DOI 10.58351/2949-2041.2025.20.3.008

Сироткин Максим Сергеевич,
Крюкова Диана Вадимовна,
Чернявский Роман Евгеньевич, Студенты,
Сахалинский государственный университет, Южно-Сахалинск

Научный руководитель: **Осипов Геннадий Сергеевич,**
д.т.н., профессор кафедры информатики,
Сахалинский государственный университет, Южно-Сахалинск

ИССЛЕДОВАНИЕ МЕТОДА БЛИЖАЙШИХ СОСЕДЕЙ В СИСТЕМЕ СИМВОЛЬНОЙ МАТЕМАТИКИ WOLFRAM MATHEMATICA

Аннотация. В статье исследуется метод ближайших соседей (k-NN) в системе символьной математики Wolfram Mathematica. Рассматриваются его возможности в задачах классификации и регрессии, проведена практическая апробация на наборах данных Fisher's Iris и Boston Homes. Представлены графические результаты, анализ влияния гиперпараметров, выявлены достоинства и недостатки метода. Отмечена его универсальность и вычислительные ограничения при работе с большими объемами данных.

Ключевые слова: машинное обучение, метод ближайших соседей, регрессия, машинное обучение, анализ данных.

Основные понятия

Метод ближайших соседей (k-Nearest Neighbors, k-NN) представляет собой один из простейших и наиболее интуитивно понятных методов машинного обучения. Он относится к классу непараметрических алгоритмов, что означает отсутствие явного этапа построения модели перед предсказанием. Вместо этого алгоритм делает выводы на основе всей обучающей выборки [1].

Главное преимущество этого метода – его универсальность. Он может быть применен как для задач классификации, так и для задач регрессии, а также адаптирован для работы с различными типами данных. Основным требованием является возможность определения меры сходства (или расстояния) между объектами, что позволяет оценивать их близость в заданном пространстве признаков [2].

Главная идея метода заключается в том, что объекты, находящиеся близко друг к другу в пространстве признаков, скорее всего, принадлежат к одному классу (в случае классификации) или имеют схожие значения целевой переменной (в случае регрессии). Таким образом, для предсказания нового объекта достаточно найти его ближайших соседей и определить класс или значение по ним.

Метод ближайших соседей не требует сложных вычислений в процессе обучения, так как отсутствует явная фаза генерации модели. Однако его слабым местом является высокая вычислительная сложность на этапе предсказания, так как каждый новый объект сравнивается со всей обучающей выборкой [1].

Среди ключевых характеристик метода можно выделить:

- Отсутствие предварительного обучения: алгоритм работает по принципу «ленивого обучения» (lazy learning).
- Зависимость от метрики расстояния: правильный выбор функции расстояния существенно влияет на качество классификации или регрессии.
- Чувствительность к размеру окрестности: число соседей или радиус окрестности – важные гиперпараметры, которые определяют степень сглаживания модели.

Этот метод часто используется как базовый инструмент при анализе данных и может служить эталоном для сравнения с более сложными моделями. Метод широко применяется в различных областях, включая компьютерное зрение, биометрию, медицину и анализ данных.

Несмотря на свою простоту, NN часто оказывается достаточно эффективным, особенно в условиях небольшого количества данных или в задачах, где выбор оптимального расстояния хорошо интерпретируем.

Описание метода.

Рассмотрим процесс классификации с использованием метода ближайших соседей на простом примере. Пусть имеется набор данных (см. рисунок 1), содержащий два числовых признака: возраст и вес, а также два возможных класса:

- cat (жёлтый треугольник –)
- dog (синий круг –)

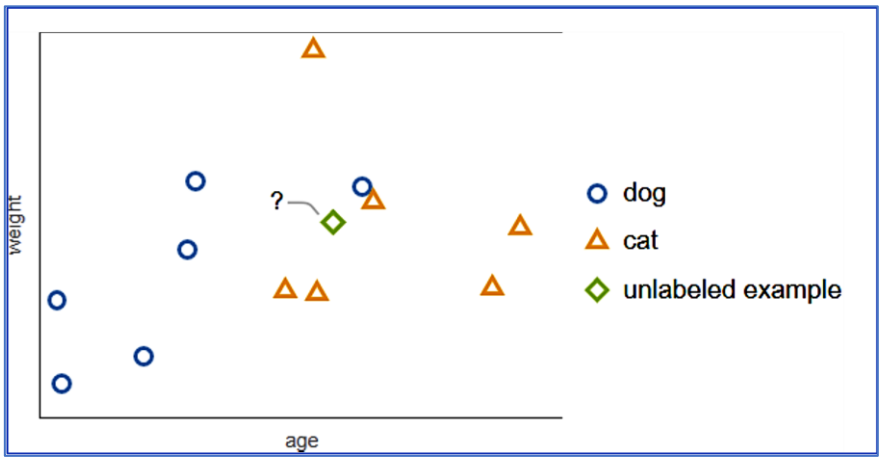

Рис. 1 Исследуемый набор данных

Допустим, что имеется новый объект (зелёный ромб –), для которого класс неизвестен. Метод ближайших соседей выполняет следующие шаги:

1. Определение ближайших соседей: вычисляется расстояние между новым объектом и всеми имеющимися примерами.

2. Голосование среди соседей: выбирается наиболее часто встречающийся класс среди k ближайших соседей.

Если, например (как показано на рисунке 2), среди четырёх ближайших соседей три принадлежат классу cat, а один – классу dog, то новому объекту присваивается класс cat.

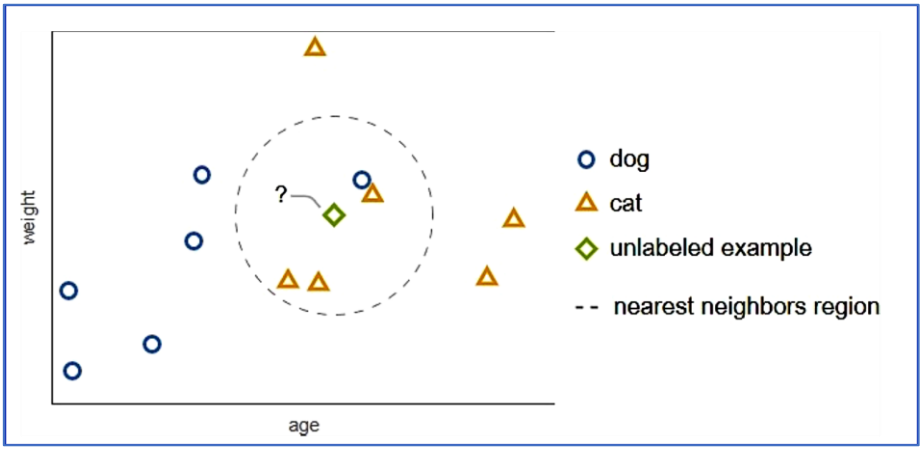

Рис. 2 Пример присвоения класса

Этот же принцип можно адаптировать для решения задачи регрессии. Если вместо классов объекты имеют числовые значения (например, скорость движения животных), прогноз может быть рассчитан как среднее значение ближайших соседей (см. рисунок 3):

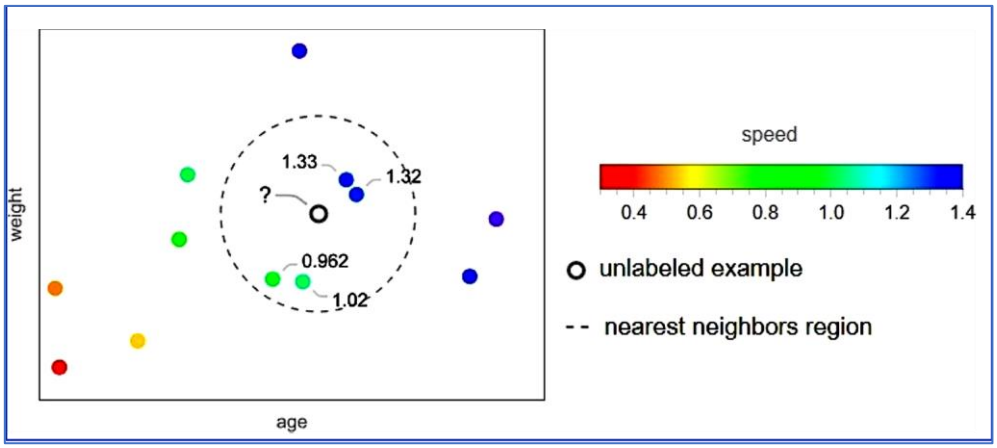

Рис. 3 Пример регрессии

$$Mean\{0.962, 1.02, 1.32, 1.33\} = 1.158$$

(Важно отметить, что в данном методе нет традиционного этапа обучения, что отличает его от параметрических моделей.)

На практике используются различные модификации метода:

- k-NN (k-Nearest Neighbors): соседями объекта считаются k ближайших точек, а не все точки в радиусе.
- Взвешенный k-NN: соседям присваиваются веса в зависимости от их удалённости от целевого объекта. Например, может использоваться экспоненциальная функция убывания:

$$weight = exp\left(-\frac{distance}{\lambda}\right)$$

- Различные функции расстояния: например, евклидово расстояние для числовых данных:

$$EuclideanDistance(\{u_1, u_2, u_3\}, \{u_1, u_2, u_3\}) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + (u_3 - v_3)^2}$$

или расстояние редактирования (Edit Distance) для строковых данных:

$$EditDistance(abcd, acd) = 1$$

Таким образом, метод ближайших соседей является гибким инструментом, который можно адаптировать под конкретные задачи, выбирая подходящие расстояния, стратегии взвешивания и количество соседей.

Обучающие выборки

1. Классический набор данных Fisher's Irises [3].

На рисунке 4 представлены операторы предварительной обработки исходных данных в системе символьной математики Wolfram Mathematica [4].

```
(*Загружаем классический набор данных Fisher's Irises*)
irises = ResourceData["Sample Data: Fisher's Irises"];
         |данные ресурса

(*Удаляем ненужные столбцы с длиной и шириной лепестков*)
irises = KeyDrop[irises, {"PetalLength", "PetalWidth"}];
         |отбросить ключи

(*Добавляем случайное отклонение к длине и ширине чашелистиков*)
irises = MapAt[# + Quantity[RandomReal[{-0.05, 0.05}], "Centimeters"] &,
         |преобраз··· |размерная·· |случайное действительное число
   irises, {All, 2 ;; 3}];
           |всё

(*Преобразуем величины к числовым значениям без единиц измерения*)
irises = QuantityMagnitude[irises];
         |модуль размерной величины

(*Отображаем первые 3 строки обработанного набора данных*)
Dataset[irises, MaxItems → 3]
|массив данных  |максимальное число элементов
```

Species	SepalLength	SepalWidth
setosa	5.08256	3.46032
setosa	4.94226	2.96009
setosa	4.65522	3.21453

rows 1–3 of 150

Рис. 4 Предварительная обработка исходных данных

Этот набор данных — один из самых известных в машинном обучении. Он содержит три вида ирисов, каждый из которых описан четырьмя числовыми признаками: длина и ширина лепестков и чашелистиков. Метод ближайших соседей здесь используется для классификации видов растений.

Главная сложность при работе с этим датасетом заключается в выборе оптимального количества соседей. Если слишком мало, модель может переобучиться и быть чувствительной к шуму в данных. Если слишком велико, модель становится слишком сглаженной и теряет способность различать границы классов.

Результирующий набор данных представлен на рисунке 5:

```
(*Создаём график рассеяния для визуализации данных*)
irisplot = ListPlot[GroupBy[irises, Key["Species"] → KeyDrop["Species"]],
           |диаграмм··· |сгруппировать по  |ключ  |отбросить ключи
   Sequence[PlotMarkers → "OpenMarkers", Frame → True],
   |последов··· |маркеры на графике  |рамка  |истина
   AspectRatio → 1, PlotRange → {{4, 8}, {1, 5}},
   |аспектное отношение |отображаемый диапазон графика
   FrameLabel → {"sepal length (cm)", "sepal width (cm)"},
   |пометка для обрамления
   PlotLegends → {"setosa examples", "versicolor examples",
   |легенды графика
      "virginica examples"}, ImageSize → 260];
                             |размер изображения
In[7]:= (*Отображаем график*)
        irisplot
```

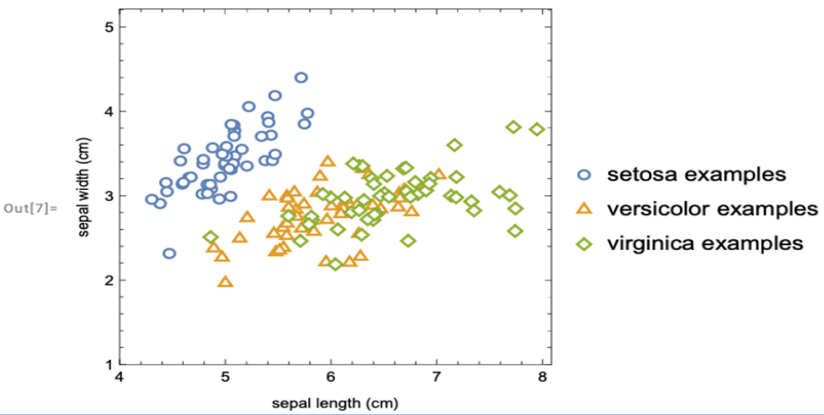

Рис. 5 Представление результирующего набора данных

2. Набор данных Boston Homes [3] (см. рисунок 6).
3.

```
(*Загрузка данных*)
houses = ResourceData["Sample Data: Boston Homes"][[All, {"RM", "AGE", "MEDV"}]];
(*Вывод первых трех строк данных*)
Dataset[houses, MaxItems → 3]
```

RM	AGE	MEDV
6.575	65.2	24
6.421	78.9	21.6
7.185	61.1	34.7

rows 1–3 of 506

Рис. 6 Набор данных Boston Homes

Этот набор данных содержит информацию о недвижимости в Бостоне, включая такие характеристики, как количество комнат, возраст здания, уровень преступности в районе и т. д. Метод ближайших соседей применяется здесь для регрессии, предсказывая стоимость жилья.

Для такого типа задач выбор правильной метрики расстояния играет важную роль. Например, евклидово расстояние хорошо работает, когда признаки имеют одинаковый масштаб, но если данные имеют разный порядок величин (например, площадь квартиры измеряется в квадратных метрах, а возраст здания в годах), то лучше использовать стандартизированные расстояния или косинусное сходство.

Результирующий набор данных:

```
(*Построение 3D-графика*)houseplot =
  ListPointPlot3D[houses, AxesLabel → {"Number of Rooms", "Age", "Price"},
   ColorFunctionScaling → False, ImageSize → 350,
   ViewPoint → {-1.810, 2.644, 1.085}, Filling → Bottom, SphericalRegion → False];
(*Вывод графика*)
houseplot
```

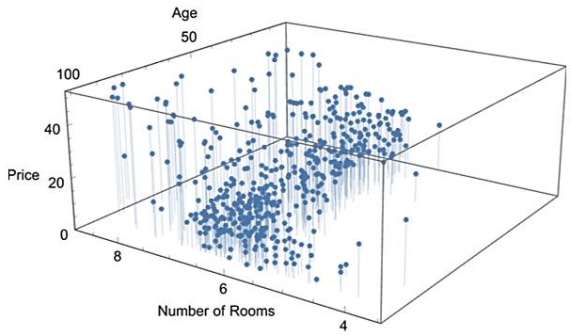

Рис. 7 Представление результирующего набора данных

Влияние гиперпараметров на обучение

При работе с методом ближайших соседей важно правильно выбирать:

- Количество соседей. Малое значение может привести к переобучению, а слишком большое — к недообучению.
- Метрику расстояния (евклидово, манхэттенское, косинусное сходство и т. д.).
- Веса соседей (взвешенные методы уменьшают влияние дальних точек).

Практическое применение метода

Классификация на наборе данных Fisher's Irises:

Рассмотрим работу метода k-NN с k=1. На рисунке 8 представлена визуализация областей классификации. Границы между классами выглядят фрагментированными, так как каждый объект формирует свою локальную область влияния. Это делает модель чувствительной к шуму и случайным выбросам, что приводит к переобучению. Результат представлен на рисунке 9.

Рис. 8 Операторы визуализации набора данных

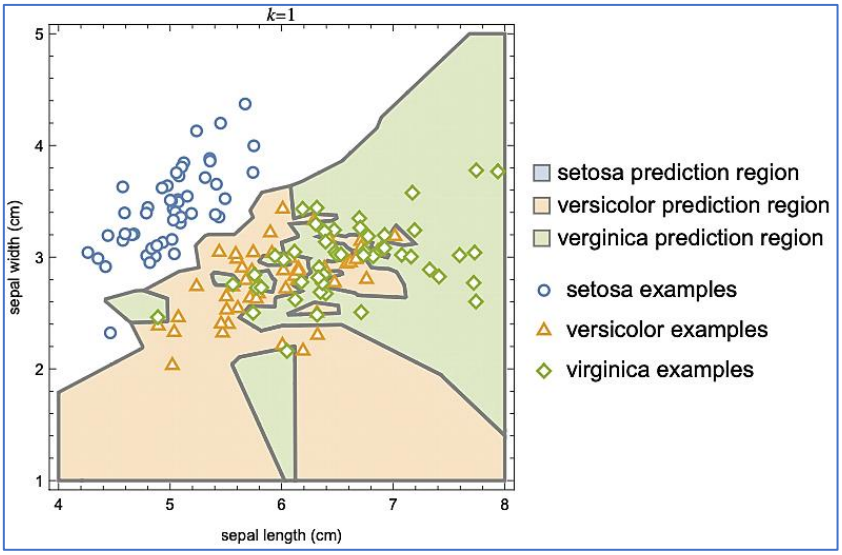

Рис. 9 Визуализация классификации

Теперь увеличим количество соседей до k=20 и снова визуализируем области классификации (см. рисунок 10). На этом этапе границы стали более плавными, а модель – менее чувствительной к отдельным точкам. Такое сглаживание уменьшает вероятность переобучения, но при этом может приводить к потере деталей в структуре данных. Результат представлен на рисунке 11.

```
In[12]:= nearest20 = Classify[irises → "Species",
                    |классифицировать
           Method → {"NearestNeighbors", "NeighborsNumber" → 20},
           |метод
           PerformanceGoal → "DirectTraining"]
           |целевая установка производительности
Out[12]=
         ClassifierFunction[ Input type:  {Numerical, Numerical }
                             Classes:  setosa, versicolor, virginica ]

In[13]:= regions = Quiet[RegionPlot[{Inactive[nearest20[{x, y}] === "setosa"],
                   |беззв… |визуализация … |неактивный
             Inactive[nearest20[{x, y}] === "versicolor"],
             |неактивный
             Inactive[nearest20[{x, y}] === "virginica"]}, {x, 4, 8}, {y, 1, 5},
             |неактивный
           Sequence[BoundaryStyle → Gray],
           |последов… |стиль границы  |серый
            FrameLabel → {"sepal length (cm)", "sepal width (cm)"},
            |пометка для обрамления
            PlotLegends →
            |легенды графика
              {"setosa prediction region", "versicolor prediction region",
               "verginica prediction region"}, PerformanceGoal → "Quality",
                                               |целевая установка производительности
            MaxRecursion → 5]];
            |максимальный уровень рекурсии
      Show[regions, irisplot, ImageSize → 290],
      |показать      |размер изображения
       PlotLabel → Row[{Style["k", Italic, FontFamily → "Times"],
       |пометка гра… |ряд |стиль  |курсив |семейство шри… |умножить
          Style["=20", FontFamily → "Times"]}]]
          |стиль  |семейство шри… |умножить
```

Рис. 10 Операторы визуализации результатов

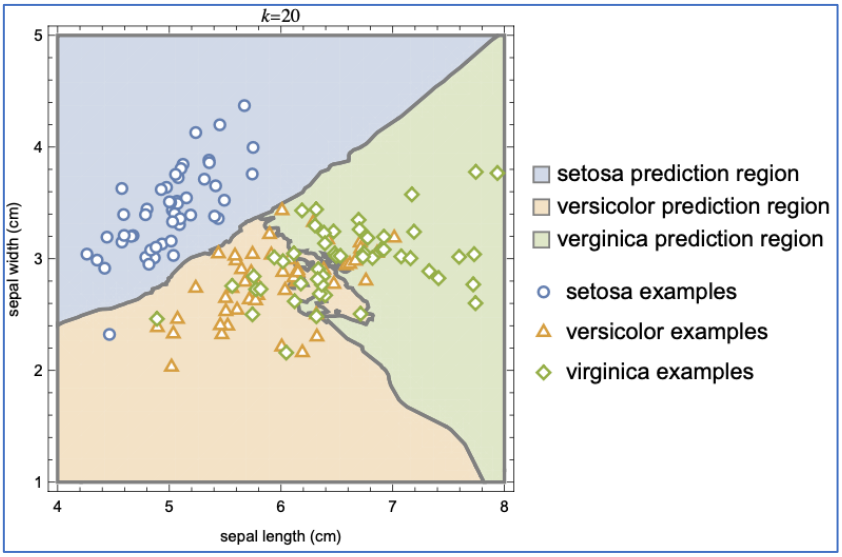

Рис. 11 результаты классификации при k=20

Визуализация вероятностей предсказания:

Дополнительно можно рассмотреть вероятности принадлежности к классам при k=20. На рисунке 12 представлена трёхмерная карта вероятностей, где видно, что вероятность принадлежности к определённому классу изменяется нелинейно. В отличие от логистической регрессии, которая строит чёткую границу, метод ближайших соседей формирует более гибкие переходные зоны.

```
points = Tuples[{Range[4, 8, 0.03], Range[1, 5, 0.03]}];
proba = Values[nearest20[points, "Probabilities"][[
    All, {"versicolor", "setosa", "virginica"}]]];

ListPlot3D[(Join[points, List /@ #1, 2] &) /@ Transpose[proba],
 PlotLegends → {"probability of versicolor",
   "probability of setosa", "probability of virginica"},
 AxesLabel → {"sepal length (cm)", "sepal width (cm)", "probability"},
 SphericalRegion → False]
```

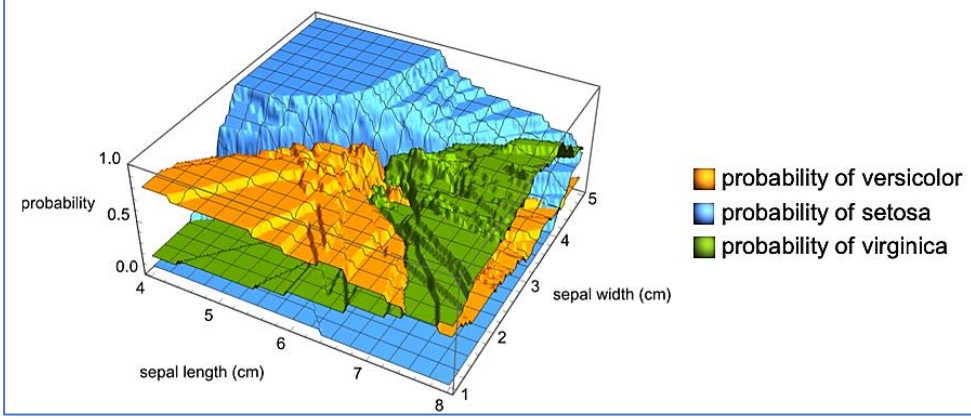

Рис. 12 вероятности отнесения к классам

Регрессия на наборе данных Boston Homes:

Метод k-NN можно использовать для прогнозирования стоимости жилья. На рисунке 13 показана трёхмерная поверхность регрессионных предсказаний. В отличие от линейной регрессии, которая строит глобальную зависимость, метод ближайших соседей выявляет локальные зависимости, обеспечивая более точные прогнозы в сложных случаях.

```
housenn = Predict[houses → "MEDV", Method → "NearestNeighbors"];
          предсказать              метод
Show[houseplot,
показать
 ListPlot3D[points = Tuples[{Range[3, 9, 0.05`], Range[0, 100]}];
 3-мерная диаграмма раз… наборы и… диапазон          диапазон
  MapThread[Append, {points, housenn[points]}],
  нанизать п… добавить в конец
  Sequence[PlotStyle → Directive[Opacity[0.8`]],
  последов… стиль графика директива прозрачность
   NormalsFunction → None, SphericalRegion → False]]]
   функция нахождения … ни о… оставлять зазор для … ложь
```

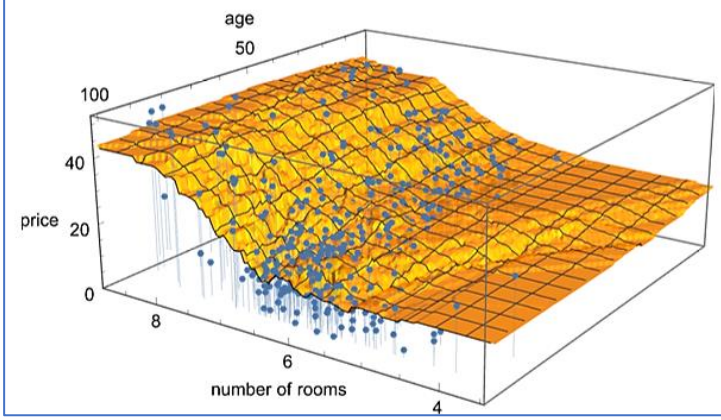

Рис. 13 Результаты регрессии

В данном примере количество соседей было автоматически определено как k=20, что позволило достичь оптимального баланса между точностью и устойчивостью модели:

$$Information[housenn, NeighborsNumber] = 20$$

Выводы и заключение

Метод ближайших соседей является одним из самых простых и интуитивно понятных алгоритмов машинного обучения. Его главное преимущество — отсутствие необходимости явного обучения модели. Однако он также имеет ряд недостатков, которые важно учитывать.

Плюсы метода:

- Простота реализации и интерпретации.
- Универсальность: применяется как для классификации, так и для регрессии.
- Возможность работать с разными типами данных и метрик расстояния.

Минусы метода:

- Высокая вычислительная сложность при больших выборках.
- Чувствительность к выбору числа соседей и метрики расстояния.
- Возможное переобучение при малых значениях.

Метод ближайших соседей стоит использовать, когда:

- данных не слишком много, и важна простота реализации.
- данные имеют естественную геометрическую интерпретацию (например, геопространственные данные).

- необходимо быстро проверить гипотезу перед использованием более сложных моделей.

Несмотря на свою простоту, метод ближайших соседей (k-NN) продолжает оставаться важным инструментом в машинном обучении, особенно в задачах начального анализа данных и построения базовых моделей.

Список литературы:

1. Кузнецов С. О. Методы и алгоритмы машинного обучения. — М.: БИНОМ, 2019.
2. Федоров А. В. Алгоритмы обработки данных: от теории к практике. — СПб.: Питер, 2021.
3. Известные наборы данных, URL: https://neerc.ifmo.ru/wiki/index.php?title=%D0%98%D0%B7%D0%B2%D0%B5%D1%81%D1%82%D0%BD%D1%8B%D0%B5_%D0%BD%D0%B0%D0%B1%D0%BE%D1%80%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85#Iris (Дата обращения 13.03.2025).
4. Wolfram Mathematica, URL: https://www.wolfram.com/mathematica/ (Дата обращения 13.03.2025).