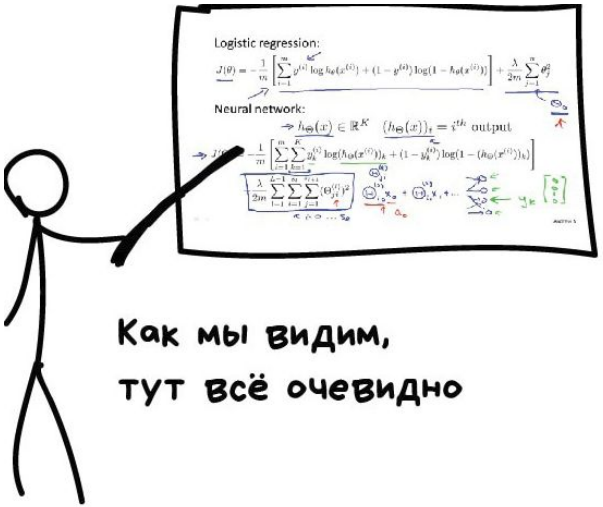
Мир машинного обучения

Маша Шеянова, masha.shejanova@gmail.com



Программисты программируют!

Датасаенс!

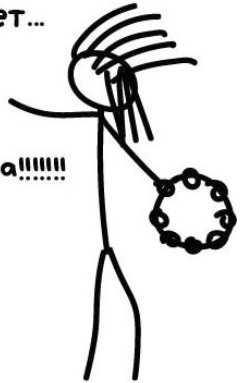
Профессия будущего!

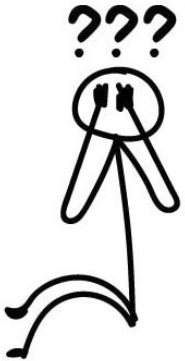
Буквально через пять лет...

Экспоненциально!!!

УМНЫЕ РОБОТЫ!

А-А-А-А-А-А-А-А-А-А-ааа!!!!!!!





Есть два типа объяснений про МО.

Я попытаюсь выбрать адекватную середину.

([Источник комикса])

Что такое машинное обучение?

Роботы всех убьют ?

МО — не про умные машины, захватывающие мир.

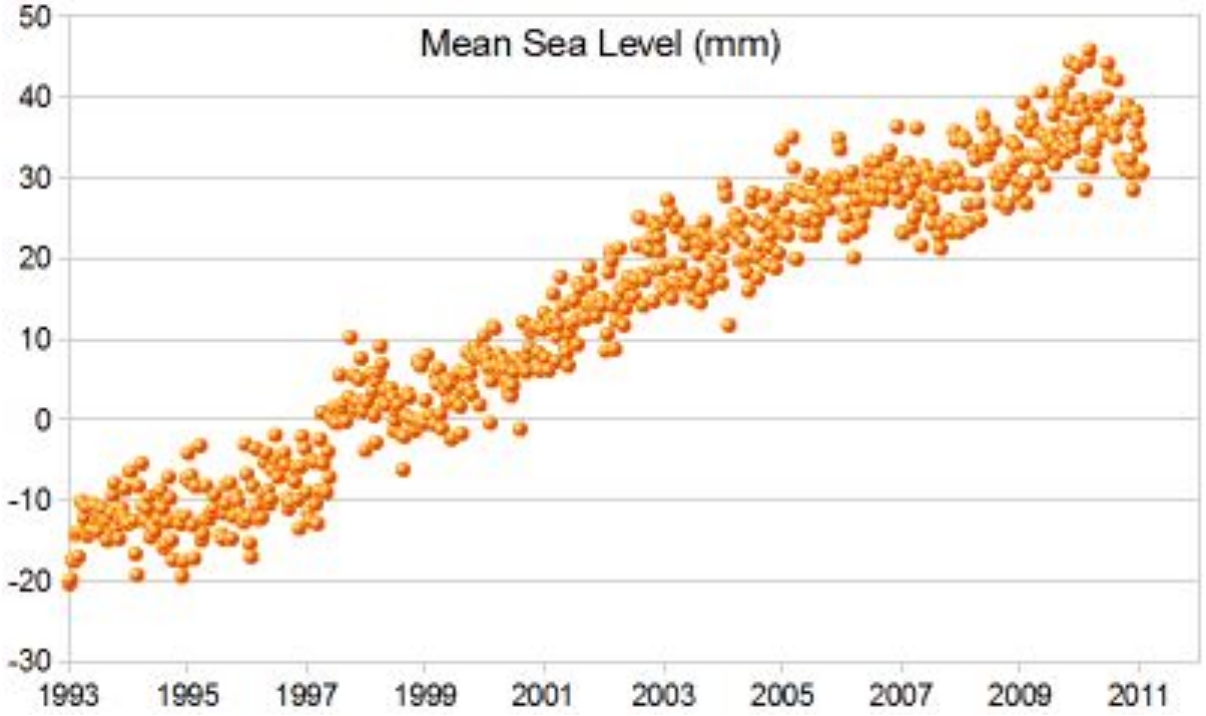
Суть МО в том, чтобы находить **закономерности** в данных.

Машина **может**:

- запоминать
- находить закономерности
- предсказывать
- ранжировать

Машина **не может**: резко поумнеть, выйти за рамки задачи, убить всех людей.

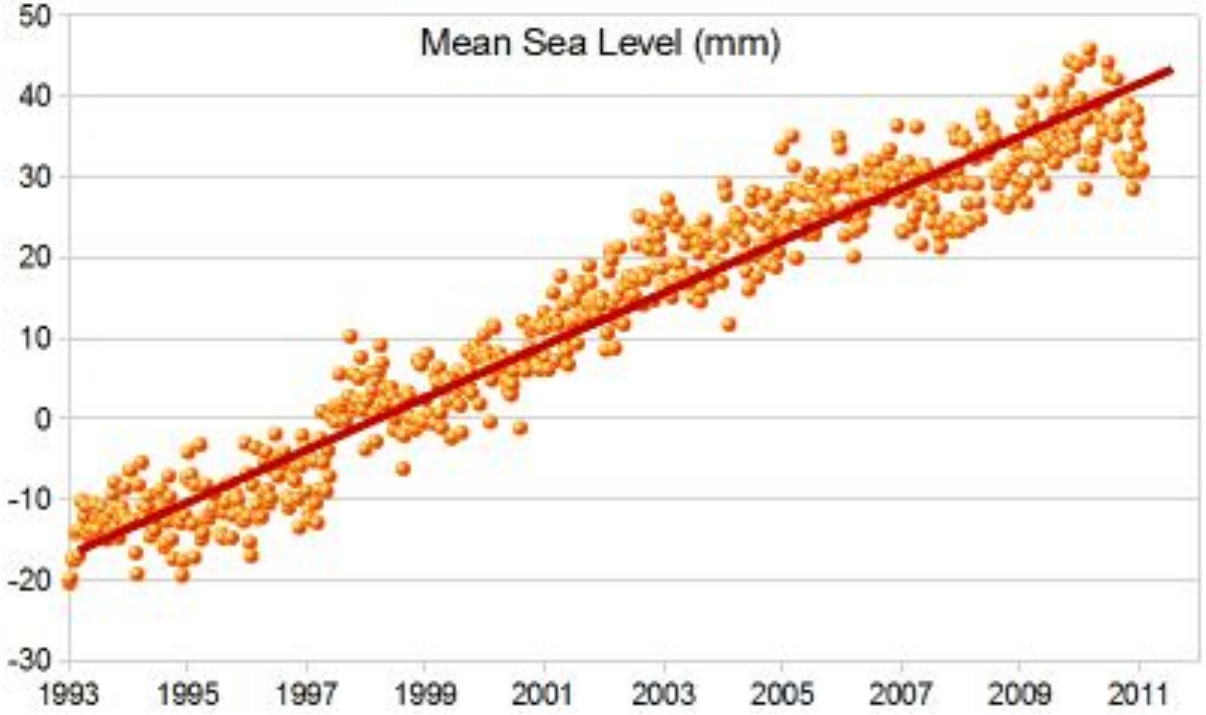
Простой пример



Зависимость уровня океана от времени.

Несложно увидеть закономерность.

Простой пример

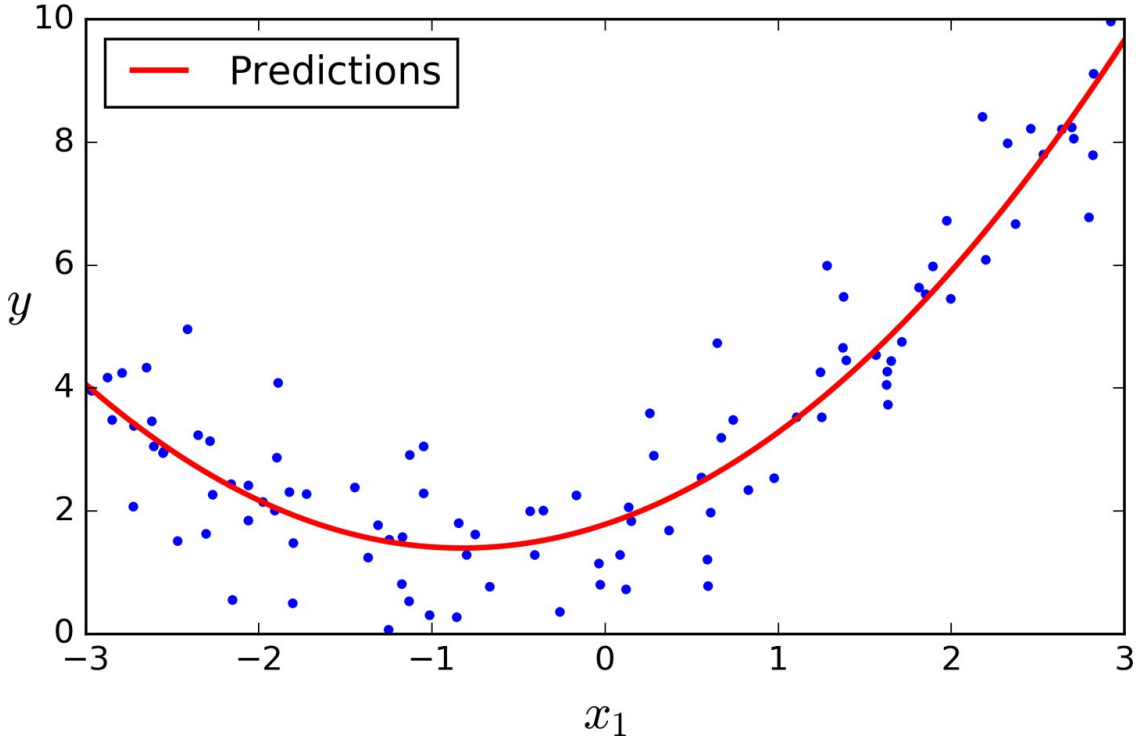


Эта линия -- наша модель, стремящаяся приблизить “закон природы”. Это функция от времени.

Функция выглядит так: $f(x) \backsimeq a * x - b$

Можно найти коэффициенты a и b.

Пример чуть посложнее

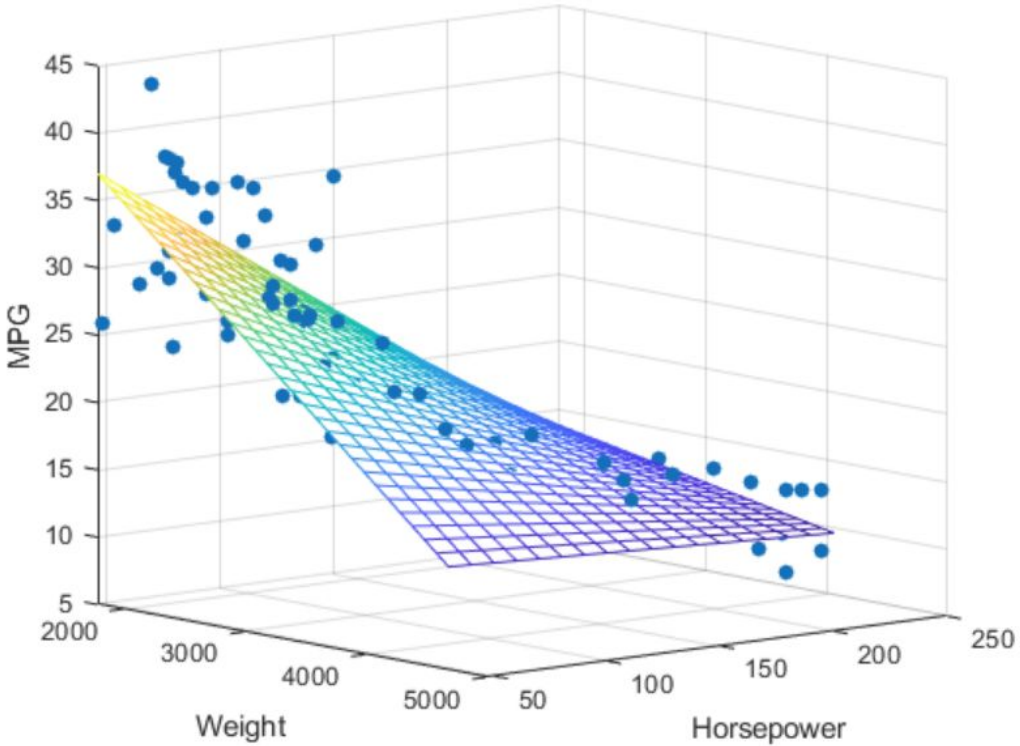


Иногда прямая линия -- не лучшее приближение “закона природы”.

Модель может быть сколь угодно сложной, например $f(x) = a * x^{12} + b * \sin(x) + \ldots$

Чем сложнее модель, тем больше опасность “подогнаться под данные”

Другой пример чуть посложнее



Чаще всего, интересующая нас переменная зависит не от одной, а от кучи переменных!

(Например, от 300)

Что такое модель?

О модели в машинном обучении можно думать как о функции в математике.

Для зависимости веса от роста и пола это будет выглядеть как-то так:

f(16, жен) ≈ 48 f(11, муж) ≈ 38

Для зависимости вероятности выжить на Титанике как-то так:

f(22, жен, 1 класс) ≈ 0.85 f(25, муж, 3 класс) ≈ 0.15

(Дисклеймер: все цифры с неба :)) (Хотя для титаника даже есть данные!)

Что нужно для машинного обучения

1. **Данные**

Какую бы задачу вы ни решали, нужно дать программе сведения о реальном мире. Нельзя обнаружить закономерности, не имея примеров.

2. **Признаки**

Это аргументы функции-модели: то, что мы можем сказать о данных.

3. **Алгоритмы**

Это подходы, позволяющие подобрать коэффициенты функции.

Что ещё важно: метрики качества

Метрики качества — то, как мы понимаем, что модель работает хорошо.

Что значит хорошо? 0% ошибок? Нет! На реальных задачах так не бывает.

Какое качество ОК — зависит от задачи: что для предсказания скачков валют отличная модель, то для распознавания лиц — ужасно.

В каждой задаче машинного обучения свои метрики, основанные на том:

- насколько близко модель предсказывает данные
- сколько раз ошибается и в какую сторону
- etc

От задачи обучения к оптимизации

Итак, нам нужно подобрать параметры модели (коэффициенты).

Как? На основании чего? Нам нужно знать **функцию потерь**.

Функция потерь — то, как мы меряем “неправильность” модели. Её мы будем минимизировать. Сначала берём коэффициенты “с потолка” — рандомно. Считаем функцию потерь и пытаемся понять, как изменить коэффициенты, чтобы функция потерь уменьшилась.

Последний шаг часто делается с помощью **градиентного спуска**.

Самое главное в ML модели

→ **обобщающая способность** ←

Идея такая: несложно научиться предсказывать результаты на данных, на которых модель учили. Про них мы и так всё знаем. Хорошо ли она работает с новыми данными — вот что важно.

Как это оценить? Делим данные на две “кучки”: одна для обучения, другая для проверки (**train** и **test**).

Чтобы нивелировать влияние от конкретного разбиения, делаем кросс-валидацию (“скользящий контроль”).

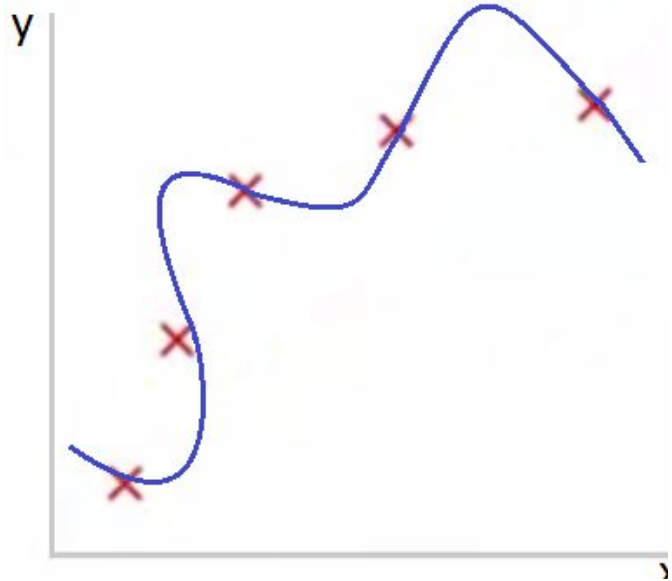
Что такое переобучение?

Переобучение означает, что модель обращает слишком много внимания на незначительные признаки, чтобы “слишком хорошо” подстроиться под данные на обучающей выборке.

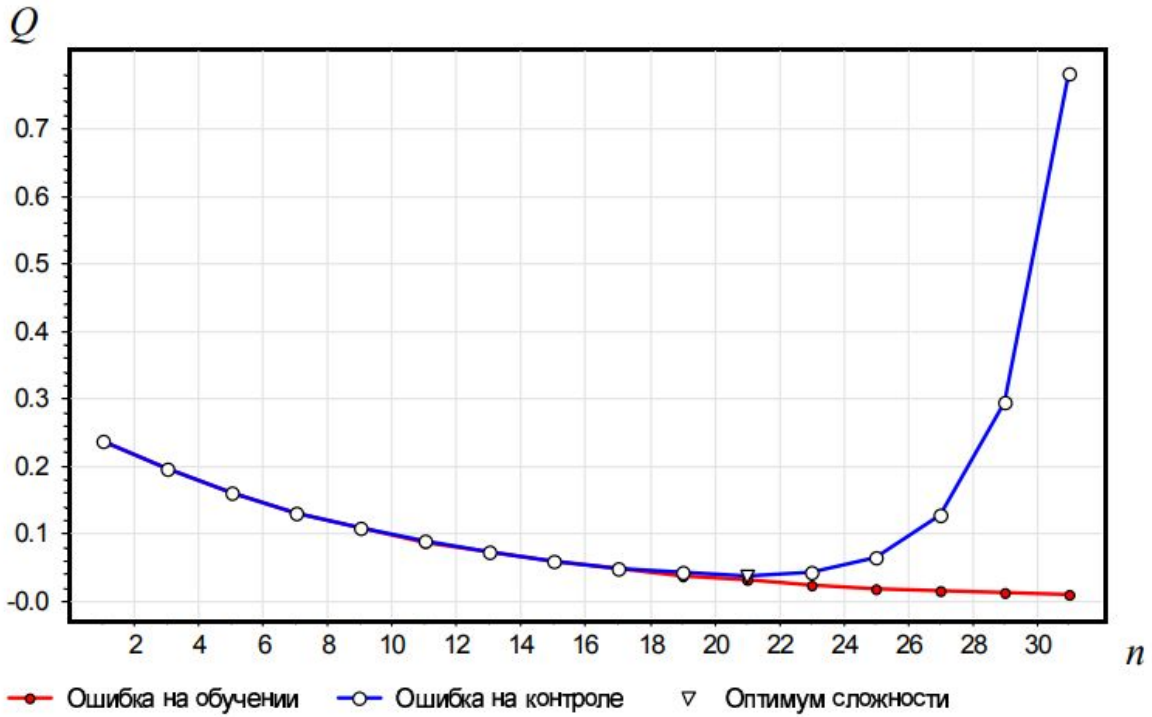
При этом модель становится излишне сложной (посмотрите на эти ненужные загогулины).

Скорее всего, она покажет плохой результат на тестовой выборке.

Как понять, что модель переобучилась?



Переобучение — это когда



Источник картинки.

Иными словами, если качество модели на обучении продолжает расти, а на тесте — падает, модель переобучилась.

Переобучение

Из-за чего возникает переобучение:

- слишком сложная модель
- ограниченность обучающих данных

Как обнаружить переобучение:

- train vs. test quality
- кросс-валидация

Избавиться полностью от переобучения нельзя. Надо его минимизировать.

Машинное обучение vs. правила

Пример: я хочу написать систему, которая отличает спам от не-спама.

На правилах: я пишу фильтр на определённые слова и явления, например:

- “бриллианты” > 3 раз => спам
- “Уважаемая Госпожа” => спам
- любое слово > 30% текста => спам

На МО: я скармливаю программе много сообщений, помеченных как “спам” и “не спам”. Указываю, как брать признаки из данных. Она разбирается сама.

Машинное обучение vs. правила

Машинное обучение:

- требует (много) данных
- и вычислительных ресурсов
- модель учится **сама**

Правила:

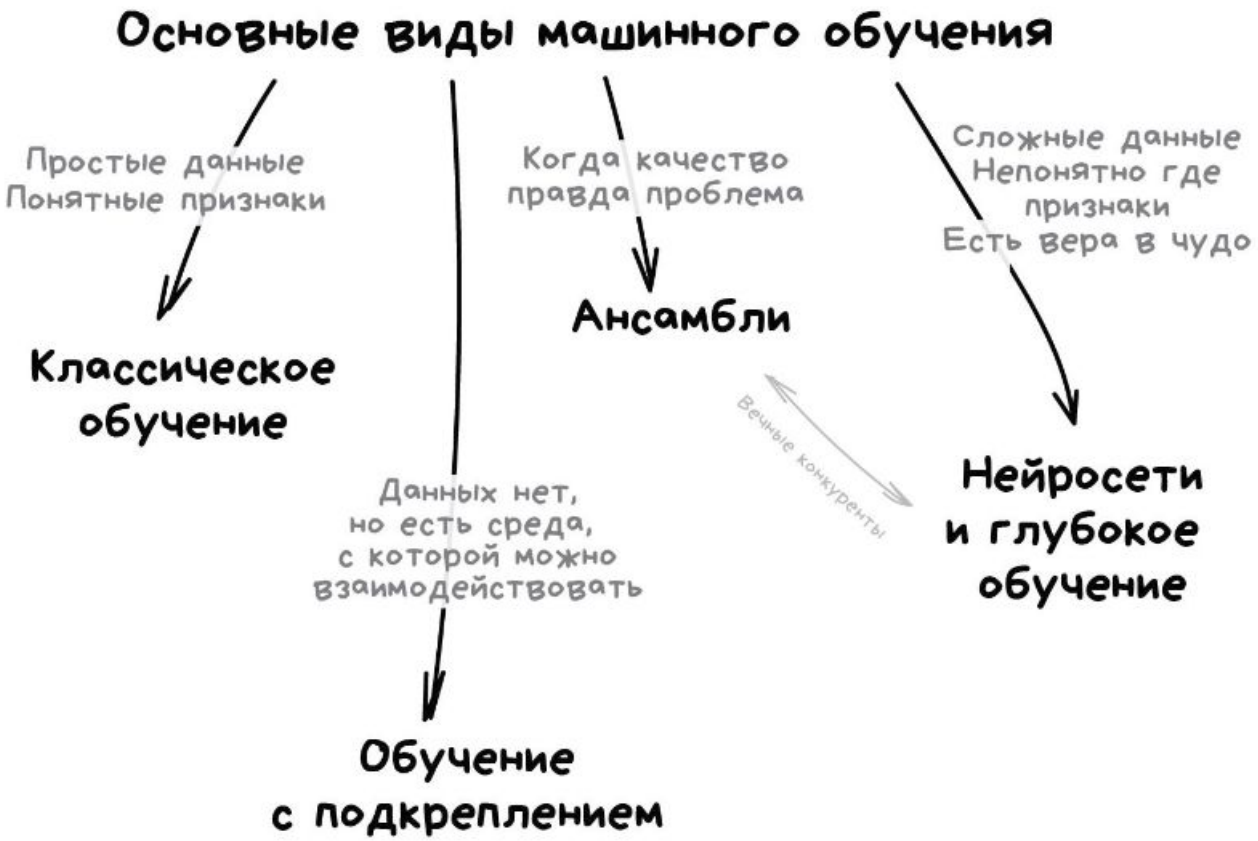
- не требуют данных вообще, главное знать, что делаешь
- надо хорошо знать, что делаешь
- и делать это, дооолго и нудно

Почему МО — это круто

- ~~позволяет чувствовать себя умным~~
- умеет обнаруживать закономерности быстрее, чем человек
- и иногда лучше, чем человек (когда речь о сложных явлениях)
- не страдает от предвзятости*
- экономит человеческое время (и затраты компаний на него)

* Если данные хорошие и задача поставлена корректно

Области машинного обучения



И снова картинка из “Машинное обучение для людей”. А здесь — подробная карта.

МО бывает разным. **Классическое МО** — основа других методов, поэтому в рамках этого курса сконцентрируемся на нём.

(Да-да, снова оттуда)

Самый типичный вариант (и тот, на который мы смотрели до сих пор) — обучение с учителем.

Обучение с учителем vs без учителя

Обучение с учителем:

- есть данные, про которые мы знаем “правильный ответ”
- мы пытаемся найти в данных закономерность, которая приводит к нужному ответу
- работает обычно гораздо лучше, чем без учителя

Обучение без учителя:

- “правильные ответы” не нужны
- пытаемся вывести закономерность “вслепую”
- гораздо сложнее и работает хуже, но зато с данными нет проблем!

Обучение с учителем: виды

- регрессия

“Найди одно число по ряду других”. Пример: угадать вес по росту и возрасту. Значение редко угадывается точно, но это и не важно.

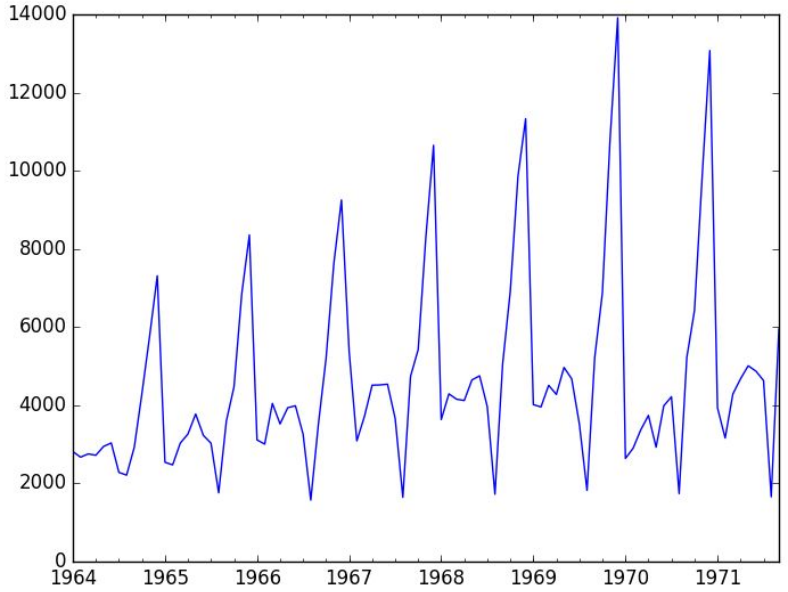
- классификация

“Разложи объекты разных форм и цветов в коробки с подписями”. Пример: является ли email спамом? Выживет ли человек на Титанике?

- ранжирование (“отранжируй страницы в выдаче поисковика”)
- прогнозирование временных рядов

Прогнозирование временных рядов

Загадка: перед вами график продаж некоторого продукта. Какого?



Обучение без учителя

- кластеризация

“Раскидай мои фотографии по папочкам. Сам реши, по каким”.

У нас есть выборка объектов, но нет заданных классов. Мы хотим разбить их на группы так, чтобы объекты в разных группах сильно отличались.

- снижение размерности

Часто приходится иметь дело с данными больших размерностей. Их сложно хранить и обрабатывать. Мы хотим снизить размерность, оставив наиболее значимые компоненты. Пример: визуализация в 2D.

Ансамбли

По большому счёту, это модели из раздела “обучение с учителем”, но более сложные: они используют простые методы как конструктор.

Например:

- взять много разных классификаторов и устроить “голосование”
- взять много разных классификаторов и заставить их порождать обучающую выборку для другого классификатора

Обучение с подкреплением

Это очень отдельная область МО.

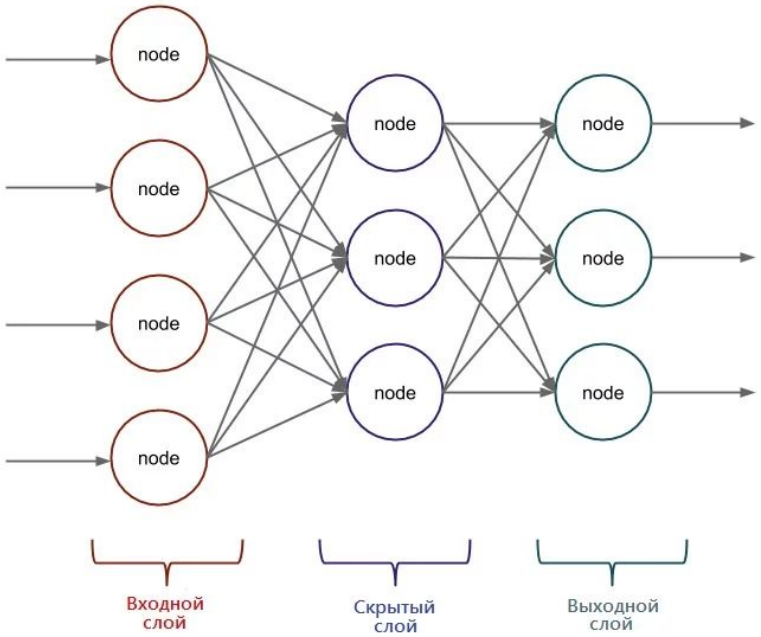
У нас нет вообще никаких данных. Зато есть среда, с которой можно взаимодействовать и учиться. В кои-то веки обучение здесь — в прямом смысле слова.

Единственная (?) область, в которой машина может обгонять человека.

Примеры:

- научить робота ходить
- научиться

Нейронные сети



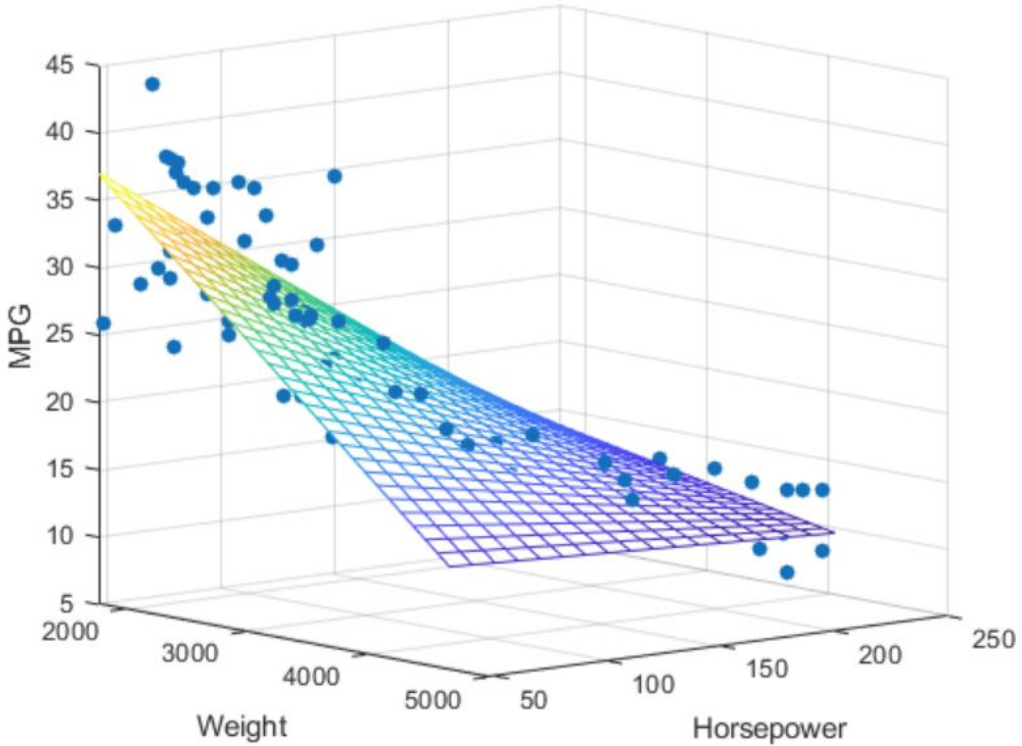
Нейросети умеют всё! Ну, очень много.

Но требуют много данных. И вычислительных ресурсов. И долго работают.

Зато с ними можно не слишком думать про признаки: разберутся сами.

Признаки

Вернёмся к нашему примеру



Здесь всё просто: расход топлива зависит от двух признаков:

- вес машины
- лошадиные силы

Почему точки не лежат на одной прямой?

- неучтённые признаки
- шум в данных

Какие признаки бывают?

- бинарные признаки

Принимают значения 1 или 0. Примеры: пол, есть ли в тексте слово “котик”

- количественные признаки

Множество действительных чисел. Пример: рост, вес, возраст.

- порядковые признаки

Конечное упорядоченное множество значений, e.g. уровень образования.

- категориальные признаки

Конечное множество значений. Пример: национальность, язык.

Как быть с категориальными признаками?

Допустим, мы предсказываем, сколько человек тратит на кофе в день, по национальности. Если мы пишем правила, нет никаких проблем:

```
if x == “english”:
    sum += 0.5
elif x == “french”:
    ...
```

Но что если мы хотим подобрать коэффициент при переменной? Что будет значить 0.2 * x?

На помощь приходит one-hot encoding.

One-hot encoding

Упорядочиваем значения категориальной переменной... и превращаем её в столько бинарных переменных, сколько у неё было значений.

значение
french
english
italian
russian

is_french	is_english	is_italian	is_russian
1	0	0	0
0	1	0	0
0	0	1	0
0	0	0	1

Теперь можно подбирать коэффициенты!

Как добыть признаки из текста?

Самый очевидный пример: вхождение слова в текст. Бинарный признак: “входит ли слово X” в мой текст — для каждого слова.

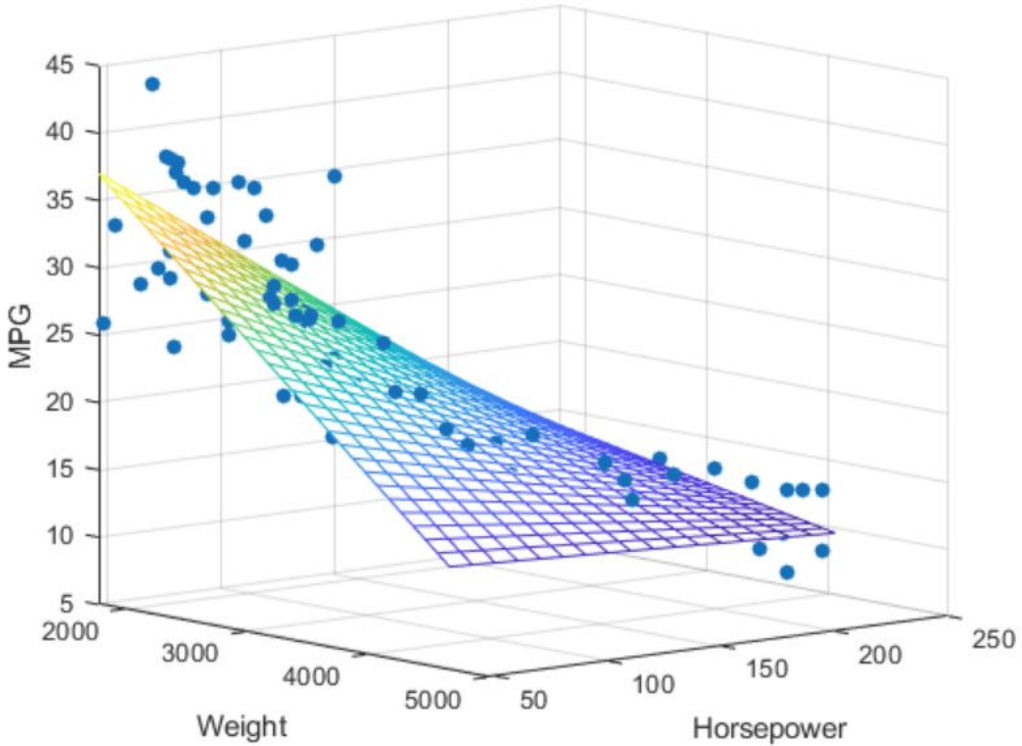
Другие признаки в тексте:

- сколько в тексте восклицательных знаков
- есть ли слова КАПСОМ
- есть ли в тексте биграмма “синие занавески”
- сколько в тексте существительных
- сколько в тексте сложноподчинённых предложений

Главное — чтобы признаки имели смысл для решаемой задачи.

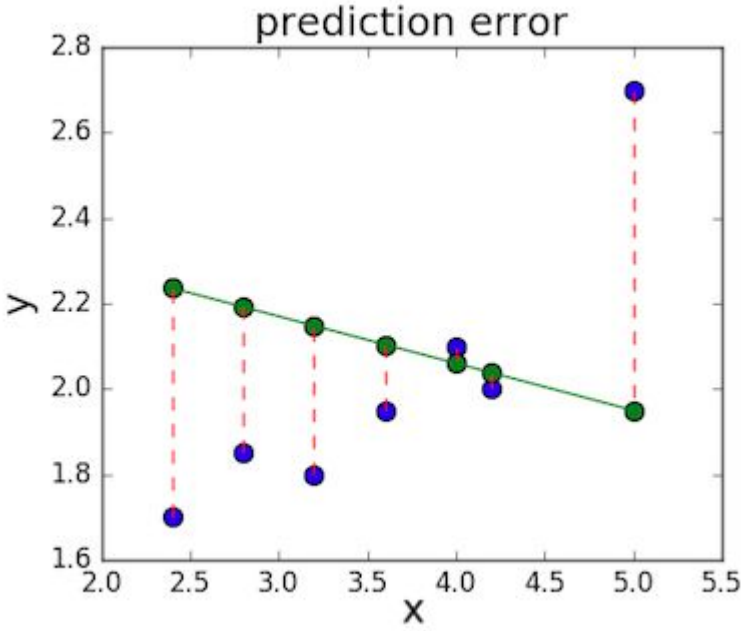
Регрессия

Вернёмся к нашему примеру ещё раз



Как посчитать, насколько хорошо модель описывает данные?

Метод наименьших квадратов



Нужно посчитать, насколько сильно наша модель “мажет” мимо реальных данных.

Как это сделать? Сумма разностей между точками?

Есть подвох. Какой?

Функции потерь для регрессии

MSE - Mean squared error - среднеквадратичная ошибка

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

MAE - Mean absolute error - средняя абсолютная ошибка

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n}$$

Полезные ресурсы

Kaggle!

Kaggle — идеальное место для человека, занимающегося МО.
(“Всемирный чемпионат по машинному обучению”)

Это платформа, на которой предприятия устраивают соревнования, открывая доступ к своим данным, и иногда вычислительным ресурсам.

На kaggle можно найти:

- туториалы по самым разным сферам МО
- соревнования с призами
- открытые датасеты
- закончившиеся соревнования и лучшие решения

Почитать / посмотреть

Статьи про МО:

- Машинное обучение для людей — про всё МО доступным языком (рус)
- Понятная статья про линейную регрессию (рус)
- Другая понятная статья про линейную регрессию (англ)
- про переобучение (англ)

Курсы и видео:

- как объяснить вашей бабушке про машинное обучение (ничего нового)
- курс КВ Воронцова в ШАДе (крутой хардкор)

Список терминов на английском

Machine Learning (ML) — машинное обучение (МО)

Overfitting — переобучение

Supervised / unsupervised ML — обучение с учителем / без учителя

Reinforcement learning — обучение с подкреплением

Loss function — функция потерь

Gradient descent — градиентный спуск