

Факультет
математики
и компьютерных
наук
СПбГУ

Машинное обучение 1

Лекция 1. Введение: устройство курса, правила. Основные термины, постановки задач и примеры применения машинного обучения

3 сентября 2021

Первый data scientist



Карл Фридрих Гаусс

математик, механик, физик, астроном и геодезист

В 24 года предсказал, где искать малую планету *Цереру*, скрывшуюся за Солнцем

In [1]:

```
%html
<style>
img.rounded {
    object-fit: cover;
    border-radius: 50%;
    height: 250px;
    width: 250px;
}
</style>
```

Команда курса



Александр
Авдюшенко
МКН, ШАД (CS
центр)

организация
курса
часть лекций
практики в группе
19.Б10-мкн



Екатерина
Гашенко
выпускница МКН
и CS центра,
Nokia

практики в группе
19.Б09-мкн
проверка ДЗ



Петр Мостовский
МКН

часть лекций
практики в группе
19.Б05-мкн
проверка ДЗ

Команда курса



Иван Долгов
выпускник CS
центра, JetBrains

проверка ДЗ



Олег Кудашев
Яндекс, группа
беспилотников

часть лекций

проверка ДЗ



Дмитрий Павлов
МКН

часть лекций

проверка
космического ДЗ

О курсе

Аннотация

Объемный годовой курс, в котором подробно рассматриваются простейшие модели машинного обучения.

Такой подход позволяет осознать основные принципы данной области в целом. Знание этих принципов даст возможность самостоятельно понимать механизмы функционирования более сложных современных моделей, обнаруживать пути улучшения уже существующих алгоритмов, а также адаптировать методы машинного обучения для решения нестандартных задач.

Пререквизиты

- Математический анализ, годовой курс
- Алгебра, годовой курс
- Теория вероятностей
- Математическая статистика
- Программирование: базовое владение языком программирования Python (ввод-вывод, циклы, рекурсия, классы, функции)

Где понадобится

- Более узкоспециализированные курсы по машинному обучению
- Научная работа в области машинного обучения
- В собеседованиях на ML-секциях
- В работе на должности Data Scientist, Machine Learning Engineer, Machine Learning Researcher

Содержание первого семестра

- Постановки задач машинного обучения
- Линейные методы классификации и регрессии
- Логические методы классификации
- Метрические методы классификации и регрессии
- Метрики качества, обобщающая способность
- Методы отбора признаков
- Основы байесовских методов

Вы научитесь

- ориентироваться в обширной области машинного обучения
- понимать на базовом уровне принципы устройства различных методов
- формализовывать задачи на языке обучения моделей
- понимать преимущества и недостатки моделей
- реализовывать базовые модели на языке программирования Python
- изучать использования таких библиотек языка программирования Python, как numpy, matplotlib и pandas
- бороться с проблемами, которые возникают при обучении моделей

Домашние задания

№ п.п.	Тема	Тип	Старт	Финиш	Баллы	Оценочное время выполнения
1	numpy, pandas	практика	3 сентября	18 сентября	14	10 часов
2	Логические алгоритмы классификации	теория	17 сентября	2 октября	10	8 часов
3	Решающие деревья	практика	17 сентября	9 октября	14	10 часов
4	Градиентный спуск своими руками	практика	24 сентября	12 октября	14	10 часов
5	Соревнование по классификации	соревна	1 октября	13 ноября	24	20 часов
6	Космическое ДЗ	практика	15 октября	4 ноября	24	20 часов
7	kNN, SVM, оценка качества	теория	22 октября	17 ноября	12	10 часов
8	Нейронная сеть на numpy	практика	19 ноября	11 декабря	24	16 часов

Экзамен в форме собеседования по машинному обучению — 24 балла

Ещё бонусные пятиминутки в начале лекции — можно получить по 1 баллу за каждую!

Критерии оценок

Итого максимально можно набрать 160 баллов = 90 (практические) + 22 (теоретические) + 24 (соревнование) + 24 (экзамен)

- «зачет» — не менее 81 балла
- 82 .. «хорошо» .. 111
- «отлично» — не менее 112

Другие курсы по машинному обучению



Воронцов Константин
Викторович
Профессор РАН

Доктор физико-
математических наук

Руководитель
лаборатории машинного
интеллекта МФТИ

Преподаватель [Школы
анализа данных \(ШАД\)](#)
(<https://yandexdataschool.ru/education>)

Курс машинного
обучения в ШАД и МФТИ
самый близкий к нашему
лекции Воронцова на
[youtube](#)
([https://www.youtube.com/playlist?
list=PLJOzdkh8T5krxc4HsHbB8g8f0hu797](https://www.youtube.com/playlist?list=PLJOzdkh8T5krxc4HsHbB8g8f0hu797))



Соколов Евгений Андреевич
Научный руководитель
Центра непрерывного
образования ВШЭ

Академический
руководитель
образовательной
программы Прикладная
математика и
информатика

Старший преподаватель
факультета
компьютерных наук ВШЭ

Курс машинного
обучения ФКН ВШЭ
несколько аналогичных
домашних заданий
курс Соколова на [github](#)
([https://github.com/esokolov/ml-
course-hse](https://github.com/esokolov/ml-course-hse))



Кашницкий Юрий
Senior Machine Learning
Scientist

NLP practitioner
Ph.D. in applied math

Курс машинного
обучения от Open Data
Science

страница курса
(<https://mlcourse.ai/>)



Andrew Ng
American computer
scientist and technology
entrepreneur focusing on
ML

One of Coursera founders

Adjunct professor at
Stanford University

страница курса на
Coursera
(<https://coursera.org/learn/machine-learning>)

Материал из Википедии — свободной энциклопедии

Машинное обучение (англ. machine learning, ML) — класс методов искусственного интеллекта, характерной чертой которых является не прямое решение задачи, а обучение за счёт применения решений множества сходных задач.

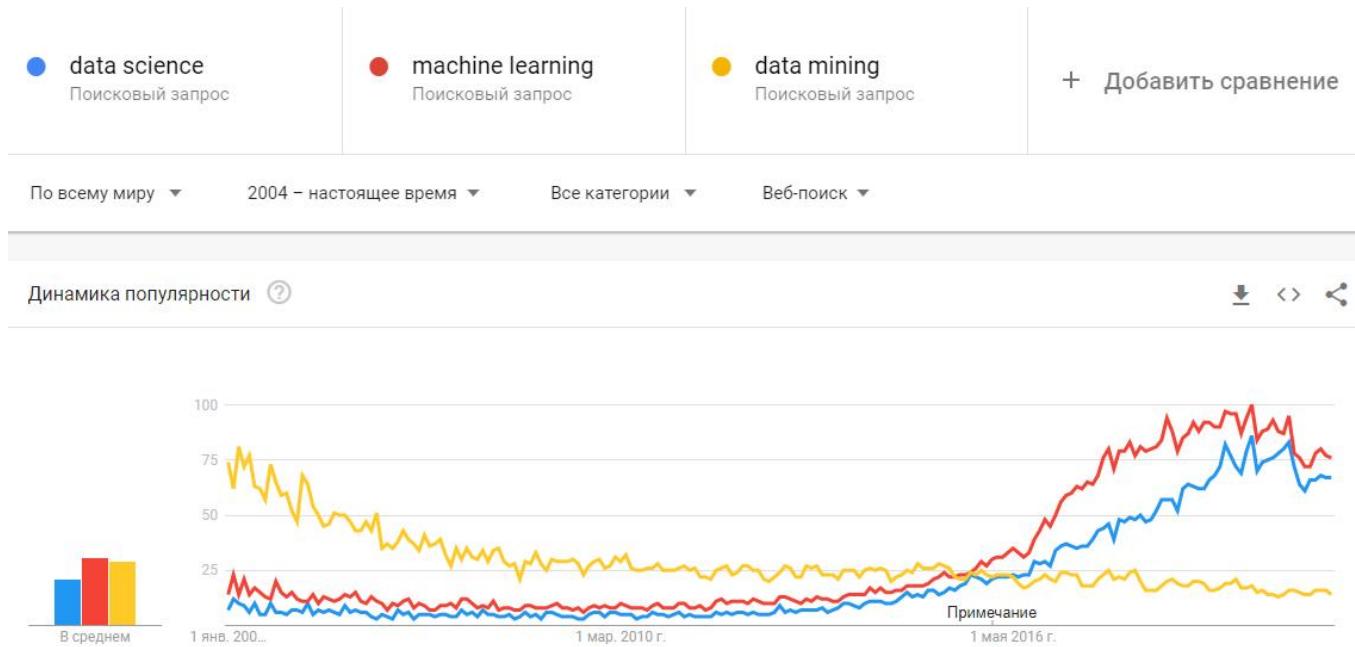
Для построения таких методов используются средства математической статистики, численных методов, математического анализа, методов оптимизации, теории вероятностей, теории графов, различные техники работы с данными в цифровой форме.

Что это?

$$\begin{cases} \frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_i)}{\partial x_i} = 0 \\ \frac{\partial(\rho u_i)}{\partial t} + \frac{\partial[\rho u_i u_j]}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j} + \rho f_i \end{cases}$$

В машинном обучении нет предзаданной модели с уравнениями...

- Различают два типа обучения:
 - Обучение по прецедентам (обучение с учителем), или индуктивное обучение, основано на выявлении эмпирических закономерностей в данных.
 - Дедуктивное обучение предполагает формализацию знаний экспертов и их перенос в компьютер в виде базы знаний.
- Дедуктивное обучение принято относить к области *экспертных систем*, поэтому *машинное обучение* ~ *обучение по прецедентам*.
- Многие методы машинного обучения разрабатывались как альтернатива классическим статистическим подходам. Многие методы тесно связаны с извлечением информации (англ. information extraction, information retrieval), интеллектуальным анализом данных (data mining).



Основные обозначения

X — множество объектов

Y — множество ответов

$y : X \rightarrow Y$ — неизвестная зависимость (target function)

Задача по обучающей выборке (training sample) $\{x_1, \dots, x_\ell\} \subset X$

с известными ответами $y_i = y(x_i)$

найти

$a : X \rightarrow Y$ — алгоритм,

решающую функцию (decision function), приближающую y на всём множестве X

Весь курс машинного обучения про это:

- как задаются объекты x_i и какими могут быть ответы y_i
- в каком смысле « a приближает y »
- как строить функцию a

Объекты и их признаки (features)

$$f_j : X \rightarrow D_j$$

Вектор $(f_1(x), \dots, f_n(x))$ — признаковое описание объекта x

Типы признаков:

- $D_j = \{0, 1\}$ — бинарный
- $\#|D_j| < \infty$ — категориальный (номинальный)
- $\#|D_j| < \infty$, D_j упорядочено — ординальный (порядковый)
- $D_j = \mathbb{R}$ — вещественный (количественный)

Матрица «объекты-признаки» (feature data)

$$F = ||f_j(x_i)||_{\ell \times n} = \begin{bmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{bmatrix}$$

Вопрос 1: Как перевести все признаки в бинарные?

In [1]:

```
import pandas as pd
from sklearn.datasets import load_boston

boston_house_prices = load_boston()

X = pd.DataFrame(data=boston_house_prices.data,
                  columns=boston_house_prices.feature_names)
Y = pd.DataFrame(data=boston_house_prices.target,
                  columns=[ "target"])

pd.concat([X, Y], axis=1).head()
```

Out[1]:

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90



In [2]:

```
print(boston_house_prices.DESCR)
```

```
.. _boston_dataset:

Boston house prices dataset
-----

**Data Set Characteristics:** 

:Number of Instances: 506

:Number of Attributes: 13 numeric/categorical predictive. Median Value
(attribute 14) is usually the target.

:Attribute Information (in order):
 - CRIM    per capita crime rate by town
 - ZN      proportion of residential land zoned for lots over 25,000 sq.ft.
 - INDUS   proportion of non-retail business acres per town
 - CHAS    Charles River dummy variable (= 1 if tract bounds river; 0 otherwise)
 - NOX     nitric oxides concentration (parts per 10 million)
 - RM      average number of rooms per dwelling
 - AGE     proportion of owner-occupied units built prior to 1940
 - DIS     weighted distances to five Boston employment centres
 - RAD     index of accessibility to radial highways
 - TAX     full-value property-tax rate per $10,000
 - PTRATIO pupil-teacher ratio by town
 - B       1000(Bk - 0.63)^2 where Bk is the proportion of blacks by town
 - LSTAT   % lower status of the population
 - MEDV    Median value of owner-occupied homes in $1000's
```

:Missing Attribute Values: None

:Creator: Harrison, D. and Rubinfeld, D.L.

This is a copy of UCI ML housing dataset.
<https://archive.ics.uci.edu/ml/machine-learning-databases/housing/>

This dataset was taken from the StatLib library which is maintained at Carnegie Mellon University.

The Boston house-price data of Harrison, D. and Rubinfeld, D.L. 'Hedonic prices and the demand for clean air', J. Environ. Economics & Management, vol.5, 81-102, 1978. Used in Belsley, Kuh & Welsch, 'Regression diagnostics ...', Wiley, 1980. N.B. Various transformations are used in the table on pages 244-261 of the latter.

The Boston house-price data has been used in many machine learning papers that address regression problems.

.. topic:: References

- Belsley, Kuh & Welsch, 'Regression diagnostics: Identifying Influential Data and Sources of Collinearity', Wiley, 1980. 244-261.
- Quinlan,R. (1993). Combining Instance-Based and Model-Based Learning. In Proceedings on the Tenth International Conference of Machine Learning, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann.

Типы задач

Классификация (classification)

- $Y = \{-1, +1\}$ — бинарная классификация
- $Y = \{1, \dots, M\}$ — многоклассовая классификация
- $Y = \{0, 1\}^M$ — многоклассовая с пересекающимися классами

Регрессия (regression)

- $Y = \mathbb{R}$ или $Y = \mathbb{R}^m$

Ранжирование (ranking)

- Y — конечное упорядоченное множество

Предсказательная модель

Модель (predictive model) — параметрическое семейство функций

$$A = \{g(x, \theta) | \theta \in \Theta\},$$

где $g : X \times \Theta \rightarrow Y$ — фиксированная функция, Θ — множество допустимых значений параметра θ

Например

Линейная модель с вектором параметров $\theta = (\theta_1, \dots, \theta_n)$, $\Theta = \mathbb{R}^n$:

$$g(x, \theta) = \sum_{j=1}^n \theta_j f_j(x) \text{ — для регрессии и ранжирования, } Y = \mathbb{R}$$

$$g(x, \theta) = \text{sign} \sum_{j=1}^n \theta_j f_j(x) \text{ — для классификации, } Y = \{-1, +1\}$$

Пример: задача регрессии, синтетические данные

$$X = Y = \mathbb{R},$$

$$l = 50,$$

$$n = 3 \text{ признака: } \{1, x, x^2\} \text{ или } \{1, x, \sin x\}$$

In [3]:

```
import numpy as np

np.random.seed(0)
l = 50

x = np.linspace(0, 30, num=l)
Y = x + 4*np.sin(x) + 3*np.random.randn(l)

X_1 = np.vstack([np.ones_like(x), x, x**2]).T
X_2 = np.vstack([np.ones_like(x), x, np.sin(x)]).T
```

In [4]:

```
import matplotlib.pyplot as plt

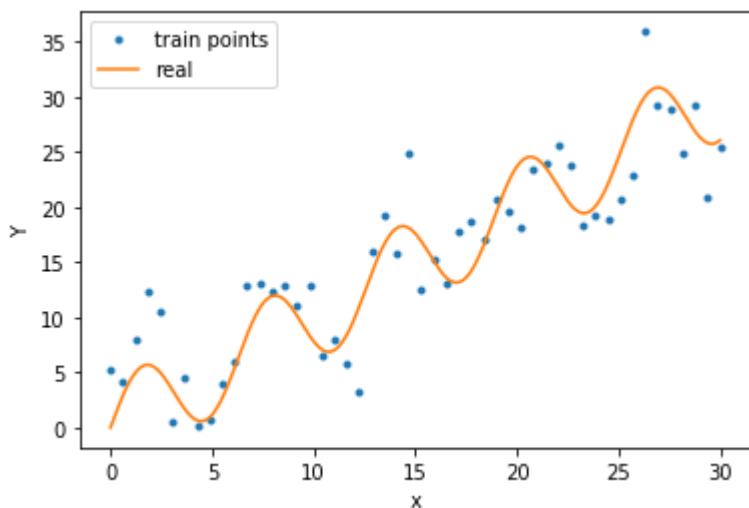
_, ax = plt.subplots()

ax.set_xlabel('x')
ax.set_ylabel('Y')

ax.plot(x, Y, '.', label='train points')

x_plot = np.linspace(0, 30, num=1000)
ax.plot(x_plot, x_plot + 4*np.sin(x_plot), label='real')

plt.legend(loc='best')
plt.show()
```



In [5]:

```
from sklearn.linear_model import LinearRegression

reg_1 = LinearRegression(fit_intercept=False)
reg_1.fit(X_1, Y)

reg_2 = LinearRegression(fit_intercept=False)
reg_2.fit(X_2, Y)
```

Out[5]:

```
LinearRegression(copy_X=True, fit_intercept=False, n_jobs=None, normalize=False)
```

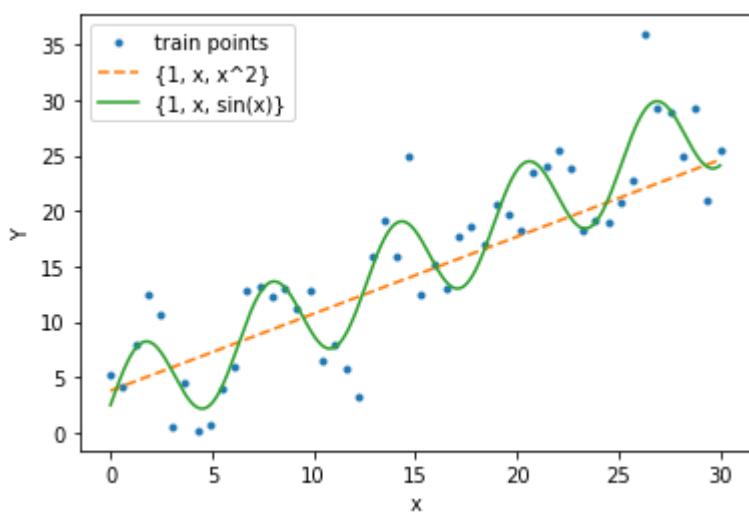
Вопрос 2: ЧТО такое intercept?

In [6]:

```
_, ax = plt.subplots()
ax.set_xlabel('x')
ax.set_ylabel('Y')
ax.plot(x, Y, '.', label='train points')

x_plot = np.linspace(0, 30, num=1000)
X_plot = np.vstack([np.ones_like(x_plot), x_plot, np.sin(x_plot)]).T
ax.plot(x_plot, reg_1.predict(X_plot), label='{1, x, x^2}', linestyle='dashed')
ax.plot(x_plot, reg_2.predict(X_plot), label='{1, x, sin(x)})')

plt.legend(loc='best')
plt.show()
```



Этап обучения (train)

метод μ по выборке $(X, Y) = (x_i, y_i)_{i=1}^\ell$ строит алгоритм $a = \mu(X, Y)$

$$\begin{bmatrix} f_1(x_1) & \dots & f_n(x_1) \\ \dots & \dots & \dots \\ f_1(x_\ell) & \dots & f_n(x_\ell) \end{bmatrix} \xrightarrow{y} \begin{bmatrix} y_1 \\ \dots \\ y_\ell \end{bmatrix} \xrightarrow{\mu} a$$

Этап применения (test)

алгоритм a для новых объектов x'_i выдаёт ответы $a(x'_i)$

Функционалы качества

$\mathcal{L}(a, x)$ — функция потерь (loss function). Величина ошибки алгоритма $a \in A$ на объекте $x \in X$.

Функции потерь для задач классификации:

- $\mathcal{L}(a, x) = [a(x) \neq y(x)]$ — индикатор ошибки

Функции потерь для задач регрессии:

- $\mathcal{L}(a, x) = |a(x) - y(x)|$ — абсолютное значение ошибки
- $\mathcal{L}(a, x) = (a(x) - y(x))^2$ — квадратичная ошибка

Эмпирический риск — функционал качества алгоритма a на X^ℓ :

$$Q(a, X^\ell) = \frac{1}{\ell} \sum_{i=1}^{\ell} \mathcal{L}(a, x_i)$$

Сведение задачи обучения к задаче оптимизации

Метод минимизации эмпирического риска

$$\mu(X^\ell) = \arg \min_{a \in A} Q(a, X^\ell)$$

Пример: метод наименьших квадратов ($Y = \mathbb{R}$, \mathcal{L} квадратична)

$$\mu(X^\ell) = \arg \min_{\theta} \sum_{i=1}^{\ell} (g(x_i, \theta) - y_i)^2$$

Проблема обобщающей способности

- Найдём ли мы «закон природы» или переобучимся, то есть подгоним функцию $g(x_i, \theta)$ под заданные точки?
- Будет ли $a = \mu(X^\ell)$ приближать функцию y на всём X ?
- Будет ли $Q(a, X^k)$ мало на новых данных — контрольной выборке $X^k = (x'_i, y'_i)_{i=1}^k$, $y'_i = y(x_i)$?

Пример переобучения

Зависимость $y(x) = \frac{1}{1+25x^2}$ на отрезке $x \in [-2, 2]$

Признаковое описание $x \rightarrow (1, x, x^2, \dots, x^n)$

Модель полиномиальной регрессии:

$a(x, \theta) = \theta_0 + \theta_1 x + \dots + \theta_n x^n$ — полином степени n

Обучение методом наименьших квадратов:

$$Q(\theta, X^\ell) = \sum_{i=1}^{\ell} (\theta_0 + \theta_1 x_i + \dots + \theta_n x_i^n - y_i)^2 \rightarrow \min_{\theta_0, \dots, \theta_n}$$

Обучающая выборка: $X^\ell = \{x_i = 4^{\frac{i-1}{\ell-1}} - 2 | i = 1, \dots, \ell\}$

Контрольная выборка: $X^k = \{x_i = 4^{\frac{i-0.5}{\ell-1}} - 2 | i = 1, \dots, \ell - 1\}$

Что происходит с $Q(\theta, X^\ell)$ и $Q(\theta, X^k)$ при увеличении n ?

In [5]:

```
import numpy as np

np.random.seed(0)
l = 50

x = np.linspace(-2, 2, num=l)
Y = 1 / (1 + 25*x**2)
```

In [6]:

```
import matplotlib.pyplot as plt

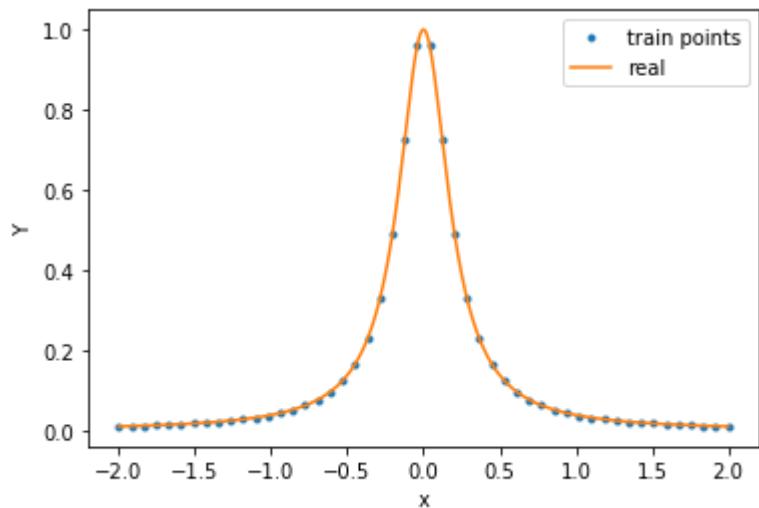
_, ax = plt.subplots()

ax.set_xlabel('x')
ax.set_ylabel('Y')

ax.plot(x, Y, '.', label='train points')

x_plot = np.linspace(-2, 2, num=1000)
y_plot = 1 / (1 + 25*x_plot**2)
ax.plot(x_plot, y_plot, label='real')

plt.legend(loc='best')
plt.show()
```



In [7]:

```
from sklearn.linear_model import LinearRegression

def X(x, n):
    res = [np.ones_like(x)]
    for i in range(1, n):
        res.append(x**i)
    return np.vstack(res).T
```

In [8]:

```
x_l = np.array([4*(i - 1)/(l - 1) - 2 for i in range(1, l + 1)])
x_k = np.array([4*(i - 0.5)/(l - 1) - 2 for i in range(1, l + 1)])
lin_reg = LinearRegression(fit_intercept=False)

train_score, test_score = [], []

ns = range(2, 40)
for n in ns:
    X_l = X(x_l, n)
    Y_l = 1 / (1 + 25*X_l**2)
    X_k = X(x_k, n)
    Y_k = 1 / (1 + 25*X_k**2)
    lin_reg.fit(X_l, Y_l)
    train_score.append(np.mean((lin_reg.predict(X_l) - Y_l)**2))
    test_score.append(np.mean((lin_reg.predict(X_k) - Y_k)**2))
```

In [9]:

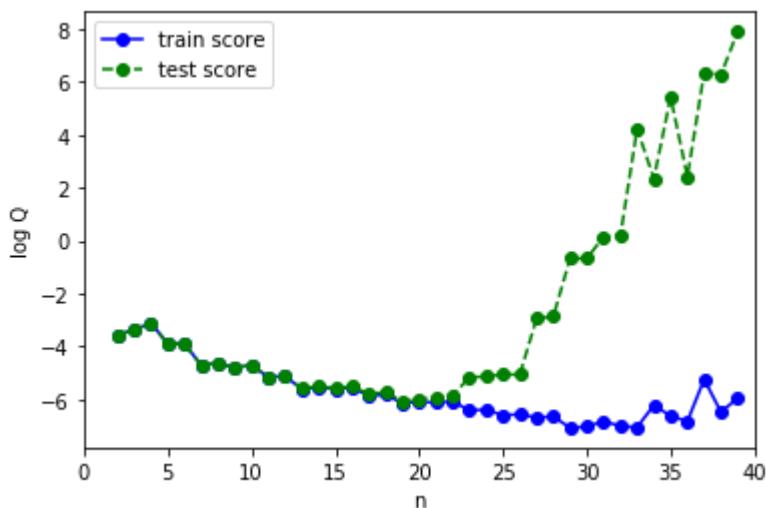
```
import matplotlib.pyplot as plt

_, ax = plt.subplots()

ax.set_xlabel('n')
ax.set_xlim(0, len(ns) + 2)
ax.set_ylabel('log Q')

ax.plot(ns, np.log(train_score), 'bo-', label='train score')
ax.plot(ns, np.log(test_score), 'go--', label='test score')

plt.legend(loc='best')
plt.show()
```



Вопрос 3: Почему такие графики? Как будет меняться точка расхождения n в зависимости от размера обучающей выборки ℓ и почему?

Переобучение — одна из главных проблем в машинном обучении

Когда `test_score >> train_score`

- Из-за чего возникает переобучение?
 - избыточная сложность пространства параметров Θ , лишние степени свободы в модели $g(x, \theta)$ «тратятся» на чрезмерно точную подгонку под обучающую выборку
 - переобучение есть всегда, когда есть оптимизация параметров по конечной (заведомо неполной) выборке
- Как обнаружить переобучение?
 - эмпирически, путём разбиения выборки на train и test
- Нельзя избавиться от него совсем. Как минимизировать?
 - минимизировать ошибку на валидации (HoldOut, Leave One Out, Cross Validation), но осторожно!
 - накладывать ограничения на θ (регуляризация)
 - минимизировать одну из теоретических оценок

Эмпирические оценки обобщающей способности

- Эмпирический риск на тестовых данных (hold-out)

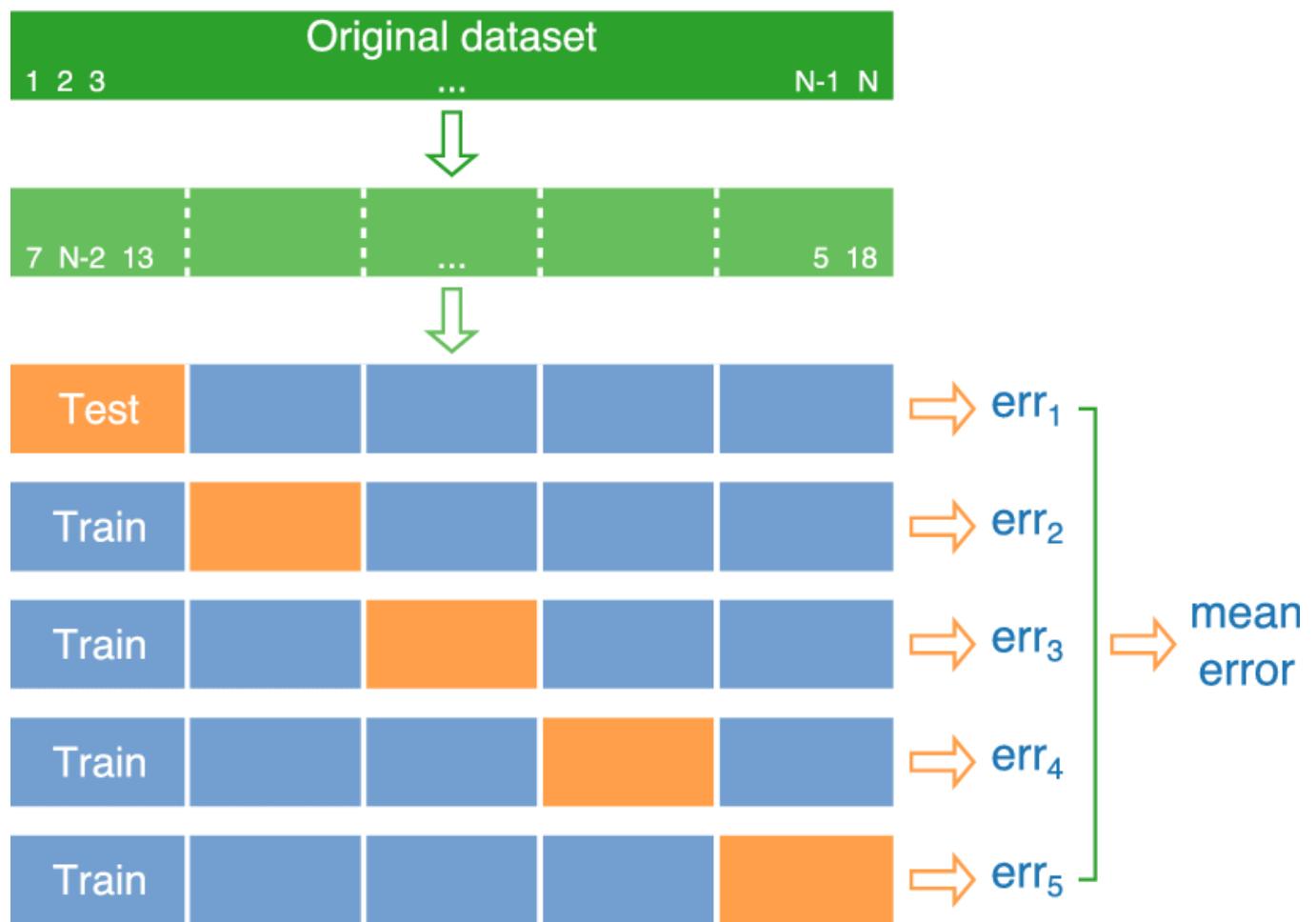
$$HO(\mu, X^\ell, X^k) = Q(\mu(X^\ell), X^k) \rightarrow \min$$

- Скользящий контроль (leave-one-out), $L = \ell + 1$

$$LOO(\mu, X^\ell) = \frac{1}{L} \sum_{i=1}^L \mathcal{L}(\mu(X^\ell \setminus \{x_i\}), x_i) \rightarrow \min$$

- Кросс-проверка (cross-validation), $L = \ell + k$, $X^L = X_n^\ell \cup X_n^k$:

$$CV(\mu, X^L) = \frac{1}{|N|} \sum_{n \in N} Q(\mu(X_n^\ell), X_n^k) \rightarrow \min$$



Знаковые события в машинном обучении

1997: IBM Deep Blue обыгрывает чемпиона мира по шахматам Гарри Каспарова

- 480 шахматных CPU
- перебор модификацией альфа-бета-отсечений
- две дебютные книги

2004: Соревнование беспилотных автомобилей: DARPA Grand Challenge

- призовой фонд \$1 млн
- в первом заезде победитель проехал 11.8 из 230 км

2006: Запуск Google Translate

- сначала статистический машинный перевод
- мобильное приложение появилось в 2010

2011: 40 лет развития DARPA CALO (Cognitive Assistant that Learns and Organizes)

- появление голосового помощника Apple Siri
- IBM Watson победил в телевизионной игре «Jeopardy!» (у нас «Своя игра»)

2011-2015: ImageNet: 25% -> 3.5% ошибок против 5% у людей

2015: Создание открытой компании OpenAI, Илон Маск и Сэм Альтман, обещают вложить \$1 млрд

2016: Google DeepMind обыграл чемпиона мира по игре Го

2018: На аукционе Christie's картина, формально нарисованная ИИ, продана за 432 500\$

2020: AlphaFold 2 предсказывает структуру белков с точностью выше 90% для примерно двух третей белков в датасете

Резюме

- Основные понятия машинного обучения:

обучение по прецедентам (с учителем), объекты, признаки, ответы, модель алгоритмов, метод обучения, эмпирический риск, переобучение

- Проблема переобучения: HoldOut, LeaveOneOut, CrossValidation
- Знаковые события в машинном обучении

Спасибо за внимание!