

UDV Summer School 2021

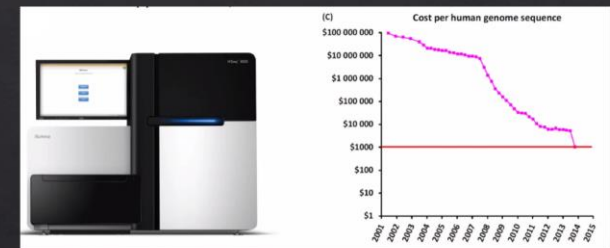
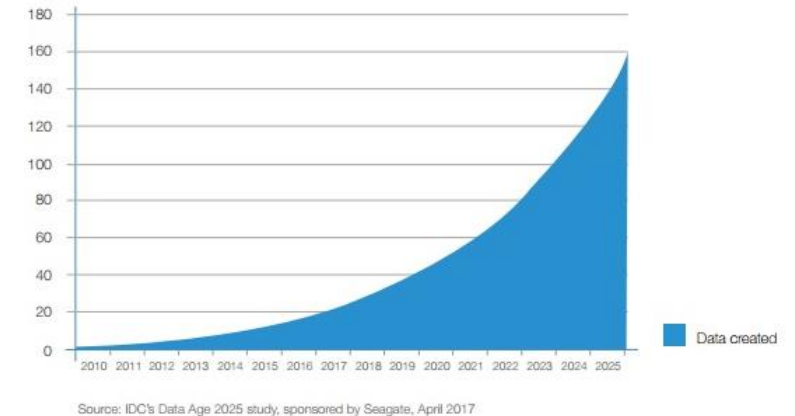
Машинное обучение

> Чернышов Юрий



> Повышается важность данных

- Разработки с середины XX века
- Сейчас много данных
- Сейчас мощное оборудование
- Недостаток ресурсов
- Развитие ИТ



Революция 2000-х

Секвенирование генома человека упало в цене от 100 миллионов до ~1000 \$

> Передовики цифровизации



Телеком



Ритейл



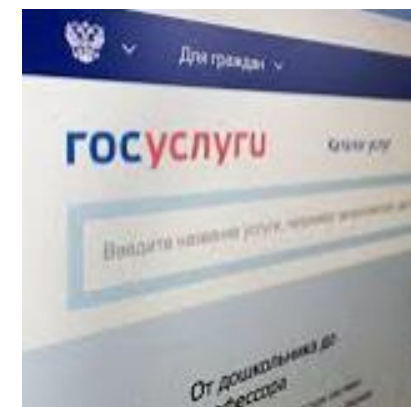
Банки



Металлурги

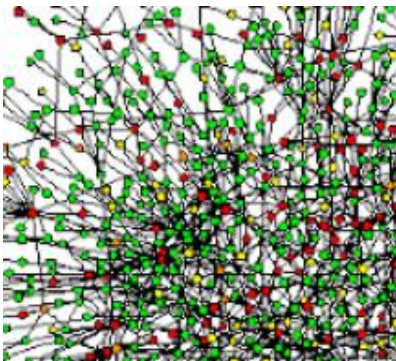


Медицина



Государство

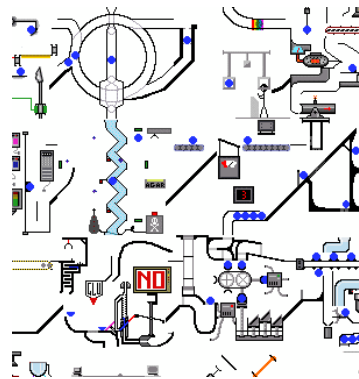
> А информационная безопасность?



Увеличивается
количество систем



Системы
устаревают



Увеличивается
сложность систем



Новые виды атак



Усложняются
взаимодействия



РОССИЙСКАЯ ФЕДЕРАЦИЯ
ФЕДЕРАЛЬНЫЙ ЗАКОН

О безопасности критической информационной
инфраструктуры Российской Федерации

Принят Государственной Думой 12 июля 2017 года
Одобрен Советом Федерации 19 июля 2017 года

Статья 1. Сфера действия настоящего Федерального закона

Настоящий Федеральный закон регулирует отношения в области обеспечения безопасности критической информационной инфраструктуры Российской Федерации (далее также – критическая информационная инфраструктура) в целях ее устойчивого функционирования при проведении в отношении ее компьютерных атак.

Статья 2. Основные понятия, используемые в настоящем
Федеральном законе

Для целей настоящего Федерального закона используются следующие основные понятия:

Регуляция
ФЗ-187, ...

> Типовые задачи при работе с данными

- Поиск
- Обработка (проверка, очистка)
- Передача и хранение
- Организация доступа
- Использование
- Создание полезного продукта
- Визуализация



> Что такое машинное обучение

- Mitchell "Machine Learning" (1997) : «Компьютерная программа обучается на основе опыта **E** для решения задач **T** с метрикой **P**, если продуктивность решения задач **T**, измеряемая по метрике **P**, увеличивается с приобретаемым опытом **E**»
- Компьютеры учатся как люди, «очеловечивание» процесса добавляет таинственности, создает шумиху, привлекает инвесторов
- В основе машинного обучения – математическое моделирование
- Не является «волшебной пилюлей» для всех задач

> Иллюстрация машинного обучения

- Различают этап обучения модели (изучение данных, поиски закономерностей, определение правила работы «черного ящика») и этап логического вывода (применение накопленных знаний).
- Вектор признаков объекта (числа). Параметры и гиперпараметры. Алгоритм обучения. Модель.
- Отбор признаков, пример характеристик автомобиля.

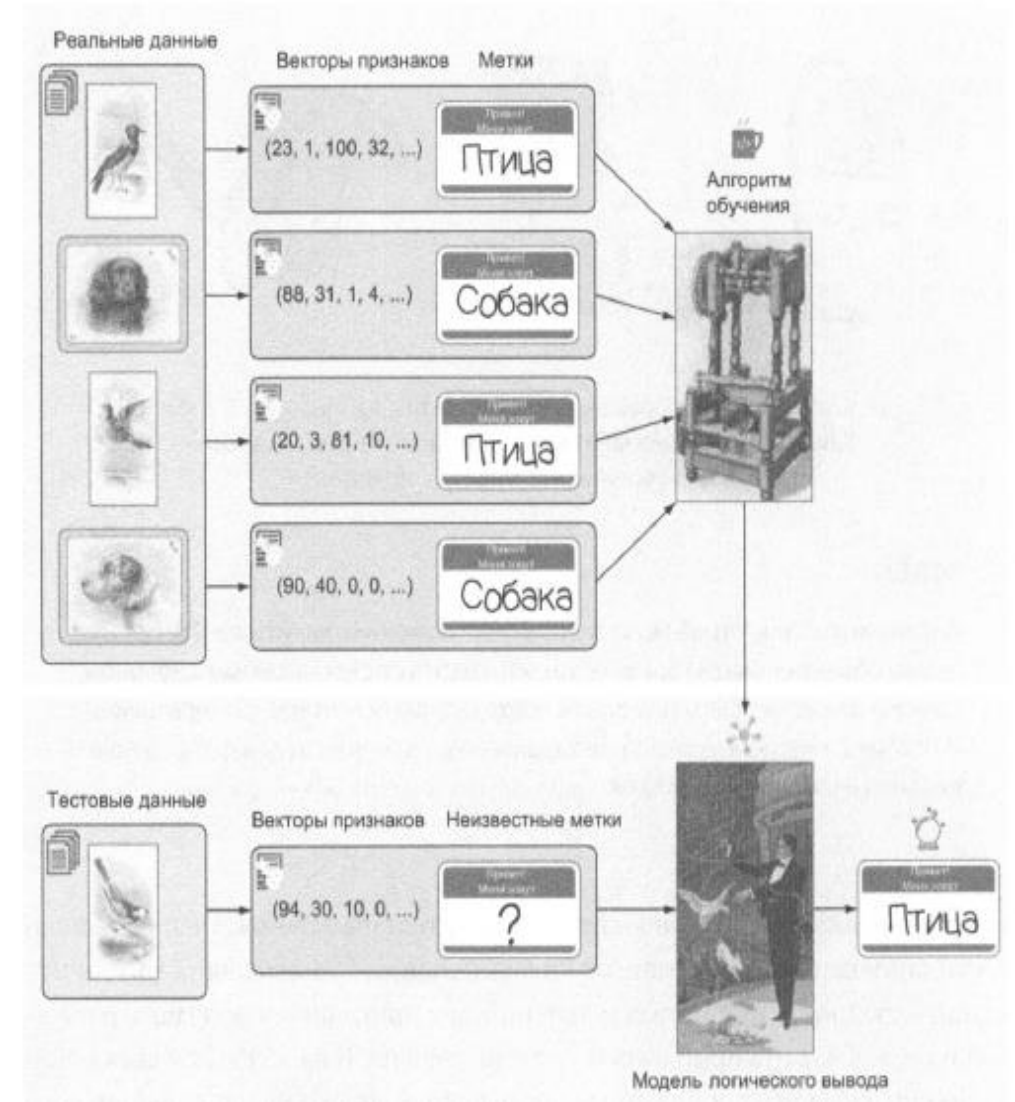
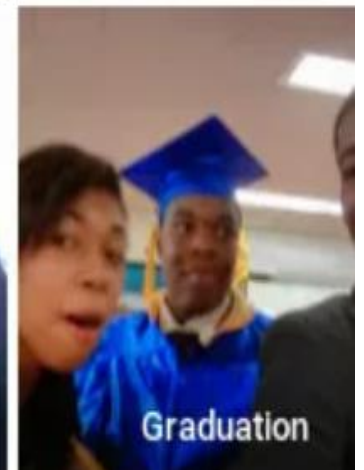
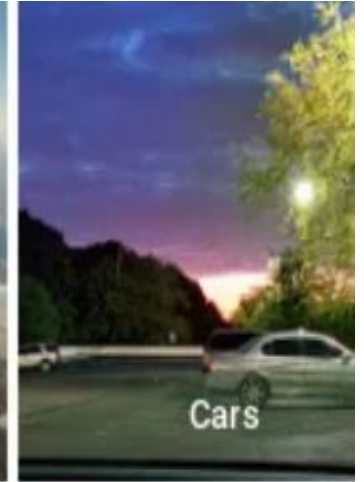


Иллюстрация из книги Н. Шакла «Машинное обучение»

> В основе обучения данные (признаки)

- важны еще больше, чем модели
- нужны для описания объектов
- должны быть числовыми
- могут быть зависимыми друг от друга (рост и вес)
- имеют разное влияние на процесс
- имеют непрерывную или дискретную природу



Google gorillas fail,
источник <https://boingboing.net/2018/01/11/gorilla-chimp-monkey-unpersone.html>

> Типовые задачи машинного обучения

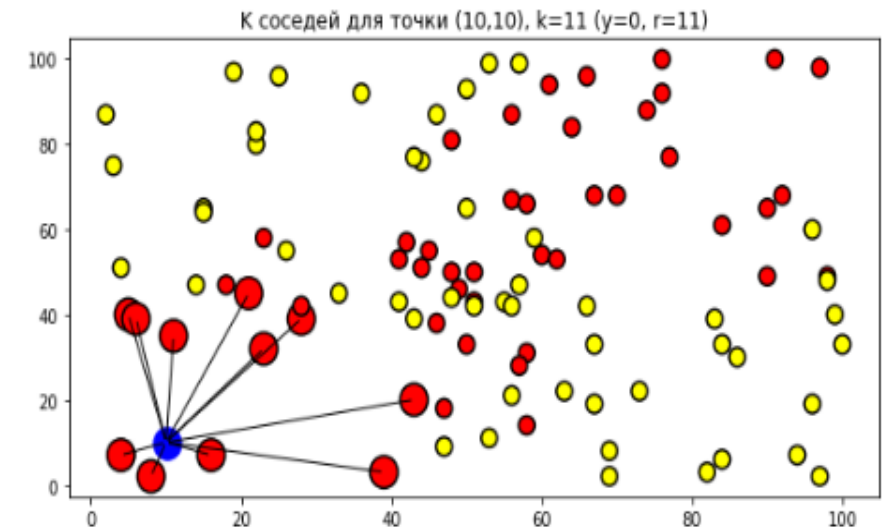
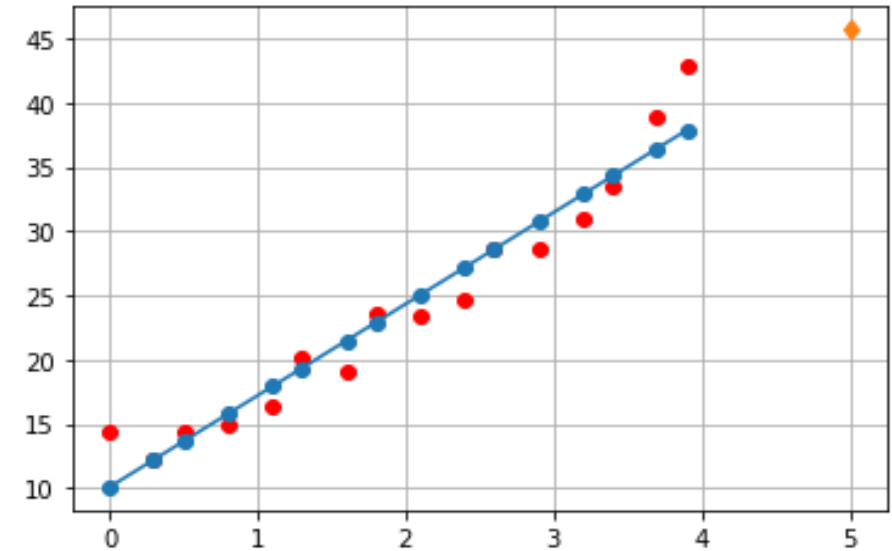
- прогнозирование (погода, курс акций)
- классификация (клиенты, пациенты)
- кластеризация (покупатели, избиратели)
- поиск аномалий (злонамеренная активность, отклонения в работе оборудования)
- моделирование (дорогие эксперименты)
- проверка гипотез (маркетинговые исследования)
- оптимизация

> Типы машинного обучения

- Supervised Learning: обучение с учителем
- Unsupervised Learning: обучение без учителя
- Reinforcement Learning: обучение с подкреплением
- Deep Learning: глубокое обучение, нейронные сети
- Transfer Learning: использование обученных моделей на одних задачах для решения других задач
- Federated Learning: распределенное обучение

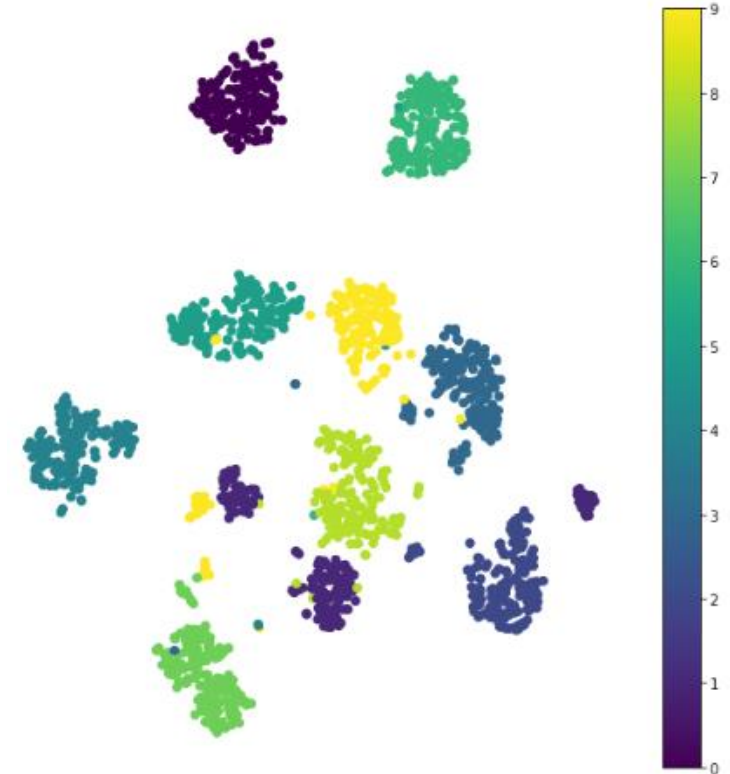
> Обучение с учителем

- Размеченные данные (известен результат, оценка, target)
- Выполняются вычисления и получившийся результат сравнивается с известной оценкой (нужна метрика)
- Проводится оптимизация (изменение параметров для улучшения качества)
- Примеры: задачи регрессии (предсказания), задачи классификации.



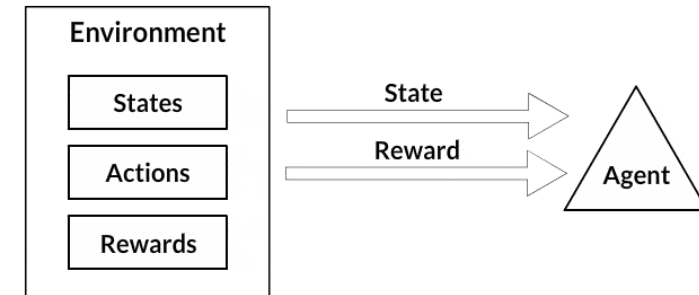
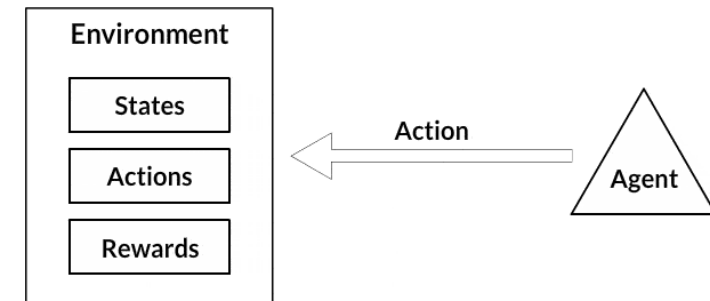
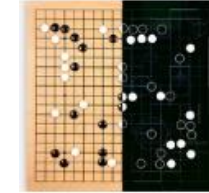
> Обучение без учителя

- Нет известных оценок
- Алгоритм сам определяет правила для решения задачи
- Проводится оптимизация (изменение параметров для улучшения качества)
- Примеры: задача кластеризации, уменьшение размерности



> Обучение с подкреплением

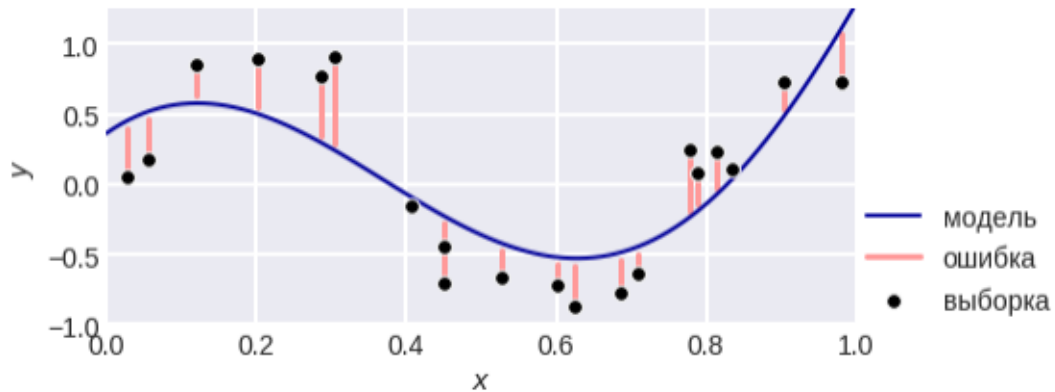
- Среда играет роль учителя, предлагая не оценки, а ответы на действия
- Есть набор состояний, действий и наград за выполненные действия
- Исследование вместо использования: алгоритм выполняет действия, получает оценку внешней среды, проводит эксперименты, меняя действия
- Примеры: самообучающиеся алгоритмы для игры в шахматы, Go, робототехника



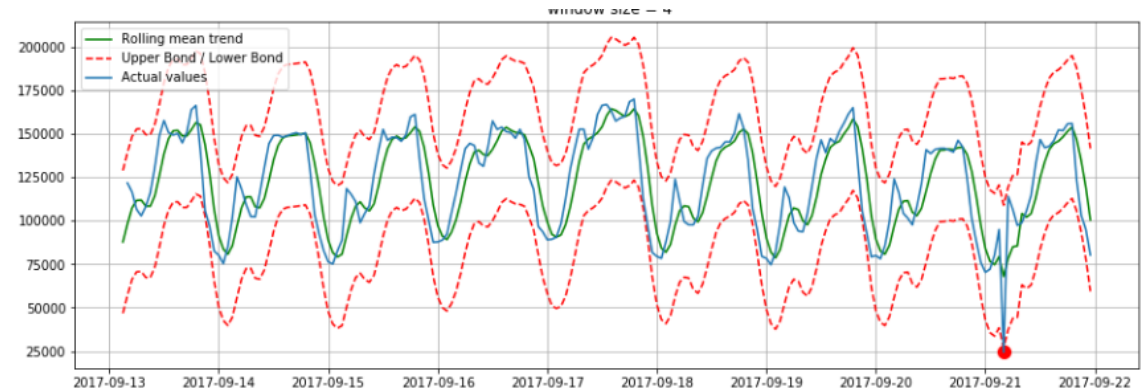
> Прогнозирование (регрессия)

Виды регрессии:

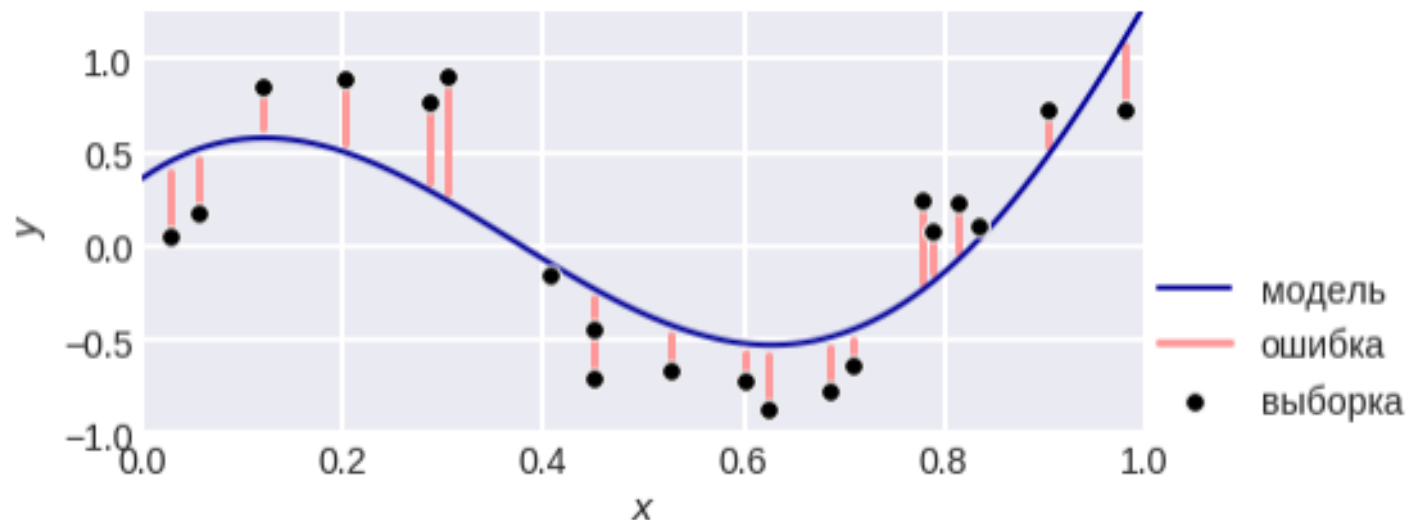
- линейная и множественная регрессии (определение зависимостей факторов)
- временные ряды (предсказание спроса, поиск аномалий)



$$y = w_0 + x_1 w_1 + \dots + w_n x_n.$$



➤ Метрики в задачах регрессии

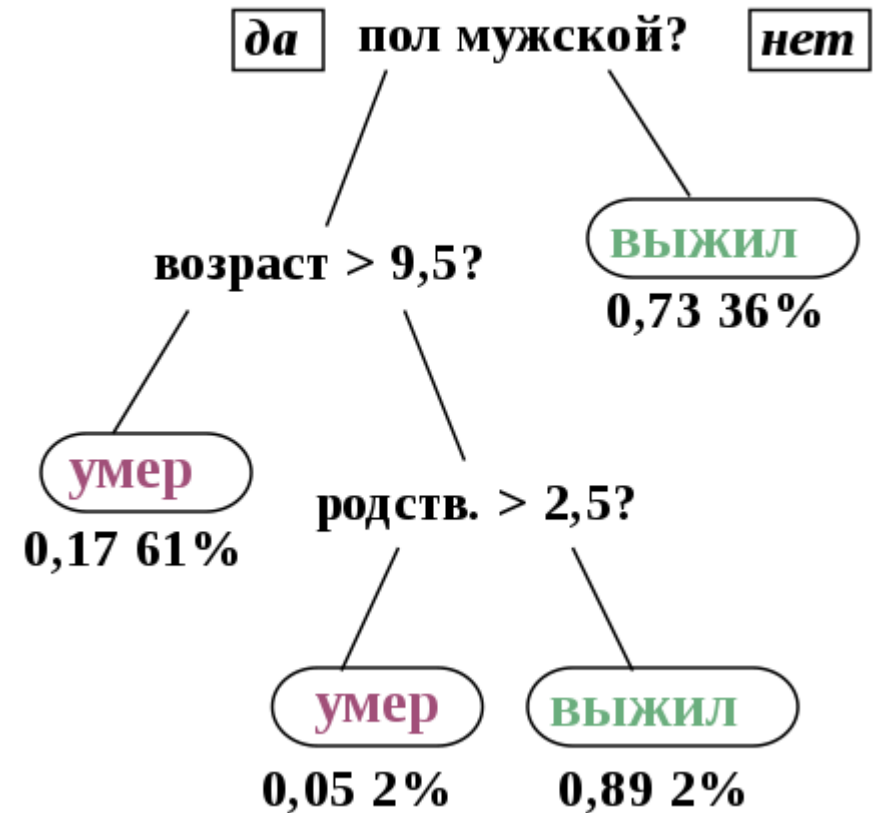


$$\text{MAE} = \frac{1}{m} \sum_{i=1}^m |a_i - y_i|.$$

$$\text{RMSE} = \sqrt{\text{MSE}} = \sqrt{\frac{1}{m} \sum_{i=1}^m |a_i - y_i|^2}$$

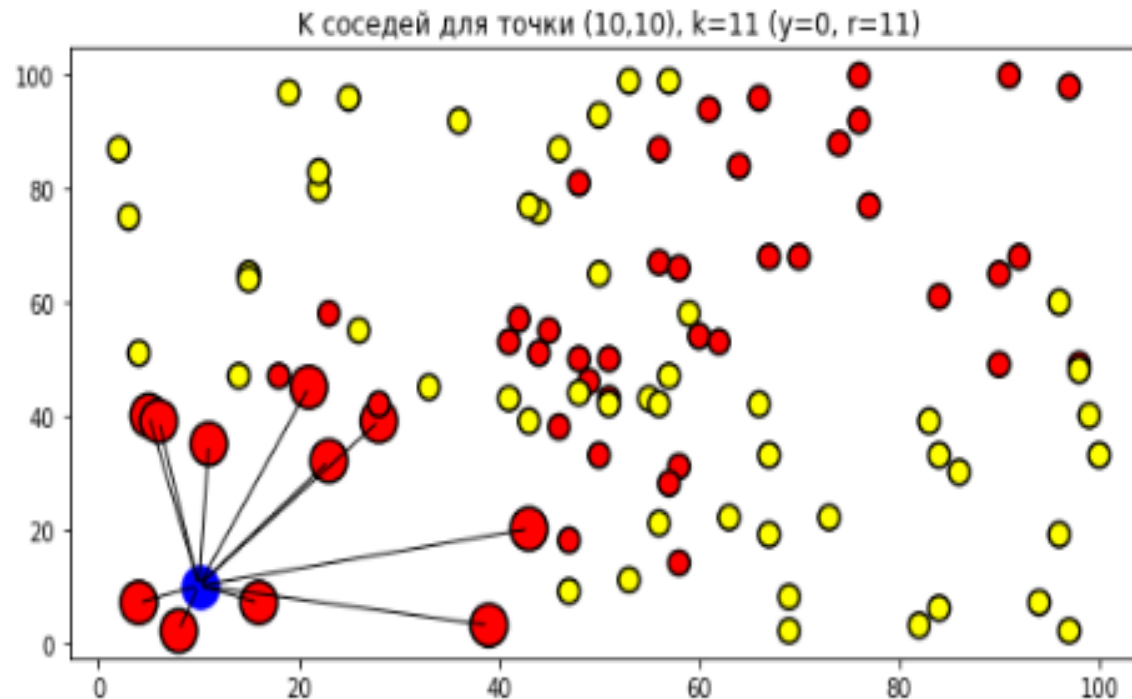
> Классификация. Деревья принятия решений

- Применения: кредитный скоринг, диагноз болезни
- Простота интерпретации
- Необходимость переобучать модель заново при изменении условий



> Классификация. Метод К ближайших соседей

- Применение: прогнозы поведения (результаты голосования, покупательская активность)



> Метрики в задачах классификации. Accuracy, precision, recall, F1- метрика.

П р о г н о з	Реальность	
	TP	FP
	FN	TN

Матрица ошибок (confusion matrix)

- Ошибка первого рода – отвергнута верная «нулевая гипотеза» (False Positive, FP)
- Ошибка второго рода – принята неверная нулевая гипотеза (False Negative, FN)
- Accuracy («Правильность») – сколько всего результатов было предсказано верно, недостаточность метрики accuracy
- Precision (Точность) – сколько положительных предсказаний оказалось верными
- Recall (Полнота) – доля правильных предсказаний для положительных результатов
- F1- метрика

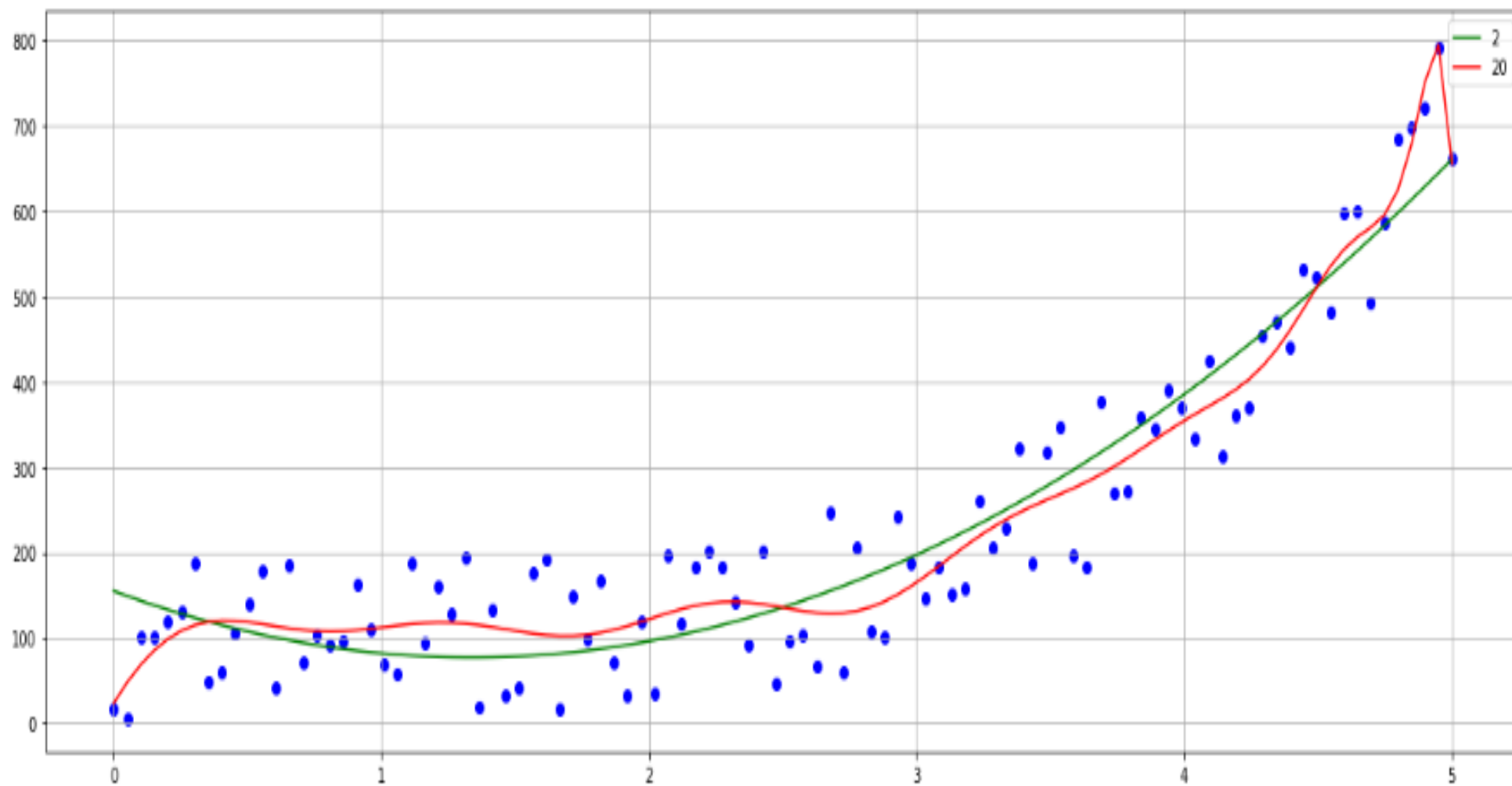
$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

$$F_{\beta} = (1 + \beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

> Проблема переобучения

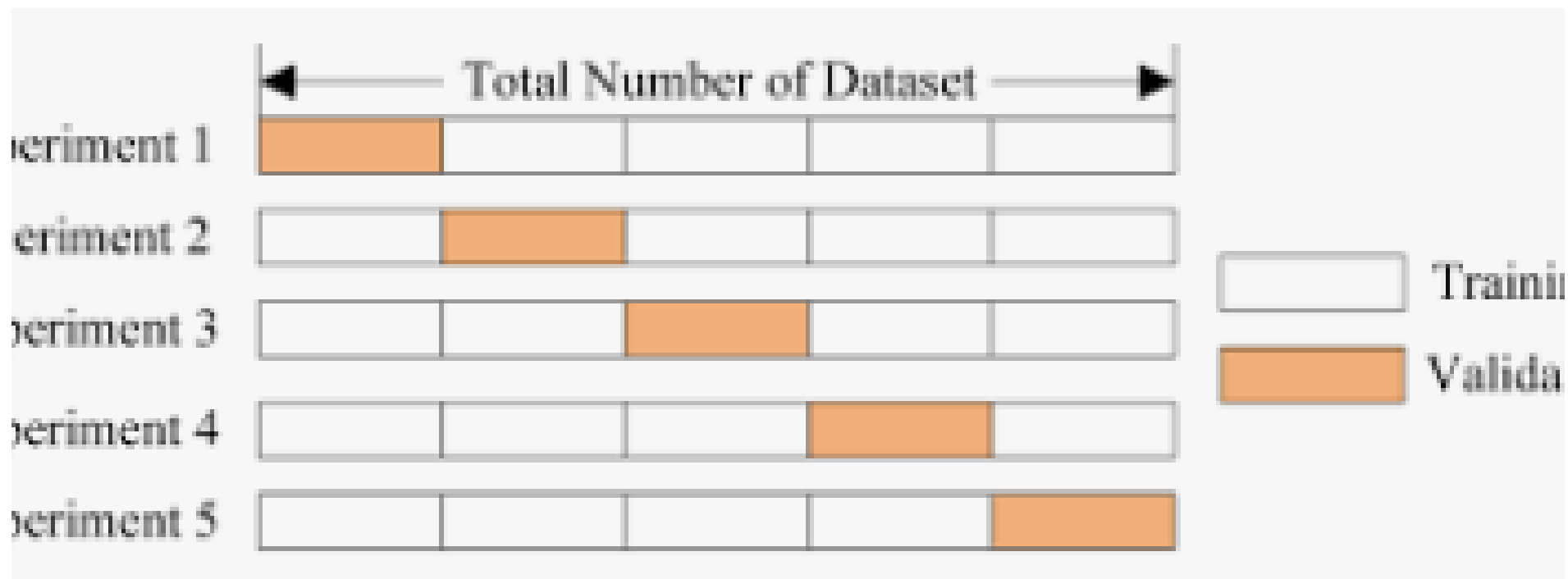


> Проверка: отложенная выборка

- Тренировочная выборка для обучения
- Тестовая выборка для проверки
- Равномерное распределение признаков
- Переобучение: высокий результат на тренировочной выборке и низкий на тестовой
- Недообучение: низкий результат на тренировочной выборке



> Проверка: кросс-валидация



> Терминология ML

- Искусственный интеллект (AI, artificial intelligence)
- Машинное обучение (ML, machine learning)
- Глубокое обучение (DL, deep learning)

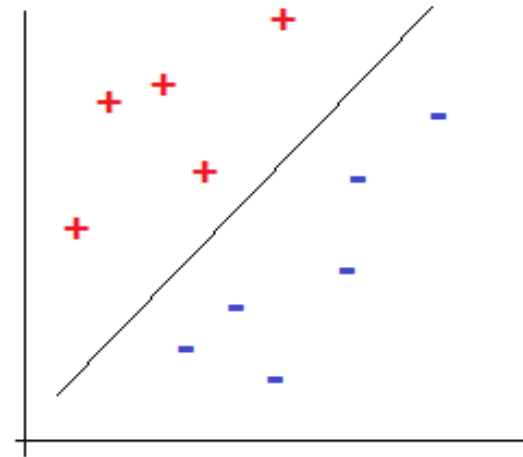
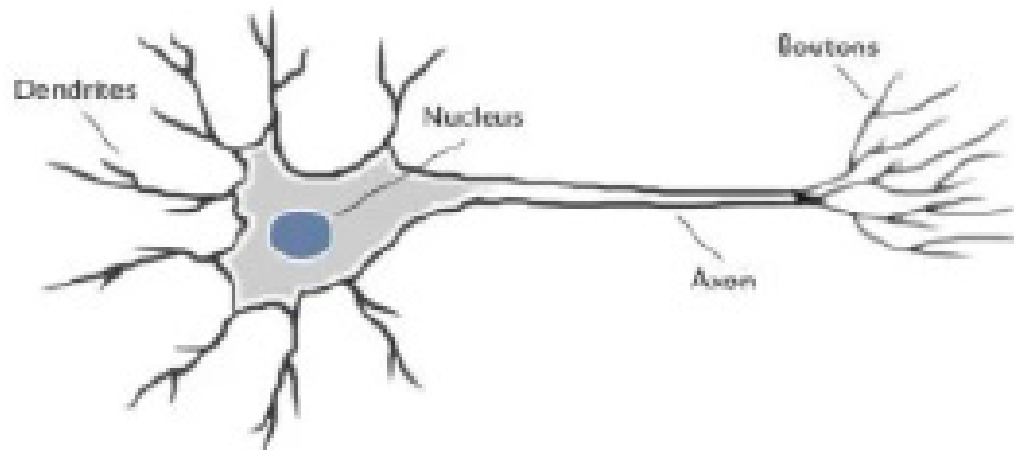


<https://netology.ru/blog/09-2019-data-science-daydjest-10>

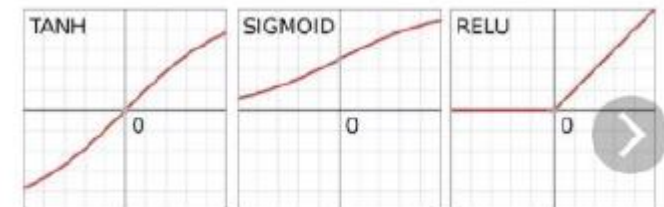
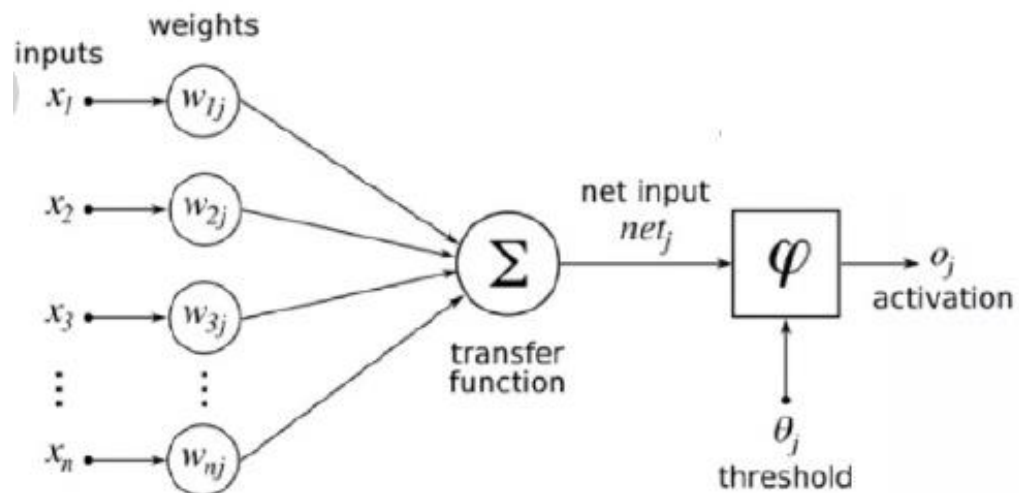
> Глубокое машинное обучение

- Модель «черного ящика»
- Тяжелые продолжительные вычисления (много параметров), несмотря на большие процессорные мощности и новые алгоритмы
- Используются нелинейные правила – позволяет получить более глубокий результат по сравнению с линейными моделями
- Не является фундаментальной наукой, основано на интуиции
- Работает на больших данных
- Как язык программирования - объясняет модели чему и как ей надо научиться

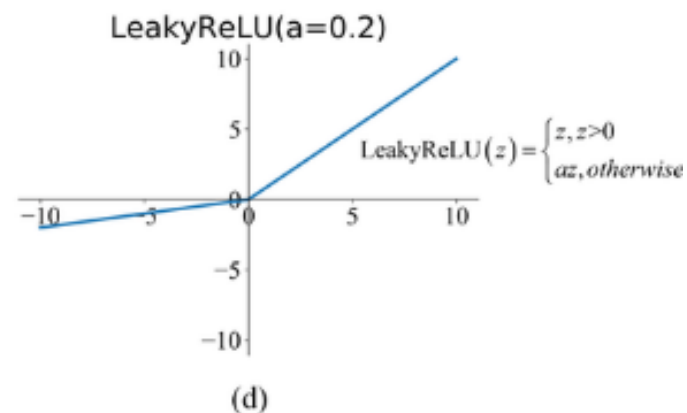
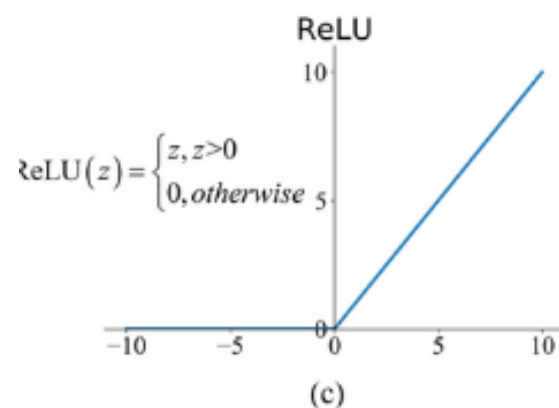
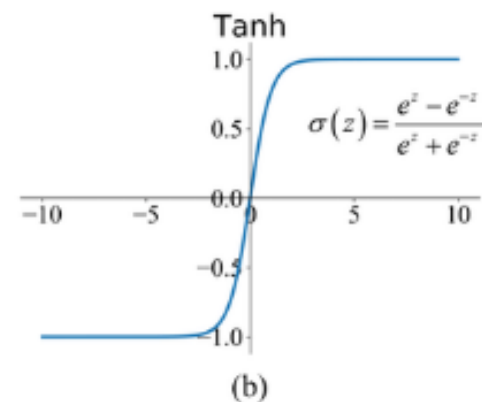
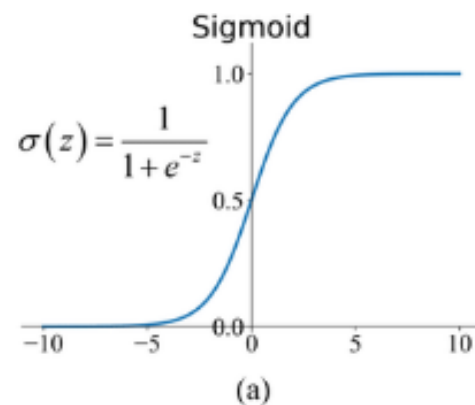
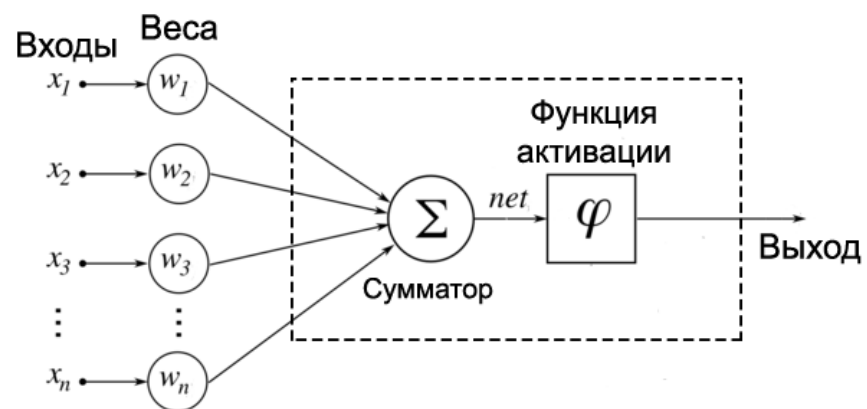
➤ Модель нейрона. Перцептрон.



$$y = w_0 + x_1 w_1 + \dots + w_n x_n.$$

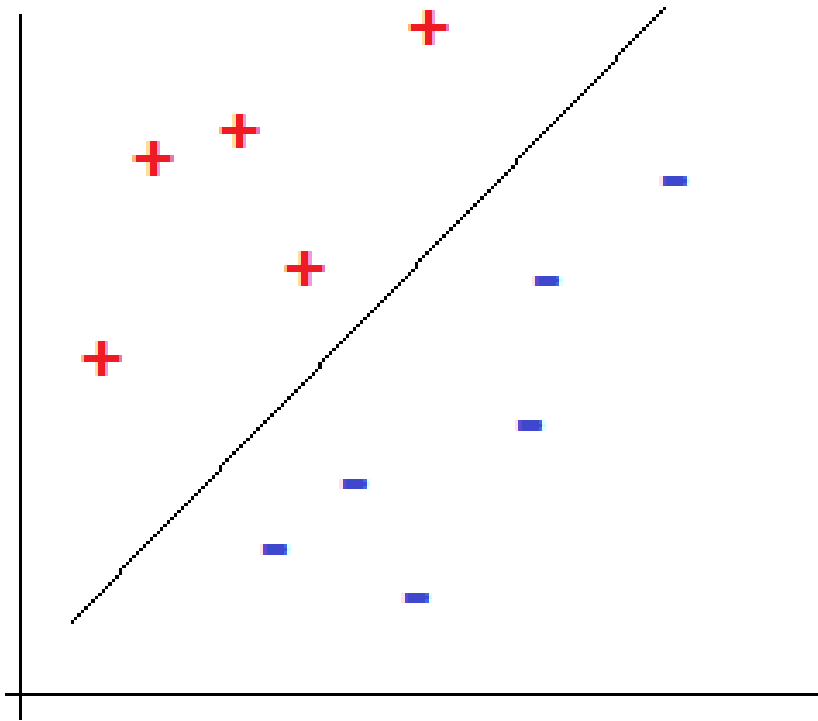


➤ Функции активации

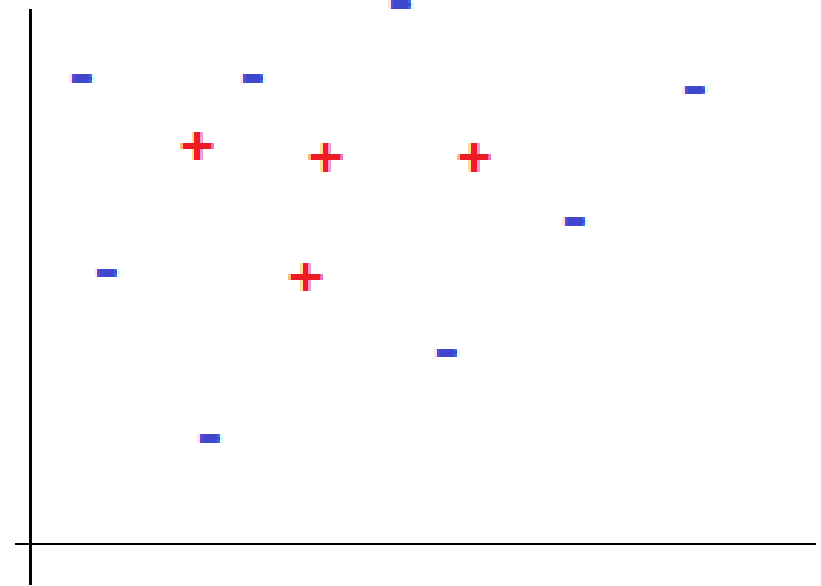


> Проблема применения регрессии

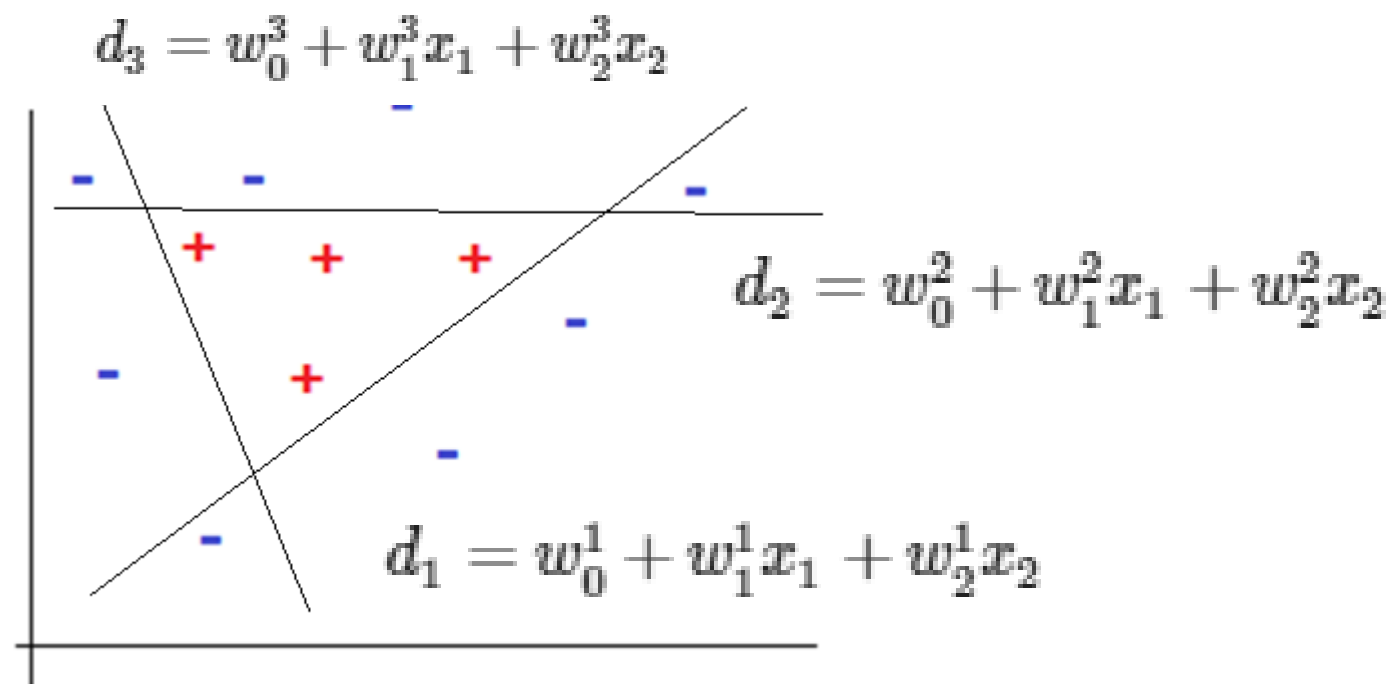
$$y = w_0 + x_1 w_1 + \dots + w_n x_n$$



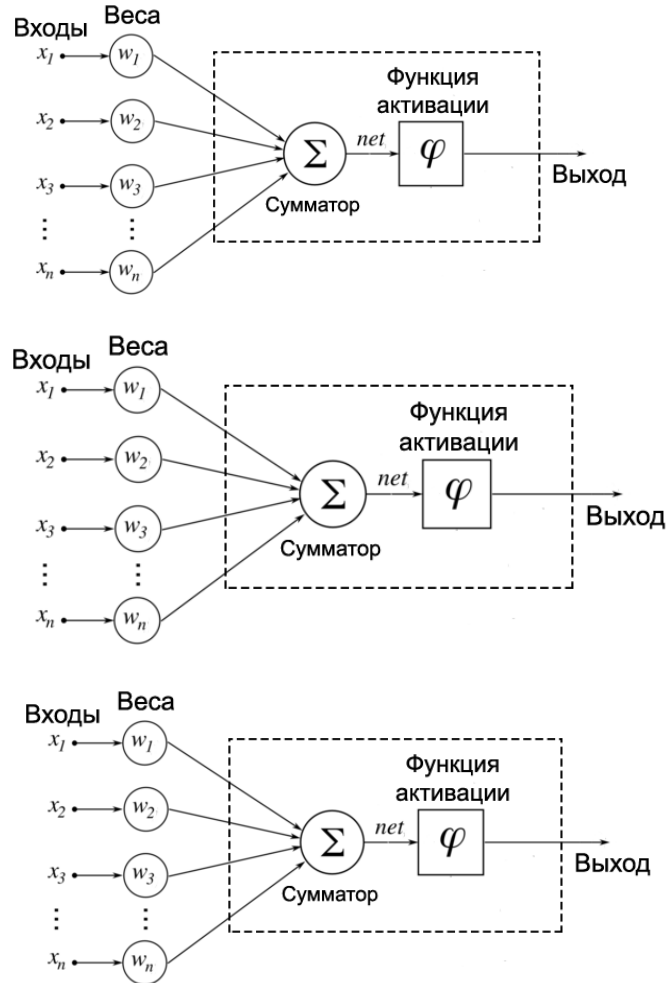
«Проблема треугольника»



> Решение проблемы треугольника

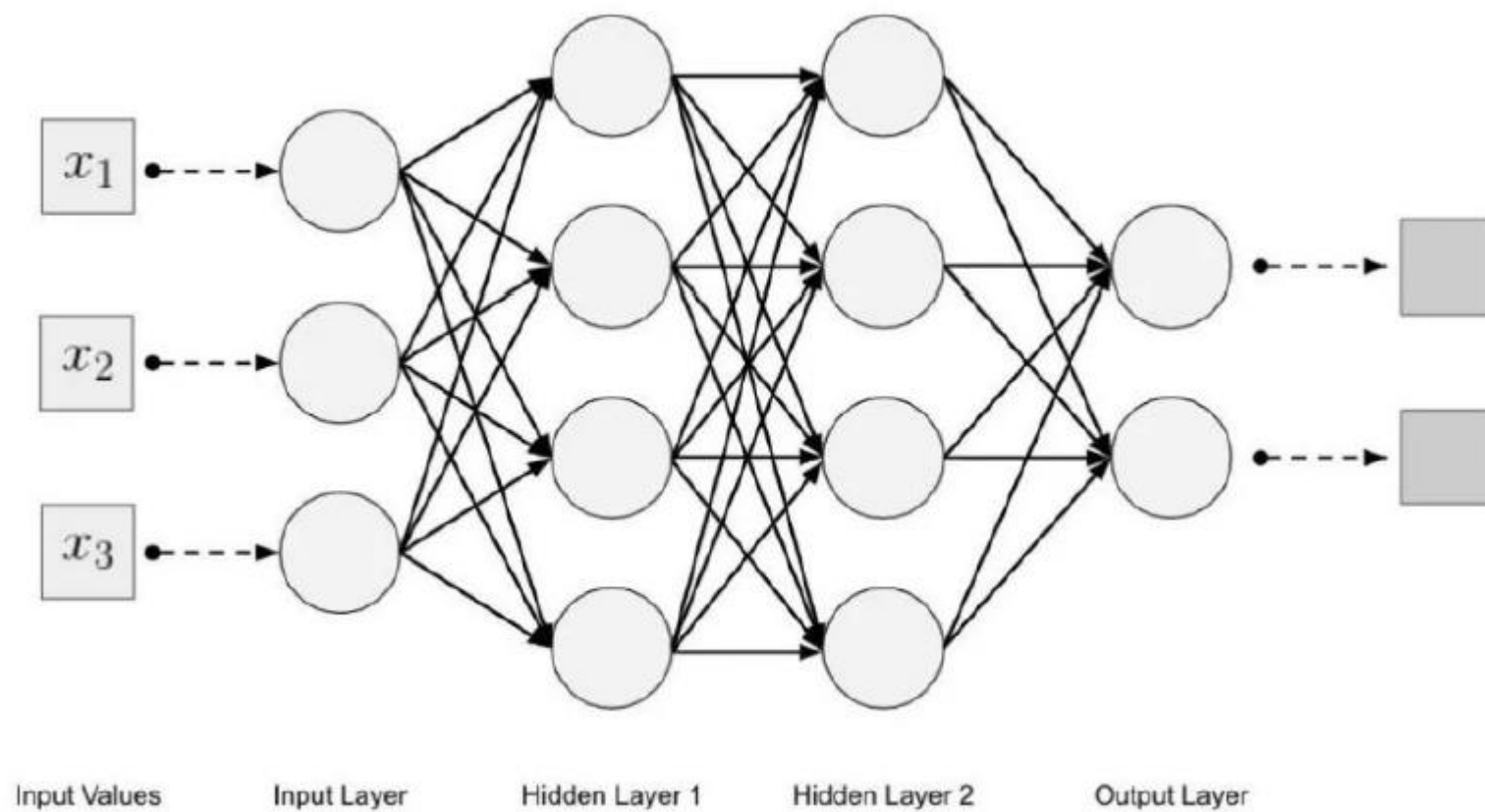


> MLP MultyLayer Perceptron

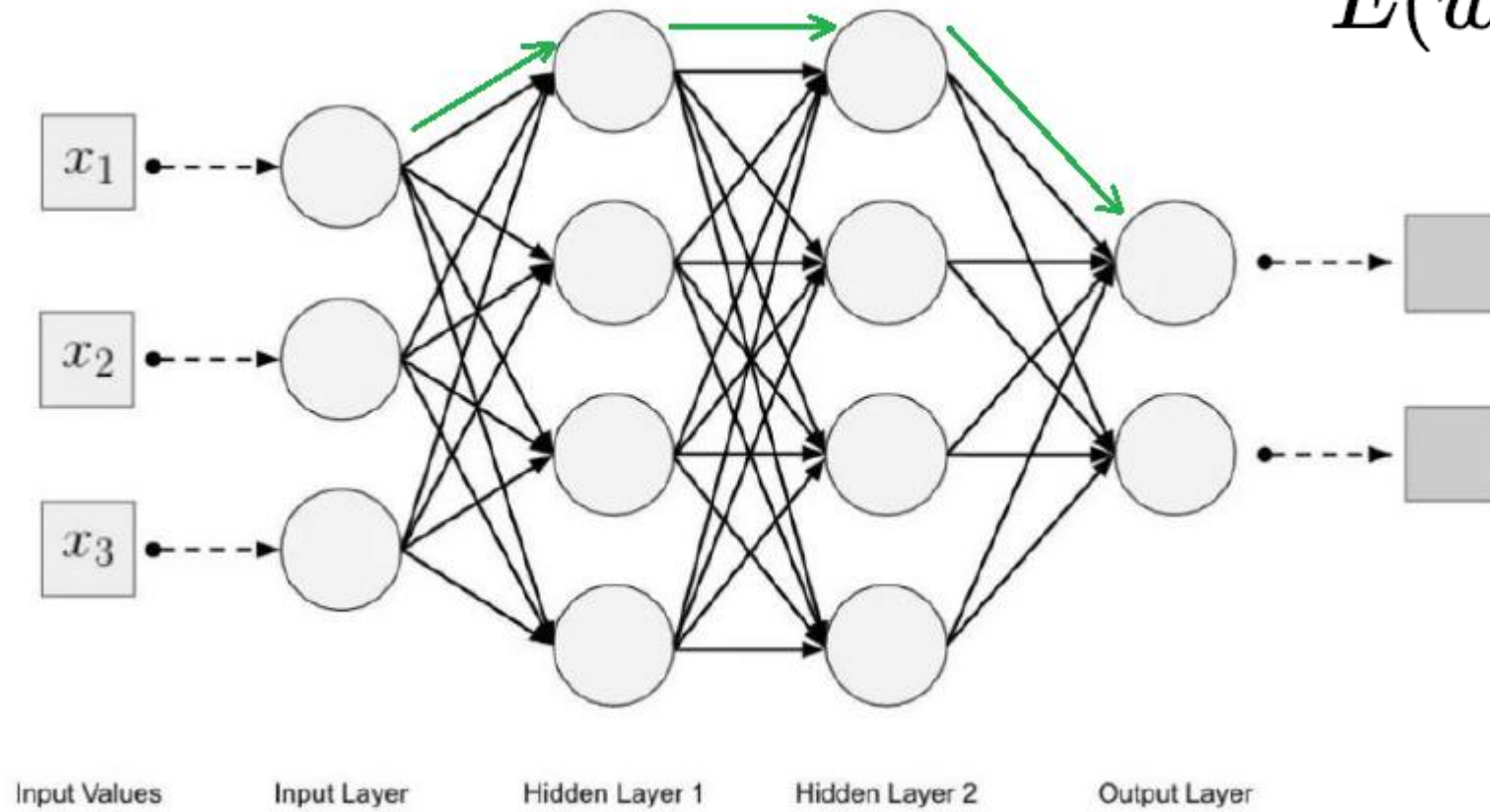


$$E(w) = \sum_i [o_i - t_i]^2$$

> Слои DNN

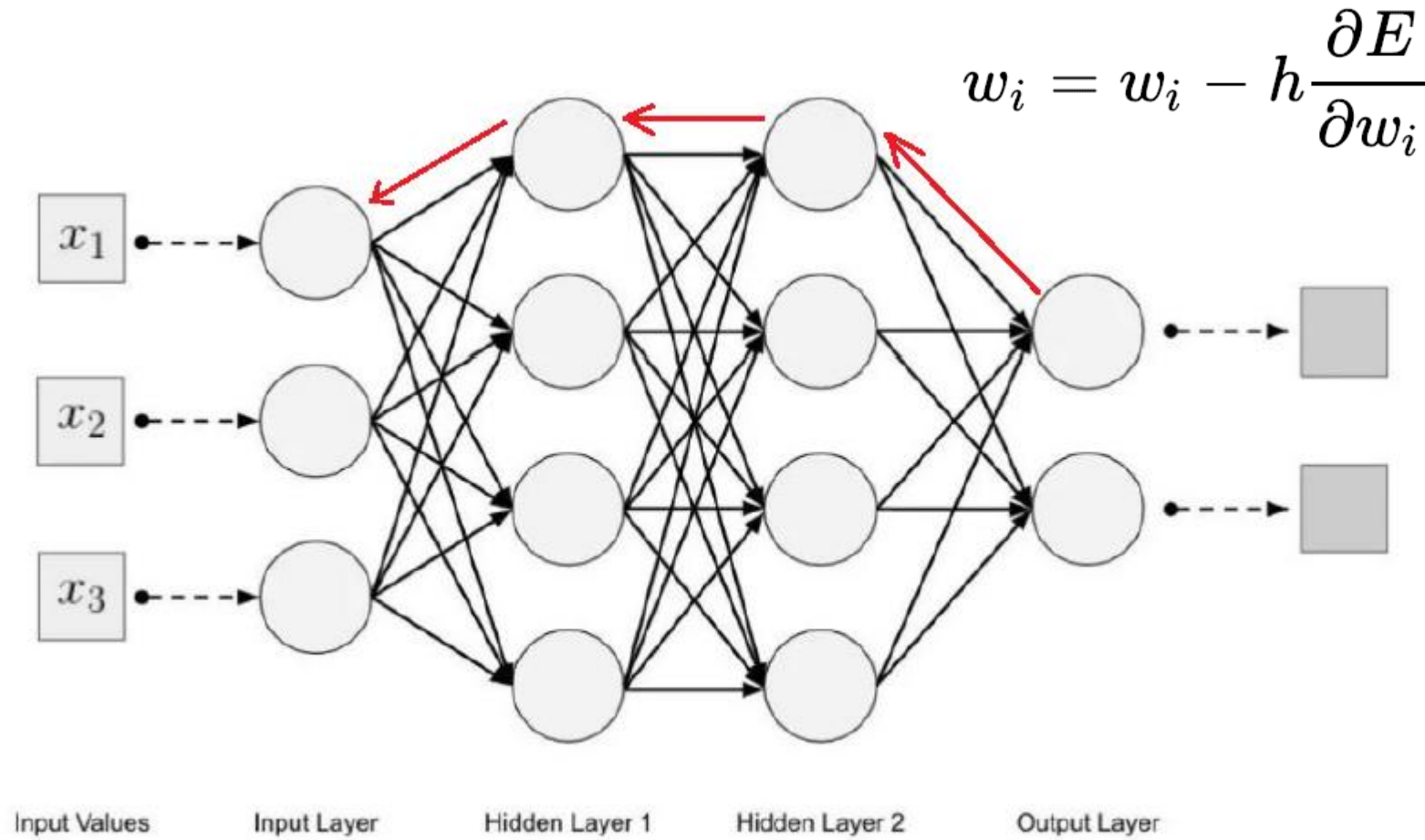


> Forward Propagation



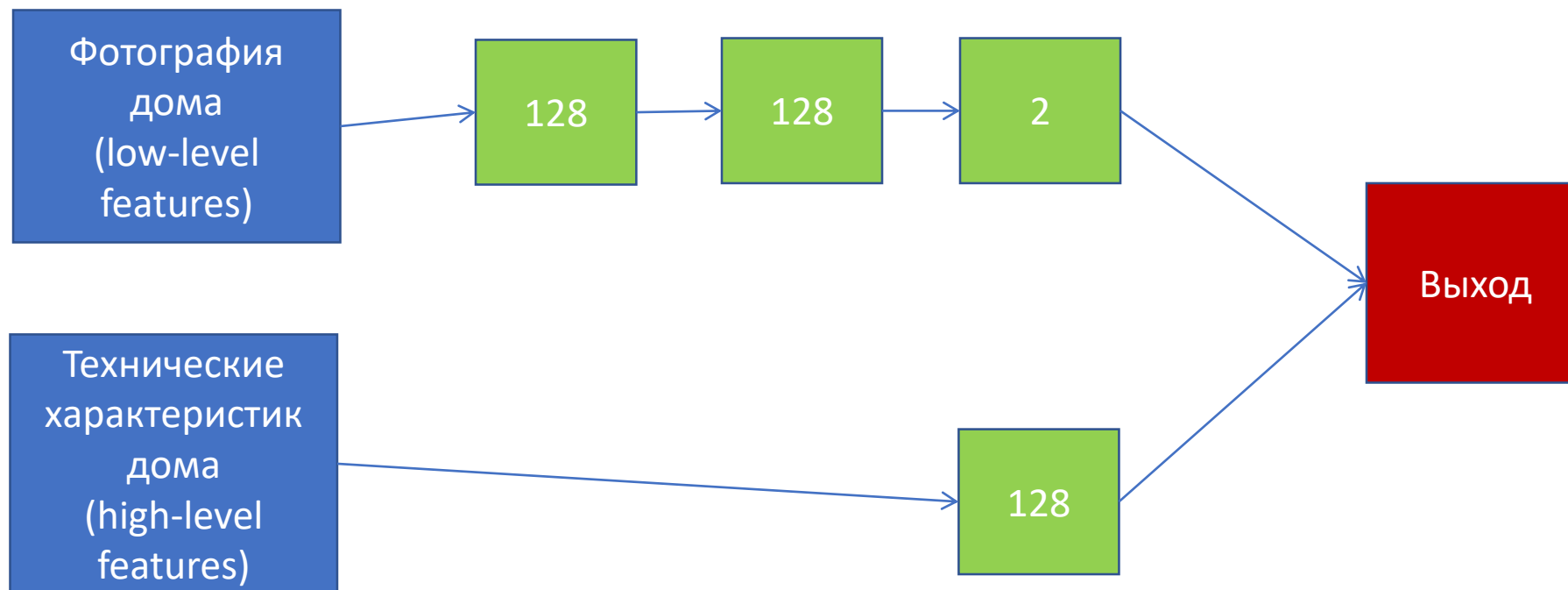
$$E(w) = \sum_i [o_i - t_i]^2$$

> Backward Propagation



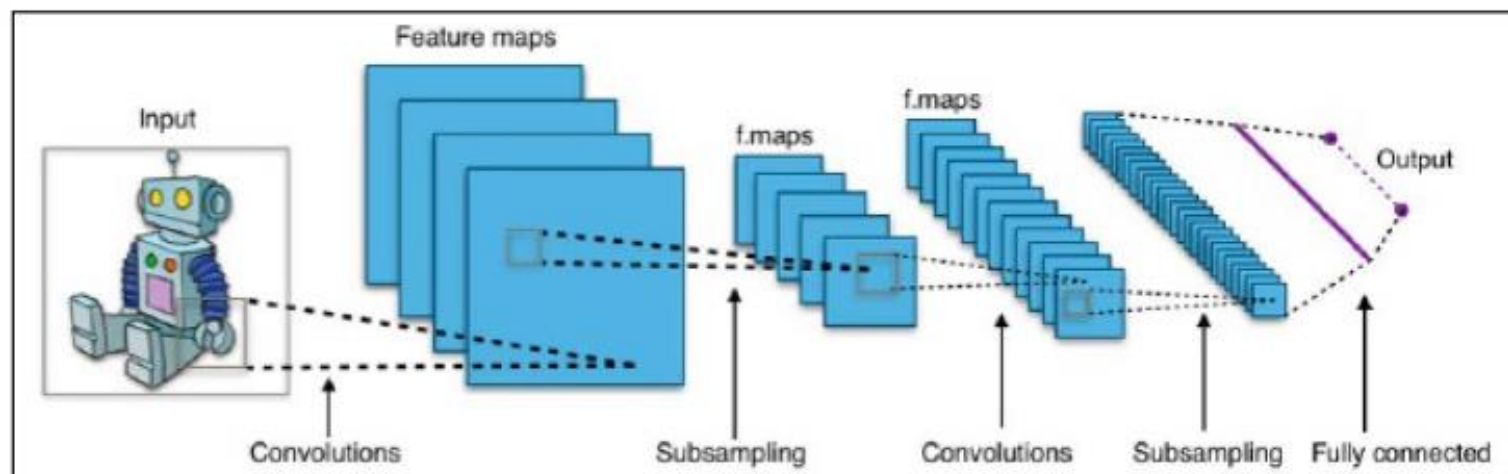
$$E(w) = \sum_i [o_i - t_i]^2$$

> Как учатся Deep Learning модели



> Сверточная нейронная сеть

Учет геометрии



Свертка

1	1	1	0	0
0	1	1	1	0
0	0	1	1	1
0	0	1	1	0
0	1	1	0	0

K

1	0	1
0	1	0
1	0	1

Convolved

4	3	4
2	4	3
2	3	4

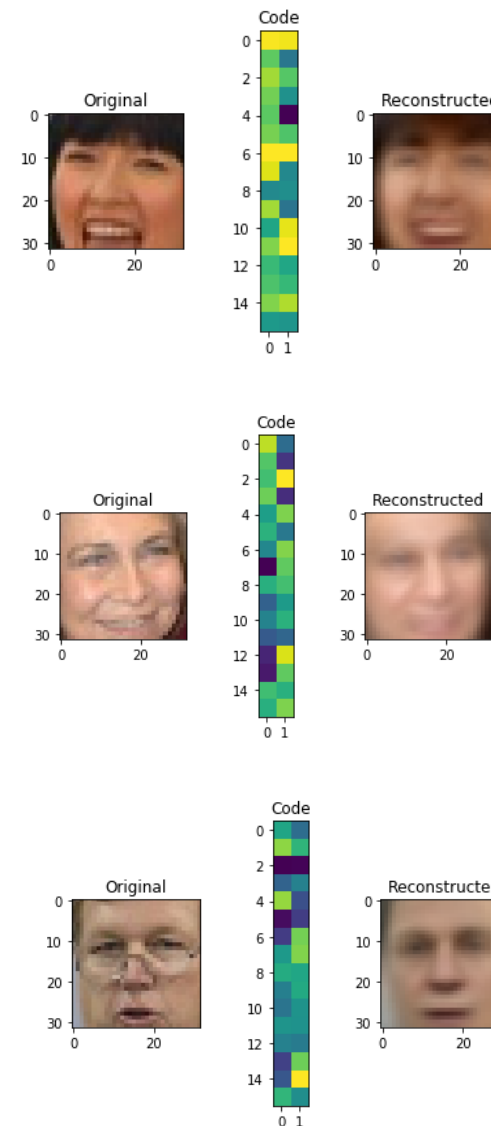
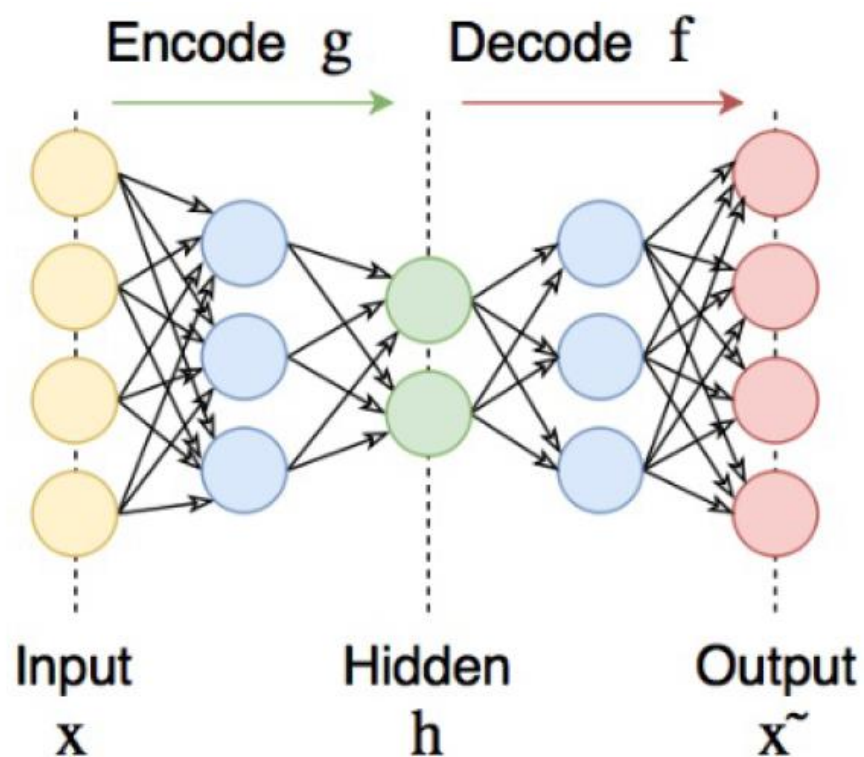
Pooling

1	0	3	6
2	4	5	2
2	6	2	0
3	4	1	7

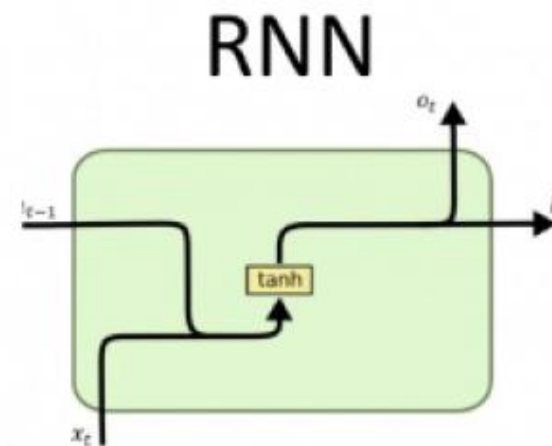
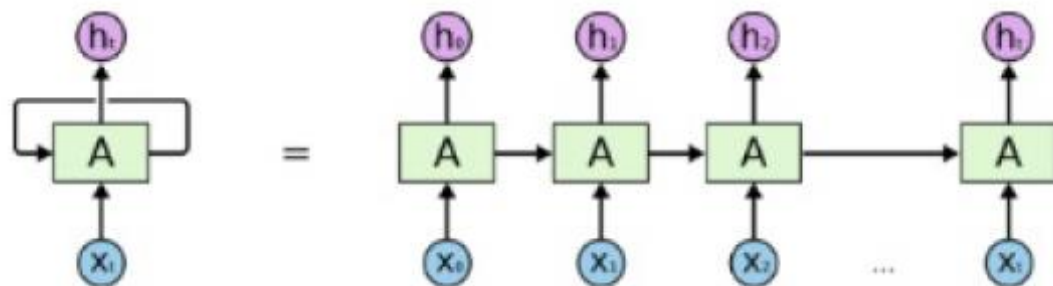
4	6
6	7

> Автокодировщик

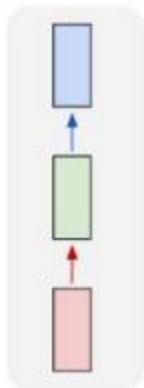
Сжатие информации



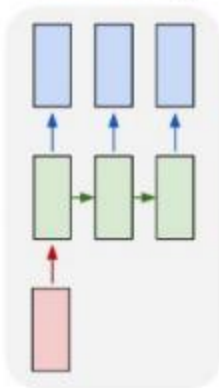
> Рекуррентные сети



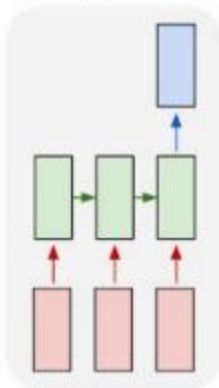
one to one



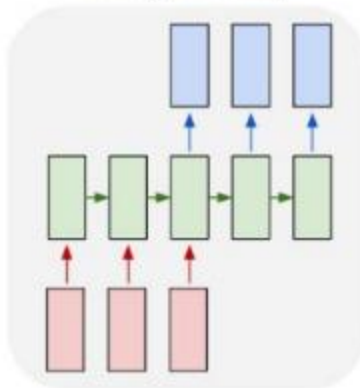
one to many



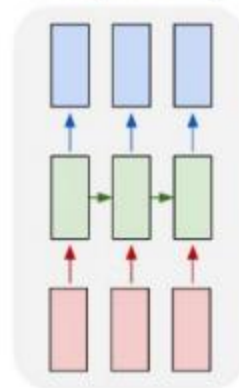
many to one



many to many



many to many



> Области применения глубокого обучения

- Обучение с учителем: классификация, регрессия/прогноз
- Обучение без учителя: понижение размерности, кластеризация, поиск паттернов (режимов, аномалий)
- Computer Vision: поиск объектов, сегментация
- NLP: моделирование языка, анализ тональности текстов, NER, POS tagging
- Генерация объектов (текстов, звуков, изображений, видео)

> Построение эффективной модели машинного обучения

- исследование процесса, определение проблемы и постановка задачи,
- сбор и предварительная обработка данных,
- выбор модели, выбор технологической и бизнес метрик оценки качества модели,
- обучение модели,
- получение и анализ результатов,
- настройка модели
- внедрение в production
- эксплуатация

Библиотеки и фреймворки

- pandas, numpy, matplotlib, scipy
- sklearn
- TensorFlow
- PyTorch
- Caffe
- ML.NET
-

> Пример: регрессия (стоимость квартир)

- `sklearn.datasets.load_boston`
- Определить как влияют факторы на цену, задача множественной регрессии

```
: from sklearn.datasets import load_boston
data = load_boston()
dir(data), data.data.shape

: ([('DESCR', 'data', 'feature_names', 'filename', 'target'], (506, 13))
```

```
: df = pd.DataFrame(data.data, columns=data.feature_names)
df['Price'] = data.target
df
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT	Price
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	4.98	24.0
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	9.14	21.6
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	4.03	34.7
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	2.94	33.4
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	5.33	36.2
...
501	0.06283	0.0	11.93	0.0	0.573	6.593	69.1	2.4786	1.0	273.0	21.0	391.99	9.67	22.4
502	0.04527	0.0	11.93	0.0	0.573	6.120	76.7	2.2875	1.0	273.0	21.0	396.90	9.08	20.6
503	0.08076	0.0	11.93	0.0	0.573	6.976	91.0	2.1675	1.0	273.0	21.0	396.90	5.64	23.9
504	0.10959	0.0	11.93	0.0	0.573	6.794	89.3	2.3889	1.0	273.0	21.0	393.45	6.48	22.0
505	0.04741	0.0	11.93	0.0	0.573	6.030	80.8	2.5050	1.0	273.0	21.0	396.90	7.88	11.9

```
: from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(data.data, data.target)
model.score(data.data, data.target)
```

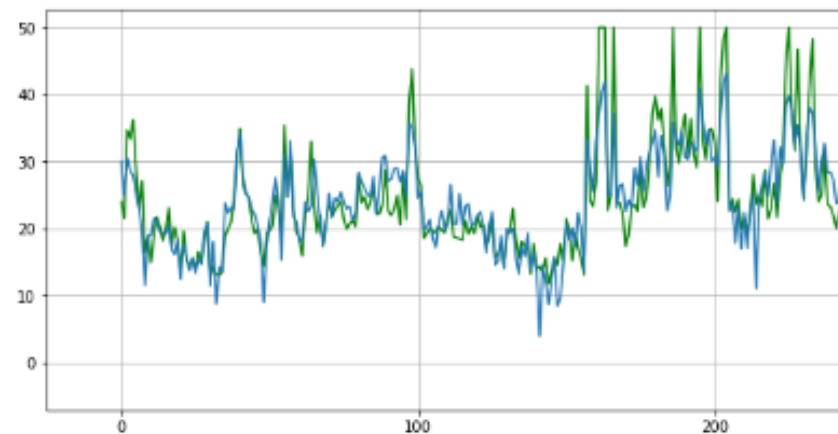
```
: 0.7406426641094095
```

```
: idx_ = 100
result = model.predict([data.data[idx_]])
print(result, data.target[idx_])
```

```
[24.58022019] 27.5
```

```
: prediction = model.predict(data.data)
```

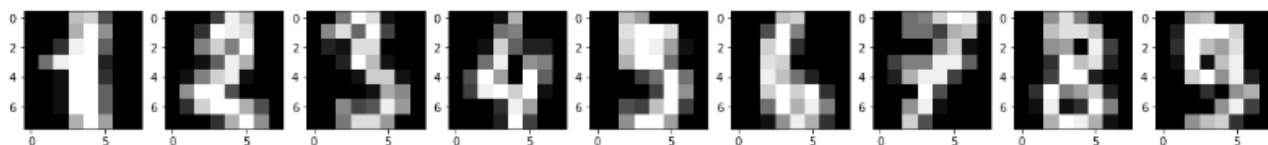
```
plt.figure(figsize=(20,5))
plt.plot(data.target, c='g')
plt.plot(prediction)
plt.grid(True)
plt.show()
```



> Пример: распознавание цифр

- `sklearn.datasets.load_digits`
рукописные цифры
- Массивы из 64 значений,
необходимо распознавать цифры
от 0 до 9

```
: plt.figure(figsize=(20,5))
for i in range(1,10):
    plt.subplot(100+90+i)
    plt.imshow(data.images[i], cmap='gray')
plt.show()
```



```
: from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(data.data, data.target)
X_train.shape, y_train.shape

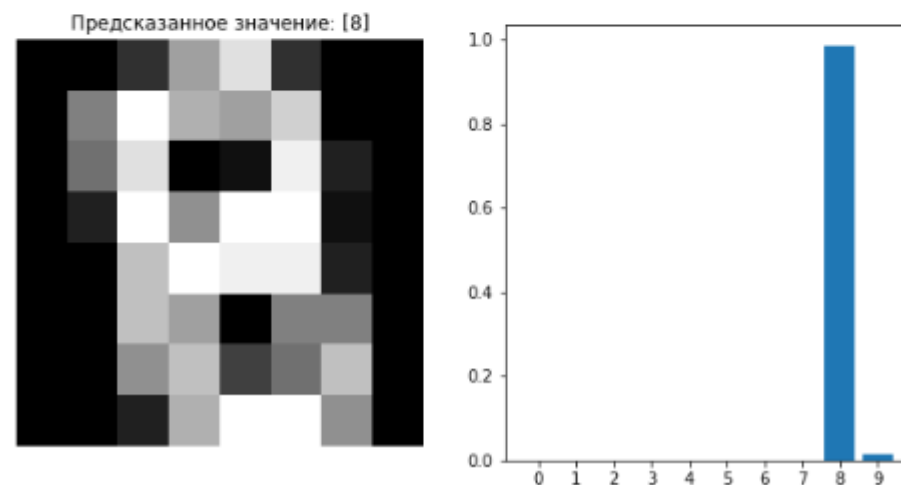
: ((1347, 64), (1347,))

: model.fit(X_train, y_train)
model.score(X_test, y_test)

idx_ = 500
plt.figure(figsize=(10,5))

plt.subplot(121)
plt.imshow(data.images[idx_], cmap='gray')
plt.title("Предсказанное значение: {}".format(model.predict([data.data[idx_]])))
plt.axis(False)

plt.subplot(122)
plt.bar(range(10), height=model.predict_proba([data.data[idx_]])[0])
plt.xticks(range(10))
plt.show()
```



➤ Пример 3: разделение на кластеры

- load_iris

```
: from sklearn.datasets import load_iris
data = load_iris()

: data.target_names
array(['setosa', 'versicolor', 'virginica'], dtype='<U10')

: pd.DataFrame(data.data, columns=data.feature_names)

:
   sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (cm)
0                5.1                3.5                1.4                0.2
1                4.9                3.0                1.4                0.2
2                4.7                3.2                1.3                0.2
3                4.8                3.1                1.5                0.2
4                5.0                3.6                1.4                0.2
...                ...                ...                ...                ...
145               6.7                3.0                5.2                2.3
146               6.3                2.5                5.0                1.9
147               6.5                3.0                5.2                2.0
148               6.2                3.4                5.4                2.3
149               5.9                3.0                5.1                1.8

150 rows x 4 columns

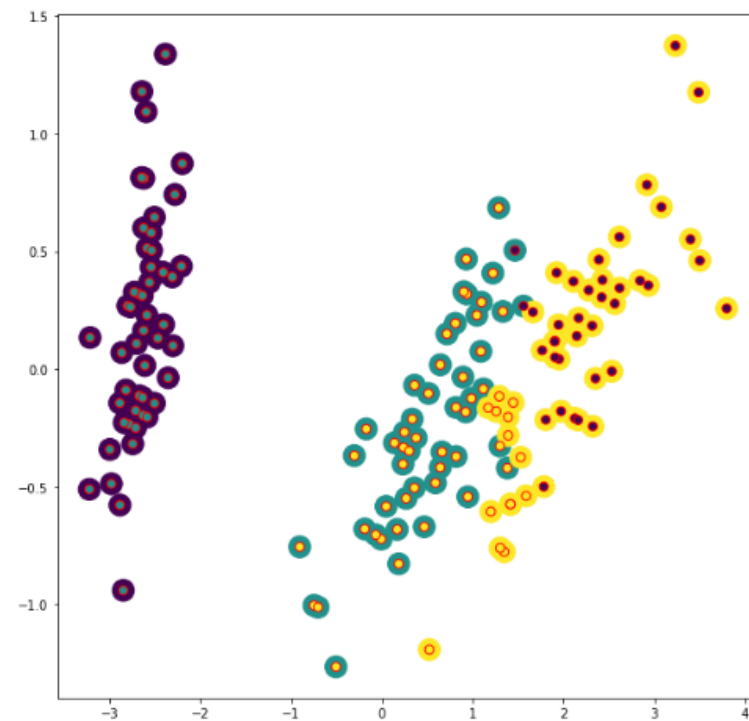
: from sklearn.cluster import KMeans
model = KMeans(3)
model.fit(data.data)
model.labels_

: array([[1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2,
0, 0, 0, 2, 2, 0, 0, 0, 0, 2, 0, 2, 0, 2, 0, 2, 0, 2, 2, 0, 0, 0,
0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0,
0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 0, 0, 2, 0, 0, 0, 0, 2, 0, 0, 0, 2], dtype=int32)
```

```
: from sklearn.decomposition import PCA

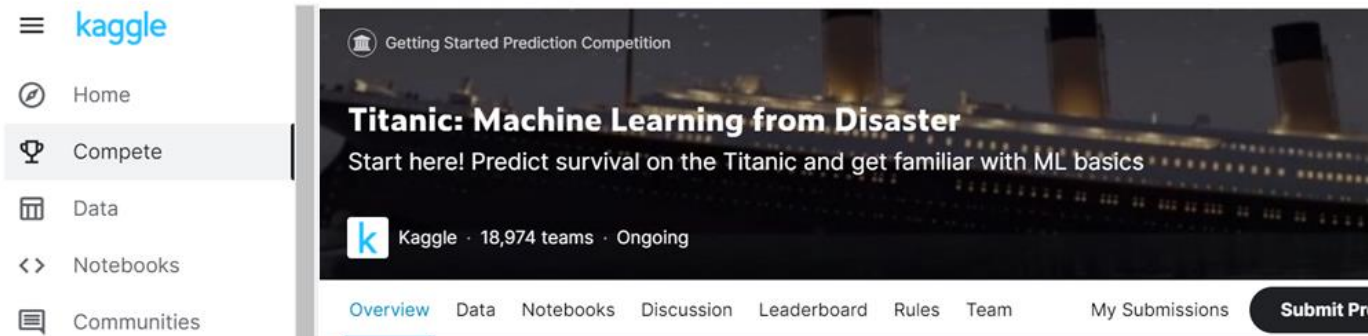
new_data = PCA(2).fit_transform(data.data)

plt.figure(figsize=(10,10))
plt.scatter(new_data[:,0], new_data[:,1], s=300, c=data.target)
plt.scatter(new_data[:,0], new_data[:,1], s=50, linewidths=1, edgecolors='r', c=model.labels_)
plt.show()
```



> Пример 4: Titanic Disaster

- <https://www.kaggle.com/c/titanic>
- Предсказание вероятности гибели пассажира «Титаника»
- Основано на реальных событиях (данных)
- Задача классификации
- Есть данные для 891 пассажира (тренировочная выборка), надо предсказать для тестовой выборки





> Как прокачаться в теме data science

- курсы (coursera.org, stepik.com, <https://github.com/yurichernyshov/Data-Science-Course-USURT>)
- статьи (habr.com, medium.com)
- telegram-каналы, сообщества в социальных сетях (FB, VC), участие в сообществе (ODS.ai)
- книги
- практика
 - kaggle.com – популярная платформа для соревнований, обмена опытом в области работы с данными и машинного обучения
 - работа в google collab


> Рекомендованные курсы coursera.org

The Coursera logo, consisting of the word "coursera" in white lowercase letters on a blue rectangular background.

- “Математика и Python для анализа данных” <https://www.coursera.org/learn/mathematics-and-python/home/welcome>
- “Введение в машинное обучение” <https://www.coursera.org/learn/vvedenie-mashinnoe-obuchenie/home/welcome>
- “Основы программирования на python” <https://www.coursera.org/learn/python-osnovy-programmirovaniya/home/welcome>



Спасибо за внимание!
Вопросы?



Чернышов Юрий, ychernyshov@ussc.ru, @yuchernyshov
Астафьева Анна, @astafevaanny

