Aprendizaje Automático

Eric Brandon García Luján

1. Introducción

Uno de los objetivos en este reporte es identificar algún algoritmo no supervisado de los que no se vieron en clase, y que además, pueda ser usado en nuestro dataset elegido (pybaseball). Se explicara el de qué trata el modelo matemático y por qué convendría usarse en nuestro dataset.

Para lo anterior, como punto prinicipal se explica que es el aprendizaje no supervisado.

El *aprendizaje no supervisado* sirve para analizar y agrupar conjuntos de datos no etiquetados(información en bruto, no ha tenido ningún tratamiento ni se ha asignado clasificación). Son algoritmos que descubren patrones ocultos o agrupaciones de datos sin necesidad de intervención humada y se vuelven una herramienta poderosa par el análisis exploratorio de datos por su capacidad de descubrir similitudes y diferencias y puede ser usada en industrias como la comercial al analizar estrategias de venta cruzada, en la segmentación de clientes o para el reconocimiento de imágenes.

2. Enfoques comunes

Estos modelos se usan normalmente para 3 tareas principalmente; agrupamiento, asociación y reducción de dimensionalidad (variables).

2.1. Agrupación en clústeres

Es una técnica en donde los datos se agrupan sin clasificarse aún, uno de los más conocidos es la agrupación en clústeres de k-Means. K-Means es un método de agrupación exclusiva en donde los datos se asignan en K grupos, donde K representa el número de agrupaciones según la. Distancia desde el centroide

2.2. Agrupación jerárquica

También conocida como análisis de agrupamiento jerárquico (HCA), puede categorizarse de dos formas: aglomerados o divisivos. La agrupación aglomerativa se considera como un "enfoque de abajo hacia arriba". Sus puntos de datos se aíslan inicialmente como agrupaciones separadas y luego se fusionan de foma iterativa según la similitud hasta que se logra crear un grupo. Norlmalmente para medir la similitud se usan cuatro métodos: método de ward, enlace promedio, enlace completo(vecino más alejado) y enlace simple(vecino más próximo). La agrupación divisiva se definiría como lo opuesto a ala aglomerativa ("de arriba hacia abajo) aunque esta normalmente no es usada.

2.3. Agrupación probabilística

Es una técnica que ayuda a resolver problemas de estimación de densidad o agrupamiento "suave". Los puntos de datos se agrupan en función de la probabilidad de que pertenezcan a una distribución particular, uno de los modelos más usados es el modelo de mezcla gaussiana (GMM).

2.4. Reglas de asociación

Es un método basado en reglas para encontrar relaciones entre variables en un conjunto de datos determinado. Este modelo puede ser usado en una rama comercial ya que le podría interesar el hábito

de consumo de las personas, el cual permite a las empresas comprender mejor las relaciones entre los diferentes productos y así desarrollar estrategias como la venta cruzada. También puede ser utilizado en industrias como la música, como en Spotify al analizar las canciones que hay en común.

2.5. Reducción de dimensionalidad

Es una técnica que se utiliza cuando el número de características o dimensiones de un conjunto de datos determinado es demasiado alto reduciendo la cantidad de entradas de datos a un tamaño manejable y preservando la integridad de los datos. Algunos de los métodos usados aquí son análisis de componentes principales, descomposición en valores singulares y codificadores automáticos.

3. DBSCAN

Para nuestro dataset, DBSCAN (Agrupación Espacial Basada en la Densidad de Aplicaciones) es un algoritmo de agrupación que podría ser útil ya que DBSCAN consiste en agrupar puntos de datos similares que estan muy juntos y tiene como objetivo principal simiplificar grandes conjuntos de datos en subgrupos significativos, identificar agrupaciones naturales en los datos y revelar pautas y estructuras ocultas. A diferencia de otros algoritmos como K-Means, DBSCAN no requiere que se especifique el número de conglomerados (K) Las ventajas de usar DBSCAN son: ...

- Flexibilidad en la forma del racimo
- Sin número predefinido de agrupaciones
- Manejo del ruido
- Visión basada en la densidad

Este algoritmo usa dos parámetros principales:

1. ϵ (épsilon): Es la distancia máxima entre dos puntos para que se consideren vecinos.
2. MinPts: Es el número mínimo de puntos necesarios para formar una región densa.

Al ajustar estos parámetros, se puede controlar el modo en que el algoritmo define los conglomerados. Los pasos que se siguen al ejecutar este algoritmo son los siguientes:

1. Selección de parámetros
   - Elige ϵ (épsilon): La distancia máxima entre dos puntos para que se consideren vecinos.
   - Elige MinPts: El número mínimo de puntos necesarios para formar una región densa.
2. Selecciona un punto de partida
   - El algoritmo comienza con un punto arbitrario no visitado del conjunto de datos.
3. Examina el barrio
   - Recupera todos los puntos dentro de la distancia ϵ del punto inicial.
   - Si el número de puntos vecinos es inferior a MinPts, el punto se etiqueta como ruido (por ahora).
   - Si hay al menos MinPts puntos dentro de una distancia ϵ, el punto se marca como punto núcleo y se forma un nuevo conglomerado.
4. Expandir el clúster
   - Todos los vecinos del punto central se añaden al clúster.
   - Si es un punto central, sus vecinos se añaden al conglomerado recursivamente.
   - Si no es un punto central, se marca como punto fronterizo y la expansión se detiene.
5. Repite el proceso

- El algoritmo se desplaza al siguiente punto no visitado del conjunto de datos.
- Los pasos 3–4 se repiten hasta que se hayan visitado todos los puntos.

6. Finalizar las agrupaciones
   - Una vez procesados todos los puntos, el algoritmo identifica todos los conglomerados.
   - Los puntos etiquetados inicialmente como ruido pueden ser ahora puntos fronterizos si están a ϵ de distancia de un punto central.
7. Manipulación del ruido
   - Los puntos que no pertenecen a ningún conglomerado permanecen clasificados como ruido.

Para nuestro conjunto de datos DBSCAN podría etiquetar un lanzamiento (pitcheo) dado su agrupamiento de lanzamientos similares sin decirle cuántos tipo de pitcheo hay, por ejemplo:

- Encontrar un grupo con alta velocidad y poco movimiento $\rightarrow$ Fastball
- Un lanzamiento con menor velocidad y más curvatura $\rightarrow$ Slider

En nuestro conjunto de datos afortunadamente si contamos con una columna llamada *"pitch type"* pero si no tuvieramos esta, podríamos crearla a través de DBSCAN.

En este mismo caso, podríamos usar el índice Silhouette ya que este mide qué tan bien estan separados los grupos encontrados. Sus valores van desde -1 hasta 1 y entre más cercanos a 1 indican una mejor separación de los clústeres y los cercanos a -1 la peor separación. Este es una alternativa a métodos como el del codo. Este índice es el ideal si utilizaramos un método como DBSCAN.

4. PCA (Principal Component Analysis)

Dado que nuestros datos provienen de una base de datos con más de 100 campos, utilizaremos el método PCA (Análisis de Componentes Principales) para reducir el número de variables a una conjunto de datos más pequeño.

4.1. ¿Por qué realizar PCA a nuestros datos?

El análisis de componentes principales sirve para reducir las variables en un conjunto grande de datos es útil para reducir las variables que pasarán a llamarse componentes principales.

Para nuestro conjunto de datos al tener más de 100 variables nos será muy útil esto. En el cuadro 1 (p. 3) se observa un resúmen del por qué será útil.

Cuadro 1: Utilidad del Análisis de Componentes Principales (PCA) en los datos de *Pybaseball*.

Objetivo	**¿Qué hace PCA?**	**¿Por qué sirve?**
Simplificar los datos	Reduce muchas variables correlacionadas a unas pocas componentes.	Evita duplicar información (por ejemplo, `release_speed`, `effective_speed` y `velocity` suelen estar correlacionadas).
Identificar patrones globales	Encuentra direcciones de máxima variación.	Muestra qué tipo de combinaciones de variables "explican" más el comportamiento de los lanzamientos.
Preparar los datos para otros algoritmos	Quita ruido y variables redundantes antes de aplicar clustering (`DBSCAN`, `KMeans`, etc.).	Mejora el desempeño y evita errores por alta dimensionalidad.

Referencias

[1] DataCamp. *DBSCAN Clustering Algorithm: Explained.* Disponible en: https://www.datacamp.com/es/tutorial/dbscan-clustering-algorithm. Consultado el 25 de octubre de 2025.

[2] IBM. *Aprendizaje no supervisado: Qué es y cómo funciona.* Disponible en: https://www.ibm.com/mx-es/topics/unsupervised-learning. Consultado el 25 de octubre de 2025.