Тестовые вопросы по AgglomerativeClustering

1. AgglomerativeClustering - это алгоритм:

**AgglomerativeClustering** — это алгоритм иерархической кластеризации, который реализует метод "снизу вверх". Он начинает с каждого объекта данных как отдельного кластера и объединяет пары кластеров, основываясь на выбранной метрике сходства, пока не будет достигнуто заданное количество кластеров или другие критерии остановки.

Основные параметры:

**n\_clusters**  
Количество кластеров, к которым алгоритм должен прийти.  
По умолчанию n\_clusters=2.

**affinity**  
Мера расстояния между точками данных. Возможные значения:

* 'euclidean' (по умолчанию),
* 'l1' (манхэттенское расстояние),
* 'l2' (евклидово расстояние),
* 'cosine' (косинусное расстояние),
* 'precomputed' (предварительно вычисленные расстояния).

**linkage**  
Метод для определения расстояния между кластерами. Возможные значения:

* 'ward' (минимизация суммарной дисперсии, работает только с 'euclidean'),
* 'complete' (максимальное расстояние между точками кластеров),
* 'average' (среднее расстояние между точками кластеров),
* 'single' (минимальное расстояние между точками кластеров).

**distance\_threshold**  
Максимальное расстояние между кластерами, при котором они объединяются. Используется, если n\_clusters не задан.

**compute\_full\_tree**  
Если True, строится полное дерево кластеризации, даже если задано n\_clusters.

**Иерархическая кластеризация** помогает визуализировать данные в виде дендрограммы, что полезно для анализа структуры данных.\

Плюсы:

Не требует задания начальных центров кластеров.

Подходит для работы с небольшими наборами данных.

Минусы:

Высокая вычислительная сложность при большом объеме данных.

Чувствительность к выбросам.

2. Какой подход использует AgglomerativeClustering для построения кластеров?

**AgglomerativeClustering** использует **агломеративный подход** (метод "снизу вверх") для построения кластеров. Этот подход заключается в следующем:

 **Инициализация:**  
Каждый объект данных рассматривается как отдельный кластер.

 **Объединение кластеров:**  
На каждом шаге алгоритм объединяет два наиболее близких кластера, основываясь на выбранной метрике сходства и методе связи (linkage).

 **Повторение:**  
Процесс продолжается до тех пор, пока все объекты не объединятся в один кластер или не будет достигнуто заданное количество кластеров (n\_clusters), либо заданный порог расстояния (distance\_threshold).

 **Результат:**  
Итоговая структура кластеров может быть представлена в виде **дендрограммы**, что позволяет визуально проанализировать, на каком этапе объединения были сформированы кластеры.

**Методы связи (linkage):**

* **Ward:** минимизирует увеличение дисперсии внутри кластеров.
* **Complete:** использует максимальное расстояние между объектами разных кластеров.
* **Average:** считает среднее расстояние между объектами кластеров.
* **Single:** минимизирует минимальное расстояние между объектами кластеров.

3. Что такое дендрограмма в контексте AgglomerativeClustering?

Дендрограмма — это графическое представление иерархического процесса кластеризации. Она показывает, как данные объединяются в кластеры на каждом шаге агломеративного алгоритма. Вертикальные оси отображают уровни расстояния или несходства между кластерами, на которых происходят их объединения. Это позволяет выбрать оптимальное количество кластеров, "обрезая" дендрограмму на определенном уровне.

4. Какой параметр в AgglomerativeClustering отвечает за количество кластеров?

Параметр n\_clusters задаёт количество кластеров, которые алгоритм должен выделить. Значение по умолчанию — 2.

5. Какой параметр в AgglomerativeClustering отвечает за метод связывания (linkage)?

Параметр linkage определяет метод, используемый для расчёта расстояния между кластерами.

6. Какие методы связывания (linkage) доступны в AgglomerativeClustering ?

Доступны следующие методы:

* ward: минимизация увеличения дисперсии внутри кластеров.
* complete: максимальное расстояние между точками разных кластеров.
* average: среднее расстояние между точками разных кластеров.
* single: минимальное расстояние между точками разных кластеров.

7. Какой метод связывания (linkage) минимизирует дисперсию внутри кластеров?

Метод ward минимизирует увеличение дисперсии внутри кластеров.

8. Какой атрибут обученного объекта AgglomerativeClustering содержит метки кластеров для каждой точки данных?

Атрибут labels\_ содержит метки кластеров для каждой точки данных.

9. Можно ли использовать AgglomerativeClustering для кластеризации данных с большим количеством признаков?

Теоретически можно, но на практике это может быть затруднительно, так как алгоритм имеет высокую вычислительную сложность (обычно O(n3)O(n^3)O(n3)) и плохо масштабируется на большие объемы данных и множество признаков. В таких случаях может потребоваться предварительное уменьшение размерности данных.

10. AgglomerativeClustering гарантированно находит глобальный оптимум?

Нет, алгоритм AgglomerativeClustering не гарантирует нахождения глобального оптимума, так как он следует жадному подходу, объединяя на каждом шаге только наиболее близкие кластеры.

Тестовые вопросы по DBSCAN

1. DBSCAN - это алгоритм:

**DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** — это алгоритм кластеризации, который группирует точки на основе плотности их расположения. Его основное преимущество заключается в том, что он автоматически определяет количество кластеров и хорошо справляется с выявлением кластеров сложной формы, а также устойчив к шумовым данным и выбросам.

* 1. Точки в кластере должны быть достаточно "плотно" сгруппированы, то есть каждая точка кластера должна находиться в пределах определённого радиуса от других точек.
  2. Точки, которые находятся в малонаселённых областях, считаются шумовыми.

Основные параметры DBSCAN:

eps (ε, радиус окрестности):

1. Определяет максимальное расстояние между двумя точками, чтобы они считались соседями.
2. Чем больше значение eps, тем больше вероятность, что точки будут объединены в один кластер.

min\_samples (минимальное количество точек):

1. Минимальное число точек в пределах радиуса eps, необходимое для того, чтобы точка считалась "основной" (core point).
2. Если число соседей меньше min\_samples, то точка считается либо граничной (border point), либо шумовой (noise point).

Типы точек в DBSCAN:

1. Основная точка (core point):

Точка, вокруг которой находится не менее min\_samples других точек в пределах радиуса eps.

1. Граничная точка (border point):

Точка, которая лежит в пределах радиуса eps от основной точки, но сама не имеет достаточно соседей, чтобы быть основной.

1. Шумовая точка (noise point):

Точка, которая не является ни основной, ни граничной и не входит в какой-либо кластер.

**Этапы работы алгоритма:**

 Выбирается начальная точка из набора данных.

 Проверяется количество соседей этой точки в пределах радиуса eps.

* Если число соседей >= min\_samples, то точка становится **основной**, и начинается формирование нового кластера.
* Если число соседей < min\_samples, то точка помечается как **шумовая**.

 Для каждой основной точки алгоритм рекурсивно добавляет соседние точки в кластер, пока не будут покрыты все плотные области данных.

 Повторяется, пока не будут обработаны все точки.

Преимущества DBSCAN:

**Выявление кластеров произвольной формы:** В отличие от KMeans, DBSCAN не ограничивается кластеризацией на основе сферических границ.

**Автоматическое определение шума:** Алгоритм автоматически исключает шумовые точки.

Не требует заранее задавать количество кластеров.

Недостатки DBSCAN:

**Чувствительность к параметрам eps и min\_samples:** Неправильный выбор параметров может привести к плохой кластеризации.

**Проблемы с высокоразмерными данными:** В таких данных сложно подобрать оптимальное значение eps.

**Плохая производительность на больших наборах данных:** Алгоритм требует вычисления расстояний между всеми парами точек.

Атрибуты обученного объекта DBSCAN:

**labels\_** — метки кластеров для каждой точки. Шумовые точки получают метку -1.

**core\_sample\_indices\_** — индексы основных точек.

**components\_** — координаты основных точек (ядра кластеров).

2. Какие два основных параметра используются в DBSCAN?

eps (эпсилон):

Радиус окрестности точки, в пределах которого алгоритм определяет плотность. Точки, находящиеся внутри этого радиуса, рассматриваются как потенциальные соседи. Чем меньше eps, тем более плотные кластеры будут обнаружены.

min\_samples (минимальное количество образцов):

Минимальное число точек, включая саму точку, которые должны находиться в пределах радиуса eps, чтобы точка считалась "основной точкой" (core point). Этот параметр определяет минимальную плотность, необходимую для формирования кластера.

3. Что такое "основная точка" (core point) в DBSCAN?

Основная точка — это точка, вокруг которой находится как минимум min\_samples точек в радиусе eps.

4. Что такое "граничная точка" (border point) в DBSCAN?

Граничная точка — это точка, которая находится в окрестности радиуса eps от основной точки, но вокруг неё самого недостаточно точек, чтобы она стала основной.

5. Что такое "шумовая точка" (noise point) в DBSCAN?

Шумовая точка — это точка, которая не является ни основной, ни граничной. Она считается выбросом и не входит ни в один кластер.

6. Какой параметр в DBSCAN отвечает за радиус окрестности?

Параметр eps.

7. Какой параметр в DBSCAN отвечает за минимальное количество точек в окрестности?

Параметр min\_samples.

8. Какие преимущества имеет DBSCAN по сравнению с KMeans?

 Не требует заранее задавать количество кластеров.

 Может обнаруживать кластеры произвольной формы.

 Устойчив к выбросам и шуму.

9. Какие недостатки имеет DBSCAN?

 Чувствителен к выбору параметров eps и min\_samples.

 Сложно работает с кластерами переменной плотности.

 Высокая вычислительная сложность на больших наборах данных.

10. Какой атрибут обученного объекта DBSCAN содержит метки кластеров для каждой точки данных?

Атрибут labels\_.

Тестовые вопросы по OPTICS

1. OPTICS - это алгоритм:

OPTICS (Ordering Points To Identify the Clustering Structure) — алгоритм кластеризации на основе плотности, предназначенный для выявления кластеров произвольной формы и шумовых данных. Он похож на DBSCAN, но предоставляет более гибкий подход к кластеризации, так как не требует фиксированного значения радиуса eps для всех кластеров.

Основная идея алгоритма заключается в построении упорядоченного списка точек данных на основе их плотности. Этот список используется для определения структуры кластеров и выделения областей с различной плотностью.

Как работает OPTICS:

Инициализация.

Алгоритм начинает с произвольной точки и вычисляет её плотность соседей (количество точек в радиусе eps).

Определение основных и достижимых расстояний.

1. **Основное расстояние (core distance):** минимальное расстояние, при котором точка становится "основной" (имеет не менее min\_samples соседей).
2. **Достижимость (reachability distance):** расстояние, необходимое для достижения другой точки через текущую.

Упорядочение точек.

Алгоритм упорядочивает точки на основе их достижимости, создавая упорядоченный список, который используется для визуализации и анализа.

Кластеризация.

Кластеры выделяются из упорядоченного списка, анализируя изменения в значениях достижимости.

Основные параметры OPTICS:

min\_samples:

1. Минимальное количество точек в радиусе eps, чтобы точка считалась "основной" (core point).
2. Этот параметр определяет минимальную плотность кластера.

max\_eps:

1. Максимальное расстояние, в пределах которого алгоритм рассматривает точки как соседей.
2. Если расстояние между точками превышает max\_eps, они не считаются связанными.

Metric:

1. Мера расстояния, используемая для определения близости между точками (например, Евклидово расстояние, Манхэттенское расстояние и т.д.).
2. По умолчанию используется Евклидово расстояние.

cluster\_method:

1. Метод выделения кластеров из упорядоченного списка.
2. xi — выделение кластеров на основе изменения плотности.
3. dbscan — выделение кластеров, как в DBSCAN.

eps (для dbscan метода):

Радиус окрестности, используемый для выделения кластеров, если выбран метод dbscan.

Xi:

Параметр для определения плотности в методе xi. Используется для выделения кластеров с различной плотностью.

leaf\_size:

Размер листа для оптимизации вычислений в KD-дереве или BallTree, которые используются для ускорения поиска соседей.

Основные преимущества OPTICS:

1. Способен выявлять кластеры с различной плотностью без необходимости фиксированного значения eps.
2. Идентифицирует шумовые точки.
3. Работает с данными произвольной формы.
4. Подходит для задач с большими объёмами данных и многомерными признаками.

Пример визуализации результата OPTICS:

Основным инструментом анализа результатов является **диаграмма достижимости (reachability plot)**. Эта диаграмма показывает, как изменяется достижимость точек, что позволяет визуально выделить кластеры (как "долины" на графике) и шумовые точки (как пики).

Недостатки OPTICS:

1. Алгоритм требует больше вычислительных ресурсов, чем DBSCAN.
2. Результаты чувствительны к выбору параметров, таких как min\_samples и xi.
3. Визуализация и интерпретация результатов может быть сложной при большом количестве данных.

2. Какое основное преимущество OPTICS по сравнению с DBSCAN?

Основное преимущество OPTICS заключается в том, что он устраняет необходимость точного задания параметра eps. OPTICS строит упорядоченный список точек с указанием их достижимости (reachability distance), что позволяет выделять кластеры с различной плотностью. Это делает алгоритм более универсальным и подходящим для задач, где плотность кластеров может варьироваться.

3. Что такое "достижимость" (reachability distance) в OPTICS?

Достижимость — это расстояние, необходимое для достижения точки из другой точки через плотные области. Формально, это максимальное значение между основным расстоянием (core distance) и евклидовым расстоянием до соседней точки. Достижимость указывает, насколько "плотно" связаны точки.

4. Что такое "основное расстояние" (core distance) в OPTICS?

Основное расстояние — это минимальное расстояние eps, при котором точка становится основной (core point), то есть имеет не менее min\_samples соседей в пределах этого радиуса. Если точка не является основной, основное расстояние не определено.

5. Какой параметр в OPTICS отвечает за минимальное количество точек в окрестности?

Параметр **min\_samples** задаёт минимальное количество точек в радиусе eps, чтобы точка считалась основной (core point).

6. Какой параметр в OPTICS отвечает за максимальное расстояние для рассмотрения соседей?

Параметр **max\_eps** определяет максимальное расстояние, в пределах которого рассматриваются соседи для вычисления плотности.

7. Что такое "упорядоченный список" (ordering) в OPTICS?

Упорядоченный список — это последовательность точек данных, отсортированных по достижимости. Этот список отражает структуру кластеров и их плотность, что позволяет визуально оценить распределение точек и выделить кластеры.

8. Как можно визуализировать результаты OPTICS?

Результаты OPTICS визуализируют с помощью **диаграммы достижимости (reachability plot)**, где по оси X откладывается индекс точки из упорядоченного списка, а по оси Y — её достижимость. Кластеры отображаются как "долины" на графике, а шумовые точки — как пики.

9. Как можно извлечь кластеры из упорядоченного списка OPTICS?

Кластеры можно извлечь путём анализа диаграммы достижимости или с использованием алгоритма **DBSCAN** на основе значений достижимости. Некоторые реализации OPTICS, такие как в библиотеке Scikit-learn, позволяют автоматически выделять кластеры на основе пороговых значений.

10. OPTICS чувствителен к выбору параметров?

OPTICS менее чувствителен к параметрам, чем DBSCAN, благодаря тому, что он не требует строгого задания фиксированного eps. Однако параметры **min\_samples** и **max\_eps** всё же могут значительно влиять на результаты, особенно в случаях сложной структуры данных.