Машинное обучение без учителя включает в себя все виды машинного обучения, когда ответ неизвестен и отсутствует учитель, указывающий ответ алгоритму. В машинном обучении без учителя есть лишь входные данные и алгоритму необходимо извлечь знания из этих данных.

ТИПЫ МАШИННОГО ОБУЧЕНИЯ БЕЗ УЧИТЕЛЯ

Рассмотрим 2 вида машинного обучения без учителя:

\*преобразования данных

\*кластеризация

**Неконтролируемые преобразования** – алгоритмы, которые создают новое представление данных, которое в отличие от исходного будет работать легче. Общераспространенное применение неконтролируемых преобразований – сокращение размерности. Мы берем высокоразмерное представление данных, состоящее из мн-ва признаков, и находим новый способ представления данных, обобщая основные характеристики и получая меньшее количество признаков. Общее применение сокращение размерности – получение 2мерного пространство в целях визуализации.

**Алгоритмы кластеризации** – разбивают данные на отдельные группы схожих между собой элементов.

ПРОБЛЕМЫ МАШИННОГО ОБУЧЕНИЯ

Главная проблема – оценка полезности инфы, извлеченной алгоритмом. Алгоритмы машинного обучения без учителя применяются к данным, которые не содержат никаких меток => мы не знаем каким должен быть правильный ответ. Поэтому сложно судить о качестве работы модели.

=> Алгоритмы машинного обучения без учителя часто используются в разведочных целях, когда хотим лучше изучить сами данные.

Изучение нового представления данных может повысить правильность алгоритмов машинного обучения с учителем или может привести к снижению времени вычислений и потребления объема памяти.

Перед тем как работать с реальными алгоритмами – рассмотрим некоторые простые методы предварительной обработки данных. Хотя предварительная обработка данных и масштабирование часто применяются вместе с алгоритмами контролируемого обучения, методы масштабирования не используют учителя, что делает их методами неконтролируемого обучения.

ПРЕДВАРИТЕЛЬНАЯ ОБРАБОТКА И МАСТАБИРОВАНИЯ

Некоторые алгоритмы (SVM, нейронные сети) очень чувствительны к масштабированию данных. Поэтому обычной практикой является преобразование признаков для того, чтобы итоговый результат был подходящим. Часто достаточно масштабирования признаков и корректировки данных. Продемонстрируем различные способы масштабирования и предварительной обработки данных [1]

График соответствует синтетическому набору данных с 2 признаками. Первый признак (ось х) принимает значения от 10 до 15. Второй признак (ось у) принимает значения в диапазоне от 1 до 9.

Эти 4 графика показывают 4 различных способа преобразования данных, которые дают более стандартные диапазоны значений.

* Применение StandartScaler гарантирует, что для каждого признака среднее будет равно 0, а дисперсия – 1, в результате чего все признаки будут иметь один и тот же масштаб. Однако это масштабирование не гарантирует получение каких-то конкретных минимальных и максимальных значений признаков.
* RobustScaler аналогичен предыдущему в том плане, что в результате применения признаки будут иметь 1 и тот же масштаб. Одна вместо среднего и дисперсии использует медиану и *квартили.* Это позволяет игнорировать т.д-х, которые сильно отличаются от остальных. Эти странные точки данных называют **выбросами** и могут стать проблемой для остальных методов масштабирования.

**Квартили –** такое число, при котором половина значение мн-ва меньше этого числа, а вторая половина – больше. Нижний квартиль – число, ниже которого находится четверть значений. Верхний квартиль – число, выше которого находится четверть значений.

* MinMaxScaler сдвигает данные таким образом, что все признаки находились строго от 0 до 1.. для двумерного набора данных это означает, что все данные помещаются в прямоугольник, образованный осью х с диапазоном значений от 0 и 1 и осью у с диапазоном значений от 0 и 1
* Normalizer масштабирует каждую точку данных таким образом, чтобы вектор признаков имел евклидову длину 1. Т.е.он проецирует точку данных на окружность с радиусом 1. Вектор умножается на иверсию своей длины. Такая нормализация используется когда важным является не длина вектора признаков, а направление.

ПРИМЕНЕНИЕ ПРЕОБРАЗОВАНИЙ ДАННЫХ

Используем набор данных рака. Методы предварительной обработки обычно применяются перед использованием алгоритма машинного обучения с учителем.

Допустим, нужно применить ядерный метод опорных векторов к набору данных рака и использовать MinMaxScaler для предварительной обработки данных. [2].

Начнем с того, что загрузим наш набор данных и разобьем его на тренировочный и тестовый для того, чтобы оценить качество модели, которую построим с помощью контролируемого обучения после предварительной обработки.

В наборе 569 точек данных, каждая представлена 30 признаками. Получаем 426 примеров в обучающей выборке и 143 примера в тестовой выборке.

Импортируем класс, который осуществляет предварительную обработку и создаем его экземляр [3].

С помощью метода fir подгоняем scaler на обучающих данных. Для MinMaxScaler этот метод вычисляет минимальное и максимальное значения каждого признака на обучающем наборе. В отличие от классификаторов и регрессоров при вызове метода scaler fit работает с обучающими данными а ответы не используются [4]

Чтобы применить преобразование, которое мы подогнали, т.е.отмасштабировать обучающие данные, необходимо воспользоваться методом transform [5].

Преобразованные данные имеют такую же форму что и исходные – признаки просто смещены и отмасштабированы. Теперь все признаки принимают значение в диапазоне от 0 до 1, как и требовалось.

Чтобы применить метод ядерных векторов к масштабированным данным, мы должны преобразовать еще тестовый набор. Это снова делается с помощью вызова метода transform, но на этот раз для текстовых данных. [6]. После масштабирования минимальные и максимальные значения признаков могут выходить за пределы 0 и 1, т.к.MinMAxScaler всегда принимают одинаковое преобразование к обучающему и тестовому наборам. Это означает, что метод transform всегда вычитает минимальное значение, вычисленное для обучающего набора и делит на ширину диапазона, также вычисленную для обучающего набора. Минимальное значение и ширина диапазона для обучающего набора могут отличаться от минимального значения и ширины диапазона для тестового набора.

МАСШТАБИРОВАНИЕ ОБУЧАЮЩЕГО И ТЕСТОВОГО НАБОРОВ ОДИНАКОВЫМ ОБРАЗОМ

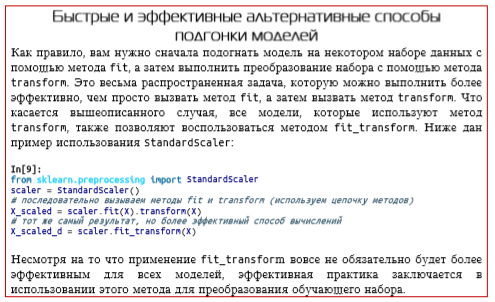
Чтобы модель контролируемого обучения работала на тестовом наборе, важно преобразовать обучающий и тестовый наборы одинаковым образом. Пример показывает, что произошло бы, если бы мы использовали минимальное значение и ширину диапазона, отдельно вычисленные для тестового набора. [7]. (большой код)

1ый график – немасштабированный 2мерный массив данных, наблюдения обучающего набора показаны кружками, а тестового – треугольниками.

2ой график – те же самые данные, но отмасштабированы с помощью MinMaxScaler. Здесь мы вызвали метод fit для обучающего набора, а затем вызвали метод transform для обучающего и тестового наборов. Набор данных на втором графике идентичен набору с первого графика, изменились лишь метки осей. Теперь все признаки принимают значения от 0 до 1.

3ий график – показывает что будет если мы отмасштабируем тестовый и обучающий наборы от отдельности. Теперь набор данных выглядит иначе. Тестовые точки сместились, т.к. масштабированы по-другому. Мы изменили положение данных произвольным образом.

НЕ НАДО



ВЛИЯНИЕ ПРЕДВАРИТЕЛЬНОЙ ОБРАБОТКИ НА МАШИННОЕ ОБУЧЕНИЕ С УЧИТЕЛЕМ

Посмотрим как влияет МинМаксСкалер на обучение методом ядерных вектором. Для сравнения снова подгоним SVC на исходных данных [8].

Теперь отмасштабируем данные с помощью MinMaxScaler перед тем как подгонять метод опорных векторов[9]. Эффект масштабирования данных весьма существенен. Хотя масштабирование данных не предполагает каких-либо сложных математических расчетов, эффективная практика заключается в том, чтобы использовать методы масштабирования, предлагаемые scikit-learn, а не создавать их заново самостоятельно, поскольку легко ошибиться даже в этих простых вычислениях.

Кроме того, можно легко заменить один алгоритм предварительной обработки на другой, сменив имя используемого класса, поскольку все классы предварительной обработки имеют один и тот же интерфейс, состоящий из методов fit и transform[10]

Одним из самых простых и наиболее широко используемых алгоритмов контролируемого обучения является анализ главных компонент (principal component analysis, PCA). Кроме того, мы рассмотрим еще два алгоритма: факторизацию неотрицательных матриц (non-negative matrix factorization, NMF), которая обычно используется для выделения признаков, и стохастическое вложение соседей с распределением Стьюдента (t-distributed stochastic neighbor embedding, t-SNE), которое обычно используется для визуализации с использованием двумерных диаграмм рассеяния.