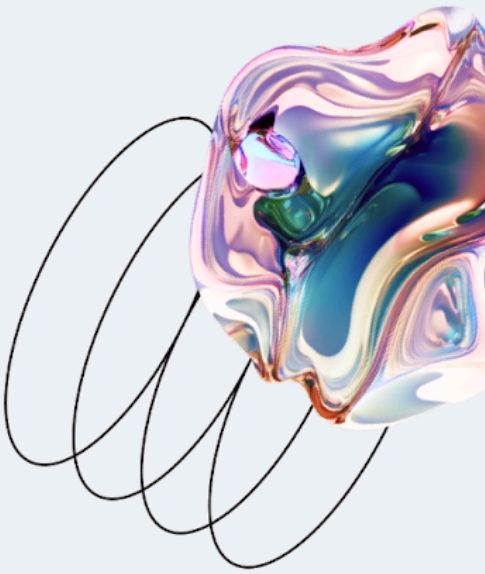
GeekBrains



Машинное обучение с учителем и без на примере классификации и кластеризации

Data Science

Оглавление

Введение

Всем привет! Это наша заключительная лекция на курсе «Введение в Data Science». Сегодня мы углубимся в тему машинного обучения и подробнее узнаем, что такое машинное обучение с учителем и без учителя. Разберемся, что такое классификация и кластеризация, что у них общего и в чем различия, а также построим модели классификации и кластеризации.

Словарь терминов

Обучение с учителем — один из основных типов машинного обучения. ML-алгоритм обучается на размеченных данных.

Неконтролируемое обучение — обучение машины с использованием информации, которая не классифицирована и не помечена. Позволяет алгоритму действовать на этой информации без руководства.

Классификация — процесс поиска модели или функции, которая помогает разделить данные на несколько категориальных классов, то есть дискретных значений.

Бустинг — это метод построения модели, который пытается построить сильный классификатор из числа слабых классификаторов.

Бэггинг — это алгоритм метаоценки, который сопоставляет каждый базовый классификатор со случайными подмножествами исходного набора данных, а затем объединяет их индивидуальные прогнозы (путем голосования или усреднения) для формирования окончательного прогноза.

Кластеризация — это задача разделения совокупности или точек данных на несколько групп таким образом, чтобы точки данных в одних и тех же группах были более похожи на другие точки данных в той же группе и отличались от точек данных в других группах.

Машинное обучение с учителем и без

Мы уже касались общего определение этих понятия в предыдущих лекциях и знаем об этих типах машинного обучения, но давайте немного углубимся в подробности.

Обучение с учителем — это контролируемое обучение: мы обучаем или тренируем машину, используя данные, которые хорошо размечены (отмечены правильным ответом). После этого машине предоставляется новый набор примеров (данных), чтобы алгоритм обучения с учителем анализировал их (набор обучающих примеров) и выдавал правильный результат из уже известных ему данных.

Предположим, что нам дали корзину фруктов. Первый шаг — обучить машину определять каждый фрукт один за другим следующим образом:

- Если форма округлая, сверху есть углубление, а цвет красный, фрукт будет помечен как **яблоко**.
- Если форма — длинный изогнутый цилиндр зелено-желтого цвета, он будет помечен как **банан**.

После обучения данных вы дали новый фрукт, скажем, банан из корзины, и попросили его идентифицировать.

Так как машина уже изучила фрукты из предыдущих данных, на этот раз она должна использовать их с умом. Сначала она классифицирует фрукт по форме и цвету, подтвердит название фрукта как «банан» и поместит его в категорию бананов. Таким образом, машина изучает данные из обучающих данных (корзина с фруктами), а затем применяет эти знания к тестовым данным (новые фрукты).
Обучение с учителем подразделяется на две категории алгоритмов:

- **Классификация** — проблема классификации возникает, когда выходная переменная представляет собой категорию, такую как «красный» или «синий», «заболевание» или «отсутствие заболевания».
- **Регрессия** — проблема регрессии возникает, когда выходная переменная представляет собой реальное значение, например «доллары» или «вес».

Типы:

- Регрессия
- Логистическая регрессия
- Классификация
- Наивные байесовские классификаторы
- K-NN (k ближайших соседей)
- Деревья решений
- Метод опорных векторов

Преимущества:

- Обучение под наблюдением позволяет собирать данные и выводить данные из предыдущего опыта.
- Помогает оптимизировать критерии эффективности с помощью опыта.

- Контролируемое машинное обучение помогает решать различные типы реальных вычислительных задач.
- Выполняет задачи классификации и регрессии.
- Позволяет оценить или сопоставить результат с новой выборкой.
- У нас есть полный контроль над выбором количества классов, которые мы хотим в обучающих данных.

Недостатки:

- Классификация больших данных может быть сложной задачей.
- Тренировка моделей обучения с учителем требует много вычислительного времени.
- Обучение с учителем не может справиться со всеми сложными задачами машинного обучения.
- Для построения моделей такого типа нужен предварительно размеченный набор данных.
- Нужен тренировочный процесс.

Неконтролируемое обучение — это обучение машины с использованием информации, которая не классифицирована и не помечена. Позволяет алгоритму действовать на этой информации без руководства. Задача машины — группировать несортированную информацию по сходствам, закономерностям и различиям без предварительной подготовки данных.

В отличие от контролируемого обучения, учитель не предоставляется. Поэтому машина ограничена возможностью самостоятельно находить скрытую структуру в неразмеченных данных.

Предположим, что алгоритму дали изображение с собаками и кошками, которых он никогда не видел. У машины нет представления об особенностях собак и кошек. Но ML-алгоритм может классифицировать их в соответствии с их сходствами, закономерностями и различиями. То есть мы можем легко разделить картинки с собаками и кошками на две части: первая — все фотографии с собаками, вторая — с кошками. Мы ничему не обучали машину раньше, не было тренировочных данных или примеров.

Это позволяет модели работать самостоятельно, чтобы обнаруживать закономерности и информацию, которые ранее не обнаруживались. В основном это касается неразмеченных данных.
Неконтролируемое обучение подразделяется на две категории алгоритмов:

- **Кластеризация** — проблема кластеризации заключается в том, что вы хотите обнаружить неотъемлемую группу данных. Например, группу клиентов по покупательскому поведению.
- **Ассоциация** — проблема изучения правил ассоциации заключается в том, что вы хотите обнаружить правила, которые описывают большие части ваших данных. Например, люди, которые покупают X, также склонны покупать Y.

Типы неконтролируемого обучения:

Кластеризация

1. Эксклюзивный (раздел)
2. Агломеративный
3. Перекрытие
4. вероятностный

Типы кластеризации:

1. Иерархическая кластеризация
2. Кластеризация K-средних
3. Анализ главных компонентов
4. Разложение по сингулярным значениям
5. Анализ независимых компонентов

Контролируемое и неконтролируемое машинное обучение:

Параметры	**Контролируемое машинное обучение**	**Неконтролируемое машинное обучение**
Входные данные	Алгоритмы обучаются на размеченных данных	Алгоритмы используются для данных, которые не размечены
Вычислительная сложность	Более простой метод	Вычислительно сложный
Точность	Высокая точность	Менее точный

Количество классов	Количество классов известно	Количество классов неизвестно
Анализ данных	Использует автономный анализ	Использует анализ данных в реальном времени
Используемые алгоритмы	Линейная и логистическая регрессия, случайный лес, метод опорных векторов, нейронная сеть и так далее	Кластеризация K-средних, иерархическая кластеризация, априорный алгоритм и так далее
Выход	Дается желаемый результат	Желаемый результат не указан
Тренировочные данные	Используйте обучающие данные, чтобы натренировать модель	Данные для обучения не используются
Сложная модель	Невозможно изучить более крупные и сложные модели	Можно изучать более крупные и сложные модели
Модель	Можем протестировать нашу модель	Не можем проверить нашу модель
Вызывается как	Обучение с учителем также называют классификацией	Неконтролируемое обучение также называют кластеризацией
Пример	Оптическое распознавание символов	Найти лицо на изображении

Преимущества обучения без учителя:

- Не требует, чтобы обучающие данные были размечены.
- Уменьшение размерности может быть легко достигнуто с помощью обучения без учителя.
- Способно находить ранее неизвестные закономерности в данных.

- **Гибкость**: неконтролируемое обучение гибкое в том смысле, что его можно применять к широкому кругу задач, включая кластеризацию, обнаружение аномалий и анализ правил ассоциации.
- **Исследование**: неконтролируемое обучение позволяет исследовать данные и обнаруживать новые и потенциально полезные закономерности, которые могут быть не очевидны с начала.
- **Низкая стоимость**: обучение без учителя часто дешевле, чем с учителем, потому что оно не требует размеченных данных, получение которых может занять много времени и средств.

Недостатки обучения без учителя:

- Трудно измерить точность или эффективность из-за отсутствия предопределенных ответов во время обучения.
- Результаты часто менее точные.
- Пользователь должен потратить время на интерпретацию и маркировку классов, которые следуют этой классификации.
- **Отсутствие руководства**: в неконтролируемом обучении отсутствуют руководство и обратная связь, обеспечиваемые помеченными данными, что может затруднить определение того, являются ли обнаруженные закономерности актуальными или полезными.
- **Чувствительность к качеству данных**: неконтролируемое обучение может быть чувствительно к качеству данных, включая пропущенные значения, выбросы и зашумленные данные.
- **Масштабируемость**: неконтролируемое обучение может быть дорогостоящим в вычислительном отношении, особенно для больших наборов данных или сложных алгоритмов, что может ограничить его масштабируемость.

Классификация

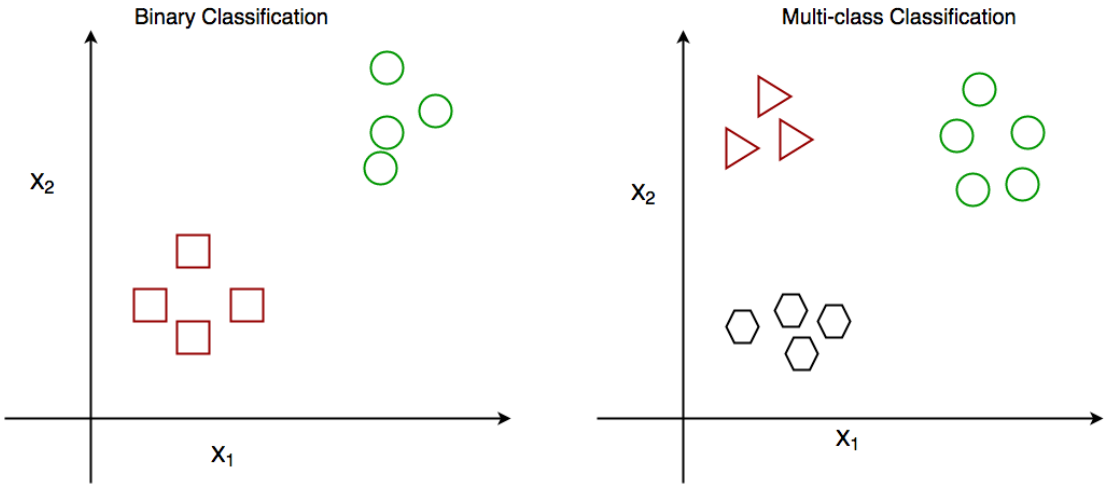
На прошлом уроке мы обсуждали регрессию и дерево решений — один из простейших методов классификации. Давайте поподробнее разберем, что такое классификация и чем она отличается от регрессии.

Классификация — это процесс поиска или обнаружения модели или функции, которая помогает разделить данные на несколько категориальных классов, то есть дискретных значений. Данные классифицируются по разным меткам в соответствии с

некоторыми параметрами, заданными во входных данных, а затем для данных прогнозируются метки.

- В задаче классификации мы должны прогнозировать дискретные целевые переменные (метки классов), используя независимые функции.
- В задаче классификации мы должны найти границу решения, которая может разделить разные классы в целевой переменной.

Производная функция отображения может быть продемонстрирована в виде правил «ЕСЛИ — ТО». Процесс классификации имеет дело с проблемами, когда данные могут быть разделены на двоичные или несколько дискретных меток. Предположим, что мы хотим предсказать возможность победы в матче команды A на основе некоторых записанных ранее параметров. Тогда будет две метки — да и нет.



Бинарная классификация и мультиклассовая классификация

Типы алгоритмов классификации

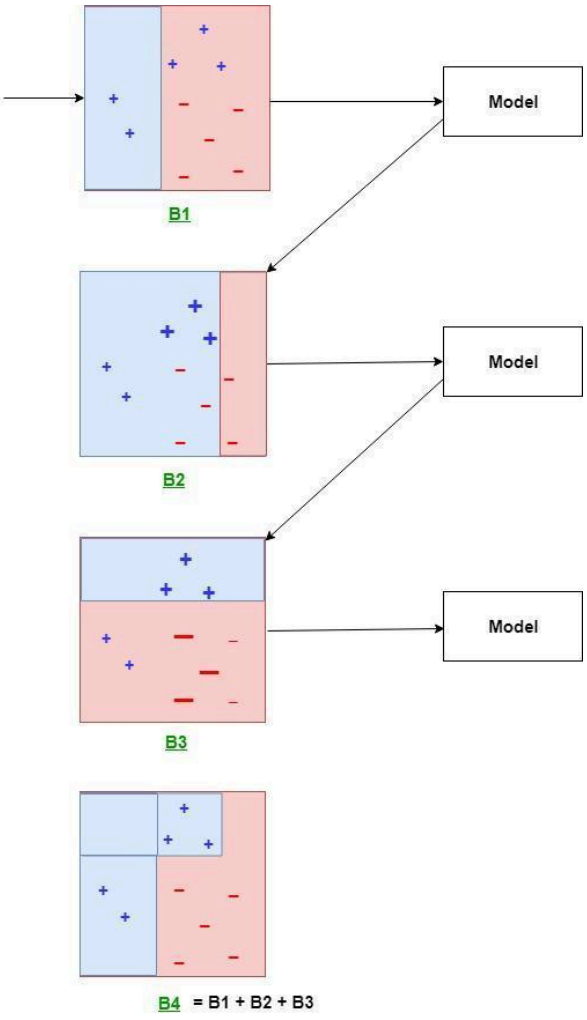
Есть разные типы алгоритмов классификации. Они были разработаны, чтобы обеспечить лучшие результаты для задач классификации за счет использования бэггинга и бустинга.

Бустинг – это метод построения сильного классификатора из числа слабых классификаторов путем построения модели с использованием слабых моделей. Модель строится на обучающих данных, затем строится вторая модель, которая пытается исправить ошибки первой модели. Эта процедура продолжается, и модели добавляются до тех пор, пока не будет правильно предсказан полный набор обучающих данных или пока не будет добавлено максимальное количество моделей.

AdaBoost (Adaptive Boosting) — популярный метод бустинга, который объединяет несколько «слабых классификаторов» в один «сильный классификатор». Его сформулировали Йоав Фройнд и Роберт Шапир. Они получили премию Гёделя 2003 года за свою работу. AdaBoost был первым действительно успешным алгоритмом бустинга, разработанным для двоичной классификации.

Алгоритм:

1. Инициализируйте набор данных и присвойте равный вес каждой точке данных.
2. Предоставьте это в качестве входных данных для модели и определите ошибочно классифицированные точки данных.
3. Увеличьте вес ошибочно классифицированных точек данных.
4. Если (получены требуемые результаты) → перейти к шагу 5
5. Иначе → перейти к шагу 2
6. Конец



Сейчас самые популярные и быстрые модели классификации работают на основе алгоритмов бустинга, с каждым годом эти алгоритмы становятся быстрее и точнее. Основная задача, которой сейчас занимается множество крупных компаний, — это ускорения бустеров для моделей классификации.

Бэггинг — это алгоритм метаоценки, который сопоставляет каждый базовый классификатор со случайными подмножествами исходного набора данных, а затем объединяет их индивидуальные прогнозы (путем голосования или усреднения) для формирования окончательного прогноза.

Такая метаоценка обычно используется как способ уменьшить дисперсию оценки черного ящика (например, дерева решений) путем введения рандомизации в процедуру ее построения и последующего создания из нее ансамбля.

Каждый базовый классификатор обучается параллельно с обучающим набором, который генерируется путем случайного рисования с заменой N примеров (или данных) из исходного набора обучающих данных, где N — размер исходного обучающего набора. Обучающая выборка для каждого из базовых классификаторов независима друг от друга. Многие из исходных данных могут повторяться в результирующем обучающем наборе, а другие могут быть опущены.

Бэггинг уменьшает переобучение (дисперсию) за счет усреднения или голосования, однако это приводит к увеличению систематической ошибки, которая компенсируется уменьшением дисперсии.

Сравнение классификации и регрессии:

Классификация	**Регрессия**
В этой постановке задачи целевые переменные дискретны	В этой постановке задачи целевые переменные непрерывны
Такие проблемы, как классификация спама по электронной почте и прогнозирование заболеваний, решаются с помощью алгоритмов классификации	Такие задачи, как прогноз цен на жилье и прогноз осадков, решаются с помощью алгоритмов регрессии
Мы пытаемся найти лучшую возможную границу решения, которая может разделить два класса с максимально возможным разделением	Мы пытаемся найти наиболее подходящую линию, которая может представлять общую тенденцию данных

Показатели оценки, такие как Precision, Recall и F1-Score, используются для оценки производительности алгоритмов классификации	Показатели оценки, такие как среднеквадратическая ошибка, R2-Score и MAPE, используются для оценки производительности алгоритмов регрессии
Мы сталкиваемся с такими проблемами, как бинарная классификация или проблемы многоклассовой классификации	Мы сталкиваемся с такими проблемами, как модели линейной регрессии, а также с нелинейными моделями
Входные данные — это независимые переменные и категориальная зависимая переменная	Входные данные — это независимые переменные и непрерывная зависимая переменная
Вывод — категориальные метки	Выходные данные являются непрерывными числовыми значениями.
Цель в том, чтобы предсказать метки категорий/классов	Цель в том, чтобы прогнозировать непрерывные числовые значения
Примеры использования: обнаружение спама, распознавание изображений, анализ настроений	Примеры использования: прогнозирование цен на акции, прогнозирование цен на жилье, прогнозирование спроса

Начало работы с классификацией

В машинном обучении и статистике классификация — это проблема определения того, к какому из набора категорий (подгрупп) принадлежит новое наблюдение на основе обучающего набора данных, содержащего наблюдения, и принадлежность к которым известна.

Классификация — это задача машинного обучения, которая включает присвоение метки класса заданным входным данным на основе набора обучающих данных. Цель классификации — построить модель, которая может точно предсказать метку класса для новых невидимых данных.

Несколько шагов, чтобы начать работу с классификацией:

1. **Понимание проблемы.** Прежде чем приступить к классификации, важно понять проблему, которую вы пытаетесь решить. Какие метки классов вы пытаетесь предсказать? Какова связь между входными данными и метками классов?

2. **Подготовка данных.** Как только вы хорошо разберетесь в проблеме, следующим шагом будет подготовка данных: их сбор и предварительная обработка, а также их разделение на обучающие, проверочные и тестовые наборы.

3. **Выбор модели.** Существует множество моделей, которые можно использовать для классификации, включая деревья решений, случайные леса, k-ближайших соседей и методы опорных векторов. Важно выбрать модель, подходящую для вашей задачи, принимая во внимание размер и сложность ваших данных, а также доступные вычислительные ресурсы.

4. **Обучение модели.** Когда вы выбрали модель, следующим шагом будет ее обучение на данных: настройка параметров модели для минимизации ошибки между прогнозируемыми метками классов и фактическими метками классов для обучающих данных.

5. **Оценка модели.** После обучения модели важно оценить ее производительность на проверочном наборе. Это даст вам представление о том, насколько хорошо модель может работать с новыми невидимыми данными.

6. **Тонкая настройка модели.** Если производительность модели неудовлетворительна, ее можно настроить, изменив параметры или попробовав другую модель.

7. **Развертывание модели.** Наконец, когда вы удовлетворены производительностью модели, вы можете развернуть ее, чтобы делать прогнозы на основе новых данных.

Это основные шаги для начала работы с классификацией. По мере накопления опыта вы, возможно, захотите изучить более продвинутые методы, такие как ансамблевые методы, глубокое обучение и трансферное обучение.

Типы классификации

Классификация бывает двух видов:

1. **Двоичная классификация** — когда нам нужно разделить данные на 2 разных класса. Например, на основании данных о здоровье человека мы должны определить, есть ли у человека определенное заболевание или нет.
2. **Мультиклассовая классификация** — количество классов больше 2. Например, на основе данных о разных видах цветов мы должны определить, к какому виду относится наше наблюдение.

Как работает классификация?

Предположим, нам нужно предсказать, есть ли у пациента определенное заболевание или нет, на основе 3 переменных, называемых признаками.
Это значит, что есть два возможных исхода:

1. У пациента есть указанное заболевание. По сути, результат с пометкой «Да» или «Верно».
2. У пациента нет указанного заболевания. Результат с пометкой «Нет» или «Ложь».

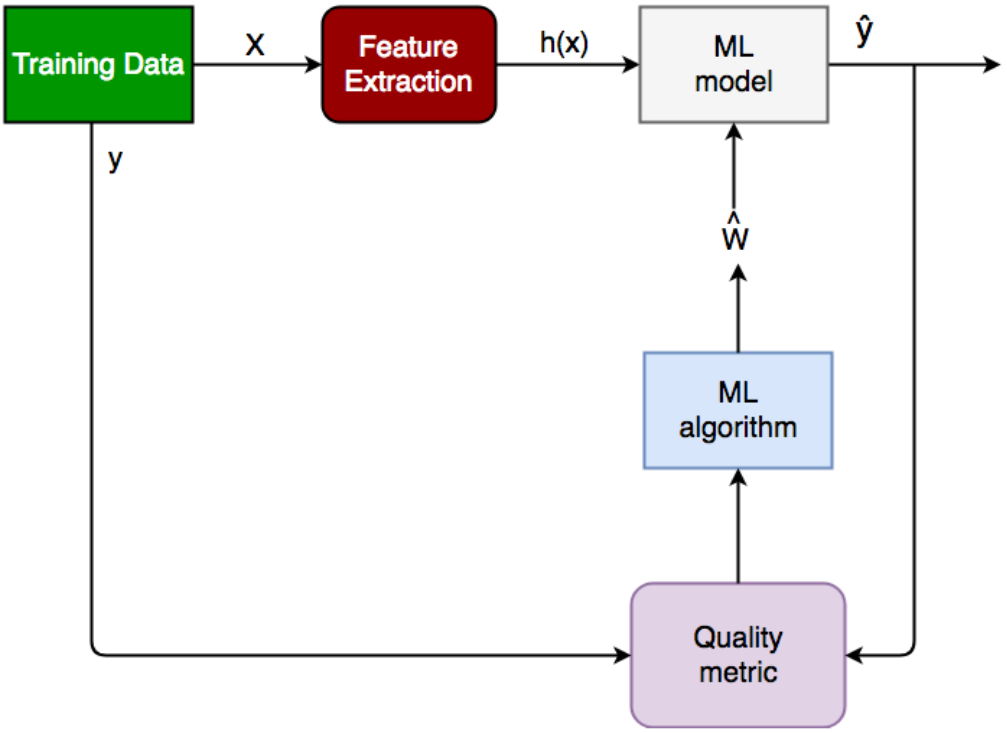
Это проблема бинарной классификации.

У нас есть набор наблюдений (набор обучающих данных), который содержит выборочные данные с фактическими результатами классификации. Мы обучаем модель под названием «Классификатор» на этом наборе данных и используем эту модель, чтобы предсказать, будет ли заболевание у определенного пациента или нет.

Результат, таким образом, теперь зависит от нескольких факторов:

1. Насколько хорошо эти функции могут «сопоставляться» с результатом.
2. Качество нашего набора данных. Под качеством я подразумеваю статистические и математические качества.
3. Насколько хорошо наш классификатор обобщает эту связь между функциями и результатом.
4. Значения x1 и x2.

Ниже приведена обобщенная блок-схема задачи классификации.



Блок-схема обобщенной классификации:

1. X: предварительно классифицированные данные в виде матрицы N*M. N — нет наблюдений, M — количество признаков.
2. y: вектор Nd, соответствующий предсказанным классам для каждого из N наблюдений.
3. Извлечение функций: извлечение ценной информации из ввода X с использованием серии преобразований.
4. Модель машинного обучения: «Классификатор», который мы будем обучать.
5. y': метки, предсказанные классификатором.
6. Метрика качества: метрика, используемая для измерения производительности модели.
7. Алгоритм ML: алгоритм, который используется для обновления весов w', который обновляет модель и итеративно «обучается».

Типы классификаторов (алгоритмы)

Существуют различные типы классификаторов. Некоторые из них :

- Линейные классификаторы: логистическая регрессия
- Древовидные классификаторы: классификатор дерева решений

- Опорные векторные машины
- Искусственные нейронные сети
- Байесовская регрессия
- Гауссовские наивные байесовские классификаторы
- Классификатор стохастического градиентного спуска (SGD)
- Методы ансамбля: случайные леса, AdaBoost, классификатор пакетов, классификатор голосования, классификатор ExtraTrees.

Практическое применение классификации

1. Беспилотный автомобиль использует методы классификации с поддержкой глубокого обучения, которые позволяют обнаруживать и классифицировать препятствия.
2. Фильтрация спама по электронной почте является одним из наиболее распространенных и общепризнанных способов использования методов классификации.
3. Обнаружение проблем со здоровьем, распознавание лиц, распознавание речи, обнаружение объектов и анализ настроений используют классификацию в своей основе.

Давайте на практике познакомимся с тем, как работает классификация. Мы собираемся изучить различные классификаторы и увидеть простое аналитическое сравнение их производительности на известном стандартном наборе данных, наборе данных Iris.

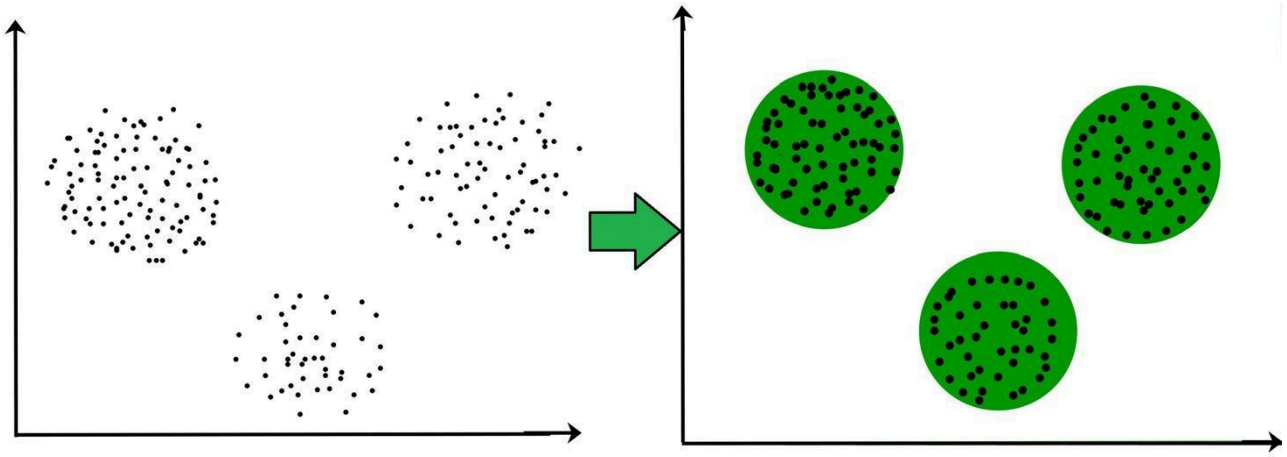
Кластеризация

Кластеризация — это основной тип метода обучения без учителя: мы получаем ссылки из наборов данных, состоящих из входных данных без помеченных ответов. Как правило, он используется как процесс для поиска значимой структуры, объяснения лежащих в основе процессов, генеративных функций и группировок, присущих набору примеров.

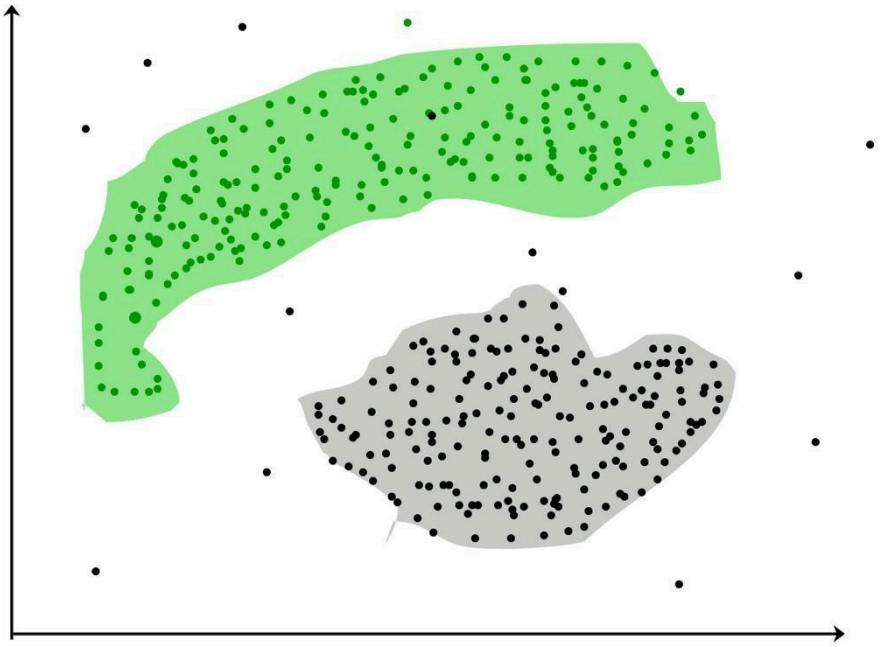
Кластеризация — это задача разделения совокупности или точек данных на несколько групп таким образом, чтобы точки в одних группах были похожи и отличались от

точек в других группах. Это совокупности объектов на основе сходства и несходства между ними.

Например, точки данных на графике ниже, сгруппированные вместе, можно классифицировать в одну группу. Мы можем различить кластеры, и мы можем определить, что на изображении ниже есть 3 кластера.



Кластеры не обязательно должны быть сферическими. Например :



DBSCAN: пространственная кластеризация приложений с шумом на основе плотности

Эти точки данных группируются с использованием базовой концепции, согласно которой точка данных находится в пределах заданного ограничения от центра кластера. Для расчета выбросов используются различные дистанционные методы.

Кластеризация очень важна, поскольку определяет внутреннюю группировку среди присутствующих неразмеченных данных. Критериев хорошей кластеризации нет, но есть метрики, позволяющие оценить, насколько четкие границы кластеров мы получили в результате кластеризации: например, silhouette score. Это зависит от пользователя и от того, какие критерии он может использовать для удовлетворения своих потребностей.

Например, нас может интересовать поиск представителей для однородных групп (редукция данных), поиск «естественных кластеров» и описание их неизвестных свойств («естественные» типы данных), поиск полезных и подходящих группировок («полезные» классы данных) или поиск необычных объектов данных (обнаружение выбросов). Этот алгоритм должен делать некоторые предположения, которые составляют сходство точек, и каждое предположение создает разные и одинаково достоверные кластеры.

Методы кластеризации:

- **Методы, основанные на плотности**, рассматривают скопления как плотную область, имеющую некоторые сходства и отличия от более низкой плотной области пространства. Эти методы обладают хорошей точностью и возможностью объединения двух кластеров.

  Примеры — DBSCAN (пространственная кластеризация приложений с шумом на основе плотности), OPTICS (точки упорядочения для определения структуры кластеризации) и так далее.

- **Методы, основанные на иерархии**, — кластеры, сформированные в этом методе, образуют структуру древовидного типа на основе иерархии. Новые кластеры формируются с использованием ранее сформированного. Делятся на две категории:

  - **Агломеративный** — подход «снизу вверх»
  - **Разделительный** — подход «сверху вниз»

  Примеры — CURE (кластеризация с использованием представителей), BIRCH (сбалансированная итеративная редуцирующая кластеризация и использование иерархий) и так далее.

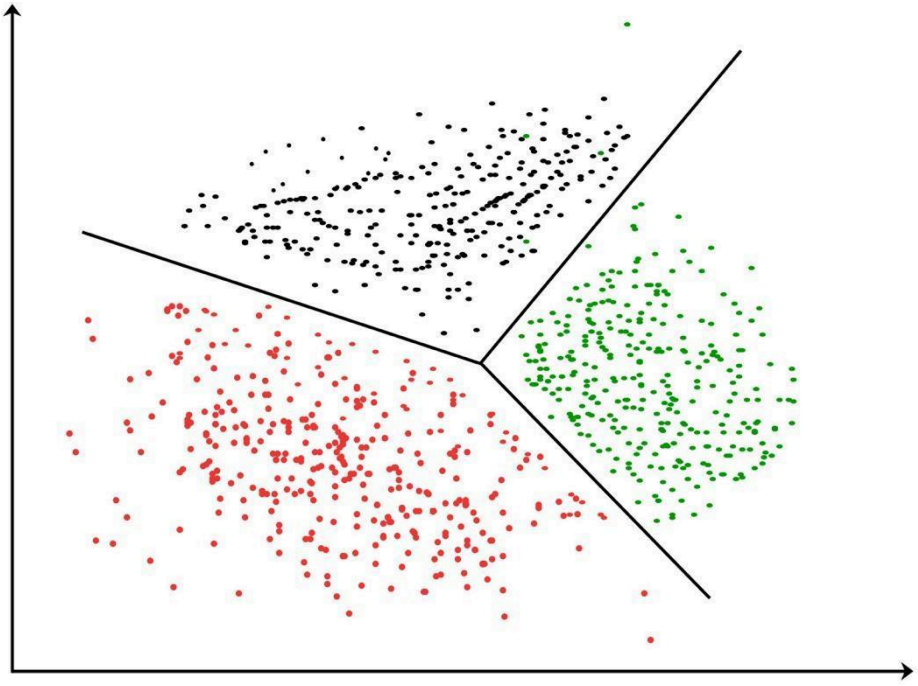
- **Методы разбиения** разбивают объекты на k кластеров, и каждый раздел образует один кластер. Этот метод используется для оптимизации функции подобия объективного критерия, например, когда расстояние является основным параметром.

Примеры — K-means, CLARANS (кластеризация больших приложений на основе случайного поиска) и так далее.

- **Методы на основе сетки** — пространство данных состоит из конечного числа ячеек, которые образуют структуру, подобную сетке. Все операции кластеризации в этих сетках выполняются быстро и не зависят от количества объектов данных.

  Примеры — STING (статистическая информационная сетка), волновой кластер, CLIQUE (кластеризация в поисках) и так далее.

Алгоритмы кластеризации — алгоритм кластеризации K-средних. Это простейший алгоритм обучения без учителя, который решает проблему кластеризации. Алгоритм K-средних разбивает n наблюдений на k кластеров, где каждое наблюдение принадлежит кластеру, а ближайшее среднее значение служит прототипом кластера.



Применение кластеризации в разных областях

- **Маркетинг.** Для характеристики и выявления сегментов клиентов в маркетинговых целях.
- **Биология.** Для классификации различных видов растений и животных.
- **Библиотеки.** Для группировки различных книг на основе тем и информации.
- **Страхование.** Для признания клиентов, их политики и выявления мошенничества.

- **Городское планирование.** Для создания групп домов и изучения их стоимости на основе их географического положения и других факторов.
- **Исследования землетрясений.** Для определения опасных зон (их можно выявить, изучая районы, пострадавшие от землетрясения).
- **Обработка изображений.** Для группировки похожих изображений, классификации изображений на основе содержимого и выявления закономерностей в данных изображения.
- **Генетика.** Для группировки генов, имеющих схожие паттерны экспрессии, и выявления генных сетей, которые работают вместе в биологических процессах.
- **Финансы.** Для определения сегментов рынка на основе поведения клиентов, выявления закономерностей в данных фондового рынка и анализа рисков в инвестиционных портфелях.
- **Обслуживание клиентов.** Для группировки запросов и жалоб клиентов по категориям, выявления общих проблем и разработки целевых решений.
- **Производство.** Для группировки похожих продуктов, оптимизации производственных процессов и выявления дефектов в производственных процессах.

Метод К-средних

Цель кластеризации в том, чтобы разделить совокупность или набор точек данных на несколько групп, чтобы точки данных в каждой группе были сопоставимы друг с другом и отличались от точек данных в других группах. По сути, это группировка объектов, основанная на том, насколько они похожи и отличаются друг от друга.

Нам дан набор элементов данных с определенными функциями и значениями этих функций (например, вектор). Задача состоит в том, чтобы разделить эти предметы на группы. Для этого мы будем использовать алгоритм K-средних; алгоритм обучения без учителя. «K» в названии алгоритма представляет собой количество групп/кластеров, по которым мы хотим классифицировать наши элементы.

Это поможет, если вы будете думать об элементах как о точках в n-мерном пространстве. Алгоритм разделит элементы на k групп или кластеров сходства. Чтобы вычислить это сходство, мы будем использовать евклидово расстояние в качестве измерения.
Алгоритм работы:

1. Мы случайным образом инициализируем k точек, называемых средними значениями или центроидами кластера.

2. Мы классифицируем каждый элемент по его ближайшему среднему значению и обновляем координаты среднего значения, которые являются средними значениями элементов, классифицированных в этом кластере на данный момент.

3. Мы повторяем процесс для заданного количества итераций, и в конце у нас есть наши кластеры.

Упомянутые выше «баллы» называются средними, потому что они представляют собой средние значения элементов, классифицированных в них. Для инициализации этих средств у нас есть много вариантов. Интуитивно понятный метод заключается в инициализации средних значений для случайных элементов в наборе данных. Другой метод заключается в инициализации средних значений случайными значениями между границами набора данных (если для функции x элементы имеют значения в [0,3], мы инициализируем средние значения для x в [0,3]).
Приведенный выше алгоритм в псевдокоде выглядит следующим образом:

Инициализировать k-средних случайными значениями

--> Для заданного количества итераций:

--> Итерация по элементам:

--> Найдите среднее значение, ближайшее к элементу, вычислив

евклидово расстояние элемента с каждым из средних

--> Присвоить элементу значение

--> Обновить среднее значение, сдвинув его к среднему значению элементов в этом кластере.

Заключение

Сегодня мы погрузились в базовые темы, связанные с машинным обучением — как с учителем, так и без. Поговорили о том, что такое кластеризация и классификация. Рассмотрели на примере построение модели классификации и кластеризации на одном и том же наборе данных и теперь попробуем построить свои модели на последнем семинаре.

До новых встреч!