



Занятие 1. Введение в машинное обучение

Колмагоров Евгений
ekolmagorov98@yandex.ru

7 октября 2024

Знакомство



Лектор
Колмагоров Евгений
MLE в ecom.tech
t.me/ekolmagorov



Ассистент
Кочарян Давид
Аналитик в MediaScope
t.me/rufos_20

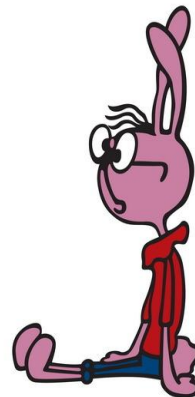
Цель курса

1. Рассмотреть основы машинного обучения: термины, классические алгоритмы, а также рассмотреть подходы к решению прикладных задач
2. На примере языка Python и готовых библиотек рассмотреть сферы применимости того или иного алгоритма
3. Попробовать на примере реальных задач посмотреть границы применимости машинного обучения

Программа курса

Очень насыщенная и интересная:

- 14 занятий по основным темам машинного обучения. На которых будут рассмотрены как теоретические так и практические аспекты
- 5 домашних заданий
- 2 контрольных проекта
- 1 Финальный по окончании курса



Правила игры

- Оценка за курс представляет собой “зачёт”/”не зачёт”
- Чтобы получить “зачёт” за курс необходимо набрать $\geq 60\%$ от максимального количества баллов
- Баллы за курс слагаются из:
 - 1) Домашних работ
 - 2) Сданных проектов
 - 3) Небольших квизов по материалам предыдущей лекции
- Кому не хватило баллов за работу в течении курса необходимо сдавать письменный экзамен по окончанию курса

Немного философских вопросов

1. Что значит вообще "обучение"?

2. Зачем вообще возникла
необходимость обучать машину?

3. Как понять, что компьютер чему-
либо "научился"?

4. Чему может научиться компьютер,
а чему принципиально нет?



Попробуем дать ответ на первый вопрос

“Целью обучения является не
получение знаний, а умение
действовать со знанием дела”

П. Я. Гальперин



Теперь попробуем ответить на второй вопрос

С самых первых дней появления компьютеров, учёные и нейрофизиологи поняли насколько общими могут быть программы для ЭВМ. То идеи об эмуляции мозговой деятельности человека не заставили себя долго ждать

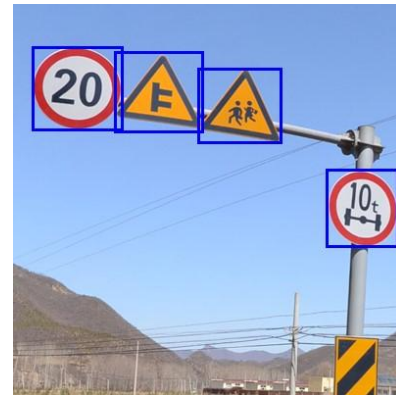


Фрэнк Розенблатт и "Марк-1"

Осталось ответить на третий вопрос

Ответить на этот вопрос не так просто на самом деле по той причине, что цель обучения - это решение конкретной задачи. Поэтому судить о том, научилась ли машина чему-либо, можно только исходя из качества решения целевой задачи.

И если задачу распознавания дорожных знаков понятно, как мерить, то качество работы голосового ассистента уже не так очевидно...



А на четвёртый вопрос пока ответ дать нельзя

На текущей стадии развития алгоритмов искусственного интеллекта сложно ответить на вопрос о границах их применения.

Наука не стоит на месте и каждый день изобретаются новые подходы способные решать всё более сложные задачи

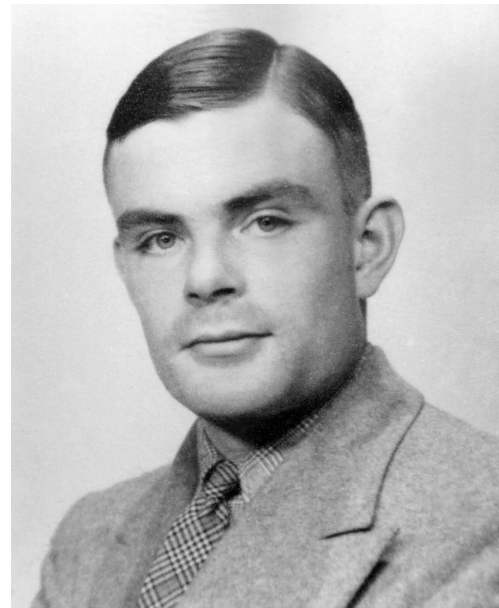


OpenAI



История развития. Первые идеи

Одни из первых, кто ещё в 1947 году высказался об идеи создания “интеллектуальных” машин, которые должны изменять свое внутреннее состояние исходя из полученного опыта, был родоначальник компьютерных наук английский математик-программист Алан Тьюринг.

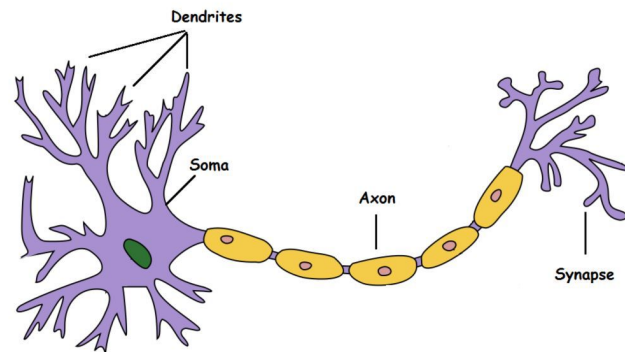


Алан Тьюринг

История развития. Создание первых алгоритмов

В 1957 организуется первые группы, которые ставят задачу создания искусственного интеллекта.

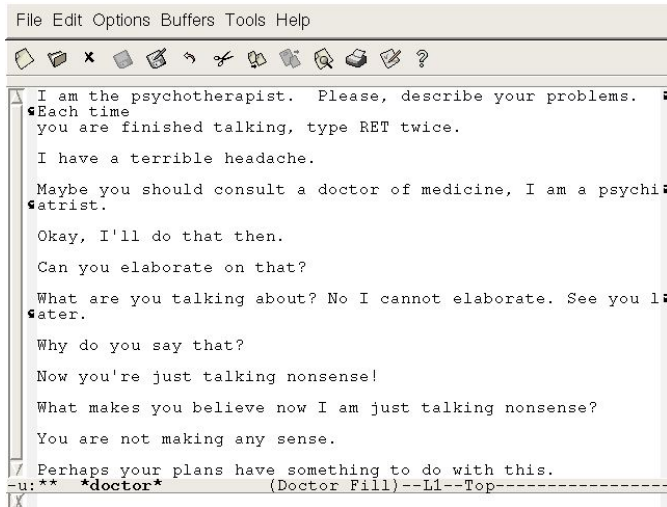
И в 1960 году был создан первый работающий пример алгоритма, моделирующий работу мозгового нейрона - перцептрон Розенблатта, на основе специально созданной машине Mark-1



История развития. Дальнейшие шаги 1960-1970

В 1966 год - создание диалоговой системы Eliza, которая моделирует разговор с психотерапевтом

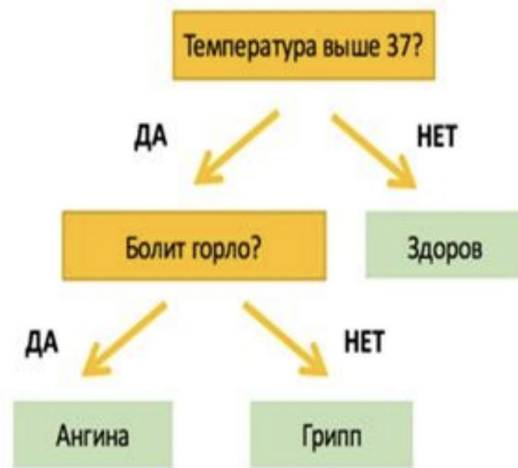
В 1970-е создаются первые “машины вывода”, цель которых производить логический вывод из фактов и правил на основе аппарата математической логики



```
File Edit Options Buffers Tools Help
[Icons]
I am the psychotherapist. Please, describe your problems.
Each time you are finished talking, type RET twice.
I have a terrible headache.
Maybe you should consult a doctor of medicine, I am a psychiatrist.
Okay, I'll do that then.
Can you elaborate on that?
What are you talking about? No I cannot elaborate. See you later.
Why do you say that?
Now you're just talking nonsense!
What makes you believe now I am just talking nonsense?
You are not making any sense.
Perhaps your plans have something to do with this.
-u:** *doctor* (Doctor Fill)--Li--Top-----
```

История развития. Экспертные системы 1980-е

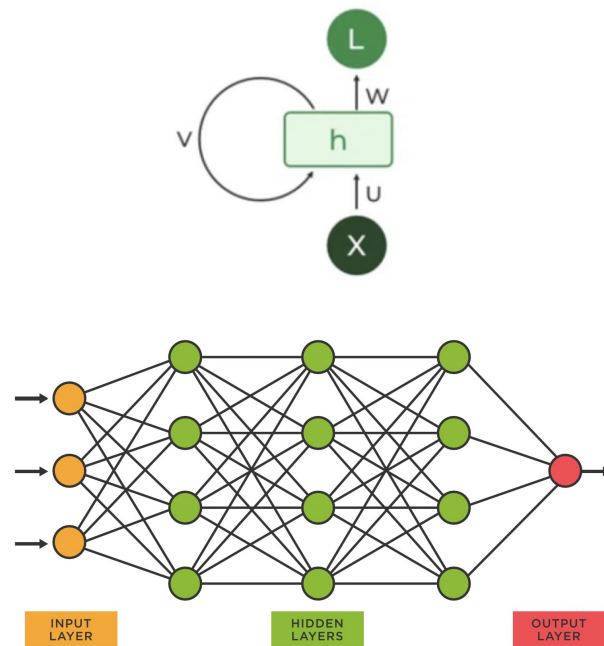
В 1980-х происходит дальнейшее расширение алгоритмов машинного обучения. Происходит расцвет подходов на основе правил (rule-based) и в 1984 создаётся алгоритм автоматического построения решающего дерева



История развития. Первые нейронные сети

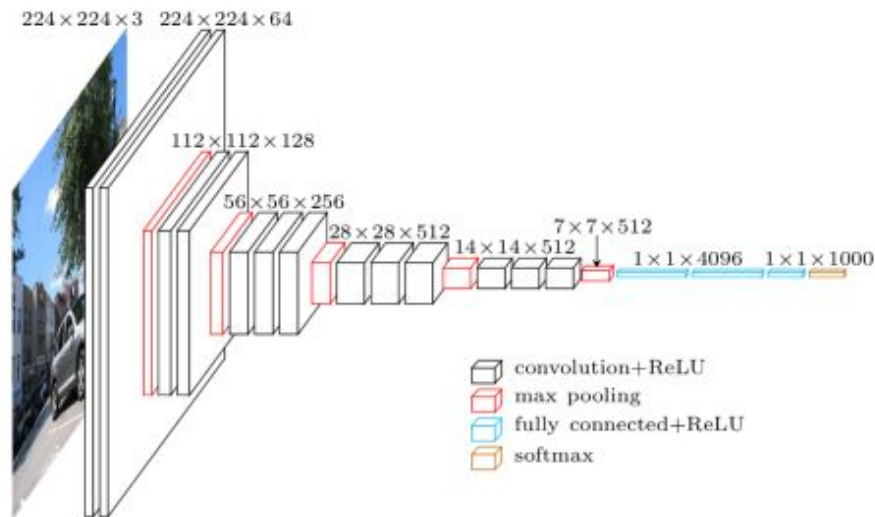
В 1990-х удаётся создать и обучить первую нейронную сеть, которая способна улавливать более сложные зависимости в данных.

Происходит создание принципиально новых подходов к обработке данных на основе нейросетевых алгоритмов и их математической базы



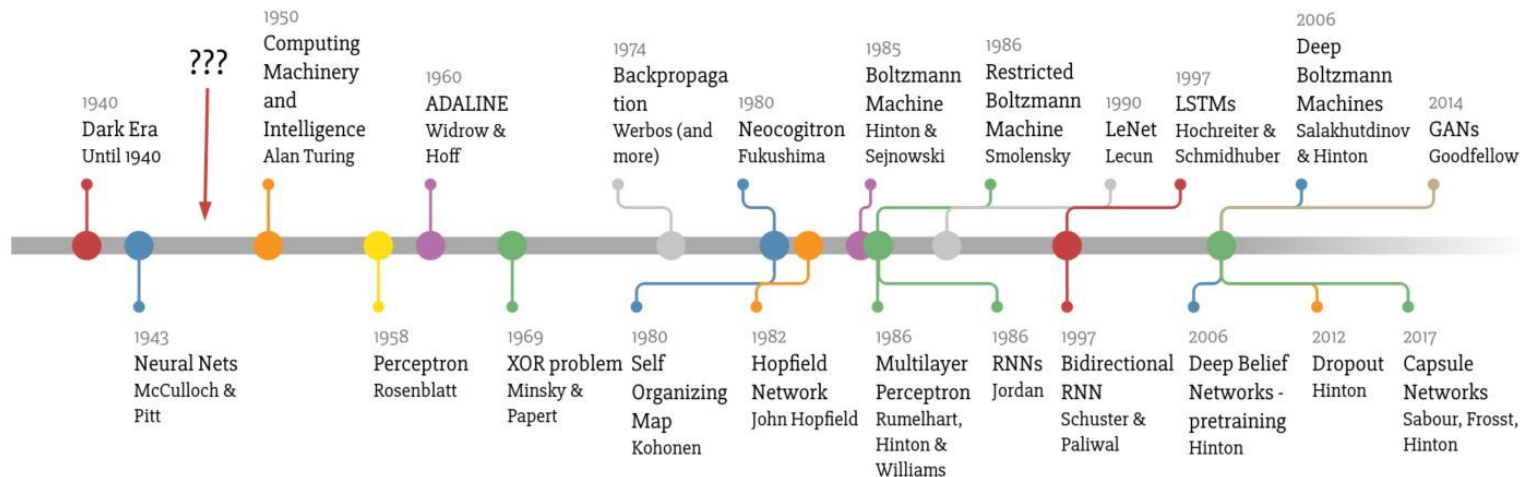
История развития. 21 век расцвет и усложнение существующих подходов

За последние годы произошло стремительное развитие вычислительных устройств и параллельных вычислений, что позволило существенно усложнить нейросетевые подходы.



История развития. Общая картина

Deep Learning Timeline



Made by Favio Vázquez

<https://maefabien.github.io/deeplearning/Perceptron/#why-neurons>

Так что же такое “Машинное обучение”

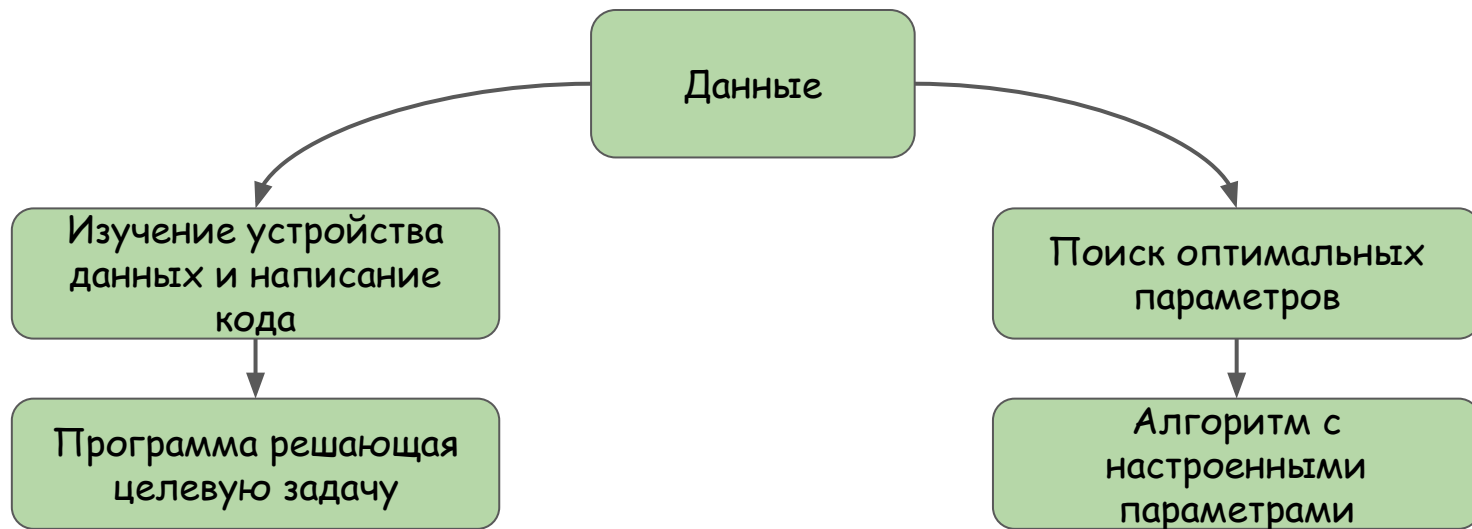
Машинное обучение – раздел науки о данных изучающий процесс, в результате которого компьютер способен показывать поведение на основе данных, которое в нём не было явно запрограммировано

В итоге в чём же заключается принципиальное отличие между программой и алгоритмом машинного обучения?



Отличие программирования от машинного обучения

Главное отличие машинного обучения от программирования заключается в том, что для решения задачи алгоритм машинного обучения **сам извлекает закономерности** из данных и на основе этих закономерностей корректирует своё поведение, в то время как при втором подходе задача извлечения закономерностей лежит на разработчике



Примеры задач. Классификация объектов

Задача **классификации** - определить к какому из ограниченного набора классов принадлежит рассматриваемый объект

iris setosa



petal

sepal

iris versicolor



petal

sepal

iris virginica



petal

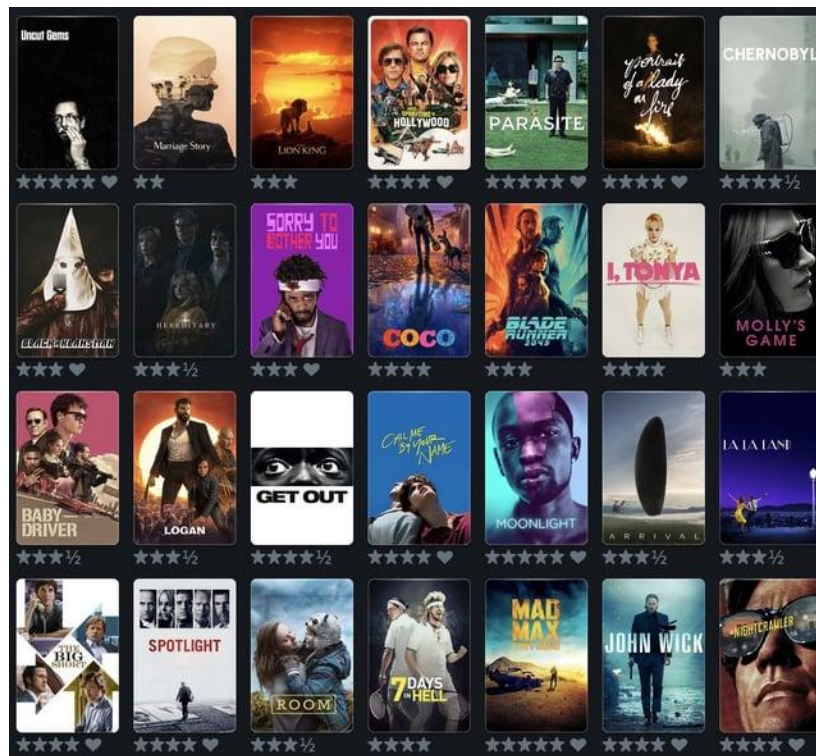
sepal

Примеры задач. Рекомендации

Задача *рекомендаций* - найти наиболее релевантные “объекты” из всего ассортимента, которые могут быть наиболее “интересны” пользователю

Под “интересом” может быть клик, покупка, просмотр, добавление в избранное и тд.

Объектами могут быть фильмы, товары на маркетплейсе, рекламные объявления, услуги и тд.



Примеры задач. Скоринг

Задача *скоринга* заключается в том, чтобы приписать каждому из объектов некоторое вещественное число (score), которое отражает некоторую физическую величину из реального мира.

В качестве таких величин может быть вероятность наступления банкротства, вероятность поломки, оценка релевантности документа поисковому запросу и тд.



Примеры задач. Глубинные подходы

С развитием нейросетевых подходов спектр решаемых задач стал шире, а сами задачи сложнее

Чтение по губам

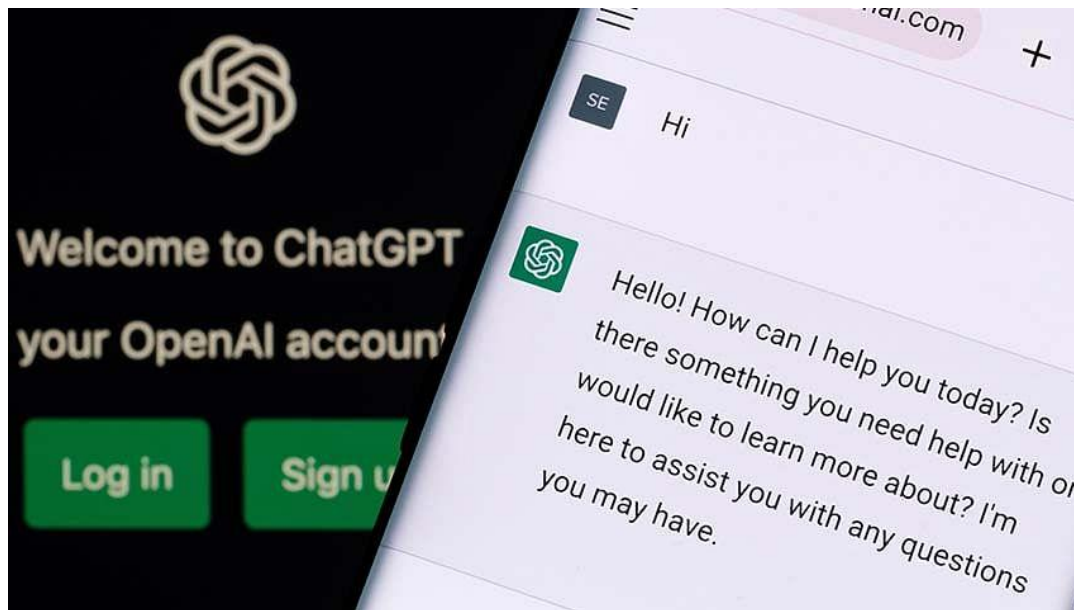
Google Deepmind в 2017 году создали модель, обученную на телевизионном датасете, которая смогла превзойти профессионального lips reader'а с канала BBC.



Примеры задач. Глубинные подходы

Диалоговые системы

В 2022 году OpenAI представила миру нейросеть ChatGPT, которая ведёт осмысленный диалог, хранит в себе базу знаний и способна держать контекст беседы на протяжении длительного времени



Огласите весь список, пожалуйста

Спектр решаемых задач с помощью машинного обучения растёт с каждым днём, и до сих пор человечество находит необычные задачи для его применения

- Прогноз спроса/выручки (Demand forecasting)
- Определение тональности текста (Sentimental analysis)
- Распознавание лиц (Face detection)
- Распознавание речи (Speech to Text)
- Диагностика болезней (Medical forecasting)
- Ранжирование Web-страниц (Page ranking)
- Обнаружение аномалий и фрода (Anomaly detection)
- Поиск похожих объектов

И много чего другого....

Формальная постановка задачи

Модельная задача

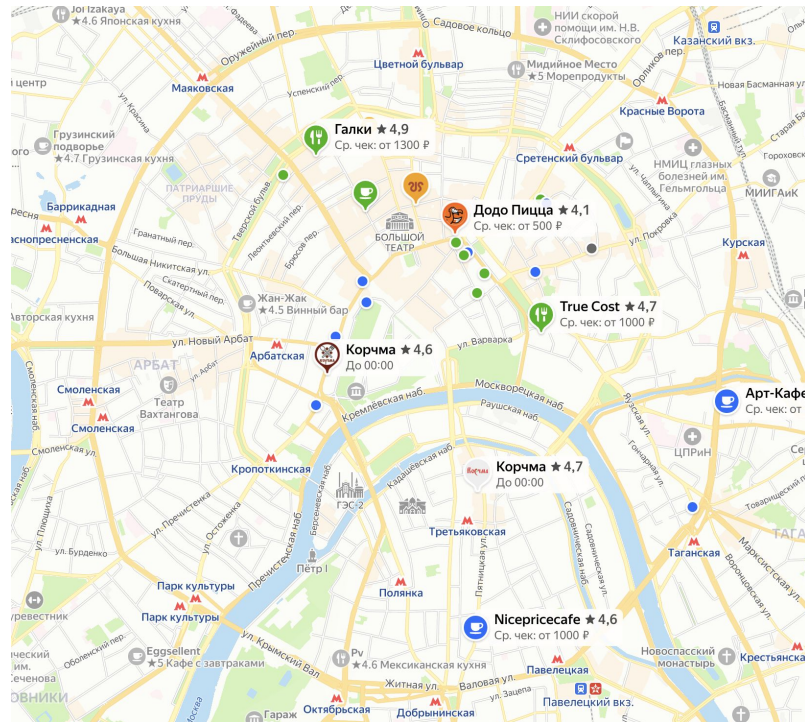
Попробуем с помощью машинного обучения найти оптимальное расположение открытия очередного заведения для некоторой сети кафе

Дано:

- Сеть кафе
- Хотим открыть ещё один
- Существует несколько вариантов размещения

Хотим ответить на вопрос:

- Какой из вариантов принесёт наибольшую прибыль?



Формализуем поставленную задачу

Обозначим

X - множество рассматриваемых **объектов**

Y - множество **ответов**, в данном случае это множество вещественных чисел R

$F: X \rightarrow Y$ неизвестная **зависимость** между местом открытия и полученной прибылью

Формализуем поставленную задачу

Поскольку у нас уже имеется открытая сеть кафе, то у нас есть выборка данных на основе, которой будет строиться алгоритм машинного обучения.

Дано:

$\{x_1, x_2, \dots, x_N\}, \subset \mathbf{X}$ - обучающая выборка размера N из существующих уже открытых заведений

$\{y_1, y_2, \dots, y_N\} \subset \mathbf{Y}$ - известные ответы (targets) для данной выборки

Найти:

$a : a(x_i) \approx y_i$ - искомая функция, которая для заданного описания кафе будет определять его прибыль

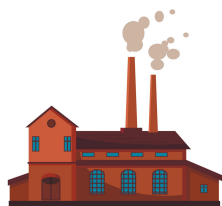
Признаковое описание

Поскольку единственный тип объект, с которым может работать компьютер, это число, то все объекты реального мира должны быть представлены в **числовом** формате

$f(x_i) = (f(x_{i1}), f(x_{i2}), \dots, f(x_{id}))$ - признаковое описание i -ого объекта



Objects



Feature
extraction



$(f(x_1), \dots, f(x_d))$

Feature vector

Матрица “объекты-признаки”

Признаковое описание объекта представляет собой вектор размерности d , где число d - количество используемых признаков (факторов) для описания объекта.

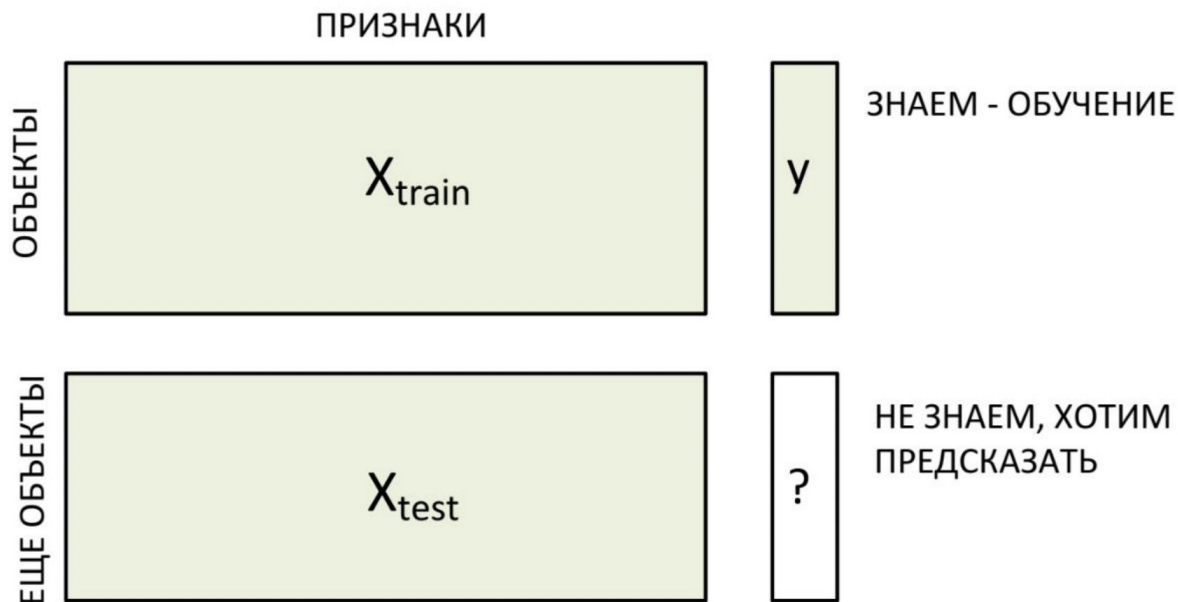
$$\begin{pmatrix} f(x_{11}) & f(x_{12}) & \dots & f(x_{1d}) \\ f(x_{21}) & f(x_{22}) & \dots & f(x_{2d}) \\ \cdot & \cdot & \cdot & \cdot \\ f(x_{N1}) & f(x_{N2}) & \dots & f(x_{Nd}) \end{pmatrix} \quad - \quad \begin{array}{l} \text{Все объекты с их} \\ \text{признаковым описанием} \\ \text{можно собрать в одну} \\ \text{матрицу “объекты-признаки”} \\ \text{размерности } N \times d \end{array}$$

Пример такой матрицы

плохой_клиент	линии	возраст	поведение_30-59_дней	Debt_Ratio	доход	число_кредитов
0	0.111673	46	0	1.329588	800.0	8
0	0.044097	69	0	0.535122	3800.0	10
0	0.047598	77	0	0.169610	3000.0	7
0	0.761149	58	1	2217.000000	NaN	4
0	0.690684	55	0	0.432552	12416.0	7

Разбиение множества на Train & Test

Те строки матрицы, для которых известны целевые переменные y используем в качестве обучения алгоритма, а для тех у кого неизвестны - для теста



Обучение с учителем

В случае если известны целевые переменные y для имеющихся прецедентов, то обучение состоит из двух этапов:

1. Этап обучения:

по имеющейся выборке $\{x_i, y_i\}$ строится алгоритм a

2. Этап тестирования:

полученный алгоритм применяют к тестовой выборке объектов $\{x_i\}$ и на ней оценивают итоговое качество полученного решения

Вернёмся к исходной модельной задаче

С учётом введенных обозначений:

- x_i – i -ый объект кафе
- $f(x_{i1}, x_{i2}, \dots, x_{id})$ – признаковое описание кафе.

В качестве признаков можно использовать различные количественные характеристики: удалённость от центра, наличие рядом метро, средняя цена квадратного метра в соседних домах и тд.

- y_i – какая была выручка у i -ого заведения за последний год

Поиск оптимального набора признаков представляет собой творческую задачу и зачастую для получения качественного результата требуются специальные доменные знания

Виды признаков

В зависимости от природы объекта встречаются следующие типы признаков:

- Числовые
- Бинарные
- Категориальные - принимают значения из неупорядоченного множества
- Ординальные - принимают значения из упорядоченного множества
- Признаки со сложной внутренней структурой. Например, изображение объекта

Виды данных

- Табличные (excel, csv, реляционные базы данных)
- Форматированные (json, yaml, xml)
- Текстовые
- Мультимедийные: изображение и видео
- Звуковые
- Логи

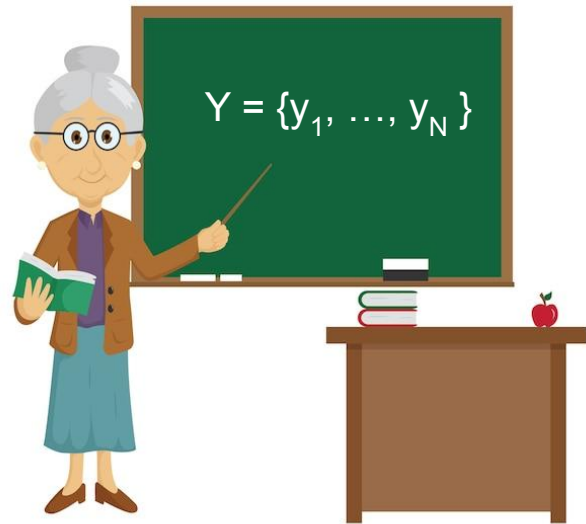
Большинство алгоритмов машинного обучения работает с числовыми данными, поэтому все виды данных необходимо переводить в числовые

Типы задач в зависимости от целевой переменной

В зависимости от наличия целевой переменной y и её природы используются различные подходы для решения задач.

В тех задачах машинного обучения, где **присутствует** переменная y , называются задачами обучения с **учителем**.

А там где переменная y **отсутствует** называются задачами обучения **без учителя**



Обучение с учителем. Классификация

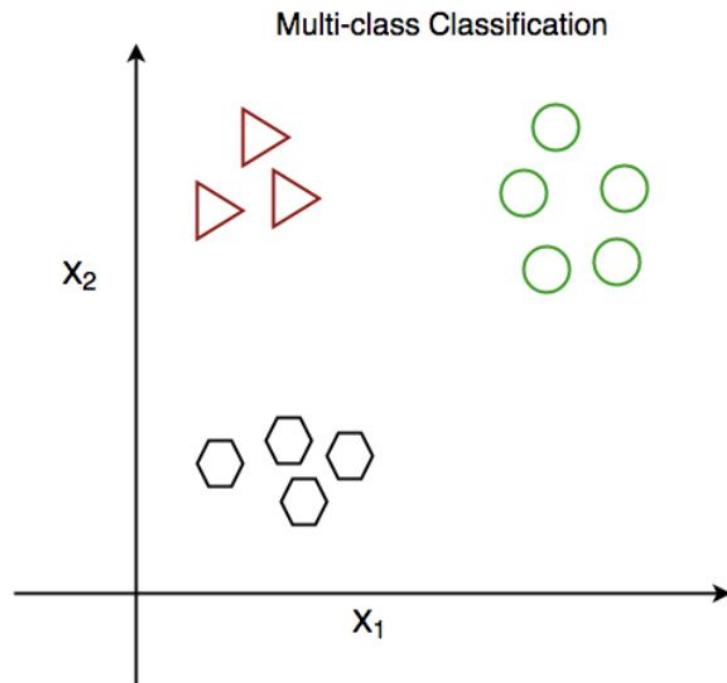
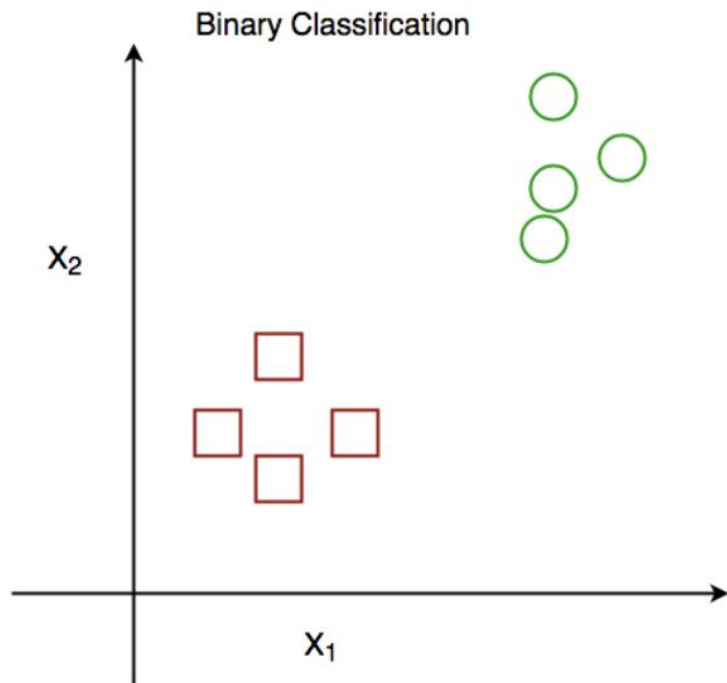
В случае если множество допустимых значений $y \subset \mathbb{N}$ - это натуральные числа, то решается задача классификации:

- Если $y = \{0, 1\}$ - бинарная классификация
- Если $y = \{1, \dots, M\}$ - многоклассовая классификация на M непересекающихся классов
- Если $y = \{0, 1\}^M$ - многоклассовая классификация с M пересекающимися классами. Например, объект с $Y = \{0, 1, 1, 0, 1\}$, относится ко 2, 3 и 5 классу

Примеры задач прикладных задач, которые сводятся к классификации

- Медицинская диагностика (Здоров/болен пациент) - бинарная классификация
- Будет ли выбран для данного клиента некоторый товар - бинарная классификация
- Тональность текста (негативный, позитивный или нейтральный) - непересекающаяся многоклассовая
- Тегирование объектов на карте (кафе, булочная, наличие веранды, pet-friendly) - многоклассовая с пересекающимися классами
- Классификация изображений - непересекающаяся многоклассовая

Геометрическое представление



Обучение с учителем. Регрессия

В случае если множество допустимых значений $y \subset \mathbb{R}$ - это вещественные числа, то решается задача регрессии:

- Если $y = \{y \subset \mathbb{R}\}$ - одномерная регрессия
- Если $y = \{y_1, \dots, y_M\}$, $y_i \subset \mathbb{R}$ - многомерная регрессия

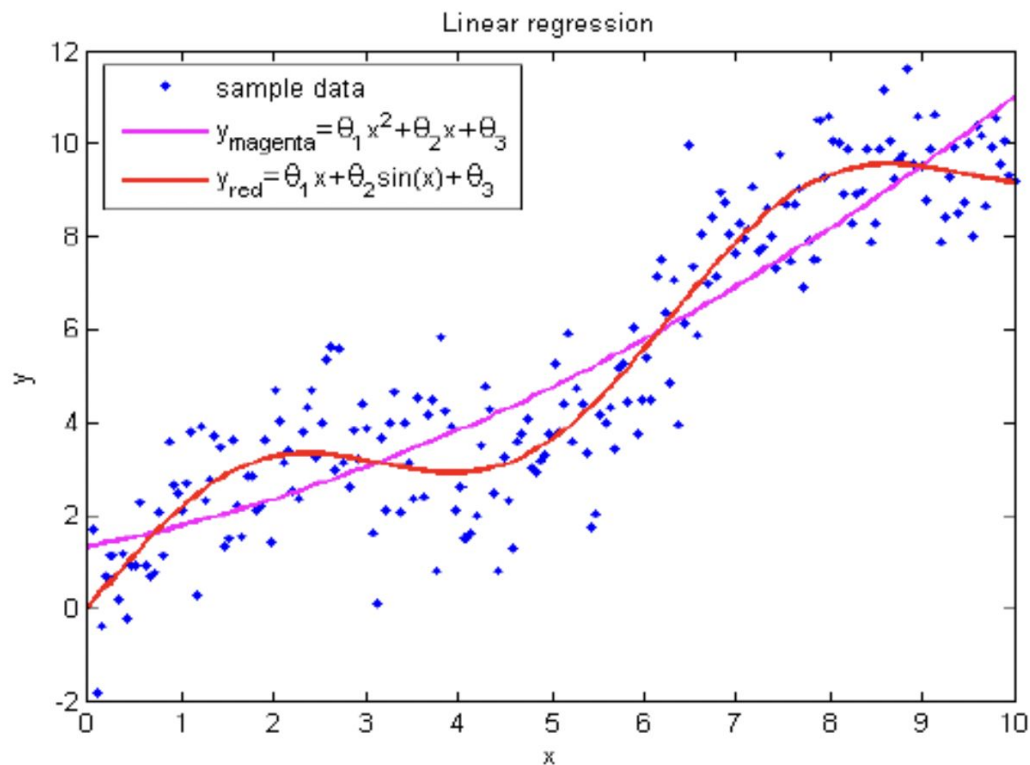
В некоторых случаях можно свести задачу многомерной регрессии к M одномерным отдельным задачам, в каждой из которых ищется оптимальное значение таргета

Примеры задач прикладных задач, которые сводятся к регрессии

- Предсказание стоимости недвижимости (стоимость квартиры в Москве)
- Предсказание прибыли ресторана
- Предсказание поведения временного ряда в будущем (стоимость акций)
- Предсказание зарплаты выпускника вуза по его оценкам

Геометрическое представление

$X = Y = \mathbb{R}$, $\ell = 200$, $n = 3$ признака: $\{x, x^2, 1\}$ или $\{x, \sin x, 1\}$



Обучение с учителем. Ранжирование

В случае если множество допустимых значений y - частично упорядоченное множество, и для i -ого объекта важно не конкретное значение y , а позиция относительно других объектов, то тогда решается задача ранжирования

Примеры задач:

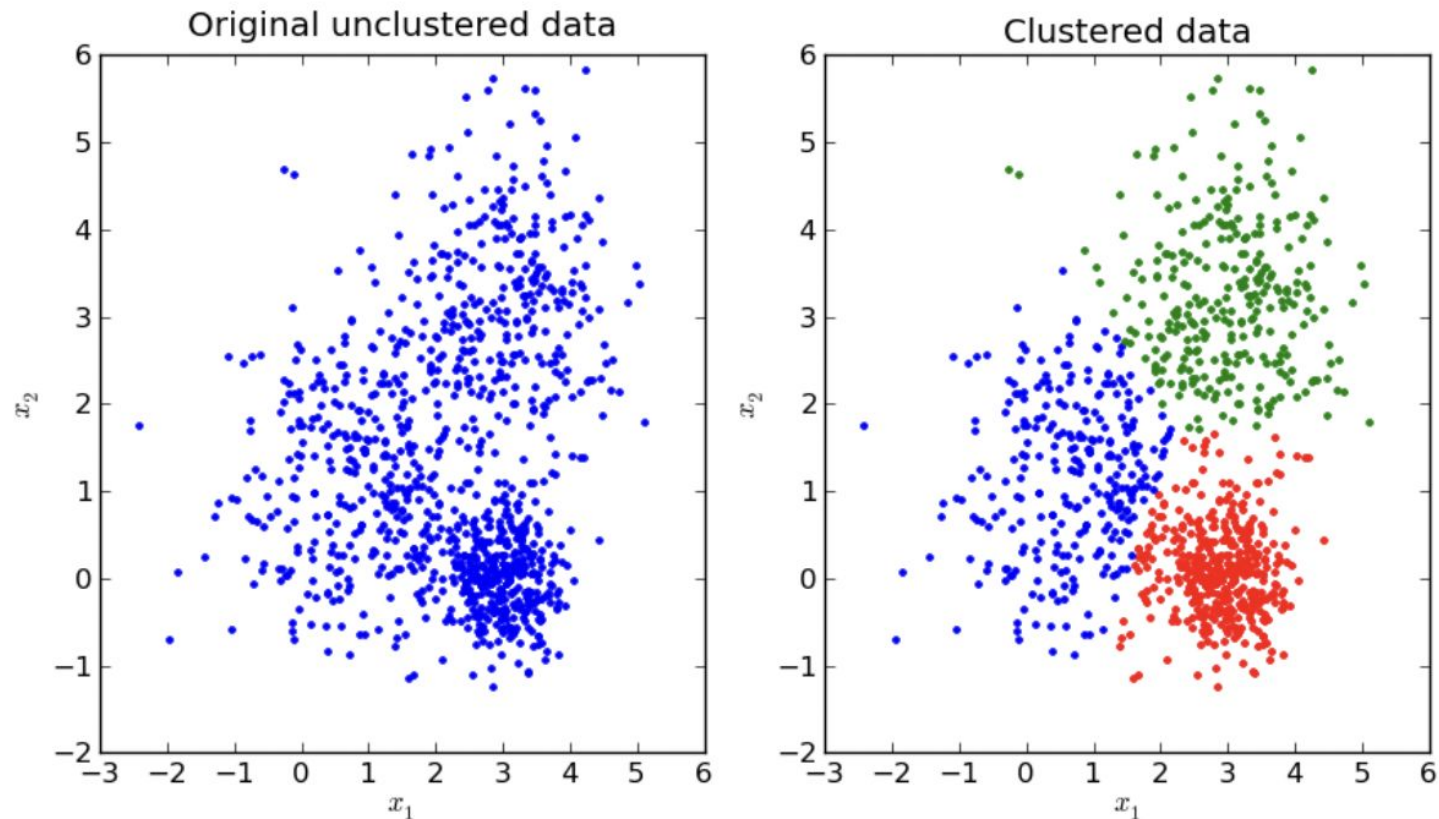
- Вывести подходящие запросу документы в порядке уменьшения релевантности
- Вывести кандидатов на должность в порядке уменьшения релевантности

Обучение без учителя. Кластеризация

В случае если **нет необходимости** делать прогноз переменной y или **её нет**, то происходит обучение **без учителя**. Как правило в таких задачах стоит цель найти закономерности в признаковом описании или произвести их визуализацию.

Одной из таких задач является задача **кластеризации**, где стоит необходимость разделения объектов на группы.

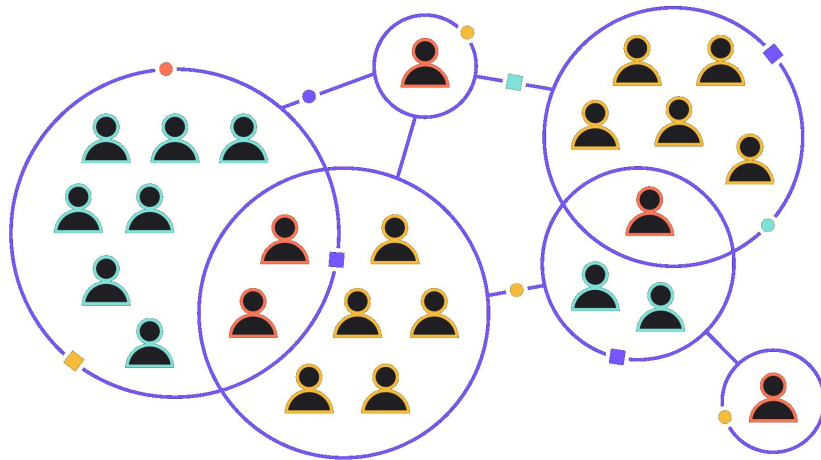
Пример работы кластеризации



Примеры задач

Как правило в формулировках таких задач употребляется ключевое слово “похожий”:

- Разбить пользователей на группы, внутри каждой из которых будут похожие пользователи
- Разбить текстовые документы на группы по схожести документов



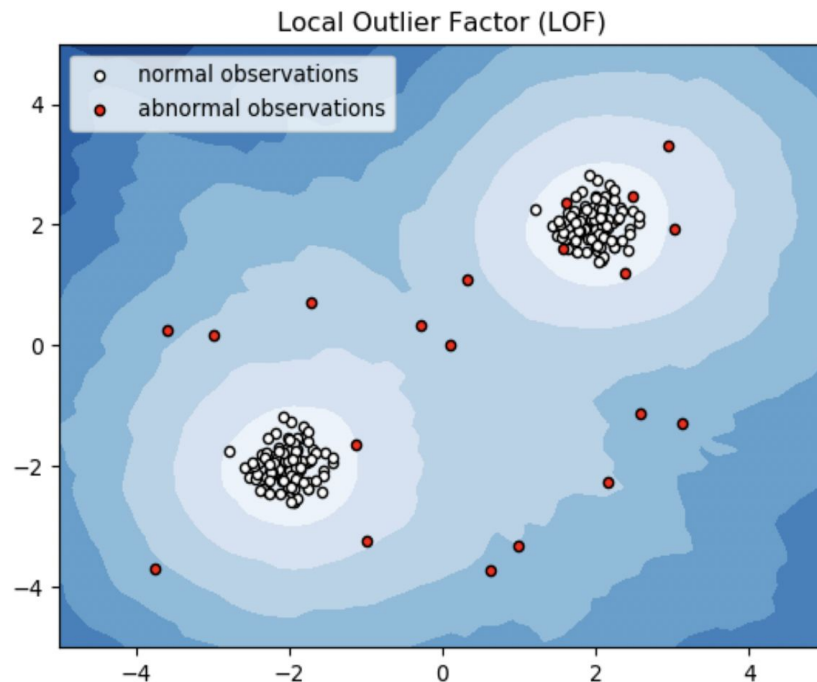
Обучение без учителя. Понижение размерности

Понижение размерности – задача генерации новых признаков (их число меньше, чем число старых), так, что с их помощью задача решается не хуже, чем с исходными.

Также понижение размерности может быть использовано для визуализации многомерных объектов на 2-D и 3-D графиках

Обучение без учителя. Оценивание плотности

Оценивание плотности – задача приближения распределения объектов.



Этапы обучения алгоритма. Оценка
предсказательной способности.

Пример задачи

Предположим, что мы хотим предсказать стоимость дома целевая переменная у по двум его признакам:

- x_1 - его площади
- x_2 - количество комнат



Выбор алгоритма

Исходя из природы данных и вида целевой переменной выбирается некоторое семейство алгоритмов A , среди, которых ищется наиболее оптимальный. То есть тот, на котором качество решения исходной задачи будет наилучшим.

$$A = \{a(x, w)\}_{w \in \mathbb{R}^k}$$

Семейство A состоит из алгоритмов одной природы, но с различными параметрами w

Пример семейства - линейные модели

Существует множество различных семейств алгоритмов, которые будут рассмотрены на дальнейших занятиях.

Сейчас же для наглядности рассмотрим семейство линейных моделей. В них предсказание представляет собой линейную функцию:

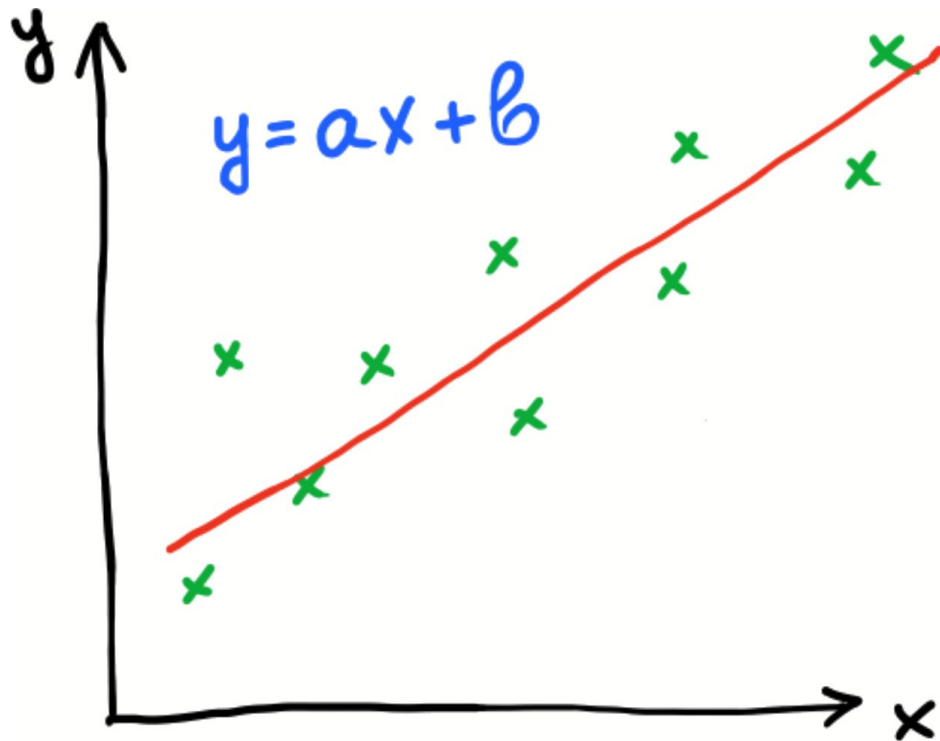
$$a(x, w) = w_0 + w_1 x_1 + \dots + w_d x_d$$

В данной задаче у линейной функции будет три слагаемых, так как имеется только два признака:

$$a(x, w) = w_0 + w_1 x_1 + w_2 x_2$$

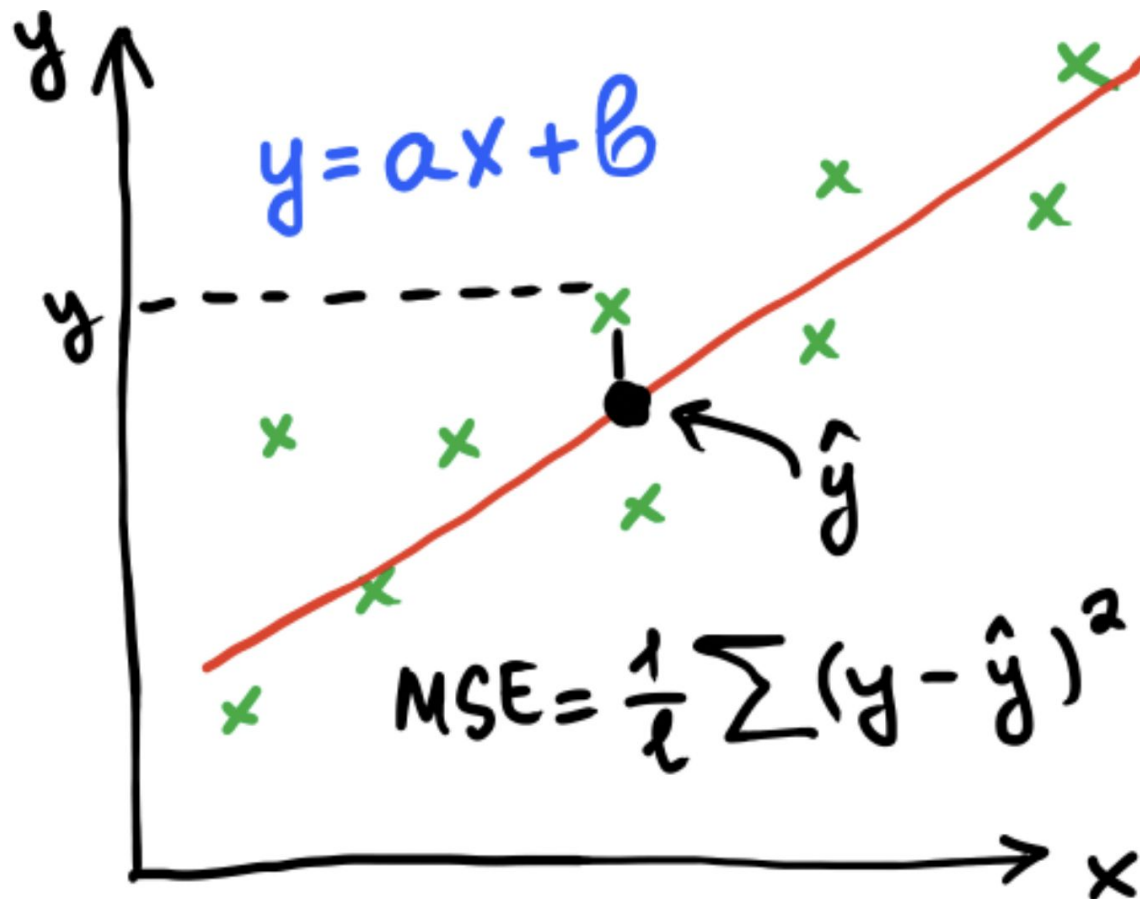
Визуализация линейной модели

Исходя из вида линейной модели нетрудно увидеть, что оно задаёт некоторую гиперплоскость в $d+1$ мерном пространстве. В случае $d=1$ - это будет прямая.



Как понять, что алгоритм работает хорошо?

Оценка качества работы



Функционал ошибки

Как измерить ошибку алгоритма на всех объектах выборки?

Функционал ошибки – функция, измеряющая качество работы алгоритма на всех объектах обучающей выборки

Пример - среднеквадратичная ошибка (MSE):

$$Q(a, X) = \frac{1}{N} \sum_{i=1}^N (a(x_i) - y_i)^2$$

$a(x_i)$ - ответ алгоритма на i -ом объекте

y_i - значение целевой переменной для i -го объекта

Обучение = оптимизация функционала ошибки

При обучении алгоритма ищется такой, набор параметров алгоритма $\{w_0, w_1, \dots, w_d\}$, на котором функционал Q достигает своего минимума:

$$Q(a, X) = \frac{1}{N} \sum_{i=1}^N (a(x_i) - y_i)^2 \rightarrow \min_{w_0, \dots, w_d}$$

Вспомним, что конкретный алгоритм семейства A определяется набором своих параметров $\{w_0, w_1, \dots, w_d\}$

Функционал ошибки для предсказания стоимости дома линейной моделью

Вспомним, что $a(x, w)$ для данной задачи определяется формулой:

$$a(x, w) = w_0 + w_1 x_1 + w_2 x_2$$

Подставим её в функционал ошибки:

$$Q(a, X) = \frac{1}{N} \sum_{i=1}^N (w_0 + w_1 x_1 + w_2 x_2 - y_i)^2 \rightarrow \min_{w_0, w_1, w_2}$$

Таким образом параметры w подбираются таким образом, чтобы на них достигался минимум функционал ошибки

Ещё раз об обучении

Процесс поиска оптимального алгоритма (оптимального набора параметров или весов) называется **обучением**.

Метрики качества

После того как прошел процесс обучения и был найден оптимальный с точки зрения функционала ошибки алгоритм необходимо провести его оценку и сравнить его качество работы с другими алгоритмами.

Абсолютная ошибка (MAE):

$$Q(a, X) = \frac{1}{N} \sum_{i=1}^N |a(x_i) - y_i|$$

Доля правильных ответов :

$$accuracy(a, X) = \frac{1}{N} \sum_{i=1}^N [a(x_i) == y_i]$$

Шаги решения задачи машинного обучения

1. Постановка задачи

Шаги решения задачи машинного обучения

1. Постановка задачи
2. Выделение множества признаков

Шаги решения задачи машинного обучения

1. Постановка задачи
2. Выделение множества признаков
3. Формирование обучающей и тестовой выборок

Шаги решения задачи машинного обучения

1. Постановка задачи
2. Выделение множества признаков
3. Формирование обучающей и тестовой выборок

Шаги решения задачи машинного обучения

1. Постановка задачи
2. Выделение множества признаков
3. Формирование обучающей и тестовой выборок
4. **Выбор семейства алгоритмов**

Шаги решения задачи машинного обучения

1. Постановка задачи
2. Выделение множества признаков
3. Формирование обучающей и тестовой выборок
4. Выбор семейства алгоритмов
5. Предобработка данных

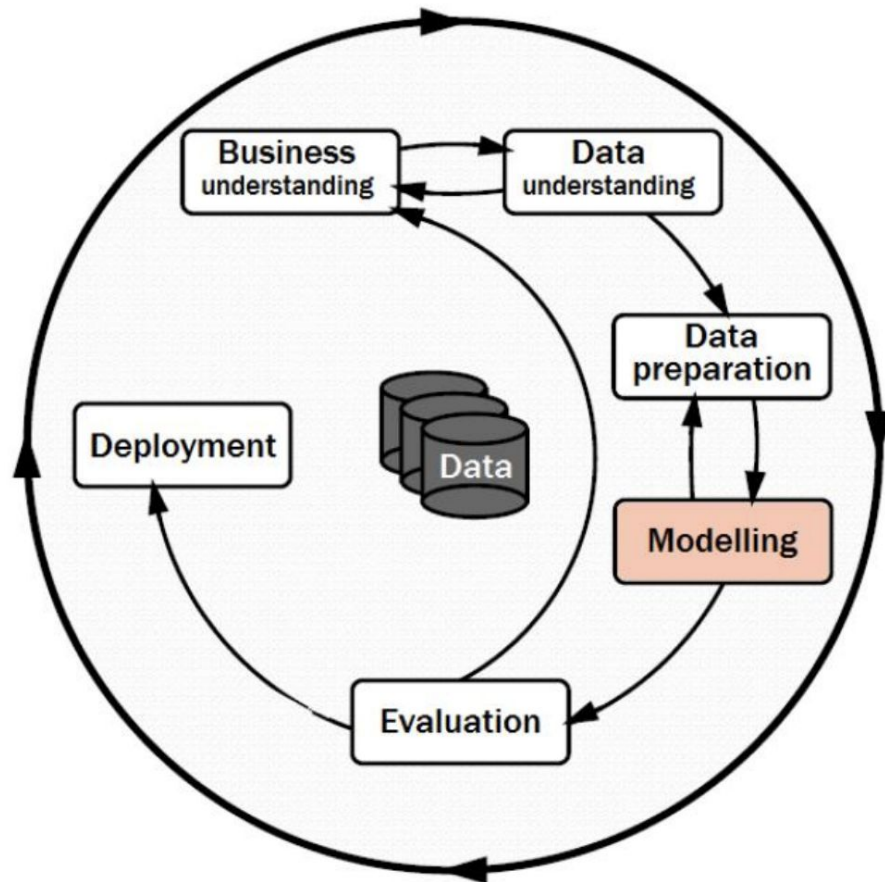
Шаги решения задачи машинного обучения

1. Постановка задачи
2. Выделение множества признаков
3. Формирование обучающей и тестовой выборок
4. Выбор семейства алгоритмов
5. Предобработка данных
6. Обучение

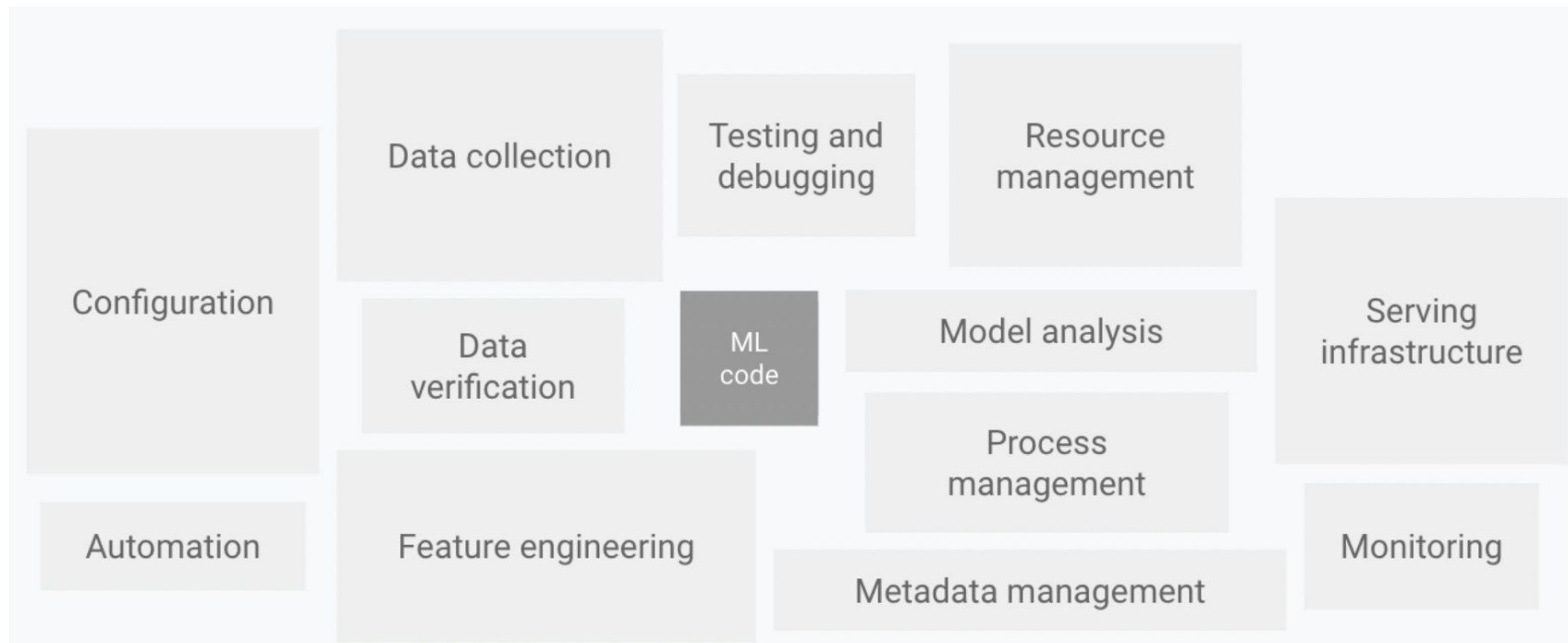
Шаги решения задачи машинного обучения

1. Постановка задачи
2. Выделение множества признаков
3. Формирование обучающей и тестовой выборок
4. Выбор семейства алгоритмов
5. Предобработка данных
6. Обучение
7. Оценка качества работы

Стадии разработки

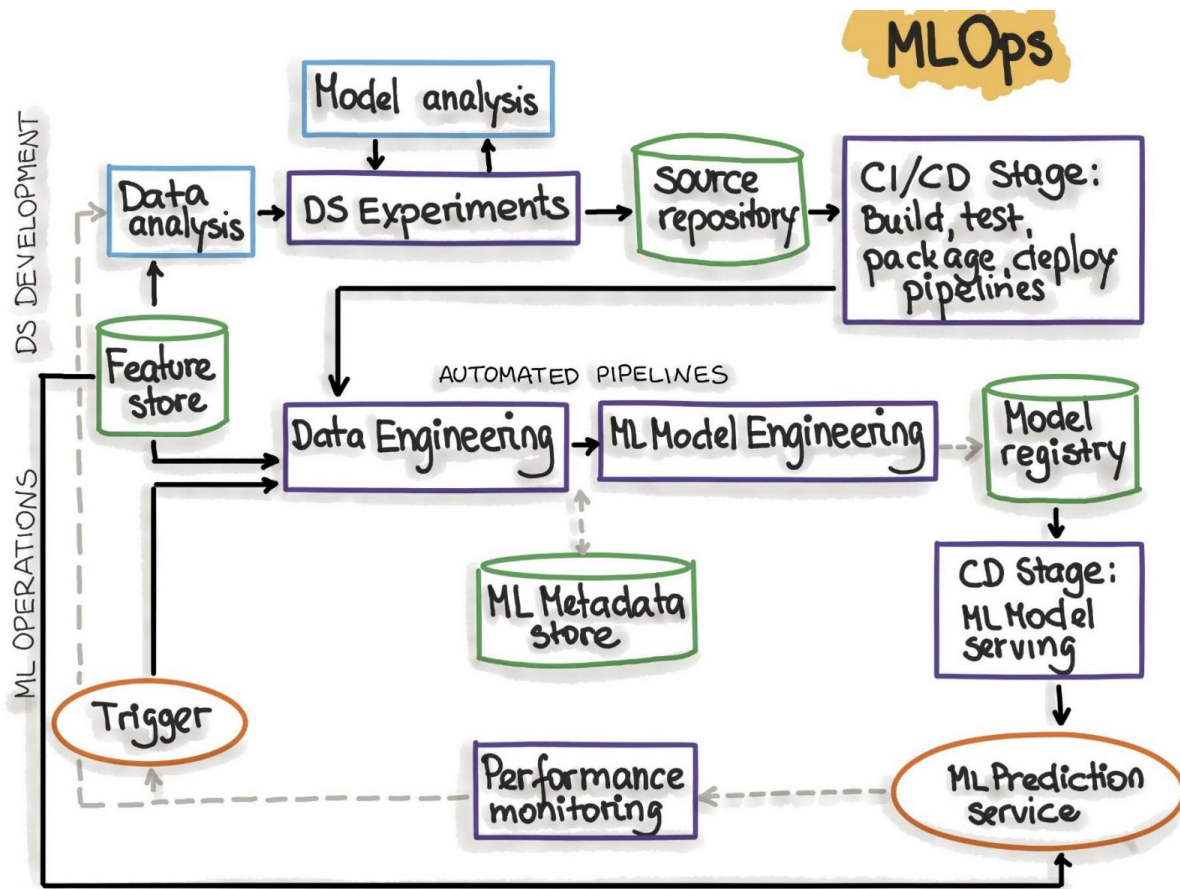


Подготовка модели состоит из множества компонент



Ошибки на каждом из этих этапов приводят к ухудшению качества работы

Инфраструктурные особенности внедряемых решений



Технологическая база курса



Язык программирования Python

<https://www.python.org/>



**Библиотека для матричных вычислений и
линейной алгебры**

<http://www.numpy.org/>



Библиотека для научных вычислений

<https://www.scipy.org/>



Библиотека для визуализации

<https://matplotlib.org/>

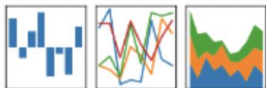


Библиотека для машинного обучения

<http://scikit-learn.org/>

pandas

$$y_{it} = \beta^t x_{it} + \mu_i + \epsilon_{it}$$



Библиотека для обработки данных

<https://pandas.pydata.org/>

Теоретическая база курса

- Линейная алгебра
- Математический анализ
- Теория вероятностей и математическая статистика