Метод k-ближайших соседей

Балакшина А.Д.
Матвеева М.А.

Дата выполнения: 13.10.25

План

- Что такое *k-NN*?
- Области применения
- Алгоритм
- Предобработка данных
- Численный пример
- Детали реализации
  - Выбор метрики
  - Выбор числа соседей
  - Поиск соседей
- Примеры применения
- Достоинства и недостатки
- В заключение
- Источники

Что такое *k-NN*?

Метод *k*-ближайших соседей *(k-Nearest Neighbor, k-NN)* используется для решения задачи классификации. Он относит объекты к классу, которому принадлежит большинство из k его ближайших соседей в многомерном пространстве признаков. Это один из простейших алгоритмов обучения классификационных моделей.

Число k — это количество соседних объектов в пространстве признаков, которые сравниваются с классифицируемым объектом. Иными словами, если $k = 10$, то каждый объект сравнивается с *10*-ю соседями.

Области применения

- Банковская система
- Рекомендательные системы
- Биоинформатика и генетика
- Компьютерное зрение и распознавание образов

Алгоритм

Пусть имеется набор данных, состоящий из *n* наблюдений $X_i, (i = 1, ..., n)$, для каждого из которых задан класс $C_j, (j = 1, ..., m)$. Тогда на его основе может быть сформировано обучающее множество, все примеры которого представляют собой пары X_i, C_j.

Обучение: алгоритм просто запоминает векторы признаков наблюдений и их метки классов. На этом этапе задаётся параметр алгоритма k, т.е. число «соседей», которые будут использоваться при классификации, и выбирается метрика расстояния.

Классификация: предъявляется новый объект, для которого метка класса не задана. Для него определяются k ближайших предварительно классифицированных наблюдений. Затем выбирается класс, которому принадлежит большинство из k ближайших примеров-соседей, и к этому же классу относится классифицируемый объект.

Алгоритм

Иллюстрация

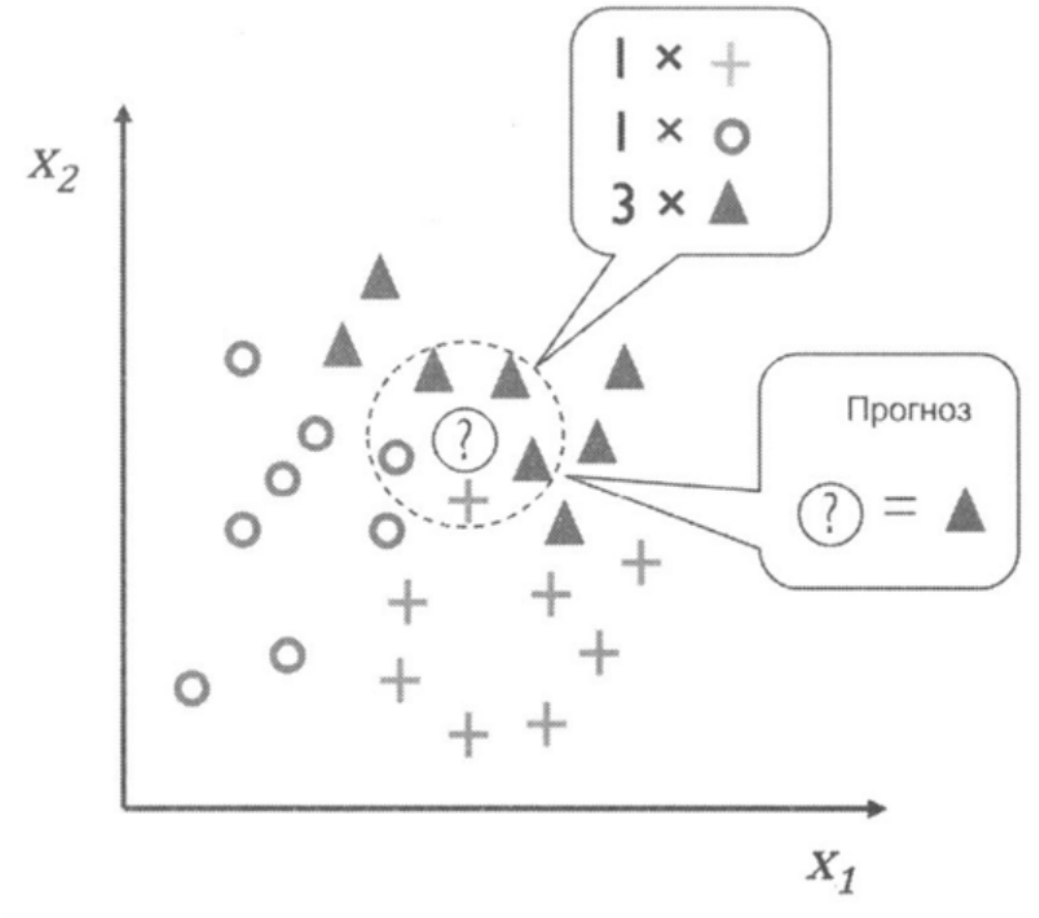

Рис 1. Принцип работы алгоритма *k*-ближайших соседей

Предобработка данных

1. В метрических методах важно выполнять *масштабирование* (стандартизацию, нормализацию), если у признаков разный диапазон числовых значений. Иначе признаки с бо́льшим диапазоном начнут доминировать, и классификация будет ошибочной.
2. Метод k-ближайших соседей сильно страдает от «проклятия размерности».

Предобработка данных

Проклятие размерности

Суть проблемы: с ростом числа признаков пространство становится чрезвычайно разреженным. Объекты в среднем оказываются далеко друг от друга, и понятие «близости», на котором основан метод, теряет смысл.

Связь между «ближайшими» соседями и целевым объектом ослабевает, что приводит к резкому падению качества точности прогноза.

На практике: данные часто лежат на многообразии меньшей размерности. Также многие признаки коррелируют друг с другом, и реальная структура данных проще, чем кажется.

Решение:

- отбор наиболее значимых признаков;
- уменьшение размерности.

Численный пример (1/5)

Пусть имеется набор данных о заёмщиках банка часть из которых допустили просрочку по платежу. Признаками являются возраст и среднемесячный доход. Метками класса в поле «Просрочено» будут «Да» и «Нет».

X_1- возраст	X_2- доход	Y- Просрочено
46	40	Нет
36	54	Нет
34	29	Да
38	23	Да

Табл. 1

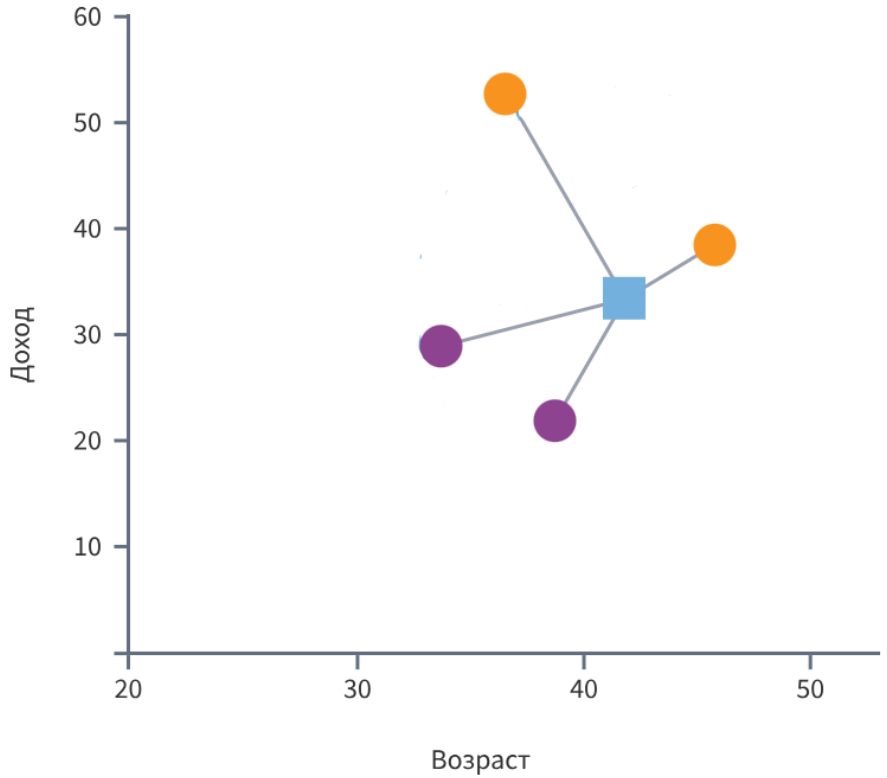

Рис. 4

Численный пример (2/5)

На рис.5 оранжевыми кружками представлены объекты класса «Нет», а фиолетовыми класса «Да». Синий квадрат - новый заёмщик. Задача заключается в том, чтобы выполнить классификацию нового заёмщика для которого $A_1 = 42$ и $A_2 = 34$ с целью оценить возможность просрочки им платежей.
Зададим значение параметра $k = 3$.

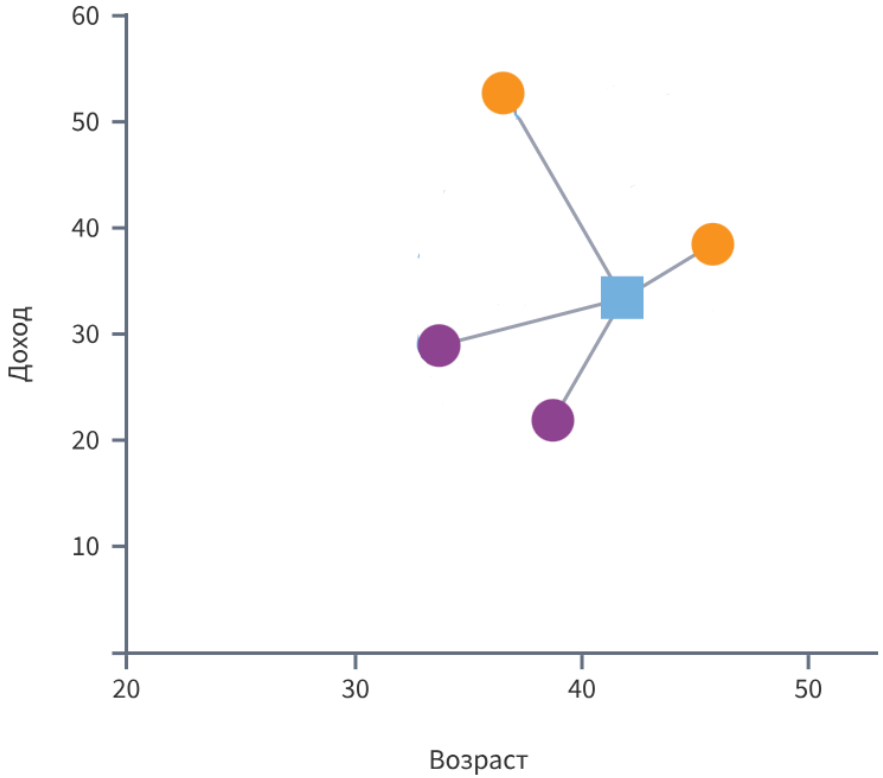

Рис. 5

Численный пример (3/5)

Рассчитаем расстояние между вектором признаков классифицируемого объекта и векторами обучающих примеров по формуле $D(A,X) = \sqrt{(A_1 - X_1)^2 + (A_2 - X_2)^2}$ и установим для каждого примера его ранг (табл. 2).

X_1	X_2	Расстояние	Ранг	Y
46	40	7.2	1	Нет
36	54	20.9	4	Нет
34	29	9.4	2	Да
38	23	11.7	3	Да

Табл. 2

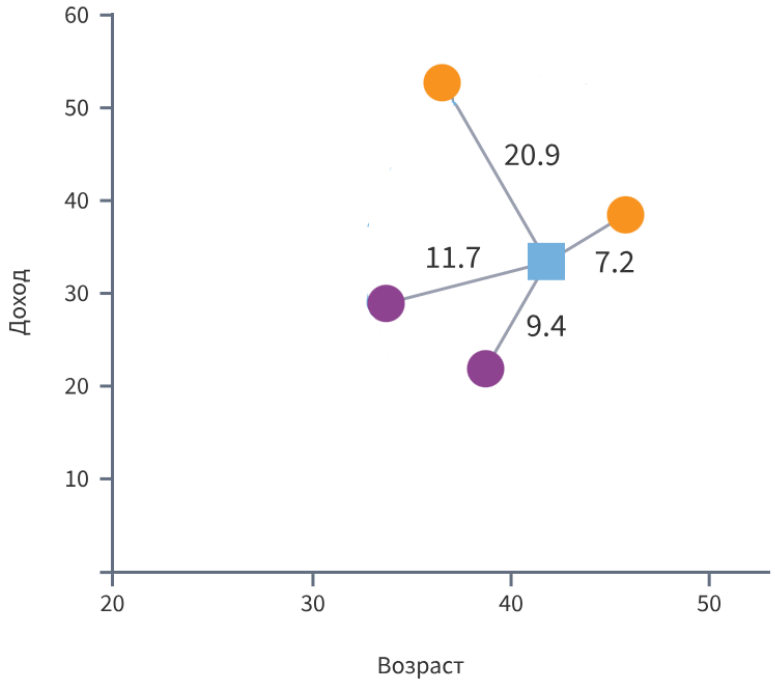

Рис. 6

Численный пример (4/5)

Исключим из рассмотрения пример, который при $k = 3$ не является соседом и рассмотрим классы оставшихся (табл. 3).

X_1	X_2	Расстояние	Ранг	Y
46	40	7.2	1	Нет
34	29	9.4	2	Да
38	23	11.7	3	Да

Табл. 3

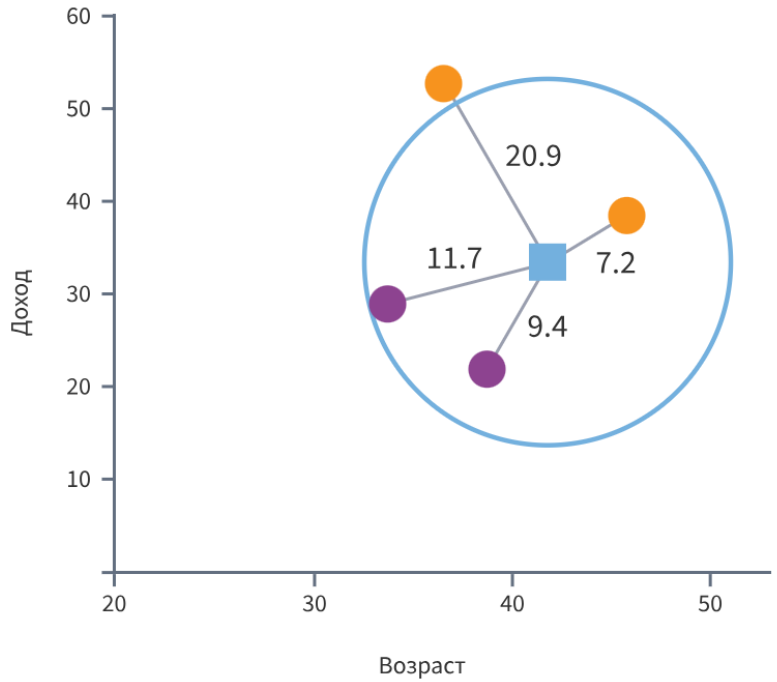

Рис. 7

Численный пример (5/5)

Таким образом, из трёх ближайших соседей (на рисунке расположены внутри круга) классифицируемого объекта два имеют класс «Да», а один — «Нет». Следовательно, путём простого невзвешенного голосования определяем его класс как «Да». На основании работы классификатора делаем вывод, что заёмщик с заданными характеристиками может допустить просрочку по выплате кредита.

Выбор метрики (1/2)

Как правильно выбрать функция расстояния ρ?

В подавляющем большинстве случаев обычное евклидово расстояние $\rho(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$ будет хорошим выбором. Однако в некоторых случаях другие функции будут подходить лучше, например:

- Манхэттенская метрика: $\rho(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} |x_i - y_i|$
  Часто используется в высокоразмерных пространствах из-за лучшей устойчивости к выбросам. Представим, что два объекта в 1000-размерном пространстве почти идентичны, но сильно отличаются по одному из признаков. Это почти наверняка свидетельствует о выбросе в этом признаке, и объекты, скорее всего, очень близки. Евклидово расстояние усилит различие в единственном признаке и сделает их более далёкими друг от друга. Этого недостатка лишена манхэттенская метрика — в ней вместо квадрата используется модуль.

Выбор метрики (2/2)

- Косинусное расстояние:
  $\rho(\mathbf{x}, \mathbf{y}) = 1 - \frac{\mathbf{x} \cdot \mathbf{y}}{\|\mathbf{x}\|\|\mathbf{y}\|} = 1 - \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2}\sqrt{\sum_{i=1}^{n} y_i^2}}$
  Эта метрика хороша тем, что не зависит от норм векторов. Такое поведение бывает полезно в некоторых задачах, например, при поиске похожих документов. В качестве признаков там часто используются количества слов. При этом интуитивно кажется, что если в тексте использовать каждое слово в два раза больше, то тема этого текста поменяться не должна. Поэтому как раз в этом случае нам не важна норма вектор-признака.
- Расстояние Жаккара $\rho(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$
  Его стоит использовать, если исследуемые объекты — это некоторые множества. Это полезно тем, что нет нужды придумывать векторные представления для этих множеств, чтобы использовать традиционные метрики.

Выбор числа соседей

Маленькое k

- Если выбрать слишком малое k: есть опасность, что единственным ближайшим объектом окажется «выброс», т.е. объект с неправильно определённым классом, и он даст неверное решение.

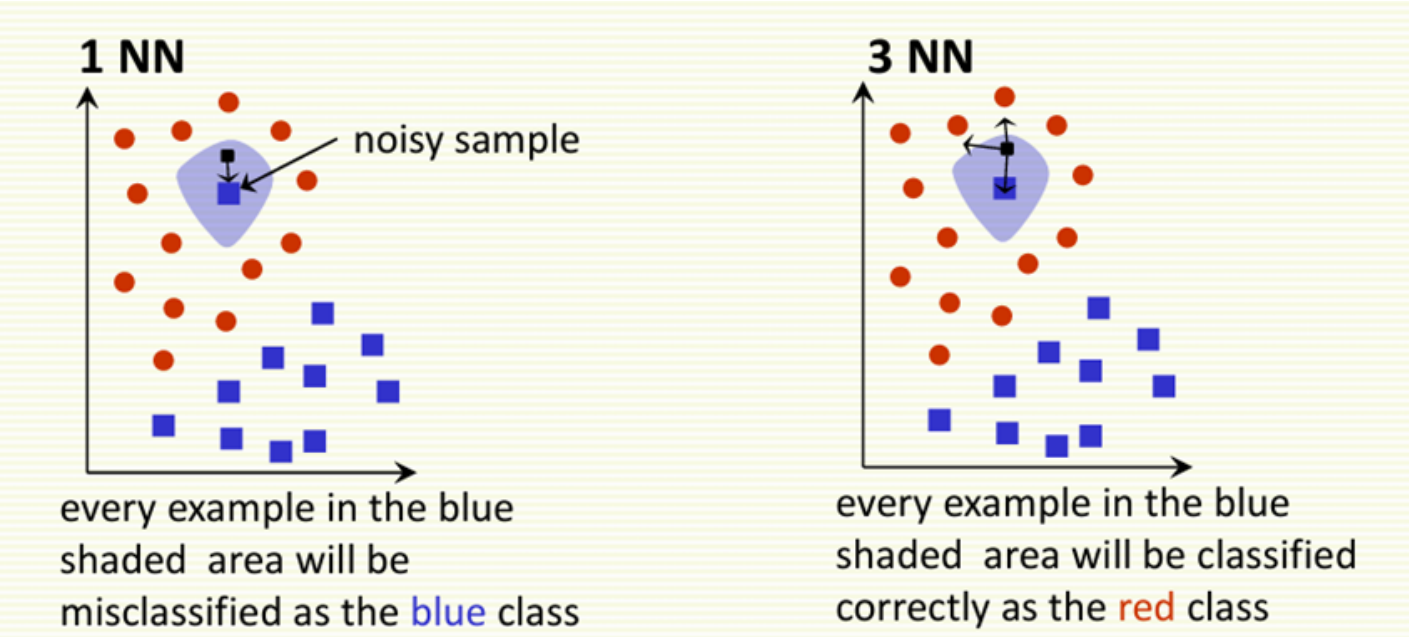

Рис 2. Малое k и его влияние на класс

Выбор числа соседей

Большое k

- Если выбрать слишком большое k: классификатор становится менее чувствительным к локальным закономерностям и в большей степени зависит от общего распределения данных. *Например*, когда k равно общему числу объектов N - понятно, что тогда «победит» самый популярный (модальный) класс, и расстояние до исследуемого объекта не будет играть вообще никакой роли.

Поиск оптимального k в k-NN - это балансировка между переобучением и недообучением.

Оптимальный выбор числа соседей (1/4)

Корректный выбор параметра *k* является ключевым фактором, определяющим эффективность алгоритма. Итоговое решение будет зависеть в том числе от специфики задачи и данных. Общие рекомендации для его выбора:

- В задачах *бинарной* классификации не следует использовать чётные значения *k*, чтобы избежать ничьи в голосовании.
- *Эвристика:* $k = \sqrt{N}$, где N — количество образцов в обучающей выборке. Это значение часто бывает неплохим начальным приближением.

Оптимальный выбор числа соседей (2/4)

Более надежные способы подбора:

- Метод локтя *(Elbow method)*
  - Задайте диапазон для k (например, [10, 40]).
  - Обучите модель, используя значения k из указанного выше диапазона.
  - Постройте график частоты ошибок от k.
  - Выберите оптимальное значение k - им будет точка K, при которой производительность значительно улучшается, прежде чем выровняться/начать снова ухудшаться на графике.

Оптимальный выбор числа соседей (3/4)

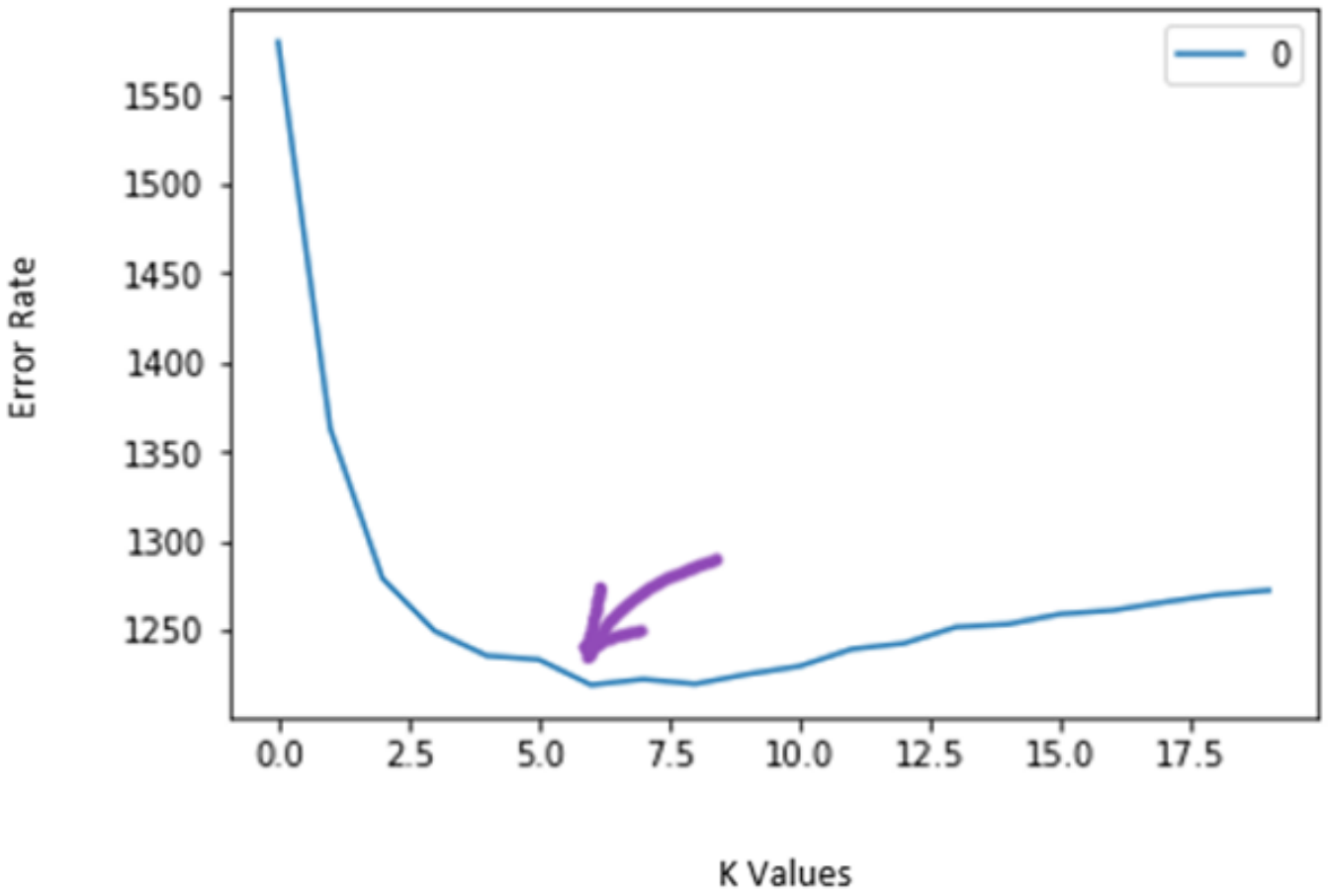

Рис 3. Пример Метода локтя (здесь оптимальное $k = 6$)

Оптимальный выбор числа соседей (4/4)

- Поиск по сетке с перекрестной проверкой *(Grid Search with Cross-Validation)*
  - Задайте диапазон значений K.
  - Используйте кросс-валидацию для поиска в указанном диапазоне значений K. *(Процесс кросс-валидации: данные разбиваются на несколько частей, модель по очереди обучается на всех частях кроме одной, которая используется для проверки, а затем вычисляется среднее арифметическое всех полученных оценок качества как итоговая метрика.)*
  - Поиск по сетке оценивает производительность для каждого значения K и выбирает оптимальное значение.

Поиск соседей

С первого взгляда в нахождении соседей нет никакой проблемы: можно просто перебрать все объекты из обучающей выборки, посчитать для каждого из них расстояние до тестового объекта и затем найти минимум.

Однако несмотря на то, что сложность такого поиска линейная по N, она также зависит и от размерности пространства признаков. Если $x \in \mathbb{R}^D$, то сложность такого алгоритма поиска $O(ND)$.

Проблема: в типичной задаче машинного обучения количество признаков может быть порядка 100, а размер выборки и вовсе может исчисляться десятками и сотнями тысяч объектов + осложняется всё тем, что данный поиск необходимо выполнять на этапе применения модели, который должен быть быстрым!!!

Всё это означает, что возникает необходимость в более быстрых методах поиска ближайших соседей, чем простой перебор.

Поиск соседей. Методы

Сегодня мы не будем подробно рассматривать эти методы, а просто перечислим основные.

Точные методы

- Полный перебор с различными эвристиками. Например, можно выбрать подмножество признаков и считать расстояние только по ним. Оно будет оценкой снизу на реальное расстояние, поэтому если оно уже больше, чем до текущего ближайшего объекта, то можно сразу отбросить этот объект и переходить к следующему
- K-d-деревья

Приближённые методы

- Random projection trees
- Locality-sensitive hashing (LSH)
- Proximity graphs Hierarchical navigable small world (HNSW)

Примеры применения

Пример из медицины (область "Биоинформатика")

- **Задача:** Определить, является ли новообразование у пациента доброкачественным.
- **Решение:** В системе есть база данных тысяч пациентов, где для каждого известны параметры опухоли (размер, форма, плотность и т.д.) и точный диагноз. Для нового пациента измеряются те же параметры. Алгоритм *k-NN* находит 5 (k=5) записей с самыми похожими параметрами. Если 4 из них были злокачественными, система порекомендует классифицировать новую опухоль как «злокачественную».

Примеры применения

Пример из интернет-магазина (область "Рекомендательные системы")

- **Задача:** Порекомендовать книгу пользователю.
- **Решение:** Алгоритм ищет пользователей, которые купили много тех же книг, что и наш целевой пользователь (т.е. его «ближайших соседей» по вкусам). Затем он анализирует, какие книги эти «соседи» покупали и любили, а наш целевой пользователь — еще нет. Эти книги и становятся рекомендациями.

Достоинства и недостатки

Достоинства

- устойчивость к выбросам и аномальным значениям
- программная реализация алгоритма относительно проста
- результаты работы алгоритма легко поддаются интерпретации
- не требует, чтобы данные следовали какому-либо распределению

Недостатки

- данный метод не создает каких-либо моделей, обобщающих предыдущий опыт
- неэффективный по памяти, поскольку нужно хранить всю обучающую выборку
- вычислительно дорогой по той же причине
- обязательное требование: стандартизация/нормализация данных
- «проклятие размерности»

В заключение

Метод *k*-ближайших соседей — это классический алгоритм, который наглядно демонстрирует базовый принцип машинного обучения: «похожие объекты ведут себя похоже». Его *сила* — в простоте и интерпретируемости, а *слабость* — в вычислительной сложности при работе с большими объемами информации.

Источники

1. Машинное обучение с PyTorch и Scikit-Leam: Пер. с англ. / С. Рашка, Ю. Лю, В. Мирджалили. - Астана: Фолиант, 2024. - 688 с.
2. CS9840: Machine Learning. Lecture 2: kNN Classification. / O. Veksler. - University of Western Ontario: 2014.
3. Классификация данных методом k-ближайших соседей // Loginom URL: `https://loginom.ru/blog/knn` (дата обращения: 12.10.2025).
4. Метод K ближайших соседей // Машинное и глубокое обучение URL: `https://deepmachinelearning.ru/docs/Machine-learning/Metric-methods/KNN` (дата обращения: 12.10.2025).
5. Метрические методы // Яндекс Образование URL: `https://education.yandex.ru/handbook/ml/article/metricheskiye-metody` (дата обращения: 12.10.2025).
6. How to Find The Optimal Value of K in KNN // Geeksforgeeks URL: `https://www.geeksforgeeks.org/machine-learning/how-to-find-the-optimal-value-of-k-in-knn/?ysclid=mgcq75ohg2382100565` (дата обращения: 12.10.2025).

Спасибо за внимание!