Машинное обучение ФИВТ

DS-поток

Лекция шестая

Работа с признаками, отбор признаков, заполнение пропусков

Работа с признаками

Числовые признаки

Порядковые и категориальные признаки

Mean Encoding

Дата/время и координаты

Масштабирование

Мультиколлиниарность

Ранг

Заменяем значения на их индексы в вариационном ряду:

$rank([-100, 0, 1e5]) = [0, 1, 2]$

$rank([1000, 1, 10]) = [2, 0, 1]$

Повторы обычно заменяются на средний ранг.

Для применения к тесту сохраним отображение из train-а в ранги и

- Возьмем ранг ближайшего объекта из train.
- Средний ранг по ближайшим объектам из выборки.
- Интерполируем ранги.

Подвигаем выборсы к остальным объектам,

$\Rightarrow$ они перестают вносить большой вклад в модель.

Иногда хорошо работает для KNN, лин. моделей, нейросетей,

особенно есть нет времени разбираться с выбросами.

Трансформации

- Логарифмическая

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

- Возведение в степень

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

- Пребразование Бокса-Кокса

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

Объекты с бОльшими значениями признаков становятся ближе к остальным объектам.

При отрицательных значениях признака нужно произвести сдвиг.

Особенно хорошо работают для нейростей.

Трансформации

Логарифмическая : $\widetilde{x} = \ln(x)$

1. После логарифмирования мултипликативный признак станет аддитивным.

   Рассмотрим лин. регрессию с одним признаком: $\widehat{y}(x) = \theta x + \theta_0$

   - Пусть x показывает во сколько раз увеличились цены.

     $\Rightarrow x$ — *мултипликативный признак*.

     Тогда $\widehat{y_1} = \theta x_1 + \theta_0 \qquad \widehat{y_2} = \theta x_2 + \theta_0$

     $$\Rightarrow \widehat{y_1} - \widehat{y_2} = \theta(x_1 - x_2)$$

     Но рассматривать $x_1 - x_2$ нелогично, лучше рассматривать x_1/x_2.

   - Пусть x показывает на сколько увеличились цены.

     $\Rightarrow x$ — *аддитивный признак*.

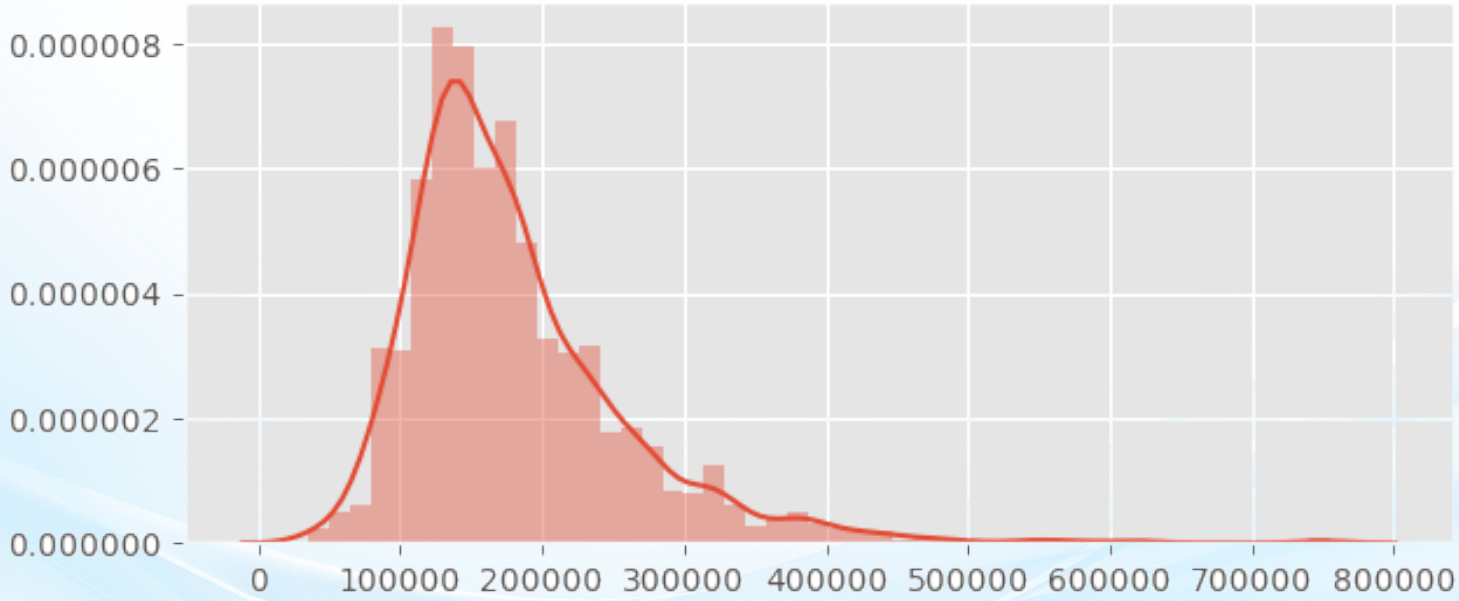
     Здесь рассматривать $x_1 - x_2$ вполне логично.

Трансформации

Логарифмическая : $\widetilde{x} = \ln(x)$

2. Преобразует искаженное распределение ближе к нормальному.

Исходное распределение.

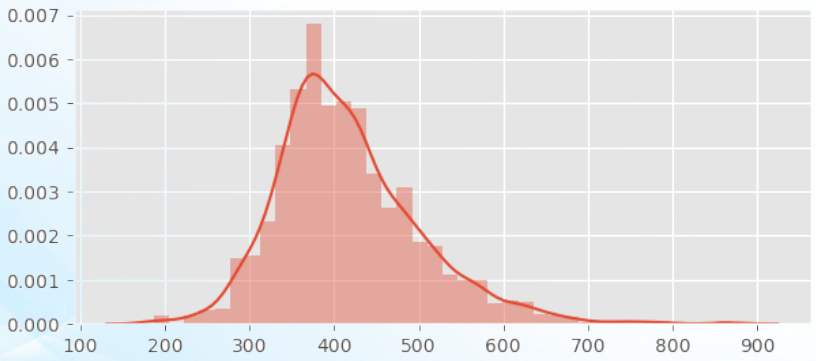


Трансформации

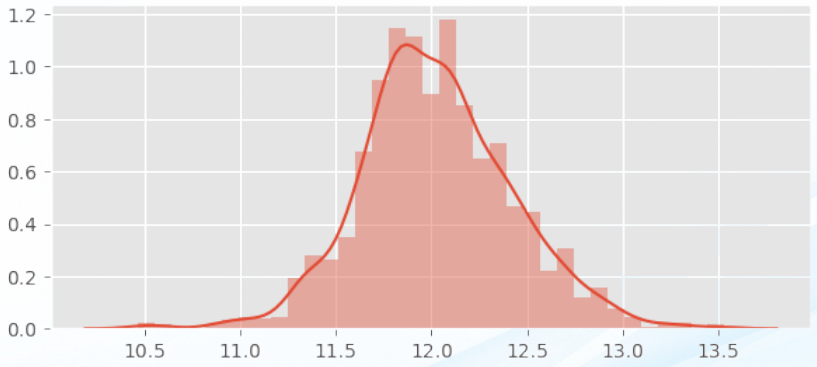
Логарифмическая : $\widetilde{x} = \ln(x)$

2. Преобразует искаженное распределение ближе к нормальному.

$\sqrt{x}$



$\log(x)$



Генерация признаков

цена	целая часть	дробная часть
0.99	0	0.99
2.49	2	0.49
1	1	0
9.99	9	0.99

Разделеним на целую и вещественную часть.

Такие преобразования очень часто делают для цены,

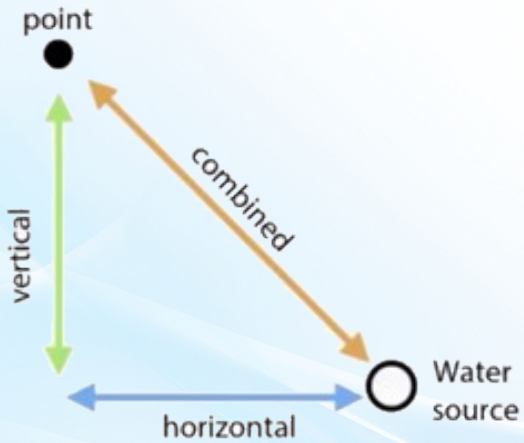
так как это отражает восприятние цены человеком.

Генерация признаков

Квадратная площадь : $55 m^2$

Цена : 107000 $

Цена for 1 m^2 : 10700 $ / 55 m^2



Расстояние по вертикали : 3 м

Расстояние по горизонтали : 2 м

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

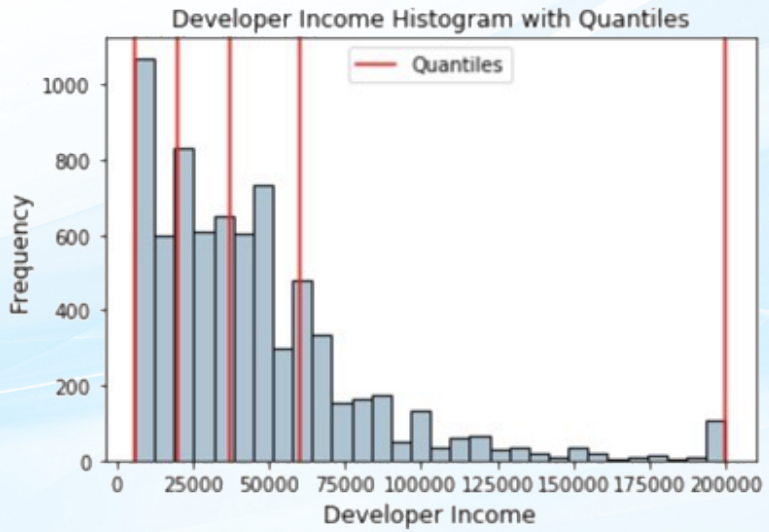
  Выбор точек разбиения вручную или равномерно

- Adaptive Binning

  Выбор точек разбиения в зависимости от выборки.

Пример:

Чтобы вероятностная масса в каждом бине была одинаковой.



Работа с признаками

Числовые признаки

Порядковые и категориальные признаки

Mean Encoding

Дата/время и координаты

Масштабирование

Мультиколлиниарность

Ординальные признаки (порядковые)

Ординальные признаки — признаки с упорядоченными состояниями.

Состояния сравнимы между собой, но расстояния между ними не определены.

Примеры:

- Класс билета : 1, 2, 3
- Образование : kindergarten, school, undergraduate, bachelor, master
- Оценка : A, B, C, D, F

Как работать?

- Label encoding

  Каждому состоянию сопоставляем число.

  Порядок состояний должен сохраниться.

Категориальные признаки

Категориальные признаки — признаки с несравнимыми значениями.

Категория - объекты с одним значением данного признака.

Примеры:

- Город : Москва, Долгопрудный, Клин
- 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.

- Count encoding и Frequency encoding

  Заменяем категорию на ее кол-во/частоту в обучении.

  Замечание :

  Если частоты у категорий похожи, то они будут неразличимы.

  Поэтому можно использовать ранги частот.

Как работать с категориальными признаками?

- One-hot encoding

  Создается $N - 1$ новых (dummy) признаков, где N - кол-во категорий.

  Каждый i-ый признак - индикатор i-ой категории.

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 исчисления и двоичные числа разбиваются на столбцы.

  Минусы: Полученные признаки могут коррелировать.

- Mean encoding (Target encoding)

  Заменяем категорию на среднее значение (другую статистику) таргета у объектов, имеющих данную категорию.

  Является очень мощным методом работы с категориальными признаками.

Работа с признаками

Числовые признаки

Порядковые и категориальные признаки

Mean Encoding

Дата/время и координаты

Масштабирование

Мультиколлиниарность

Mean Encoding

Пример

Label encoding

id	job	job_label	target
1	Doctor	1	1
2	Doctor	1	0
3	Doctor	1	1
4	Doctor	1	0
5	Teacher	2	1
6	Teacher	2	1
7	Engineer	3	0
8	Engineer	3	1
9	Waiter	4	1
10	Driver	5	0

Mean encoding

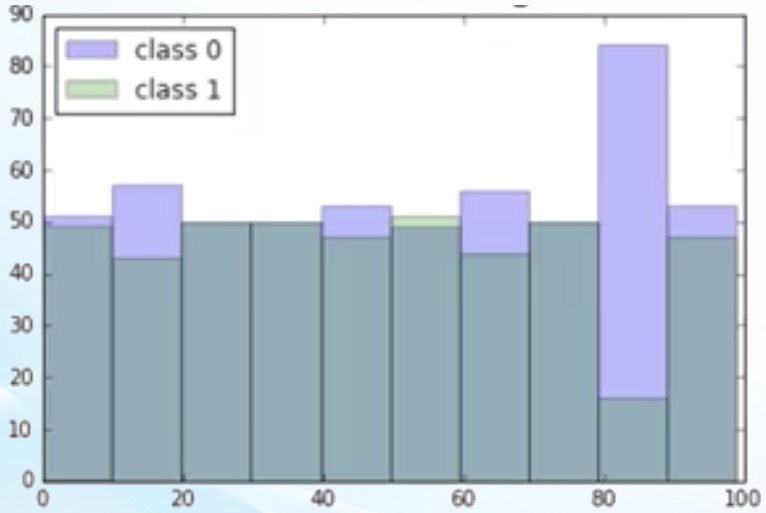
id	job	job_mean	target
1	Doctor	0,50	1
2	Doctor	0,50	0
3	Doctor	0,50	1
4	Doctor	0,50	0
5	Teacher	1	1
6	Teacher	1	1
7	Engineer	0,50	0
8	Engineer	0,50	1
9	Waiter	1	1
10	Driver	0	0

Mean Encoding

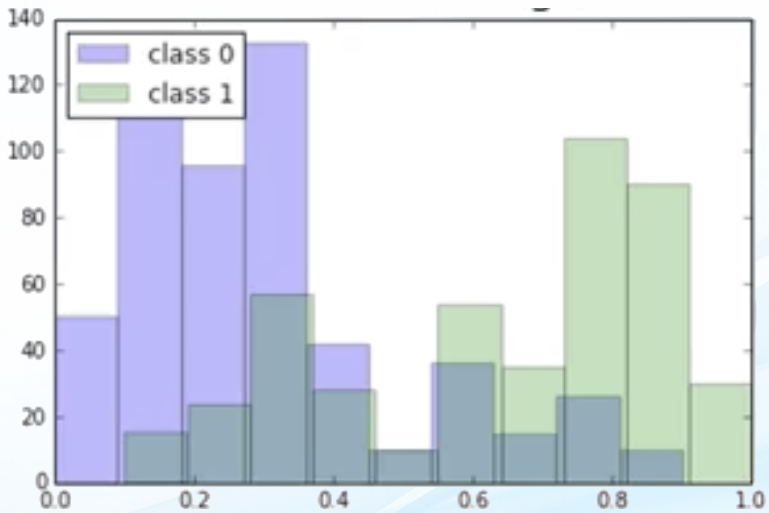
Будем пока рассматривать случай бинарной классификации.

Почему Mean Encoding работает?

Label encoding



Mean encoding



Посмотрим на гистограммы признака для класса 0 и класса 1.

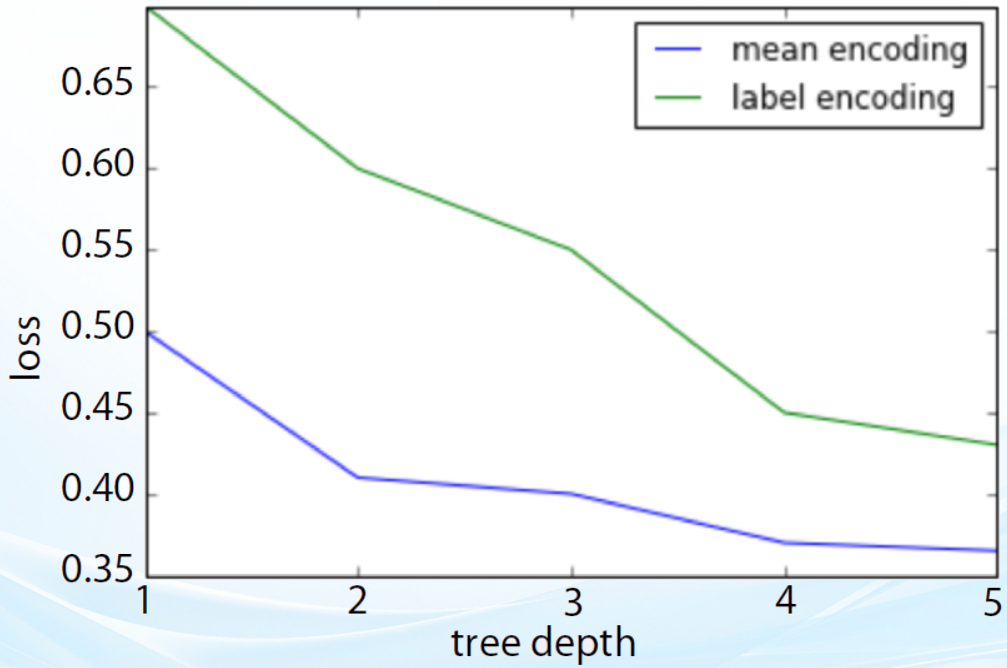
В случае Label encoding получаем случайную картину.

В случае Mean encoding классы выглядят более разделимыми.

Mean Encoding

Почему Mean Encoding работет?

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



Mean encoding решает эту проблему.

Достигается меньшая ошибка при меньшей глубине.

Mean Encoding

Какой бывает Mean Encoding?

Для задачи бинарной классификации:

- Частота:

  $$mean(target) = \frac{\#\{class\ 1\}}{\#\{class\ 0\}+\#\{class\ 1\}}$$

- Логарифмические шансы

  $$\ln(\frac{\#\{class\ 1\}}{\#\{class\ 0\}}) \cdot 100$$

- Кол-во объектов класса 1 : $\#\{class\ 1\}$

- Разница кол-ва объектов из класса 1 и 0:

  $$\#\{class\ 1\} - \#\{class\ 0\}$$

- Любой ваш вариант

Mean Encoding

Проблемы

- Статистики, которые нашли в обучении,

  не всегда верны для теста.

  Если количество объектов в категории мало, то оценка

  статистики будет очень шумной.

- При подсчете статистики используем таргет

  $\Rightarrow$ при обучении на объекте x_i у модели есть информация о Y_i.

Эти проблемы вызывают переобучение модели.

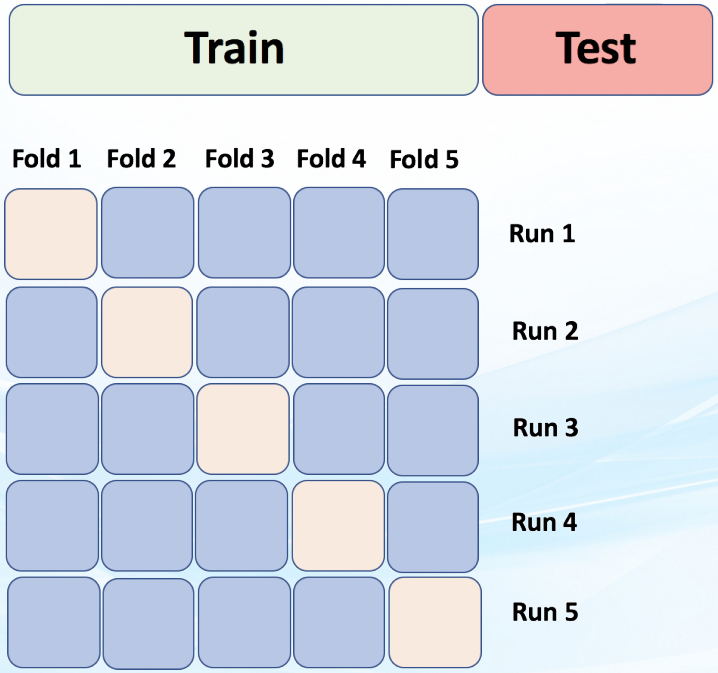
Регуляризация

- CV loop

  1. Разбиваем данные на k фолдов.
  2. Для получения статистики для k-ого фолда используем таргеты всех фолдов, кроме k-ого.

     Для подсчета статистики для теста используется весь train.

$\Rightarrow$ Для объекта x_i не используем его таргет y_i в статистике.



Регуляризация

- CV loop

  В некоторых случаях все равно может произойти переобучение.

  Рассмотрим пример Leave-One-Out.

	feature	feature_mean	target
0	Moscow	0.50	0
1	Moscow	0.25	1
2	Moscow	0.25	1
3	Moscow	0.50	0
4	Moscow	0.50	0

  По новому признаку можно однозначно восстановить ответ.
  Однако для теста значение статистики для всех объектов этой категории $= 0.4$, т.е. восстановить ответ уже нельзя.

  $\Rightarrow$ утечка таргета всё таки есть.

Регуляризация

- Сглаживание

  Пусть $\overline{y_c}$ — среднее значение таргета в категории c в обучении.

  n_c — количество объектов категории c в обучающей выборке.

  Заменим значение кат. признака на

$$S_c = \frac{\overline{y_c} \cdot n_c + \overline{y} \cdot \alpha}{n_c + \alpha}$$

  $\alpha = 0 \Rightarrow$ нет регуляризации.

  $\alpha \to \infty \Rightarrow$ статистика стремится к глобальному среднему.

Регуляризация

- Expanding mean

  Зафиксируем некоторый порядок объектов в трейне.

  Для подсчета статистики для x_i используются только $y_1, ..., y_{i-1}$.

  Плюсы:

  - Самое маленькая утечка таргета среди всех методов.
  - Нет гиперпараметров.
  - Используется в CatBoost для обработки категориальных признаков.

Mean Encoding

- Регрессия

  Больший набор статистик :

  выборочная дисперсия, квантили, максимум,

  распределение по корзинам.

  В остальном подход аналогичен бинарной классификации.

- Многоклассовая классификация.

  Для каждого признака вводем K енкодеров,

  где K - число классов.

  k-ый енкодер работает с таргетом вида $I\{Y_i = k\}$.

  В итоге имеем K новых признаков.

Работа с признаками

Числовые признаки

Порядковые и категориальные признаки

Mean Encoding

Дата/время и координаты

Масштабирование

Мультиколлиниарность

Дата и время

Пусть имеется признак, показывающий время события.

Полезно добавить дополнительные признаки:

- Время сейчас:
  Год, сезон, день месяца, день недели, час, минуты, секунды, праздничный ли день и какой праздник.
- Время с определенного события:
  Считаем время, прошедшее с некоторого момента.
  - Момент одинаковый для всех строк.
    Например, 17 сентября 1951 года.
  - Момент зависит от строки.
    Примеры :
    Кол-во дней с прошлых выходных.
    Кол-во дней с последней покупки.
- Синус от времени с периодом кратным длине сезона/дня.

Замечание: Не забывайте, что такие признаки как день недели являются категориальными!

Координаты

Пусть имеется признак, показывающий местоположение.

- Если есть доп данные.

  Можно добавить расстояния до ближайшего магазина, больницы, школы и прочего.

Координаты

- Если доп. данных нет.

  Придумаем местоположения сами по имеющимся данным.

  Допустим, у нас данные квартир.

  - 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.

  Пример статистик : кол-во квартир, средняя цена квартиры.

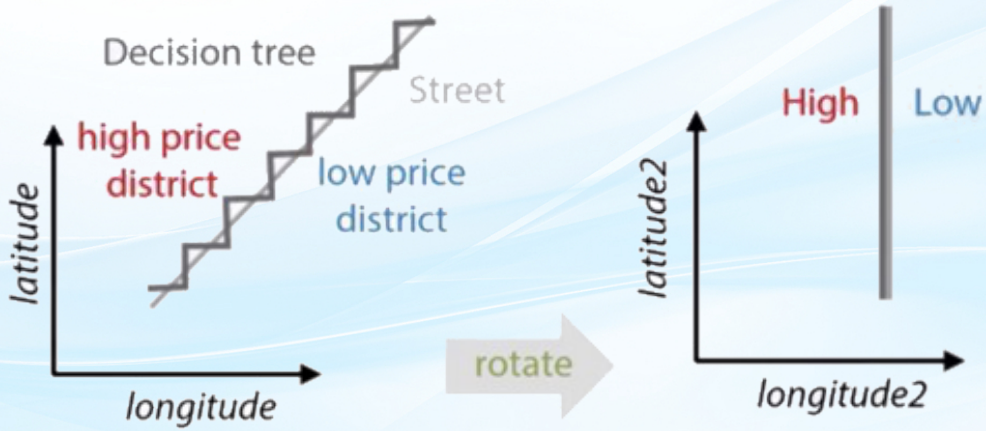
- Поворот координат

  Повернем координаты и возьмем это как новые признаки.

  Неясно как лучше повернуть

  $\Rightarrow$ можно сделать много разных поворотов.

  Полезно для tree-based методов.



Работа с признаками

Числовые признаки

Порядковые и категориальные признаки

Mean Encoding

Дата/время и координаты

Масштабирование

Мультиколлиниарность

Масштабирование

Масштабирование - приведение признаков к единому масштабу.

Важно для

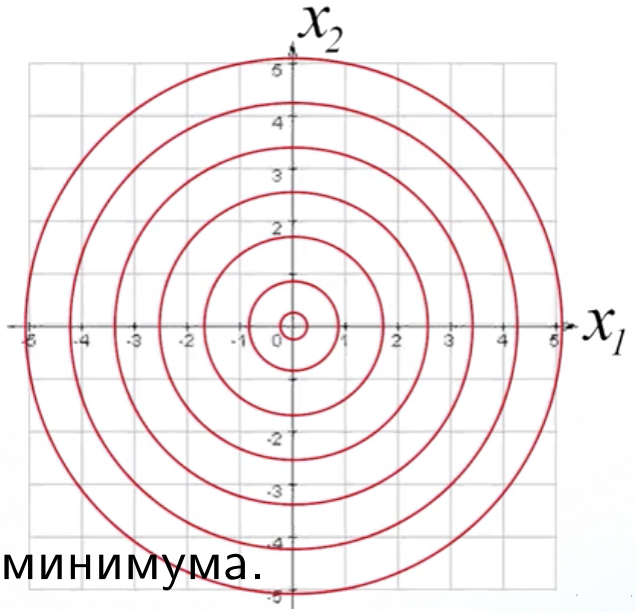
- линейных моделей с регуляризацией
- метрических моделей
- градиентного спуска (как следствие для нейросетей)

Не важно для решающих деревьев.

Почему?

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

Масштабирование



Пример:

Градиентный спуск для $w_1^2 + w_2^2 \to \min\limits_{w_1, w_2}$

$$-\frac{\partial \mathcal{L}}{\partial w}(1,1) = (-2,-2)$$

Вектор антиградиента проходит через точку минимума.

Градиентный спуск для

$w_1^2 + 100 w_2^2 \to \min\limits_{w_1, w_2}$

$$-\frac{\partial \mathcal{L}}{\partial w}(1,1) = (-2,-200)$$

Вектор антиградиента направлен практически вниз

и проходит мимо точки минимума.

Масштабирование

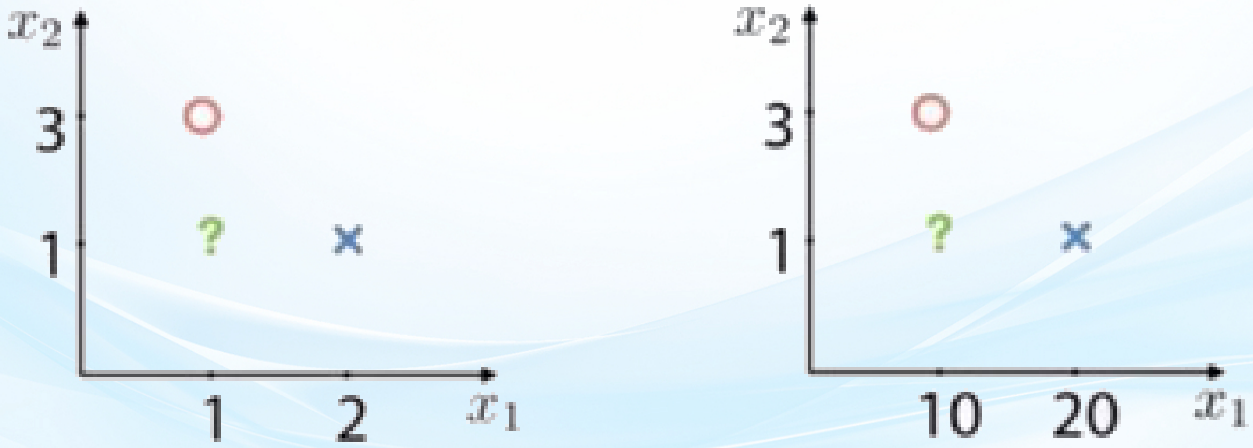
Пример:

KNN на следующих двух картинках даст разный ответ.

В первом случае ? классифицируется как x.

На второй как o.

Однако поменялся только масштаб!



Масштабирование

- Standardization

  Убираем различия в сдвигах и масштабах.

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

  Чаще применяется для линейных моделей и нейросетей

- Min-max normalization (rescaling)

  Масштабируем на отрезок $[0, 1]$:

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

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

  Чаще применяется для метрических моделей

Масштабирование

- Mean normalization

$$m_j = min(x_1^j, ..., x_n^j)$$
$$M_j = max(x_1^j, ..., x_n^j)$$
$$avg_j = average(x_1^j, ..., x_n^j)$$

$$\widetilde{x}_i^j = \frac{x_i^j - avg_j}{M_j - m_j}$$

- Normalization

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

Замечание:

Масштабируем и тест, и трейн одинаково.

Все статистики подбираем по трейну!

Работа с признаками

Числовые признаки

Порядковые и категориальные признаки

Mean Encoding

Дата/время и координаты

Масштабирование

Мультиколлиниарность

Мультиколлинеарность

Мультиколлинеарность — корреляционная взаимосвязь признаков.

Рассмотрим задачу линейной регрессии :

$$Y = X\theta + \varepsilon$$

$$\widehat{\theta} = (X^T X)^{-1} X^T Y$$

Мультиколлинеарность означает, что среди строк X есть коррелирующие.

Мультиколлинеарность

Последствия:

- Матрица $X^T X$ становиться близка к вырожденной.
- Если $X^T X$ близка к вырожденной, то коэффициенты будут неустойчивы.
- Большой разброс значений коэффициентов.

  Становится невозможно судить о важности признаков
- Опасность переобучения, так как снижается обобщающая способность модели.

Решения:

- Посмотреть на графики корреляций.
- Воспользоваться методами отбора признаков.

Отбор признаков

Отбор признаков

Зачем отбирать признаки?

Одномерный отбор признаков

Перебор признаков

Зачем отбирать признаки?

Шумовые признаки - признаки не связанные с таргетом.

Пример:

Добавим к нашим признакам 100 шумовых признаков из $\mathcal{N}(0, 1)$:

$$X^1 \sim \mathcal{N}(0,1), \quad X^2 \sim \mathcal{N}(0,1), ..., \quad X^{100} \sim \mathcal{N}(0,1)$$

Новых признаков много.

$\Rightarrow$ С большой вероятностью хотя бы один немного коррелирует

с таргетом на обуч. выборке.

$\Rightarrow$ Модель может решить, что он важный и использовать его.

На других данных такой корреляции уже не будет.

$\Rightarrow$ Качество будет страдать.

Зачем отбирать признаки?

Много признаков

- Пусть есть 1000 признаков.
- Обучаем решающее дерево.
- Чтобы учесть каждый признак хотя бы по одному разу,

  нужно дерево глубины ≥ 10.

  У такого дерева минимум 1000 листьев.
- В каждый лист должно попасть достаточное число объектов,

  иначе - риск переобучения.

  $\Rightarrow$ Должно быть много объектов.

Замечание: Если хочется учесть каждый признак больше,

чем один раз, то дерево должно быть еще глубже.

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.

Отбор признаков

Зачем отбирать признаки?

Одномерный отбор признаков

Перебор признаков

Одномерный отбор признаков

Одномерный отбор работает по следующему принципу:

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

  или

  Отбираем признаки, у которых информативность выше порога.

Одномерный отбор признаков

Коэффициент корреляции

$$R_j = \widehat{corr}(X^j, Y) = \frac{\sum_{i=1}^{n}(X_i^j - \overline{X}^j)(Y_i - \overline{Y})}{\sqrt{\sum_{i=1}^{n}(X_i^j - \overline{X}^j)^2 \sum_{i=1}^{n}(Y_i - \overline{Y})^2}}$$

Одномерный отбор признаков

Коэффициент корреляции Какой?

$$R_j = corr(x^j, y) = \frac{\sum_{i=1}^{n}(x_i^j - \overline{x}^j)(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i^j - \overline{x}^j)^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

Одномерный отбор признаков

Коэффициент корреляции Пирсона

$$R_j = corr(x^j, y) = \frac{\sum_{i=1}^{n}(x_i^j - \overline{x}^j)(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i^j - \overline{x}^j)^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

Одномерный отбор признаков

Коэффициент корреляции Пирсона

$$R_j = corr(x^j, y) = \frac{\sum_{i=1}^{n}(x_i^j - \overline{x}^j)(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i^j - \overline{x}^j)^2 \sum_{i=1}^{n}(y_i - \overline{y})^2}}$$

Возьмем коэффициент корреляции в качестве информативности.

Чем больше $|R_j|$, тем информативнее признак x^j.

- Учитывает только линейную зависимость.
- Для вещественных признаков и ответов.
- Для бинарных признаков и ответов можно использовать значения $\{-1, 1\}$

Одномерный отбор признаков

Другие коэффициенты

- Вещественные признак и отклик.

  Корреляции Пирсона, Спирмена, Кендала.

- Бинарные признак и отклик.

  Коэффициент ассоциации и коэффициент контингенции.

- Категориальные признак и отклик.

  Хи-квадрат, коэф-т Крамера.

- Вещественные vs Категориальные

  дискритизируем вещественную переменную.

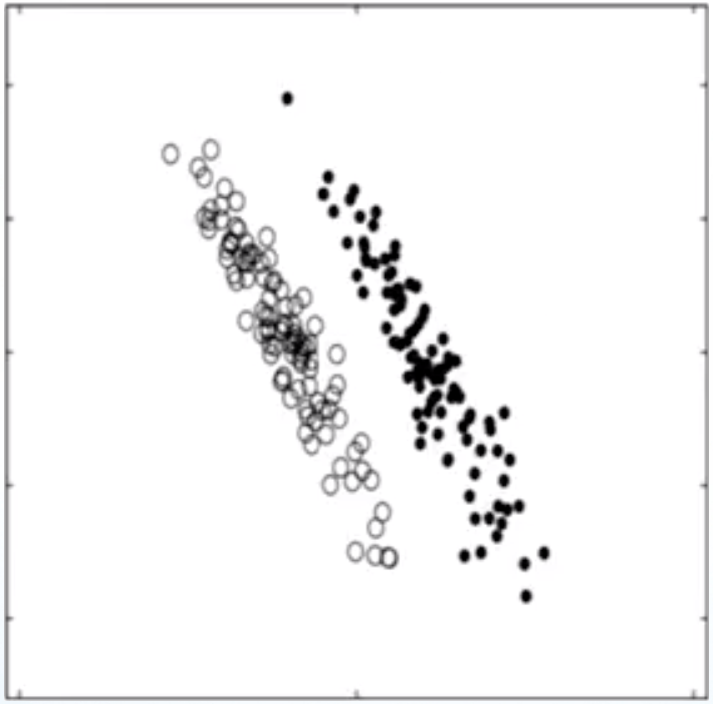
Одномерный отбор признаков

Построение модели для каждого признака

- Для каждого признака строим модель над одним признаком.
- Измеряем качество модели.
- Сортируем все признаки по качеству.
- Выбираем лучшие признаки.

Одномерный отбор признаков

Проблема : Не учитываение сложных закономерностей



По двум признакам можно идеально разделить классы.

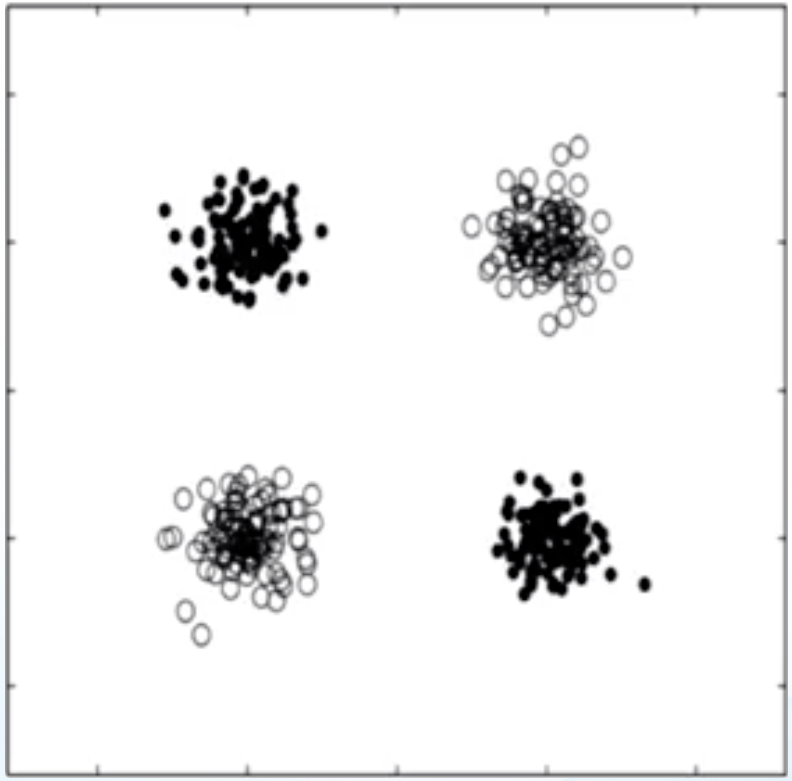
По x_1 данные можно как-то разделить $\Rightarrow$ x_1 — информативный.

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 : Не учитываение сложных закономерностей



По двум признакам можно идеально разделить классы (например, деревом).

При одномерном отборе и x^1, и x^2 будут неинформативными.

Отбор признаков

Зачем отбирать признаки?

Одномерный отбор признаков

Перебор признаков

Перебор признаков

Принцип:

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

Перебор признаков

Полный перебор

Пробуем все подмножества признаков и выбираем лучшее.

Плюсы:

- Находит точное решение.

Минусы:

- Перебирает 2^d вариантов, где d - кол-во признаков.

  $\Rightarrow$ Подходит только для малого числа признаков.

Перебор признаков

Жадное добавление

Пусть F_t - множество информативных признаков на итерации t.

$Q(F)$ - ошибка модели, обученной на признаках из мн-ва F.

1. $F_0 = \{\}$
2. Повторить для t от 1 до d:
3. Находим признак x^j, при добавлении которого к F_{t-1} получим наименьшую ошибку модели:

   $$j = \arg\min_j Q(F_{t-1} \cup x^j)$$
4. $F_t = F_{t-1} \cup x^j$
5. пока ошибка уменьшается

Перебор признаков

Жадное добавление

Плюсы:

- Работает достаточно быстро.

  Требует всего 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 J_t он там навсегда останется.

  Нет возможности убрать признак после добавления.

Перебор признаков

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

$$a(x) = \sum_{j=1}^{d} w_j x^j + w_0$$

- Если признаки отмасштабированы:

  Веса можно использовать как показатели информативности.

  Чем больше $|w_j|$, 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

Решающее дерево — разбирали ранее.

Случайный лес — разбирали ранее.

Заполнение пропусков

Что может быть пропуском?

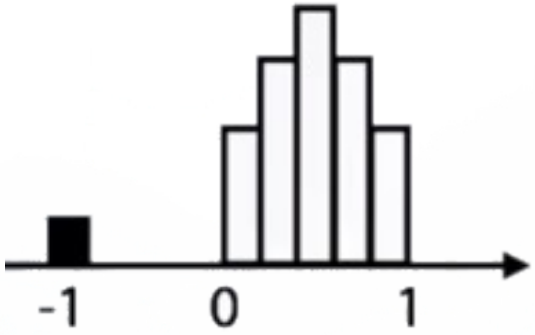
Пропуском может быть :

- NaN
- Пустая строка
- -
- -1
- 1000000
- -99999
- 999

Как понять что является пропуском?

Посмотрим на гистограмму.

Все пропущенные значения были заменены на -1.



А что произошло здесь?

Пропущенные значения были заменены на среднее значение признака.

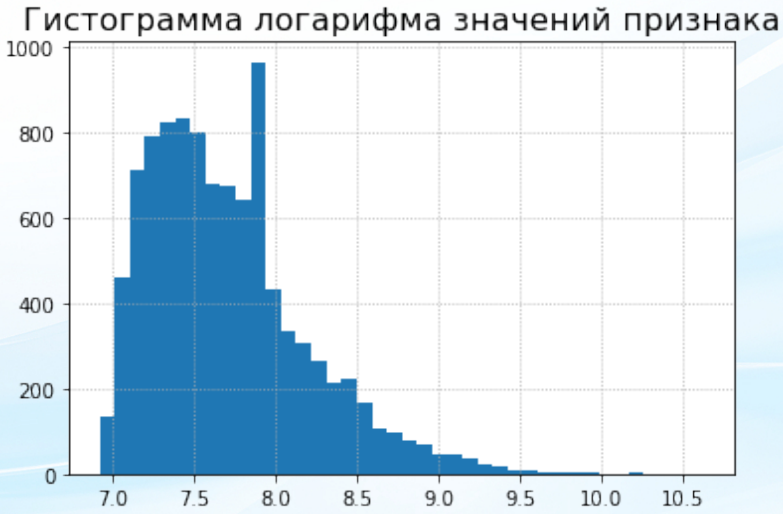
Как понять, что является пропуском?

Что можно понять здесь?

Хмм, ничего не понятно...

Прологарифмируем значения признака.

Теперь пропуски отчетливо видны.



Какие бывают пропуски?

Рассмотрим датасет из двух признаков:

x^1 - болеет ли студент, x^2 - оценка за контрольную по статистике.

Значение x^2 пропущено у некоторого объекта.

- Missing Completely at Random (случайный пропуск)
  Событие {признак пропущен} не зависит ни от других признаков, ни от значения пропущенного признака.

  Пример: Учитель поленился проставить оценку в таблицу.

- Missing at Random (неслучайный пропуск)
  Событие {признак пропущен} не зависит от значения пропущ. признака, но зависит от значения других признаков.

  Пример: Студент болел, поэтому не пошел на контрольную.

- Missing not at Random (неслучайный пропуск)
  Событие {признак пропущен} зависит от значения пропущенного признака.

  Пример: Студент не знает статистику и поэтому не пошел на кр.

Что делать с пропусками?

Неслучайные пропуски:

- Завести отдельный бинарный признак : $I\{x^j - \text{пропущено}\}$.
- Для категориальных признаков завести новую категорию.
- Закодировать каким-то невероятным значением.

  Хорошо работает для tree-based моделей

  т.к. позволяет сделать разделение на пропущенные и непропущенные.
- Использовать модели, умеющие работать с пропусками.

Что делать с пропусками?

Случайные пропуски:

- Удалить все строки или столбцы с пропущенными значениями.
- Использовать наиболее вероятное значение признака.

  Среднее или медиана для вещественных переменных,
  для категориальных - самое частое значение.
  Неплохо работает на линейных моделей и нейросетей.
- Обучить модель предсказывать пропущенные значения.

  Самые популярные варианты - Linear Regression and KNN.
- Multiple Imputation

  Обучить несколько разных моделей предсказывать пропуски.
  Усреднить результаты разных моделей.

  Обычно берется 5-10 похожих моделей.
- Использовать модели, умеющие работать с пропусками.

Заполнение пропусков

Замечание: Нужно быть очень аккуратным с заменой пропущенных значений до feature generation.

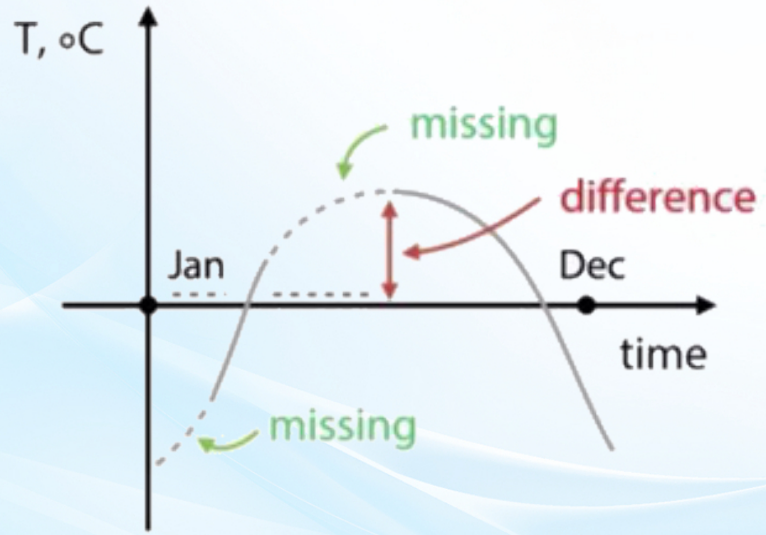
Пример:

Заменим пропуски средним значением температуры за год.

Заполнение пропусков

Замечание: Нужно быть очень аккуратным с заменой пропущенных значений до feature generation.

Пример:



Заменим пропуски средним значением температуры за год.

Добавим признак : разница значений температуры с предыдущим днем.

Заполнение пропусков

Замечание: Нужно быть очень аккуратным с заменой пропущенных значений до feature generation.

Пример:

categorical _feature	numeric _feature	numeric_ feature_filled	categorical _encoded
A	1	1	1.5
A	4	4	1.5
A	2	2	1.5
A	-1	-1	1.5
B	9	9	-495
B	NaN	-999	-495

Заменим пропуски на что-то out-of-range.

Закодируем категориальный признак средним значением вещественного признака.

При подсчете среднего использовали out-of-range значения(

Решение:

Игнорировать пропущенные значения при подсчете средних и проч.


ВСЁ!