**Agrupamiento, clusterial**

- Proceso de agrupar objetos similares en clusters

- Se clasifica como aprendizaje no supervisado xq no usan etiquetas

- Para entender la estructura y similitud entre datos

- Utilizado en motores de recomendacion, analisis de redes sociales, riesgo credito, clasificacion de geners, riesgos medicos

**K-Means Clustering:**

-Definir k centroides (uno para cada grupo)

-Tomar cada punto de la base de datos

-Situarlo en la clase de su centroide más cercano.

EL punto se va a la zona densa (scatter) mas cercana?

**Mean-Shift Clustering:**

A sliding-window-based algorithm

that attempts to find dense areas of data points.

Recorre (scatter) y busca zona mas densa de puntos?

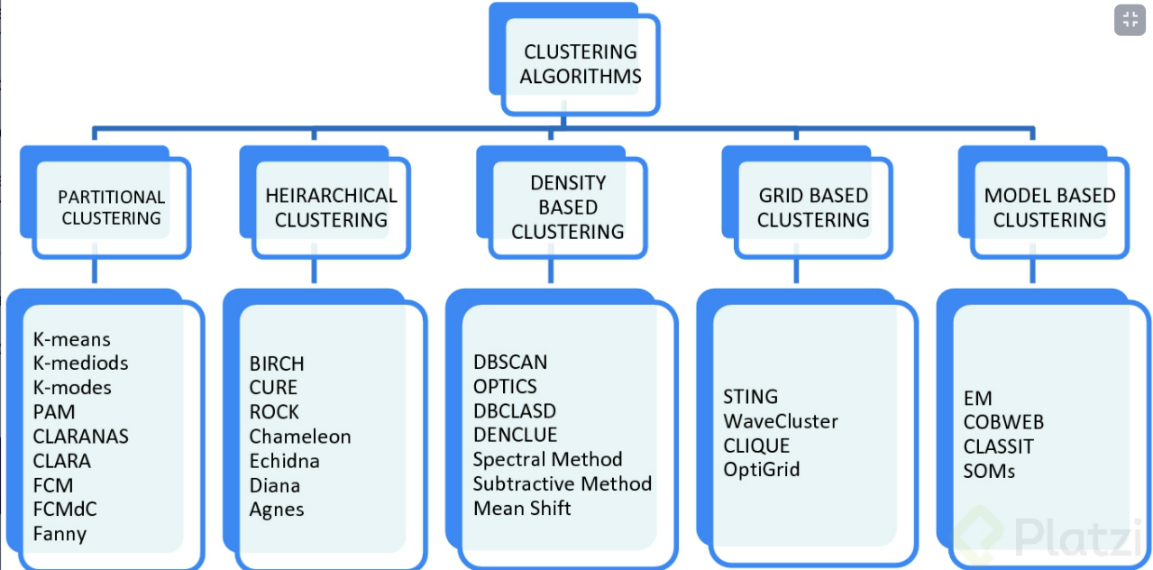
**DBSCAN:**

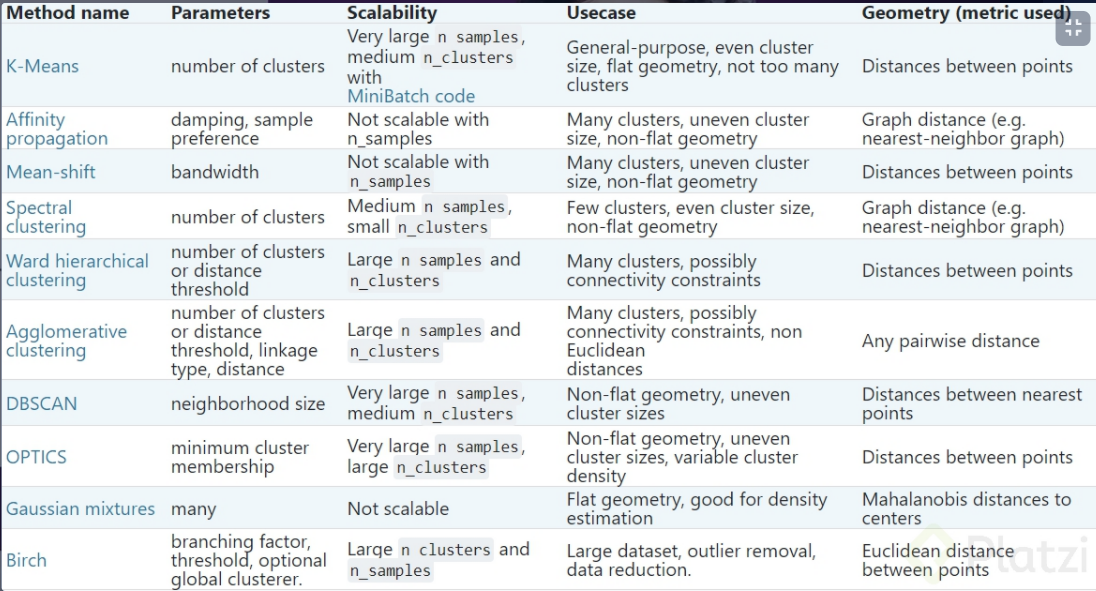
A density-based clustered algorithm similar to mean-shift,

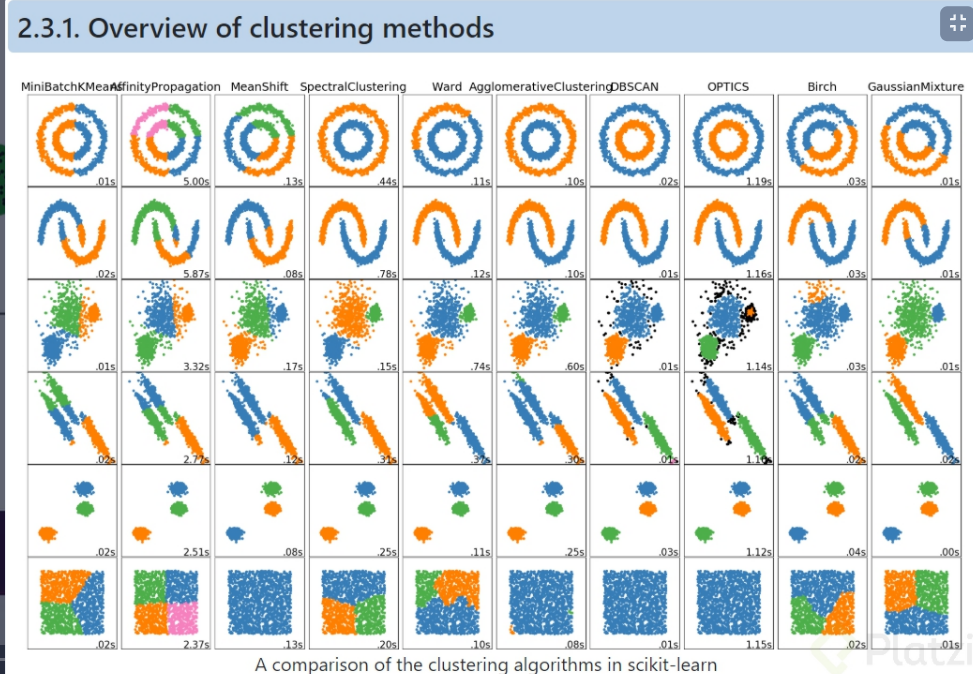
but with a couple of notable advantages.

Expectation–Maximization (EM) Clustering using Gaussian Mixture Models (GMM):

Gaussian Mixture Models (GMMs) give us more flexibility than K-Means.







***ALGORITMOS DE AGRUPAMIENTO***  
Clustering es una técnica de Machine Learning que implica la agrupación de puntos de datos. Dado un conjunto de puntos de datos, podemos utilizar un algoritmo de agrupación para clasificar cada punto de datos en un clúster específico. En teoría, los puntos de datos que están en el mismo clúster deben tener propiedades y/o características similares, mientras que los puntos de datos en diferentes clústeres deben tener propiedades y/o características muy diferentes. La agrupación es un método de Aprendizaje no Supervisado y es una técnica común para el análisis de datos estadísticos utilizada en muchos campos.

***Agrupamiento K Means***  
K Means es probablemente el algoritmo de clustering más conocido, es fácil de entender e implementar en código.

1. Para empezar, seleccionamos un número de clústeres para usar e inicializamos aleatoriamente sus respectivos puntos centrales. Para calcular el número de clústeres a utilizar, es bueno echar un vistazo rápido a los datos y tratar de identificar cualquier agrupación distinta. Los puntos centrales son vectores de la misma longitud que cada vector de puntos de datos.
2. Cada punto de datos se clasifica calculando la distancia entre ese punto y cada centro del clúster, y luego clasificando el punto que estará en el clúster cuyo centro está más cerca de él.
3. Basándonos en estos puntos clasificados, recalculamos el centro del clúster tomando la media de todos los vectores del clúster.
4. Repite estos pasos para un número determinado de iteraciones o hasta que los centros de clústeres no cambien mucho entre iteraciones. También puedes optar por inicializar aleatoriamente los centros de grupo unas cuantas veces, y luego seleccionar la ejecución que parezca que proporcionó los mejores resultad

K Means tiene la ventaja de que es bastante rápido, ya que todo lo que estamos haciendo es calcular las distancias entre puntos y centros de grupo, por lo tanto, son muy pocos cálculos.

Por otro lado, K Means tiene un par de desventajas. En primer lugar, tienes que seleccionar cuantos clústeres hay. Esto no siempre es trivial e idealmente con un algoritmo de clustering nos gustaría que lo resolviera por nosotros porque el objetivo es obtener alguna información de los datos. K Means también comienza con una selección aleatoria de centros de clústeres y, por lo tanto, puede producir diferentes resultados de clústeres en diferentes ejecuciones del algoritmo. Por lo tanto, los resultados puedes no ser repetibles y carecer de consistencia. Otros métodos de clustering son más consistentes.

K Medians es otro algoritmo de clustering relacionado con K Means, excepto que en lugar de volver a calcular los puntos centrales del clúster usando la media, usamos el vector mediano del grupo. Este método es menos sensible a los valores atípicos, debido al uso de la mediana, pero es mucho más lento en el caso de conjuntos de datos más grandes, ya que se requiere la clasificación en cada iteración cuando se calcula el vector de la mediana.

***Agrupamiento Mean Shift***  
La agrupación Mean Shift es un algoritmo basado en ventanas deslizantes que intenta encontrar áreas densas de puntos de datos. Es un algoritmo basado en el centroide, lo que significa que el objetivo es localizar los puntos centrales de cada clúster, lo que funciona actualizando a los candidatos para que los puntos centrales sean la media de los puntos dentro de la ventana deslizante.

Estas ventanas candidatas son filtradas en una etapa de post procesamiento para eliminar los duplicados cercanos, formando el conjunto final de puntos centrales y sus correspondientes grupos.

1. Para explicar el agrupamiento Mean Shift consideremos un conjunto de datos en un espacio bidimensional. Comenzamos con una ventana circular deslizante centrada en el punto C, seleccionada aleatoriamente, y con el radio r como núcleo. Mean Shift es un algoritmo de escalada de colinas que implica el desplazamiento iterativo de este núcleo a una región de mayor densidad en cada paso hasta la convergencia.
2. En cada iteración, la ventana deslizante se desplaza hacia regiones de mayor densidad desplazando el punto central a la media de los puntos dentro de la ventana. La densidad dentro de la ventana corredera es proporcional al número de puntos dentro de ella. Naturalmente, al cambiar a la media de los puntos en la ventana, gradualmente se moverá hacia áreas de mayor densidad de puntos.
3. Seguimos desplazando la ventana corrediza de acuerdo con la media hasta que no hay dirección en la que un desplazamiento pueda acomodar más puntos dentro del núcleo.  
   Este proceso de los pasos 1 al 3 se realiza con muchas ventanas hasta que todos los puntos se encuentran dentro de una ventana. Cuando se superponen varias ventanas, se conserva la ventana que contiene la mayor cantidad de puntos. A continuación, los puntos de datos se agrupan según la ventana deslizante en la que residen.

A diferencia de la agrupación K Means, no es necesario seleccionar el número de clústeres, ya que el desplazamiento medio lo descubre automáticamente. Es una gran ventaja. El hecho de que los clustering converjan hacia los puntos de máxima densidad también es muy deseable, ya que es bastante intuitivo de entender y encaja bien en un sentido naturalmente basado en datos. El inconveniente es que la selección del tamaño/radio “r” de la ventana puede ser no trivial.

***DBSCAN***  
DBSCAN es un algoritmo de agrupamiento basado en la densidad similar a Mean Shift, pero con un par de ventajas notables. Las siglas DBSCAN significan agrupamiento espacial basado en densidad de aplicaciones con ruido.

1. DBSCAN comienza con un punto de datos de inicio arbitrario que no ha sido visitado. El vecindario de este punto se extrae usando una distancia épsilon, todos los puntos que están dentro de la distancia de épsilon son puntos de vecindario.
2. Si hay un número suficiente de puntos dentro de este vecindario, entonces el proceso de agrupación comienza y el punto de datos actual se convierte en el primer punto del nuevo grupo. De lo contrario, el punto será etiquetado como ruido, más tarde este punto ruido se podría convertirse en la parte del clúster. En ambos casos, este punto se marca como visitado.
3. Para este primer punto en el nuevo clúster, los puntos dentro de su vecindario distante épsilon también pasan a formar parte del mismo clúster. Este procedimiento de hacer que todos los puntos en el vecindario de épsilon pertenezcan al mismo grupo se repite para todos los nuevos puntos que acaban de ser agregados al clúster del grupo.
4. Este proceso de pasos 2 y 3 se repite hasta que todos los puntos en el clúster son determinados, es decir, todos los puntos dentro del vecindario épsilon del clúster han sido visitados y etiquetados.
5. Una vez que terminamos con el clúster actual, se recupera y procesa un nuevo punto no visitado, lo que conduce al descubrimiento de otro clúster o ruido. Este proceso se repite hasta que todos los puntos se marcan como visitados. Dado que al final de esta visita se han visitado todos los puntos, cada uno de ellos se marcará como perteneciente a un clúster o como ruido.

DBSCAN presenta algunas grandes ventajas sobre otros algoritmos de agrupación en clúster. En primer lugar, no requiere un número determinado de clúster. También identifica los valores atípicos como ruidos, a diferencia del desplazamiento medio, que simplemente los coloca en un grupo, incluso si el punto de datos es muy diferente. Además, puede encontrar clústeres de tamaño y forma arbitraria bastante bien.

El principal inconveniente de DBSCAN es que no funciona tan bien como otros cuando los clústeres son de densidad variable. Esto se debe a que la configuración del umbral de distancia épsilon para identificar los puntos de vecindario variará de un clúster a otro cuando la densidad varíe. Esta desventaja también ocurre con los datos de muy alta dimensión, ya que de nuevo el umbral de distancia épsilon se vuelve difícil de estimar.

***Agrupamiento utilizando modelos de mezcla gaussiana***  
Los modelos de mezcla gaussiana (GMM, por sus siglas en ingles) nos dan más flexibilidad que los K Means. Con los GMM suponemos que los puntos de datos están distribuidos por Gauss, esto es una suposición menos restrictiva que decir que son circulares usando la media. De esta manera, tenemos dos parámetros para describir la forma de los clústeres, la media y la desviación estándar. Tomando un ejemplo en dos dimensiones, esto significa que los clústeres pueden tomar cualquier tipo de forma elíptica, ya que tenemos una desviación estándar tanto en la dirección X como en la Y. Por lo tanto, cada distribución gaussiana se asigna a un solo grupo.

Para encontrar los parámetros del gaussiano para cada clúster, por ejemplo, la media y la desviación estándar, utilizaremos un algoritmo llamado maximización de expectativas (EM, por sus siglas en inglés). Entonces podemos proceder con el proceso de agrupación de maximización de expectativas usando GMM.

1. Comenzamos seleccionando el número de clúster, como se hace en K Means e inicializando aleatoriamente los parámetros de distribución gaussiana para cada clúster. Uno puede tratar de proporcionar una buena estimación de los parámetros iniciales echando un vistazo rápido a los datos también. Aunque hay que tener en cuenta, esto no es 100% necesario, ya que los gaussianos empiezan como muy pobres, pero se optimizan rápidamente.
2. Dadas estas distribuciones gaussianas para cada clúster, calcular la probabilidad de que cada punto de datos pertenezca a un clúster en particular. Cuanto más cerca esté un del centro de los gaussianos, más probable será que pertenezca a ese grupo. Esto debería tener sentido intuitivo ya que con una distribución Gaussiana estamos asumiendo que la mayoría de los datos se encuentran más cerca del centro del clúster.
3. Basándonos en estas probabilidades, calculamos un nuevo conjunto de parámetros para las distribuciones gaussianas de manera que maximicemos las probabilidades de los puntos de datos dentro de los clústeres. Calculamos estos nuevos parámetros utilizando una suma ponderada de las posiciones de los puntos de datos, donde las ponderaciones son las probabilidades del punto de datos que pertenece a ese clúster en particular.
4. Los pasos 2 y 3 se repiten iterativamente hasta la convergencia, donde las distribuciones no cambian mucho de iteración en iteración.

El uso de los modelos de mezcla gaussiana presenta dos ventajas clave. En primer lugar, son mucho más flexibles en términos de covarianza de clústeres que los K Means, debido al parámetro de desviación estándar, los clústeres pueden adoptar cualquier forma de elipse, en lugar de limitarse a círculos. K Means es en realidad un caso especial de los modelos de mezcla gaussiana en el que la covarianza de cada clúster a lo largo de todas las dimensiones se acerca a 0. En segundo lugar, dado que los modelos de mezcla gaussiana utilizan probabilidades, pueden tener conglomerados múltiples por punto de datos. Por lo tanto, si un punto de datos está en medio de dos grupos superpuestos, podemos simplemente definir su clase diciendo que pertenece al X por ciento de la 1 y al Y por ciento de la 2. Es decir, los modelos de mezcla gaussiana apoyan una composición mixta.

***Agrupamiento Jerárquico***  
Los algoritmos de agrupación jerárquica se dividen en dos categorías: de arriba hacia abajo o de abajo hacia arriba. Los algoritmos ascendente tratan cada punto de datos como un solo clúster al principio y luego fusionan sucesivamente o aglomeran pares de clústeres hasta que todos los clústeres se hayan fusionado a un único clúster que contenga todos los puntos de datos. Por lo tanto, la agrupación jerárquica ascendente se denomina agrupación algomerativa jerárquica. Esta jerarquía de clúster se representa como un árbol o dendograma. La raíz del árbol es el único clúster que recoge todas las muestras, siendo las hojas los clústeres con una sola muestra.

1. Comenzamos tratando cada punto de datos como un solo clúster, es decir, si hay X puntos de datos en nuestro conjunto de datos, entonces tenemos X clústeres. Luego seleccionamos una métrica de distancia que mide la distancia entre dos clústeres.
2. En cada iteración, combinamos dos grupos en uno. Los dos grupos que se combinarán se seleccionan como los que tiene la vinculación media más pequeña. Es decir, de acuerdo con nuestra métrica de distancia seleccionada, estos dos clústeres tienen la menor distancia entre sí y por lo tanto son más más similares y deben ser combinados.
3. El paso 2 se repite hasta llegar a la raíz del árbol, es decir, solo tenemos un clúster que contiene todos los puntos de datos. De esta manera podemos seleccionar cuántos clústeres queremos al final, simplemente eligiendo cuándo dejar de combinar los clústeres, es decir, cuando dejar de construir el árbol.

La agrupación jerárquica no requiere que especifiquemos el número de clústeres e incluso podemos seleccionar qué número de clústeres se ve mejor ya que estamos construyendo un árbol. Además, el algoritmo no es sensible a la elección de la métrica de distancia, todos ellos tienden a funcionar igualmente bien, mientras que, con otros algoritmos de agrupamiento, la elección de la métrica de distancia es crítica. Un caso particularmente útil de los métodos de agrupamiento de jerárquica es cuando los datos subyacentes tienen una estructura jerárquica y se desea recuperar la jerarquía, otros algoritmos de agrupación no pueden hacer esto. Estas ventajas de la agrupación jerárquica vienen a costa de una menor eficiencia, a diferencia de la complejidad lineal de K Means y el modelo de mezcla gaussiana.