

# Основы машинного обучения

Лекция 1: Введение в машинное обучение

Полина Полунина  
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:**

1. Отбор и генерация признаков (Feature Engineering). Поиск и оптимизация модели (Grid Search).
2. Стекинг, вотинг. Градиентный бустинг. Пакеты XGBoost/Catboost/LightGBM
3. Соревнования по анализу данных, обзор решений, статей и актуальных методов.
4. Соревнования по анализу данных, обзор решений, статей и актуальных методов. Recap курса.

# Как будет строиться занятие по курсу?

< 30 минут: рекап прошлой лекции, проверка домашнего задания

< 60 минут: лекция

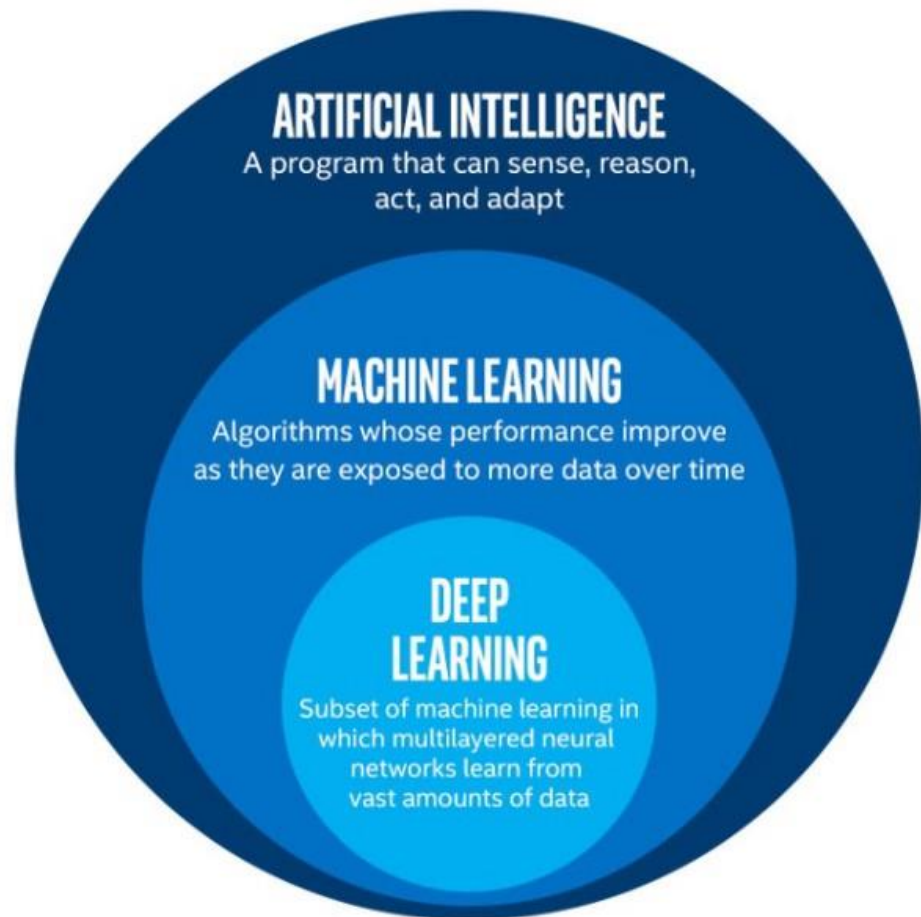
< 60 минут: практика

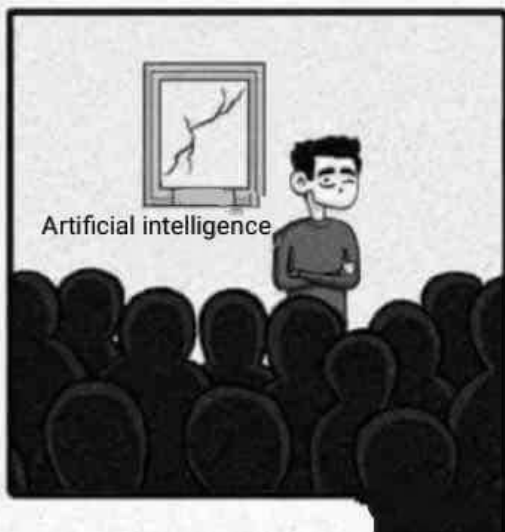
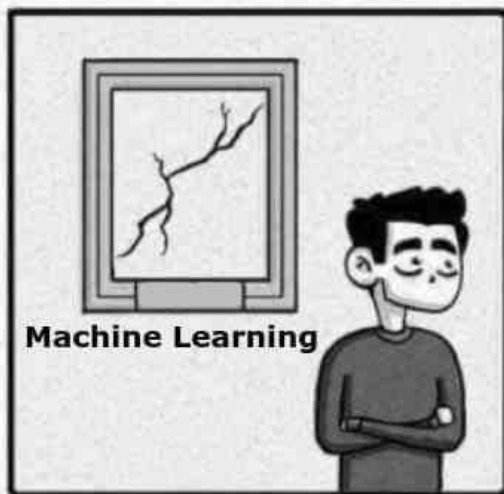
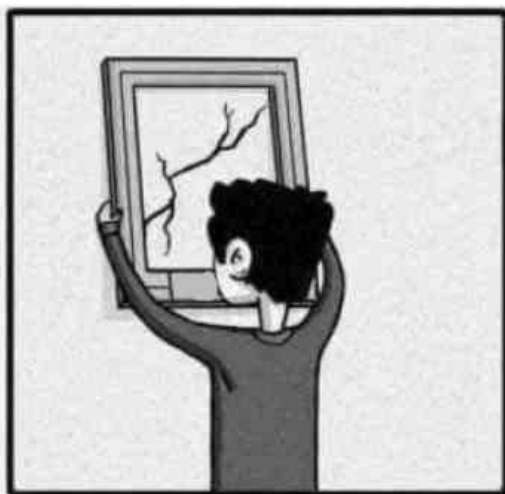
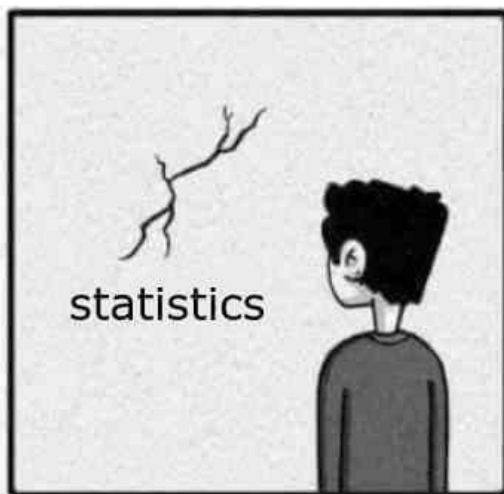
< 30 минут: мини-контест с лидербордом

# 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

# Машинное обучение?





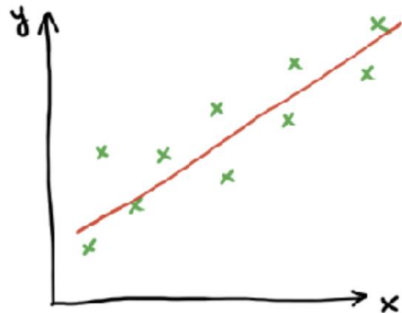
Как решить задачу с ML?



# 1. Тип задачи. Обучение с учителем

- Нужно предсказать число - **задача регрессии**

x	y
1	2
3	5
-1	-2
5	?

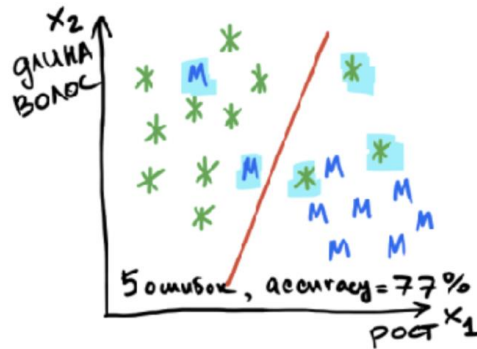


Например: определение возраста человека по фото

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

- Нужно предсказать класс - **задача классификации**

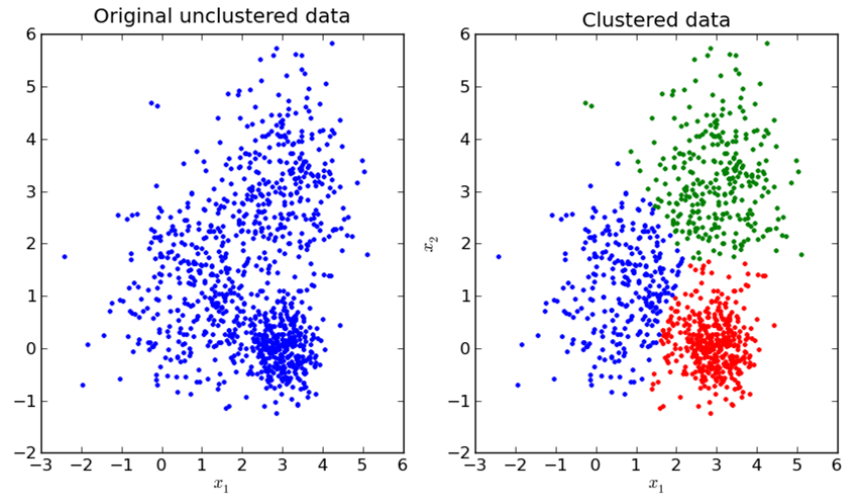
$x_1$	$x_2$	y
180	5	М
170	20	Ж
160	5	М
190	30	?



Например: распознавание букв или цифр

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

# 1. Пример специальных задач. Обучение без учителя



## Кластеризация

Например: кластеризация аудитории сайта

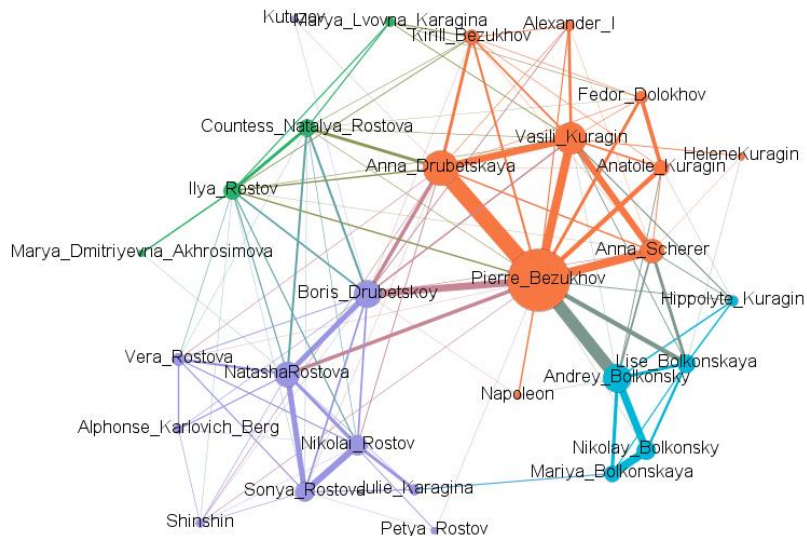


## Обучение с подкреплением

Например: кластеризация аудитории сайта

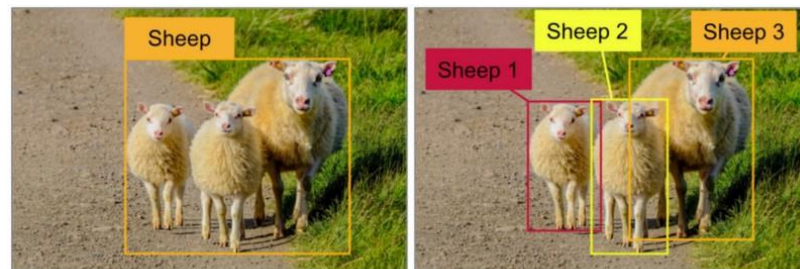
<https://www.youtube.com/watch?v=1wy2jtS5qck>

# 1. Пример специальных задач



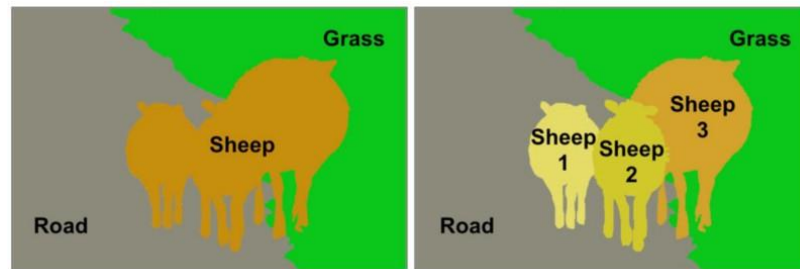
## Анализ графов

Например: Лев Толстой и сетевой анализ



**Classification + Localization**

**Object Detection**

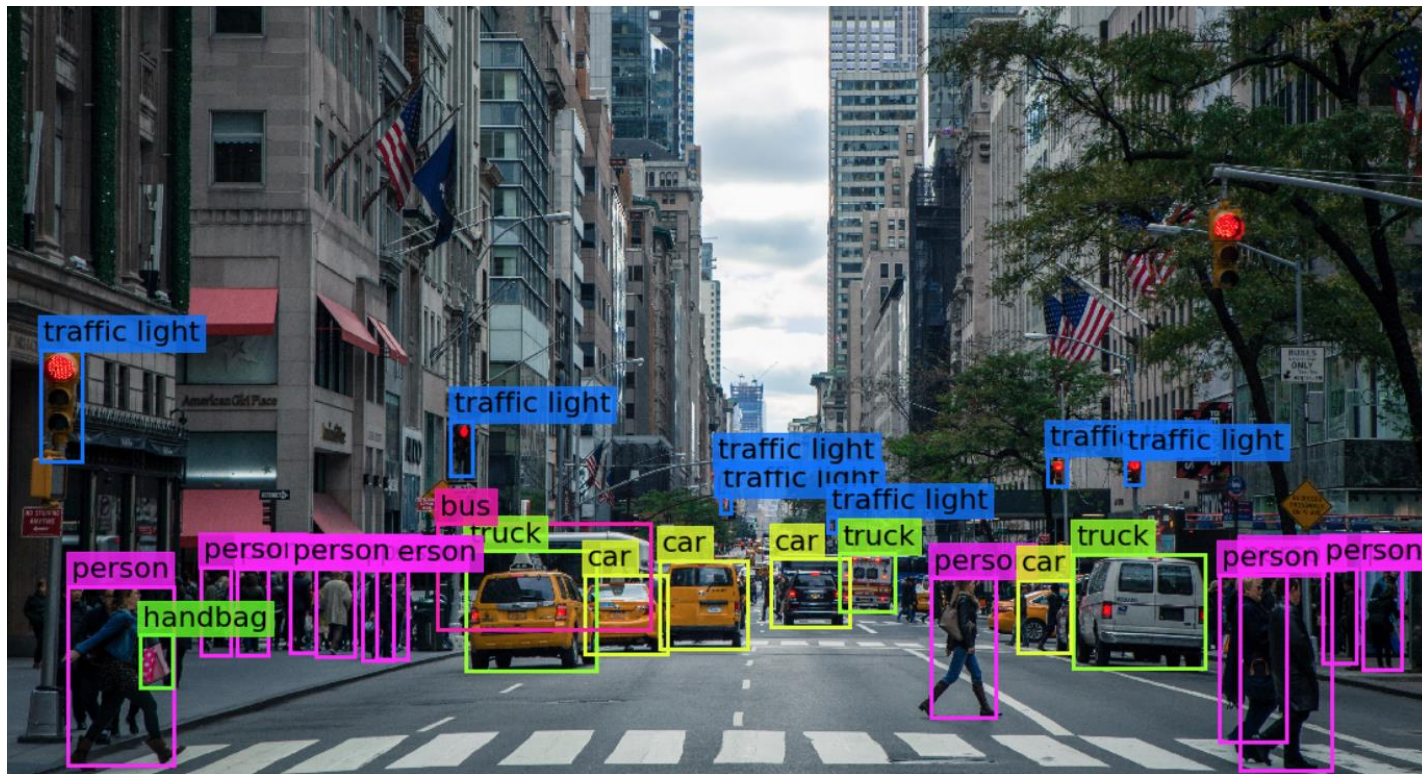


**Semantic Segmentation**

**Instance Segmentation**

## Компьютерное зрение

Например: Сегментация



Что за задача?

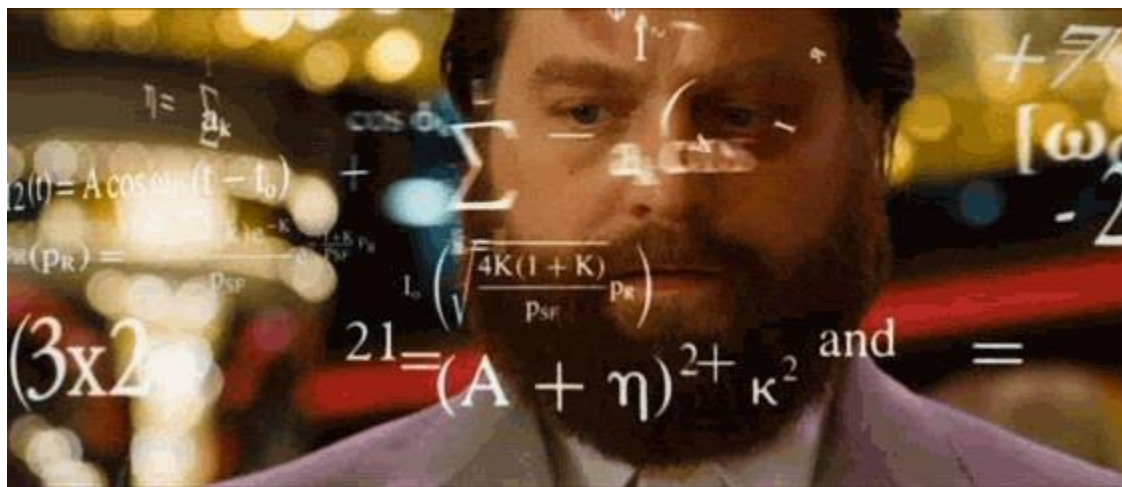
## 2. Организация данных

The diagram illustrates the structure of a CSV file. A blue bracket on the left side of the table rows is labeled "index labels". A red bracket above the table columns is labeled "column names". An orange bracket on the right side of the table rows is labeled "data".

	Mountain	Height (m)	Range	Coordinates	Parent mountain	First ascent	Ascents bef. 2004	Failed attempts bef. 2004
0	Mount Everest / Sagarmatha / Chomolungma	8848	Mahalangur Himalaya	27°59'17"N 86°55'31"E	NaN	1953	>>145	121.0
1	K2 / Qogir / Godwin Austen	8611	Baltoro Karakoram	35°52'53"N 76°30'48"E	Mount Everest	1954	45	44.0
2	Kangchenjunga	8586	Kangchenjunga Himalaya	27°42'12"N 88°08'51"E	Mount Everest	1955	38	24.0
3	Lhotse	8516	Mahalangur Himalaya	27°57'42"N 86°55'59"E	Mount Everest	1956	26	26.0
4	Makalu	8485	Mahalangur Himalaya	27°53'23"N 87°05'20"E	Mount Everest	1955	45	52.0
5	Cho Oyu	8188	Mahalangur Himalaya	28°05'39"N 86°39'39"E	Mount Everest	1954	79	28.0
6	Dhaulagiri I	8167	Dhaulagiri Himalaya	28°41'48"N 83°29'35"E	K2	1960	51	39.0
7	Manaslu	8163	Manaslu Himalaya	28°33'00"N 84°33'35"E	Cho Oyu	1956	49	45.0
8	Nanga Parbat	8126	Nanga Parbat Himalaya	35°14'14"N 74°35'21"E	Dhaulagiri	1953	52	67.0
9	Annapurna I	8091	Annapurna Himalaya	28°35'44"N 83°49'13"E	Cho Oyu	1950	36	47.0

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






## 2. Организация данных. Разметка данных

**Яндекс Толока**

**Простые задания  
за вознаграждения**

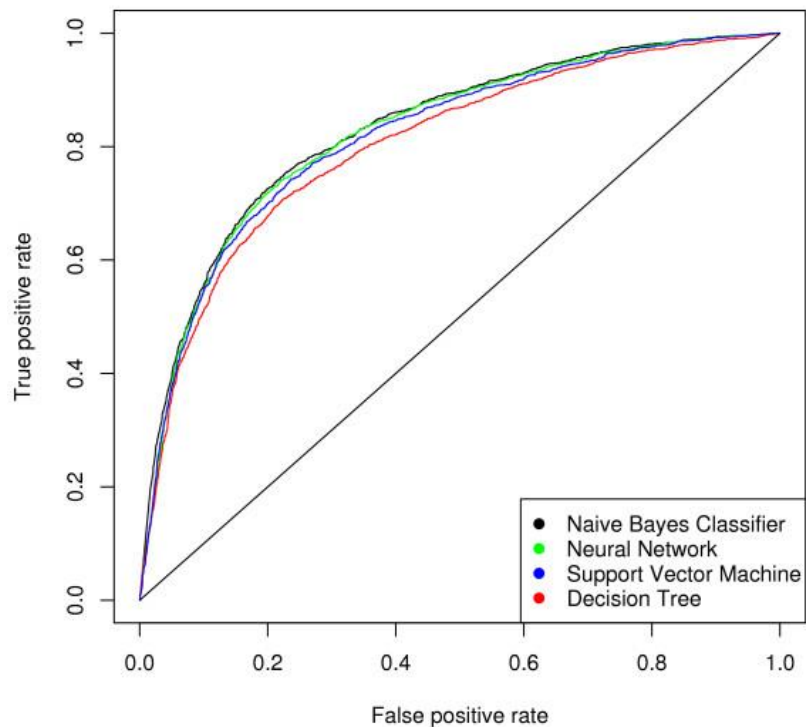
Присоединиться



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' (Yandex logo), and '1,0\$'. The background includes stylized clouds and green trees.

Если нужна разметка данных вручную

### 3. Метрика оценивания



**Метрики оценивания модели:**

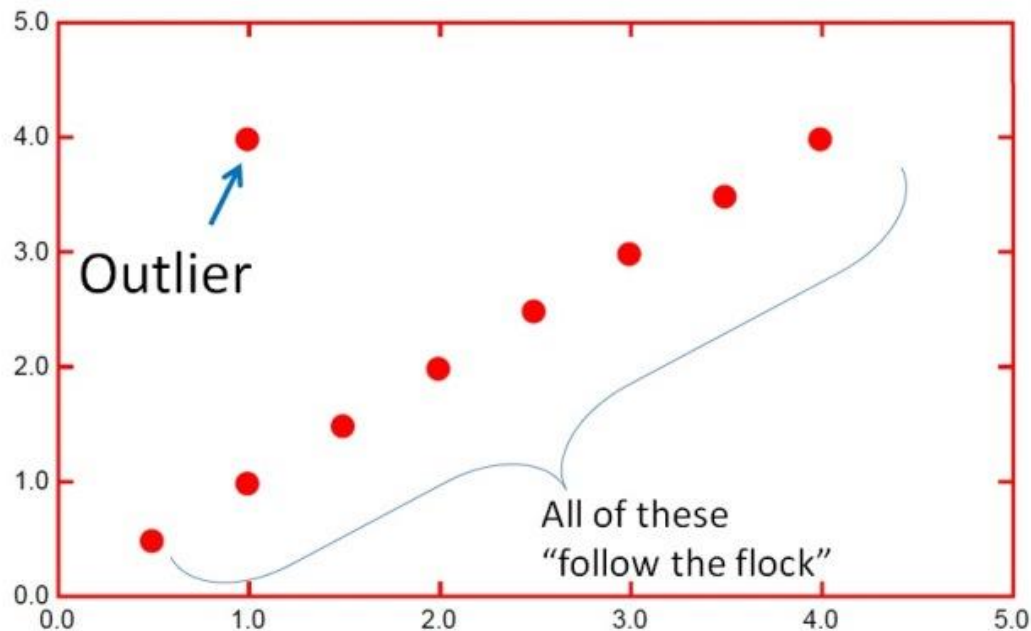
Точность %, Ошибки 1 или 2 рода, MSE

**Метрики оценивания в бизнесе:**

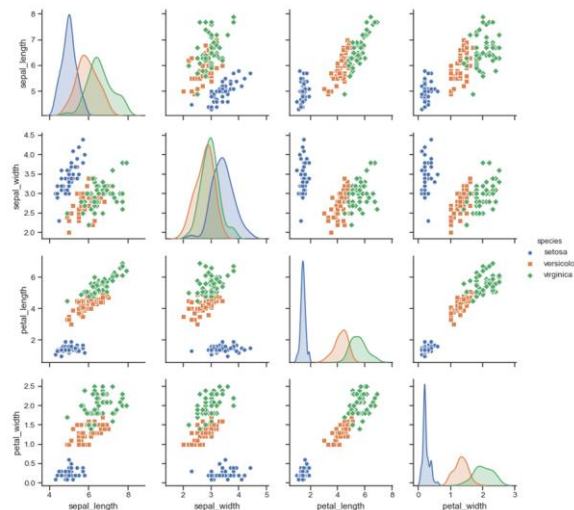
Деньги, ОТТОК



## 4. Предобработка данных



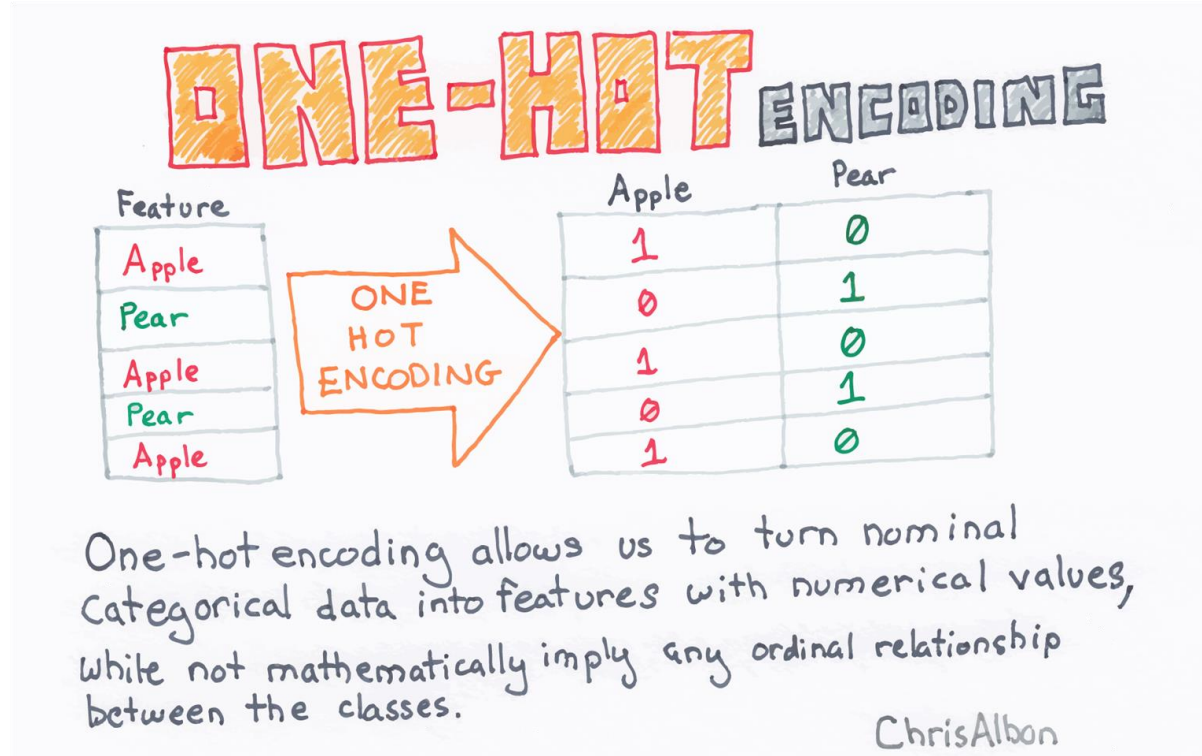
Never mind what the axes mean...



Изучение и визуализация данных.

Поиск артефактов.

## 4. Предобработка данных



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:

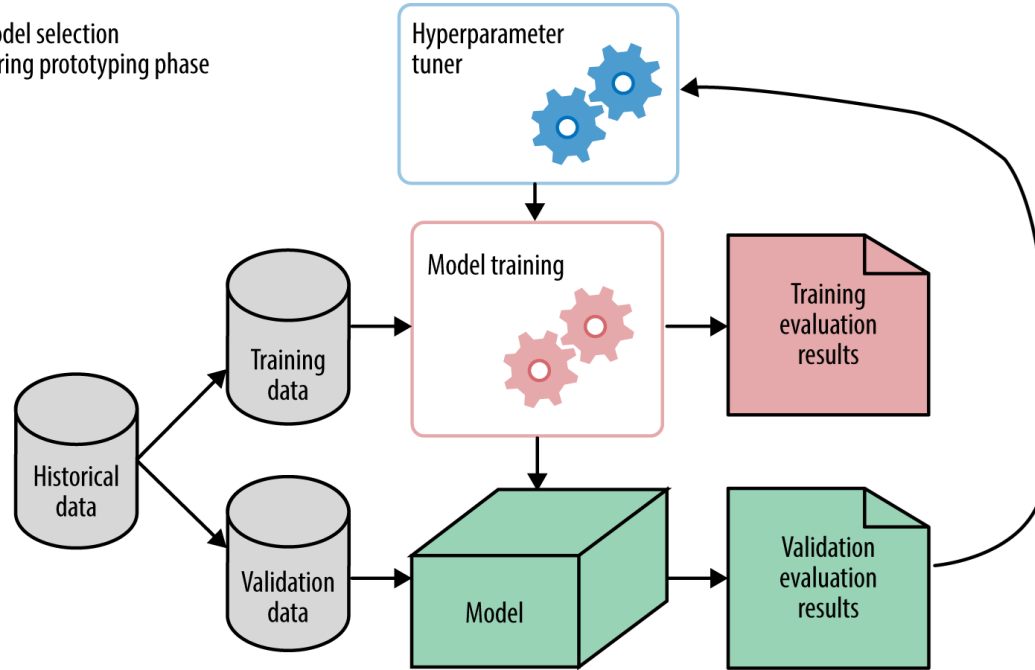
state of the art: статьи с *NIPS* <https://nips.cc/>,

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



## 5. Оптимизация модели

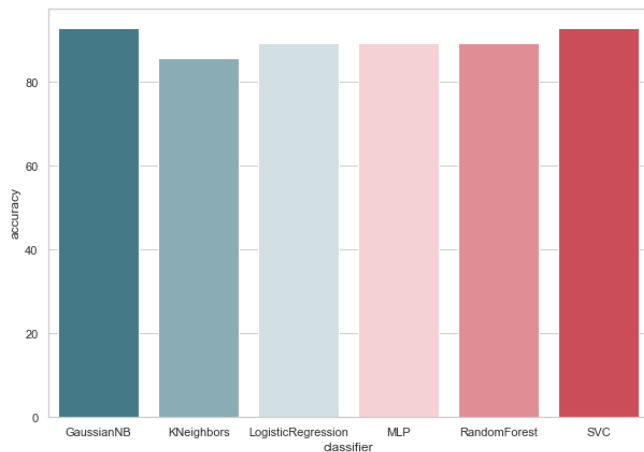
Model selection  
during prototyping phase



Оптимизация гипер параметров выбранной модели. Кросс Валидация

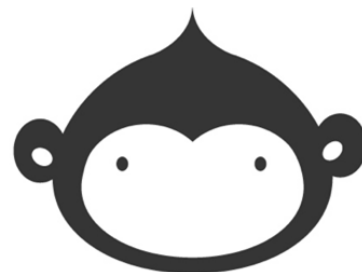
## 5. Оптимизация модели

Accuracy is 92.86% for GaussianNB  
Accuracy is 92.86% for SVC  
Accuracy is 85.71% for KNeighbors  
Accuracy is 89.29% for LogisticRegression  
Accuracy is 89.29% for RandomForest  
Accuracy is 89.29% for MLP



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



## 7. Продакшн



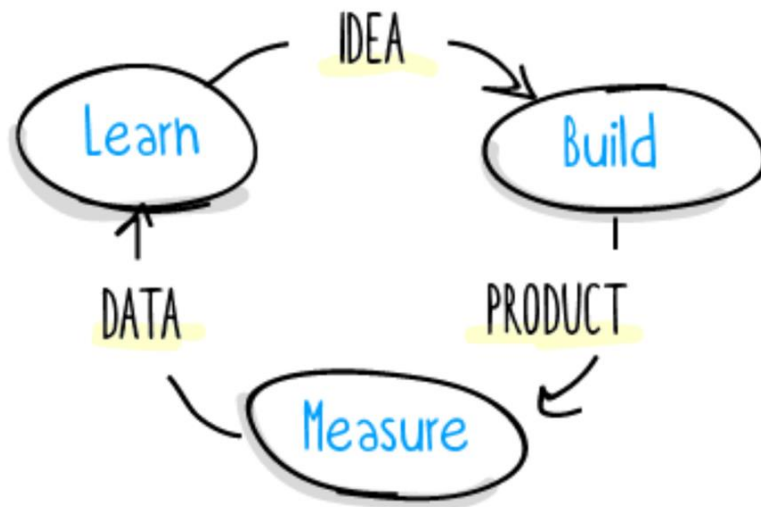
Воспроизводимые результаты python - докер:  
[https://en.wikipedia.org/wiki/Docker\\_\(software\)](https://en.wikipedia.org/wiki/Docker_(software))



Компиляция кода в C++



## 7. Продакшн



Оптимизируем модель: корректируем параметры, добавляем новые данные.

Валидируем модель в реальных условиях, получаем оценку итогового качества.

# 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 ( $E=mc^2$ )
- 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:**

Express ML курс: [https://github.com/Slinkolgor/express\\_ml](https://github.com/Slinkolgor/express_ml)

МЛ кукбук: <https://chrisalbon.com/>

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: <https://github.com/demidovakatya/vvedenie-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>

