# Приклад роботи алгоритму DBSCAN

Алгоритм DBSCAN - це алгоритм кластеризації даних, який ґрунтується на щільності даних. Основна ідея алгоритму DBSCAN полягає у виявленні щільних областей (кластерів) всередині даних та ідентифікації викидів (шуму) на основі розподілу точок у просторі.

#### Основні концепції та ідеї алгоритму DBSCAN:

* DBSCAN починає з вибору випадкової точки даних і визначає, чи є ця точка "ядром" (core point) щільного кластера. Точка вважається ядром, якщо в заданому радіусі (epsilon, ε) від неї міститься не менше певної кількості точок (MinPts).
* Якщо точка є ядром, а всі точки в її радіусі також є ядрами (або досяжними з інших ядер), ці точки утворюють одну щільну область (кластер).
* Точки, що знаходяться всередині радіусу (ε) від ядра, але самі не є ядрами, вважаються граничними точками. Вони можуть бути частиною кластера, якщо пов'язані з ядром, але не можуть утворити окремі кластери.
* Точки, які не є ядрами і не знаходяться всередині радіусу (ε) від жодного ядра, вважаються викидами або шумом і не включаються до кластерів.

Для кращого розуміння алгоритму нижче наведено наочний приклад.

У банкетному залі великий колектив проводить святковий корпоратив. Хтось стоїть поодинці, але більшість - з колегами. Деякі компанії просто товпляться юрбою, деякі – танцюють у великому колі або в довгому ланцюжку. Припустимо, потрібно розділити людей в залі на групи (рис. 4.2).

Рис. 4.2. Наочний приклад
роботи алгоритму DBSCAN

Але як виділити групи настільки різної форми, та ще й не забути про одинаків? Потрібно оцінити щільність натовпу навколо кожної людини. Напевно, якщо щільність натовпу між двома людьми вище за певний поріг, то вони належать до одної компанії. Наприклад, люди, які утворюють танцювальний ланцюжок, будуть відноситися до однієї групи, навіть якщо щільність ланцюжка між ними змінюється в деяких межах.

Визначимо, що поруч з деякою людиною зібрався натовп, якщо близько до неї знаходиться кілька інших осіб. Очевидно, що потрібно задати два параметри. Що значить «близько»? Для цього обирається певна інтуїтивно зрозуміла відстань. Наприклад, якщо люди можуть доторкнутися один до одного, то вони знаходяться близько (до одного метра). Тепер, скільки саме «кілька інших осіб»? Припустимо, три людини.

Всі люди, в кого є хоча б три сусіда в радіусі метра від нього, беруть в руки зелені прапорці. Тепер вони є кореневими елементами, що формують групи. Наступними є особи, в околі яких менше за трьох сусідів. Вибираються ті, в яких принаймні один з сусідів тримає зелений прапор. Зазвичай, ці люди є на межі груп і вони отримують жовті прапори (4.3).



Рис. 4.3. Визначення міри сусідства

Залишилися одинаки, в околі яких немає сусідів, їм роздаються червоні прапори. Вважається, що вони не належать до жодної групи.

Таким чином, якщо від однієї людини до іншої можна створити ланцюжок «зелених» людей, то ці дві людини належать до одної групи. Очевидно, що всі подібні скупчення розділені або порожнім простором або людьми з жовтими прапорами. Їх можна пронумерувати: кожен в групі №1 може досягти за ланцюжком рук кожного іншого в групі №1, але нікого в №2, №3 і так далі. Те ж для інших груп.

Якщо поруч з людиною з жовтим прапорцем є тільки один «зелений» сусід, то він буде належати до тої групи, до якої належить його сусід. Якщо таких сусідів кілька, і в них різні групи, то доведеться вибирати. Тут можна скористатися різними методами – наприклад, подивитися, хто з сусідів найближчий. Доведеться якось обходити крайові випадки, але нічого страшного (рис. 4.4).



Рис. 4.4. Визначення груп за мірою сусідства

Як правило, немає сенсу позначати всі кореневі елементи натовпу відразу. Оскільки від кожного кореневого елемента групи можна провести ланцюжок до кожного іншого, то не важливо, з якого починати обхід. Тут краще підходить ітеративний варіант:

1. Обирається випадкова людина з натовпу.

2. Якщо поруч з нею менше за три сусіди, людина переноситься до переліку можливих одинаків і вибирається хтось інший.

3. інакше:

* Ця людина вилучається з переліку людей, яких треба обійти.
* Ця людина отримує зелений прапорець і створює нову групу, в якій вона поки що є єдиною особою.
* Обходимо всіх сусідів даної людини. Якщо сусід вже є в переліку потенційних одинаків або поруч з ним мало інших людей, то це свідчить про межу скупчення. Для простоти можна відразу помітити сусіда жовтим прапором, приєднати до групи і продовжити обхід. Якщо сусід теж виявляється «зеленим», то він не започатковує нову групу, а приєднується до вже створеної. Крім того, він додається до переліку обходу сусідів сусіда. Цей пункт повторюється, поки перелік обходу не стане порожнім.

4. Кроки 1-3 повторюються, поки не будуть обійдені всі люди.

5. Розглядається перелік одинаків. Якщо на кроці 3 вже позначено осіб, які знаходяться на межах груп, то в переліку залишилися тільки викиди-одинаки. На цьому алгоритм завершено.

Як і класифікація, кластеризація може використовуватися як детектор аномалій. Тобто, до аномалій відносять одинаків, що не попадають в жодний кластер.

#### Алгоритм DBSCAN має кілька ключових переваг:

* Спроможний виявляти кластери довільної форми та розміру.
* Здатний виділяти викиди та ігнорувати їх при формуванні кластерів.
* Його ефективність дозволяє працювати з більшими обсягами даних.

Однак для успішного застосування DBSCAN важливо правильно налаштувати параметри, такі як радіус (ε) та мінімальна кількість точок (MinPts), що може вимагати деякого досвіду та розуміння структури даних.