

# Машинное обучение

Лекция 3

Метод k ближайших соседей

Линейная регрессия

Сергей Корпачев

НИУ ВШЭ, 2026

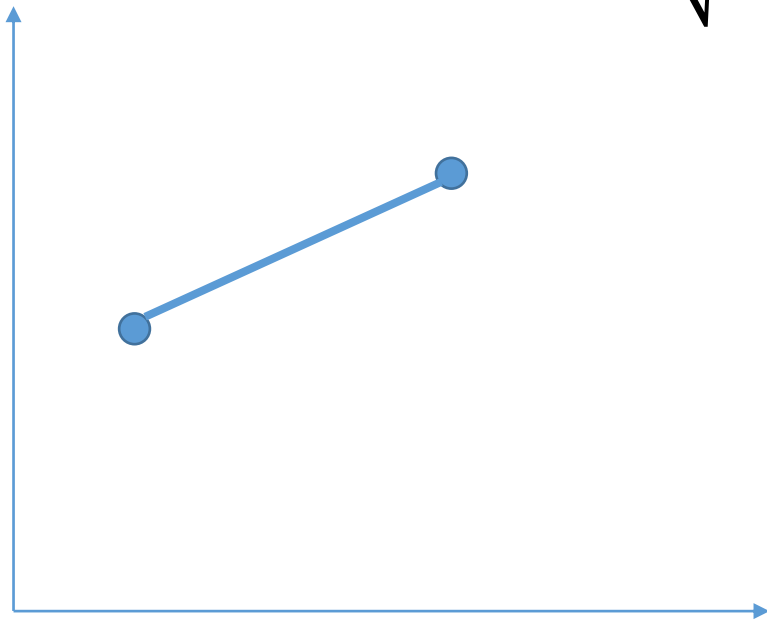
Сравнение объектов и метрики

# 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
...	...	...	...	...

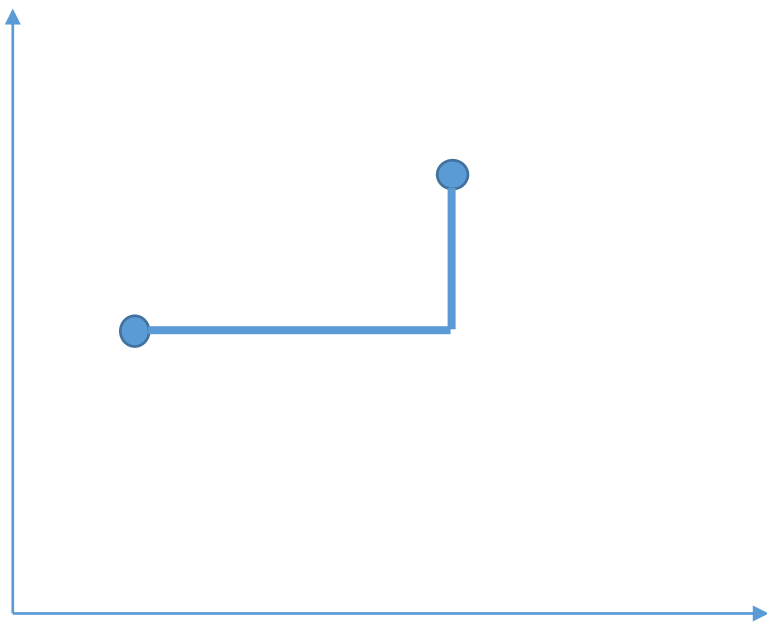
# Евклидова метрика

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



# 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 к?

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

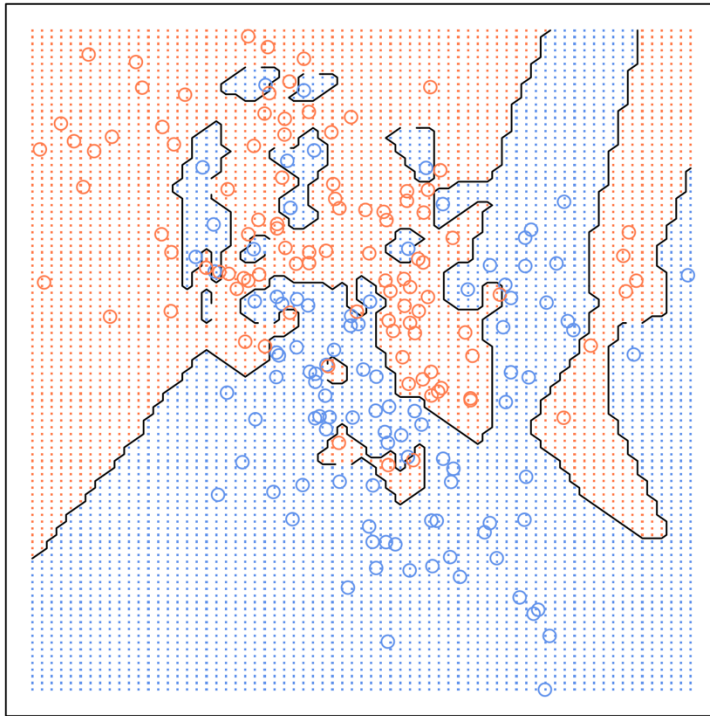
Применяем модель:

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

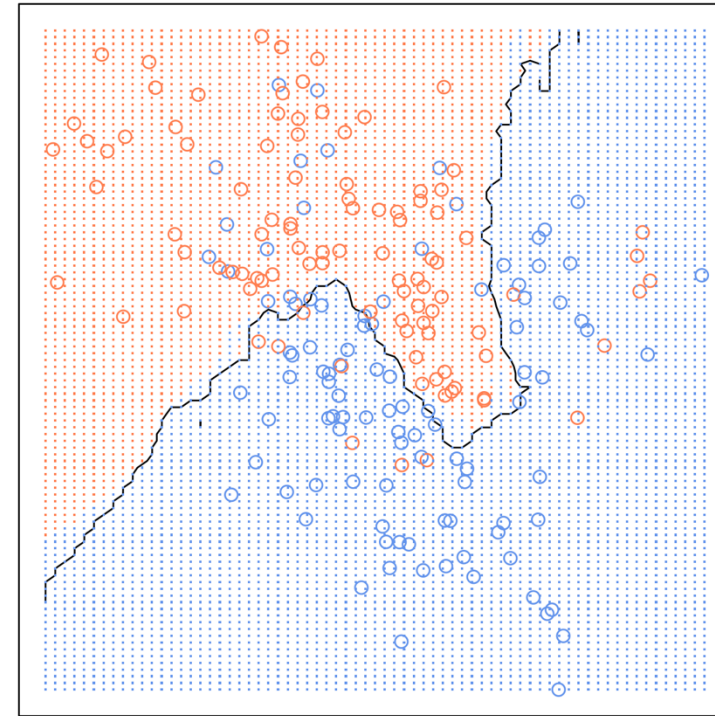
С точки зрения качества на обучающей выборке лучший выбор  $k = 1$

# 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

Переобучение (overfitting)

Хорошее качество на обучении  
Низкое качество на новых данных

Разобраться в предмете и  
усвоить алгоритмы решения  
задач

Обобщение (generalization)

Хорошее качество на обучении  
Хорошее качество на новых  
данных

# Отложенная выборка

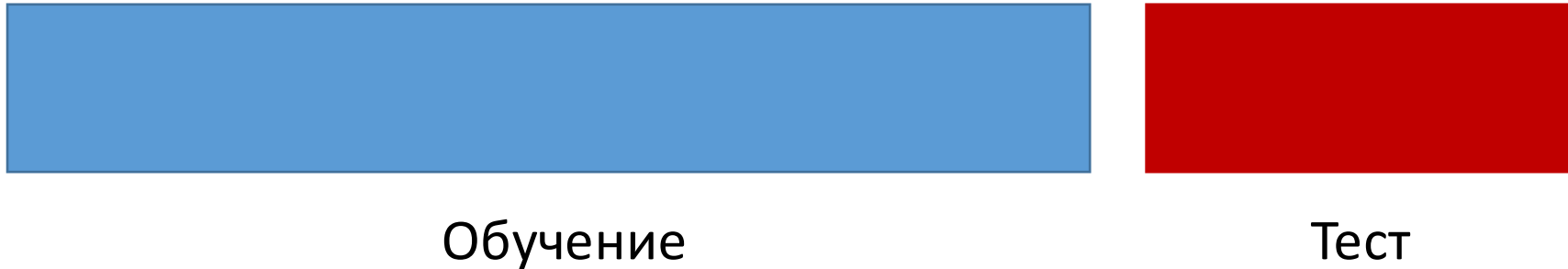


Обучение



Тест

# Отложенная выборка



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

# Кросс-валидация

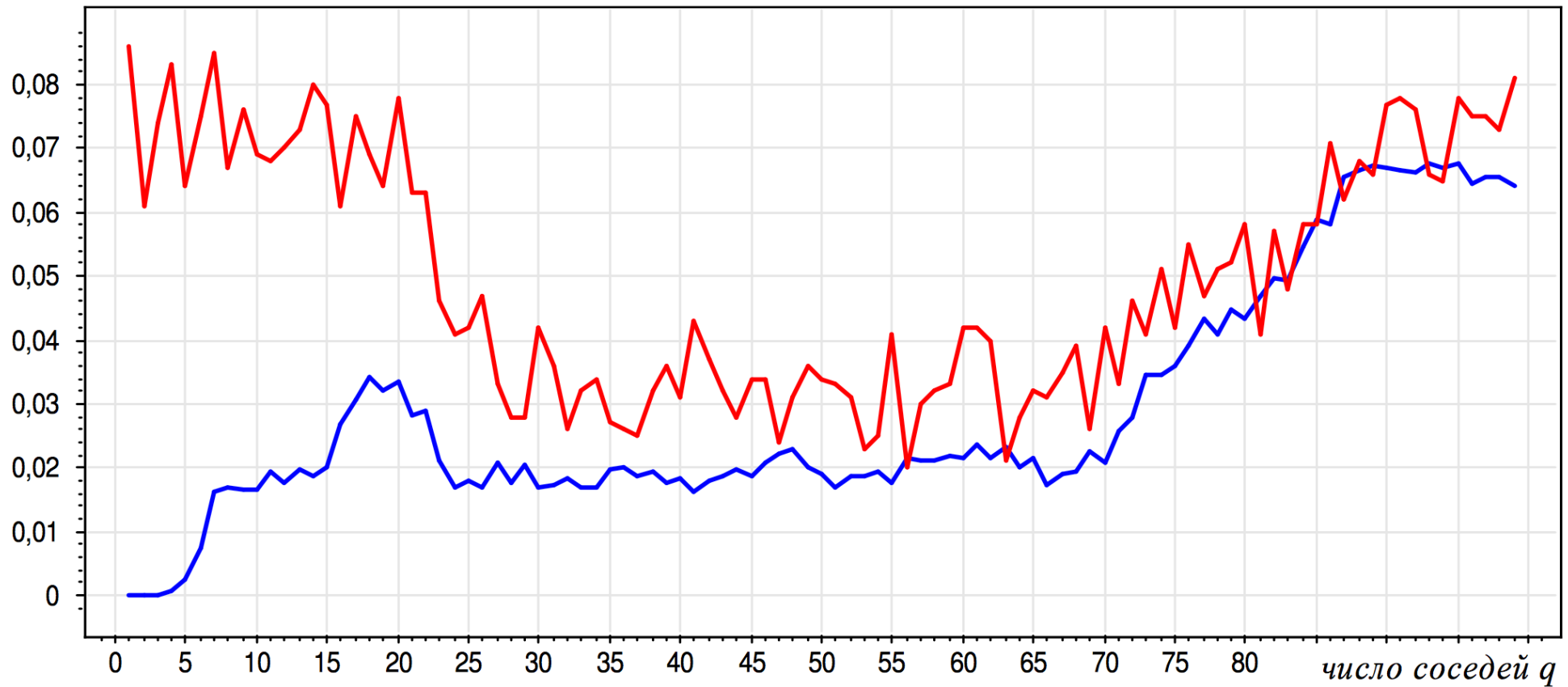


# Кросс-валидация

- 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*



<http://www.machinelearning.ru/wiki/index.php?title=MO>

# 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

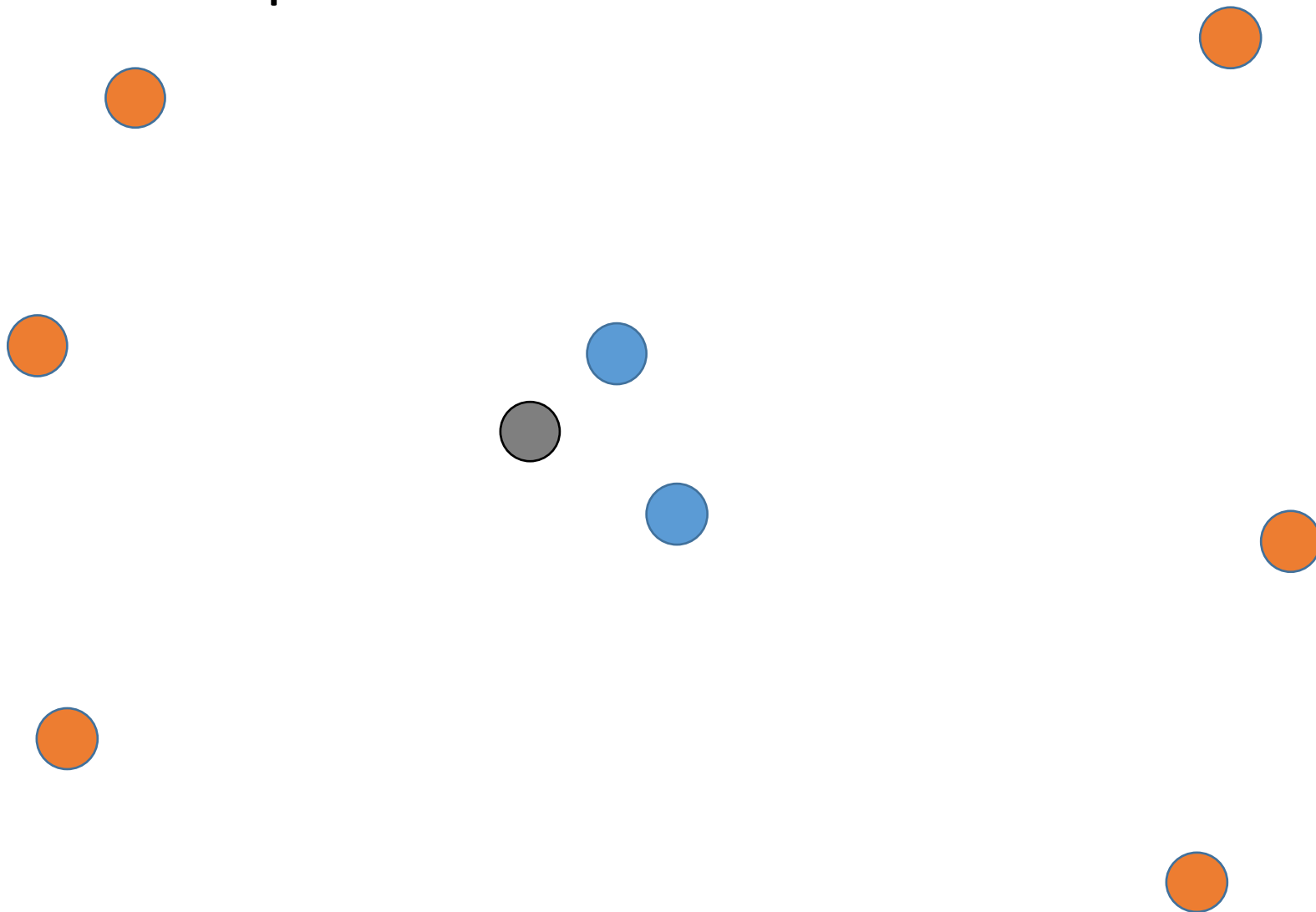
Дано: новый объект  $x$

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:

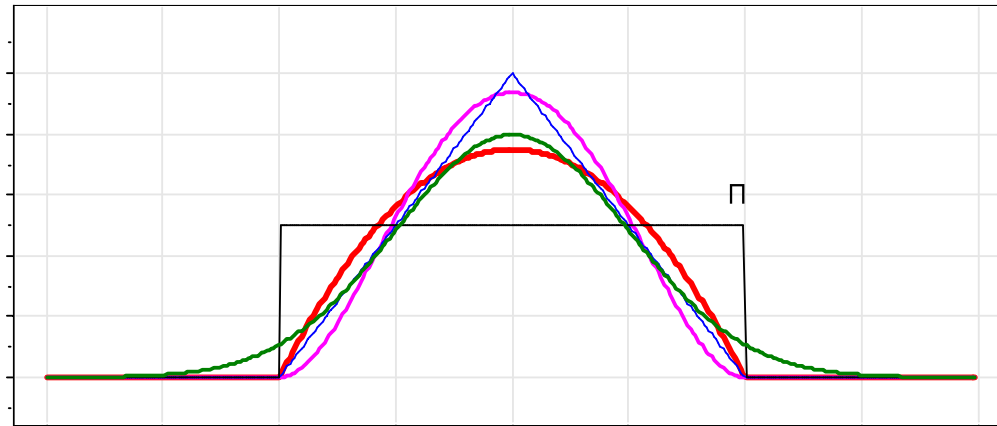
- $w_i = K \left( \frac{\rho(x, x_{(i)})}{h} \right)$
- $K$  — ядро
- $h$  — ширина окна

# Ядра для весов

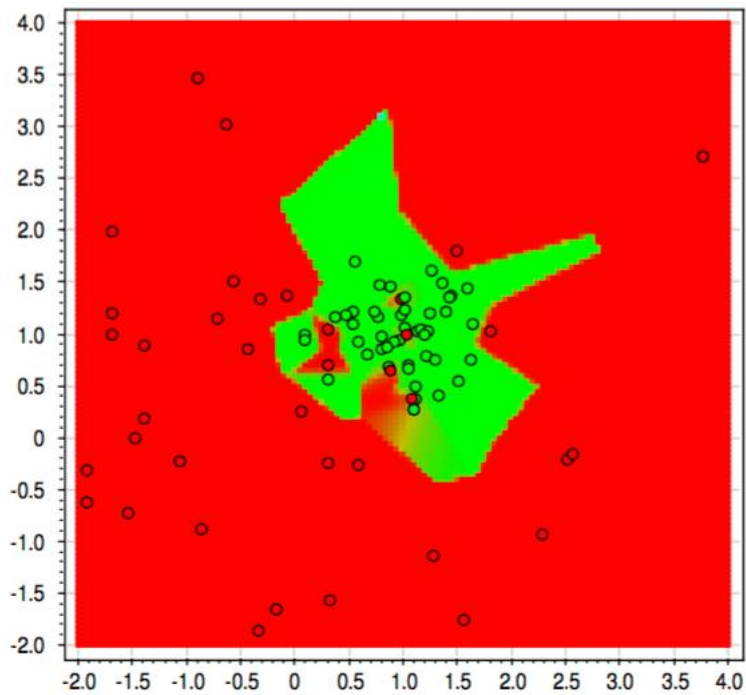
- Гауссовское ядро:

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

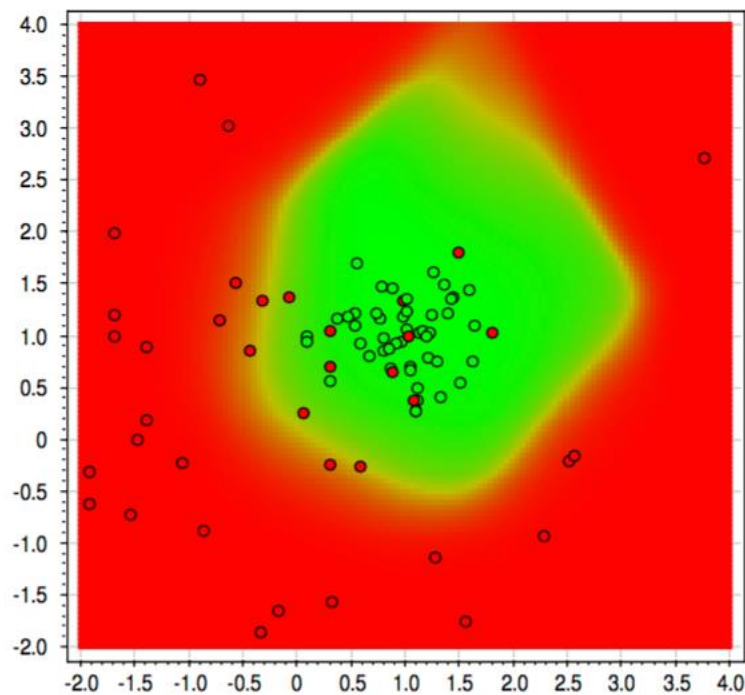
- И много других:



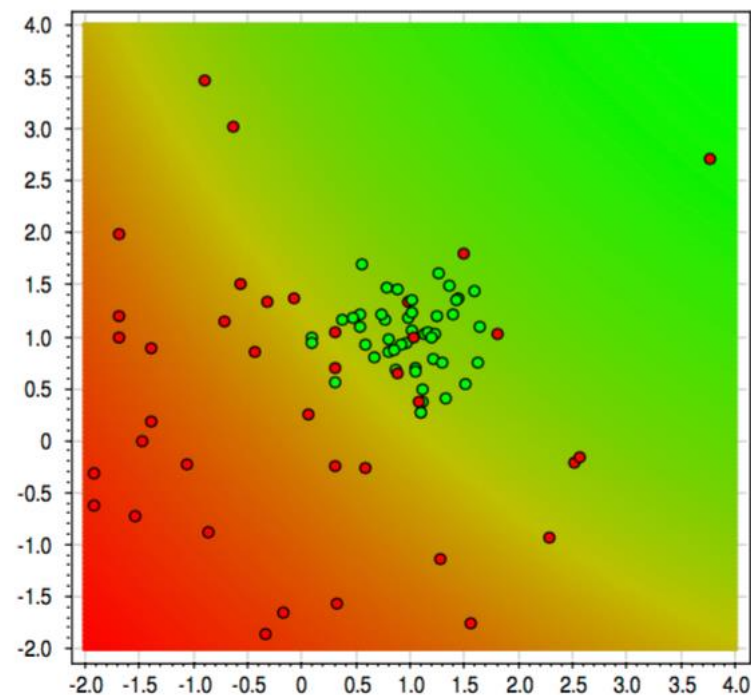
# Ядра для весов



$h = 0.05$



$h = 0.5$



$h = 5$

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:

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

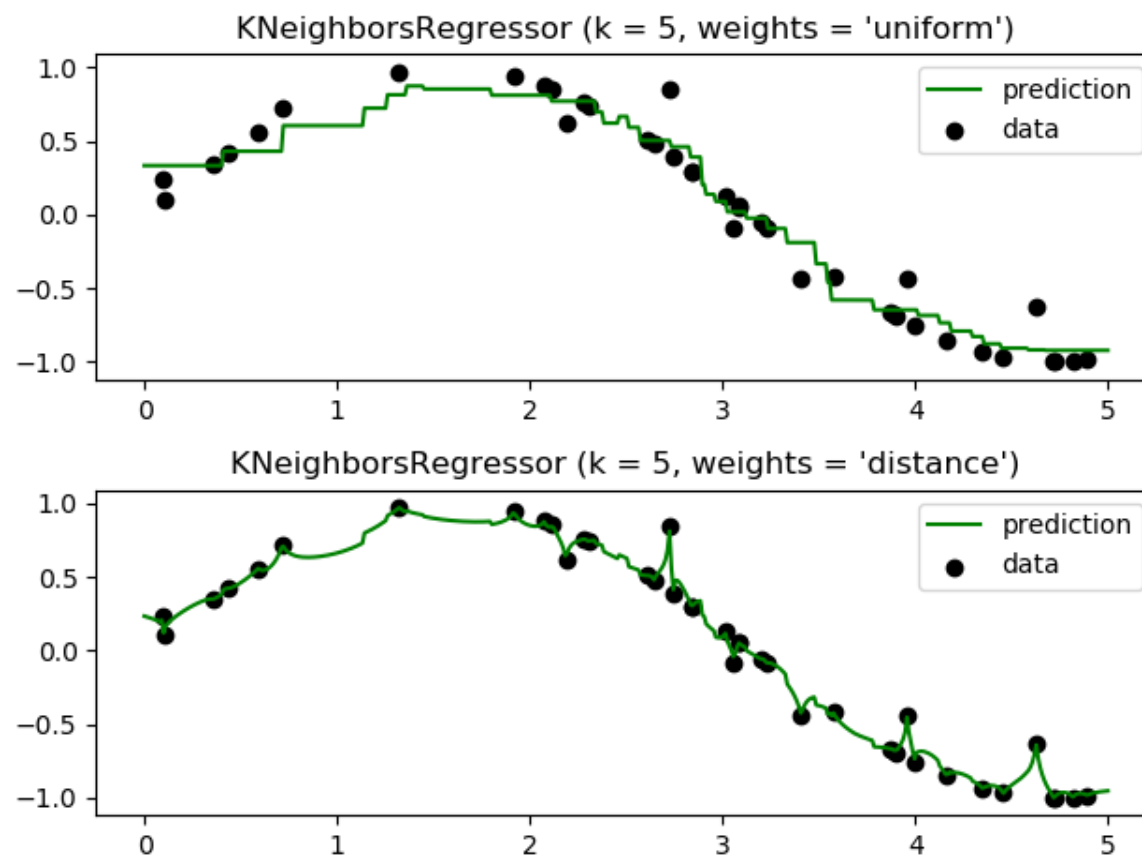
# kNN: применение

- 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

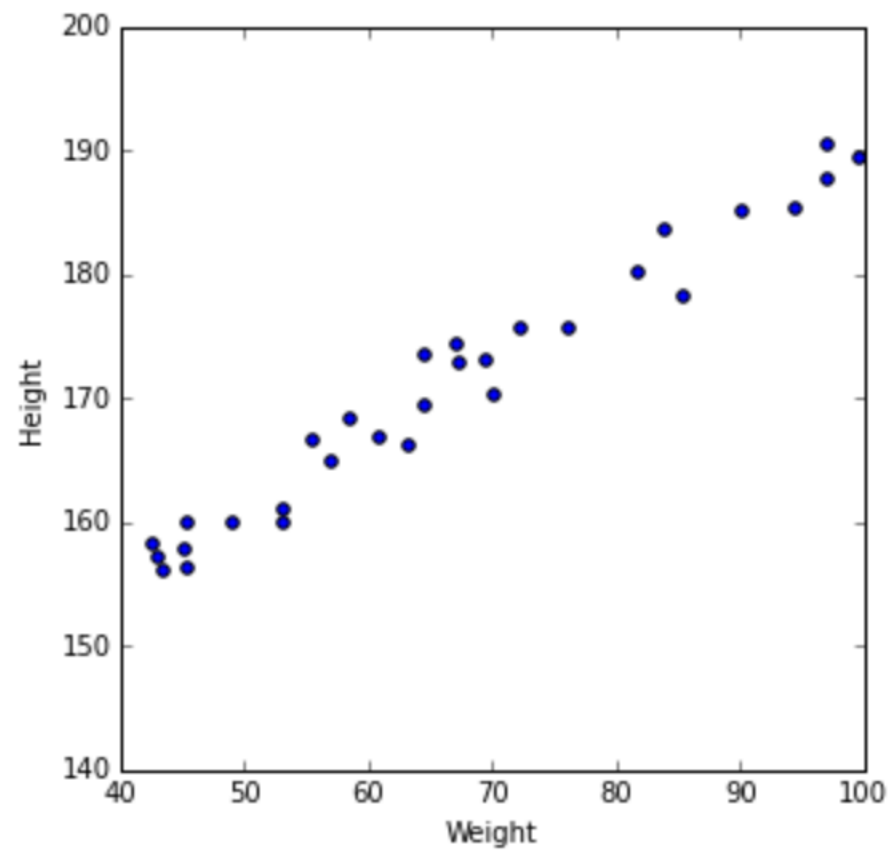
# Минусы kNN

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

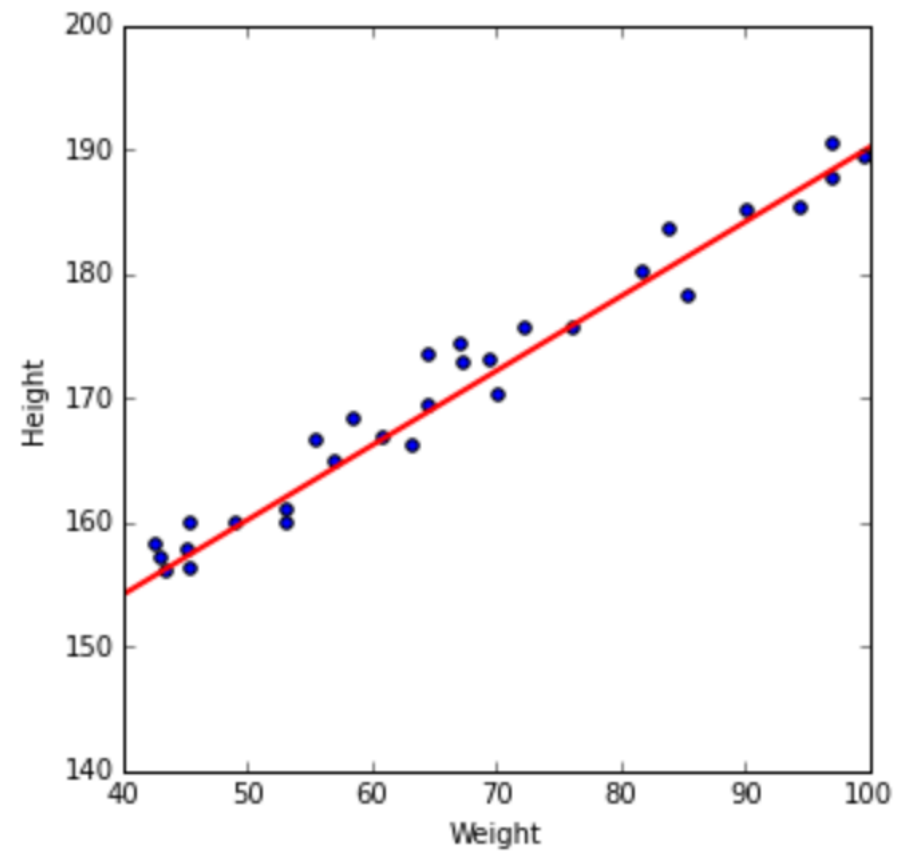


# Линейная регрессия

# Парная регрессия



# Парная регрессия



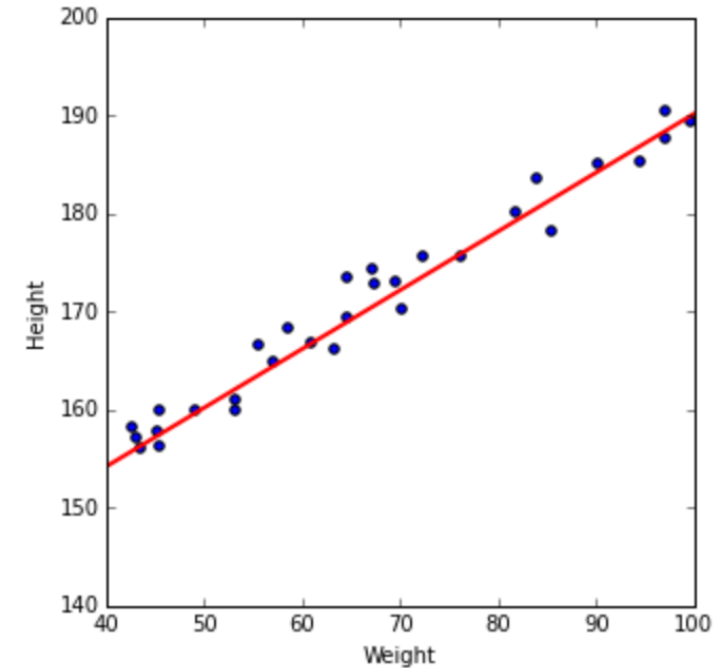
# Парная регрессия

- Простейший случай: один признак
- Модель:  $a(x) = w_1 x + w_0$
- Два параметра:  $w_1$  и  $w_0$
- $w_1$  — тангенс угла наклона
- $w_0$  — где прямая пересекает ось ординат

# Почему модель линейная?

$$a(x) = 2x + 1$$

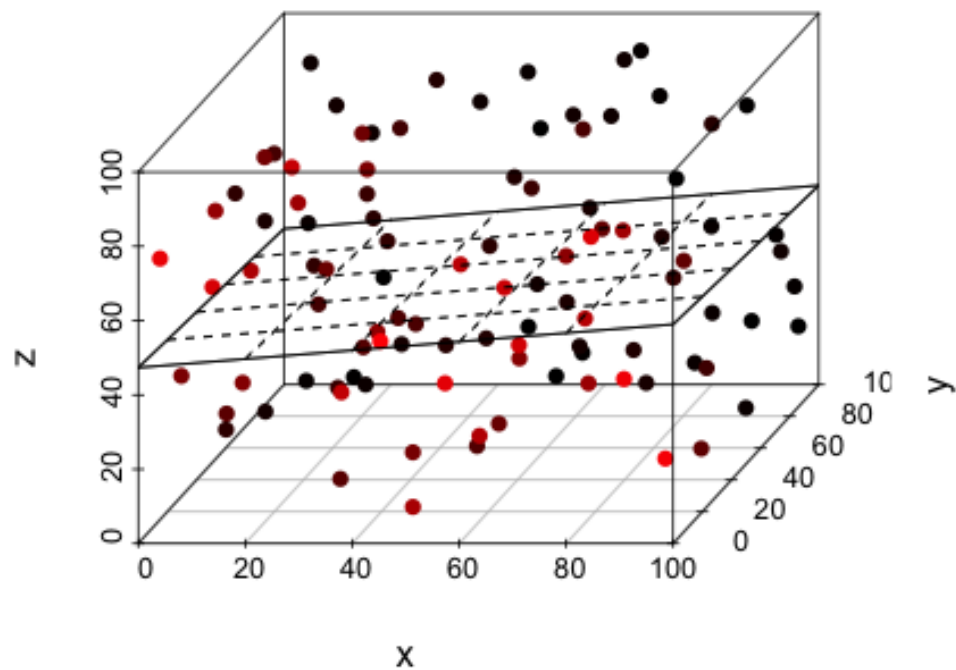
- $x = 1, a(x) = 3$
- $x = 2, a(x) = 5$
- $x = 10, a(x) = 21$
- $x = 20, a(x) = 41$



# Два признака

- Чуть более сложный случай: два признака
- Модель:  $a(x) = w_0 + w_1 x_1 + w_2 x_2$
- Три параметра

# Два признака



# Много признаков

- Общий случай:  $d$  признаков
- Модель

$$a(x) = w_0 + w_1x_1 + \dots + w_dx_d$$

- Количество параметров:  $d + 1$



# Много признаков

- Общий случай:  $d$  признаков
- Модель

$$a(x) = w_0 + w_1x_1 + \dots + w_dx_d$$

Свободный коэффициент/сдвиг/bias

Веса/коэффициенты

- Количество параметров:  $d + 1$

# Много признаков

$$\begin{aligned} a(x) &= w_0 + w_1 x_1 + \dots + w_d x_d = \\ &= w_0 + \langle w, x \rangle \end{aligned}$$

- Будем считать, что есть признак, всегда равный единице:

$$\begin{aligned} a(x) &= w_1 x_1 + \dots + w_d x_d = \\ &= w_1 * 1 + w_2 x_2 + \dots + w_d x_d = \\ &= \langle w, x \rangle \end{aligned}$$

# Применимость линейной регрессии

# Модель линейной регрессии

$$a(x) = w_1x_1 + \dots + w_dx_d = \langle w, x \rangle$$

- Нет гарантий, что целевая переменная именно так зависит от признаков
- Надо формировать признаки так, чтобы модель подходила

# Предсказание стоимости квартиры

- Признаки: площадь, район, расстояние до метро
- Целевая переменная: рыночная стоимость квартиры
- Линейная модель:

$$\begin{aligned} a(x) = & w_0 + w_1 * (\text{площадь}) \\ & + w_2 * (\text{район}) \\ & + w_3 * (\text{расстояние до метро}) \end{aligned}$$

# Предсказание стоимости квартиры

$$\begin{aligned} a(x) = & w_0 + w_1 * (\text{площадь}) \\ & + w_2 * (\text{район}) \\ & + w_3 * (\text{расстояние до метро}) \end{aligned}$$

# Предсказание стоимости квартиры

$$\begin{aligned} a(x) = & w_0 + w_1 * (\text{площадь}) \\ & + w_2 * (\text{район}) \\ & + w_3 * (\text{расстояние до метро}) \end{aligned}$$

- За каждый квадратный метр добавляем  $w_1$  к прогнозу

# Предсказание стоимости квартиры

$$\begin{aligned} a(x) = & w_0 + w_1 * (\text{площадь}) \\ & + w_2 * (\text{район}) \\ & + w_3 * (\text{расстояние до метро}) \end{aligned}$$

- Что-то странное

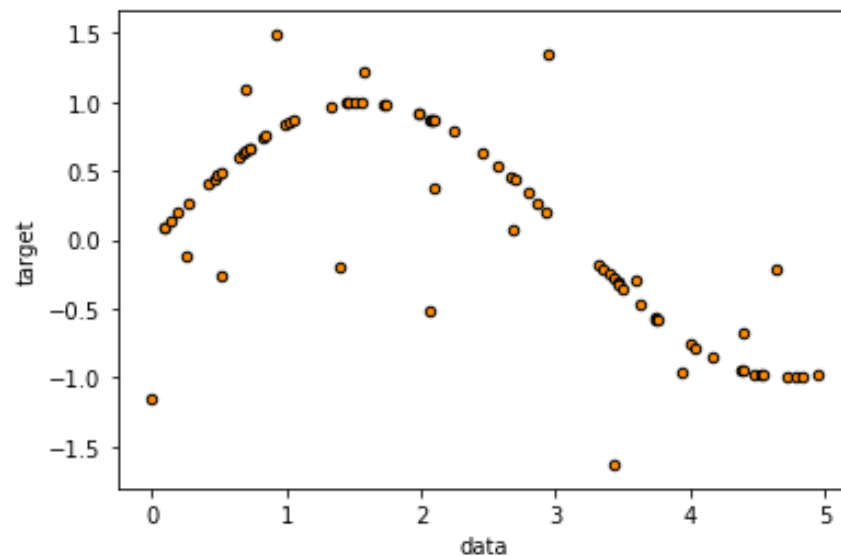


# Предсказание стоимости квартиры

$$a(x) = w_0 + w_1 * (\text{площадь})$$

$$+ w_2 * (\text{район})$$

$$+ w_3 * (\text{расстояние до метро})$$




# Кодирование категориальных признаков

- Значения признака «район»:  $U = \{u_1, \dots, u_m\}$
- Новые признаки вместо  $x_j$ :  $[x_j = u_1], \dots, [x_j = u_m]$
- One-hot кодирование

# Кодирование категориальных признаков

Район	ЦАО	ЮАО	САО
ЦАО	1	0	0
ЮАО	0	1	0
ЦАО	1	0	0
САО	0	0	1
ЮАО	0	1	0

# Кодирование категориальных признаков

Район		ЦАО	ЮАО	САО
ЦАО		1	0	0
ЮАО		0	1	0
ЦАО		1	0	0
САО		0	0	1
ЮАО		0	1	0

$$a(x) = w_0 + w_1 * (\text{площадь})$$

$$+ w_2 * (\text{квартира в ЦАО?})$$

$$+ w_3 * (\text{квартира в ЮАО?})$$

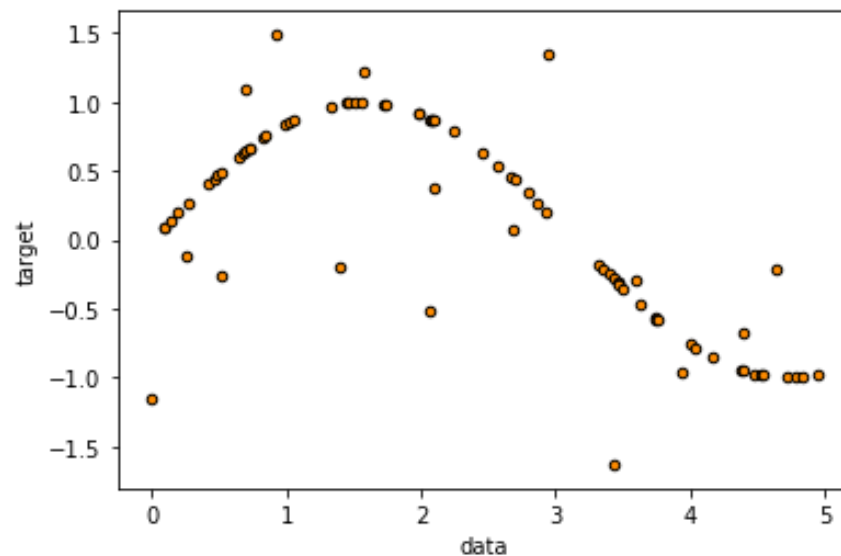
$$+ w_4 * (\text{квартира в САО?})$$

# Предсказание стоимости квартиры

$$a(x) = w_0 + w_1 * (\text{площадь})$$

$$+ w_2 * (\text{район})$$

$$+ w_3 * (\text{расстояние до метро})$$

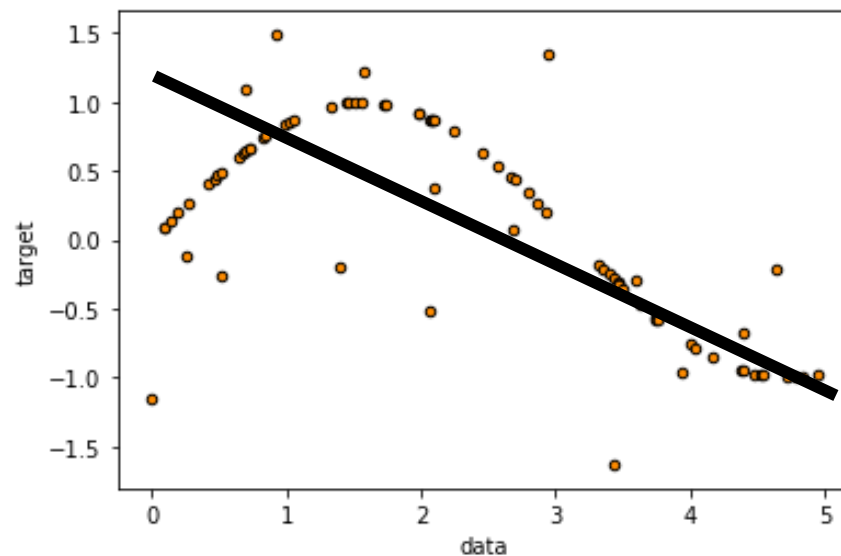


# Предсказание стоимости квартиры

$$a(x) = w_0 + w_1 * (\text{площадь})$$

$$+ w_2 * (\text{район})$$

$$+ w_3 * (\text{расстояние до метро})$$

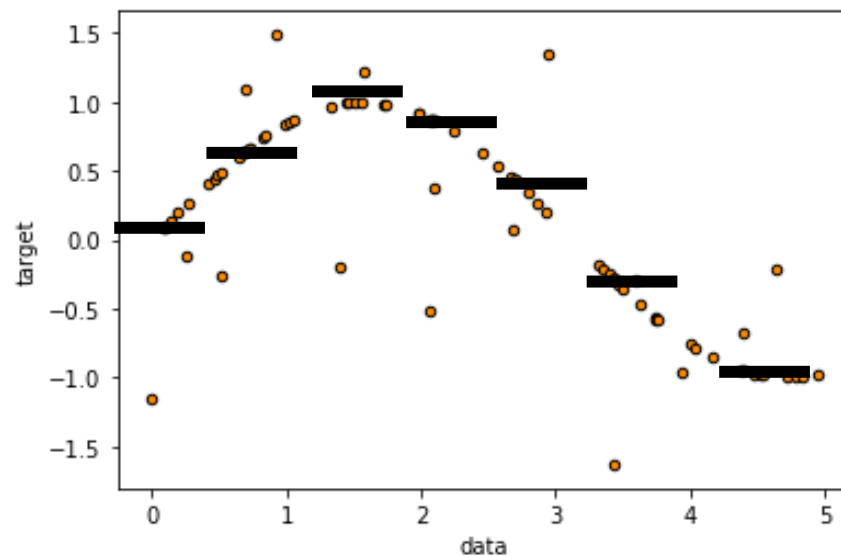


# Предсказание стоимости квартиры

$$a(x) = w_0 + w_1 * (\text{площадь})$$

$$+ w_2 * (\text{район})$$

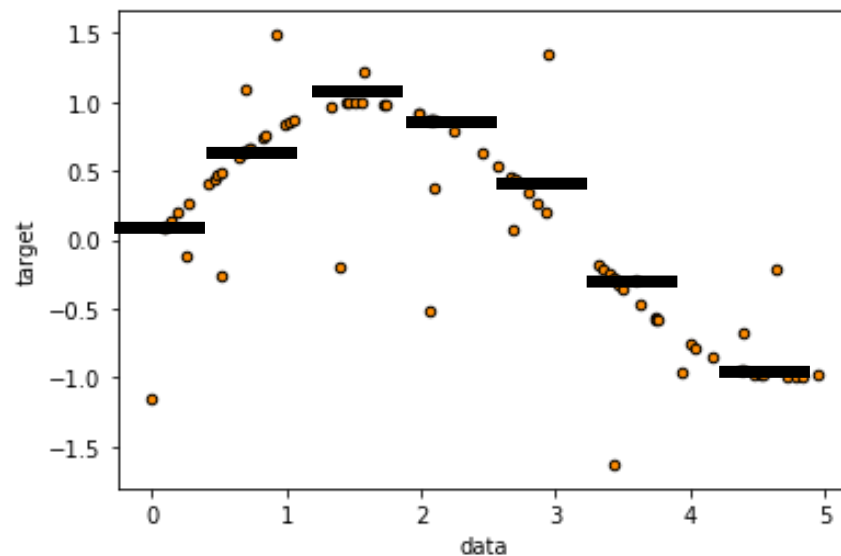
$$+ w_3 * (\text{расстояние до метро})$$



# Предсказание стоимости квартиры

$$a(x) = w_0 + w_1 * (\text{площадь}) \\ + w_2 * (\text{район})$$

$$+ w_3 * [t_0 \leq x_3 < t_1] + \dots + w_{3+n} [t_{n-1} \leq x_3 < t_n]$$





# Линейные модели

- Модель линейной регрессии хороша, если признаки сделаны специально под неё
- Пример: one-hot кодирование категориальных признаков или бинаризация числовых признаков

# Линейная регрессия в векторном виде

# Модель линейной регрессии

$$a(x) = \langle w, x \rangle$$

- Среднеквадратичная ошибка и задача обучения:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} (\langle w, x_i \rangle - y_i)^2 \rightarrow \min_w$$

# Матрицы

- Матрица — таблица с числами (для простоты)
- Матрица «объекты-признаки»:

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\ell 1} & x_{\ell 2} & \cdots & x_{\ell d} \end{pmatrix} \in \mathbb{R}^{\ell \times d}$$

# Матрицы

- Матрица — таблица с числами (для простоты)
- Матрица «объекты-признаки»:

объект и его признаки

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\ell 1} & x_{\ell 2} & \cdots & x_{\ell d} \end{pmatrix}$$

# Матрицы

- Матрица — таблица с числами (для простоты)
- Матрица «объекты-признаки»:

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\ell 1} & x_{\ell 2} & \cdots & x_{\ell d} \end{pmatrix}$$

значения признака на всех объектах

# Векторы

- Вектор размера  $d$  — тоже матрица
- Вектор-строка:  $w = (w_1, \dots, w_d) \in \mathbb{R}^{1 \times d}$
- Вектор-столбец:  $w = \begin{pmatrix} w_1 \\ \dots \\ w_d \end{pmatrix} \in \mathbb{R}^{d \times 1}$

# Применение линейной модели

- $a(x) = \langle w, x \rangle = w_1 x_1 + \dots + w_d x_d$
- Как применить модель к обучающей выборке?

$$\begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{\ell 1} & x_{\ell 2} & \cdots & x_{\ell d} \end{pmatrix}$$

$$\begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{pmatrix}$$

$$\begin{pmatrix} \sum_{i=1}^d w_i x_{1i} \\ \sum_{i=1}^d w_i x_{2i} \\ \vdots \\ \sum_{i=1}^d w_i x_{\ell i} \end{pmatrix}$$



# Матричное умножение

- Только для матриц  $A \in \mathbb{R}^{m \times k}$  и  $B \in \mathbb{R}^{k \times n}$
- Результат:  $AB = C \in \mathbb{R}^{m \times n}$
- Правило:

$$c_{ij} = \sum_{p=1}^k a_{ip} b_{pj}$$

Пример

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 10 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} & & \\ & & \\ & & \end{pmatrix}$$

Пример

$$\begin{pmatrix} \boxed{1} & \boxed{2} \\ 0 & 1 \\ 10 & 0 \end{pmatrix} * \begin{pmatrix} \boxed{1} & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} \boxed{1} & & \\ & & \\ & & \end{pmatrix}$$

Пример

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 10 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \end{pmatrix}$$

Пример

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 10 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 4 \\ & & \end{pmatrix}$$

Пример

$$\begin{pmatrix} 1 & 2 \\ 0 & 1 \\ 10 & 0 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 4 \\ 0 & & \end{pmatrix}$$

# Модель линейной регрессии

- Среднеквадратичная ошибка и задача обучения:

$$\frac{1}{\ell} \sum_{i=1}^{\ell} (\langle w, x_i \rangle - y_i)^2 \rightarrow \min_w$$

# Вычисление ошибки

- Отклонения прогнозов от ответов:

$$Xw - y = \begin{pmatrix} \langle w, x_1 \rangle - y_1 \\ \vdots \\ \langle w, x_\ell \rangle - y_\ell \end{pmatrix}$$



# Вычисление ошибки

- Евклидова норма:

$$\|z\| = \sqrt{\sum_{j=1}^n z_j^2}$$

$$\|z\|^2 = \sum_{j=1}^n z_j^2$$

# Вычисление ошибки

- Отклонения прогнозов от ответов:

$$Xw - y = \begin{pmatrix} \langle w, x_1 \rangle - y_1 \\ \vdots \\ \langle w, x_\ell \rangle - y_\ell \end{pmatrix}$$

- Среднеквадратичная ошибка:

$$\frac{1}{\ell} \|Xw - y\|^2 = \frac{1}{\ell} \sum_{i=1}^{\ell} (\langle w, x_i \rangle - y_i)^2$$

# Обучение линейной регрессии

$$\frac{1}{\ell} \|Xw - y\|^2 \rightarrow \min_w$$

- Вычисление MSE в NumPy:

```
np.square(X.dot(w) - y).mean()
```