



Работа с признаками



Работа с признаками

Числовые признаки

Дата/время и координаты

Порядковые и категориальные признаки

Масштабирование

Отбор признаков



Ранг

Идея: заменить значения X_1, \dots, X_n на ранги R_1, \dots, R_n , где ранг R_i — порядковый номер X_i в упорядоченном наборе.

X_i	7.3	2.2	0.3	6.2	1.6	6.2	9.6
R_i	6	3	1	4.5	2	4.5	7

Для нового объекта x

- ▶ ранг ближайшего объекта из train.
- ▶ средний ранг по ближайшим объектам из выборки.
- ▶ взвешенный средний ранг по ближайшим объектам из выборки.

Смысл:

Подвигаем выборки к остальным объектам,
⇒ они перестают вносить большой вклад в модель.

Применение:

Иногда хорошо работает для KNN, лин. моделей, нейросетей, особенно если нет времени разбираться с выбросами.



Трансформации

- ▶ Логарифмическая

$$\tilde{x} = \ln(x)$$

- ▶ Возведение в степень

$$\tilde{x} = \sqrt{x + 1}$$

- ▶ Преобразование Бокса-Кокса

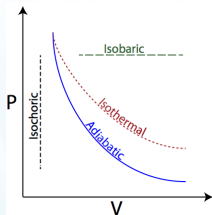
$$\tilde{x} = \begin{cases} \frac{x^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \ln(x), & \lambda = 0 \end{cases}$$

Смысл: Объекты с большими значениями признаков становятся ближе к остальным объектам. Особенно хорошо работают для нейростей.

Замечание: при отрицательных значениях нужно произвести сдвиг.



Генерация признаков



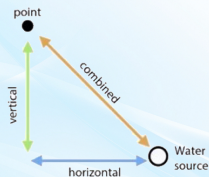
Имеем:

Объем: $2m^3$

Давление: 10 кПа

Получаем:

PV: 20 Дж



Имеем:

Расстояние по вертикали: 3 м

Расстояние по горизонтали: 2 м

Получаем:

Полное расстояние: 3.60 м

Такие признаки помогут многим моделям,
т.к. модели плохо умеют умножать/делить фичи друг на друга.



Квантование (Binning)

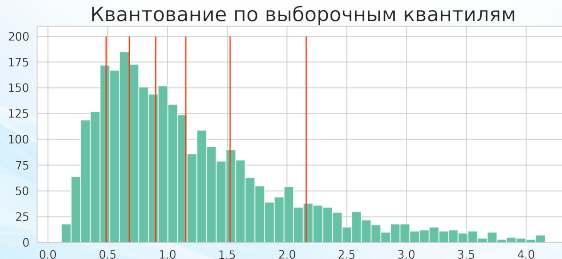
Разбиение множества значений признака на интервалы (бины) и замена признака на категориальную переменную.

- **Fixed-Width Binning**

Выбор точек разбиения вручную или равномерно

- **Adaptive Binning**

Выбор точек разбиения в зависимости от выборки.





Работа с признаками

Числовые признаки

Дата/время и координаты

Порядковые и категориальные признаки

Масштабирование

Отбор признаков



Дата и время

Дата/timestamp в чистом виде не достаточно информативны.
Полезно добавить дополнительные признаки.

- ▶ **Характеристики даты**

Год, сезон, день месяца, день недели, час, минуты, секунды

- ▶ **Время с момента некоторого события**

Кол-во дней с прошлого дождя.

Кол-во секунд с последнего столкновения частиц.

Кол-во дней от фиксированной даты X.

- ▶ **Синус от времени с периодом кратным длине сезона/дня.**



Координаты

Местоположение обычно характеризуется широтой и долготой.
В чистом виде они недостаточно информативны.
Рассмотрим разные случаи.

- ▶ **Есть дополнительные данные**

Можно добавить расстояния до значимых объектов





Координаты

- ▶ **Дополнительных данных нет**

Придумаем местоположения сами по имеющимся данным.

Пример для анализа цен на квартиры.

- ▶ Разделим карту на квадраты.
В каждом квадрате найдем самую дорогую квартиру.
Для объектов в квадрате добавим расстояние до этой квартиры.
- ▶ Организуем имеющиеся точки в кластеры.
Найдем центры кластеров.
Будем использовать их как важные местоположения.
- ▶ Найдем район с старыми строениями.
Посчитаем расстояние до него.





Координаты

► Агрегирующие статистики

Посчитаем статистики по объектам, находящимся рядом.

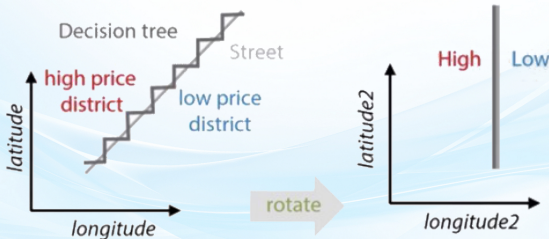
Примеры: средняя масса частиц, средняя скорость, плотность потока

► Поворот координат

Повернем координаты и возьмем это как новые признаки.

Можно сделать несколько разных поворотов.

Полезно для tree-based методов.





Работа с признаками

Числовые признаки

Дата/время и координаты

Порядковые и категориальные признаки

Масштабирование

Отбор признаков



Порядковые признаки

Для таких признаков значения сравнимы между собой,
но *расстояния между ними не определены.*

Примеры:

- ▶ Класс билета
A, B, C
- ▶ Образование
школа, бакалавриат, магистратура, аспирантура
- ▶ Оценка
1, 2, 3, ..., 9, 10

Обработка: **Label encoding**

Каждому состоянию сопоставляем число.

Порядок состояний должен сохраниться.



Категориальные признаки

Значения категориальных признаков не сравнимы друг с другом.

Категория — одно значение данного признака.

Примеры:

- ▶ Спектральный класс звезды

O, B, A, F, G, K, M

- ▶ Пол

мужской, женский



Как работать с категориальными признаками?

► Label encoding

Каждому состоянию сопоставляем число.

`sklearn.preprocessing.LabelEncoder` — алфавитный порядок.

`pandas.factorize` — в порядке встречаемости значения.

Минусы:

Линейные модели плохо работают с такими признаками.

Деревья могут работать, но потребуется глубокое дерево.



Как работать с категориальными признаками?

► One-hot encoding

Создается $K - 1$ новых бинарных признаков,
где K — кол-во категорий.

Замечание 1:

Если есть пара вещественных, очень значимых признаков,
то деревьям и KNN будет трудно обращать на них внимание
из-за большого кол-ва новых one-hot признаков.

Замечание 2:

Если у кат. фичи много уникальных значений,
то добавим много новых признаков, в которых,
возможно, только пара ненулевых элементов.

Тогда обычно хранят только ненулевые элементы — **sparse matrix**.



Как работать с категориальными признаками?

► **Binary encoding**

Применяется Label encoding.

Полученные номера переводятся в *двоичную систему* счисления и двоичные числа разбиваются на столбцы.

Минусы: Полученные признаки могут коррелировать.

► **Mean encoding** (Target encoding)

Заменяем категорию на *ср. значение* или другую статистику таргета у объектов, имеющих данную категорию.

Является очень мощным методом работы с кат. признаками.

Разберем подробнее.



Mean Encoding: какие статистики выбирать?

Бинарная классификация

- ▶ Частота класса 1
- ▶ Логиты
- ▶ Кол-во объектов класса 1
- ▶ Разница кол-ва объектов между классами

Многоклассовая классификация

Для кат. признака введем K признаков, где K — число классов.

k -ый признак строится по таргету вида $I\{Y_i = k\}$.

Регрессия

- ▶ Среднее
- ▶ Дисперсия
- ▶ Квантили
- ▶ Максимум
- ▶ Распределение по бинам

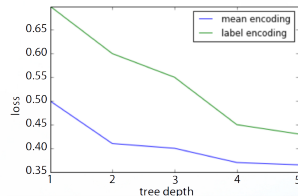


Mean Encoding

Модели на основе решающих деревьев

Деревьям трудно работать с кат. признаками с большим кол-вом уникальных значений: нужна большая глубина.

Mean encoding решает эту проблему: меньшая ошибка при меньшей глубине.



Проблемы

- ▶ Статистики по обучающей выборке, не всегда верны для теста.
Например, если количество объектов в категории мало, то оценка статистики будет очень шумной.
- ▶ При подсчете статистики используем таргет
⇒ при обучении на объекте x_i у модели есть информация о Y_i .

Эти проблемы вызывают **переобучение** модели.



Работа с признаками

Числовые признаки

Дата/время и координаты

Порядковые и категориальные признаки

Масштабирование

Отбор признаков



Масштабирование

Масштабирование — приведение признаков к единому масштабу.

Важно для

- ▶ линейных моделей с регуляризацией
- ▶ метрических моделей
- ▶ градиентного спуска (как следствие для нейросетей)

Не важно для решающих деревьев.

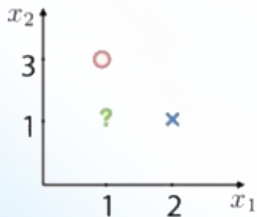
Почему?

- ▶ Регуляризация имеет тенденцию штрафовать параметры при признаках меньшего масштаба.
- ▶ Начинают учитываться только крупномасштабные признаки.

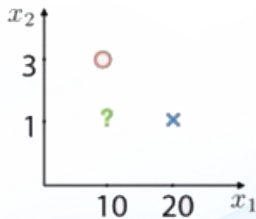


Масштабирование

Как классифицируется "?" методом ближайшего соседа?



Как "x"



Как "o"

Однако поменялся только масштаб!



Масштабирование

► Стандартизация

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_{ij} \quad \sigma_j = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_{ij} - \mu_j)^2} \quad \tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}$$

Чаще применяется для линейных моделей и нейросетей

► Min-max нормализация

Масштабируем на отрезок $[0, 1]$:

$$m_j = \min(x_{1j}, \dots, x_{nj}) \quad M_j = \max(x_{1j}, \dots, x_{nj})$$

$$\tilde{x}_{ij} = \frac{x_{ij} - m_j}{M_j - m_j}$$

Чаще применяется для метрических моделей



Масштабирование

► Нормализация средним

$$m_j = \min(x_{1j}, \dots, x_{nj})$$

$$M_j = \max(x_{1j}, \dots, x_{nj})$$

$$\mu_j = \frac{1}{n} \sum_{i=1}^n x_{ij}$$

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{M_j - m_j}$$

► Нормализация

$$\tilde{x}_i = \frac{x_i}{\|x_i\|^2}$$

Замечание:

Масштабируем и тест, и трейн одинаково.

Все статистики подбираем по трейну!



Работа с признаками

Числовые признаки

Дата/время и координаты

Порядковые и категориальные признаки

Масштабирование

Отбор признаков



Зачем отбирать признаки?

Ускорение модели

- ▶ Чем больше признаков, тем сложнее модель.
- ▶ Чем сложнее модель, тем дольше она вычисляет прогнозы и обучается.
- ▶ В некоторых задачах могут быть жесткие ограничения на скорость работы и обучения.
Например, в онлайн-моделях.



Одномерный отбор признаков

Принцип

- ▶ Измеряем связь (информативность) каждого признака с целевой переменной отдельно.
- ▶ Отбираем лучшие по информативности.

Как оценить информативность?

1. Важность признаков
2. Корреляции
3. Качество моделей, обученных по каждому из признаков отдельно



Важность признаков в общем случае

Permutation feature importance

1. Обучим модель и измерим метрику на валидации.
2. Для одного выбранного признака перемешаем все его значения в датасете, на котором до этого измерили метрику.
3. Измерим метрику на видоизмененном датасете.
4. Определим важность данного признака как *разницу между исходным и новым значением метрики*.
5. Сделаем пункты 2-4 для всех признаков.

Плюсы:

- ▶ Подходит для любых моделей.
- ▶ Требуется одного обучения модели.
- ▶ Использует тестовое множество и является более надежным, чем MDI для деревьев.

Минусы:

- ▶ Более вычислительно затратно, чем MDI для деревьев.
- ▶ Переоценивает важность для скоррелированных признаков. (Strobl et al (2008))



Важность признаков в общем случае

Drop Column feature importance

Сравним 2 модели:

- ▶ Модель, обученная на датасете со всеми признаками
- ▶ Модель, обученная на данных без одного признака.

Важность этого признака — разница метрик на тесте/валидации для этих моделей.

Плюсы:

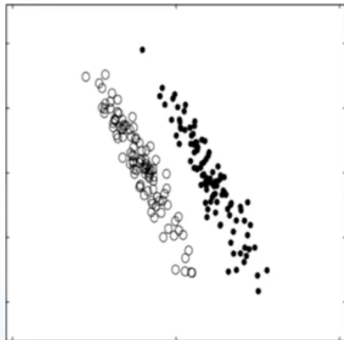
- ▶ Самая точная важность признаков.
- ▶ Подходит для любых моделей.

Минусы:

- ▶ Вычислительно сложно.
Требует обучения большого количества разных моделей.

Одномерный отбор признаков

Проблема: сложные закономерности



По двум признакам можно идеально разделить классы.

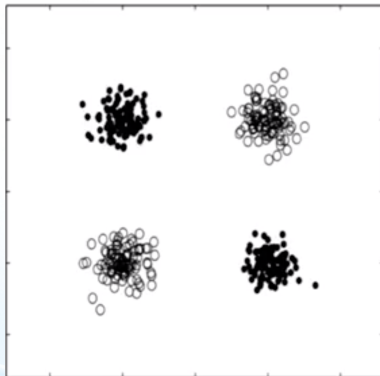
По x_1 данные можно как-то разделить $\Rightarrow x_1$ — информативный.

По x_2 данные нельзя разделить $\Rightarrow x_2$ будет неинформативным.

\Rightarrow Останется только признак x_1 , сильно теряем в качестве.

Одномерный отбор признаков

Проблема: сложные закономерности



По двум признакам можно идеально разделить классы.

При одномерном отборе x_1 и x_2 будут неинформативными.



Перебор признаков

Принцип

- ▶ Каким-то методом перебираем комбинации признаков
- ▶ Для каждой комбинации обучаем модель
- ▶ Выбираем комбинацию, дающую лучшую модель

Полный перебор

Пробуем все подмножества признаков и выбираем лучшее.

Свойства:

- ▶ Находит точное решение.
- ▶ Перебирает 2^d вариантов.
⇒ Подходит только для малого числа признаков.



Жадное добавление

Пусть F_t — множество информативных признаков на итерации t .

Принцип:

1. Сначала F_0 — пустое
2. Находим признак x_j , при добавлении которого к F_{t-1} получим наименьшую ошибку модели
3. Повторяем до тех пор, пока ошибка уменьшается

Плюсы:

- ▶ Работает достаточно быстро — требует d итераций.
На каждой итерации t происходит обучение $(d - t)$ моделей.
 \Rightarrow обучается всего $\frac{d(d-1)}{2}$ моделей.

Минусы:

- ▶ Слишком жадно.
После добавления признака в F_t он там навсегда останется.
Нет возможности убрать признак после добавления.



Add-Del

- ▶ Жадное добавление.
Добавляем по одному признаку пока ошибка уменьшается.
- ▶ Жадное удаление.
Удаляем по одному признаку пока ошибка уменьшается.
- ▶ Повторяем стадии добавления и удаления,
пока ошибка уменьшается.

Может исправлять ошибки, сделанные в процессе перебора ранее.



Отбор на основе моделей

Линейные модели

$$y(x) = \sum_{j=1}^d \theta_j x_j$$

- ▶ Если признаки отмасштабированы:
Веса можно использовать как показатели информативности.
Чем больше $|\theta_j|$, тем больший вклад вносит признак x_j .
- ▶ Если признаки не отмасштабированы:
Веса нельзя использовать как показатели информативности.

Для повышения числа нулевых весов — L_1 -регуляризация



ВСЁ!