

# Основы машинного обучения

Лекция 3

Метод k ближайших соседей

Евгений Соколов

[esokolov@hse.ru](mailto:esokolov@hse.ru)

НИУ ВШЭ, 2024

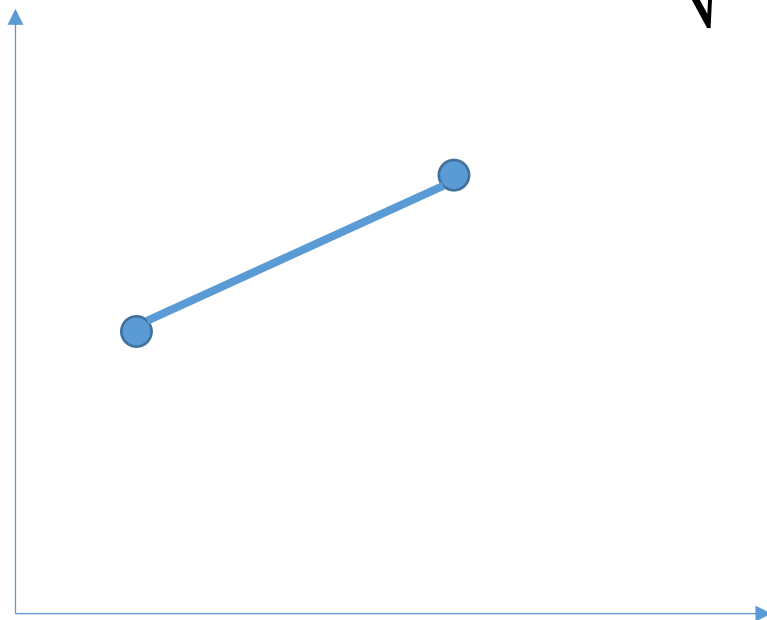
Сравнение объектов и метрики

# Числовые данные

Сколько раз в день вызывает такси	Средние расходы на такси в день	Как часто вызывал комфорт	Возраст	Согласился повысить категорию?
2	400	0.3	29	да
0.3	80	0	28	нет
...	...	...	...	...

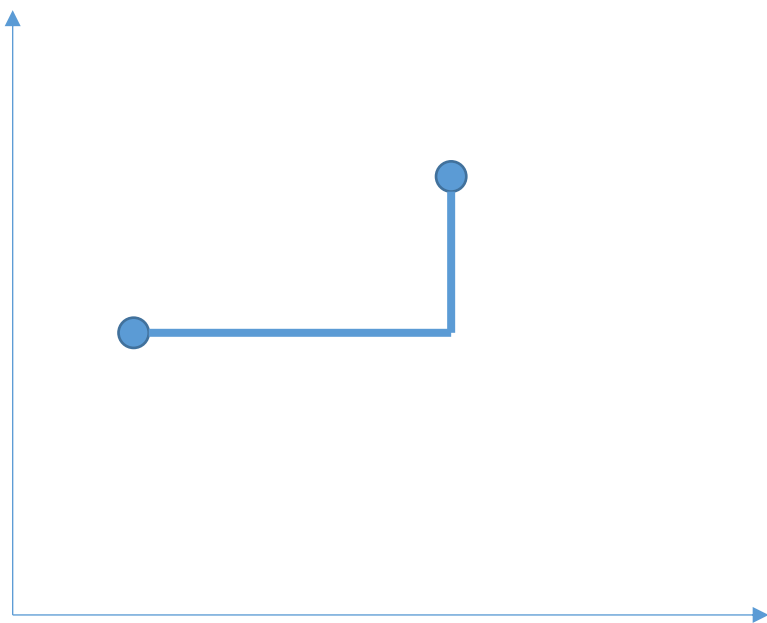
# Евклидова метрика

$$\rho(x, z) = \sqrt{\sum_{j=1}^d (x_j - z_j)^2}$$



# Манхэттенская метрика

$$\rho(x, z) = \sum_{j=1}^d |x_j - z_j|$$



# Обобщение

$$\rho(x, z) = \sqrt[p]{\sum_{j=1}^d |x_j - z_j|^p}$$

- Метрика Минковского
- Можно подбирать  $p$  под конкретную задачу

# Категориальные данные

На каком классе чаще всего ездит	Ближайшее к дому метро	Способ оплаты	Согласился повысить категорию?
Эконом	Таганская	Карта	да
Комфорт	Юго-Западная	Наличные	нет
...	...	...	...

# Считающая метрика

- Простейшая метрика: подсчёт различий

$$\rho(x, z) = \sum_{j=1}^d [x_j \neq z_j]$$



# Что ещё?

- Текстовые данные — чуть-чуть изучим в курсе, подробно потом
- Изображения — потом

Измерение ошибки модели

# Вопросы

- Как сравнить две модели?
- Как подобрать  $k$  и метрику?

# Функция потерь для классификации

- Частый выбор — бинарная функция потерь

$$L(y, a) = [a \neq y]$$

- Функционал ошибки — доля ошибок (error rate)

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) \neq y_i]$$

- Нередко измеряют долю верных ответов (accuracy):

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) = y_i]$$

# Функция потерь для классификации

ВАЖНО

Accuracy — не точность!

# Accuracy

$a(x)$	$y$
-1	-1
+1	+1
-1	-1
<b>+1</b>	<b>-1</b>
+1	+1

# Accuracy

$a(x)$	$y$
-1	-1
+1	+1
-1	-1
<b>+1</b>	<b>-1</b>
+1	+1

Доля ошибок: 0.2

Доля верных ответов: 0.8

# Accuracy

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) \neq y_i]$$

Решаем задачу выявления редкого заболевания

- 950 здоровых ( $y = +1$ )
- 50 больных ( $y = -1$ )

Модель:  $a(x) = +1$

**Доля ошибок: 0.05**



# Accuracy

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} [a(x_i) \neq y_i]$$

- Всегда смотрите на баланс классов!
- Доля верных ответов не обязательно меняется от 0.5 до 1 для разумных моделей

# Как выбрать k?

Обучающая выборка

На каком классе чаще всего ездит	Ближайшее к дому метро	Способ оплаты	Согласился повысить категорию?
Эконом	Таганская	Карта	да
Комфорт	Юго-Западная	Наличные	нет
Комфорт	Строгино	Карта	да

Применяем модель:

Эконом	Таганская	Карта	?
--------	-----------	-------	---

# Как выбрать $k$ ?

Обучающая выборка

На каком классе чаще всего ездит	Ближайшее к дому метро	Способ оплаты	Согласился повысить категорию?
Эконом	Таганская	Карта	да
Комфорт	Юго-Западная	Наличные	нет
Комфорт	Строгино	Карта	да

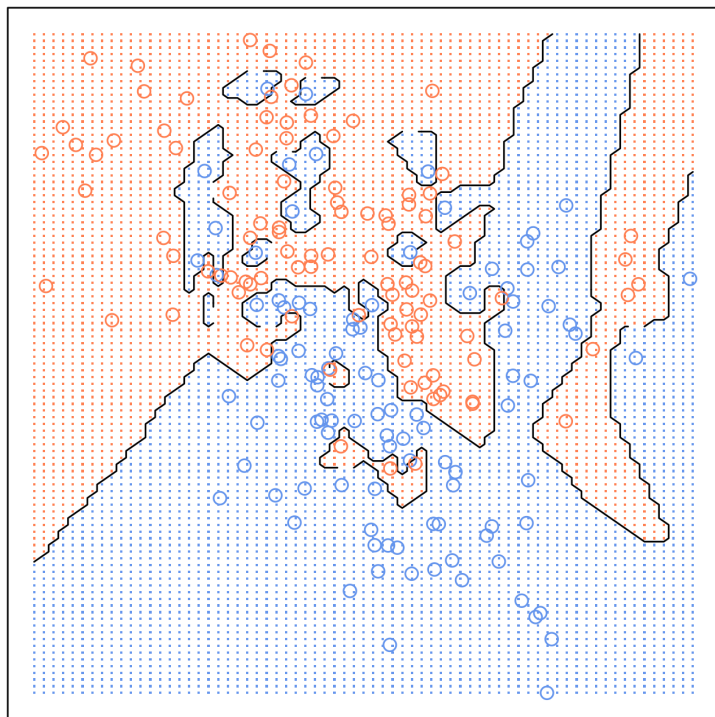
Применяем модель:

Эконом	Таганская	Карта	да
--------	-----------	-------	----

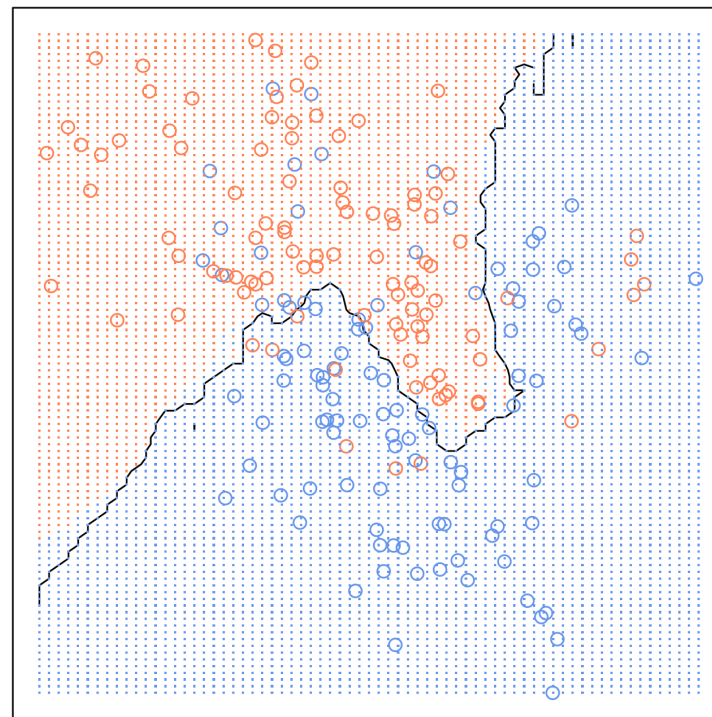
С точки зрения качества на обучающей выборке лучший выбор  $k = 1$

# Как выбрать $k$ ?

1-nearest neighbours



20-nearest neighbours



<https://kevinzakka.github.io/2016/07/13/k-nearest-neighbor/>

# Гиперпараметры

- Нельзя подбирать  $k$  по обучающей выборке — **гиперпараметр**
- Нужно использовать дополнительные данные

Обобщающая способность

# Обобщающая способность

Как готовиться к экзамену?

Заучить все примеры с  
занятий

Разобраться в предмете и  
усвоить алгоритмы решения  
задач

# Обобщающая способность

Как готовиться к экзамену?

Заучить все примеры с  
занятий

Разобраться в предмете и  
усвоить алгоритмы решения  
задач

Переобучение (overfitting)

Обобщение (generalization)



# Обобщающая способность

Как готовиться к экзамену?

Заучить все примеры с  
занятий

Разобраться в предмете и  
усвоить алгоритмы решения  
задач

Переобучение (overfitting)

Обобщение (generalization)

Хорошее качество на обучении  
Низкое качество на новых данных

Хорошее качество на обучении  
Хорошее качество на новых  
данных

# Отложенная выборка



Обучение



Тест

# Отложенная выборка



- Слишком большое обучение — тестовая выборка нерепрезентативна
- Слишком большой тест — модель не сможет обучиться
- Обычно: 70/30, 80/20

# Кросс-валидация

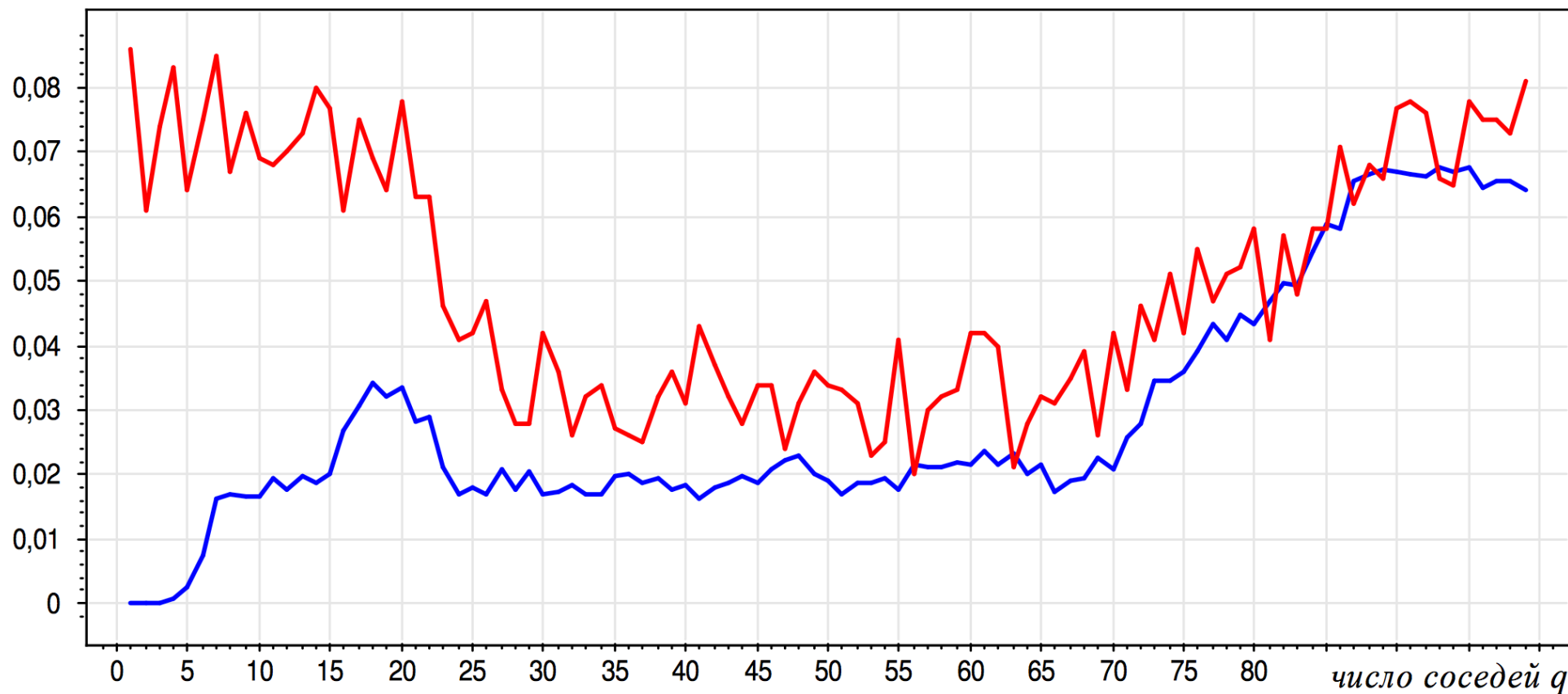


# Кросс-валидация

- Надёжнее отложенной выборки, но медленнее
- Параметр — количество разбиений  $n$  (фолдов, folds)
- Хороший, но медленный вариант —  $n = \ell$  (leave-one-out)
- Обычно:  $n = 3$  или  $n = 5$  или  $n = 10$

# Подбор числа соседей

*частота ошибок*



# Чуть больше терминов

- После подбора всех гиперпараметров стоит проверить на совсем новых данных, что модель работает
- Обучающая выборка — построение модели
- Валидационная выборка — подбор гиперпараметров модели
- Тестовая выборка — финальная оценка качества модели

Метод  $k$  ближайших соседей с  
весами



# kNN: применение

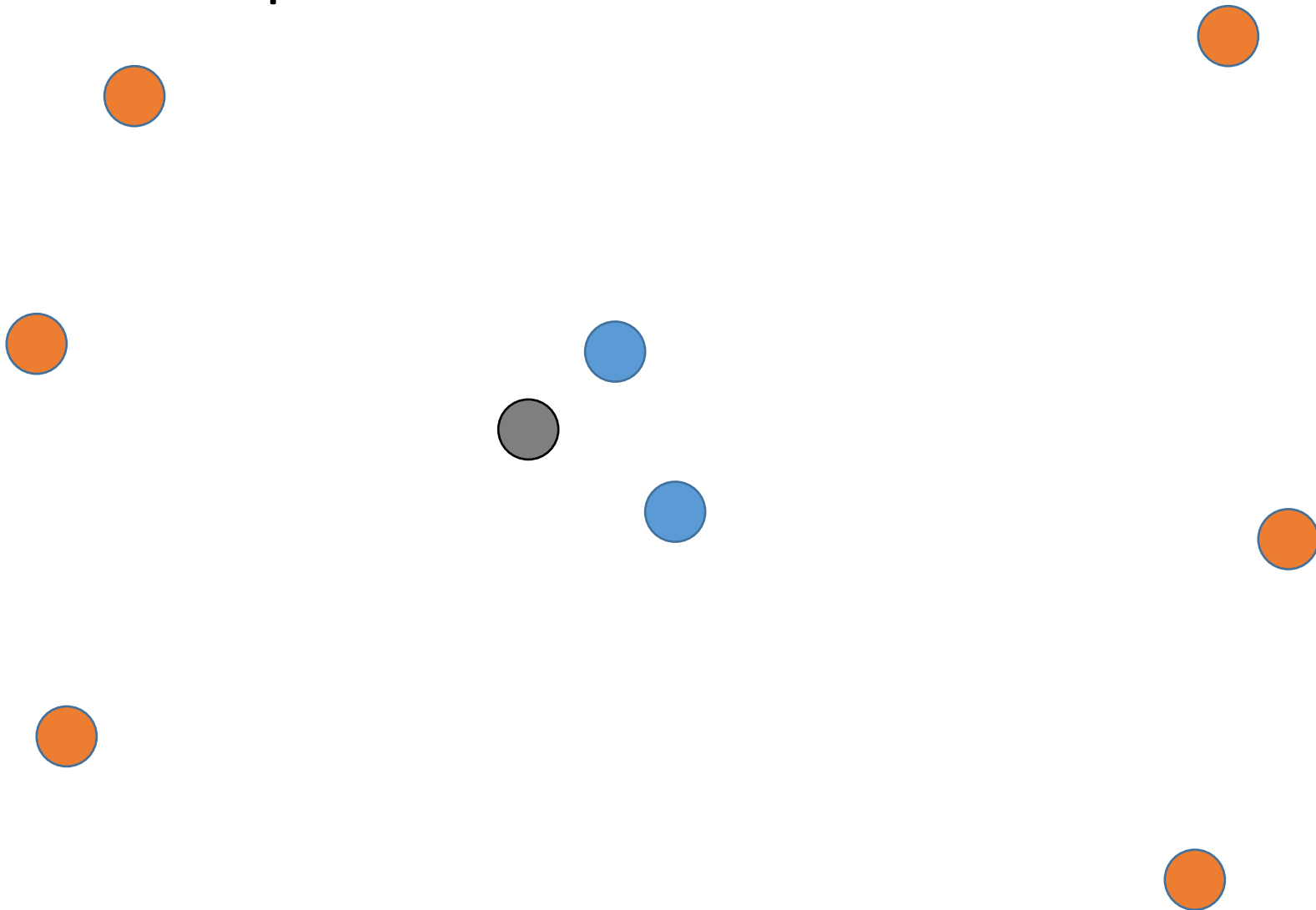
Дано: новый объект  $x$

Применение модели:

- Сортируем объекты обучающей выборки по расстоянию до нового объекта:  
 $\rho(x, x_{(1)}) \leq \rho(x, x_{(2)}) \leq \dots \leq \rho(x, x_{(\ell)})$
- Выбираем  $k$  ближайших объектов:  $x_{(1)}, \dots, x_{(k)}$
- Выдаём наиболее популярный среди них класс:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{i=1}^k [y_{(i)} = y]$$

# Проблема с расстояниями



# Взвешенный knn

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{i=1}^k w_i [y_{(i)} = y]$$

Варианты:

- $w_i = \frac{k+1-i}{k}$
- $w_i = q^i$
- Не учитывают сами расстояния

# Взвешенный knn

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{i=1}^k w_i [y_{(i)} = y]$$

Парзеновское окно:

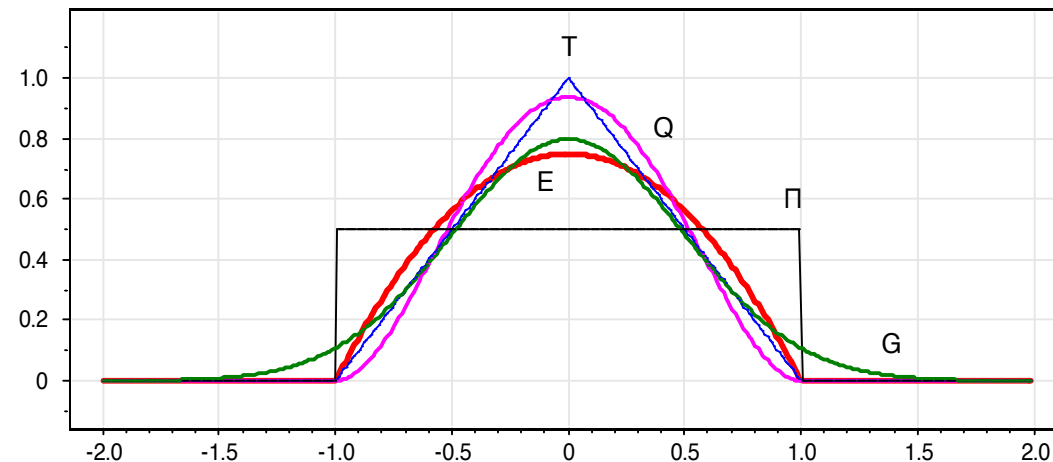
- $w_i = K \left( \frac{\rho(x, x_{(i)})}{h} \right)$
- $K$  — ядро
- $h$  — ширина окна

# Ядра для весов

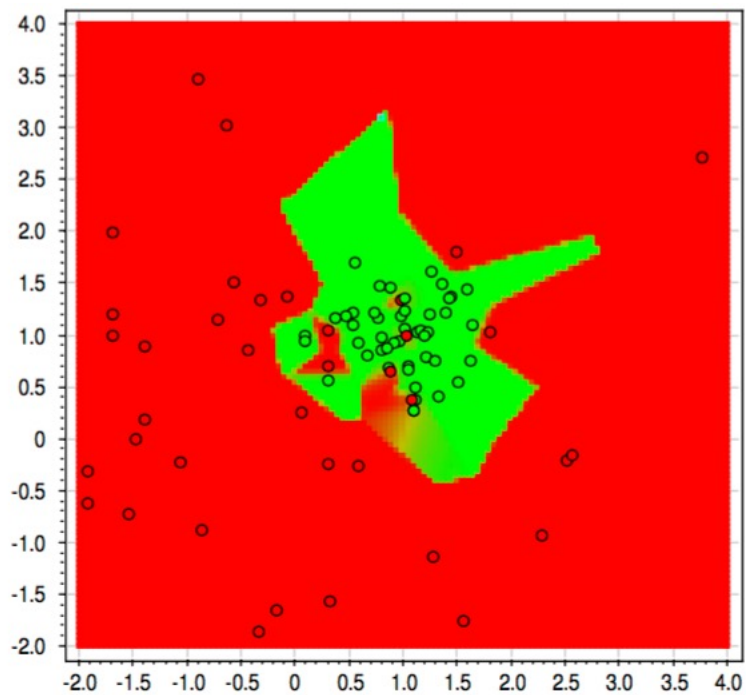
- Гауссовское ядро:

$$K(z) = (2\pi)^{-0.5} \exp\left(-\frac{1}{2}z^2\right)$$

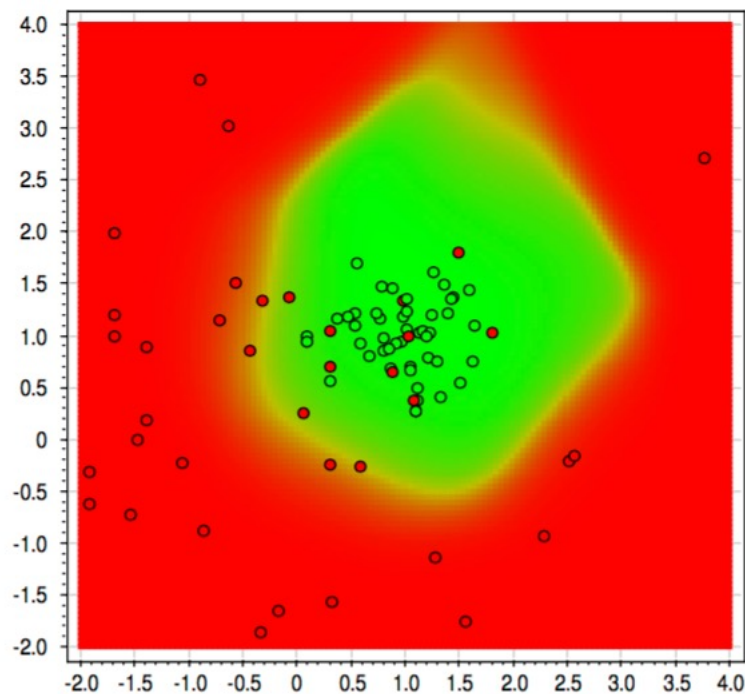
- И много других:



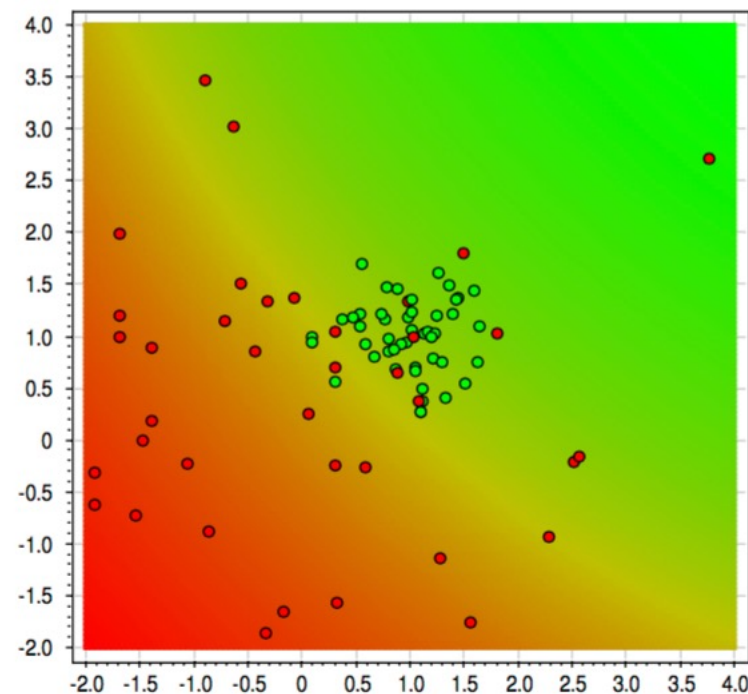
# Ядра для весов



$h = 0.05$



$h = 0.5$



$h = 5$

kNN для регрессии

# kNN: обучение

- Дано: обучающая выборка  $X = (x_i, y_i)_{i=1}^{\ell}$
- Задача регрессии (ответы из множества  $\mathbb{Y} = \mathbb{R}$ )
- Обучение модели:
  - Запоминаем обучающую выборку  $X$



# kNN: применение

Дано: новый объект  $x$

Применение модели:

- Сортируем объекты обучающей выборки по расстоянию до нового объекта:  
 $\rho(x, x_{(1)}) \leq \rho(x, x_{(2)}) \leq \dots \leq \rho(x, x_{(\ell)})$
- Выбираем  $k$  ближайших объектов:  $x_{(1)}, \dots, x_{(k)}$
- Усредняем ответы:

$$a(x) = \frac{1}{k} \sum_{i=1}^k y_{(i)}$$

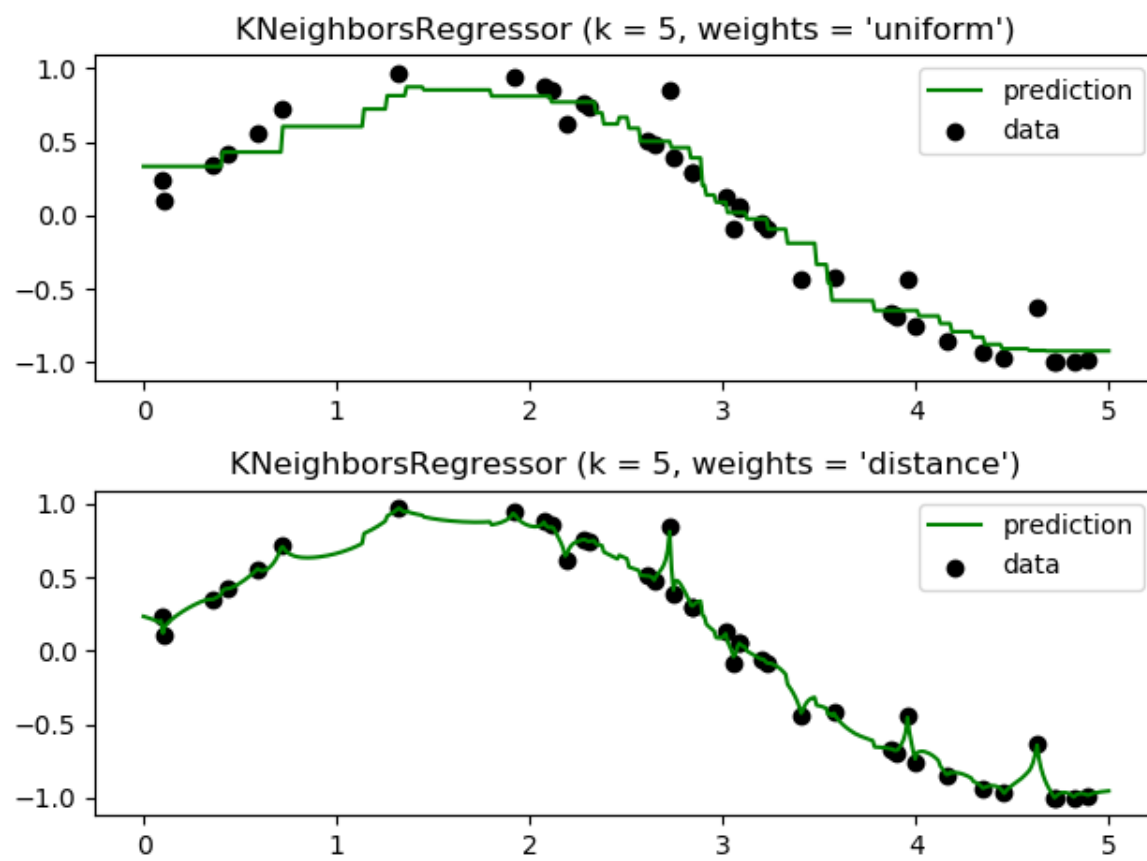
# kNN: применение

- Можно добавить веса:

$$a(x) = \frac{\sum_{i=1}^k w_i y_{(i)}}{\sum_{i=1}^k w_i}$$

- $w_i = K \left( \frac{\rho(x, x_{(i)})}{h} \right)$
- Формула Надарая-Ватсона

# kNN: применение



# Функция потерь для регрессии

- Частый выбор — квадратичная функция потерь

$$L(y, a) = (a - y)^2$$

- Функционал ошибки — среднеквадратичная ошибка (mean squared error, MSE)

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2$$

# Функция потерь для регрессии

- Ещё один вариант — средняя абсолютная ошибка (mean absolute error, MAE)

$$Q(a, X) = \frac{1}{\ell} \sum_{i=1}^{\ell} |a(x_i) - y_i|$$

- Слабее штрафует за серьёзные отклонения от правильного ответа

Резюме

# Плюсы kNN

- Если данных много и для любого объекта найдётся похожий в обучающей выборке, то это лучшая модель
- Очень простое обучение
- Мало гиперпараметров
- Бывают задачи, где гипотеза компактности уместна
  - Классификация изображений
  - Классификация текстов на много классов

# Минусы kNN

- Часто другие модели оказываются лучше
- Надо хранить в памяти всю обучающую выборку
- Искать  $k$  ближайших соседей довольно долго
- Мало способов настроить модель