**Тезисы главы 3 «Машинного обучения без учителя»**

**Два вида машинного обучения без учителя**: неконтролируемые преобразования, алгоритмы кластеризации.

**Неконтролируемые преобразования**

Применение неконтролируемых преобразований:

1. Сокращение размерности;
2. Поиск компонент из которых состоят данные (выделение тем из коллекций текстовых документов).

Алгоритмы машинного обучения без учителя часто используются:

1. в разведочных целях, когда специалист хочет лучше изучить сами данные
2. они служат этапом предварительной обработки данных для алгоритмов машинного обучения с учителем

Одним из самых простых и наиболее широко используемых алгоритмов не контролируемого обучения является **анализ главных компонентов** (principal component analysis, PCA).

Анализ главных компонент (PCA) используется для **сокращения размерности** и **выделение признаков**.

Работа алгоритма **PCA** основана на вращение данных, а затем удаление направлений, объясняющих незначительную дисперсию данных. Важно отметить, что РСА является методом машинного обучения без учителя и не использует какой-либо информации о классах при поиске поворота. Он просто анализирует корреляционные связи в данных. Корреляционная связь говорит о взаимосвязанности данных параметров.

Для **сокращения размерности** до 2-х измерений, используют PCA c первыми двумя главными компонентами. Идея, лежащая в основе выделения признаков, заключается в поиске нового представления данных, которое в отличие от исходного лучше подходит для анализа.

Метод «Собственных лиц» для **выделения признаков** заключается в использовании алгоритма PCA как преобразования способом поворота данных с последующим удалением компонент («выбеливание»), имеющих низкую дисперсию.

**Факторизация неотрицательных матриц (NMF)** – еще один алгоритм машинного обучения без учителя, цель которого – выделить полезные характеристики. Он работает так же, как PCA, а также его можно использовать для уменьшения размерности.

Факторизация — это операция разложения объекта (число, матрица) на его простые составляющие.

**t-SNE** -- ещё один алгоритм неконтролируемого преобразования – это множественное обучение. Существует класс алгоритмов визуализации, называемых алгоритмами множественного обучения (manifold learning algorithms), которые используют гораздо более сложные графические представления данных и позволяют получить визуализации лучшего качества. Особенно полезным является алгоритм t-SNE.

Алгоритмы множественного обучения в основном направлены на визуализацию и поэтому редко используются для получения более двух новых характеристик. Некоторые из них, в том числе t-SNE, создают новое представление обучающих данных, но при этом не осуществляют преобразования новых данных. Это означает, что данные алгоритмы **нельзя применить к тестовому набору**, они могут преобразовать лишь те данные, на которых они были обучены. Множественное обучение может использоваться для разведочного анализа данных, но редко используется в тех случаях, когда конечной целью является применение модели машинного обучения с учителем. Идея, лежащая в основе алгоритма t-SNE, заключается в том, чтобы найти двумерное представление данных, сохраняющее расстояния между точками наилучшим образом. t-SNE начинает свою работу со случайного двумерного представления каждой точки данных, а затем пытается сблизить точки, которые в пространстве исходных признаков находятся близко друг к другу, и отдаляет друг от друга точки, которые находятся далеко друг от друга. При этом t-SNE уделяет большее внимание сохранению расстояний между точками, близко расположенными друг к другу. Иными словами, он пытается сохранить информацию, указывающую на то, какие точки являются соседями друг другу.

Имейте в виду, что этот метод не использует информацию о метках классов: он является полностью неконтролируемым. Тем не менее он может найти двумерное представление данных, которое четко разграничивает классы, используя лишь информацию о расстояниях между точками данных в исходном пространстве.

**Кластеризация**

**Кластеризация k-средних** – один из самых простых и наиболее часто используемых алгоритмов кластеризации. Сначала выбирается число кластеров k. После выбора значения k алгоритм k-средних отбирает точки, которые будут представлять центры кластеров (cluster centers). Затем для каждой точки данных вычисляется его евклидово расстояние до каждого центра кластера. Каждая точка назначается ближайшему центру кластера. Алгоритм вычисляет центроиды (centroids) – центры тяжести кластеров. Каждый центроид – это вектор, элементы которого представляют собой средние значения характеристик, вычисленные по всем точкам кластера. Центр кластера смещается в его центроид. Точки заново назначаются ближайшему центру кластера. Этапы изменения центров кластеров и переназначения точек итеративно повторяются до тех пор, пока границы кластеров и расположение центроидов не перестанут изменяться, т.е. на каждой итерации в каждый кластер будут попадать одни и те же точки данных.

**Недостатки алгоритма** **k-средних.**

Каждый кластер определяется исключительно его центром, это означает, что каждый кластер имеет выпуклую форму. В результате этого алгоритм k-средних может описать относительно простые формы. Кроме того, алгоритм k-средних предполагает, что все кластеры в определенном смысле имеют одинаковый «диаметр», он всегда проводит границу между кластерами так, чтобы она проходила точно посередине между центрами кластеров.

Кроме того, алгоритм k-средних плохо работает, когда кластеры имеют более сложную форму. Например, случай двух кластеров в форме полумесяцев.

**Алгомеративная кластеризация** (agglomerative clustering) относится к семейству алгоритмов кластеризации, в основе которых лежат одинаковые принципы: алгоритм начинает свою работу с того, что каждую точку данных заносит в свой собственный кластер и по мере выполнения объединяет два наиболее схожих между собой кластера до тех пор, пока не будет удовлетворен определенный критерий остановки.

Критерий остановки, реализованный в scikit-learn – это количество кластеров, поэтому схожие между собой кластеры объединяются до тех пор, пока не останется заданное число кластеров. Есть несколько критериев связи (linkage), которые задают точный способ измерения «наиболее схожего кластера». В основе этих критериев лежит расстояние между двумя существующими кластерами.

Изначально количество кластеров равно количеству точек данных. Затем на каждом шаге объединяются два ближайших друг к другу кластера.

Из-за своего способа работы алгоритм агломеративной кластеризации **не может вычислить прогнозы для новых точек** данных. Поэтому алгоритм агломеративной кластеризации не имеет метода predict.

**Иерархическая кластеризация.**

Результатом агломеративной кластеризации является иерархическая кластеризация (hierarchical clustering). Кластеризация выполняется итеративно, и каждая точка совершает путь от отдельной точки-кластера до участника итогового кластера. На каждом промежуточном шаге происходит кластеризация данных (с разным количеством кластеров).

К сожалению, алгоритм агломеративной кластеризации по-прежнему не в состоянии обработать сложные данные типа набора two\_moons. Чего нельзя сказать о DBSCAN

**DBSCAN** – плотностный алгоритм кластеризации пространственных данных с присутствием шума.

Основные преимущества алгоритма DBSCAN заключаются в том, что пользователю *не нужно заранее задавать количество кластеров*, алгоритм может выделить кластеры сложной формы и способен определить точки, которые не принадлежат какому-либо кластеру. DBSCAN работает немного медленнее, чем алгоритм агломеративной кластеризации и алгоритм k-средних, но также может масштабироваться на относительно большие наборы данных.

DBSCAN определяет точки, расположенные в «густонаселенных» областях пространства характеристик, когда многие точки данных расположены близко друг к другу. Эти области называются плотными (dense) областями пространства характеристик. Идея алгоритма DBSCAN заключается в том, что *кластеры образуют плотные области данных, которые отделены друг от друга относительно пустыми областями*.

Параметр **eps** чуть более важен, поскольку он определяет, что подразумевается под «близостью» точек друг к другу. Очень маленькое значение eps будет означать отсутствие ядерных точек и может привести к тому, что все точки будут помечены как шумовые. Очень большое значение eps приведет к тому, что все точки сформируют один кластер.

Значение **min\_samples** главным образом определяет, будут ли точки, расположенные в менее плотных областях, помечены как выбросы или как кластеры. Если увеличить значение min\_samples, все, что могло бы стать кластером с количеством точек, не превышающим min\_samples, будет помечено как шум. Поэтому значение min\_samples задает минимальный размер кластера.

Несмотря на то, что в алгоритме **DBSCAN** не нужно явно указывать количество кластеров, значение eps неявно задает количество выделяемых кластеров. Иногда подобрать оптимальное значение eps становится проще после масштабирования данных с помощью StandardScaler или MinMaxScaler, так как использование этих методов масштабирования гарантирует, что все характеристики будут иметь одинаковый масштаб.

**Тезисы главы 4 «Типы данных и конструирование признаков»**

Все признаки делятся на **непрерывные** и **категорийные**. Непрерывные представлены числом с плавующей точкой. Категорийные признаки известны как дискретные, поскольку не имеют числовых значений.

Вопрос оптимальной подготовки данных для конкретного прикладного применения известен под названием feature engineering (**конструирование признаков**) и является одной из главных задач для специалистов по машинному обучению, пытающихся решить реальные проблемы.

**Прямое кодирование (дамми-переменные).**

На сегодняшний момент наиболее распространенным способом представления категориальных переменных является прямое кодирование или, если перевести дословно, кодирование с одним горячим состоянием (one-hot-encoding или one-out-of-N encoding). Идея, лежащая в основе прямого кодирования, заключается в том, чтобы заменить категориальную переменную одной или несколькими новыми признаками, которые могут принимать значения 0 и 1. Значения 0 и 1 придают смысл формуле линейной бинарной классификации (а также всем остальным моделям в scikit-learn) и с помощью дамми-переменных мы можем выразить любое количество категорий, вводя по одному новому признаку для каждой категории.