**1. Что такое обучение без учителя?**

* **Определение:** Это тип машинного обучения, где алгоритмы работают с данными, не имеющими заранее известных "правильных ответов" или меток.
* **Отличие от обучения с учителем:** В обучении с учителем есть размеченные данные (входные данные + правильные ответы). Здесь же есть только входные данные.
* **Цель:** Алгоритму необходимо самостоятельно найти скрытые закономерности, структуру, группы или аномалии в данных. Он "учится", извлекая знания из самих данных.

**2. Основные типы обучения без учителя (рассмотренные в ЛР)**

В данной работе рассматриваются два ключевых направления:

**2.1. Неконтролируемые преобразования (Unsupervised Transformations)**

* **Суть:** Алгоритмы, которые создают новое, измененное представление исходных данных. Цель – сделать данные более удобными для анализа человеком или для других алгоритмов машинного обучения.
* **Основные применения:**
  + **Сокращение размерности (Dimensionality Reduction):**
    - **Проблема:** Данные могут иметь очень много признаков (размерностей). Это усложняет анализ, требует больше вычислений и может ухудшать качество моделей ("проклятие размерности").
    - **Решение:** Уменьшение количества признаков при сохранении наиболее важной информации.
    - **Пример:** Визуализация многомерных данных в 2D или 3D пространстве.
  + **Поиск компонент (Component Extraction) / Выделение признаков (Feature Extraction):**
    - **Проблема:** Исходные признаки могут быть неоптимальны для решения задачи.
    - **Решение:** Создание новых, более информативных признаков из исходных.
    - **Пример:** Выделение тем из коллекции текстовых документов (например, "политика", "спорт").

**2.2. Алгоритмы кластеризации (Clustering Algorithms)**

* **Суть:** Алгоритмы, которые автоматически группируют схожие объекты данных в кластеры.
* **Принцип:** Объекты внутри одного кластера должны быть максимально похожи друг на друга, а объекты из разных кластеров – максимально отличаться.
* **Неизвестные параметры:** Алгоритм не знает заранее, сколько кластеров должно быть или какие характеристики определяют принадлежность к кластеру.
* **Пример:** Группировка фотографий людей в социальной сети, где все фотографии одного человека попадают в один кластер.

**3. Проблемы и особенности обучения без учителя**

* **Оценка качества:** Главная сложность. Так как нет "правильных ответов", трудно объективно оценить, насколько хорошо алгоритм справился с задачей. Часто требуется экспертная оценка или анализ полезности результата для последующих задач.
* **Разведочный анализ данных:** Часто используется как первый шаг для понимания структуры данных, выявления выбросов или интересных закономерностей.
* **Предварительная обработка:** Результаты обучения без учителя могут служить входными данными для моделей обучения с учителем, улучшая их производительность.

**4. Предварительная обработка и масштабирование данных**

Это важный этап, часто предшествующий применению многих ML-алгоритмов. В контексте ЛР, методы масштабирования рассматриваются как неконтролируемые, так как они не используют метки классов.

* **Зачем нужно масштабирование?**
  + Многие алгоритмы (например, SVM, нейронные сети, методы, основанные на расстоянии) чувствительны к масштабу признаков.
  + Признаки с большими значениями могут непропорционально сильно влиять на результат, даже если они не более важны.
  + Масштабирование приводит все признаки к сопоставимому диапазону.
* **Основные методы масштабирования (скейлеры) из scikit-learn:**
  + StandardScaler: Делает так, чтобы у каждого признака среднее значение стало 0, а стандартное отклонение (разброс) – 1.
  + RobustScaler: Похож на StandardScaler, но использует медиану и квартили. Менее чувствителен к редким аномальным значениям (выбросам).
  + MinMaxScaler: Самый простой – "сжимает" или "растягивает" каждый признак так, чтобы все его значения лежали строго в диапазоне от 0 до 1.
  + Normalizer: Работает по-другому! Он масштабирует не каждый *признак* (столбец), а каждый *объект* (строку) так, чтобы его "длина" (евклидова норма) стала равна 1. Это полезно, когда важно направление вектора признаков, а не его величина.
* **Ключевое правило применения скейлеров (и любых препроцессоров):**
  + **Обучение (fit) скейлера:** Скейлер "обучается" (вычисляет необходимые статистики, например, min/max или среднее/std) **только на обучающей выборке (X\_train)**.
  + **Применение (transform) скейлера:** Обученный скейлер используется для преобразования **и обучающей (X\_train), и тестовой (X\_test) выборок**.
  + **Почему это критически важно?**
    - **Избегание утечки данных:** Информация из тестового набора не должна влиять на процесс обучения (включая настройку скейлера). Тестовый набор имитирует "новые, невидимые" данные.
    - **Сохранение структуры данных:** Преобразование должно быть одинаковым для всех частей данных, чтобы модель, обученная на преобразованных обучающих данных, могла корректно работать с аналогично преобразованными тестовыми данными.
  + **Иллюстрация (Рис. 2 в ЛР):** Показывает, что если масштабировать обучающий и тестовый наборы отдельно (обучая скейлер на каждом из них), то относительное положение точек данных искажается, что делает сравнение и оценку модели некорректными.
* **Метод fit\_transform():**
  + Удобная комбинация методов fit() и transform().
  + Применяется **только к обучающей выборке**: X\_train\_scaled = scaler.fit\_transform(X\_train).
  + Для тестовой выборки всегда используется только transform(): X\_test\_scaled = scaler.transform(X\_test).

**5. Влияние предварительной обработки на обучение с учителем**

* **Практический результат:** Правильная предварительная обработка (например, масштабирование) часто **значительно улучшает** качество моделей обучения с учителем.
* **Пример из ЛР (SVM):**
  + Точность SVM на исходных, немасштабированных данных: ~0.63.
  + Точность SVM на данных, масштабированных с помощью MinMaxScaler или StandardScaler: ~0.96-0.97.
* **Причина (для SVM):** SVM чувствителен к расстояниям между точками. Масштабирование обеспечивает, что все признаки вносят сопоставимый вклад в расчет расстояний, что позволяет SVM построить более оптимальную разделяющую гиперплоскость.