GeekBrains

Классическое обучение. Основные понятия

Машинное обучение

Оглавление

Введение

Сегодня мы начинаем знакомство с невероятным направлением в современном мире IT на стыке математики и программирования — машинным обучением. Звучит довольно сложно, но не стоит бояться: каждый из вас встречается с машинным обучением ежедневно. Социальные сети предлагают новости на основе того, что вы чаще всего смотрите, маркетплейсы советуют вам товары, а голосовой помощник включает ваши любимые песни по запросу. Все это — машинное обучение, одно из направлений искусственного интеллекта.

Меня зовут Евгений Абумов, я буду вашим преподавателем на протяжении этого курса. С машинным обучением я познакомился 10 лет назад. Тогда еще не было таких нейросетевых алгоритмов, которые позволяли бы генерировать текст или картинки по текстовому описанию. Это было время первых голосовых помощников, и именно эта тема заинтересовала меня. Было очень интересно узнать, как же речь превращается в конкретный запрос, который затем обрабатывается машиной. Затем был продолжительный путь изучения данной темы, путь нелегкий.

Если вспоминать мое обучение — больше всего трудностей и опасений было с математикой. Возможно, потому что местами не хватало знаний или было убеждение, оставшееся после школы и университета, что все что связано с математикой — трудно постижимо человеку, который никогда не интересовался

данным предметом. Возможно, и вы ощущаете похожие сомнения. Но не стоит переживать! На нашем курсе, конечно, будут понятия и формулы из математики, но они всегда будут подкреплены простыми и понятными примерами. Кстати, во время изучения и работы в машинном обучении я понял: математика это очень интересно!

Теперь давайте поговорим про то, что вас ждет на этом курсе. В рамках задачи обучения с учителем мы рассмотрим с вами линейные модели машинного обучения и задачи, которые они решают — задачи регрессии и классификации. Вместе с этим, мы будем говорить о том, как улучшать наши модели, каким образом вообще их обучают, и конечно, про плюсы и минусы использования разных моделей.

Далее мы изучим деревья решения и создадим из простых алгоритмов сложный — это называется ансамбль. Значительную часть курса мы посвятим оптимизации модели и работе с признаками. Каждая лекция будет сопровождаться практикой и домашней работой, а затем на семинаре вы сможете применить знания на практике и решить одну из прикладных задач. Задачи будут интересные!

На данном, первом занятии, мы изучим базовые понятия, дадим определения основным терминам в машинном обучении, а на практике научимся считать точность модели машинного обучения с помощью функции потерь.

Для лучшего усвоения и понимания материала советую вам вспомнить линейную алгебру, основы статистики и математического анализа и теорию вероятности. Так же будет не лишним уметь работать с массивами, классами и объектами на языке программирования Python.

Ну что же, давайте начинать!

Термины, используемые в лекции

Искусственный интеллект (artificial intelligence) – автоматизация интеллектуальных задач, обычно выполняемых людьми

Машинное обучение (machine learning) – обучение, при котором люди вводят данные и ответы, соответствующие этим данным, а на выходе получают правила. Эти правила можно применить к новым данным для получения оригинальных ответов.

Глубокое обучение (deep learning) – подход к представлению данных, делающий упор на изучение последовательных слоев. Другими словами, глубокое обучение основано не на конкретных алгоритмах, а на идее отображения входа в желаемый выход.

Объекты – это то, что определяется задачей, и для чего мы создаем модель машинного обучения. Например, в задаче прогнозирования цены дома, объекты — это дома.

Ответы – это то, что является решением для объектов в конкретной задаче. Например, в задаче определения стоимости дома ответом является цена дома, в задаче определения спама ответом является класс сообщения — спам это или нет.

Целевая переменная – это то, что мы прогнозируем. Например, в задаче прогнозирования цен на дома, целевая переменная — это цена дома.

Выборка – множество объектов и множество ответов к ним.

Признаки (фичи) – это то, чем характеризуются объекты. Например, для автомобиля это может быть цвет, количество лошадиных сил, марка, тип кузова. Признаки сильно зависят от того, что мы хотим получить от модели машинного обучения.

Бинарные признаки – это признаки, у которых всего два значения, например 0 или 1, True или False. Это все характеристики объекта, на которые можно однозначно ответить — да или нет.

Категориальные признаки – это признаки, которые принимают значение из некого конечного множества. Например, цвет машины, или тип бытовой техники.

Числовые признаки – это признаки, которые имеют числовое выражение. Это может быть цена, рейтинг товара, размер квартиры и так далее.

Обучение с учителем (supervised learning) – у нас есть историческая выборка о каком то явлении, или истории того, как раньше принимал решения человек. Эта история и есть наш учитель.

Обучение без учителя (unsupervised learning) – задача, в которой нам самим нужно найти признаки, разделяющие группу объектов.

Регрессия – задача машинного обучения, целью которой является поиск числа.

Классификация – задача машинного обучения, в которой нужно определить класс объект.

Функционал модели – это некоторая функция, которая на вход принимает модель машинного обучения и обучающую выборку. Затем измеряет ошибку на каждом объекте, и усредняет ее.

Модель машинного обучения – функция зависимости данных от ответов, которая обучается на известных примерах для подбора весов.

Немного истории

Когда мы говорим про машинное обучение, складывается впечатление, что оно появилось недавно, но это не так.

Вообще понятие ИИ возникло не десять и даже не 20 лет назад. Еще в Древней Греции, например, был бог Гефест, который создал механического бронзового титана и наделил его душой. В китайском писании третьего века до нашей эры есть история изобретателя, который представил королю механического человека, способного ходить и петь. Делаем вывод: идея искусственного интеллекта уже долгое время сидела в головах людей.

Термин “Искусственный интеллект” начал употребляться после смерти Алана Тьюринга. Этот великий человек, сделавший огромный вклад в информатику, алгоритмизацию и искусственный интеллект. Еще в 1947 году он публично говорил о “машине, которая способна учиться на собственном опыте”. Его метод для

определения способности машины мыслить как человек, известный как тест Тьюринга, до сих пор используется разработчиками ИИ.

Когда речь идет о современных итерациях “искусственного интеллекта”, мы используем слова, придуманные в 1956 году Джоном Маккарти, 28-летним профессором Дартмутского колледжа. Термин возник на конференции по машинному обучению, организованной Маккарти и другими профессорами из Дартмута. Они планировали пригласить всего несколько участников, но вместо этого на конференцию пришли десятки исследователей из разных научных областей. Это показало, что к исследованиям в области ИИ не только есть интерес, но что у них есть и реальный потенциал.

Не только математики в те времена интересовались искусственным интеллектом. Фрэнк Розенблатт преподавал научную психологию в авиационной лаборатории Корнелла и первым использовал естественные науки, чтобы вдохновить людей на исследования в области искусственного интеллекта. В 1958 году он изобрел перцептрон — электронное устройство, которое имитирует нейронные сети в человеческом мозге и активирует систему распознавания образов. Розенблатт впервые смоделировал персептрон на ранней версии компьютера ЭВМ, а позже его усовершенствовали в американском ВМС. Газета The New York Times назвала его технологию “зародышем электронного компьютера“, который должен был “уметь ходить, говорить, видеть, писать, воспроизводить себя и осознавать свое существование”.

За последние два десятилетия появилось несколько резонансных примеров превосходства ИИ над простыми смертными. В 1997 году суперкомпьютер Deep Blue, созданный компанией IBM для игры в шахматы, победил мирового чемпиона по шахматам Гарри Каспарова, став первой машиной, обыгравшей действующего чемпиона мира.

Еще одно ключевое событие произошло в 2011 году, когда компьютерная система под названием Watson выиграла 1 миллион долларов в американском телевизионном шоу “Своя игра”. А в 2015 году технология AlphaGo от Google разгромила в древней китайской настольной игре Го лучшего европейского игрока Фан Хуи. Однако не всегда все шло так гладко. Взять хотя бы случай в 2016 году с гуманоидным роботом Софией, которая во время демонстрации на конференции South by Southwest заявила, что “уничтожит человечество”. Так робот ответила на, по всей видимости, шутливый вопрос своего создателя Дэвида Хансона.

В итоге сейчас мы живем в мире chatGPT, множества рекомендательных систем, нейросетей. Но как видите, большинство идей было придумано довольно давно. Способна ли повторить когнитивные способности человеческого мозга модель

машинного обучения? То есть способно ли человечество создать искусственный интеллект, неотличимый от естественного интеллекта? Я думаю, когда-нибудь мы сможем говорить об этом!

Даем определения

Так что же такое машинное обучение? И как оно относится к искусственному интеллекту? Давайте дадим определения.

Итак, **искусственный интеллект (artificial intelligence)** — автоматизация интеллектуальных задач, обычно выполняемых людьми.

Машинное обучение (machine learning) — обучение, при котором люди вводят данные и ответы, соответствующие этим данным, а на выходе получают правила. Эти правила можно применить к новым данным для получения оригинальных ответов. В нашем курсе мы будем работать именно с классическим машинным обучением, так как это основа, на базе которой строится **глубокое обучение,** или как его еще называют **глубинное обучение (deep learning)** — подход к представлению данных, делающий упор на изучение последовательных слоев.

Другими словами, глубокое обучение основано не на конкретных алгоритмах, а на идее отображения входа в желаемый выход. Современные технологии, которые привлекают внимание всего мира, основаны именно на глубоком обучение. Вы еще могли слышать такое понятие как нейросети — это и есть глубокое обучение.

Другими словами машинное обучение — это алгоритмы, которые извлекают зависимости из данных. В машинном обучении не стоит цель решить задачу прямым способом. Нам нужно выявить правила, по которым решаются все задачи такого типа.

Давайте начнем с примера, на котором мы разберем основные определения, чтобы разговаривать на одном языке.

Итак, давайте представим, что мы с вами являемся владельцами сервиса для туристов, на котором можно забронировать жилье посуточно. Одни люди, заходя на наш сайт, могут снять жилье, а другие — назовем их арендодатели — могут предложить свое жилье на нашей площадке.

Цена проживания может зависеть от различных факторов, плюс арендодатели могут ее как завышать, так и занижать. Наша задача — построить справедливую рекомендательную систему, которая будет оценивать стоимость проживания в конкретном объекте размещения. Как же машинное обучение может помочь нам в этой задаче? Давайте разбираться.

Для начала, давайте определим, что является объектом в этой задаче. Объекты — это то, что определяется задачей, и для чего мы создаем модель машинного обучения. Например, в задаче обнаружения спама, объектами являются электронные письма, а в задаче прогнозирования болезни — пациенты. В нашей задаче объекты это дома, квартиры, комнаты, то есть места размещения. Объекты обозначаются буквой x(икс).

Мы выяснили, что является объектами, теперь нужно понять, что мы прогнозируем. В этой задаче нас интересует цена размещения в объекте. Это называется целевая переменная. Это формальное определение, часто целевую переменную называют еще target (таргет). Давайте в нашем примере считать целевой переменной конкретную цену проживания в объекте недвижимости за одну ночь.

В нашей задаче мы будем говорить не про один объект, а про некое множество объектов, для которых мы будем делать предсказание. И точно так же у нас будет множество ответов. Про ответы мы поговорим чуть позже.

Мы ввели определение объекта, у нас это объект недвижимости, то есть дом, квартира, комната. Но пока не очень понятно, чем характеризуется этот объект. Конечно, можно предположить, что, допустим, если это квартира в центре города, то посуточное проживание будет дорогим. А если она еще и находится недалеко от

метро, то проживание будет еще дороже. Но, к сожалению, компьютер такое описание не поймет. Компьютер умеет оперировать только цифрами. Поэтому в машинном обучении каждый объект описывают какими-то числами — характеристиками, на основе которых мы будем делать прогноз. Введем еще одно понятие.

Признаки, или как их еще называют факторы, feachers, фичи — это характеристики объекта. В нашем случае признаки будут описывать объект размещения.

Ну что же, давайте немного разомнемся! Подумайте, какие характеристики могут описывать наши объекты — дома, комнаты, квартиры, и какие из придуманных характеристик на ваш взгляд больше всего буду влиять на стоимость проживания.

Я уверен, вы придумали много различных признаков. Теперь давайте классифицируем все признаки.

- **Бинарные** — это признаки, у которых всего два значения, например 0 или 1, True или False. Это все характеристики объекта, на которые можно однозначно ответить — да или нет. Атрибуты зависят от задачи. Например, оплата по карте или наличными, метро рядом или нет и так далее.
- **Количественные**, или числовые — это признаки, которые имеют числовое выражение. Это может быть цена, рейтинг товара, размер квартиры и так далее. Числовое выражение признака принято брать конечным. Здесь надо быть осторожным — не всегда число, это количественный признак. Например, рейтинг товара можно рассматривать как положительный и отрицательный.
- **Категориальные** — это признаки, которые принимают значение из некого конечного множества. Например, цвет машины, или тип бытовой техники. Чуть позже мы с вами будем говорить про линейные модели, и про то, что они не могут работать с категориальными признаками, поэтому нам придется переводить их в другой тип.

Есть и более сложные типы признаков, а точнее структуры, например картинки. В практике машинного обучения есть способы перевода картинок в числовое описание, но на данном курсе мы не будем это рассматривать, так как это больше касается глубокого обучения.

Далее, если объект мы назвали x, то описание такого объекта с помощью признаков будет выглядеть так:

x = (x1, x2, ... xd), где d - это количество признаков.

Такое описание объекта называют признаковым описанием.

На этом курсе мы всегда будем считать, что нам даны данные. Что из себя представляют данные? Это разные объекты в зависимости от задачи — например дома, товары, акции и так далее, и ответы на них. То есть сколько, например, стоит дом в этом районе, или сколько стоит товар. В нашей текущей задаче мы тоже считаем, у нас есть набор данных — объекты-ответы — который поможет нам с созданием модели машинного обучения. Это называется обучающая выборка или датасет (dataset).

Обратите внимание, что наши данные представлены в таблице. Табличные данные — один из самых удобных форматов отображения данных. Лучше всего, если все

признаки являются численными — тогда с таблицей можно работать, как с объектом линейной алгебры — матрицей объекты-признаки.

Создание признакового описание очень для дальнейшей работы. Но нам могут встретиться и некоторые проблемы:

- **Пропуски** — или пропущенные значения. Например, нам неизвестна стоимость дома в какой-то строке, или нет численного описания площади объекта размещения. Конечно, пропуски можно удалять из выборки, но мы можем пропустить слишком много информации. Кстати наличие пропуска само по себе может нести информацию: например, это может говорить о систематической проблеме в сборе данных.
- **Выбросы** — объекты, которые резко отличаются от большинства остальных. Например, в задаче определения стоимости ноутбука, ноутбук будет стоить 100 рублей. Иногда выбросы — это действительно аномалии, ошибка при сборе данных, но бывают случаи когда выбросы правдивы, просто их нужно обрабатывать отдельно.
- **Ошибка разметки** — такие ошибки в основном допускают люди при разметке датасета.

В реальных задачах данные приходится искать. Это могут быть таблицы exel, базы данных или иной источник. Так же, данные часто бывают не размеченными, то есть нет пар объект-ответ. Такое часто встречается, когда мы работаем с картинками. Поэтому такие картинки(или вообще любые неразмеченные данные отдают на краудсорсинг — это ручная разметка данных людьми. Вообще все что связано с данными, это большая часть работы в машинном обучении. От данных зависит какая модель получится в итоге.

> 💡 Чуть ранее, мы с вами придумывали признаки. Это одна из самых творческих задач в машинном обучении, и даже есть целое направление — фичер инжиниринг. Это направление занимается поиском, созданием, придумыванием признаков.

Теперь поговорим про модель машинного обучения. На данный момент у нас есть: объекты, ответы, и стоит задача — на основании уже известных объектов создать рекомендательную модель, то есть модель, которая будет предсказывать цену на

недвижимость. А что такое модель? Представьте, что вы сами зашли на наш сайт и хотите разместить объект недвижимости. Как вы определите цену своего объекта? Очевидно, что вы откроете карту в районе нахождения вашей недвижимости, и начнете смотреть похожие объекты, то есть квартиры. А затем, на основе ваших наблюдений, сделаете вывод, какой должна быть цена на собственную квартиру.

В этом и заключается машинное обучение. Модель повторяет ваши действия, она точно так же смотрит на объекты недвижимости и их цены, то есть учится, а затем может сделать предположение о цене, но уже по иному объекту. Если говорить более формально, модель — это функция, отображающая объекты в предсказания. Например, в нашей задаче мы отображаем недвижимость в цену.

Теперь нужно выбрать, к какому типу задач относится наша. В машинном обучении есть несколько типов задач:

Обучение с учителем — у нас есть историческая выборка о каком то явлении, или истории того, как раньше принимал решения человек. Эта история — наш учитель. Например, мы хотим предсказать цену , и у нас есть история, то есть данные за какой-то прошлый период, о том, как цена зависела от марки, процессора, размера экрана и так далее. Наша задача — построить модель, которая будет предсказывать цену так же, как раньше делал это человек, то есть у нас есть учитель.

Рассмотрим другой пример. Допустим мы хотим дать скидку той категории людей, которые почти готовы купить ноутбук. Но как определить такую группу? Кто должен войти в эту группу? По каким критериям отбирать таких людей? Этой информации у нас нет, поэтому в таком виде учителя у нас тоже нет. Мы должны самостоятельно найти такую категорию людей и самостоятельно определить их критерии. Такая постановка вопроса называется задачей машинного обучения без учителя.

Теперь поговорим про то, какие задачи решает обучение с учителем.

- **Регрессия** — примерами задач регрессии может быть предсказание стоимости дома, спрос на конкретный товар, температура воздуха. По сути, та задача, которую мы рассматривали в течении занятия и есть задача линейной регрессии.
- **Бинарная классификация** — мы можем предсказывать например, вернет ли клиент кредит, как будет происходить оплата — наличными или картой. В общем это задача когда нужно определить однозначный класс объекта из двух.

- **Многоклассовая классификация** — задача, при которой у нас больше чем два класса. Например, определить жанр музыки по словам и мелодии.
- **Многоклассовая классификация с пересекающимися классами** — более сложный вариант классификации, когда классы могут повторяться.
- **Ранжирование** — задача некой сортировки, когда сравнивают объекты друг с другом.
- **Понижение размерности** — задача, при которой уменьшают количество признаков для модели, делая при этом признаки мощнее.

На самом деле множество ответов и определяет тип задачи. А это в свою очередь определяет то, как мы будем измерять ошибку, какую модель мы будем строить.

Регрессия

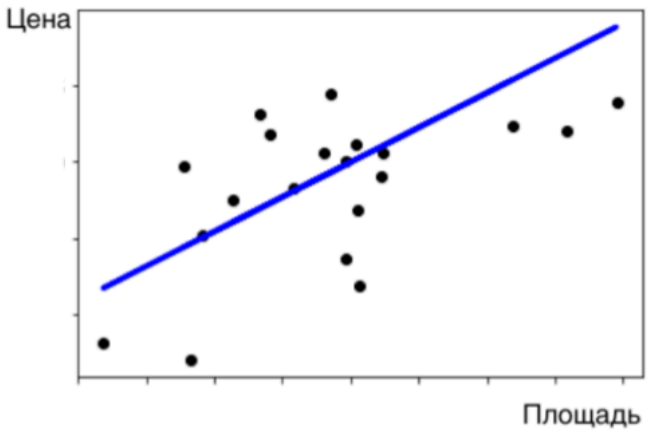
Итак, мы выяснили, что модель — это математическая функция, которая принимает на вход признаковое описание объектов. Рассмотрим подробнее задачу регрессии. Во-первых, определим выход модели — если модель возвращает число от минус бесконечность до плюс бесконечность — это задача регрессии.

В школе вы наверняка встречались с регрессией, когда смотрели на график, и пытались оценить значение в новой точке. Тогда ваш график был описан некой функцией. В машинном обучении функция неизвестна, нам нужно ее найти.

Давайте посмотрим на график, и предположим, что цена проживания за одну ночь из нашего примера зависит всего лишь от одного параметра — от площади объекта размещения.

Линейная регрессия

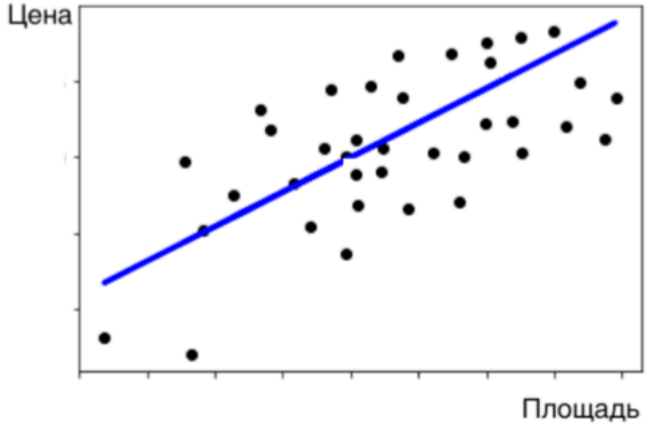
Пример



Мы можем просто соединить все точки, но это будет неверно. Цена наверняка будет отличаться, даже на похожие объекты. Давайте добавим больше точек.

Линейная регрессия

Пример



Мы видим, что точки действительно выстраиваются в ряд, подтверждая нашу гипотезу о зависимости цены от площади объекта недвижимости. То есть при построении модели мы хотим определить такую линию, от которой цены на бронь жилья будут не сильно отличаться.

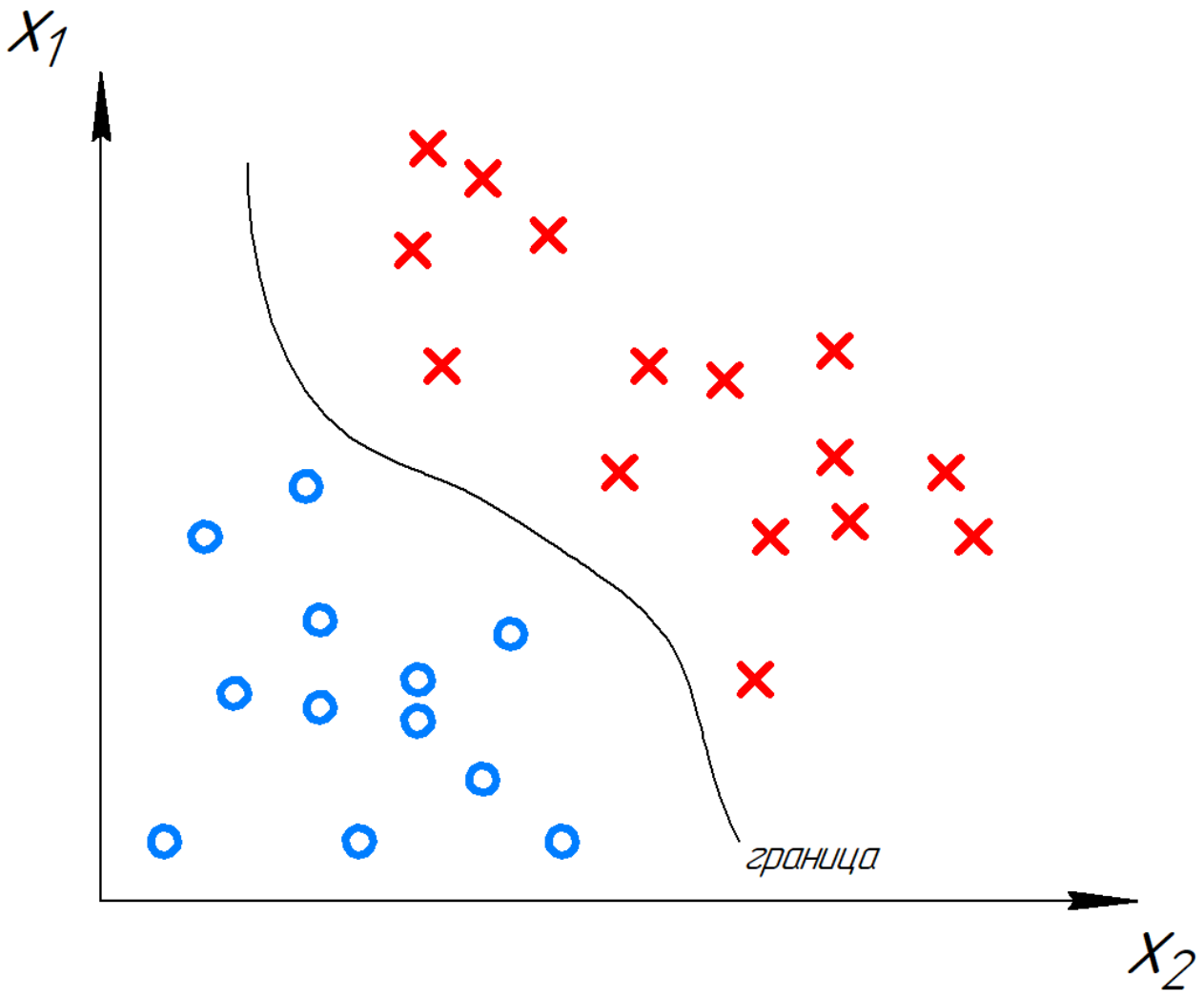
Конечно, не всегда зависимость можно отобразить с помощью прямой линии, но это будет уже называться нелинейной регрессией — линии в таких графиках не будет прямой, но все равно будет отображать зависимость.

Классификация

Сейчас мы рассмотрим еще одну задачу обучения с учителем — классификацию. Если нам например нужно вернуть в результате не число, то это не задача регрессии.

Допустим мы работаем с товаром в онлайн магазине и хотим определить, дорогой это товар или нет. То есть нам нужна не сама цена товара, а принадлежность этого товара к одной из групп — дорогой или дешевый. В этом случае мы работаем с задачей классификации — нам нужно определить к какому классу относится товар.

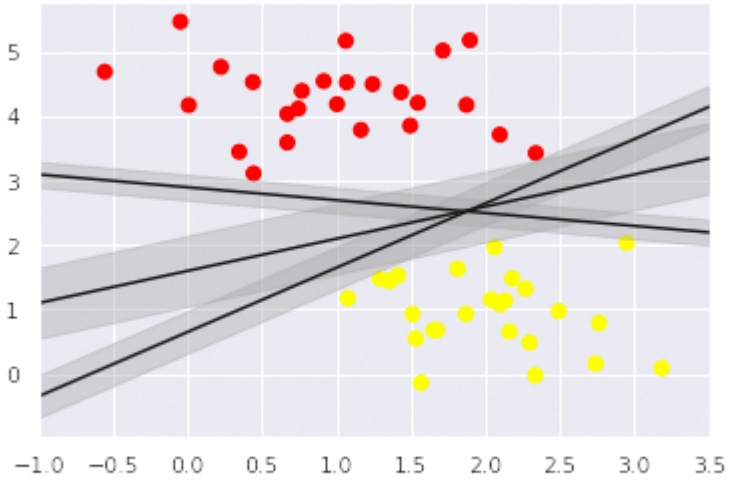
То есть в задаче классификации имеется множество объектов, разделенных некоторым образом на классы. Классовая принадлежность других объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества



В данном примере на слайде у нас есть некие объекты, которые обладают своими характеристиками — x1 и x2. Мы знаем для объектов кто их них x1 а кто x2. Нам нужно определить правила, которые определят классы неизвестных объектов.

Классифицировать объект — значит указать номер или наименование класса, к которому относится данный объект. То есть сам класс это и есть результат функции, который мы хотим определить. А правильные метки классов в исторической выборке — наш учитель. Поэтому это задача обучения с учителем.

Сама классификация объектов предполагает создание некоторой разделяющей плоскости по характеристикам объектов, чтобы можно было однозначно определить объект в тот или иной класс.



Посмотрите на картинку, здесь представлены три различных варианта проведения линии, с одной стороны которой будут объекты одного класса, а с другой стороны другого класса. Очевидно, что таких плоскостей, в данном случае линий, может быть еще несколько вариантов. Наша задача — найти лучший.

Функция потерь

Итак, все основные понятия практически рассмотрены. Осталось еще одно, не менее важное. И это понятие функция потерь. Что это такое? Давайте разбираться.

В процессе обучения нам нужно подобрать такую функцию, которая приближала бы предсказания к ответам из исторической выборки. Естественно, на глаз мы это не сделаем. Нужно сделать это как-то математически. Для этих целей существует функция потерь, которая может посчитать ошибку, и чем больше будет эта ошибка, тем хуже модель справилась с предсказанием.

Самый простой способ — вычесть предсказание из истины, но это не всегда возможно. Сумма, иногда усредненная, функции потерь по всем объектам, называется эмпирическим риском. Чем меньше эмпирический риск, тем точнее работает модель машинного обучения. Далее рассмотрим несколько несложных функций потерь для задачи регрессии.

Рассмотрим простейшие функции потерь для задачи регрессии
Для начала импортируем необходимые библиотеки:

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
```

Начнем с примера.

Пусть имеется набор точек x = [0, 1, 2, 3, 4, 5] и
y = [0, 1, 2, 3, 4, 5]

```
x = np.array([0, 1, 2, 3, 4, 5])
y = np.array([0, 1, 2, 3, 4, 5])
```

Отобразим их на графике

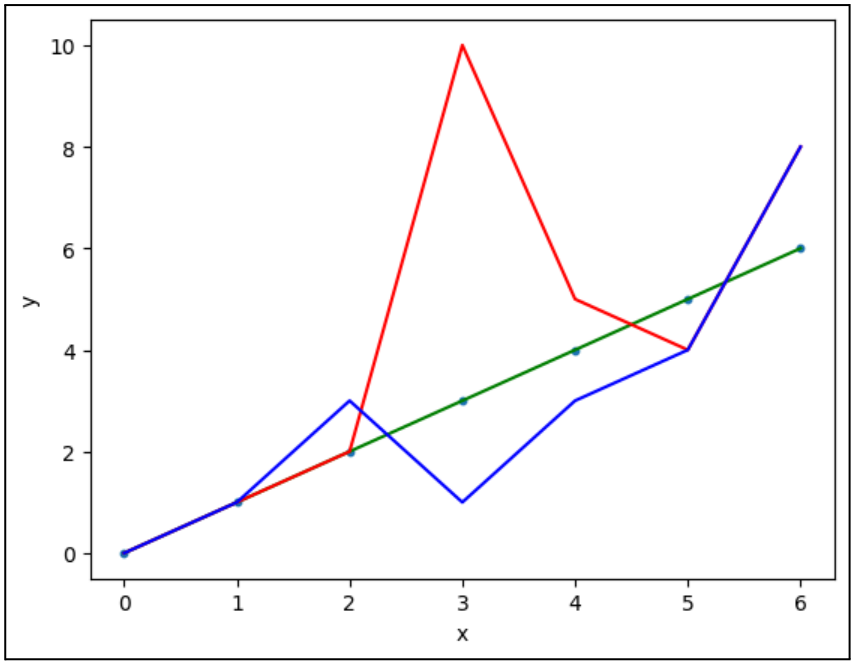
```
plt.scatter(y, y, s=30)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

Очевидно, что точки лежат на одной линии, но представим, что мы этого не знаем и нам надо определить эту линию.

Вариантов может быть достаточно много:

```
y1 = np.array([0, 1, 2, 3, 4, 5])
y2 = np.array([0, 1, 2, 10, 4, 5])
y3 = np.array([0, 1, 4, 4, 5, 5])

plt.scatter(y, y, s=30)
plt.plot(y1, 'g')
plt.plot(y2, 'r')
plt.plot(y3, 'r')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```



Видим, что зеленая линия — это то, что мы хотели найти.

Осталось доказать математически, что это наилучшее приближение функции (линии) к исходному набору данных (точкам).

Воспользуемся самой простой идеей и вычтем из правильных ответов предсказанные и сложим результат.

Для y3 получим (давайте устно посчитаем ошибку):

(0-0)+(1-1)+(2-4)+(3-4)+(4-5)+(5-5) = -4

Аналогично посчитаем для y1 и y2, а использование типа данных numpy array упростит наши вычисления:

```
 e1 = sum(y - y1)
 e2 = sum(y - y2)
 e3 = sum(y - y3)

print(e1)
print(e2)
print(e3)

0
-6
-4
```

И для получения средней ошибки для каждого объекта разделим полученное значение на число объектов.

Для y3 получим следующее:

((0-0)+(1-1)+(2-4)+(3-4)+(4-5)+(5-5)) / 6 = -4 / 6 = -0.66

```
e1 = sum(y - y1)/len(y)
e2 = sum(y - y2)/len(y)
e3 = sum(y - y3)/len(y)

print(e1)
print(e2)
print(e3)

0.0
-1.0
-0.6666666666666666
```

Функция потерь в машинном обучении является мерой того, насколько точно полученная модель способна предсказать целевую функцию (правильные ответы).

В рассмотренном примере: мы посчитали самую простую функцию потерь и получили, что в y1 наиболее точные предсказания.

Функция потерь по всем объектам называется эмпирическим риском. Чем меньше эмпирический риск, тем точнее работает модель машинного обучения.

Только рассмотренную функцию используют редко, а вот ее небольшое видоизменение довольно-таки часто: называется такая функция потерь MAE — mean absolute error — средняя абсолютная ошибка. Это еще одна функция потерь, используемая в регрессионных моделях.

MAE — это сумма абсолютного значения разницы (то есть по модулю) между целевым значением и прогнозируемым значением. Он только измеряет среднюю длину модуля предсказанной ошибки значения, независимо от направления, и диапазон значений составляет от 0 до положительной бесконечности.

То есть, чем ближе MAE к 0, тем модель лучше.

Рассчитаем MAE для нашего примера:

```
e1 = sum(abs(y - y1))/len(y)
e2 = sum(abs(y - y2))/len(y)
e3 = sum(abs(y - y3))/len(y)

print(e1)
print(e2)
print(e3)

0.0
1.3333333333333333
0.6666666666666666
```

Аналогичный функционал есть в библиотеке sklearn:

```
from sklearn.metrics import mean_absolute_error

mean_absolute_error(y, y3)
```

```
0.6666666666666666
```

Рассмотрим следующий пример.

Пример 2

Сгенерируем большее количество точек, для это воспользуемся функционалом numpy random.

Команда np.random.rand (10, 1) сгенерирует 10 случайных чисел со значениями 0 до 1

```
np.random.rand(10, 1)

   array([[0.89984445],
          [0.65435709],
          [0.76107974],
          [0.4546634 ],
          [0.51032297],
          [0.44049049],
          [0.3941815 ],
          [0.25895252],
          [0.90411821],
          [0.22347952]])
```

Команда np.random.randn (10, 1) сгенерирует 10 случайных чисел, среднее которых будет равно 0

```
 np.random.randn(10, 1)

    array([[ 0.0616352 ],
           [-0.12105073],
           [ 0.87713122],
           [-0.12899918],
           [ 0.94563367],
           [-0.01826823],
```

```
           [ 1.74189031],
           [-0.26926341],
           [ 1.32004379],
           [ 0.4835326 ]])
```

Пусть нам заданы 100 точек:

```
np.random.seed(0)    # для воспроизведения результатов
x = np.random.rand(100, 1)
y = 1 + 5 * x + np.random.randn(100, 1)

plt.scatter(x, y, s=10)
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

У нас вновь может быть несколько вариантов того, как модель может выглядеть: наша задача определить, какая модель будет лучше — то есть с минимальной функцией потерь.

```
y1 = 1 + 5 * x
y2 = 3 + 5 * x
y3 = 1 + 1 * x

plt.scatter(x, y, s=10)
plt.plot(x, y1, 'g')
plt.plot(x, y2, 'r')
plt.plot(x, y3, 'r')
plt.xlabel('x')
plt.ylabel('y')
plt.show()
```

Определим, какая модель лучше с помощью уже известной нам MAE:

```
e1 = mean_absolute_error(y, y1)
e2 = mean_absolute_error(y, y2)
e3 = mean_absolute_error(y, y3)

print(e1)
print(e2)
print(e3)

0.8623845994287561
1.8153285514201543
2.180005707537805
```

Получаем, что ответы модели y1 (зеленая линия на графике) дают минимальное значение MAE, что означает, что прогноз, согласно модели 1, наиболее точный.

Полный код вы можете найти в блокноте по ссылке:

https://colab.research.google.com/drive/1BtVOHUOmh0JMFtCGJb_JYjUjrMb_PaZ1?usp=sharing

Заключение

Так же давайте введем еще одно определение, которым я буду пользоваться на следующих лекциях. Функция потерь по всем объектам называется эмпирическим риском.

В заключении занятия давайте еще раз повторим основные тезисы, которые вы услышали сегодня. Это поможет вам не запутаться на следующих занятиях, когда мы будем говорить про более алгоритмы и иные подходы в машинном обучении.
Здесь нужно понимать, что пока мы в роли универсальных инженеров. На курсе мы делаем все из ниже перечисленного. На практике есть команда, и у каждого человека своя роль.

Итак, прежде всего машинное обучение — это данные. Данные определяют какой алгоритм мы используем для решения задач машинного обучения. Данные — это

большая работа, так как их нужно не только собрать, но еще и правильно обработать. Про данные мы будем говорить чуть позже, когда пройдем все линейные методы.

Далее признаки — характеристики объектов. Какие бы данные мы с вами не собрали, если признаки плохие — модель точно не получится. Признаки это то, что влияет на модель. Правильный подбор признаков — важная задача

Подведем итог: Что же такое модель машинного обучения? Это функция. Когда мы обучаем модель — мы минимизируем эту функцию. Есть уже готовые алгоритмы, и по сути инженеру машинного обучения нужно лишь выбрать, каким алгоритмом воспользоваться.

Минимизация функции происходит всегда по-разному — например в линейной регрессии, так как у нас прямая линия, мы ищем квадрат расстояния между разделяющей плоскостью и объектами. Для логистической регрессии вообще нет явного минимизатора функции потерь (об этом мы поговорим на третьем занятии).

Получается, что какой бы линейный алгоритм мы не рассматривали — у нас всегда будет функция, которую мы будем минимизировать, а также способ минимизации этой функции.

Сегодня мы рассмотрели основные термины машинного обучения, узнали чем отличаются разные подходы в машинном обучении, и рассмотрели на примере процесс обучения модели. Мы научились искать минимум функции с помощью функции потерь. Это отличный старт для вашего обучения! На следующем занятии мы будем знакомиться с первом алгоритмом, рассмотрим его применение на практике, и поговорим про подводные камни, которые сопровождают нас в процессе обучения модели. До встречи!

Что можно почитать еще?

1. https://gb.ru/blog/mashinoe-obuchenie/ - введение в машинное обучение
2. https://habr.com/ru/companies/ods/articles/322076/ - введение
3. https://www.youtube.com/playlist?list=PLk4h7dmY2eYHHTyfLyrl7HmP-H3mMAW08 - курс лекций по машинному обучению и математике в машинном обучении

Используемая литература

1. Бринк Х., Ричардс Дж., Феверолф М. - Машинное обучение, 2017 год
2. Андрей Бурков — Машинное обучение без лишних слов, 2020 год
3. Андреас Мюллер, Сара Гвидо — Введение в машинное обучение с помощью Python, 2017 год
4. Джереми Уатт, Реза Борхани, Аггелос Катсаггелос — Машинное обучение: основы, алгоритмы и практика применения