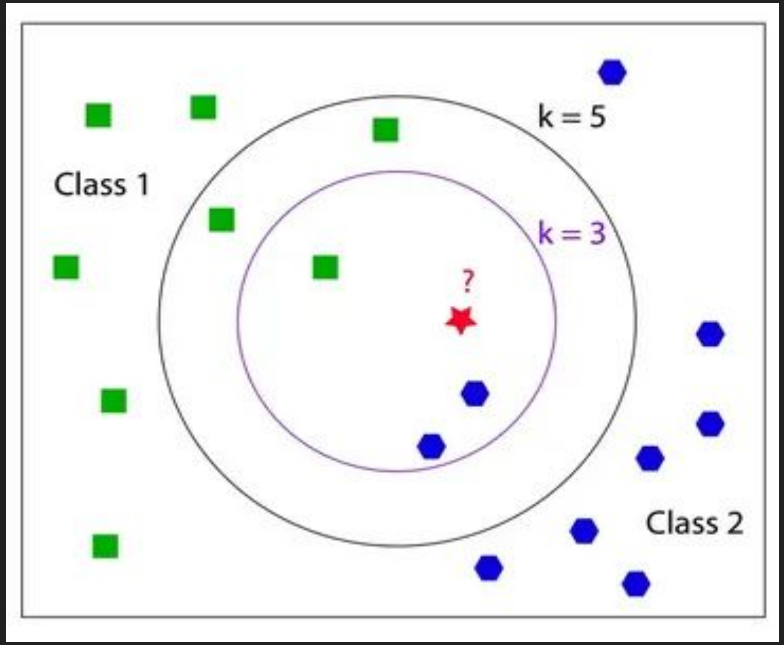
KNN, K-MEANS, NAIVE-BAYES

KNN

Алгоритм:

1. Выбираем гиперпараметр K
2. Выбираем как будем считать расстояние
3. Вычисляем расстояния для всех объектов на обучающей выборке
4. Определяем метку класса (или числовое значение)



Варианты расчета расстояния

- Косинусное расстояние

$$\rho(x, y) = 1 - \cos\theta = 1 - \frac{x \cdot y}{|x||y|}$$

- Манхэттенское расстояние (Манхэттенская метрика)

$$\rho(x, y) = \sum_i |x_i - y_i|$$

- Расстояние Жаккара

$$\rho(A, B) = 1 - \frac{|A \cap B|}{|A \cup B|}$$

- Метрика Минковского

$$\rho(x, y) = \left(\sum_i |x_i - y_i|^p \right)^{1/p}$$

- Евклидово расстояние

$$\rho(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

Плюсы

- Можем использовать как в задачах регрессии, так и в задачах классификации
- Довольно неплохо позволяет решить задачу (крепкий бейзлайн, который можно быстро сделать)
- Высокая универсальность из-за возможности выбора алгоритма расчета расстояния
- Легко интерпретируемый

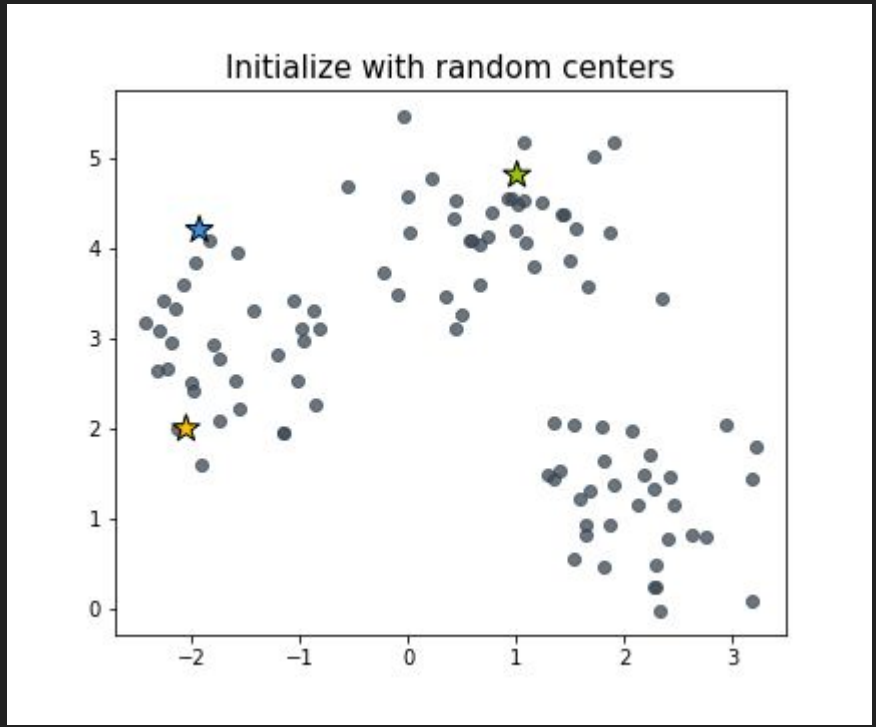
Минусы

- Дорого по памяти и вычислениям
- Данные сильно влияют(масштаб, выбросы, кол-во данных и тд)
- Нужно правильно подобрать метрику, чтобы получить корректный результат (в других алгоритмах это не влияет на их корректность работы напрямую)

K-means

Алгоритм:

1. случайно инициализируем центры кластеров
2. находим ближайшее скопление точек
3. реинициализируем центр кластеров
4. повторяем пункты 2 и 3 пока не придем к окончательному решению



Плюсы

- Высокая универсальность из-за возможности выбора алгоритма расчета расстояния
- Легко интерпретируемый
- Работает относительно быстро
- Не зависит от типа данных (с оговорками)

Минусы

- Вопрос: как выбрать K?
- Что делать с кластерами сложной формы
- ОЧЕНЬ чувствителен к выбору начальных центров
- Результаты могут сильно отличаться от запуска к запуску

Вспомним что это за покемон

- Априорная вероятность?
- Апостериорная вероятность?
- Условная вероятность?
- Теорема Байеса?

Naive bayes

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

где:

- P(A|B) — апостериорная вероятность события A при условии выполнения события B;
- P(B|A) — условная вероятность события B при условии выполнения события A;
- P(A) и P(B) — априорные вероятности событий A и B соответственно.

Еще страшная формула

$$P(y_i = C_k|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|y_i = C_k)P(y_i = C_k)}{P(\mathbf{x}_i)}$$

Optimal class label:

$$C^* = \arg\max_k P(y_i = C_k|\mathbf{x_i})$$

Алгоритм работы

На этапе обучения наивный байесовский классификатор строит вероятностную модель, основываясь на тренировочных данных

- Вычисление априорных вероятностей классов
- Вычисление Условная вероятность признаков
- Лапласовское сглаживание

Инференс (предсказание)

После обучения модель можем классифицировать новые объекты. Для этого используем **теорему Байеса**:

Выбор класса:

Классификатор выбирает класс, для которого вероятность встретить именно эти признаки максимальна

Плюсы

- Можем найти выбросы в данных
- Имеем представление о распределение
- Ему не надо много данных
- Быстрый
- Если нет корреляции, то это не беда (а фича)

Минусы

- В реальном мире признаки всё-таки зависят друг от друга
- Проблемы с новыми признаками