

# Лекция 2

## Метрические классификаторы

---

Екатерина Тузова

# Разбор летучки

---

## Мотивирующий пример

---

# Мотивирующий пример

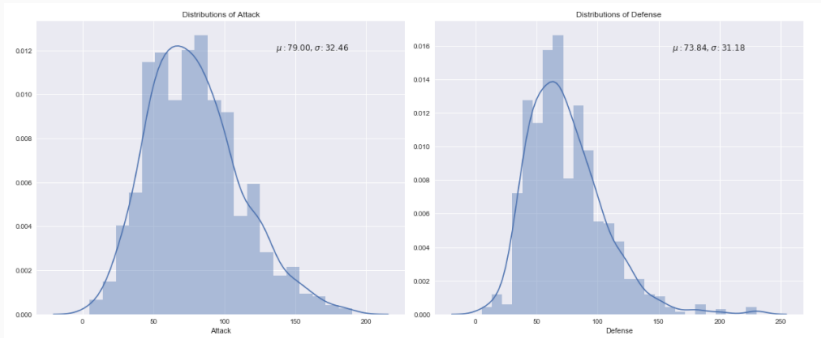


```
In [4]: pokemons.head()
```

```
Out[4]:
```

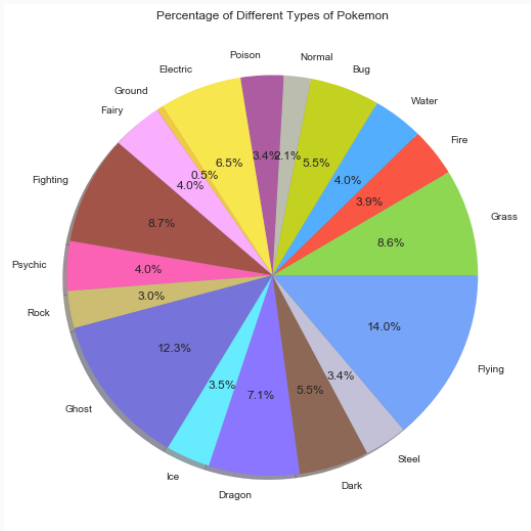
|   | Name                     | Type 1 | Type 2 | Total | HP | Attack | Defense | Sp. Atk | Sp. Def | Speed | Generation | Legendary |
|---|--------------------------|--------|--------|-------|----|--------|---------|---------|---------|-------|------------|-----------|
| 0 | Bulbasaur                | Grass  | Poison | 318   | 45 | 49     | 49      | 65      | 65      | 45    | 1          | False     |
| 1 | Ivysaur                  | Grass  | Poison | 405   | 60 | 62     | 63      | 80      | 80      | 60    | 1          | False     |
| 2 | Venusaur                 | Grass  | Poison | 525   | 80 | 82     | 83      | 100     | 100     | 80    | 1          | False     |
| 3 | VenusaurMega<br>Venusaur | Grass  | Poison | 625   | 80 | 100    | 123     | 122     | 120     | 80    | 1          | False     |
| 4 | Charmander               | Fire   | NaN    | 309   | 39 | 52     | 43      | 60      | 50      | 65    | 1          | False     |

# Распределения



Number of Pokemon = 800

# Типы покемонов



Какие признаки есть в датасете?

---



$$f : X \rightarrow D_f$$

- Бинарные ( $D_f = \{0, 1\}$ )
- Номинальные ( $D_f$  – конечное множество)
- Порядковые ( $D_f$  – конечное упорядоченное множество)
- Количественные ( $D_f = \mathbb{R}$ )

– Бинарные (Legendary)

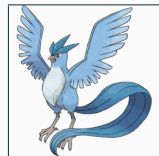
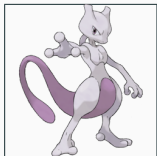
- Бинарные (Legendary)
- Номинальные (Type 1, Type 2)

- Бинарные (Legendary)
- Номинальные (Type 1, Type 2)
- Порядковые (Generation)

- Бинарные (Legendary)
- Номинальные (Type 1, Type 2)
- Порядковые (Generation)
- Количественные (Attack, Defense, ...)

**Легендарный покемон** это чрезвычайно редкий и зачастую очень могущественных покемон, о нем слагаются мифы и легенды в мире покемонов.

# Легендарность



|     | Name      | Type 1  | Type 2 | Total | HP  | Attack | Defense | Sp. Atk | Sp. Def | Speed | Generation | Legendary |
|-----|-----------|---------|--------|-------|-----|--------|---------|---------|---------|-------|------------|-----------|
| 0   | Bulbasaur | Grass   | Poison | 318   | 45  | 49     | 49      | 65      | 65      | 45    | 1          | False     |
| 24  | Rattata   | Normal  | NaN    | 253   | 30  | 56     | 35      | 25      | 35      | 72    | 1          | False     |
| 28  | Ekans     | Poison  | NaN    | 288   | 35  | 60     | 44      | 40      | 54      | 55    | 1          | False     |
| 32  | Sandshrew | Ground  | NaN    | 300   | 50  | 75     | 85      | 20      | 30      | 40    | 1          | False     |
| 35  | Nidorina  | Poison  | NaN    | 365   | 70  | 62     | 67      | 55      | 55      | 56    | 1          | False     |
| 156 | Articuno  | Ice     | Flying | 580   | 90  | 85     | 100     | 95      | 125     | 85    | 1          | True      |
| 162 | Mewtwo    | Psychic | NaN    | 680   | 106 | 110    | 90      | 154     | 90      | 130   | 1          | True      |
| 799 | Volcanion | Fire    | Water  | 600   | 80  | 110    | 120     | 130     | 90      | 70    | 6          | True      |



# Задача классификации

$X$  - множество объектов

$Y$  - множество классов

Обучающая выборка:  $X^l = (x_i, y_i)_{i=1}^l$

**Задача:** Построить алгоритм  $a: X \rightarrow Y$ , способный классифицировать произвольный объект  $x \in X$ .

# Задача классификации в нашем контексте

$X$  - покемоны

$Y$  - легендарность

Обучающая выборка:  $X^l = (x_i, y_i)_{i=1}^l$

**Задача:** Построить алгоритм  $a: X \rightarrow Y$ , способный определить, является ли покемон легендарным.

# Гипотеза компактности

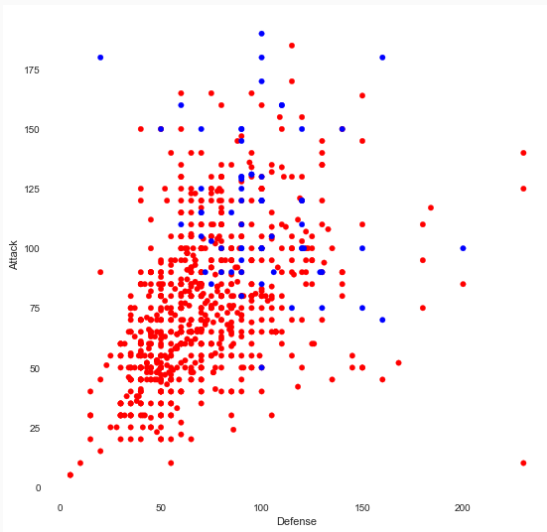
---

Схожие объекты, как правило, лежат в одном классе.

Схожие объекты, как правило, лежат в одном классе.

Как определить **схожесть** объектов?

# Пример



Схожие объекты, как правило, лежат в одном классе.

Схожесть = Функция расстояния

$$\rho : X \times X \rightarrow [0, \infty)$$

# Функции расстояния

---



$$\rho(u, v) = \sqrt{\sum_{j=1}^n |u^j - v^j|^2}, \quad u, v \in X^l$$

Признаковые описания объектов:

$$u = \{u^1, u^2, \dots, u^n\}$$

$$v = \{v^1, v^2, \dots, v^n\}$$

$$\rho(u, v) = \sum_{j=1}^n |u^j - v^j|, \quad u, v \in X^l$$

Признаковые описания объектов:

$$u = \{u^1, u^2, \dots, u^n\}$$

$$v = \{v^1, v^2, \dots, v^n\}$$

Обобщение евклидова расстояния и расстояния городских кварталов

$$\rho(u, v) = \left( \sum_{j=1}^n |u^j - v^j|^q \right)^{1/q}, \quad u, v \in X^l$$

Признаковые описания объектов:

$$u = \{u^1, u^2, \dots, u^n\}$$

$$v = \{v^1, v^2, \dots, v^n\}$$

Минимальное количество операций вставки одного символа, удаления одного символа и замены одного символа на другой, необходимых для превращения одной строки в другую.

## Расстояние Левенштейна

|   |   | Е | Л | Е | Р | Н | А | Н | Т |
|---|---|---|---|---|---|---|---|---|---|
|   | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Р | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Е | 2 | 1 | 2 | 2 | 3 | 4 | 5 | 6 | 7 |
| Л | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| Е | 4 | 3 | 2 | 1 | 2 | 3 | 4 | 5 | 6 |
| В | 5 | 4 | 3 | 2 | 2 | 3 | 4 | 5 | 6 |
| А | 6 | 5 | 4 | 3 | 3 | 3 | 3 | 4 | 5 |
| Н | 7 | 6 | 5 | 4 | 4 | 4 | 4 | 3 | 4 |
| Т | 8 | 7 | 6 | 5 | 5 | 5 | 5 | 4 | 3 |

# Метрический классификатор

---

# Обобщенный метрический классификатор

$u \in X$  - произвольный объект, который собираемся классифицировать.

# Обобщенный метрический классификатор

$u \in X$  - произвольный объект, который собираемся классифицировать.

Отсортируем объекты из  $X^l$  относительно  $u$ :

$$\rho(u, x_1) \leq \rho(u, x_2) \leq \dots \leq \rho(u, x_l)$$

$x_i$  -  $i$ -й сосед объекта  $u$

$y_i$  - класс  $i$ -го соседа  $u$



# Обобщенный метрический классификатор

$$\rho(u, x_1) \leq \rho(u, x_2) \leq \dots \leq \rho(u, x_l)$$

$x_i$  –  $i$ -й сосед объекта  $u$

$y_i$  – класс  $i$ -го соседа  $u$

$$\rho(u, x_1) \leq \rho(u, x_2) \leq \dots \leq \rho(u, x_l)$$

$x_i$  –  $i$ -й сосед объекта  $u$

$y_i$  – класс  $i$ -го соседа  $u$

**Идея 1:** Посмотрим на ближайшие объекты и отнесем  $u$  к доминирующему классу.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$w(i, u)$  - вес  $i$ -го соседа  $u$ , неотрицателен

## Метод ближайшего соседа

Объект относится к тому классу, к которому относится ближайший в выборке.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i = 1]$$

## Метод ближайшего соседа

Объект относится к тому классу, к которому относится ближайший в выборке.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i = 1]$$

+ Простота реализации (lazy learning)

## Метод ближайшего соседа

Объект относится к тому классу, к которому относится ближайший в выборке.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i = 1]$$

- + Простота реализации (lazy learning)
- + Интерпретируемость решения

## Метод ближайшего соседа

Объект относится к тому классу, к которому относится ближайший в выборке.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i = 1]$$

- + Простота реализации (lazy learning)
- + Интерпретируемость решения
- Неустойчивость к шуму

# Метод ближайшего соседа

Объект относится к тому классу, к которому относится ближайший в выборке.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i = y} w(i, u)$$

$$w(i, u) = [i = 1]$$

- + Простота реализации (lazy learning)
- + Интерпретируемость решения
- Неустойчивость к шуму
- Отсутствие настраиваемых параметров



# Метод ближайшего соседа

Объект относится к тому классу, к которому относится ближайший в выборке.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i = 1]$$

- + Простота реализации (lazy learning)
- + Интерпретируемость решения
- Неустойчивость к шуму
- Отсутствие настраиваемых параметров
- Низкое качество классификации

# Метод ближайшего соседа

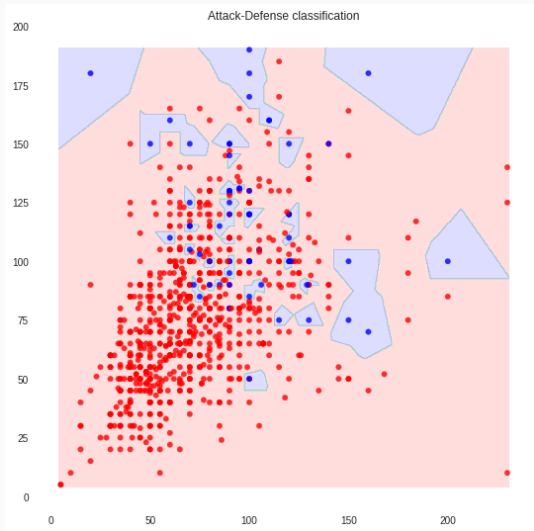
Объект относится к тому классу, к которому относится ближайший в выборке.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i = 1]$$

- + Простота реализации (lazy learning)
- + Интерпретируемость решения
- Неустойчивость к шуму
- Отсутствие настраиваемых параметров
- Низкое качество классификации
- **Необходимость хранить всю выборку целиком**

# Пример



## Метод k ближайших соседей

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i \leq k]$$

+ Менее чувствителен к шуму

## Метод k ближайших соседей

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i \leq k]$$

- + Менее чувствителен к шуму
- + Появляется настраиваемый параметр k

## Метод k ближайших соседей

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$$w(i, u) = [i \leq k]$$

- + Менее чувствителен к шуму
- + Появляется настраиваемый параметр k
- Неоднозначность при  $\sum_{y_i=y} w(i, u) = \sum_{y_i=s} w(i, u) \quad y \neq s$

# Подбор параметров

---

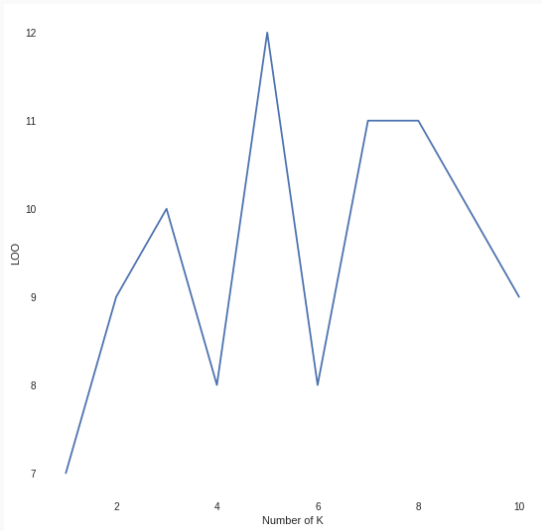
Функционал скользящего контроля (leave-one-out):

$$LOO(k, X^l) = \sum_{i=1}^l [a(x_i; X^l \setminus \{x_i\}, k) \neq y_i] \rightarrow \min_k$$



Правда ли нужно выбрасывать один объект?

# Пример



$$\rho(u, x_1) \leq \rho(u, x_2) \leq \dots \leq \rho(u, x_i)$$

$x_i$  –  $i$ -й сосед объекта  $u$

$y_i$  – класс  $i$ -го соседа  $u$

**Идея 1:** Посмотрим на ближайшие объекты и отнесем  $u$  к доминирующему классу.

# Обобщенный метрический классификатор

$$\rho(u, x_1) \leq \rho(u, x_2) \leq \dots \leq \rho(u, x_l)$$

$x_i$  –  $i$ -й сосед объекта  $u$

$y_i$  – класс  $i$ -го соседа  $u$

**Идея 1:** Посмотрим на ближайшие объекты и отнесем  $u$  к доминирующему классу.

**Идея 2:** Более близкие объекты важнее для классификации.

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$w(i, u) = [i \leq k] * w_i$ , где  $w_i$  это вес, зависящий только от номера соседа

## Метод $k$ взвешенных соседей

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$w(i, u) = [i \leq k] * w_i$ , где  $w_i$  это вес, зависящий только от номера соседа

Возможные эвристики:

- $w_i = \frac{k+1-i}{k}$  – линейные убывающие веса

# Метод $k$ взвешенных соседей

$$a(u, X^l) = \arg \max_{y \in Y} \sum_{y_i=y} w(i, u)$$

$w(i, u) = [i \leq k] * w_i$ , где  $w_i$  это вес, зависящий только от номера соседа

Возможные эвристики:

- $w_i = \frac{k+1-i}{k}$  – линейные убывающие веса
- $w_i = q^i$  – экспоненциально убывающие веса

Как более обоснованно задать веса?



## Метод окна Парзена

$K(r)$  – ядро, невозрастающее, положительное на  $[0, \infty]$

## Метод окна Парзена

$K(r)$  – ядро, невозрастающее, положительное на  $[0, \infty]$

Фиксированной ширины:

$$a(u, X^l, h, K) = \arg \max_{y \in Y} \sum_{y_i=y} K\left(\frac{\rho(u, x_i)}{h}\right) \quad h - \text{ширина окна}$$

## Метод окна Парзена

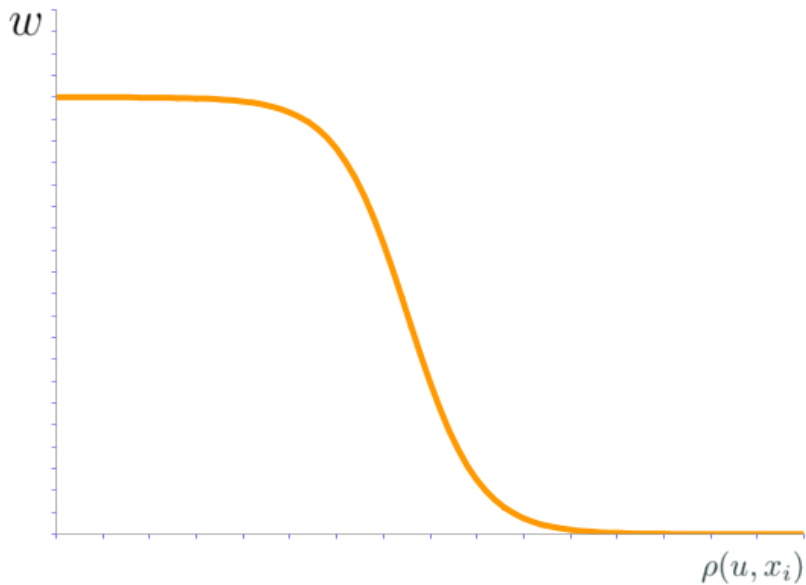
$K(r)$  – ядро, невозрастающее, положительное на  $[0, \infty]$

Фиксированной ширины:

$$a(u, X^l, h, K) = \arg \max_{y \in Y} \sum_{y_i=y} K\left(\frac{\rho(u, x_i)}{h}\right) \quad h - \text{ширина окна}$$

Переменной ширины:

$$a(u, X^l, k, K) = \arg \max_{y \in Y} \sum_{y_i=y} K\left(\frac{\rho(u, x_i)}{\rho(u, x_k)}\right)$$



# Выбор метрики

---

**Идея:** Максимизировать сумму расстояний между объектами разных классов при этом сохраняя сумму расстояний между объектами одного класса небольшой.

**Идея:** Максимизировать сумму расстояний между объектами разных классов при этом сохраняя сумму расстояний между объектами одного класса небольшой.

$$\max \sum_{x_i \in D, x_j \in F} \rho(x_i, x_j) \quad D \neq F$$

**Идея:** Максимизировать сумму расстояний между объектами разных классов при этом сохраняя сумму расстояний между объектами одного класса небольшой.

$$\max \sum_{x_i \in D, x_j \in F} \rho(x_i, x_j) \quad D \neq F$$

$$\sum_{x_i, x_j \in S} \rho^2(x_i, x_j) \leq 1$$



Если используемая метрика  $\rho(u, x_i)$  основана на суммировании различий по всем признакам, а число признаков очень велико, то все точки выборки могут оказаться практически одинаково далеки друг от друга.

Набор признаков объекта генерируется подбрасыванием честной монетки  $n$  раз. Соответственно каждый объект описывается вектором  $[0, 1]^n$ . При таких условиях все объекты будут равноудалены.

# Предобработка

---

Что произойдет, если признаки представлены в разном масштабе?

Все признаки должны быть представлены **в одном масштабе**.  
В противном случае признак с наибольшими числовыми значениями будет доминировать в метрике.

# Отбор признаков

---

1.  $\rho_j(u, x_i) = |u^j - x_i^j|$  – расстояние по  $j$ -му признаку  
 $LOO(j) \rightarrow \min$

1.  $\rho_j(u, x_i) = |u^j - x_i^j|$  – расстояние по  $j$ -му признаку  
 $LOO(j) \rightarrow \min$
2. Добавляем признак и строим  $\rho'$   
 $\rho'(u, x_i) = \rho(u, x_i) + w_j \rho_j(u, x_i)$   
 $LOO(j, w_j) \rightarrow \min$



# Жадное добавление признаков

1.  $\rho_j(u, x_i) = |u^j - x_i^j|$  – расстояние по j-му признаку  
 $LOO(j) \rightarrow \min$

2. Добавляем признак и строим  $\rho'$   
 $\rho'(u, x_i) = \rho(u, x_i) + w_j \rho_j(u, x_i)$   
 $LOO(j, w_j) \rightarrow \min$

3. Заменяем признак  
 $\rho'(u, x_i) = \rho(u, x_i) - w_k \rho_k(u, x_i) + w_j \rho_j(u, x_i)$

# Жадное добавление признаков

1.  $\rho_j(u, x_i) = |u^j - x_i^j|$  – расстояние по  $j$ -му признаку  
 $LOO(j) \rightarrow \min$
2. Добавляем признак и строим  $\rho'$   
 $\rho'(u, x_i) = \rho(u, x_i) + w_j \rho_j(u, x_i)$   
 $LOO(j, w_j) \rightarrow \min$
3. Заменяем признак  
 $\rho'(u, x_i) = \rho(u, x_i) - w_k \rho_k(u, x_i) + w_j \rho_j(u, x_i)$
4. Добавляем признаки, пока LOO не увеличивается

- Проблема хранения

- Проблема хранения
- Проблема быстрого поиска ближайших соседей

# Отбор эталонов

---

$$a(u, X^l) = \arg \max_{y \in Y} \underbrace{\sum_{y_i=y} w(i, u)}_{\Gamma_y(u)}$$

$w(i, u)$  - вес  $i$ -го соседа  $u$ , неотрицателен

$\Gamma_y(u)$  - оценка близости объекта  $u$  к классу  $y$

$\Gamma_y(u) = \sum_{y_i=y} w(i, u)$  – оценка близости объекта  $u$  к классу  $y$

Отступ показывает степень **типичности объекта**.

**Отступом** объекта  $x_i \in X^l$  относительно классификатора  $a$  называется величина:

$$M(x_i) = \Gamma_{y_i}(x_i) - \max_{y \in Y \setminus y_i} \Gamma_y(x_i)$$

## 1. Эталонные



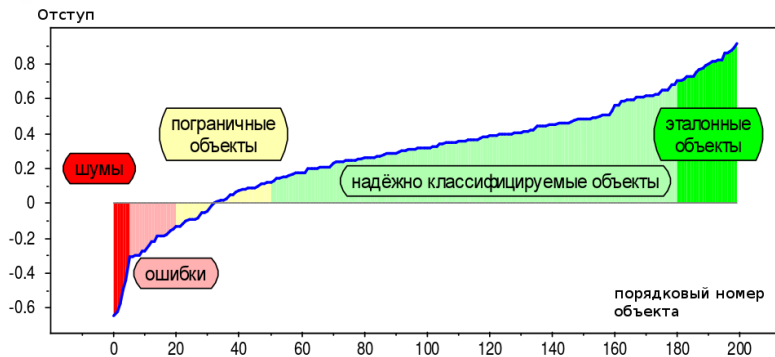
1. Эталонные
2. Надёжно классифицируемые (неинформативные)

1. Эталонные
2. Надёжно классифицируемые (неинформативные)
3. Пограничные

1. Эталонные
2. Надёжно классифицируемые (неинформативные)
3. Пограничные
4. Ошибочные

1. Эталонные
2. Надёжно классифицируемые (неинформативные)
3. Пограничные
4. Ошибочные
5. Шумовые

# Типы объектов



**Задача:** Выбрать оптимальное подмножество эталонов  $\Omega \subseteq X^l$

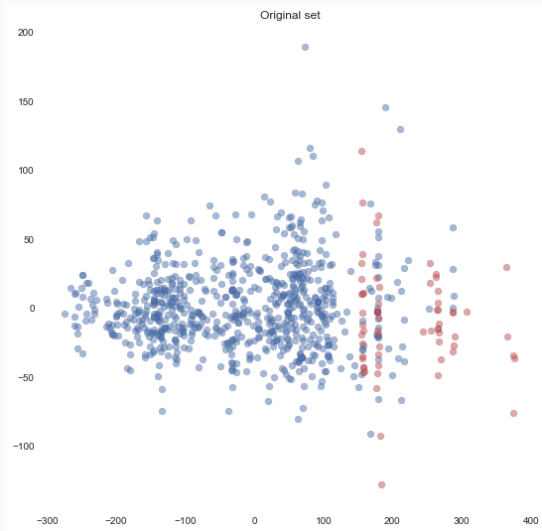
Классификатор будет иметь вид:

$$a(u, \Omega) = \arg \max_{y \in Y} \sum_{x_i \in \Omega} [y_i = y] w(i, u)$$

# Condensed Nearest Neighbor

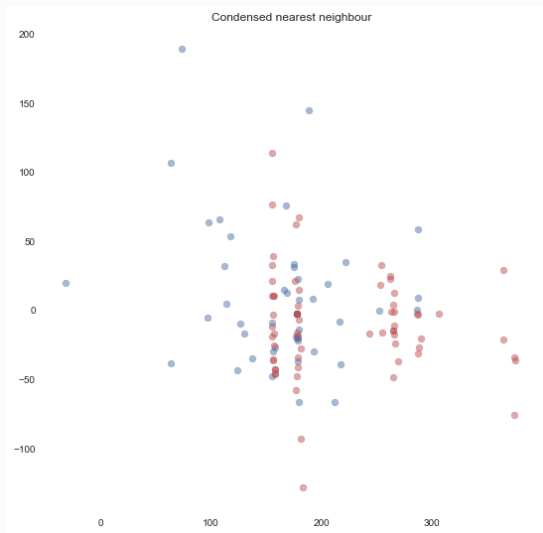
1. Исключить ошибочные, шумовые и пограничные объекты
2. Найти по одному эталону в каждом классе
3. Добавлять каждый следующий объект  $x$  в  $\Omega$ , если классификация с текущим набором эталонов ошибается на нём
4. Продолжать до тех пор пока  $\Omega$  не перестанет пополняться

# Condensed Nearest Neighbor





# Condensed Nearest Neighbor



+ Сокращается число хранимых объектов

# Condensed Nearest Neighbor

- + Сокращается число хранимых объектов
- + Сокращается время классификации

# Condensed Nearest Neighbor

- + Сокращается число хранимых объектов
- + Сокращается время классификации
- + Объекты разделяются по величине отступа

# Condensed Nearest Neighbor

- + Сокращается число хранимых объектов
- + Сокращается время классификации
- + Объекты разделяются по величине отступа
- Очень медленный

# Condensed Nearest Neighbor

- + Сокращается число хранимых объектов
- + Сокращается время классификации
- + Объекты разделяются по величине отступа
- Очень медленный
- Классификация на тестовом наборе может отличаться

Вопросы?

# Быстрый поиск ближайших соседей

---



## Быстрый поиск ближайших соседей

- граф ближайших соседей
- k-d дерево
- хеширование (LSH)

**Идея:** разложим множество по которому будем искать в бинарное дерево с простыми условиями и конкретными точками в узлах.

**Идея:** разложим множество по которому будем искать в бинарное дерево с простыми условиями и конкретными точками в узлах.

1. По циклу, или рандомно выбираем ось.
2. Ищем медиану (точку, разбивающую множество на как можно более равные части).
3. Повторяем 1-2 для каждого из получившихся подмножеств

# k-d дерево

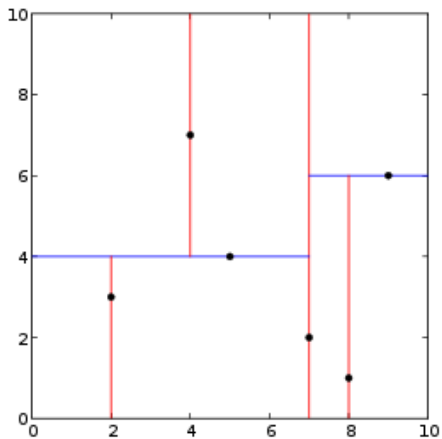
**Идея:** разложим множество по которому будем искать в бинарное дерево с простыми условиями и конкретными точками в узлах.

1. По циклу, или случайно выбираем ось.
2. Ищем медиану (точку, разбивающую множество на как можно более равные части).
3. Повторяем 1-2 для каждого из получившихся подмножеств

Сложность построения:  $O(n \log n)$

Сложность поиска: в лучшем случае  $O(\log n)$ , в худшем –  $O(n)$

## 2-d дерево



+ Один из наиболее простых методов

## k-d дерево. Особенности

- + Один из наиболее простых методов
- Работает только при малом количестве параметров

## k-d дерево. Особенности

- + Один из наиболее простых методов
- Работает только при малом количестве параметров
- Затратный алгоритм перестроения



# Locality Sensitive Hash

**Задача:** Найти похожие документы в интернете

# Locality Sensitive Hash

**Проблема:** Сколько сравнений нам понадобится для того, чтобы найти похожие среди  $N$  документов?

# Locality Sensitive Hash

**Проблема:** Сколько сравнений нам понадобится для того, чтобы найти похожие среди  $N$  документов?

$$C = \frac{N(N-1)}{2}$$

$$N = 10^6 \Rightarrow C = 5 * 10^{11}$$

# Locality Sensitive Hash

**Идея:** Давайте от каждого документа (строки из нулей и единиц) возьмем хэш  $h$ :

1. Если документы  $C_1$  и  $C_2$  похожи, то с большой вероятностью  $h(C_1) == h(C_2)$

# Locality Sensitive Hash

**Идея:** Давайте от каждого документа (строки из нулей и единиц) возьмем хэш  $h$ :

1. Если документы  $C_1$  и  $C_2$  похожи, то с большой вероятностью  $h(C_1) == h(C_2)$
2. Иначе – с большой вероятностью  $h(C_1) \neq h(C_2)$

# Locality Sensitive Hash

## Идея:

1. Разбить документ на  $n$ -граммы
2. Взять от каждого  $n$ -грамма хэш
3. Получим представление документа в виде строки из нулей и единиц. Длина такого вектора = количество всевозможных  $n$ -грамм.
4. Посчитаем документы похожими, если у них много совпадающих  $n$ -грамм

# Min Hash

Перестановка

|   |   |   |
|---|---|---|
| 5 | 3 | 6 |
| 1 | 2 | 4 |
| 2 | 4 | 1 |
| 7 | 1 | 2 |
| 6 | 7 | 7 |
| 4 | 5 | 5 |
| 3 | 6 | 3 |

|   | Документ 1 | Документ 2 |   |   |
|---|------------|------------|---|---|
| 1 | 1          | 0          | 1 | 0 |
| 2 | 1          | 0          | 0 | 0 |
| 3 | 0          | 1          | 0 | 1 |
| 4 | 0          | 1          | 0 | 1 |
| 5 | 0          | 1          | 0 | 1 |
| 6 | 1          | 0          | 1 | 0 |
| 7 | 1          | 0          | 1 | 1 |

Номер первой строки с единицей в перестановке



Наличие i-го n-грамма

|   |   |   |   |
|---|---|---|---|
| 1 | 2 | 3 | 2 |
| 2 | 1 | 3 | 1 |
| 3 | 1 | 3 | 1 |

## Что почитать по этой лекции

- Tom Mitchell "Machine Learning". Chapter 8
- К. Воронцов "Лекции по метрическим алгоритмам классификации"
- Обзор методов поиска ближайших соседей
- Ullman, Leskovec, Rajaraman "Mining of Massive Datasets" Chapter 3.4—3.8



# Что происходит сейчас в области knn

ICML'16: Fast  $k$ -Nearest Neighbour Search via Dynamic Continuous Indexing

NIPS'16:  $k^*$ -Nearest Neighbors: From Global to Local

NIPS'16: Finite-Sample Analysis of Fixed- $k$  Nearest Neighbor Density Functional Estimators

## На следующей лекции

- Кластеризация. K-means.
- Цели кластеризации.
- Типы кластерных структур.
- Функционал качества кластеризации
- K-средних
- Иерархическая кластеризация.