НИЯУ МИФИ

Методы статистического анализа

Основы МО и примеры в ФЭЧ

Ильясов Айдар
Аспирант 2 курса НИЦ КИ, МНС ЛФРП ОФН КИ

План занятий

Занятие 1 (27.10.2022) - Теоретическое

•Основы машинного обучения
•Основные модели машинного обучения

Занятие 2 (03.11.2022) – Практическое, вариативное

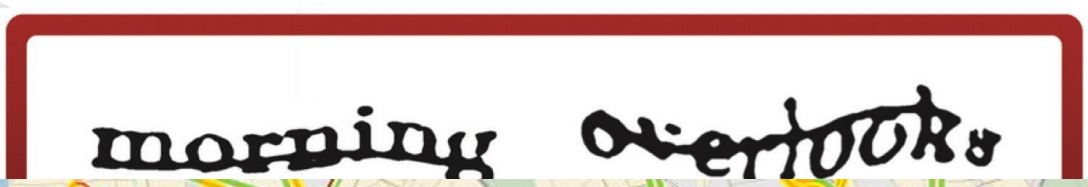
- «Усложнённые» модели машинного обучения
- Использование машинного обучения в физике частиц
  - C++
  - Python
- Пример результата работы машинного обучения на C++
- Пример(ы) результата работы машинного обучения на Python
- Источники информации
- Другое? ilyasovaid@yandex.ru

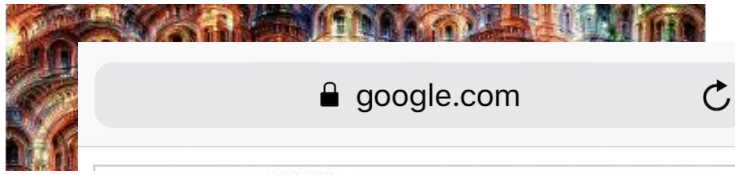
Методы статистического анализа

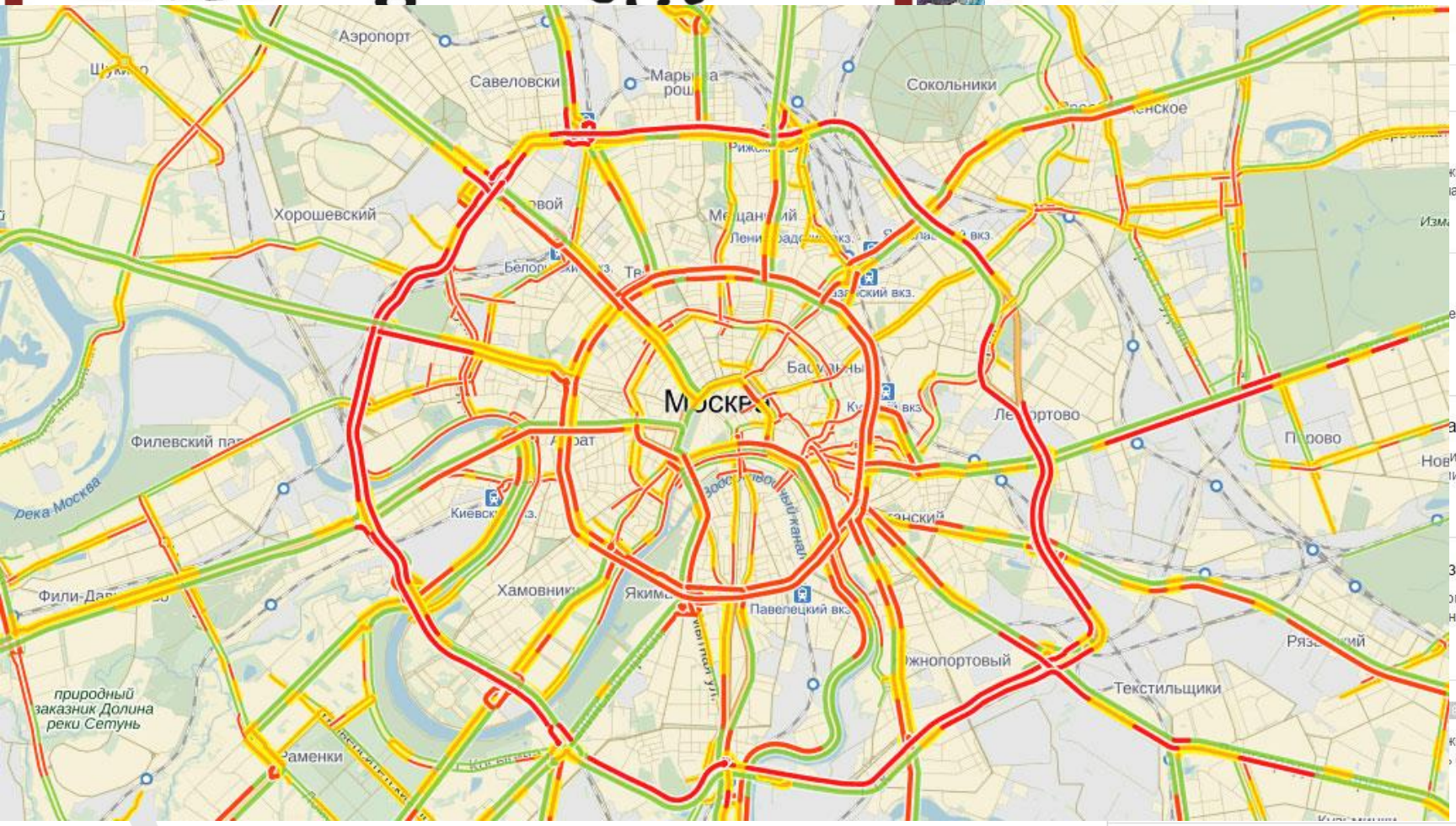
Часть 1: Основы

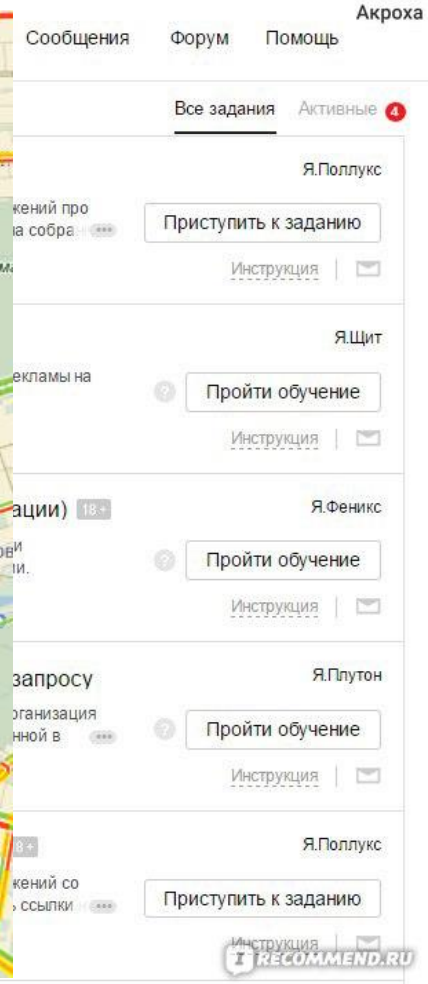
27.10.2022

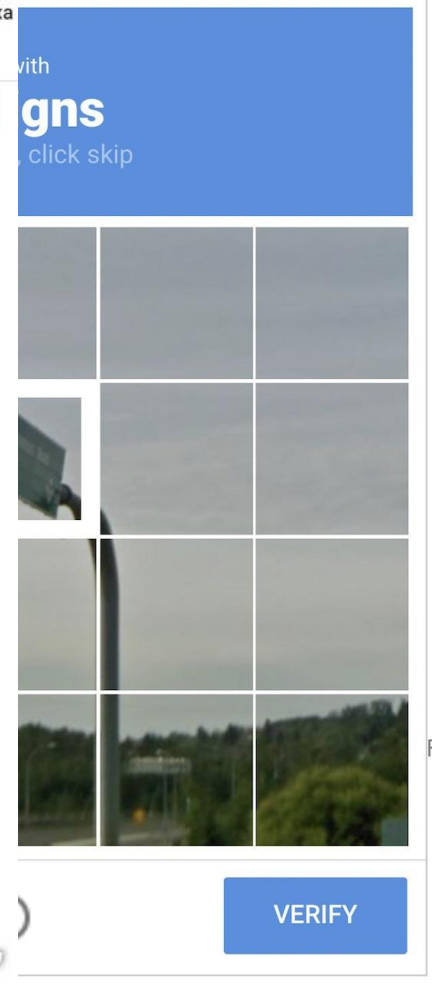
Примеры из жизни





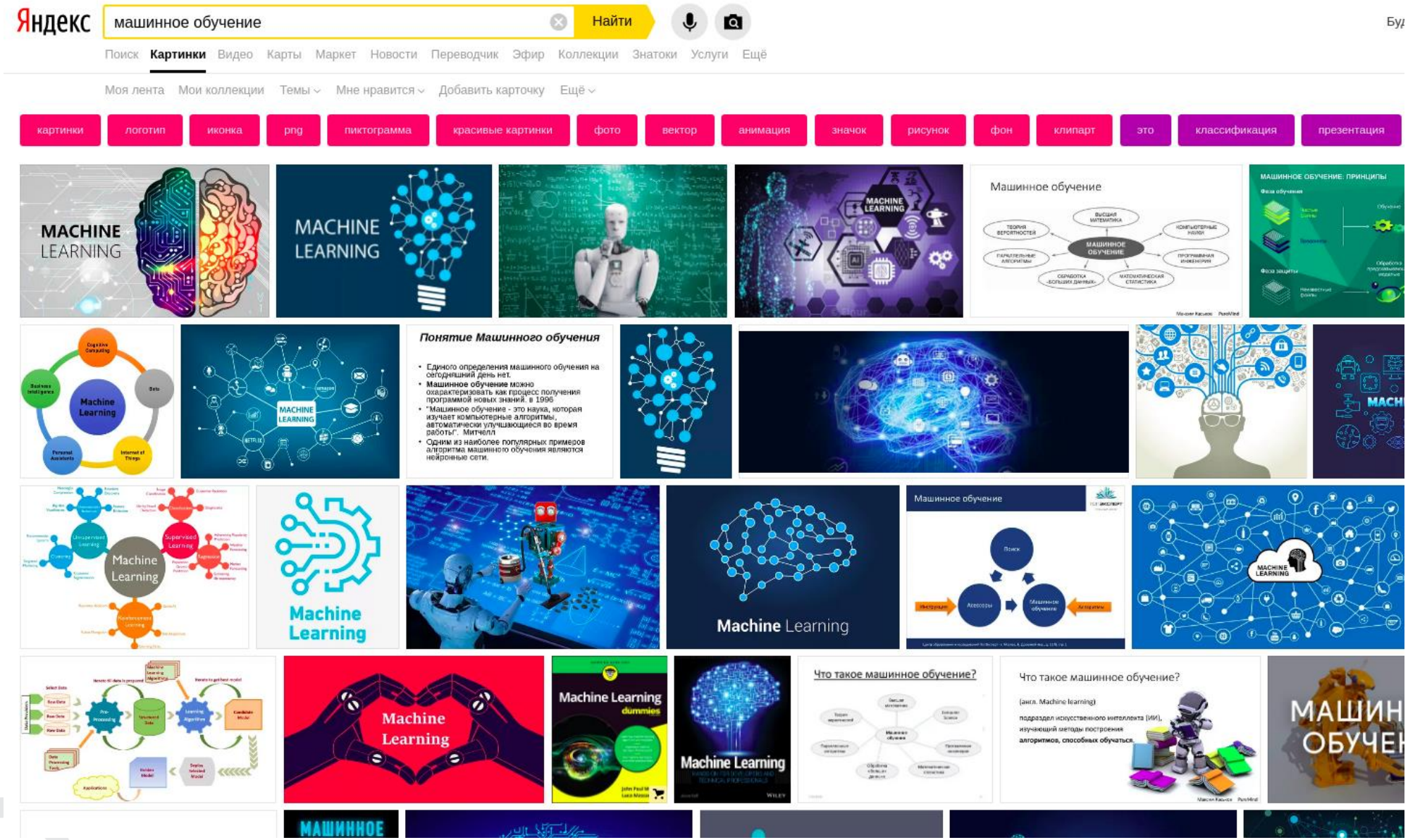




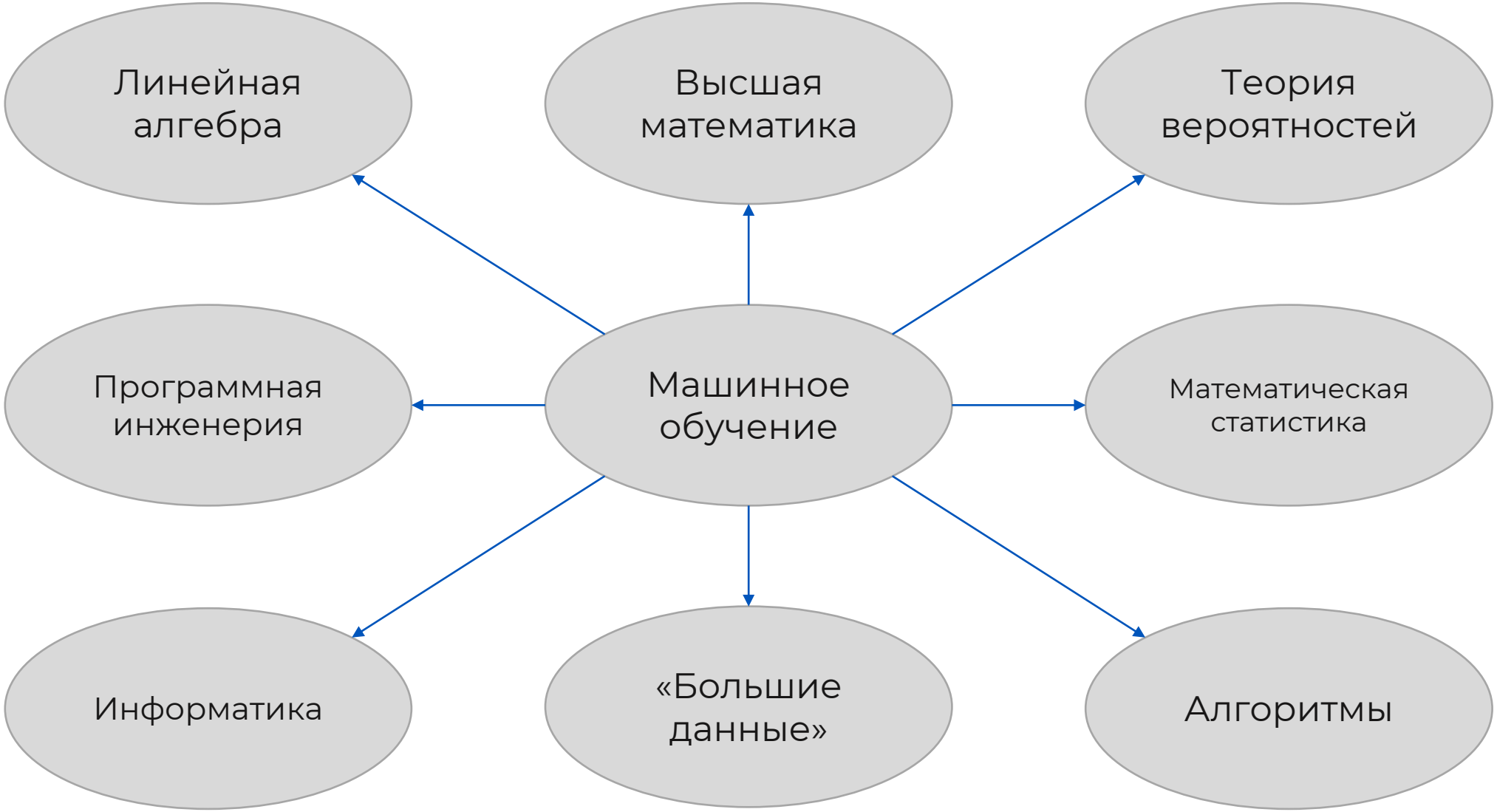


Примеры из жизни

Что такое машинное обучение?

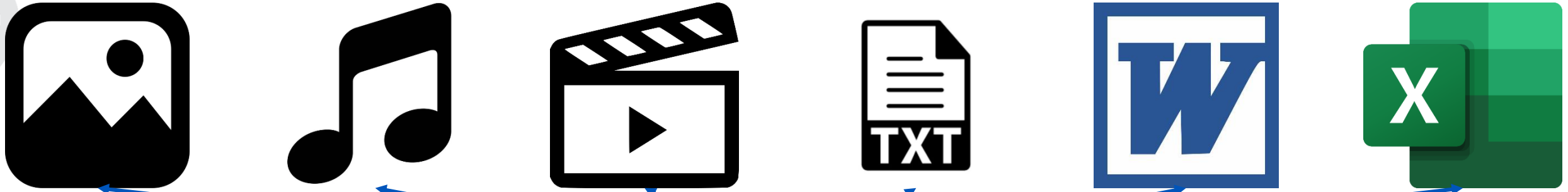


Что такое машинное обучение?



Что такое машинное обучение?

Типы данных



Машинное обучение

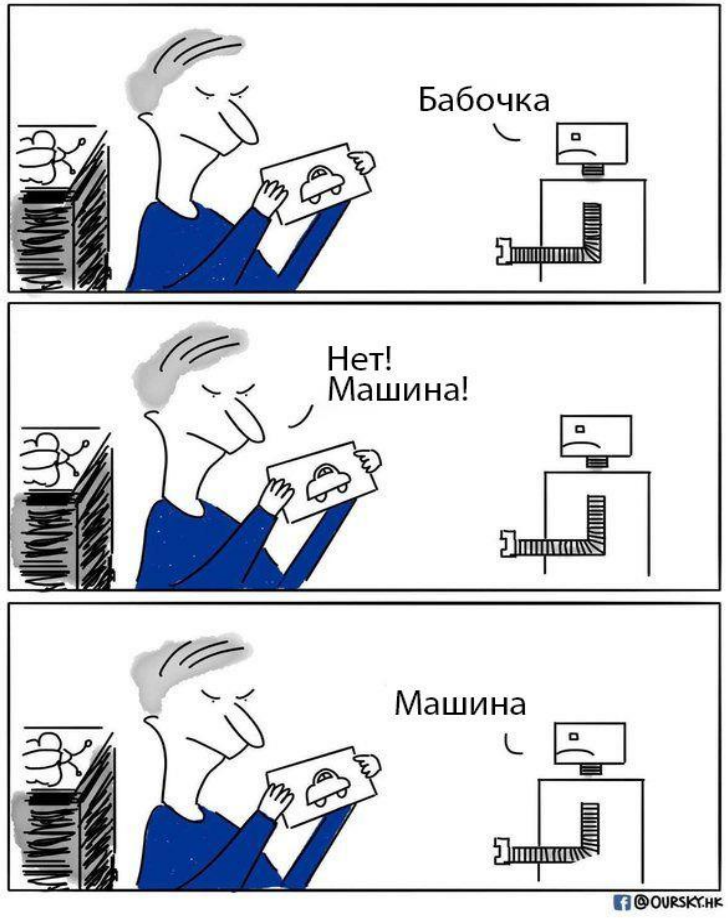
- Кошка/собака
- Дождь/гром
- Птица/самолет
- Протон/нейтрон

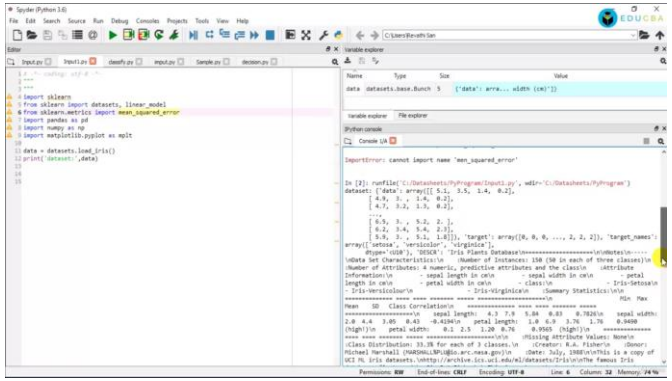
А на самом деле?

linear.py

```python
def Cost_Function(theta):
    sumOfSqErr = 0
    for i in range(0,8):
        x = X[i]
        hi = Hypothesis(theta,x)
        yi = y[i]
        diff = (hi - yi)
        sqErr = diff**2
        sumOfSqErr += sqErr
    m = len(y) + 1
    constant = float(1) / float(2*m)
    J = constant * sumOfSqErr
    return J

```

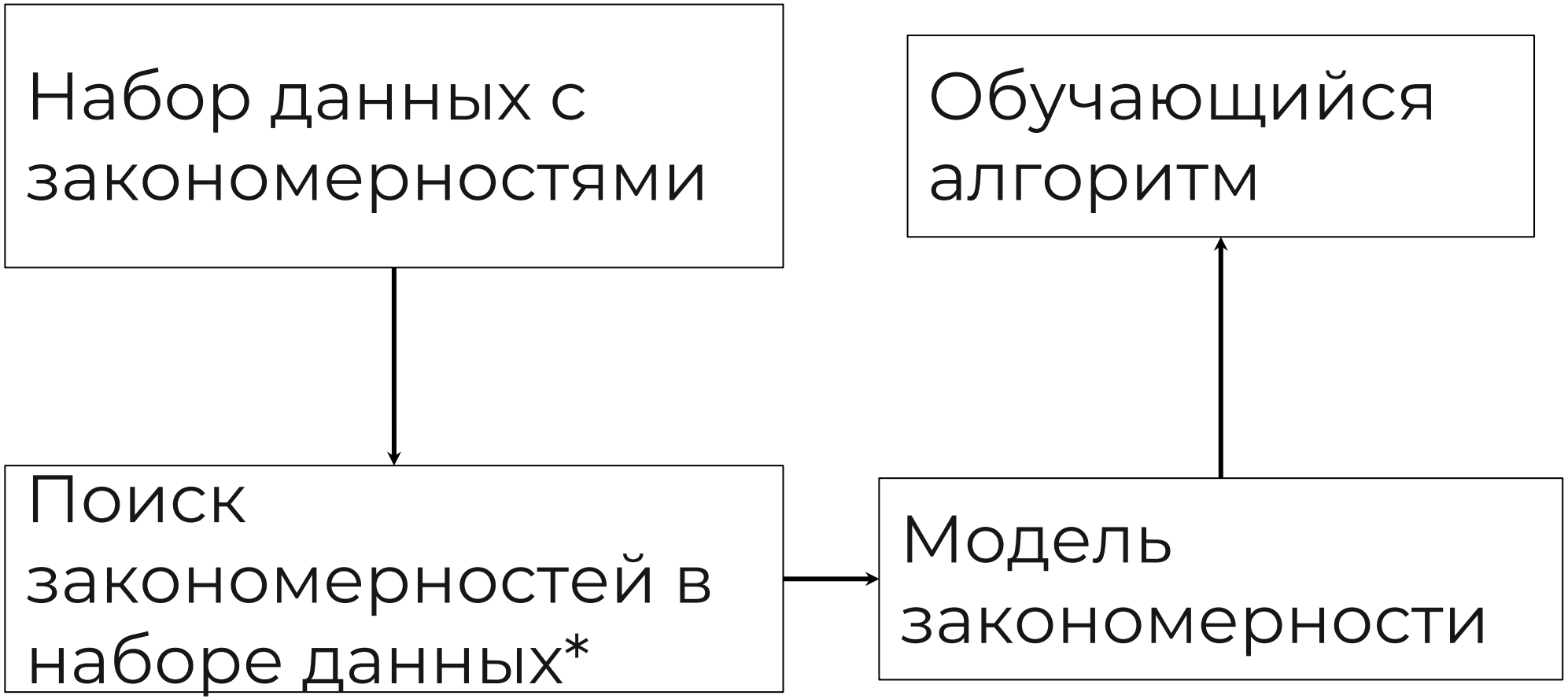
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-115-74de34368e06> in <module>
      1 logreg = LogisticRegression()
      2
----> 3 logreg.fit(X_train_dtm, y_train)

d:\programs\python\lib\site-packages\sklearn\linear_model\logistic.py in fit(self, X, y, sample_weight)
   1282
   1283         X, y = check_X_y(X, y, accept_sparse='csr', dtype=_dtype, order="C",
-> 1284                          accept_large_sparse=solver != 'liblinear')
   1285         check_classification_targets(y)
   1286         self.classes_ = np.unique(y)

d:\programs\python\lib\site-packages\sklearn\utils\validation.py in check_X_y(X, y, accept_sparse, accept_large_sparse, dtype,
 order, copy, force_all_finite, ensure_2d, allow_nd, multi_output, ensure_min_samples, ensure_min_features, y_numeric, warn_on_
dtype, estimator)
    755         y = y.astype(np.float64)
    756
--> 757     check_consistent_length(X, y)
    758
    759     return X, y

d:\programs\python\lib\site-packages\sklearn\utils\validation.py in check_consistent_length(*arrays)
    228     if len(uniques) > 1:
    229         raise ValueError("Found input variables with inconsistent numbers of"
--> 230                          " samples: %r" % [int(l) for l in lengths])
    231
    232

ValueError: Found input variables with inconsistent numbers of samples: [11, 133401]
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-115-74de34368e06> in <module>
      1 logreg = LogisticRegression()
      2
----> 3 logreg.fit(X_train_dtm, y_train)

d:\programs\python\lib\site-packages\sklearn\linear_model\logistic.py in fit(self, X, y, sample_weight)
   1282
   1283         X, y = check_X_y(X, y, accept_sparse='csr', dtype=_dtype, order="C",
-> 1284                          accept_large_sparse=solver != 'liblinear')
   1285         check_classification_targets(y)
   1286         self.classes_ = np.unique(y)

d:\programs\python\lib\site-packages\sklearn\utils\validation.py in check_X_y(X, y, accept_sparse, accept_large_sparse, dtype,
 order, copy, force_all_finite, ensure_2d, allow_nd, multi_output, ensure_min_samples, ensure_min_features, y_numeric, warn_on_
dtype, estimator)
    755         y = y.astype(np.float64)
    756
--> 757     check_consistent_length(X, y)
    758
    759     return X, y

d:\programs\python\lib\site-packages\sklearn\utils\validation.py in check_consistent_length(*arrays)
    228     if len(uniques) > 1:
    229         raise ValueError("Found input variables with inconsistent numbers of"
--> 230                          " samples: %r" % [int(l) for l in lengths])
    231
    232

ValueError: Found input variables with inconsistent numbers of samples: [11, 133401]
```





```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
<ipython-input-115-74de34368e06> in <module>
      1 logreg = LogisticRegression()
      2
----> 3 logreg.fit(X_train_dtm, y_train)

d:\programs\python\lib\site-packages\sklearn\linear_model\logistic.py in fit(self, X, y, sample_weight)
   1282
   1283         X, y = check_X_y(X, y, accept_sparse='csr', dtype=_dtype, order="C",
-> 1284                          accept_large_sparse=solver != 'liblinear')
   1285         check_classification_targets(y)
   1286         self.classes_ = np.unique(y)

d:\programs\python\lib\site-packages\sklearn\utils\validation.py in check_X_y(X, y, accept_sparse, accept_large_sparse, dtype,
 order, copy, force_all_finite, ensure_2d, allow_nd, multi_output, ensure_min_samples, ensure_min_features, y_numeric, warn_on_
dtype, estimator)
    755         y = y.astype(np.float64)
    756
--> 757     check_consistent_length(X, y)
    758
    759     return X, y

d:\programs\python\lib\site-packages\sklearn\utils\validation.py in check_consistent_length(*arrays)
    228     if len(uniques) > 1:
    229         raise ValueError("Found input variables with inconsistent numbers of"
--> 230                          " samples: %r" % [int(l) for l in lengths])
    231
    232

ValueError: Found input variables with inconsistent numbers of samples: [11, 133401]
```

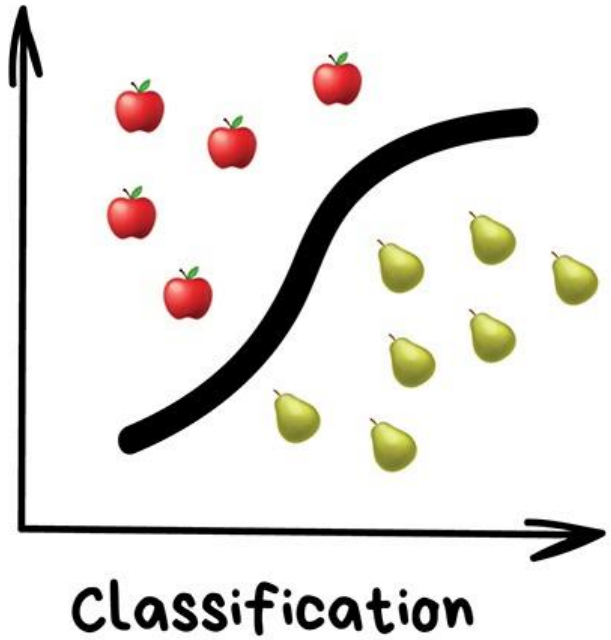
А на самом деле?



*Умный маленький человечек внутри компьютера

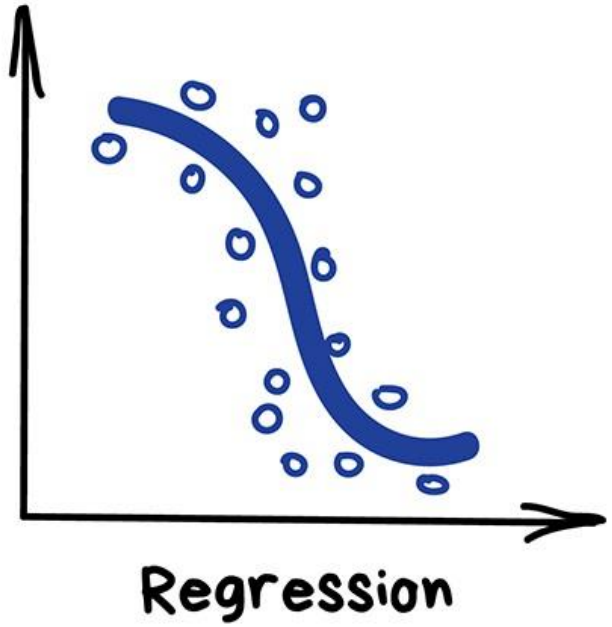
Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
    - Кошка-собака
    - 1-0
    - Птица-Самолёт-Вертолет
    - Мальчик-Девочка
    - Цветок-Животное



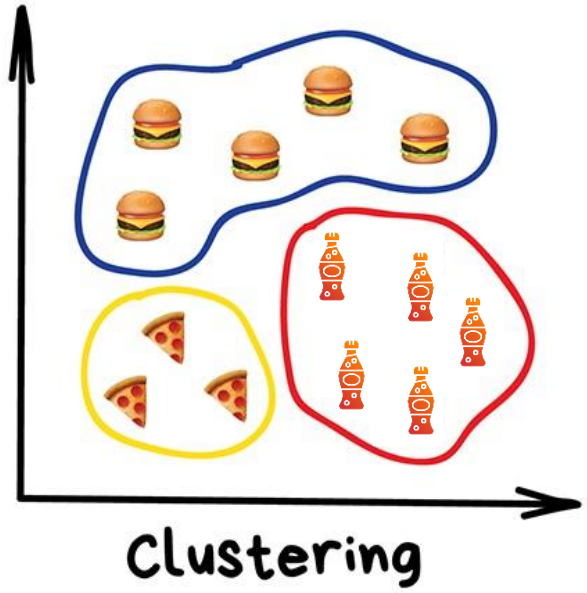
Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
  - Регрессия
    - Какого цвета конфета?
    - Сколько будет стоить квартира?
    - Какая будет цена на нефть в 2025 году?



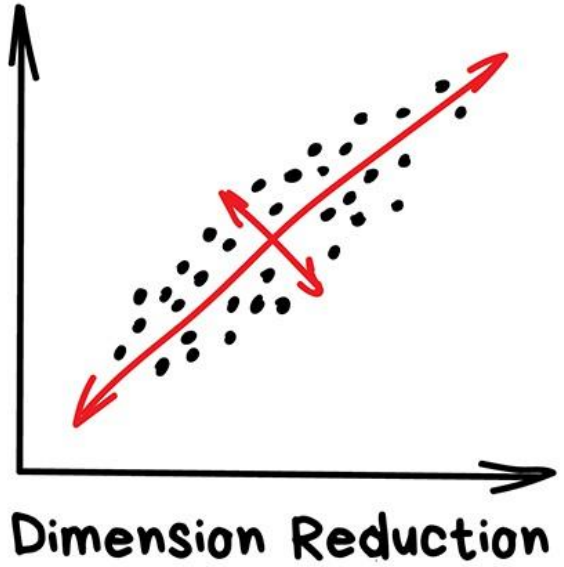
Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
  - Регрессия
- **Без учителя:**
  - Кластеризация
    - Машинное обучение - финансы - игры
    - Разложить похожие вещи по кучкам



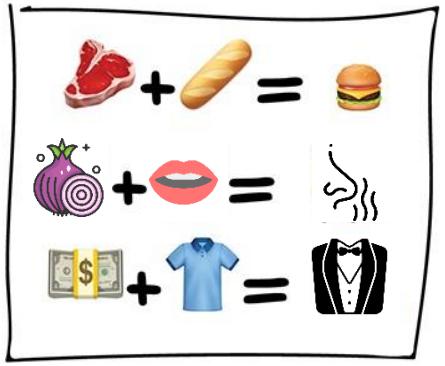
Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
  - Регрессия
- **Без учителя:**
  - Кластеризация
  - Задача уменьшения размерности
    - Анализ параметров наборов данных



Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
  - Регрессия
- **Без учителя:**
  - Кластеризация
  - Задача уменьшения размерности
  - Задача поиска правил
    - Анализ товаров покупаемых вместе
    - Прогноз распродаж



Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
  - Регрессия
- **Без учителя:**
  - Кластеризация
  - Задача уменьшения размерности
  - Задача поиска правил

} Классическое обучение

Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
  - Регрессия
- **Без учителя:**
  - Кластеризация
  - Задача уменьшения размерности
  - Задача поиска правил

Классическое обучение

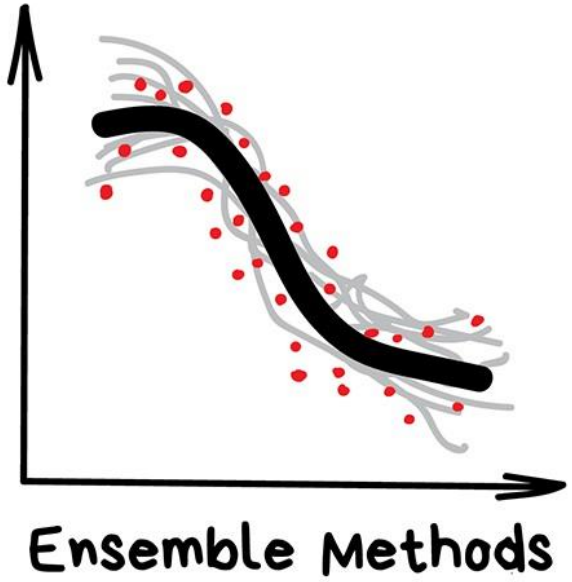
- **Обучение с подкреплением**

Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
  - Регрессия
- **Без учителя:**
  - Кластеризация
  - Задача уменьшения размерности
  - Задача поиска правил

Классическое обучение

- **Обучение с подкреплением**
- **Ансамбли**

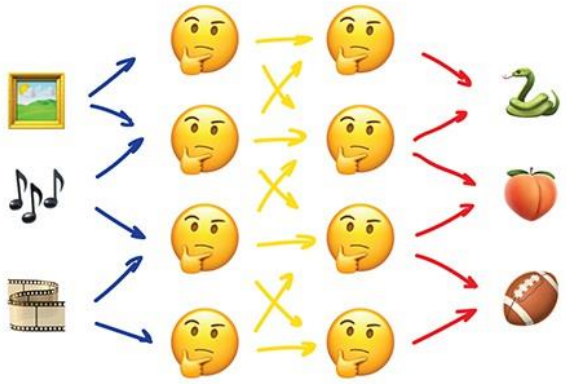


Основные типы задач машинного обучения

- **С учителем:**
  - Классификация
  - Регрессия
- **Без учителя:**
  - Кластеризация
  - Задача уменьшения размерности
  - Задача поиска правил

Классическое обучение

- **Обучение с подкреплением**
- **Ансамбли**
- **Нейросети**



«Иерархия потребностей» науки о данных

Факт:
У человека появилась идея применения МО **в новой области**

Вопрос:

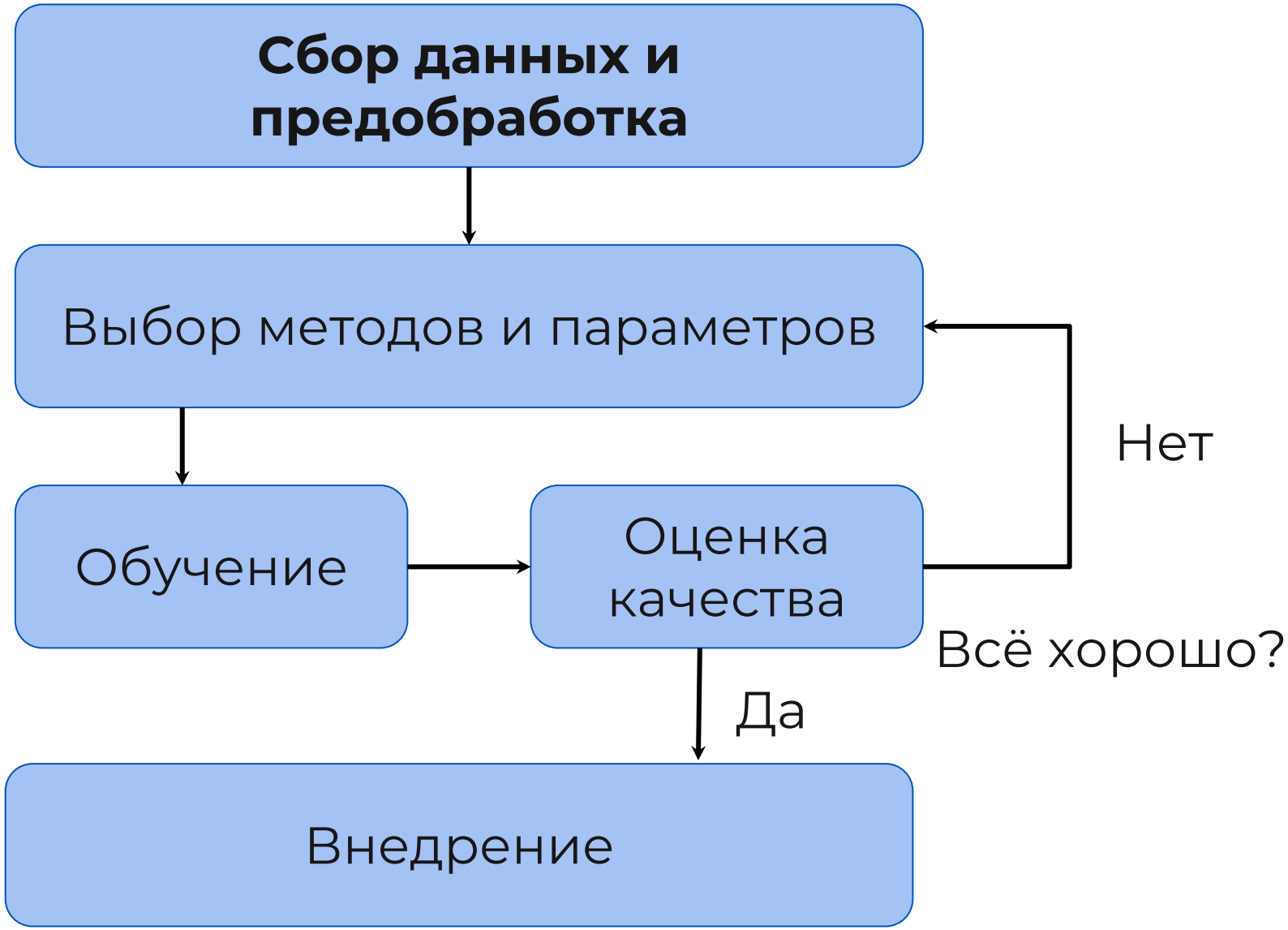
Что нужно сделать человеку чтобы **с нуля** разработать свой алгоритм машинного обучения?

«Иерархия потребностей» науки о данных

Выбор методов и параметров

Сбор и предобработка данных

Ход работы


Сбор данных и предобработка
Выбор методов и параметров
Обучение
Оценка качества
Нет
Всё хорошо?
Да
Внедрение

Сбор и предобработка данных

НИЯУ МИФИ

Сбор данных

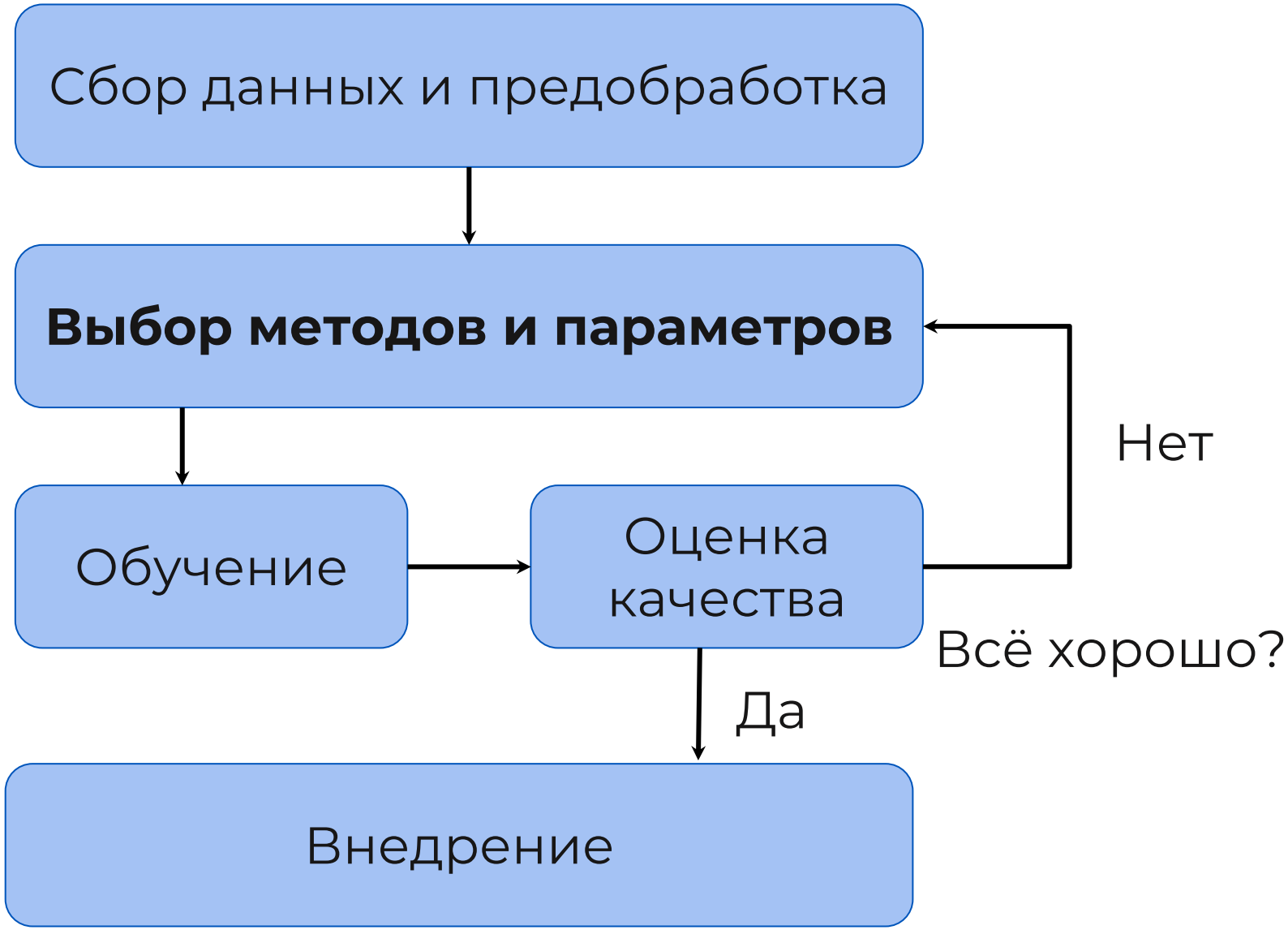
Сбор данных и предобработка

- MNIST
- Открытые базы данных
- Моделирование событий
- Данные с детекторов
- Каптча

Предобработка

- Исключение выбросов и NaN событий
- Исключение шумовых данных

Ход работы


Сбор данных и предобработка
Выбор методов и параметров
Обучение
Оценка качества
Нет
Всё хорошо?
Да
Внедрение

Какой алгоритм выбрать?

НИЯУ МИФИ

Выбор методов и параметров

«Базовый» список моделей машинного обучения

- Наивный байесовский классификатор
- Дерево решений
- Метод опорных векторов
- Регрессия
- Логистическая регрессия
- Принцип главных компонент
- Другие

Какой алгоритм выбрать?

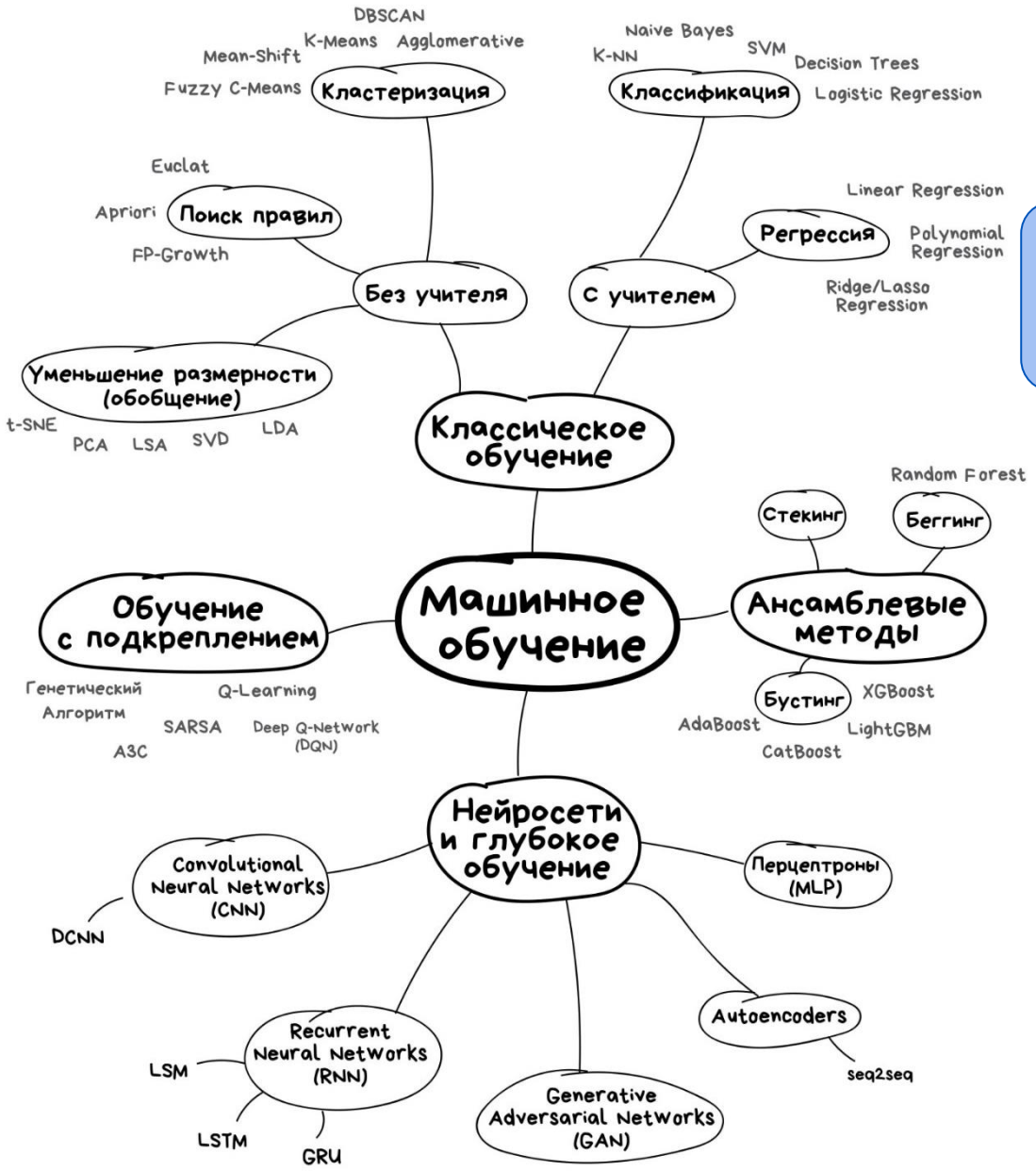
НИЯУ МИФИ

«Расширенный» список моделей машинного обучения

- Нейронная сеть
- Свёрточная нейронная сеть
- K-средних (kNN)
- Случайный лес
- Бустинг над деревьями решений
- Ансамбль моделей
- Другие

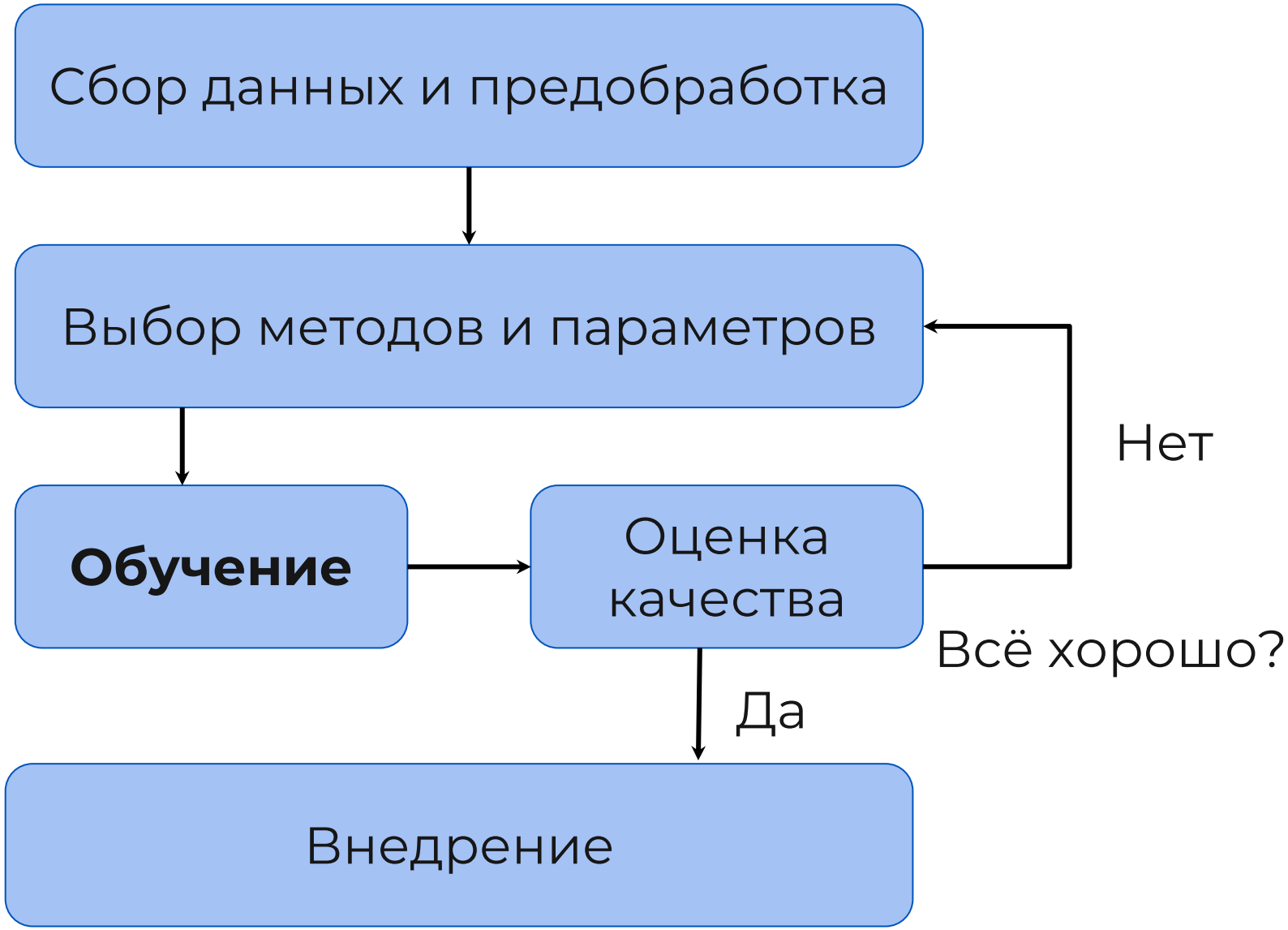
Выбор методов и параметров

Какой алгоритм выбрать?



Выбор методов и параметров

Ход работы


Сбор данных и предобработка
Выбор методов и параметров
Обучение
Оценка качества
Нет
Всё хорошо?
Да
Внедрение

Тренировка модели

Тренировка модели с учителем

- Данные размечены
- Данные разделены на тренировочный и тестовый наборы
- Необходимо либо предсказать класс, либо предсказать значение

Обучение

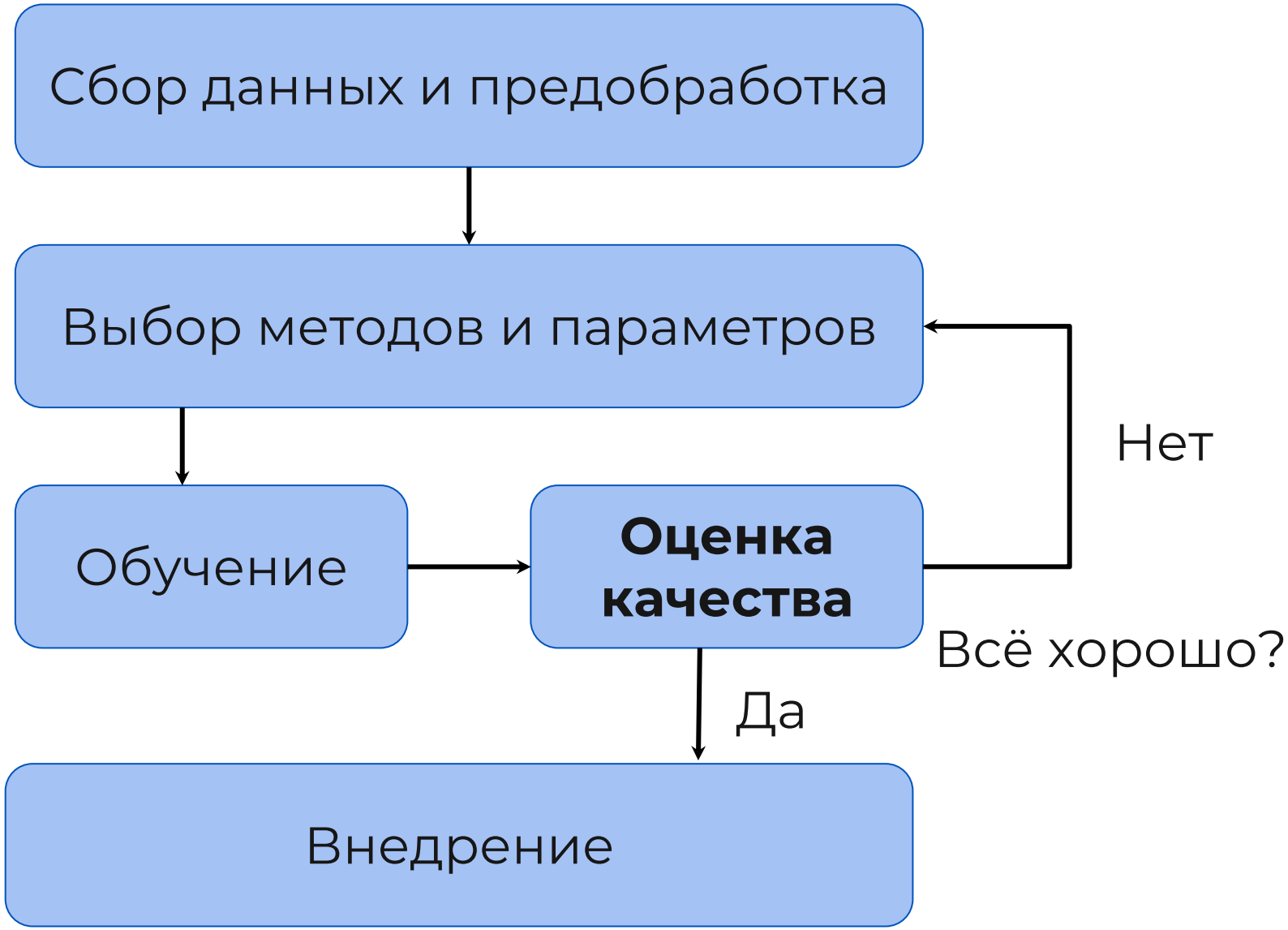
Тренировка модели без учителем

- Данные не размечены
- Единый набор данных для (само)обучения модели
- Необходимо найти классы объектов, зависимости в данных, выявить последовательности

Тренировка модели

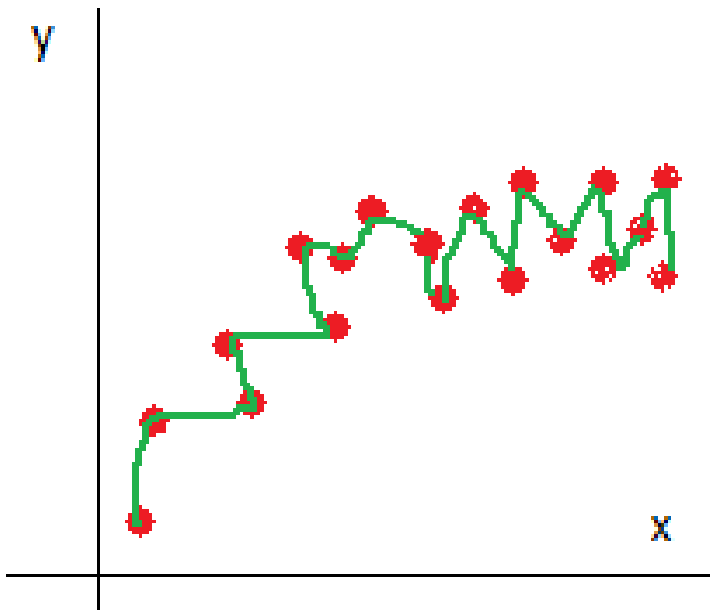
Обучение

Ход работы

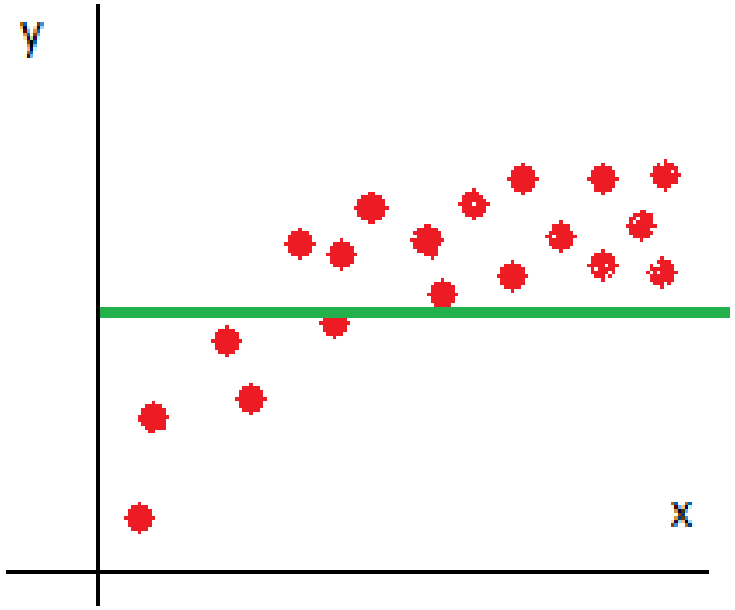

Сбор данных и предобработка
Выбор методов и параметров
Обучение
Оценка качества
Нет
Всё хорошо?
Да
Внедрение

Как оценивать эффективность модели?

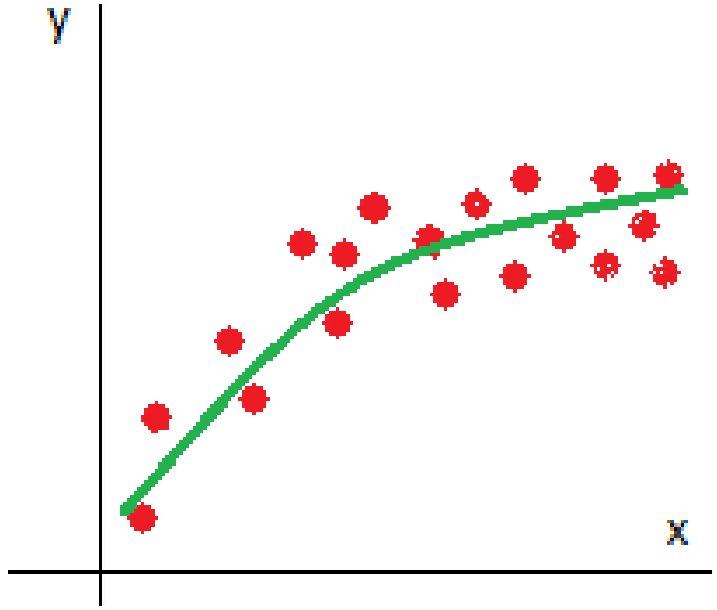
Оценка качества



Overfitting (High Variance)



Underfitting (High Bias)



Just Right

Метрики

Оценка качества

НИЯУ МИФИ

	Y = 1	Y = 0
X = 1	True Positive (TP)	False Positive (FP) (Type I error)
X = 0	False Negative (FN) (Type II error)	True Negative (TN)

X – результат работы алгоритма,
Y – истинная метка

$$\text{Accuracy} = \frac{1}{n}\sum_{i=1}^{n}\{x_i = y_i\} = \frac{\boldsymbol{TP + TN}}{\boldsymbol{TP + TN + FP + FN}}$$

$$\text{Precision} = \frac{\sum_{i=1}^{n}\{x_i = y_i = 1\}}{\sum_{i=1}^{n}\{x_i = 1\}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FP}}$$

$$\text{Recall} = \frac{\sum_{i=1}^{n}\{x_i = y_i = 1\}}{\sum_{i=1}^{n}\{y_i = 1\}} = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}}$$

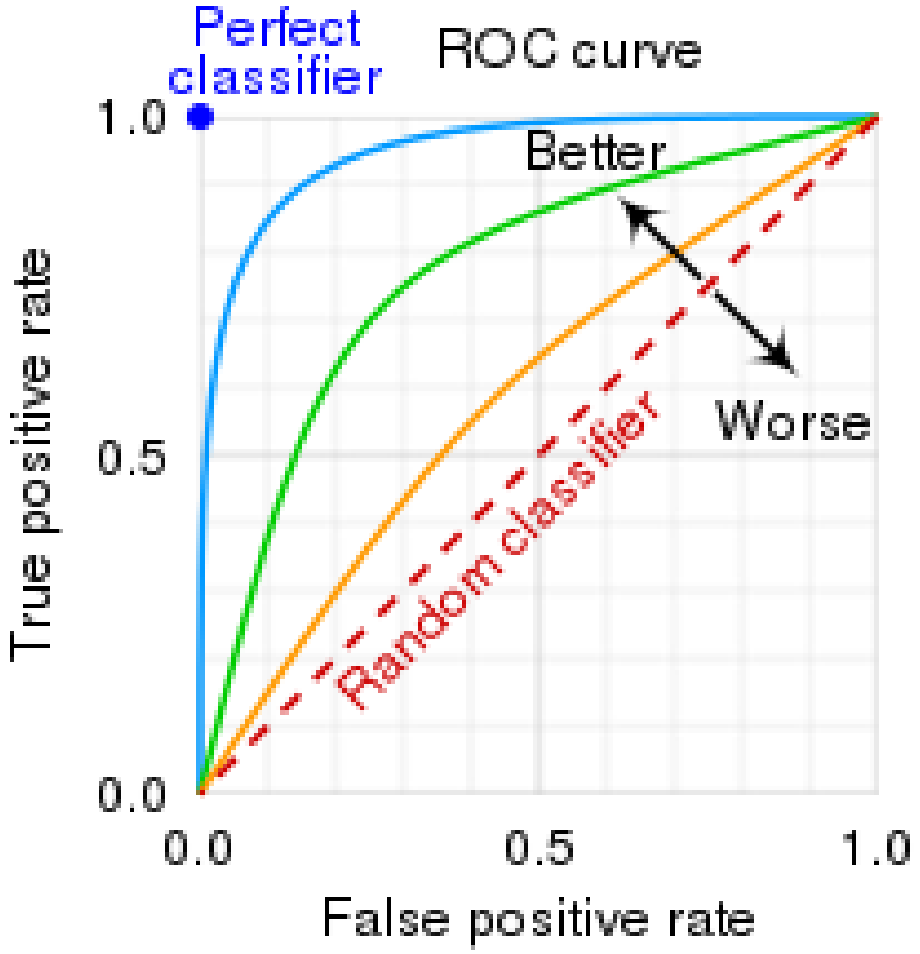
$$\text{F} - \text{score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

Метрики задачи классификации

Оценка качества

НИЯУ МИФИ

ROC – кривая и AUC



X – результат работы алгоритма, Y – истинная метка

	Y = 1	Y = 0
X = 1	True Positive (TP)	False Positive (FP) (Type I error)
X = 0	False Negative (FN) (Type II error)	True Negative (TN)

$$TPR = \frac{TP}{TP + FN} = Recall$$

$$FPR = \frac{FP}{FP + TN}$$

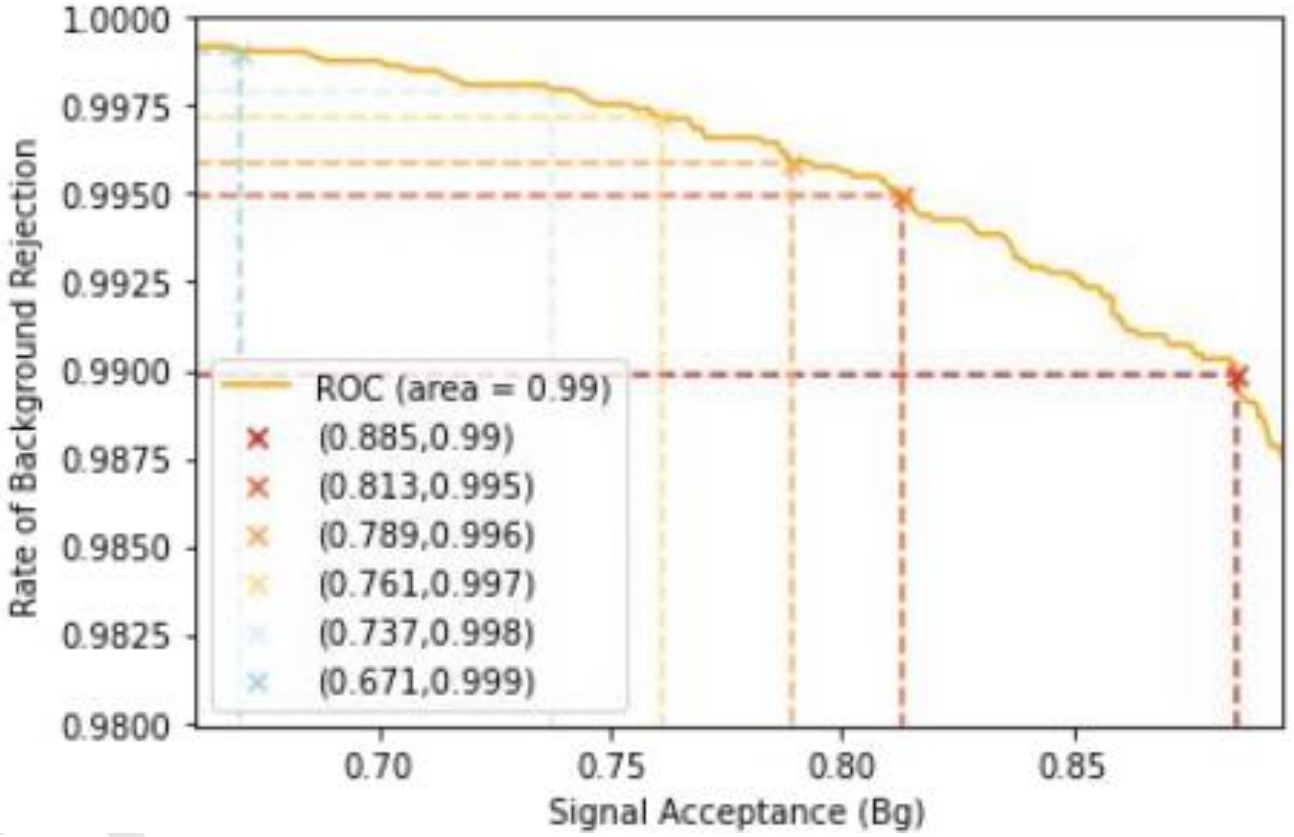
Метрики задачи классификации

Оценка качества

НИЯУ МИФИ

ROC – кривая и AUC

$$BRR = 1 - FPR = 1 - \frac{FP}{FP + TN} = \frac{\boldsymbol{TN}}{\boldsymbol{FP + TN}}$$



$$SA = TPR = \frac{\boldsymbol{TP}}{\boldsymbol{TP + FN}} = \text{Recall}$$

	Y = 1	Y = 0
X = 1	True Positive (TP)	False Positive (FP) (Type I error)
X = 0	False Negative (FN) (Type II error)	True Negative (TN)

X – результат работы алгоритма, Y – истинная метка

Метрики задачи регрессии

Оценка качества

НИЯУ МИФИ

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - x_i|$$

y – истинное значение целевой переменной
x – предсказанное значение целевой переменной

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2$$

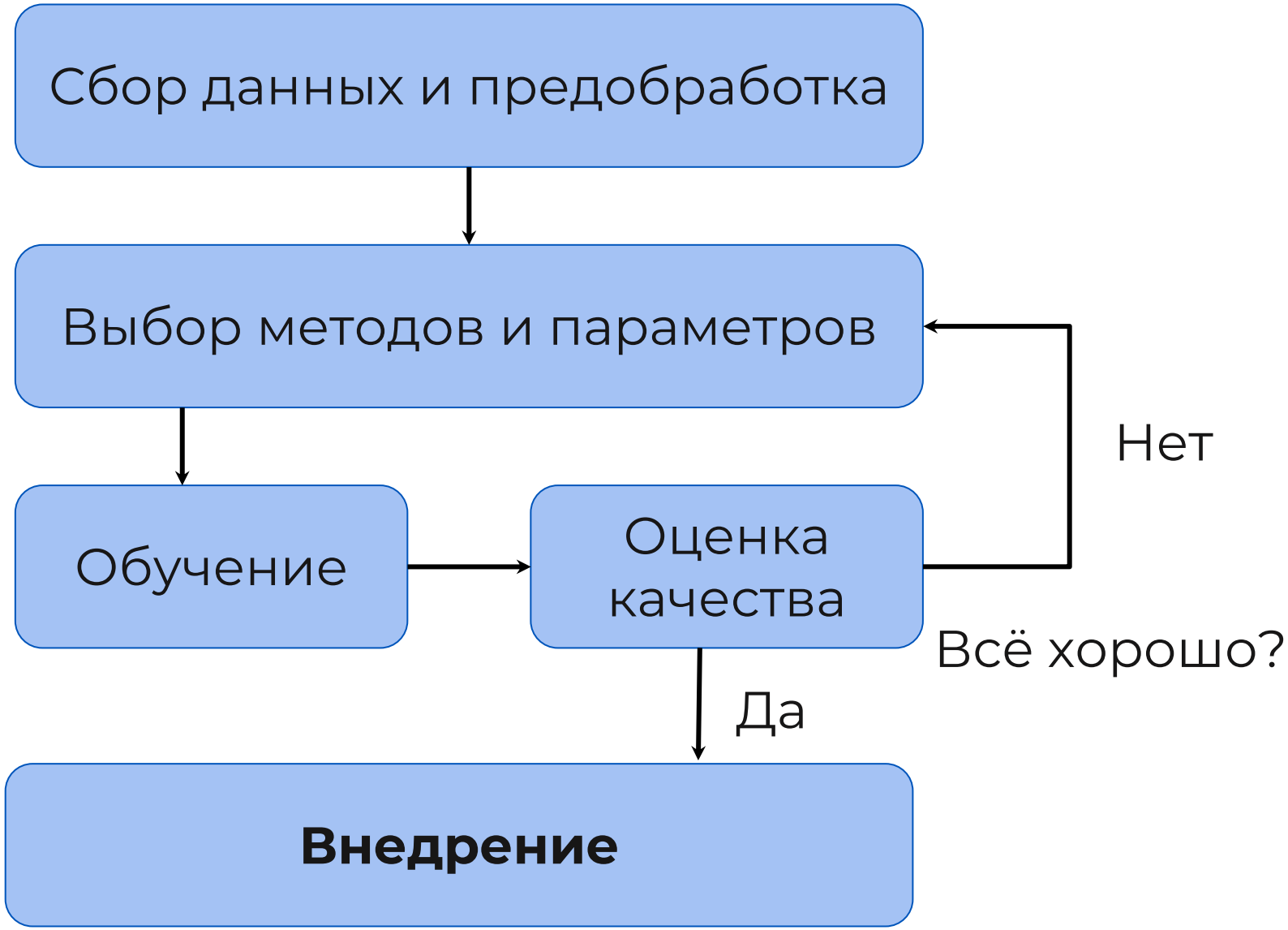
$$R^2 \equiv 1 - \frac{\sum_i (y_i - x_i)^2}{\sum_i (y_i - \bar{y})^2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{y_i - x_i}{y_i}\right|$$

$$\text{My metric} = \frac{1}{n}\sum \ldots$$

Ход работы


Сбор данных и предобработка
Выбор методов и параметров
Обучение
Оценка качества
Нет
Всё хорошо?
Да
Внедрение

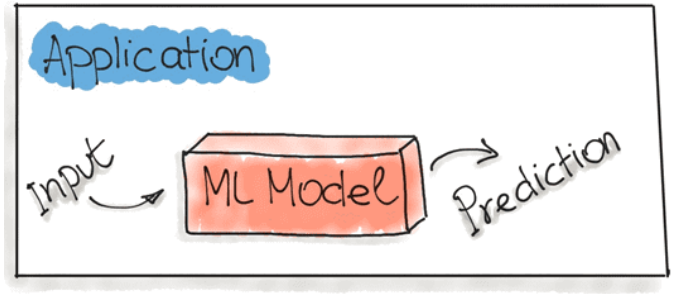
Развертывание модели

Внедрение

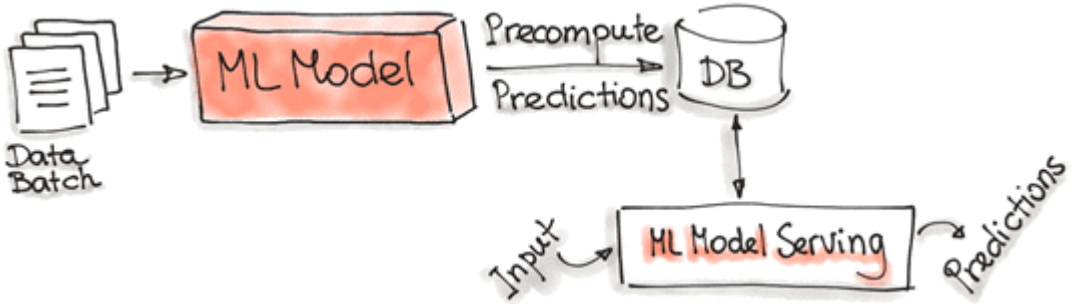
НИЯУ МИФИ

Модель как услуга

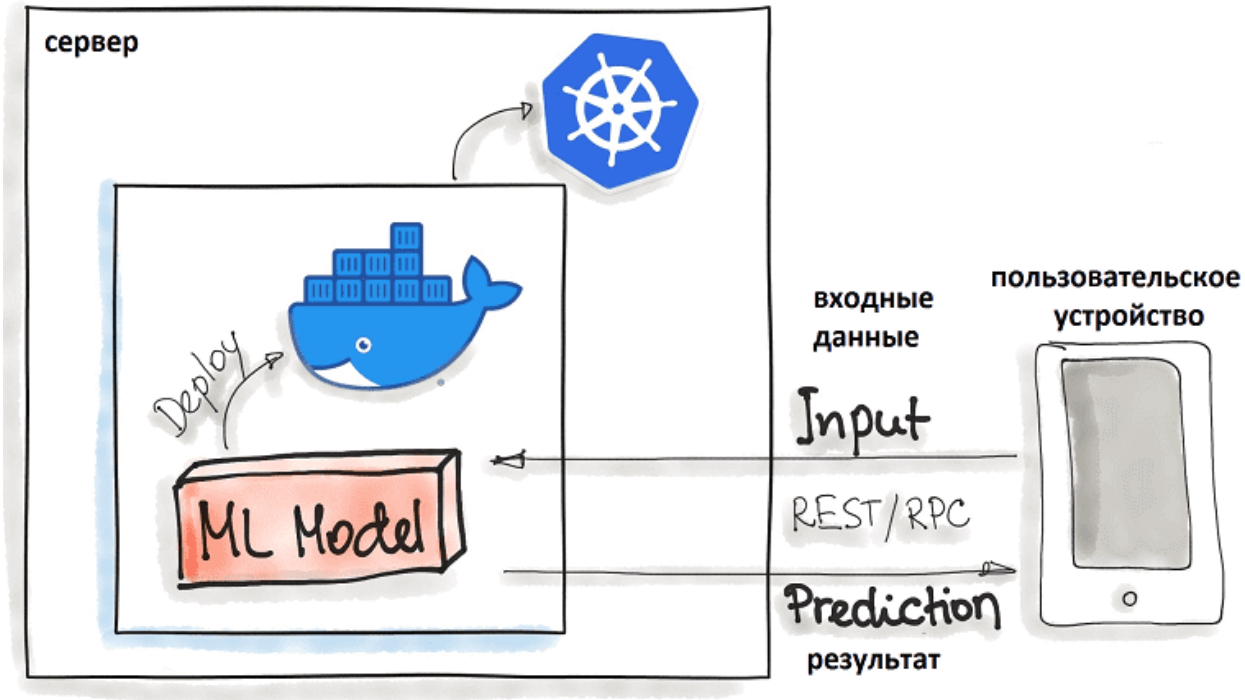
Модель как зависимость



Предварительный расчет



Развертывание с помощью Docker-контейнеров



Конец 1 части.
Вопросы?

Методы статистического анализа

Часть 2: Модели

27.10.2022

План второй части

Будут рассмотрены следующие модели

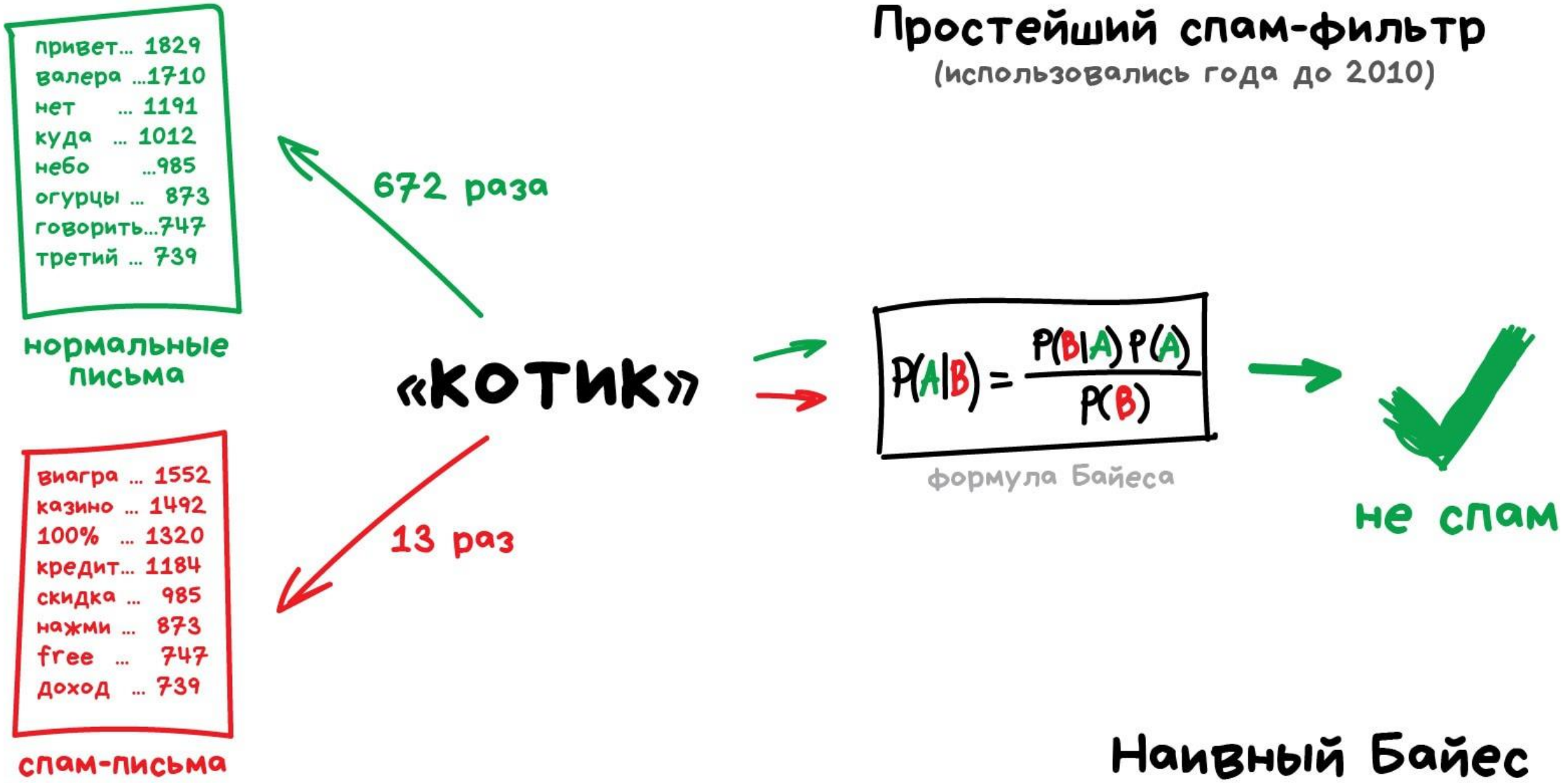
- Наивный байесовский классификатор
- Дерево решений
- Метод опорных векторов
- Регрессия
- Логистическая регрессия
- Принцип главных компонент

План второй части

Будут рассмотрены следующие модели

- **Наивный байесовский классификатор**
- Дерево решений
- Метод опорных векторов
- Регрессия
- Логистическая регрессия
- Принцип главных компонент

Наивная байесовская классификация



Наивная байесовская классификация

Алгоритм работы

Пусть имеется набор данных в формате «Погодные условия» - «Игра». Задача: на основе погодных условий спрогнозировать возможность проведения матча. Для этого необходимо проделать следующие шаги:

- **Преобразовать** набор данных в частотную таблицу.
- Создать **таблицу** правдоподобия, рассчитав соответствующие вероятности.

Например: P(облачно) = 0,37, P(игра состоится) = 0.55

- Рассчитать **апостериорную** вероятность для каждого класса по теореме Байеса:

$$\boldsymbol{P}(\boldsymbol{A}|\boldsymbol{B}) = \frac{\boldsymbol{P}(\boldsymbol{B}|\boldsymbol{A})\,\boldsymbol{P}(\boldsymbol{A})}{\boldsymbol{P}(\boldsymbol{B})}$$

Задача: состоится ли матч при солнечной погоде?

$$\boldsymbol{P}(\text{состоится}|\text{солнечно}) = \frac{\boldsymbol{P}(\text{солнечно}|\text{состоится})\,\boldsymbol{P}(\text{состоится})}{\boldsymbol{P}(\text{солнечно})}$$

Проблема: отравление Байеса

План второй части

Будут рассмотрены следующие модели

- Наивный байесовский классификатор
- **Дерево решений**
- Метод опорных векторов
- Регрессия
- Логистическая регрессия
- Принцип главных компонент

Дерево принятия решений

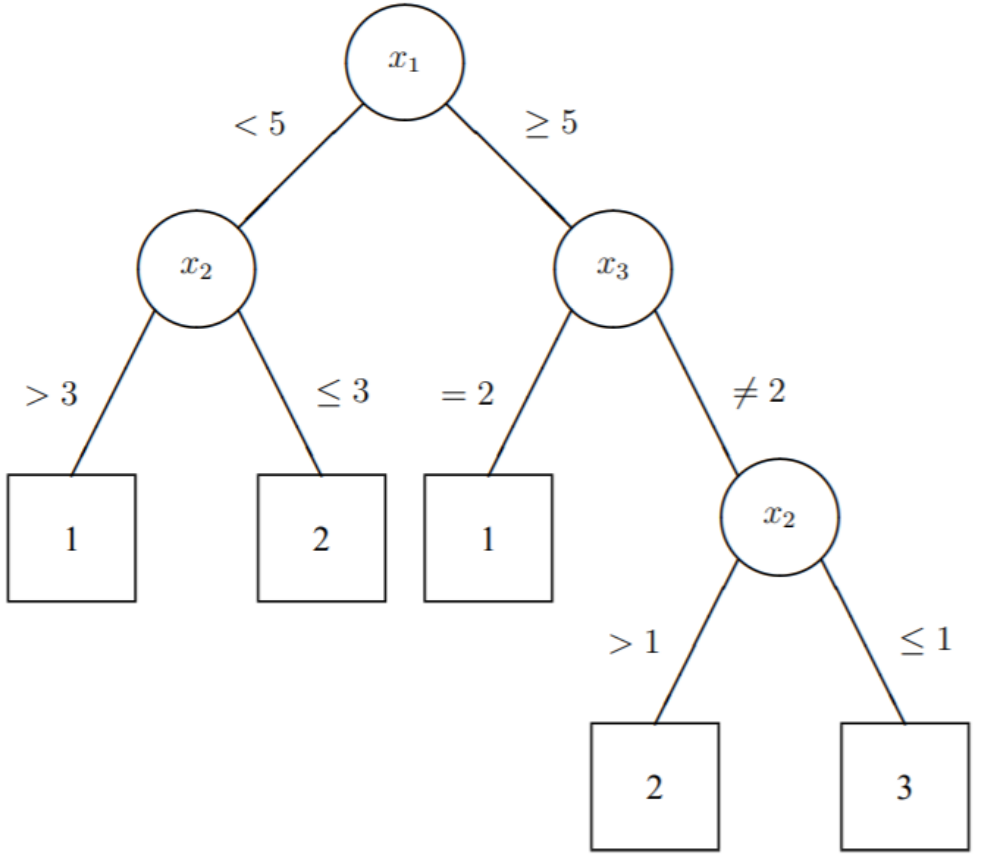
Дерево Решений

Дерево принятия решений

Алгоритм построения

Построение дерева складывается из следующих шагов:

- **Цикл** по всем признакам:
    - Цикл по упорядоченным **значениям** каждого признака:
        - **Поиск** наилучшего разделения среди всех значений этого признака на основании некоторого критерия.
- **Разбиение** на левую и правую ветвь.
- **Повторение** процедуры внутри каждой из ветвей вплоть до срабатывания критерия остановки.

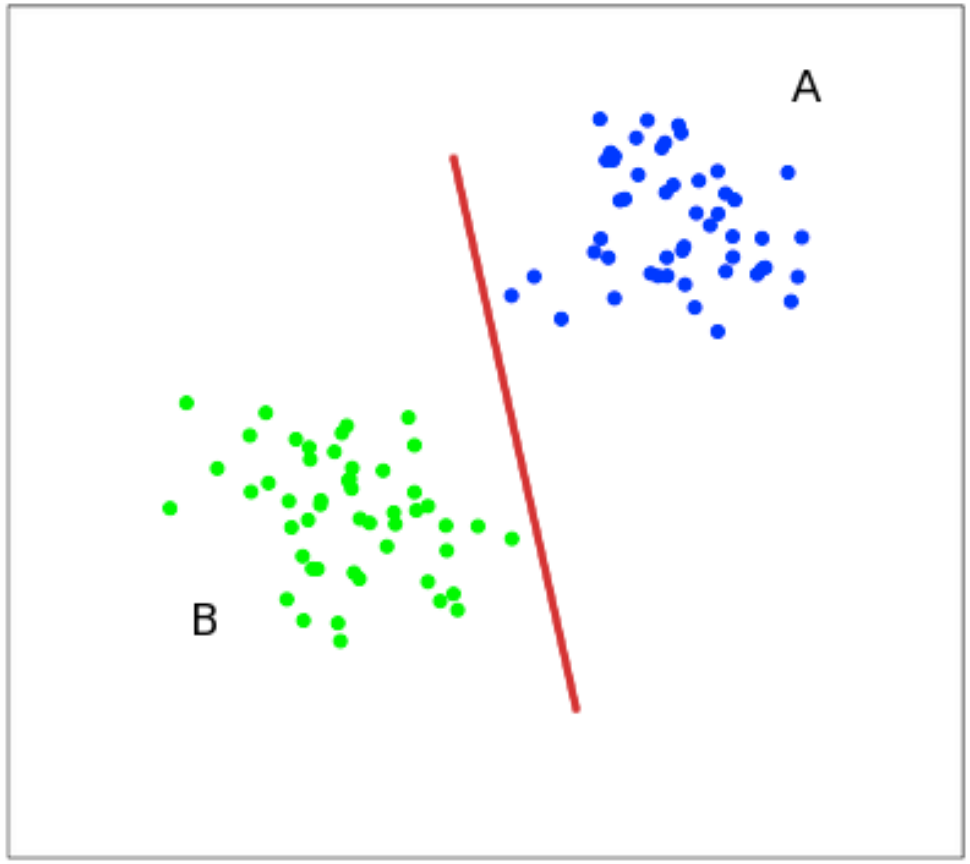


План второй части

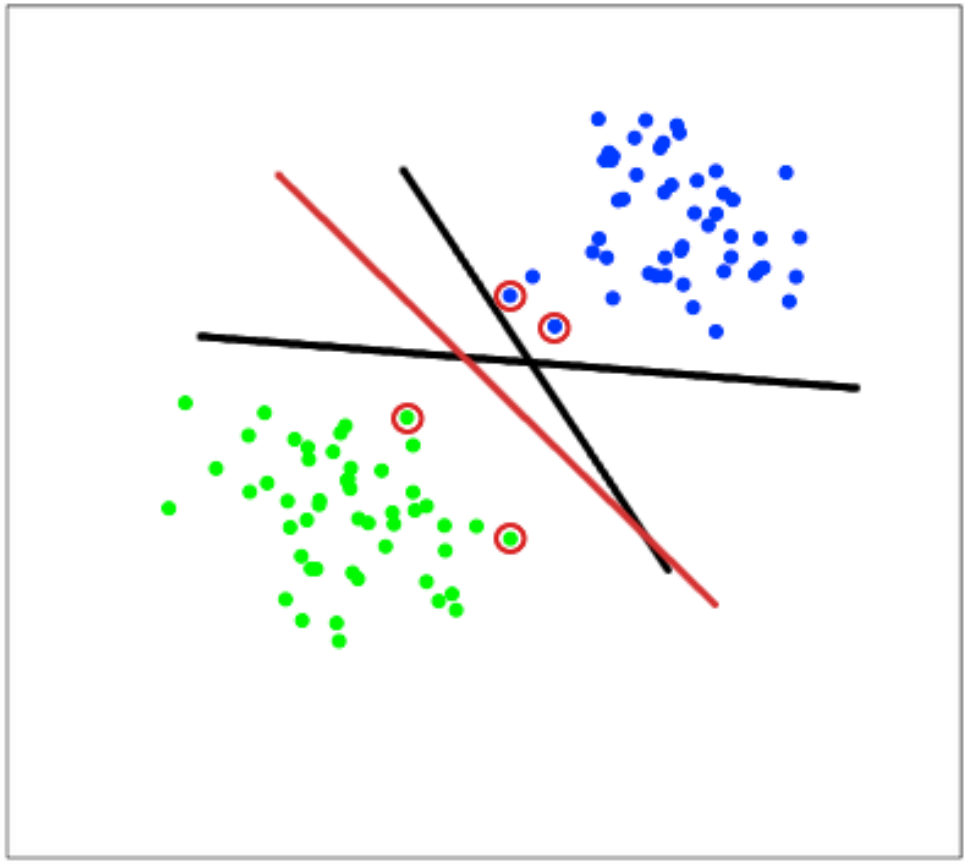
Будут рассмотрены следующие модели

- Наивный байесовский классификатор
- Дерево решений
- **Метод опорных векторов**
- Регрессия
- Логистическая регрессия
- Принцип главных компонент

Метод опорных векторов



?



Метод опорных векторов

НИЯУ МИФИ

Принцип работы

Пусть есть обучающая выборка $(\boldsymbol{x_1}, \boldsymbol{y_1}), \ldots, (\boldsymbol{x_m}, \boldsymbol{y_m}), \boldsymbol{x_i} \in \boldsymbol{R^n}, \boldsymbol{y_i} \in \{-\mathbf{1}, \mathbf{1}\}$

$\boldsymbol{F(\vec{x}) = sign\big((\vec{w}, \vec{x}) + b\big)}$, где

w – нормальный вектор к разделяющей (гипер)плоскости,

b – вспомогательный вектор

$\boldsymbol{F(\vec{x}) = 1}$ или $\boldsymbol{-1}$ в зависимости от класса события

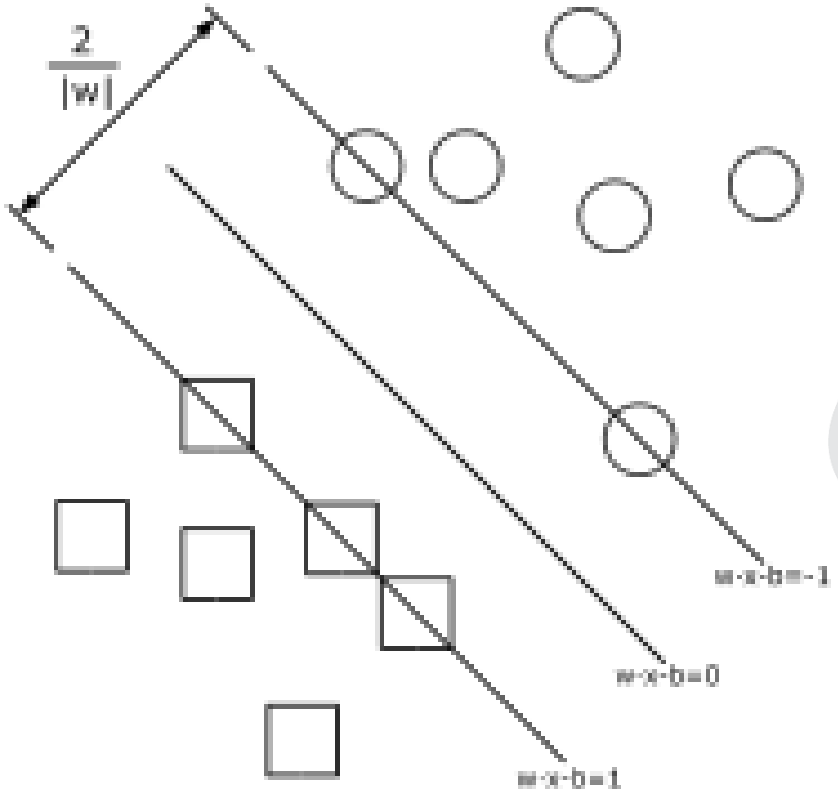
Метод опорных векторов

Принцип работы

- Максимальное расстояние до опорных векторов
- Точки не должны попасть в область полосы

$$\begin{cases} \boldsymbol{arg}\min_{\vec{w},b}\|\boldsymbol{w}\|^2, \\ \boldsymbol{y_i}\left((\vec{\boldsymbol{w}},\vec{\boldsymbol{x}})+\boldsymbol{b}\right)\geq \boldsymbol{1}, \boldsymbol{i}=\overline{\boldsymbol{1,m}} \end{cases} \Rightarrow$$

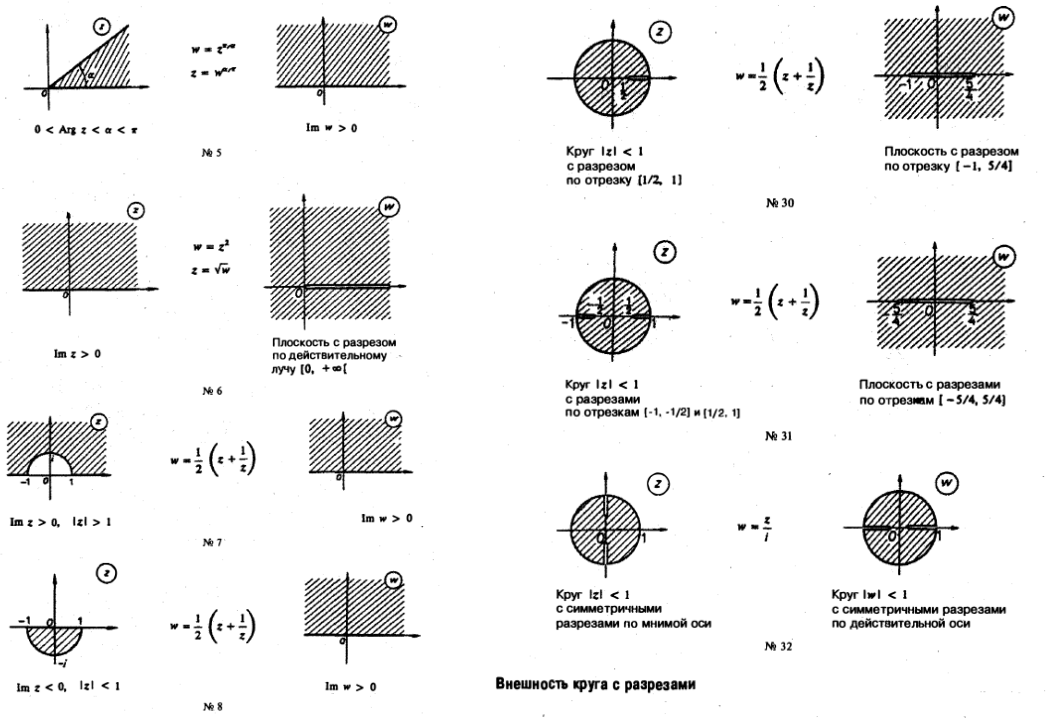
метод множителей Лагранжа

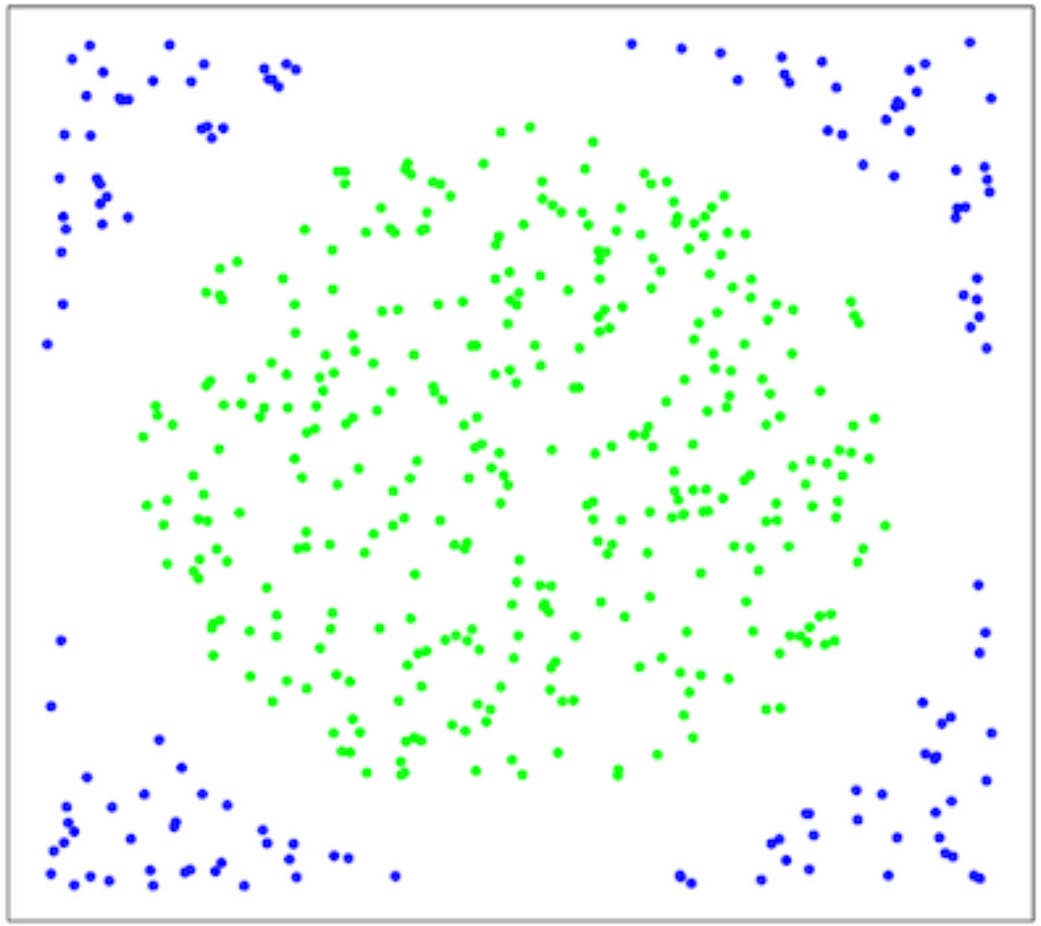


Метод опорных векторов

Вопрос: А как быть если данные не совсем хорошо линейно разделимы?

Ответ: обратиться к ТФКП



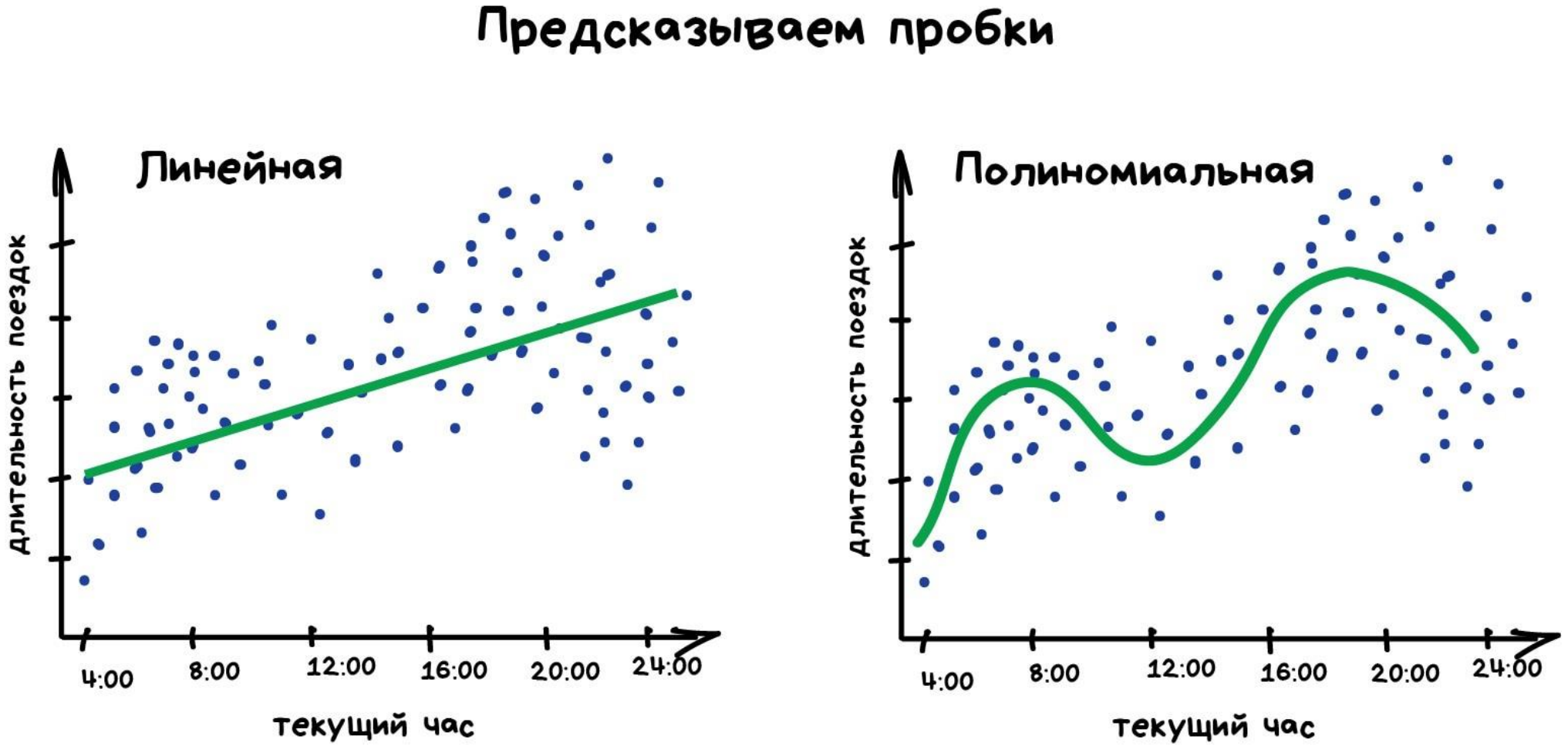


План второй части

Будут рассмотрены следующие модели

- Наивный байесовский классификатор
- Дерево решений
- Метод опорных векторов
- **Регрессия**
- Логистическая регрессия
- Принцип главных компонент

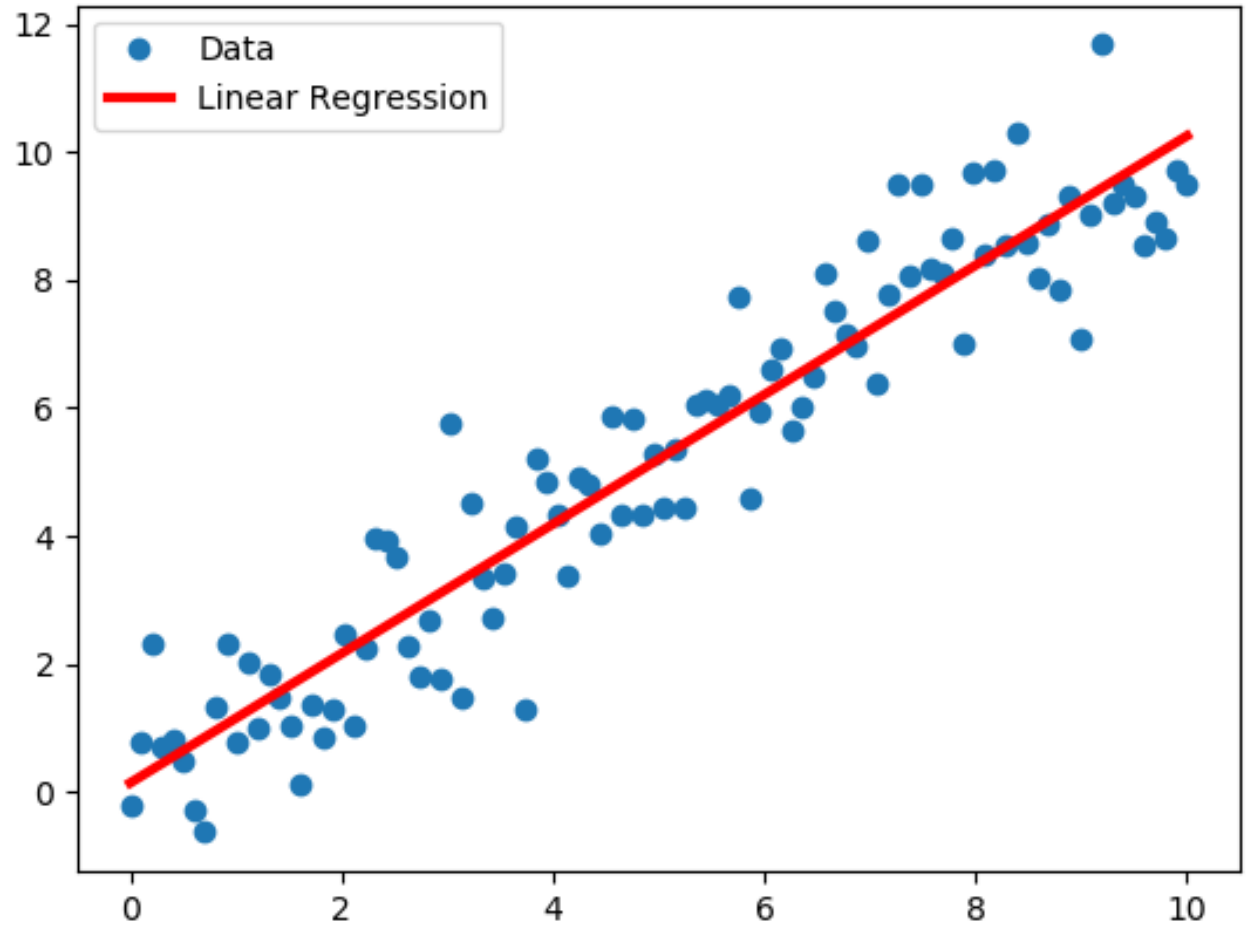
Регрессия

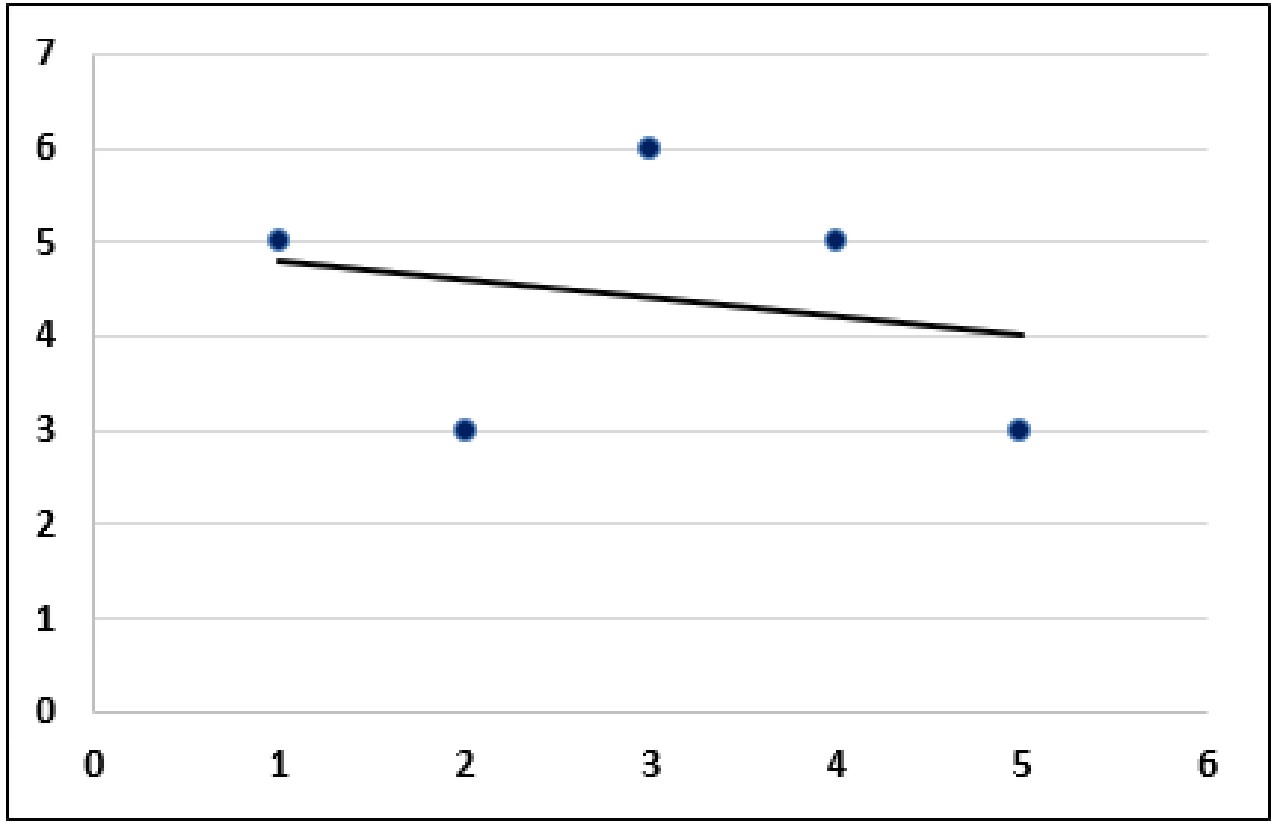


Вопрос: Что Вы знаете о линейной регрессии?

Регрессия

Метод наименьших квадратов





Метод наименьших квадратов

Алгоритм решения

Задача: минимизация функции $F(a,b) = \sum_{i=1}^{n}\left(y_i - (ax_i + b)\right)^2$

Решение:

$$\begin{cases} \dfrac{\partial F(a,b)}{\partial a} = 0 \\ \dfrac{\partial F(a,b)}{\partial b} = 0 \end{cases} \Leftrightarrow \begin{cases} -2\sum_{i=1}^{n}\left(y_i - (ax_i + b)\right)x_i = 0 \\ -2\sum_{i=1}^{n}\left(y_i - (ax_i + b)\right) = 0 \end{cases} \Leftrightarrow \begin{cases} a\sum_{i=1}^{n} x_i^2 + b\sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i \\ a\sum_{i=1}^{n} x_i + \sum_{i=1}^{n} b = \sum_{i=1}^{n} y_i \end{cases}$$

Ответ:

$$\begin{cases} a = \dfrac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2} \\ b = \dfrac{\sum_{i=1}^{n} y_i - a\sum_{i=1}^{n} x_i}{n} \end{cases}$$

Полиномиальная регрессия

Задача: найти коэффициенты $\boldsymbol{\beta_i}$ в разложении:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \ \ldots + \beta_m x_i^m + \epsilon_i$$

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \ldots & x_1^m \\ 1 & x_2 & x_2^2 & \ldots & x_2^m \\ 1 & x_3 & x_3^2 & \ldots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \ldots & x_5^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ \epsilon_3 \\ \vdots \\ \epsilon_n \end{bmatrix} \Leftrightarrow \vec{y} = X\vec{\beta} + \vec{\epsilon}$$

Тогда примерно коэффициенты полиномиальной регрессии можно оценить как: $\widehat{\vec{\beta}} = \left(X^T X\right)^{-1} X^T \vec{y}$

План второй части

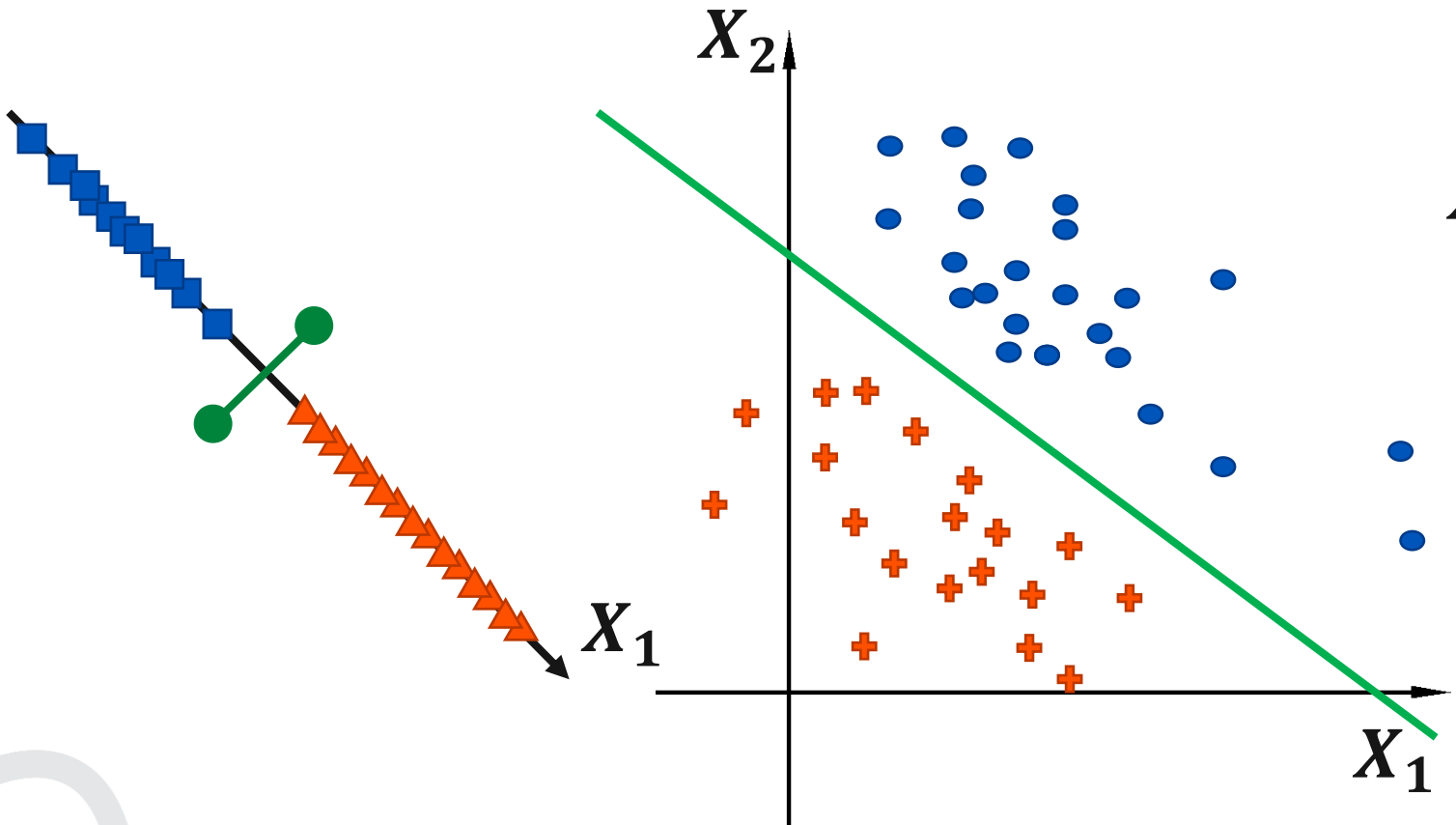
Будут рассмотрены следующие модели

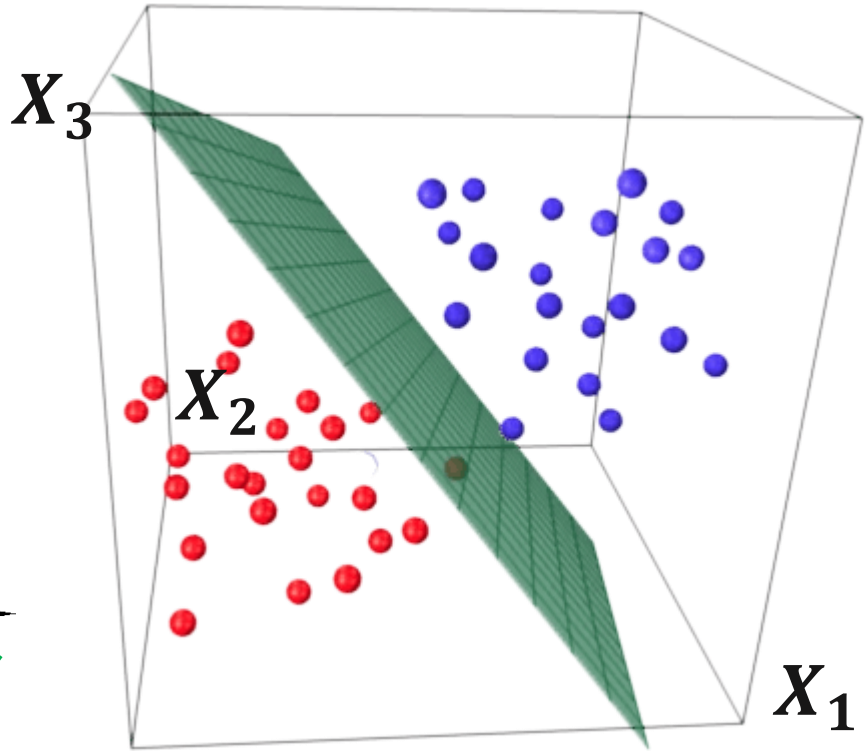
- Наивный байесовский классификатор
- Дерево решений
- Метод опорных векторов
- Регрессия
- **Логистическая регрессия**
- Принцип главных компонент

Логистическая ~~регрессия~~ классификация

Пусть есть **2** класса событий: **0** и **1**. P_0 и P_1 — вероятность принадлежности к определенному классу.

$$P_1 = 1 - P_0; P_1, P_2 \in [0, 1]$$





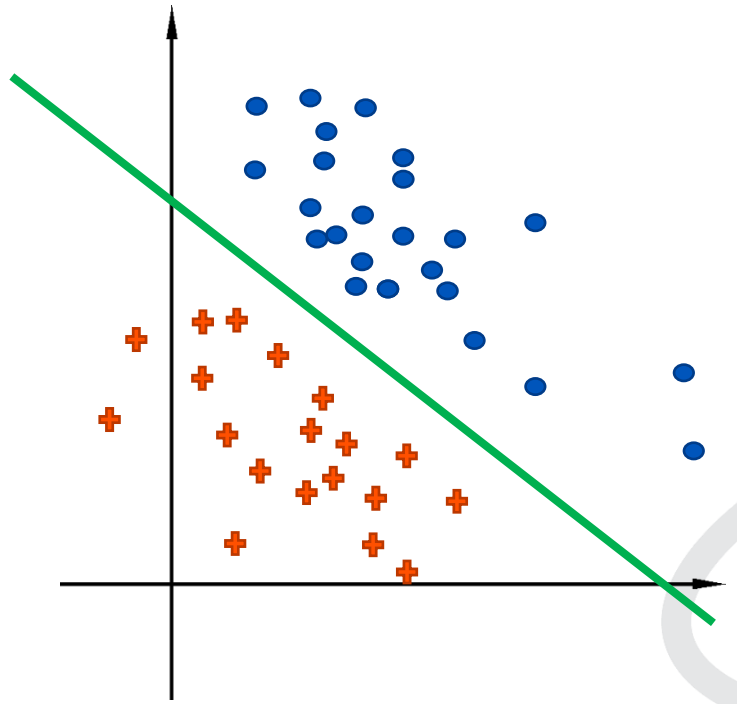
Логистическая регрессия

Рассмотрим задачу в двумерном случае

Пусть найдено уравнение прямой: $\boldsymbol{\beta_0 + \beta_1 x_1 + \beta_2 x_2 = 0}$

Рассмотрим точку $(\boldsymbol{a}, \boldsymbol{b})$. Подставим ее в уравнение прямой:

$\boldsymbol{t = \beta_0 + \beta_1 a + \beta_2 b}$, причем t может принимать 3 значения:



$$\begin{cases} \boldsymbol{t < 0}, \text{тогда } (\boldsymbol{a}, \boldsymbol{b}) \text{ принадлежит классу } \boldsymbol{0}, \boldsymbol{P_1} \in [\boldsymbol{0}, \boldsymbol{0.5}) \\ \boldsymbol{t > 0}, \text{тогда } (\boldsymbol{a}, \boldsymbol{b}) \text{ принадлежит классу } \boldsymbol{1}, \boldsymbol{P_1} \in (\boldsymbol{0.5}, \boldsymbol{1}] \\ \boldsymbol{t = 0}, \text{тогда } (\boldsymbol{a}, \boldsymbol{b}) \text{ лежит на прямой}, \boldsymbol{P_1 = 0.5} \end{cases}$$

Логистическая регрессия

Имеем: функция с выходным значением $(-\infty, +\infty)$.
Вопрос: как перевести выходное значение в вероятность?

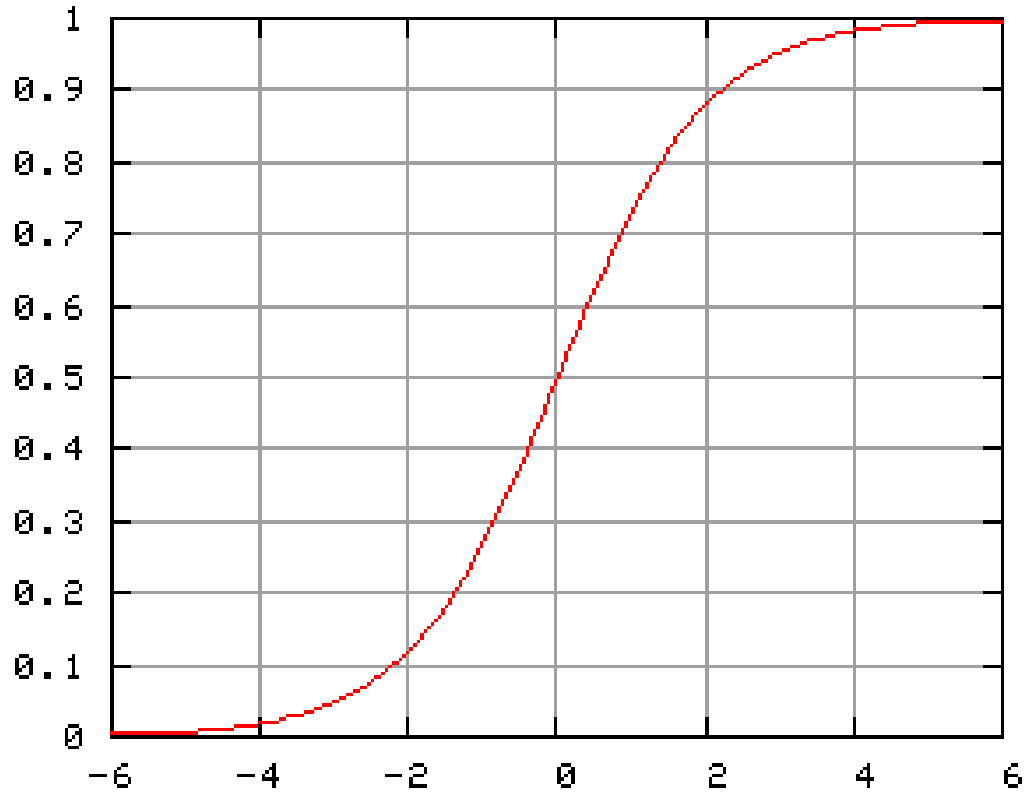
$$\boldsymbol{OR(X)} = \frac{\boldsymbol{P(X)}}{\boldsymbol{1 - P(X)}} \in [\boldsymbol{0}, +\infty)$$

$$\boldsymbol{log(OR(X))} = \boldsymbol{log}(\frac{\boldsymbol{P(X)}}{\boldsymbol{1 - P(X)}}) \in (-\infty, +\infty)$$

Приняв $\boldsymbol{log\big(OR(X)\big) = t, t} \in (-\infty, +\infty)$

Получаем: $e^t = OR(X) = \frac{\boldsymbol{P(X)}}{\boldsymbol{1 - P(X)}}$, тогда

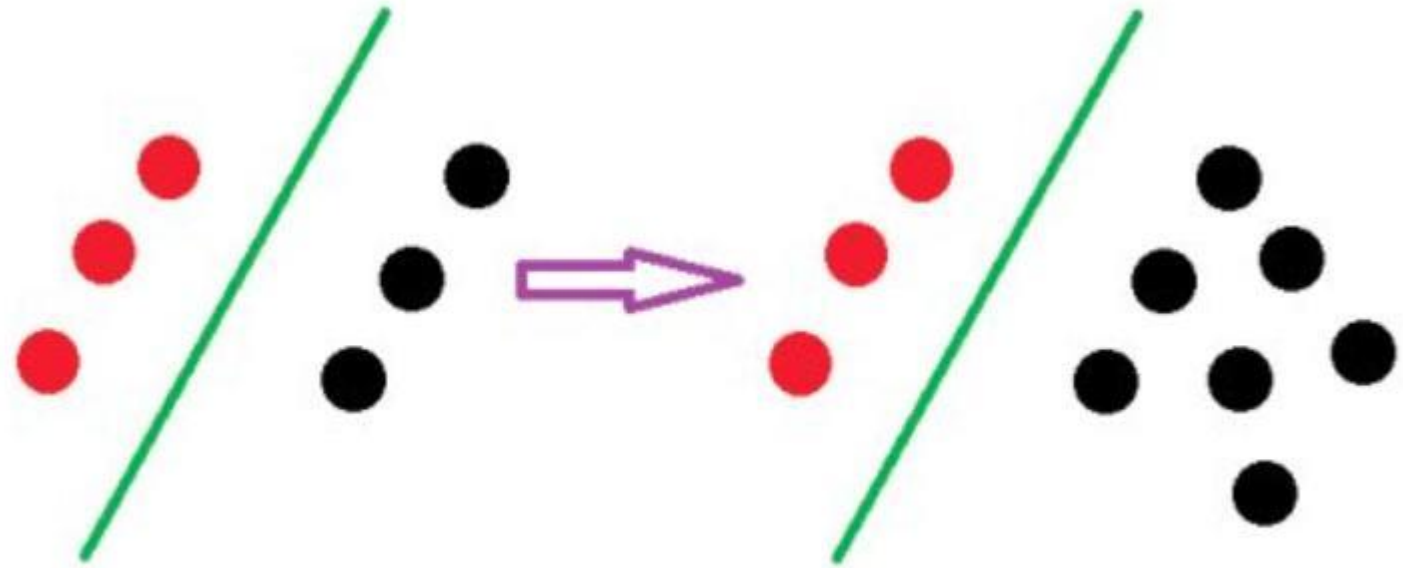
$P(X) = \frac{e^t}{1+e^t} = \frac{1}{1+e^{-t}}$ - логистическая функция



LR vs. SVM

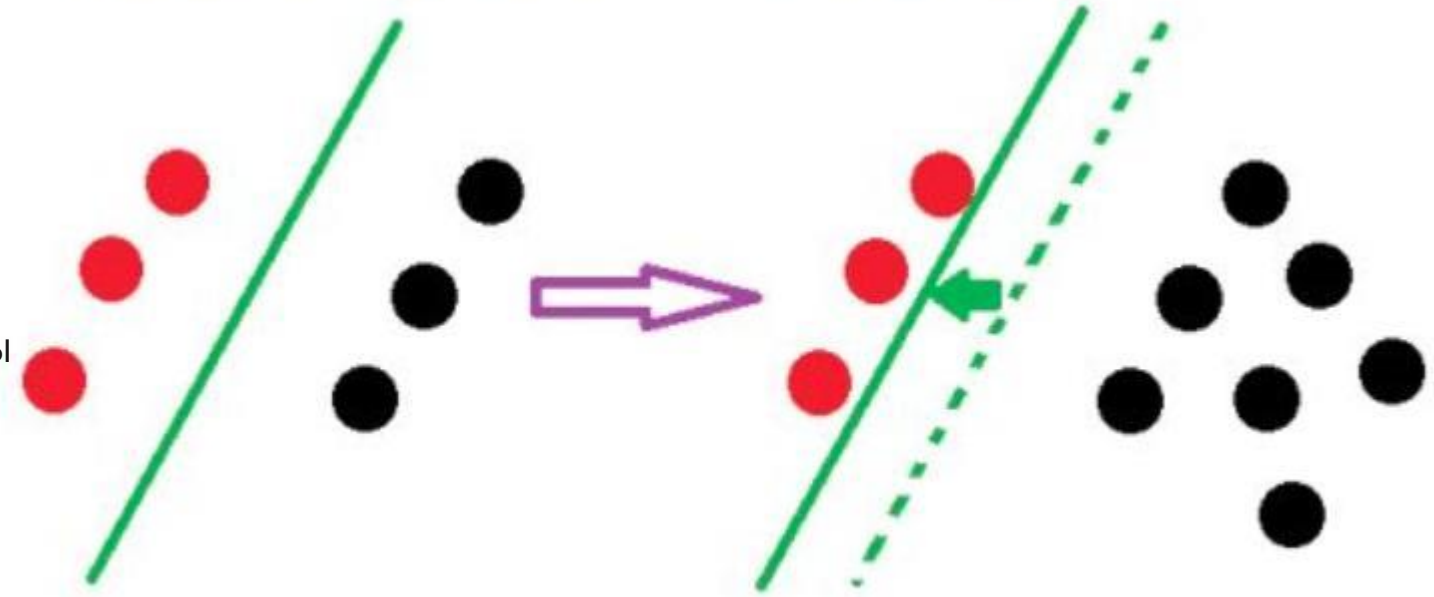
SVM

Геометрические свойства



LR

Статистические подходы



План второй части

Будут рассмотрены следующие модели

- Наивный байесовский классификатор
- Дерево решений
- Метод опорных векторов
- Регрессия
- Логистическая регрессия
- **Принцип главных компонент**

Принцип уменьшения размерности

Овчарка

Метод главных компонент

Принцип работы алгоритма

Пусть имеется матрица переменных X (I x J), где
I – число образцов, J – чисто признаков, J >>1.
Обозначим $\boldsymbol{t_\alpha = p_{\alpha 1} x_1 + \ \ldots + p_{\alpha J}\, x_J}$ - линейная комбинация исходных переменных $\mathbf{x_j}$.

$$
I = \begin{matrix} e_{(G)} \\ \\ \\ \\ \\ \\ \\ \end{matrix}
\begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \end{matrix}
\begin{bmatrix}
1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\
0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
\end{bmatrix}
\begin{matrix} v_{(G)} \\ \\ \\ \Leftarrow \\ \\ \\ \end{matrix}
\begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{matrix}
$$

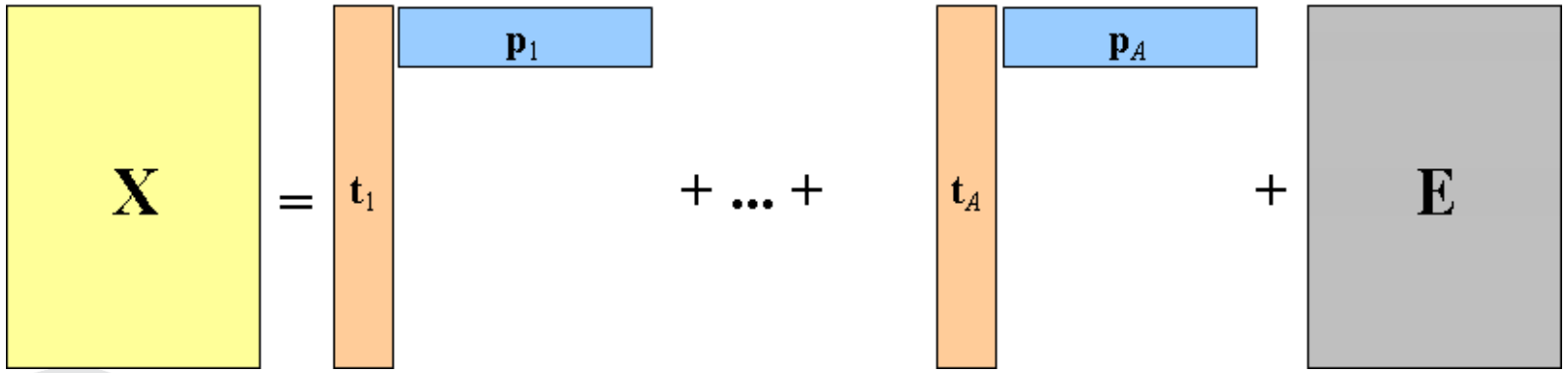
Метод главных компонент

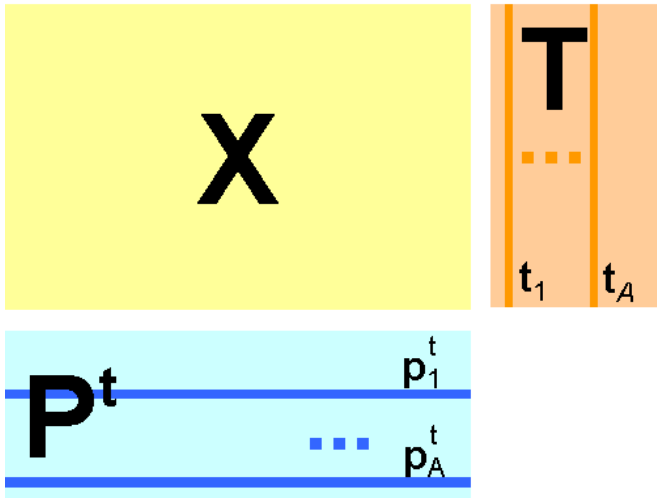
$$t_\alpha = p_{\alpha 1} x_1 + \ \ldots + p_{\alpha J}\, x_J$$

Тогда исходную матрицу X можно разложить следующим образом:

$$X = TP^T + E = \sum_{\alpha=1}^{A} t_\alpha p_\alpha^T + E$$

Матрица **T** называется матрицей *счетов* (scores). Ее размерность **(*I*×*A*)**.
Матрица **P** называется матрицей *нагрузок* (loadings). Ее размерность **(*J*×*A*)**.
E – это матрица *остатков*, размерностью (*I*×*J*).
Важное свойство – **ортогональность**.





Конец 2 части.
Вопросы?


ВСЁ!

Источники

Красивые картиночки:
https://vas3k.ru/blog/machine_learning/
Внедрение:
https://www.bigdataschool.ru/blog/mlops-deployment-patterns-and-strategies.html
Регрессионные модели:
https://habr.com/ru/company/ods/blog/323890/
Различие SVM vs. LR:
https://progler.ru/blog/razlichiya-mezhdu-mashinoy-opornyh-vektorov-i-logisticheskoy-regressiey