Técnicas de aprendizaje no supervisado (2da parte)

UNIVERSIDAD CATÓLICA DE CÓRDOBA
JESUITAS

Dr. Francisco Arduh
2023

DBSCAN

Define los clusters como regiones continuas de alta densidad

1. Para cada instancia el algoritmo cuenta cuantas instancias están ubicadas en la vecindad ε.
2. Si una instancia tiene al menos un número `min_sample` en su vecindad ε, se lo considera una instancia *core*.
3. Todas las instancias en la vecindad de una *core* pertenecen al mismo cluster.
4. Toda instancia no core y que no esté en la vecindad de una core es considerada una anomalía.

Ver: https://miro.medium.com/v2/resize:fit:1280/1*kUBIldisxX6hGFEJpCisMQ.gif

Clustering Jerárquico

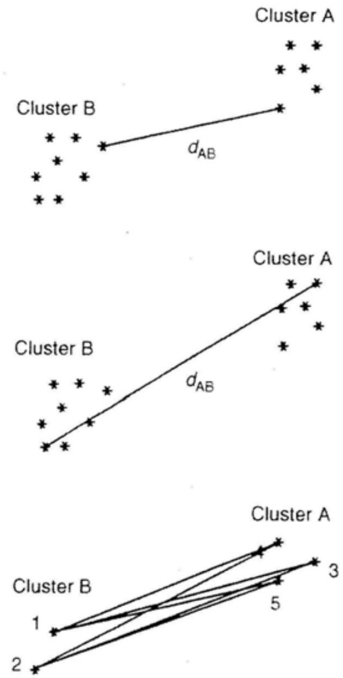
Se construye de abajo a arriba de la siguiente forma:

1. Se toma cada instancia como un clúster.
2. Se toman las dos instancias más cercanas y se genera un nuevo cluster.
3. Se toman los clústers <u>más cercanos</u> y se los agrupa en un clúster
4. Se repite el paso anterior hasta que quede un clúster.

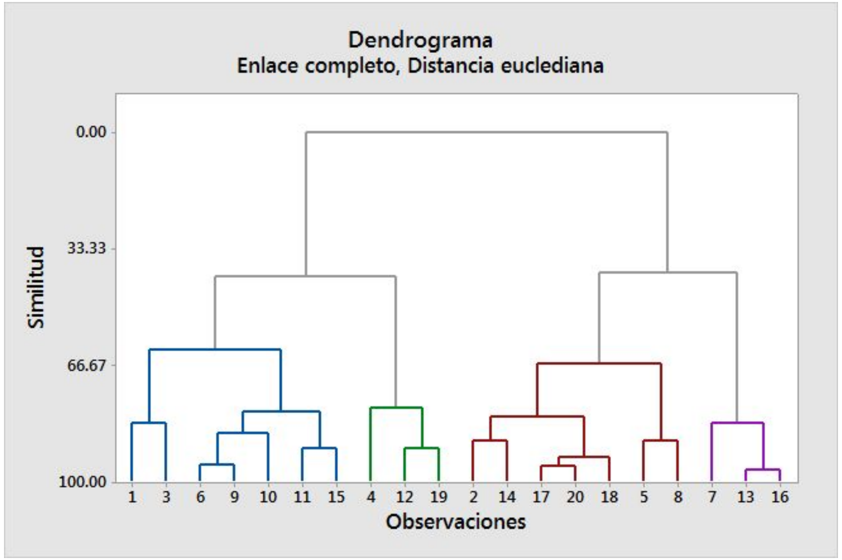
Ejemplo: https://miro.medium.com/v2/resize:fit:257/0*iozEcRXXWXbDMrdG.gif

Clustering Jerárquico: Distancia entre clústers

1. Entre puntos más cercanos.
2. Entre puntos más lejanos.
3. Distancia promedio.
4. Distancia entre los centroides.



Clustering Jerárquico: Dendrograma



¿Cómo se construye?

https://miro.medium.com/v2/resize:fit:700/1*ET8kCcPpr893vNZFs8j4xg.gif

Más algoritmos de clustering

- BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies)
- Mean-Shift
- Affinity propagation
- Spectral clustering

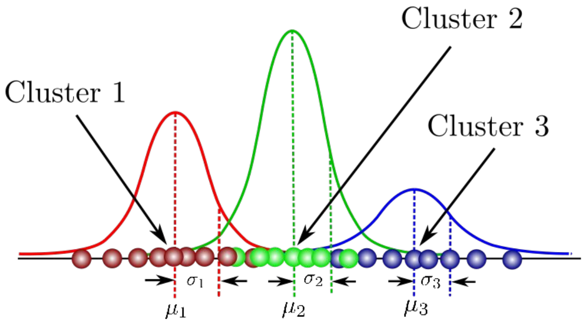
Guía scikit-learn
https://scikit-learn.org/stable/modules/clustering.html

Gaussian Mixtures

Puede ser utilizado como estimador de densidad, clustering o detección de anomalías.

Este modelo asume que las instancias son generadas por una mezcla de *k* (hiperparámetro) distribuciones gaussianas con pesos y parámetros desconocidos.

Los pesos, la media $\boldsymbol{\mu}$ y matriz de covarianza $\boldsymbol{\Sigma}$ son parámetros del modelo a determinar.



Bayesian Gaussian Mixture Model

Una variante de GMM que encuentra el número de cluster óptimo de forma automática llevando a cero los pesos de los clúster innecesarios.

```
>>> from sklearn.mixture import BayesianGaussianMixture
>>> bgm = BayesianGaussianMixture(n_components=10, n_init=10)
>>> bgm.fit(X)
>>> np.round(bgm.weights_, 2)
array([0.4 , 0.21, 0.4 , 0.  , 0.  , 0.  , 0.  , 0.  , 0.  , 0.  ])
```

Algoritmos para detección de anomalía o novedades

- PCA (y otras técnicas de reducción de dimensionalidad con el método `inverse_transform`)
- Fast-MCD: Implementado como la clase `EllipticEnvelope`.
- Isolation Forest.
- Local Outlier Factor (LOF)
- One-class SVM

Reglas de asociación

User ID	Movies liked
46578	Movie1, Movie2, Movie3, Movie4
98989	Movie1, Movie2
71527	Movie1, Movie2, Movie4
78981	Movie1, Movie2
89192	Movie2, Movie4
61557	Movie1, Movie3

Potential Rules:

Movie1 ⇒ Movie2

Movie2 ⇒ Movie4

Movie1 ⇒ Movie3

Reglas de asociación

Transaction ID	Products purchased
46578	Burgers, French Fries, Vegetables
98989	Burgers, French Fries, Ketchup
71527	Vegetables, Fruits
78981	Pasta, Fruits, Butter, Vegetables
89192	Burgers, Pasta, French Fries
61557	Fruits, Orange Juice, Vegetables
87923	Burgers, French Fries, Ketchup, Mayo

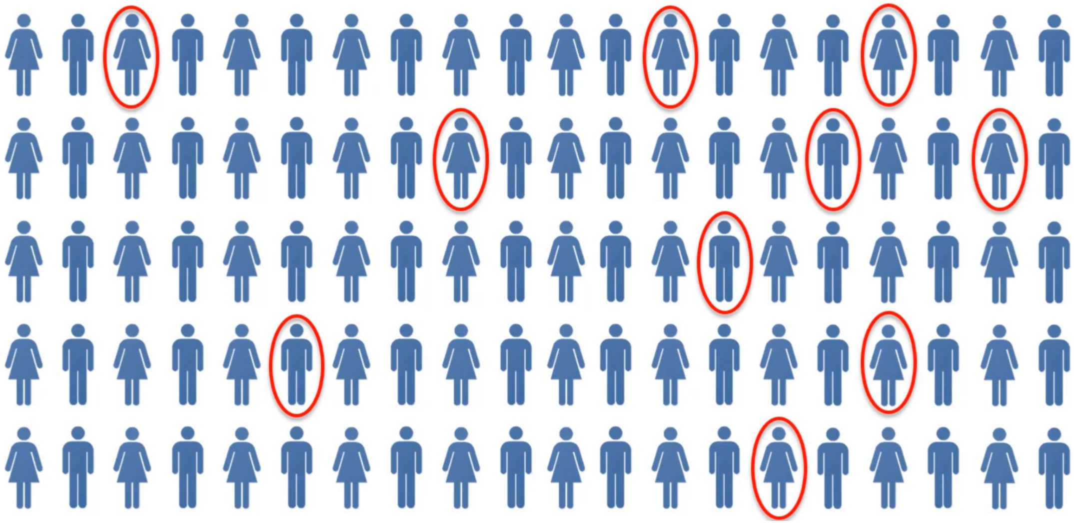
Potential Rules:

- Burgers ⇒ French Fries
- Vegetables ⇒ Fruits
- Burgers, French Fries ⇒ Ketchup

A priori: support

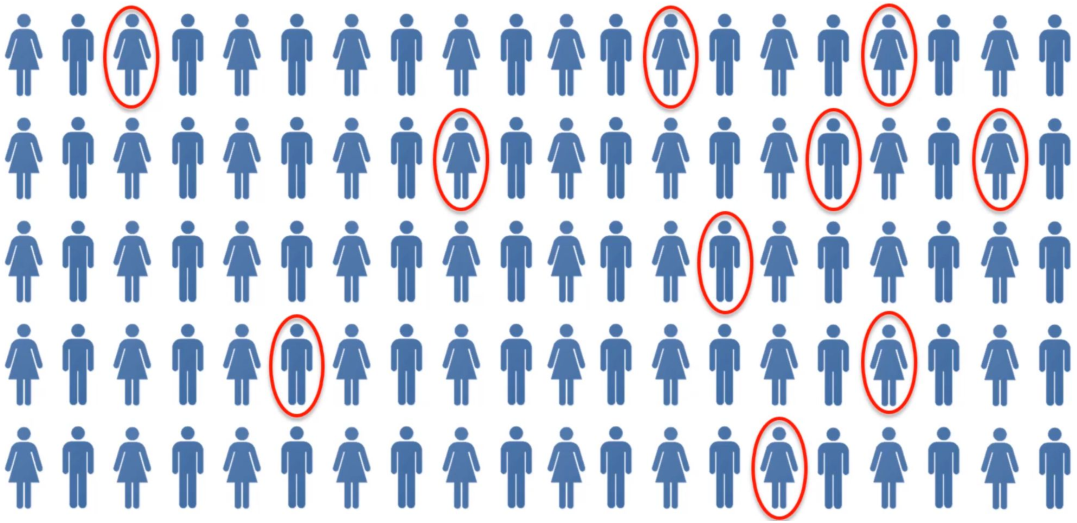
Movie Recommendation:

$$\text{support}(\boldsymbol{M}) = \frac{\#\ \text{user watchlists containing}\ \boldsymbol{M}}{\#\ \text{user watchlists}}$$



A priori: support

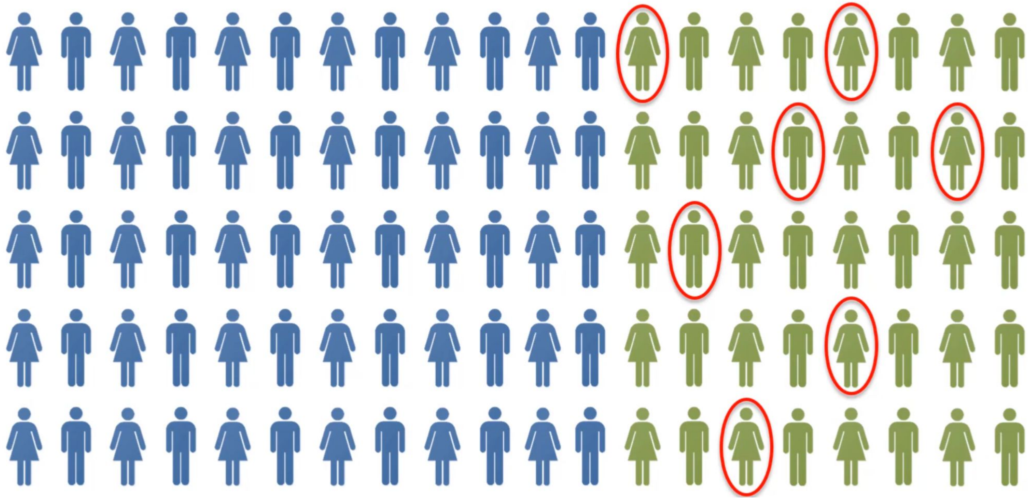
Movie Recommendation: $$\text{support}(\boldsymbol{M}) = \frac{\#\ \text{user watchlists containing } \boldsymbol{M}}{\#\ \text{user watchlists}}$$



Support = 10%

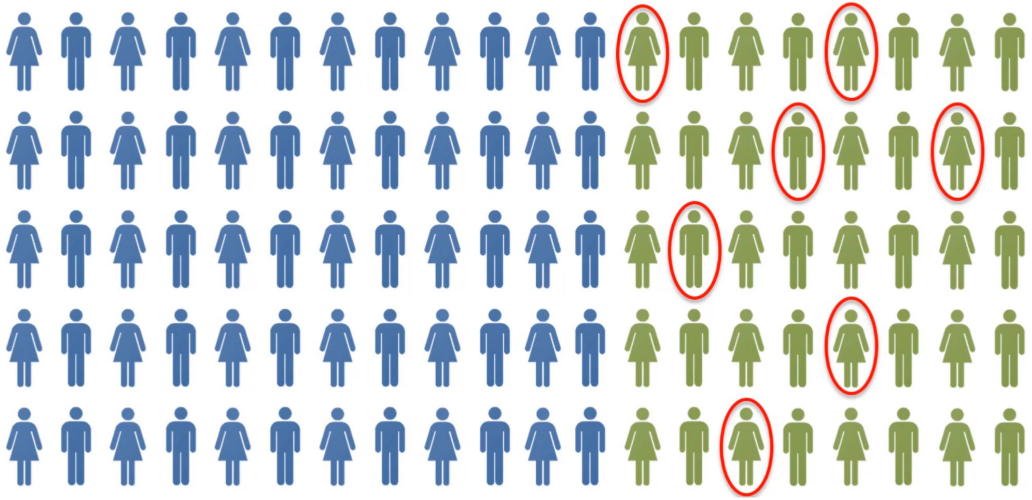
A priori: confidence

Movie Recommendation: $\text{confidence}(\boldsymbol{M_1} \to \boldsymbol{M_2}) = \dfrac{\#\ \text{user watchlists containing } \boldsymbol{M_1} \text{ and } \boldsymbol{M_2}}{\#\ \text{user watchlists containing } \boldsymbol{M_1}}$



A priori: confidence

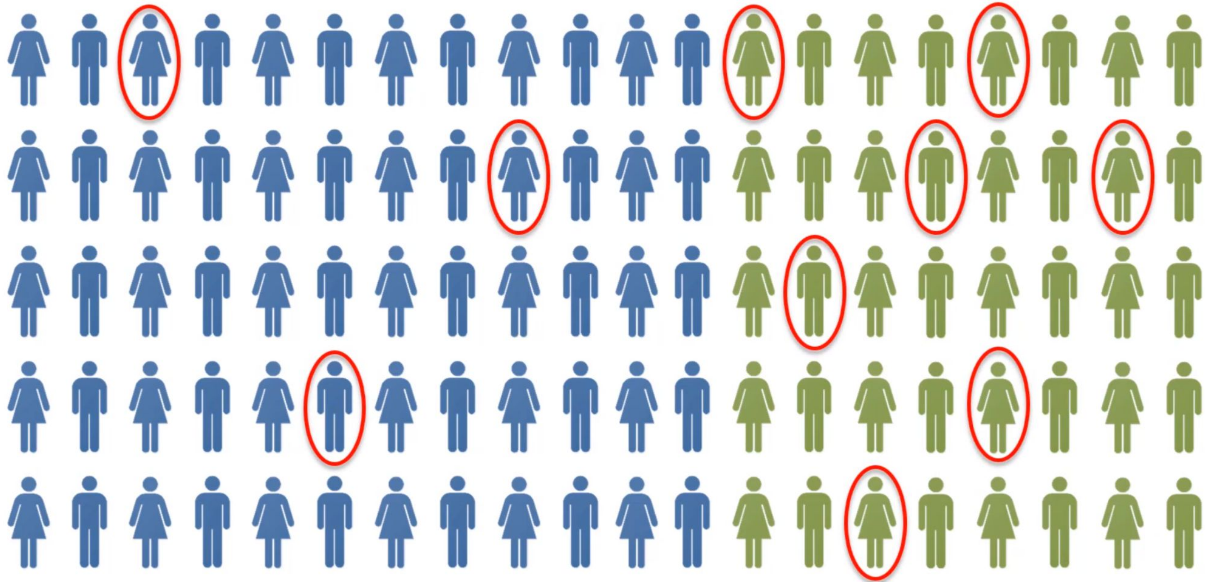
Movie Recommendation: $\text{confidence}(\boldsymbol{M_1} \rightarrow \boldsymbol{M_2}) = \dfrac{\#\text{ user watchlists containing } \boldsymbol{M_1} \text{ and } \boldsymbol{M_2}}{\#\text{ user watchlists containing } \boldsymbol{M_1}}$



confidence = 7/40 = 17,5%

A priori: lift

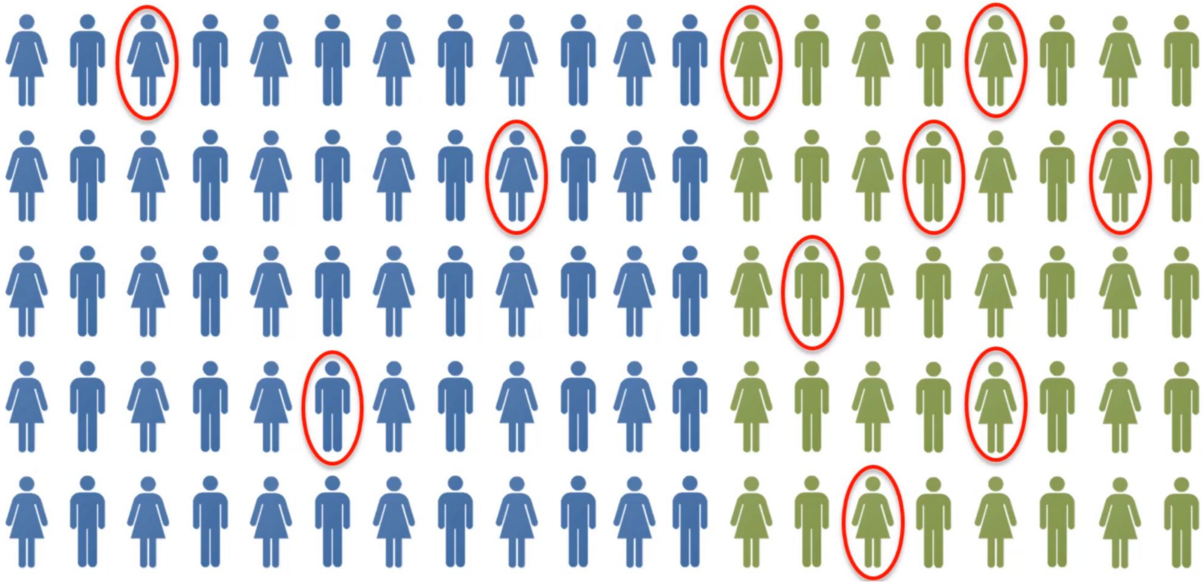
Movie Recommendation: $$\text{lift}(M_1 \rightarrow M_2) = \frac{\text{confidence}(M_1 \rightarrow M_2)}{\text{support}(M_2)}$$



A priori: lift

Movie Recommendation:

$$\text{lift}(M_1 \rightarrow M_2) = \frac{\text{confidence}(M_1 \rightarrow M_2)}{\text{support}(M_2)}$$



lift = 17,5%/10% = 1,75

A priori: pasos

1. Elegir un número mínimo de support y confidence.
2. Tomar con un subset de datos con un support superior al elegido
3. De subset anterior quedarse con un subset de datos con un confidence superior al elegido.
4. Ordenar por lift

¿Dudas?