Тестовые вопросы по association\_rules

1. Для чего используется функция ***association\_rules*** в библиотеке ***mlxtend.frequent\_patterns*** ?

Формирование ассоциативных правил

2. Какой тип данных принимает функция ***association\_rules*** в качестве основного аргумента?

Основным аргументом функции является DataFrame, содержащий частые наборы (frequent itemsets), который обычно получается в результате работы функций, таких как apriori или fpgrowth. Этот DataFrame должен содержать столбцы с элементами наборов (itemsets) и их поддержкой (support).

3. Какой параметр в функции ***association\_rules*** отвечает за метрику, используемую для оценки правил?

Параметр metric определяет метрику, которая будет использоваться для оценки правил.

metric="confidence"

4. Какие метрики можно использовать для оценки ассоциативных правил с помощью функции ***association\_rules*** ?

**Поддержка (Support):** Доля транзакций, содержащих определенный набор элементов.

Например, если набор {молоко, хлеб} встречается в 20% транзакций, его поддержка

равна 0.2.

**Доверие (Confidence):** Условная вероятность того, что транзакция, содержащая набор

элементов A, также содержит набор элементов B. Вычисляется как support(A ∪ B) /

support(A) . Например, если поддержка {молоко, хлеб} равна 0.2, а поддержка {молоко}

равна 0.5, то доверие правила {молоко} → {хлеб} равно 0.2 / 0.5 = 0.4.

**Подъем (Lift):** Показывает, насколько часто элементы A и B встречаются вместе по

сравнению с тем, как часто они встречались бы, если бы были независимыми.

Вычисляется как support(A ∪ B) / (support(A) \* support(B)) . Значение подъема больше

1 указывает на положительную корреляцию, меньше 1 - на отрицательную, а равное 1 – на независимость.

leverage (рычаг)

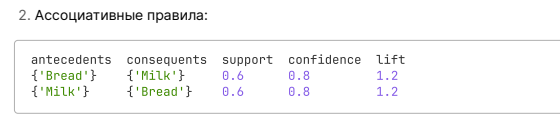
conviction (убежденность)

5. Какой параметр в функции ***association\_rules*** отвечает за минимальный порог метрики, при котором правило считается интересным?

Параметр min\_threshold

6. Какой тип данных возвращает функция ***association\_rules*** ?

Функция возвращает объект pandas.DataFrame, содержащий сгенерированные ассоциативные правила и их метрики.



7. Какие столбцы обычно присутствуют в DataFrame, возвращаемом функцией ***association\_rules*** ?

antecedents (левые части правил)

consequents (правые части правил)

support (поддержка всего правила)

confidence (доверие)

lift (подъем)

antecedent support (поддержка левой части правила)

consequent support (поддержка правой части правила)

Дополнительные метрики: leverage, conviction (если применимо).

8. Что представляют собой столбцы antecedents и consequents в DataFrame, возвращаемом функцией ***association\_rules*** ?

antecedents (левые части правил)

consequents (правые части правил)

antecedents — это наборы элементов, которые выступают в качестве условий (левой части правила).

consequents — это наборы элементов, которые следуют из условий (правой части правила).

9. Как можно отфильтровать правила, сгенерированные функцией ***association\_rules*** , по значению метрики (например, lift)?

rules = association\_rules(frequent\_itemsets, metric="lift", min\_threshold=1.0)

filtered\_rules = rules[rules['lift'] > 1.5]

10. Что нужно сделать перед использованием функции ***association\_rules*** , если у вас есть только исходные транзакционные данные?

Генерация частых наборов с порогом поддержки с помощью библиотеки apriori

Преобразовать транзакционные данные в формат one-hot encoding (например, с помощью TransactionEncoder).

Сгенерировать частые наборы элементов с помощью функции apriori или fpgrowth, указав минимальный порог поддержки (min\_support).

Использовать полученные частые наборы в функции association\_rules.

Тестовые вопросы по mlxtend.frequent\_patterns.fpmax

1. Что является основной задачей алгоритма FPMax?

FP-Max — это вариант FP-Growth, который фокусируется на получении максимальных наборов элементов. **Набор элементов X называется максимальным, если он является часто встречающимся и не существует часто встречающегося супершаблона, содержащего X.**

2. Что такое "максимальный часто встречающийся набор элементов" в контексте FPMax?

**Набор элементов X называется максимальным, если он является часто встречающимся и не существует часто встречающегося супершаблона, содержащего X.**

Другими словами, часто встречающийся шаблон X не может быть подшаблоном более крупного часто встречающегося шаблона, чтобы соответствовать определению *максимального набора элементов*.

3. Какой тип данных принимает функция fpmax из библиотеки mlxtend.frequent\_patterns ?

Функция fpmax принимает DataFrame в формате one-hot encoding, где строки представляют транзакции, а столбцы — возможные элементы, со значениями 1 (элемент присутствует) и 0 (элемент отсутствует).

4. Какой параметр в функции fpmax отвечает за минимальную поддержку?

Параметр min\_support

5. Какой параметр в функции fpmax позволяет использовать названия столбцов DataFrame вместо индексов?

Параметр use\_colnames=True (по умолчанию: False)

6. Какой тип данных возвращает функция fpmax ?

Функция возвращает объект pandas.DataFrame, который содержит максимальные часто встречающиеся наборы элементов и их поддержку.

7. Какой столбец в возвращаемом DataFrame содержит найденные максимальные часто встречающиеся наборы элементов?

Столбец itemsets

8. Какой столбец в возвращаемом DataFrame содержит значение поддержки для каждого максимального часто встречающегося набора?

Столбец support

9. По сравнению с алгоритмом FPGrowth, FPMax обычно:

Находит только максимальные часто встречающиеся наборы, что сокращает объем данных.

Может быть более эффективным в задачах, где интересны только максимальные наборы.

Упускает информацию о поддержке подмножеств, в отличие от FPGrowth.

10. Что нужно сделать перед использованием функции fpmax , если данные представлены в виде списка транзакций с текстовыми элементами?

Преобразовать данные в формат one-hot encoding, используя TransactionEncoder или аналогичный инструмент.

Убедиться, что DataFrame соответствует требованиям функции fpmax (значения 0 и 1).

Тестовые вопросы по KMeans

1. KMeans - это алгоритм:

KMeans — это алгоритм кластеризации, который относится к методам обучения без учителя (unsupervised learning).

2. Что является основной целью алгоритма KMeans?

KMeans — разделить данные на k кластеров таким образом, чтобы объекты внутри одного кластера были максимально похожи, а объекты из разных кластеров — максимально различны.

3. Как KMeans определяет сходство между объектами?

KMeans определяет сходство между объектами с помощью **евклидова расстояния** (по умолчанию), измеряя расстояние между точками данных и центроидами кластеров.

4. Что такое центроиды в алгоритме KMeans?

Центроиды — это центральные точки кластеров, которые представляют собой среднее значение всех объектов, принадлежащих к данному кластеру.

5. Как KMeans выбирает начальное расположение центроидов?

Начальное расположение центроидов выбирается случайным образом из данных. Для улучшения качества начальной инициализации часто используется метод **KMeans++**, который выбирает стартовые центроиды таким образом, чтобы они были максимально удалены друг от друга.

6. Что такое метод локтя (elbow method) в KMeans?

**Метод локтя** — это подход для определения оптимального числа кластеров k. Он основан на анализе зависимости значения инерции (суммы квадратов расстояний от точек до их центроидов) от числа кластеров. Оптимальное k соответствует "изгибу" на графике, где уменьшение инерции становится менее значительным.

7. Что такое метод силуэта (silhouette method) в KMeans?

Метод силуэта измеряет качество кластеризации, оценивая, насколько каждый объект хорошо относится к своему кластеру по сравнению с соседними кластерами.

Коэффициент силуэта принимает значения от -1 до 1:

* Значения близкие к 1 означают, что объект правильно классифицирован.
* Значения близкие к 0 указывают на неопределенность.
* Отрицательные значения означают, что объект классифицирован неверно.

8. Какой из следующих недостатков характерен для KMeans?

Чувствительность к выбросам и шуму.

Неоптимальность при неправильной инициализации центроидов.

Требование заранее задавать число кластеров k.

Предположение, что кластеры имеют сферическую форму и примерно равны по размеру.

9. Как можно уменьшить влияние выбросов на KMeans?

Использовать метод KMedoids, который минимизирует сумму расстояний до медианы вместо среднего.

Применять предварительную обработку данных (удаление выбросов).

Ограничить влияние выбросов с помощью алгоритмов, подобных MiniBatchKMeans.

10. KMeans гарантированно находит глобальный оптимум?

Нет, KMeans не гарантирует нахождение глобального оптимума, поскольку результат зависит от начальной инициализации центроидов.