Машинное обучение
Метод К-ближайших соседей
(KNN - K Nearest Neighbors)

Д.Ю. Хартьян

Метод К-ближайших соседей

- KNN (k-nearest neighbors) – один из простейших алгоритмов машинного обучения.
- Обзор раздела:
  - Теория KNN
  - Пример кода для KNN
  - Упражнения по KNN
  - Решения упражнений по KNN

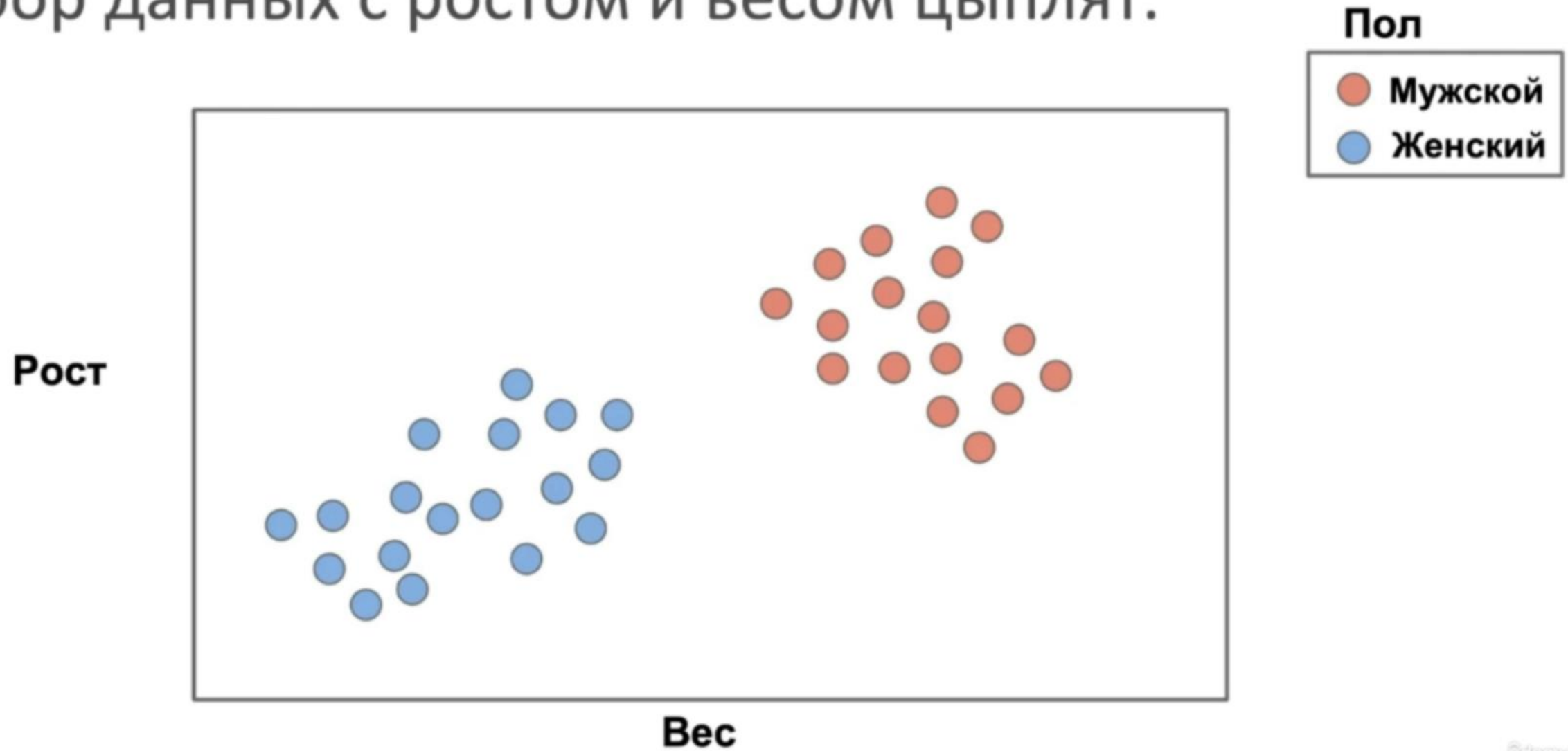
Метод К-ближайших соседей

- KNN (k-nearest neighbors) – один из простейших алгоритмов машинного обучения.
- Для каждой точки этот алгоритм присваивает значение, основываясь на **расстоянии** между старыми данными и новыми данными
- Посмотрим на примере...

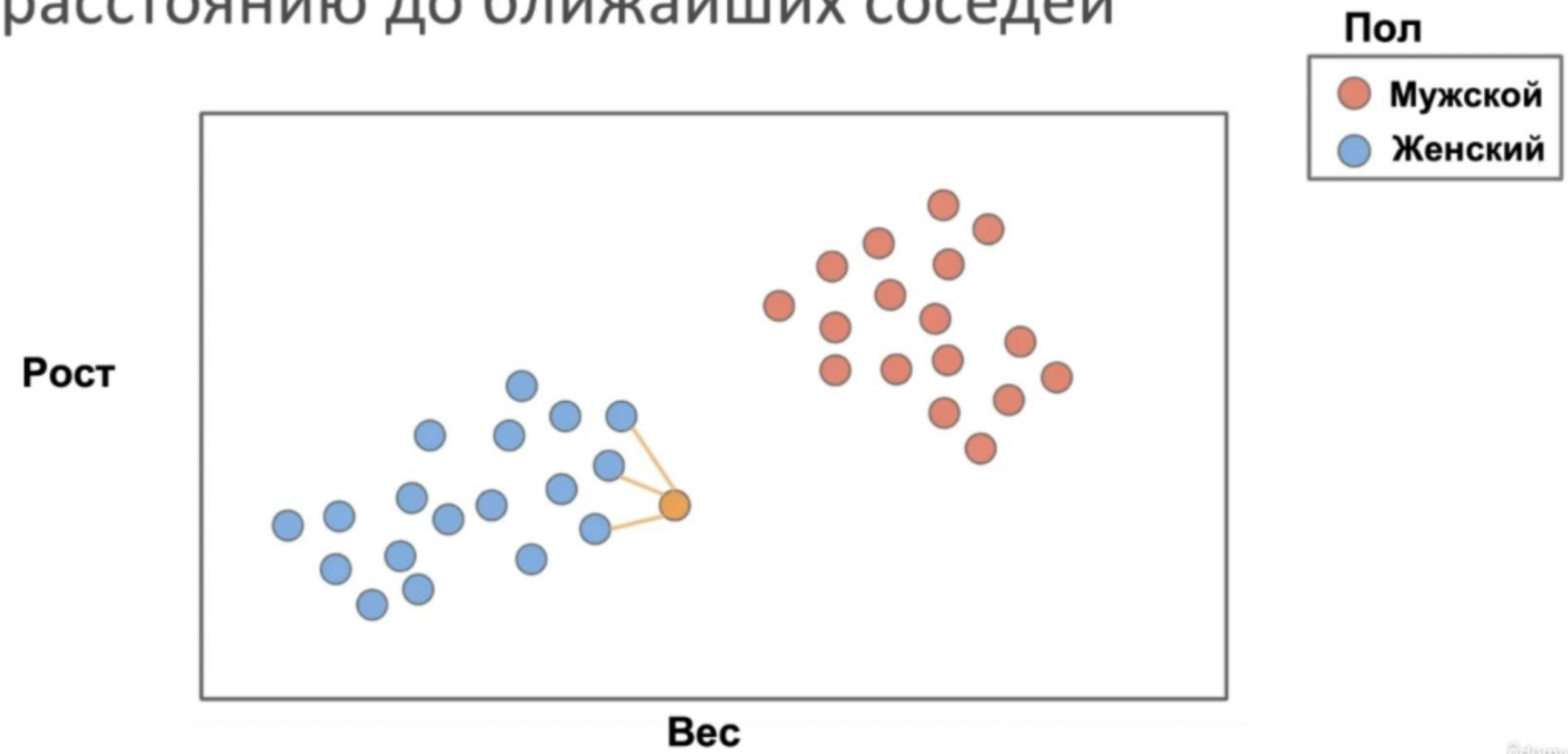
Метод К-ближайших соседей. П ример

- В птицеводстве есть задача определять пол цыплят.
- Предположим, что у нас есть набор данных с ростом и весом цыплят.
- Можем ли мы написать алгоритм, определяющий пол цыплёнка на основе его роста и веса?

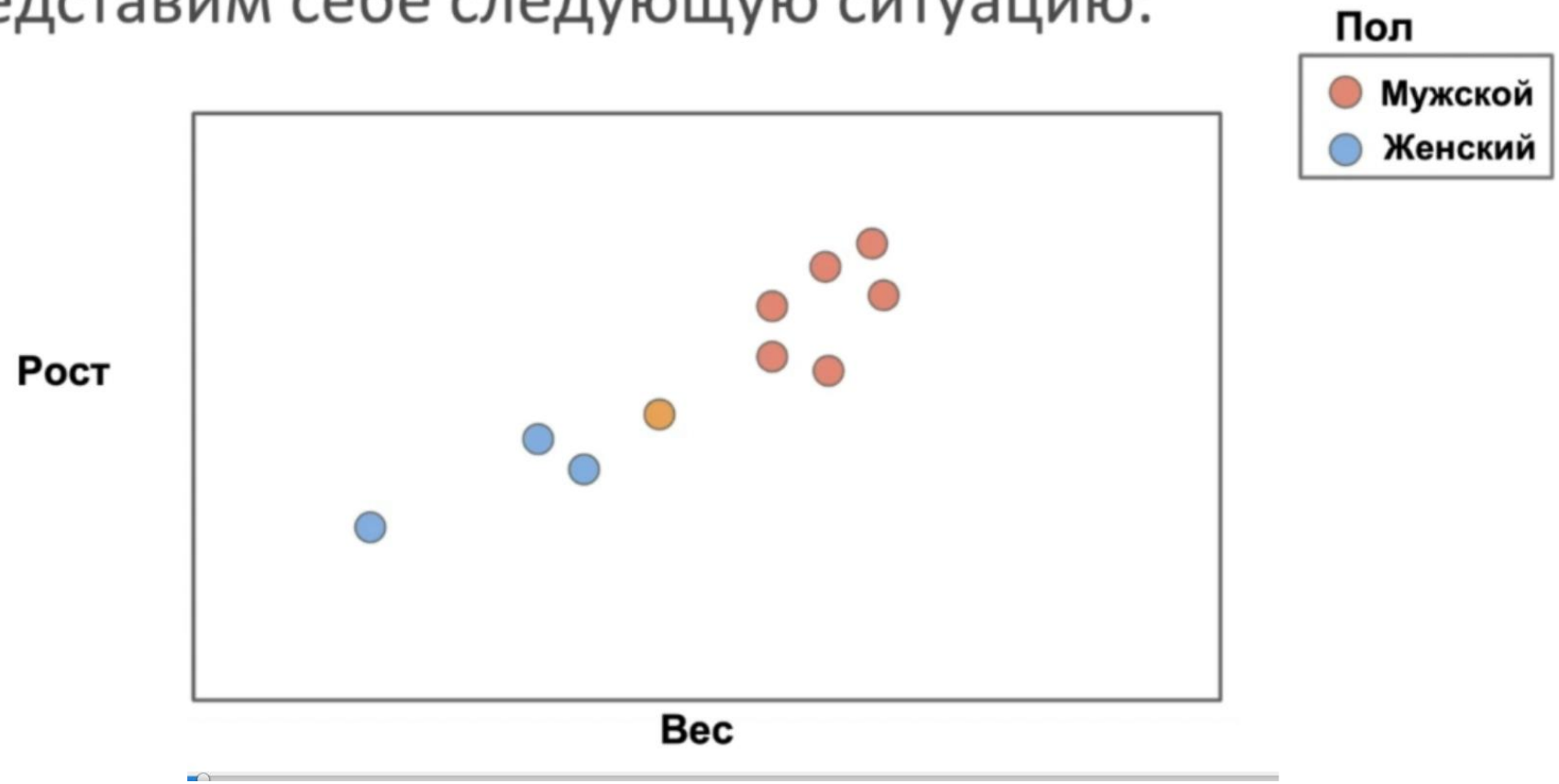
- Набор данных с ростом и весом цыплят.



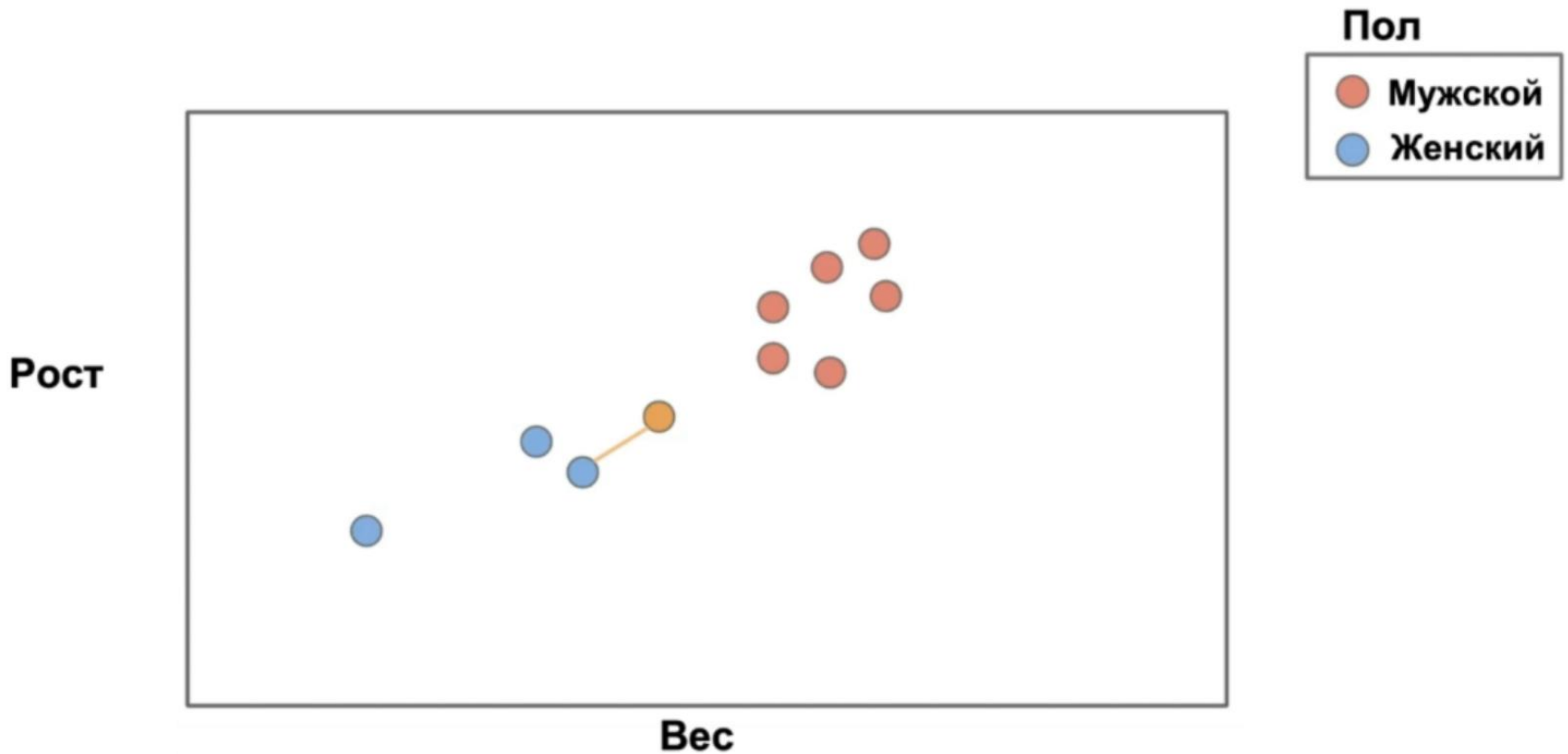
- По расстоянию до ближайших соседей



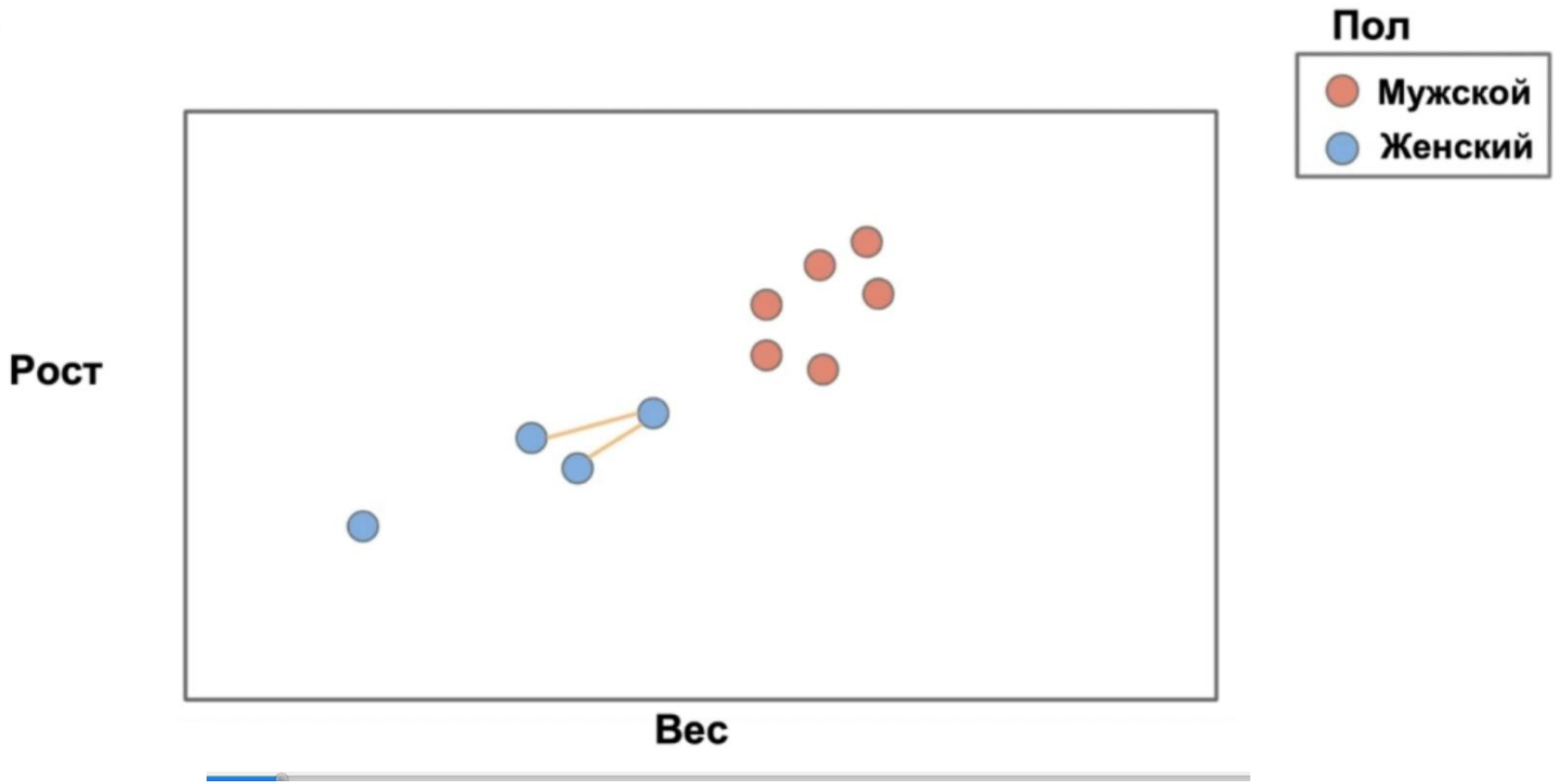
- Представим себе следующую ситуацию:



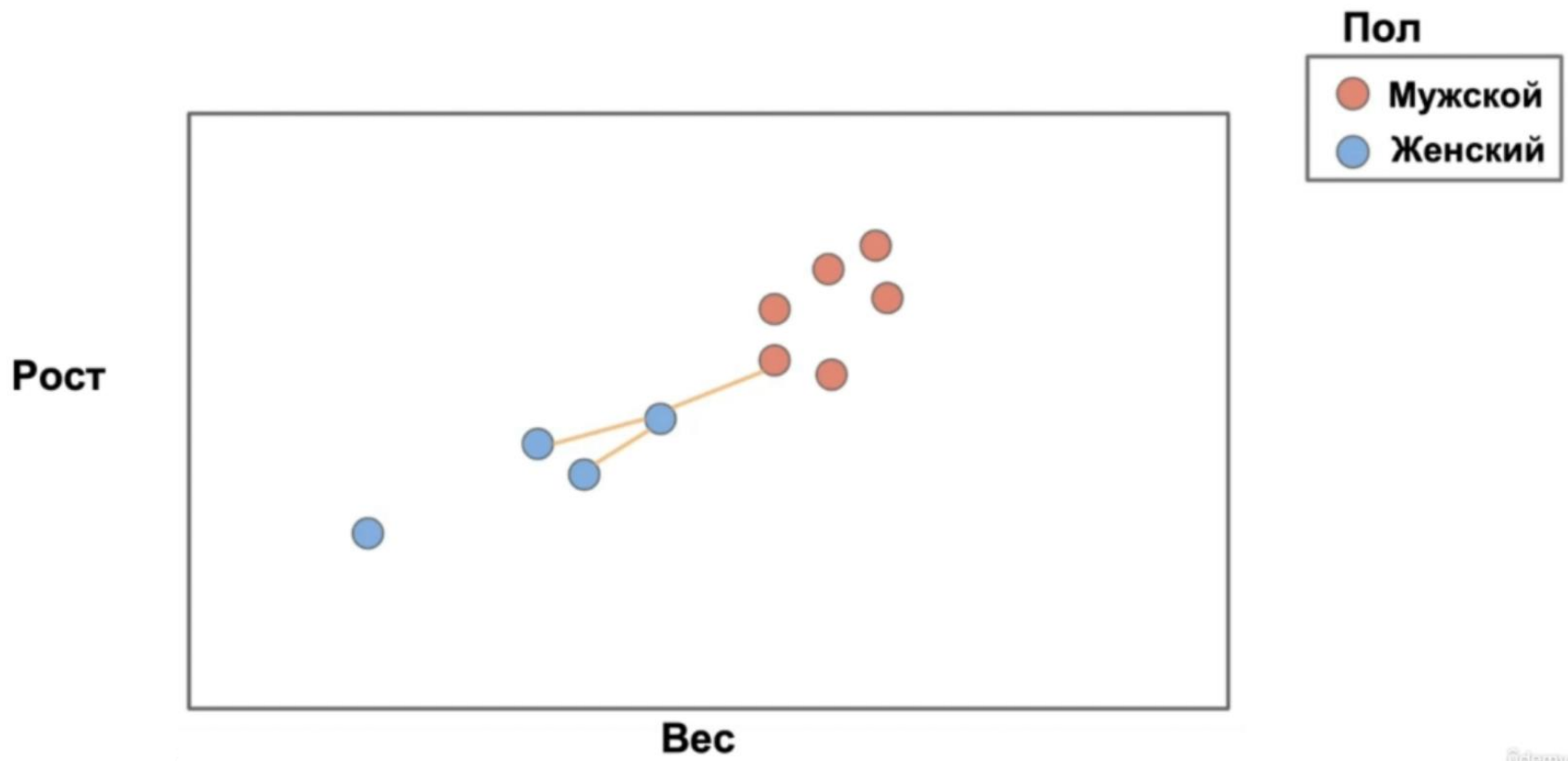
- K=1



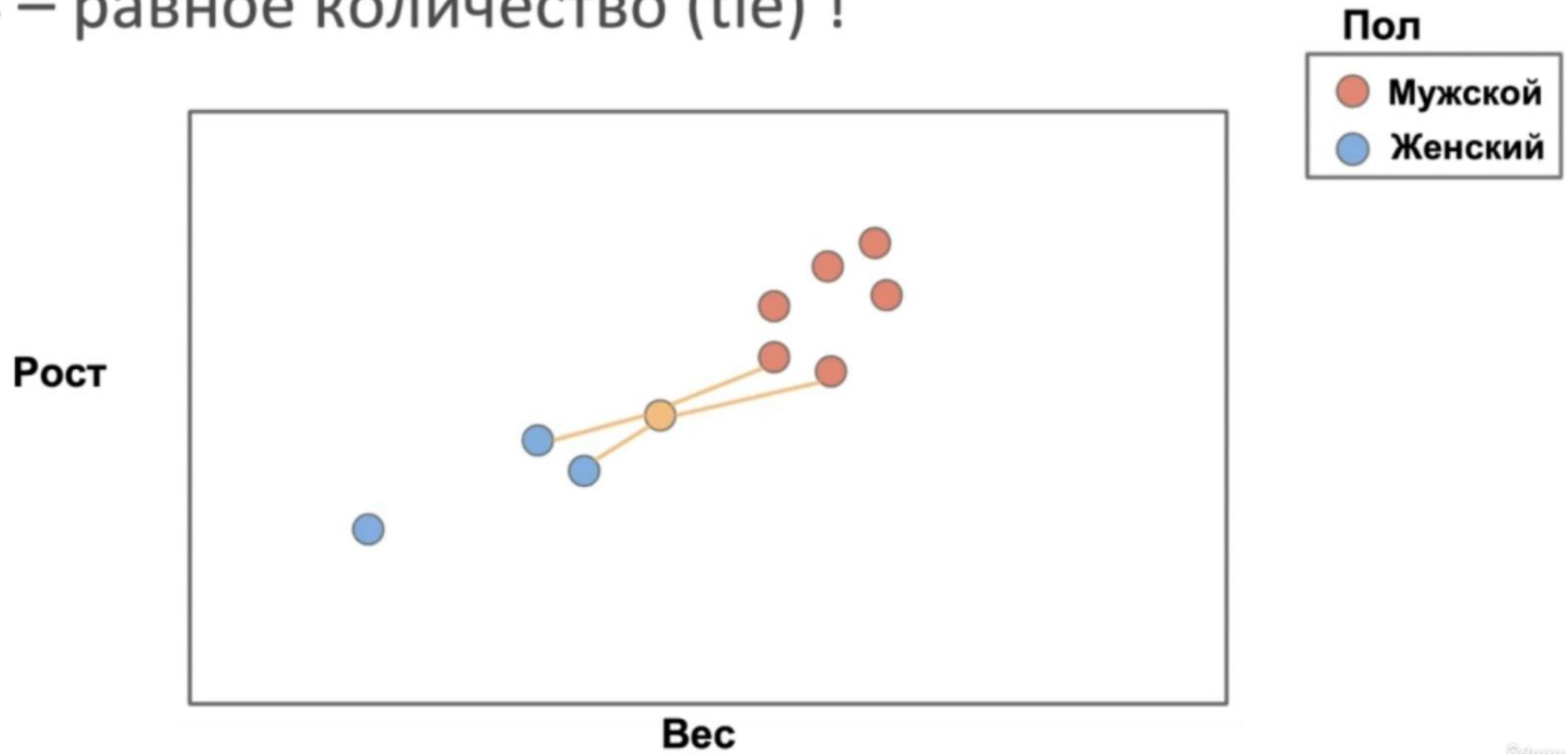
- K=2



- K=3



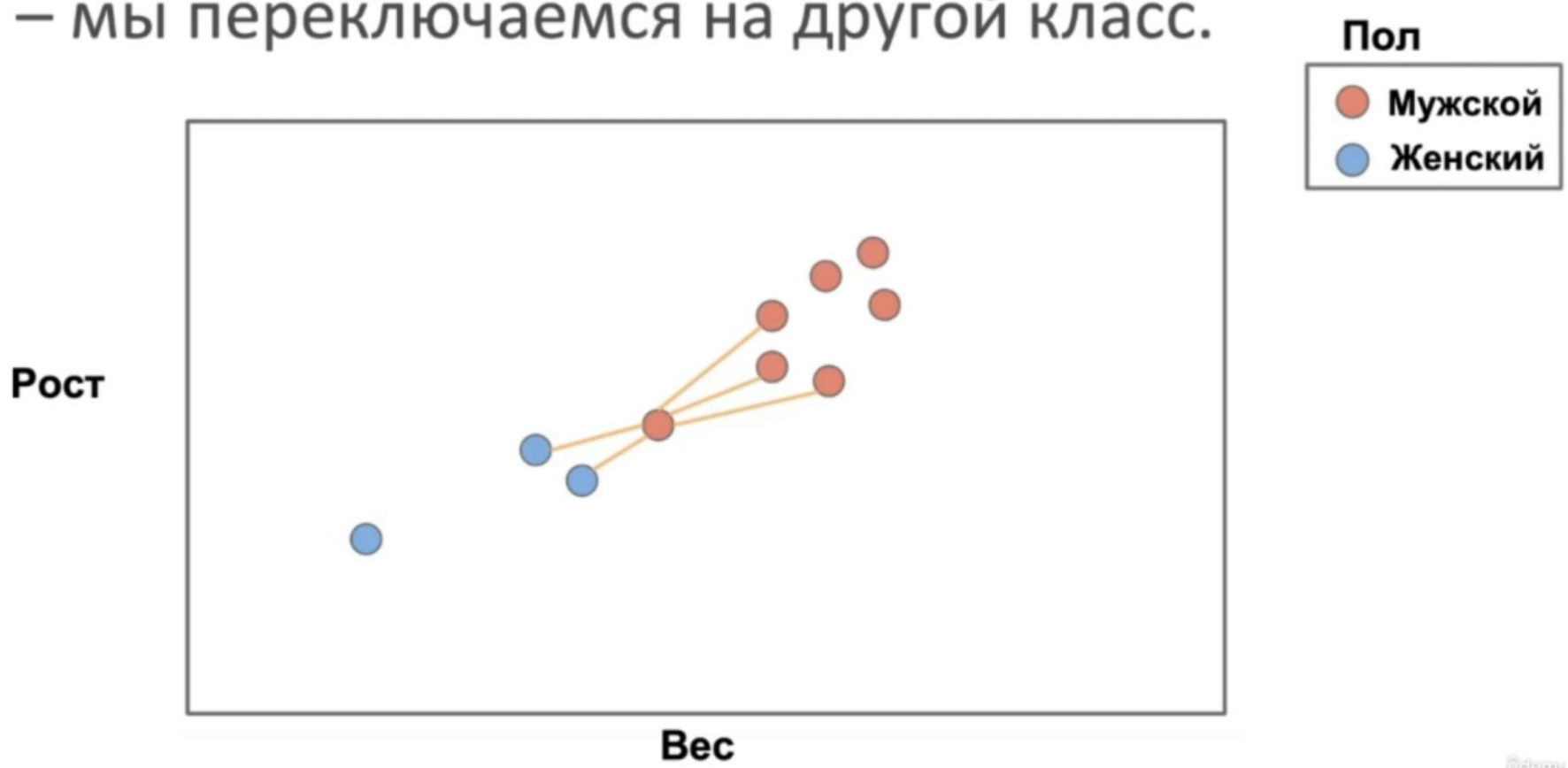
- K=4 – равное количество (tie) !



- Варианты:
  - Всегда выбирать нечётные значения К
  - В случае равенства, уменьшить К на 1
  - Случайно выбрать тот или иной вариант
  - Выбрать ближайшую точку

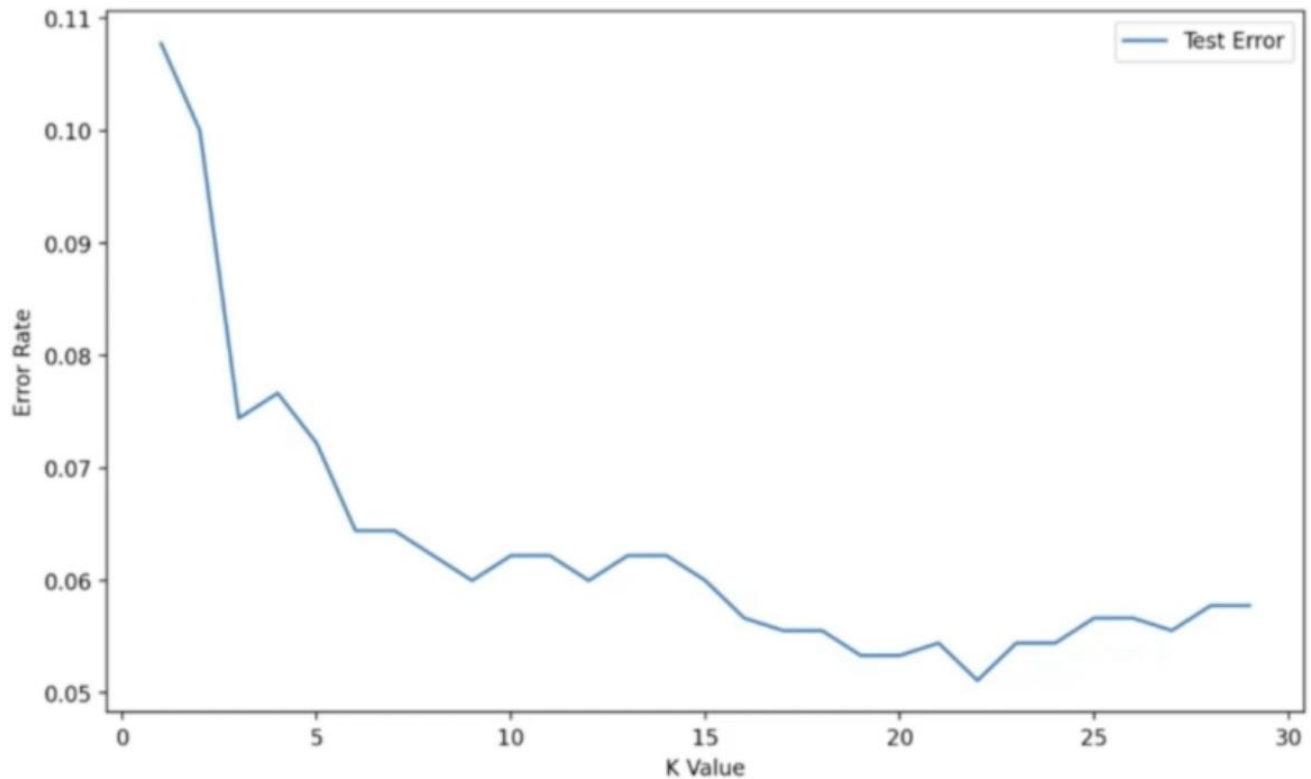
- Как с такими ситуациями работает Scikit-Learn?
  - В случае равенства (ties) будет выбран класс, идущий первым в множестве соседей.
  - Результаты отсортированы по расстоянию, так что будет выбран класс ближайшей точки.

- K=5 – мы переключаемся на другой класс.

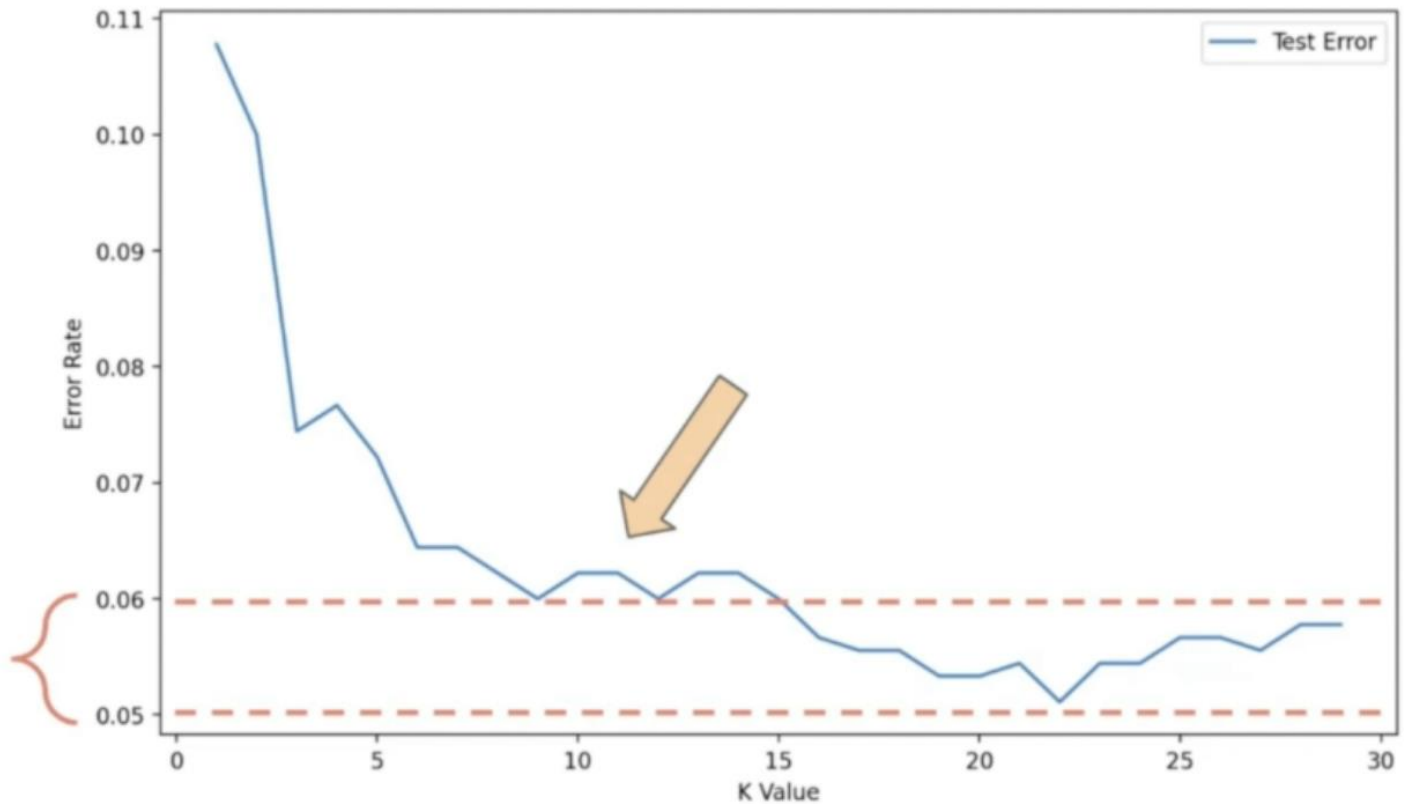


- Как выбрать число K?
  - Хотим минимизировать error = 1 - accuracy
- Два метода:
  - Метод локтя (elbow method)
  - Кросс-валидация - перебор различных значений K по сетке (grid search), чтобы найти значение K с наименьшими ошибками

- Метод локтя (elbow method)



- Метод локтя (elbow method)



- Кросс-валидация ищет значение K, ориентируясь только на минимизацию ошибок
- Это может привести к более сложной модели (более высокому значению K)
- Учитывайте специфику задачи, чтобы понять, насколько приемлемо увеличение значения K

- Алгоритм KNN
  - Выбираем значение К
  - Сортируем вектора признаков (в N-мерном пространстве) по метрике расстояния
  - Выбираем класс точек на основе К ближайших векторов признаков

Метрики расстояния

Расстояние

$$a, b \in \mathbb{R}^m$$

$L^2 = d_2(a,b) = \{\sum_{i=1}^{m} |a_i - b_i|^2\}^{1/2}$ Евклидово расстояние

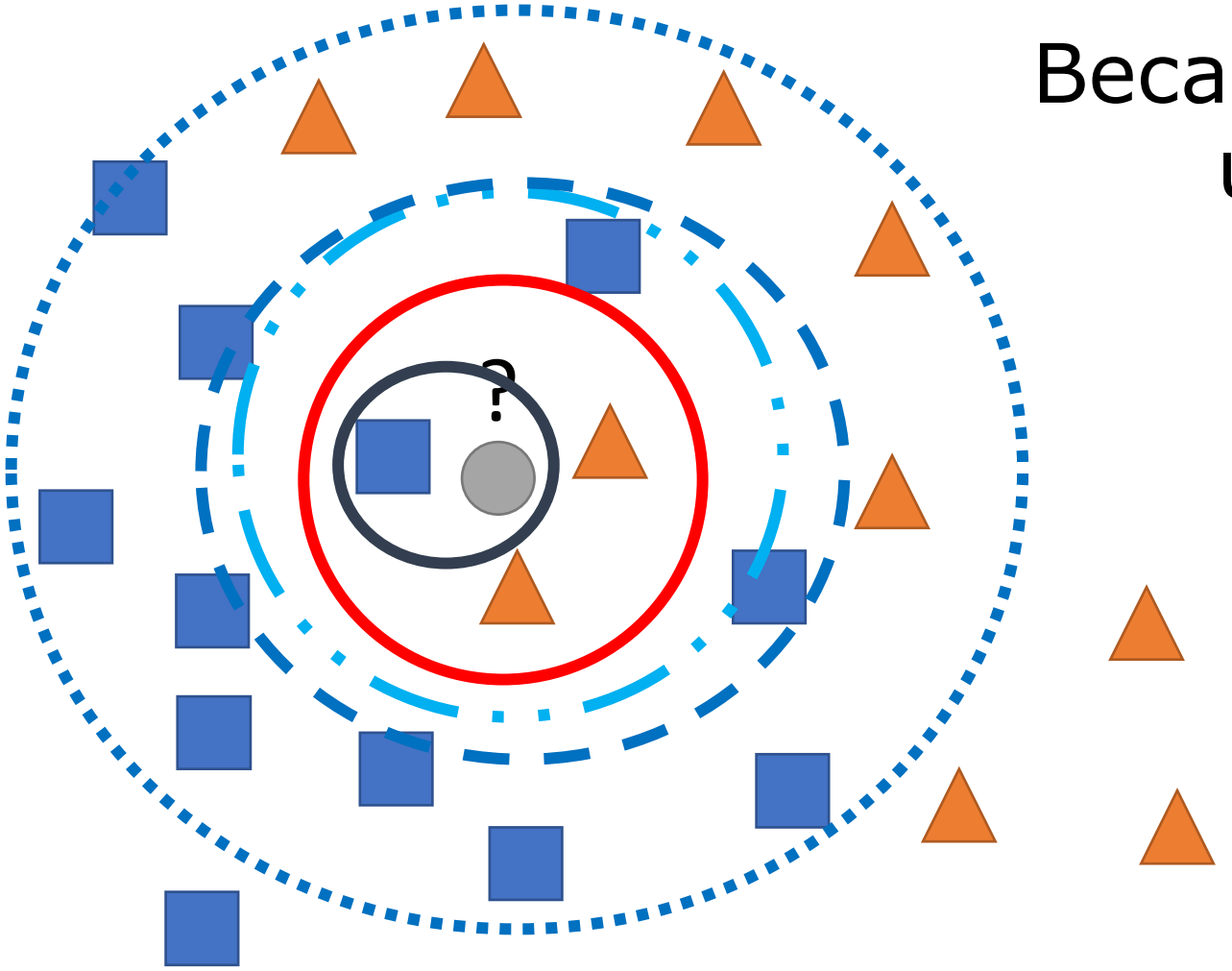
$L^1 = d_1(a,b) = \sum_{i=1}^{m} |a_i - b_i|$ Манхэттенское расстояние

$L^\infty = d_\infty(a,b) = \max(|a_i - b_i|)$ Расстояние Чебышева

$L^p = d_p(a,b) = \{\sum_{i=1}^{m} |a_i - b_i|^p\}^{1/p}$ Расстояние Минковского

Классификация к-Ближайших Соседей .

Выбор числа k



Веса – однородные
uniform

Классификация к-Ближайших Соседей

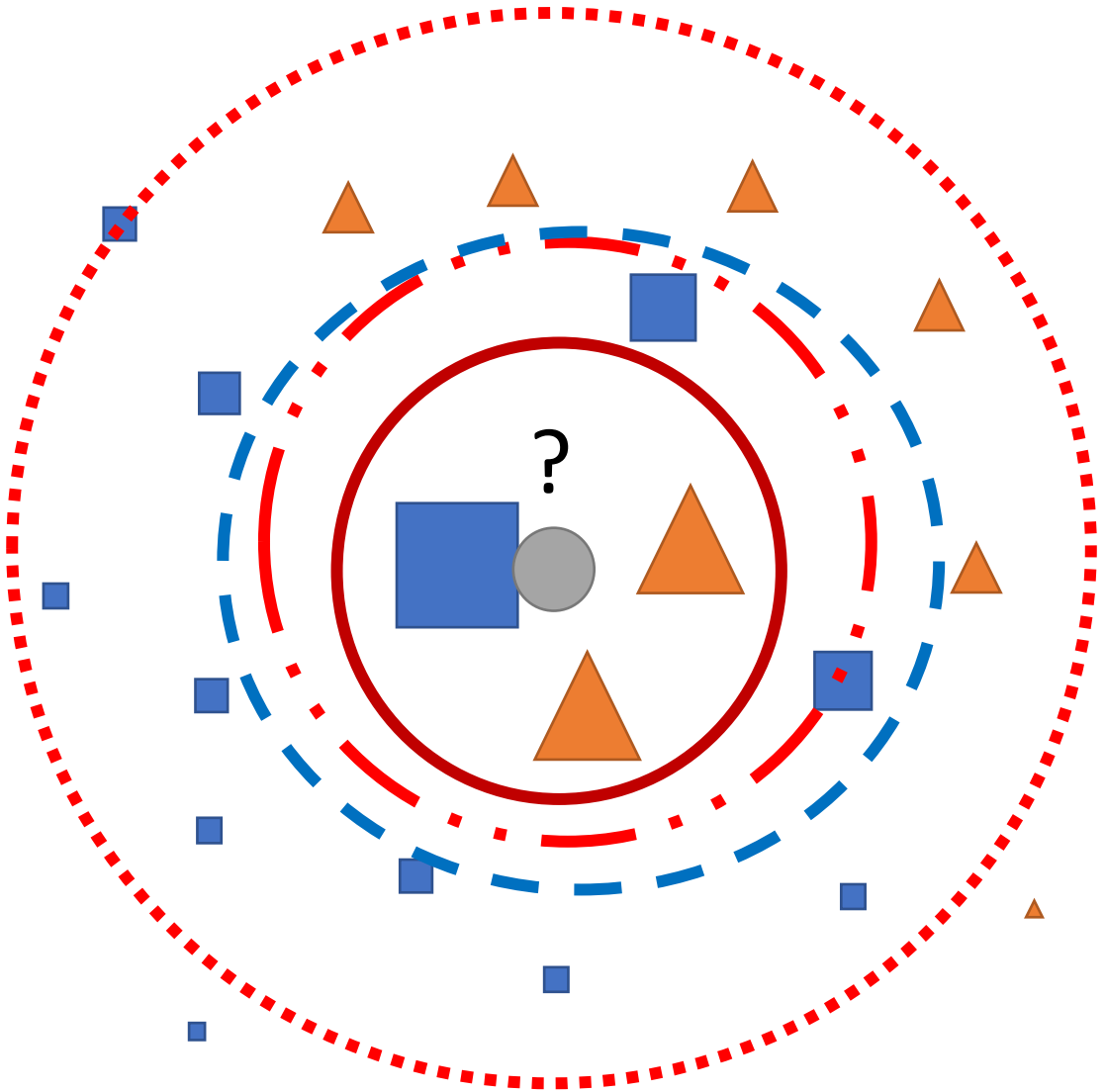
Веса

чем дальше пример, тем меньше вклад учитывается его «голосом»

$$w_i \sim \frac{1}{d(x,i)}$$

Разный вес для разных классов

$$w_{blue} = a \qquad w_{red} = b$$



Преимущества и недостатки KNN:

- **Преимущества**:
  - Прост в понимании и реализации.
  - Нетребователен к предварительным предположениям о распределении данных.
  - Модель обновляется немедленно при добавлении новых данных.
- **Недостатки**:
  - Неэффективен для больших наборов данных.
  - Чувствителен к несбалансированным данным и шуму.
  - Требует масштабирования признаков.