.

- Si s de un punto es grande, la distancia media del punto dentro del clúster es menor que la distancia promedio a los puntos en el cluster vecino => por lo que el punto está bien clasificado.

- Si es pequeña, la distancia media del punto dentro del grupo es mayor que la distancia promedio al objetos en el clúster vecino, por lo que el punto se ha clasificado erróneamente.

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

Donde:

"a" es la distancia media entre el punto y todos los demás puntos del mismo cluster.

"b" es la distancia media entre el punto y todos los puntos del cluster vecino más cercano.

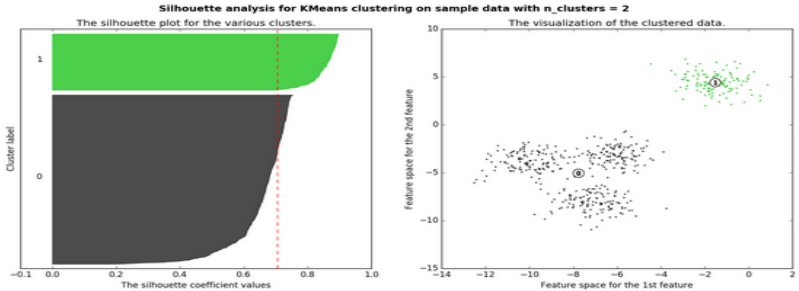
SILHOUTTE SCORE

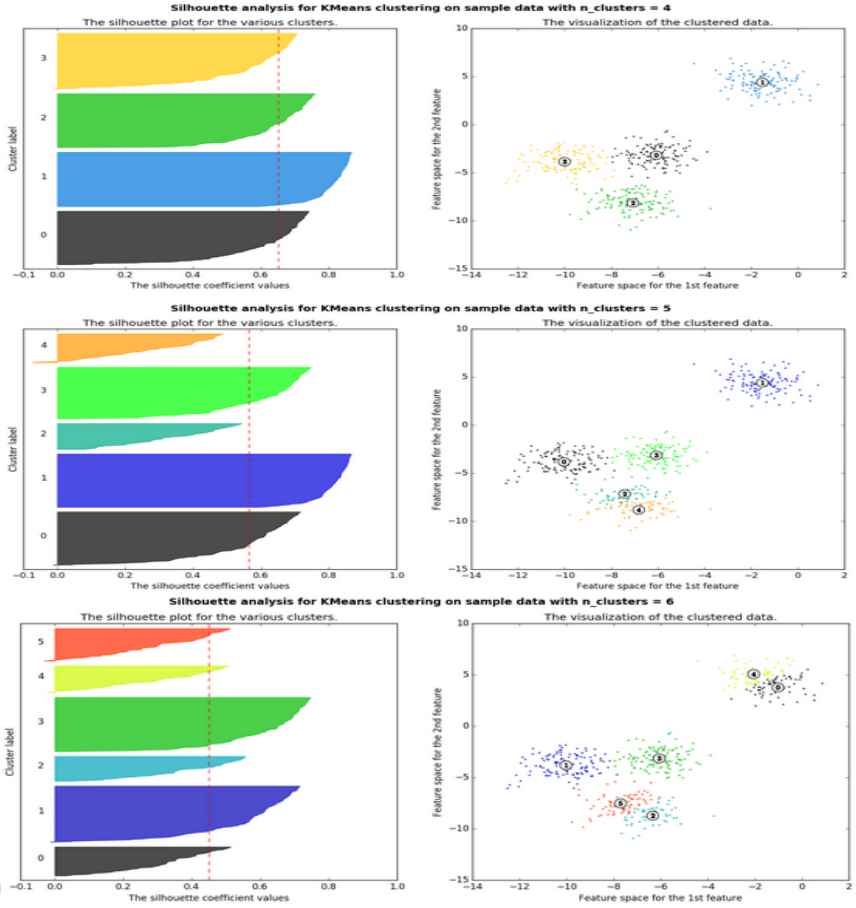
- Esta métrica permite evaluar qué tan bueno es el proceso de clusterización tanto en un solo elemento como en el agregado
- El coeficiente de silueta de todo el set es dado por la media de los coeficientes calculados para cada punto.

$$sil(C) = \overline{sil(k)} = \frac{1}{k}\sum_{i=1}^{k} sil(C_i)$$

DigitalHouse > Coding School

Ejemplo de Análisis de SC para distintos números de k-cluster





Observar y comparar **tamaño de los clusters** y valores de **sc por encima de la media.**
Para k= 3,5 y 6 es una mala elección. Entre 2 y 4 es más ambiguo. k=4 parece ser la correcta.

En este ejemplo de **python**, medimos la distancia entre puntos usando la **distancia euclidiana,** la distancia directa típica entre dos puntos.

```
metrics.silhouette_score(y, predicted, metric='euclidean')
```

(La métrica también se puede calcular utilizando Manhattan Distance, sin embargo, seguiremos usando Euclidean.)

Calinski-Harabaz Index

- Se define como la razón entre la dispersión entre clusters y la dispersión al interior de los clusters.
- Cuanto mayor sea el score, mejor es el modelo de clustering: la dispersión entre clusters es mayor que la dispersión al interior de los mismos.
  - Esto es bueno porque el cálculo tiene relación directa con el concepto de cluster

$$s(k) = \frac{\mathrm{Tr}(B_k)}{\mathrm{Tr}(W_k)} \times \frac{N-k}{k-1}$$

Donde:

"Bw" es la dispersión entre clusters

"Wk" es la dispersión al interior de los clusters.

Medidas de Validación Externas

DigitalHouse > Coding School

F-MEASURE

La medida F (o F1-score) **se define como la media armónica entre las medidas "Precision y Recall"**

Precision (o exhaustividad)**:** Fracción de clasificaciones correctas sobre la cantidad de predicciones hechas de la clase.
Recall (o sensibilidad)**:** Fracción de clasificaciones correctas sobre la cantidad verdadera de la clase

En Python podemos calcular este coeficiente con una simple línea:

```
metrics.classification_report(y_test, predicted)
```

CONFUSION MATRIX

- A pesar del nombre, una matriz de confusión es todo menos confusa!!!.
- Es un gráfico cuadrante sencillo con métricas que se ve así:

0	1	2	
0	28	22	Iris-setosa
47	3	0	Iris-versicolor
50	0	0	Iris-virginica

DigitalHouse > Coding School

CONFUSION MATRIX

¿Cómo interpretar esto?

0	1	2	
0	28	22	Iris-setosa
47	3	0	Iris-versicolor
50	0	0	Iris-virginica

Por definición, una **matriz de confusión C** es tal que elemento C_{ij} es igual al número de observaciones que se sabe que pertenecen a la clase *i*, pero que se predijo de la clase *j*.

Supongamos las clases:

0=setosa
1=versicolor
2=virginia

C_{01} es la cantidad de observaciones que pertenecen a la clase "setosa" y se predijeron de la clase "versicolor"

CONFUSION MATRIX

¿Dónde entra el nombre "confusión"?

0	1	2	
0	28	22	Iris-setosa
47	3	0	Iris-versicolor
50	0	0	Iris-virginica

Una clase **no confundida** tendrá grandes valores en la diagonal, de modo que las clases predichas coincidan con las clases reales, mientras que una clase **"confusa"** tendrá valores en todo lugar.

Si miramos en nuestro gráfico de comparación, podemos ver esto en acción!!!

Las clases predichas no coinciden bien con las clases reales, por lo que tenemos clases "confusas"!

CONFUSION MATRIX

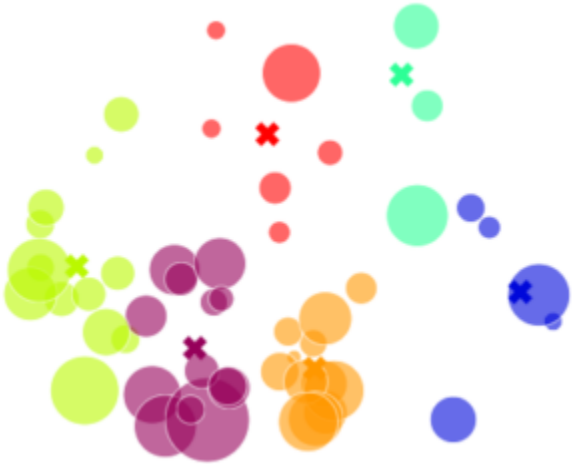
En Python podemos calcular este coeficiente con una simple línea:

```
metrics.confusion_matrix(y_test, predicted)
```

DigitalHouse > Coding School

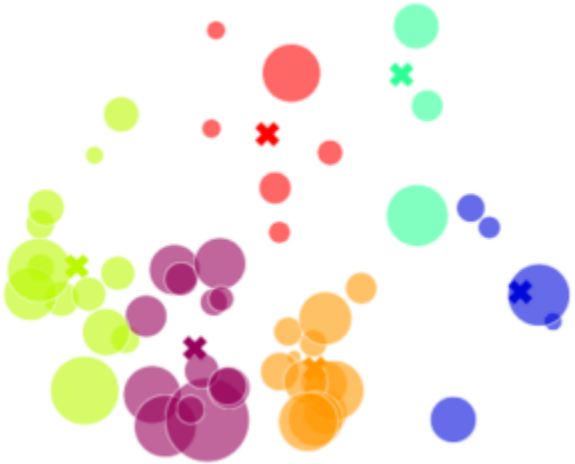
Práctica Guiada

(Realizar un análisis K-Means
y evaluar los clusters)

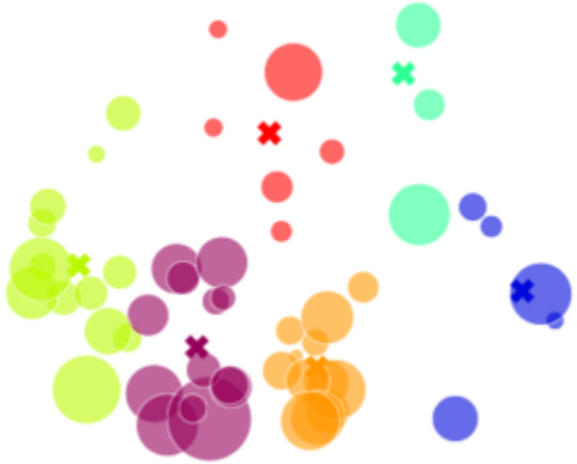


LAB

(Realizar un análisis K-Means y evaluar los clusters)



CONCLUSIONES



- Existen diferentes métodos para evaluar la calidad de nuestro análisis (incluyendo visualización, silhouette scores, F-metrics, y matrices de confusión.
- Después de analizar los Cluster, es posible que haya que redefinir la cantidad "k" de clusters buscados.
- Siempre es conveniente examinar múltiples métricas para entender la calidad del análisis realizado.