

# Основы машинного обучения

Лекция 1: Введение в машинное обучение

Екатерина Кондратьева

# Структура курса

## Основы МЛ:

1. Введение в Методы Машинного Обучения. Практикум по Python
2. Обучение с учителем: Линейная и логистическая регрессия. Ядра.
3. Обучение с учителем: Регуляризация в линейных моделях. Метод Ближайших Соседей (KNN)
4. Обучение с учителем: Метод опорных векторов (SVM) для задач классификации и регрессии.  
Kernel SVM
5. Обучение с учителем: Деревья решений (Decision Trees). Случайный лес (Random Forest).
6. Оценка качества алгоритмов машинного обучения. Кросс-валидация. Поиск аномалий и артефактов в выборке.
7. Обучение без учителя: кластеризация. Снижение размерности данных PCA.

# Структура курса

## Продвинутые методы МЛ:

8. Отбор и генерация признаков (Feature Engineering). Поиск и оптимизация модели (Grid Search).
  9. Стекинг, вотинг. Градиентный бустинг. Пакеты XGBoost/Catboost/LightGBM. Соревнования по анализу данных, обзор решений, статей и актуальных методов.
- + ***Самостоятельная работа*** (задача классификации/  
задача регрессии/  
данные без разметки)



# Как будет строиться занятие по курсу?

< 30 минут: рекап прошлой лекции, обсуждение домашнего задания

< 40 минут: лекция

< 60 минут: практика

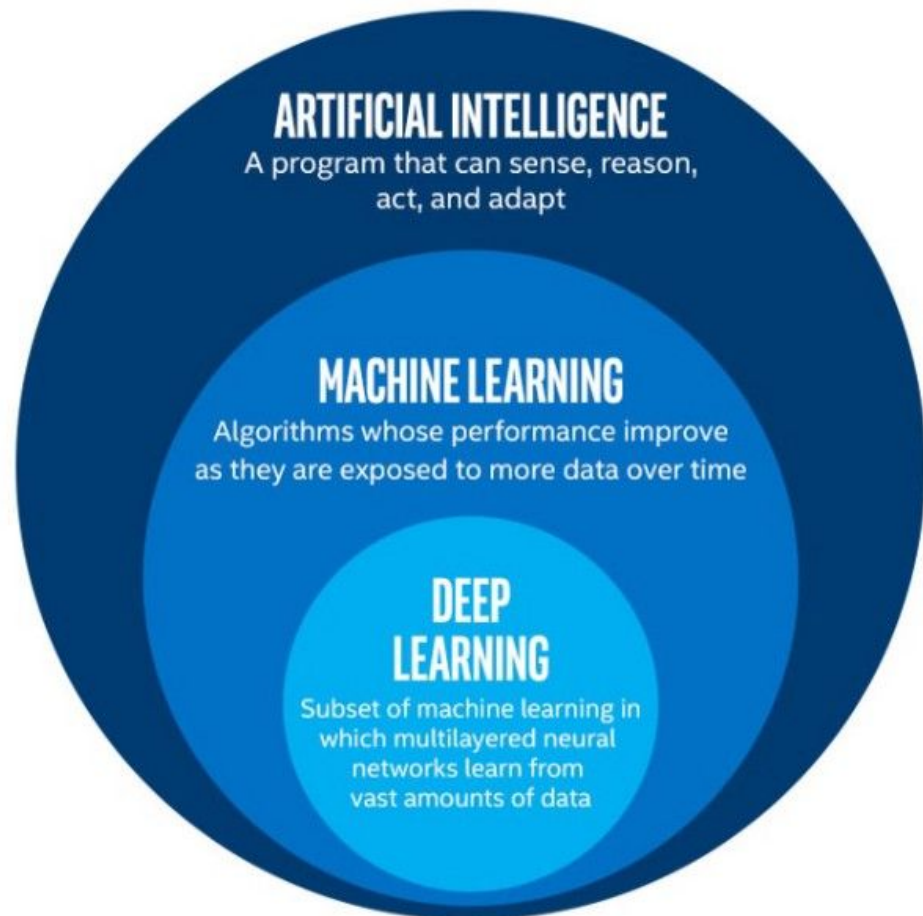
< 30 минут: вопросы, ответы, неконструктивные споры и обвинения

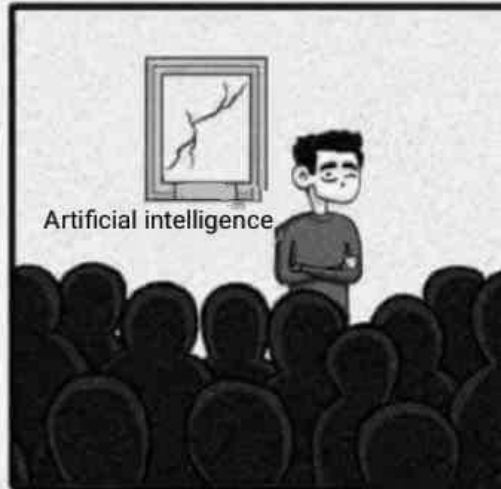
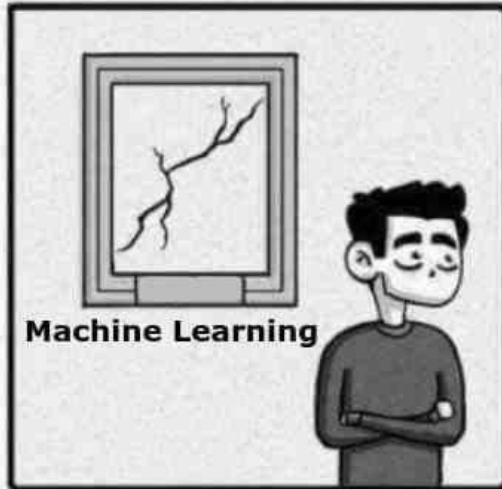
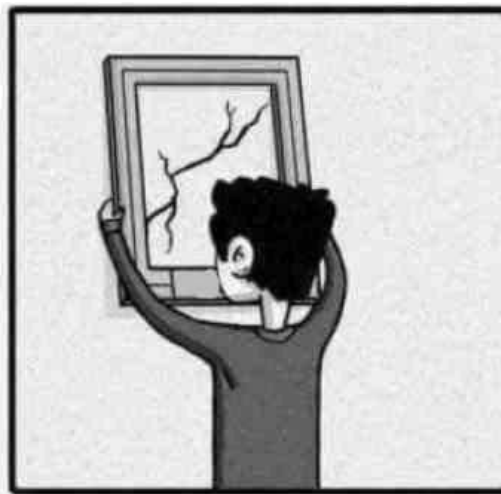
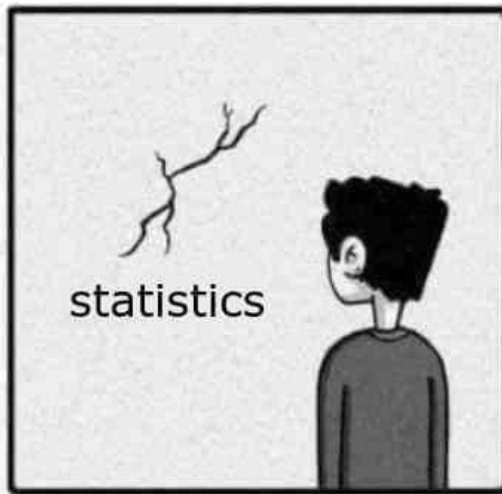
# Что нам понадобится на курсе:

- Python > 3.7, Jupyter Notebook:  
[https://repo.anaconda.com/archive/Anaconda3-2018.12-MacOSX-x86\\_64.pkg](https://repo.anaconda.com/archive/Anaconda3-2018.12-MacOSX-x86_64.pkg)
- Репозиторий группы  
<https://github.com/kondratevakate/machine-learning-with-love>
- Соревновательный дух, крепкий сон и здоровый аппетит

Что такое **машинное обучение**?

# Машинное обучение?







Как решить задачу с **ML**?

# 1. Тип задачи. Обучение с учителем

- Нужно предсказать число - задача **регрессии**

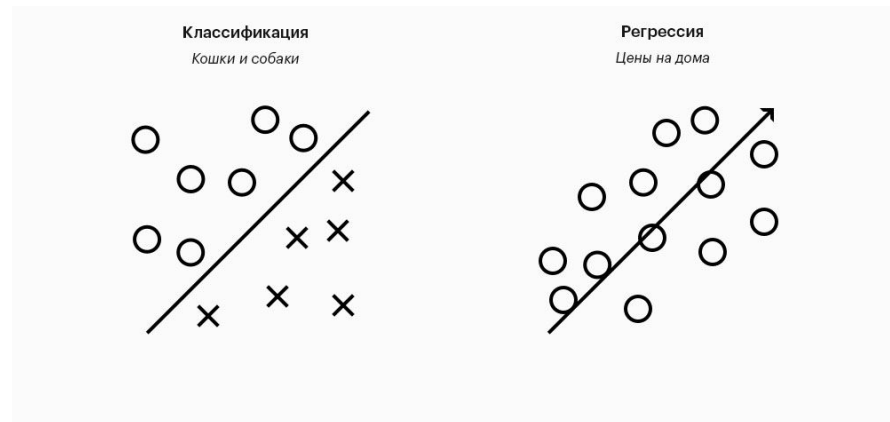
Например: определение возраста человека по фото

<https://arxiv.org/ftp/arxiv/papers/1709/1709.01664.pdf>

- Нужно предсказать класс - задача **классификации**

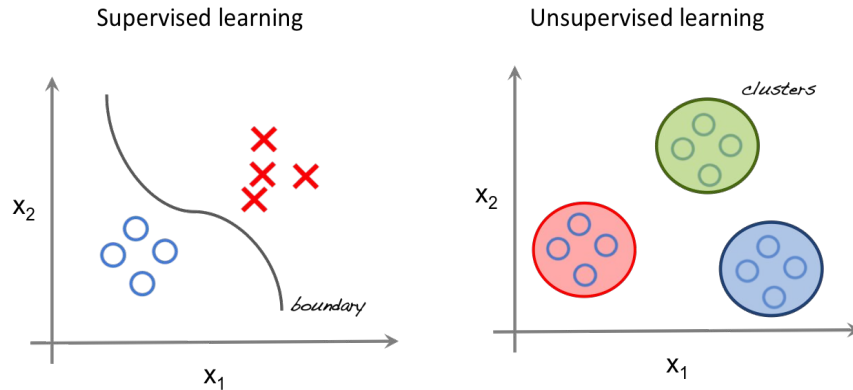
Например: распознавание букв или цифр

<https://github.com/rois-codh/kmnist>



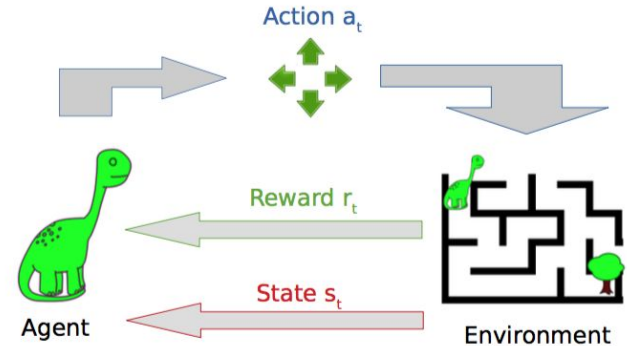
<https://awdee.ru/wp-content/uploads/2018/07/5.jpg>

# 1. Пример специальных задач. Обучение без учителя



## Кластеризация

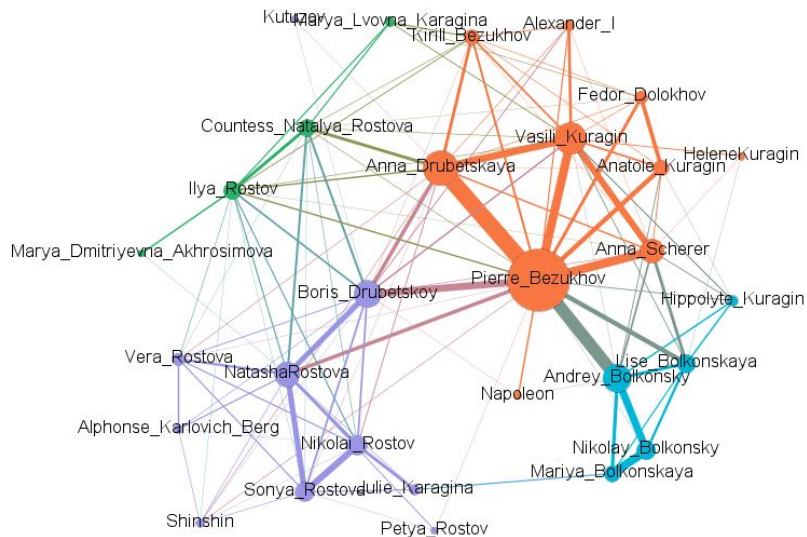
Например: кластеризация аудитории сайта



## Обучение с подкреплением

Например: обучение виртуального робота

# 1. Пример специальных задач



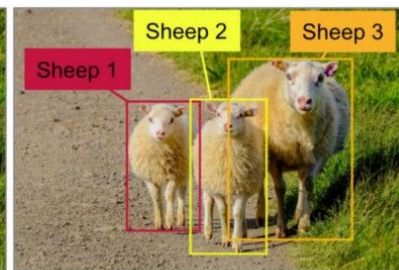
Анализ графов

Например: Лев Толстой и сетевой анализ

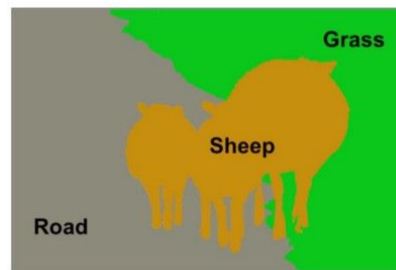
<https://cfi.hse.ru/data/2017/04/24/1168847268/fig2.png>



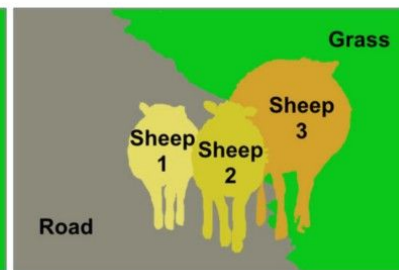
**Classification + Localization**



**Object Detection**



**Semantic Segmentation**



**Instance Segmentation**

Компьютерное зрение

Например: Сегментация

<https://towardsdatascience.com/detection-and-segmentation-through-convnets-47a442de27ea>



Что за задача?

## 2. Организация данных

	city	class	degree	income	city + degree
0	Moscow	A	1	10.2	Moscow + 1
1	London	B	1	11.6	London + 1
2	London	A	2	8.8	London + 2
3	Kiev	A	2	9.0	Kiev + 2
4	Moscow	B	3	6.6	Moscow + 3
5	Moscow	B	3	10.0	Moscow + 3
6	Kiev	A	1	9.0	Kiev + 1
7	Moscow	A	1	7.2	Moscow + 1

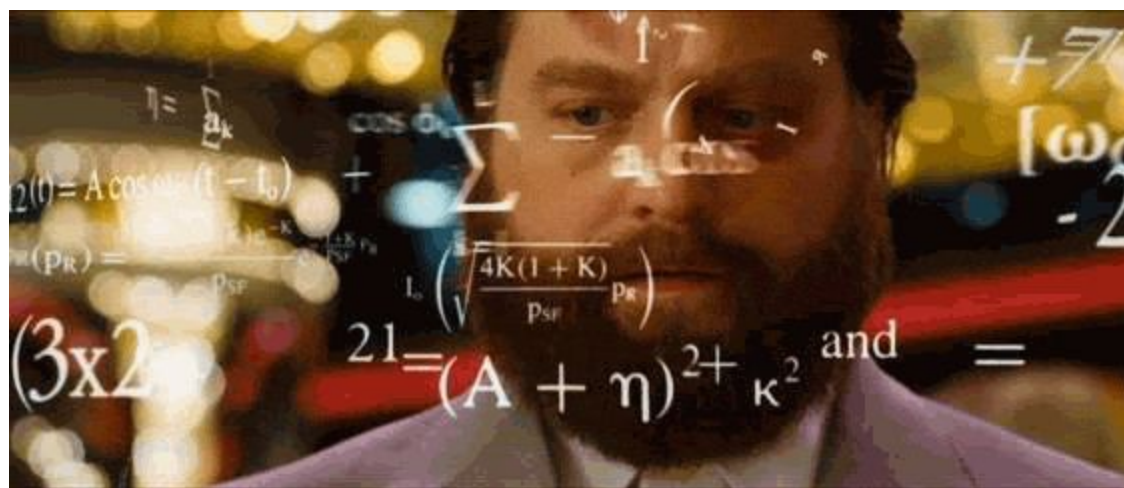
```
# конъюнкция двух признаков
def make_conj(data, feature1, feature2):
    data[feature1 + ' + ' + feature2] =
        data[feature1].astype(str) +
        ' + ' +
        data[feature2].astype(str)

    return (data)

# пример использования
make_conj(data, 'city', 'degree')
```

<https://alexanderdyakonov.files.wordpress.com/2016/08/pic021.png?w=700>

Часто текстовые данные приводят к формату таблиц (\*.csv)




## 2. Организация данных. Разметка данных

**Яндекс Толока**

**Простые задания  
за вознаграждения**

Присоединиться

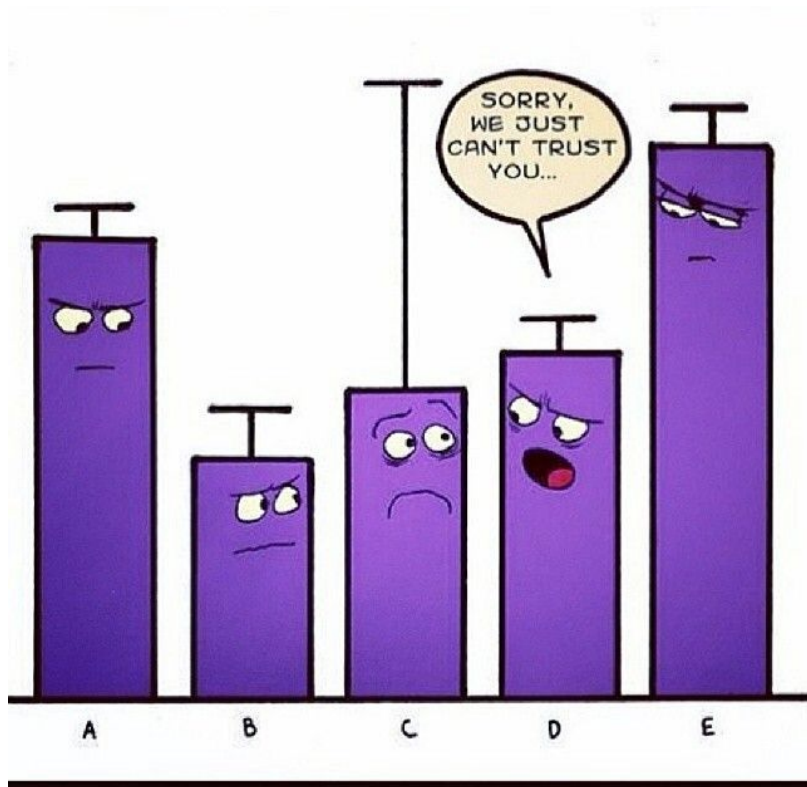


The illustration shows a woman with red hair tied in a bun, wearing a blue jacket over a white shirt. She is holding a black smartphone in her right hand and a red disposable cup in her left. Above her is a yellow map overlay with several blue circular callouts containing text: '0,9\$', '0,8\$', 'Я' (in a red circle), and '1,0\$ 7'. The background includes stylized clouds and green trees.

Если нужна разметка данных вручную



### 3. Метрика оценивания



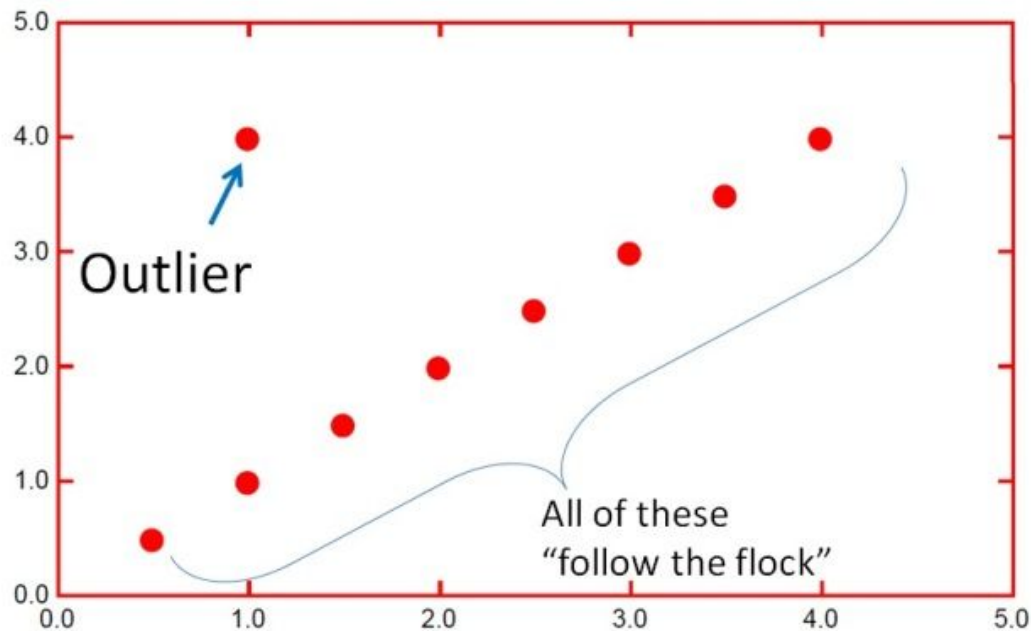
**Метрики оценивания модели:**

Точность %, Ошибки 1 или 2 рода, MSE

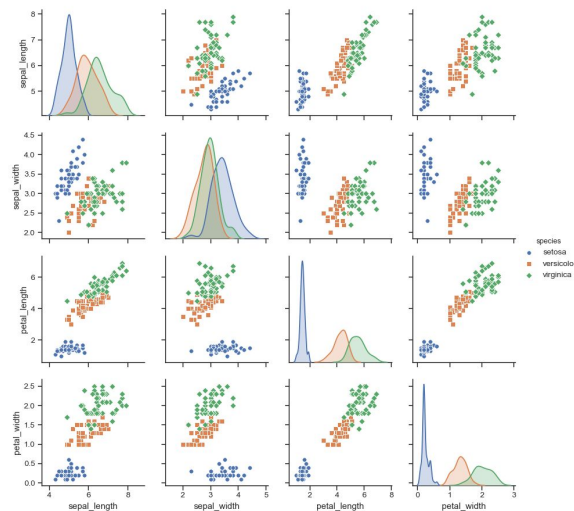
**Метрики оценивания в бизнесе:**

Деньги, отток

## 4. Предобработка данных



Never mind what the axes mean...



Изучение и визуализация данных.

Поиск артефактов.

## 4. Предобработка данных



[https://cs2.pikabu.ru/post\\_img2/2014/01/30/0/1391028003\\_254382516.jpg](https://cs2.pikabu.ru/post_img2/2014/01/30/0/1391028003_254382516.jpg)

Преобразованием категориальные признаки, заполняем пропуски

# 5. Выбор модели

## Канонические модели

baseline методы: LR, SVC, RFC, KNN

реализованы в *sklearn* <https://scikit-learn.org/>



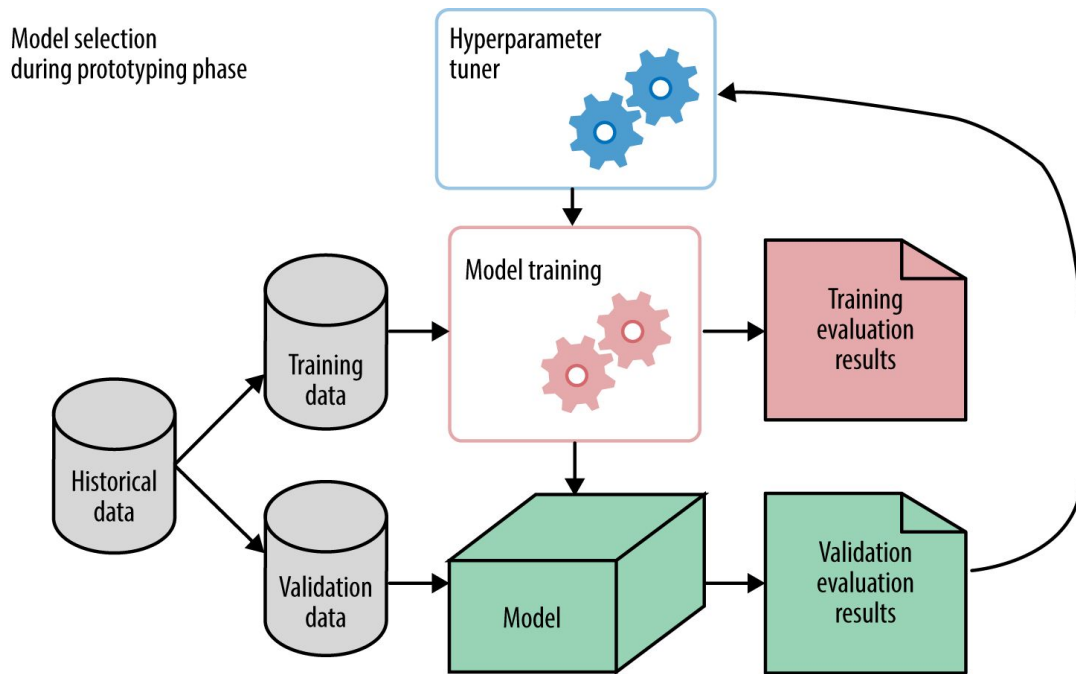
## Продвинутые методы:

state of the art: статьи с *NIPS* <https://nips.cc/>,

лучшие решения с *kaggle* <https://www.kaggle.com/>

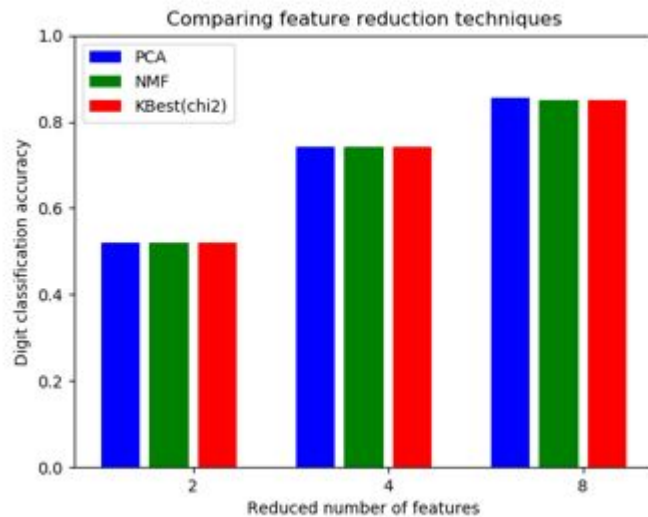


## 5. Оптимизация модели



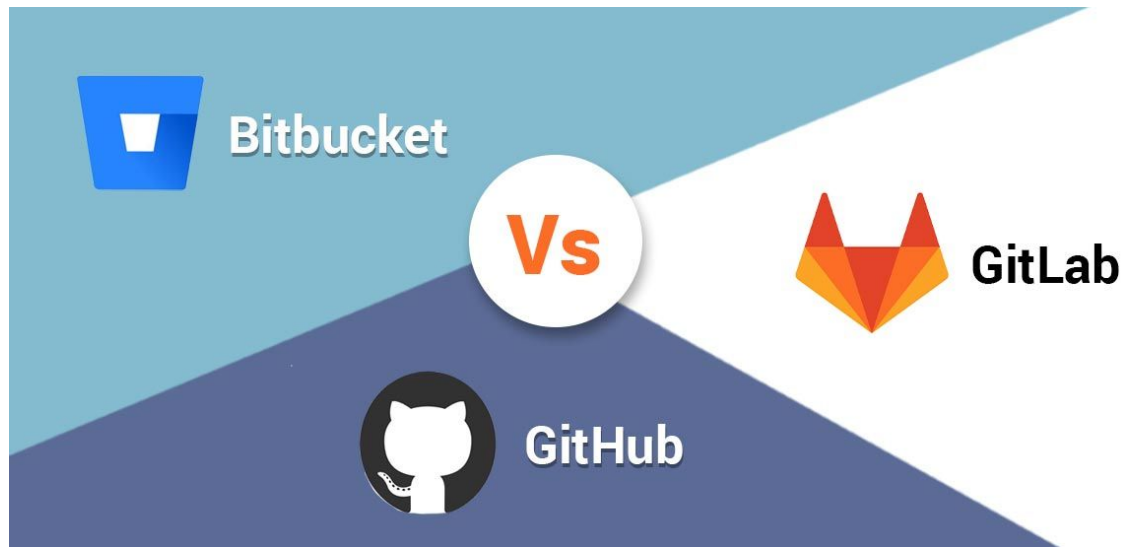
Оптимизация гипер параметров выбранной модели. Кросс Валидация

## 5. Оптимизация модели



Получаем предварительные результаты точности на кросс-валидации

## 6. Разработка и организация кода. Что такое **git**?



Система контроля версий: <https://en.wikipedia.org/wiki/Git>

## Floor is software development best practices





# Google collaboratory:

<https://colab.research.google.com/notebooks/welcome.ipynb>

## How can I search Collaboratory notebooks?

Use [Drive's](#) search box. Clicking on the Collaboratory logo at the top left of the notebook view will show all notebooks in Drive. You can also search for notebooks that you have opened recently using **File->Open Recent**.

## Where is my code executed? What happens to my execution state if I close the browser window?

Code is executed in a virtual machine dedicated to your account. Virtual machines are recycled when idle for a while, and have a maximum lifetime enforced by the system.

## How can I get my data out?

You can download any Collaboratory notebook that you've created from Google Drive following these [instructions](#), or from within Collaboratory's File menu. All Collaboratory notebooks are stored in the open source Jupyter notebook format ( .ipynb).

## How may I use GPUs and why are they sometimes unavailable?

Collaboratory is intended for interactive use. Long-running background computations, particularly on GPUs, may be stopped. Please do not use Collaboratory for [cryptocurrency mining](#). Doing so is unsupported and may result in service unavailability. We encourage users who wish to run continuous or long-running computations through Collaboratory's UI to use a [local runtime](#).

## How can I reset the virtual machine(s) my code runs on, and why is this sometimes unavailable?

The "Reset all runtimes" entry in the "Runtime" menu will return all managed virtual machines assigned to you to their original state. This can be helpful in cases where a virtual machine has become unhealthy e.g. due to accidental overwrite of system files, or installation of incompatible software. Collaboratory limits how often this can be done to prevent undue resource consumption. If an attempt fails please try again later.

## Hardware spec:

[https://colab.research.google.com/notebook#fileId=1dint4ly-7h8Trw0XRJ1uhC\\_VKe\\_wDJfY](https://colab.research.google.com/notebook#fileId=1dint4ly-7h8Trw0XRJ1uhC_VKe_wDJfY)

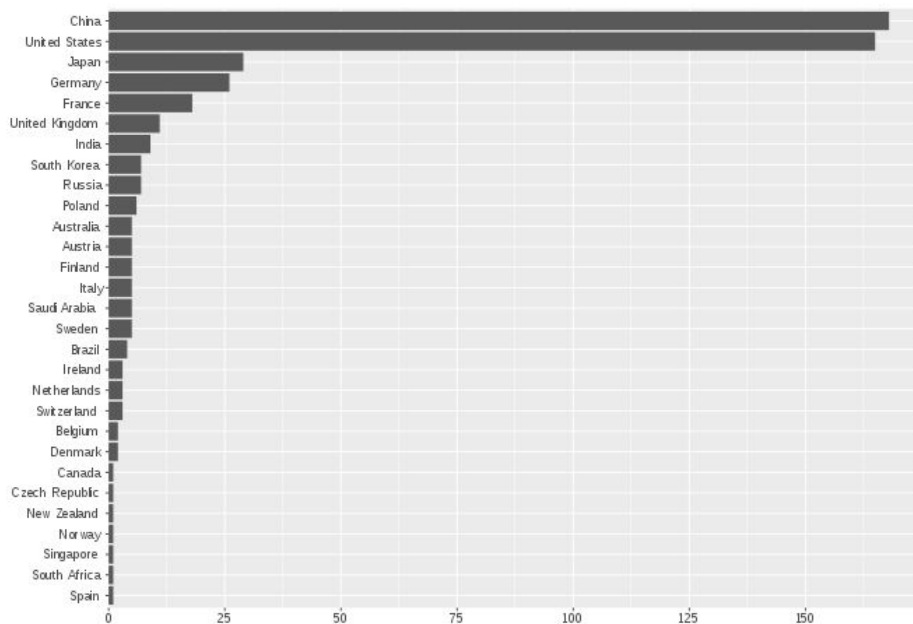
## 6. Разработка и организация кода. Докер



Воспроизводимые результаты python - докер:

[https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))

## 6. Разработка и организация кода. НРС

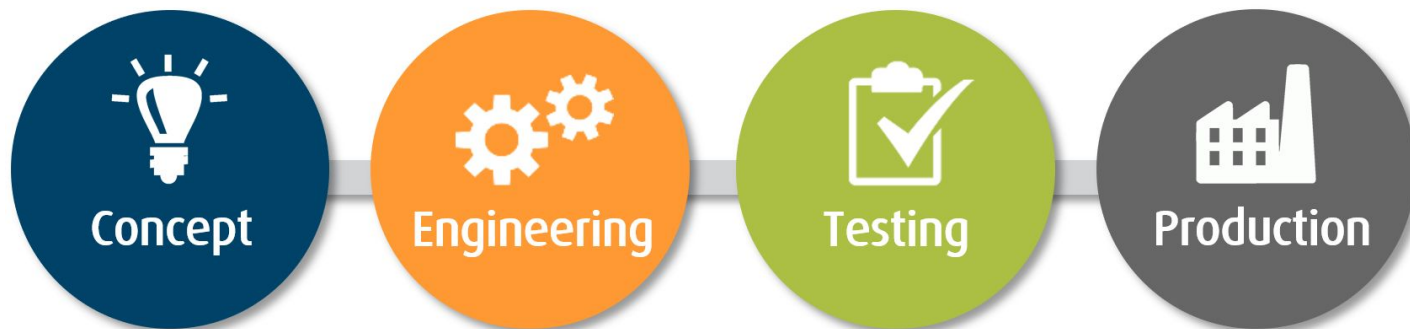


<https://ru.wikipedia.org/wiki/Top500>



<https://singularity.lbl.gov/>

## 7. Продакшн



[http://www.code-ps.com/public/codeps/media/image/code%20product%20solutions/general/Concept\\_Production.png](http://www.code-ps.com/public/codeps/media/image/code%20product%20solutions/general/Concept_Production.png)

Оптимизируем модель: корректируем параметры, добавляем новые данные.

Валидируем модель в реальных условиях, получаем оценку итогового качества.

**ВЖУХ-ВЖУХ**



**И В ПРОДАКШН**

## 7. Продакшн. Организация данных



## 7. Продакшн. Код



Компиляция кода в C++

# Пайплайн целиком:

1. Формулировка задачи
2. Подготовка датасета и разметки в соответствии с задачей
3. Определение критериев достижения успеха модели (метрика оценивания)
4. Предобработка данных
5. Выбор модели и оптимизация
6. Организация кода и разработка
7. Обертка модели в продакшн



Какие задачи **не нужно** решать с ML?

# Какие задачи **не нужно** решать с ML?

- Можно вывести зависимость исходя из знаний об устройстве мира (и эксперт может установить прямую зависимость)
- Зависимость предсказуемой величины имеет простой вид и его можно подобрать вручную, имея экспертное знание
- Нельзя набрать достаточное количество примеров из прошлого

# Полезные ссылки:

- **Введение в ML:**

Машинное обучение ВШЭ: <https://github.com/esokolov/ml-course-hse/>

Python: <https://stepik.org/course/Программирование-на-Python-67>

Статистика: <https://stepik.org/course/Основы-статистики-76>

Подборка ресурсов по машинному обучению (куча всего):  
<https://github.com/demidovakatyavvedenie-mashinnoe-obuchenie>

- **Соревнования:**

Самая популярная площадка: <https://kaggle.com>

Все соревнования здесь: <http://mltrainings.ru>

# Полезные ссылки:

- **ML Тусовка:**

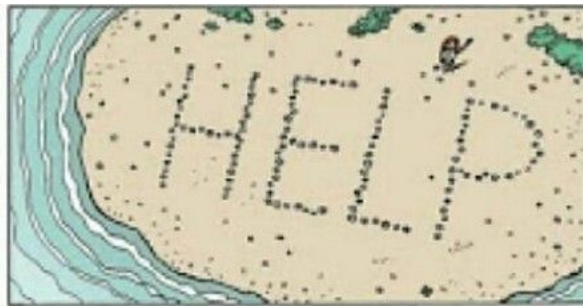
Slack датасайнс комьюнити: <http://ods.ai>

Группа express\_ml в Facebook: <https://www.facebook.com/groups/expressml/>

- **Новые разработки в области машинного обучения:**

Топовые конференции: <https://nips.cc/>, <https://icml.cc/>

Препринты публикаций: <https://arxiv.org/list/stat.ML/recent>





**CHANGED THE  
PARAMETERS IN THE SVM**

**CLAIMS IS AN ENTIRELY  
NEW WORK**