Машинное обучение

DS-поток

Лекция 5

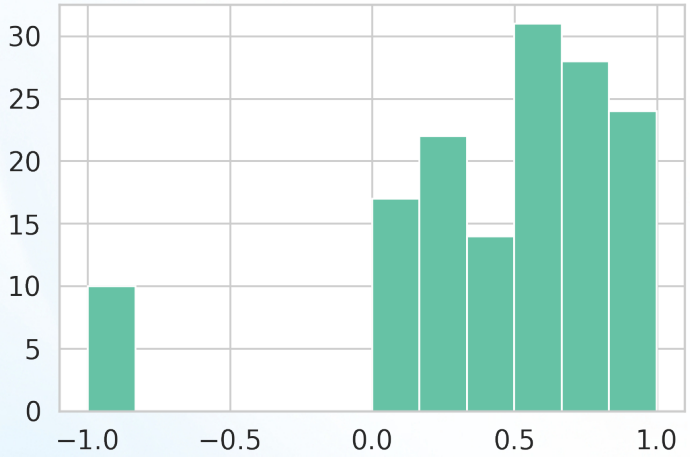
Обработка пропусков в данных

Что может быть пропуском?

Пропуском может быть:

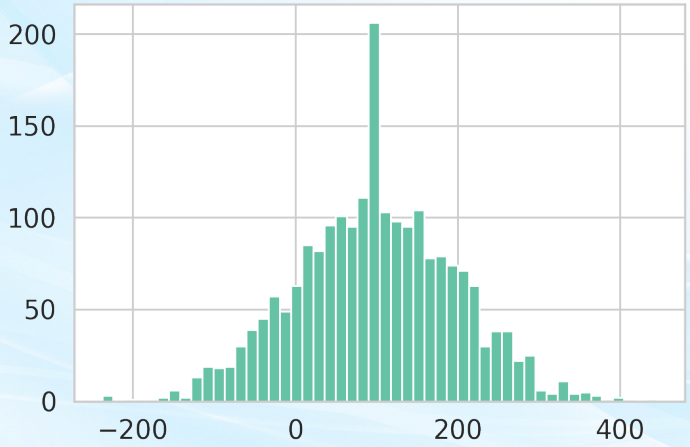
- NaN
- "nan"
- Пустая строка
- -
- ?
- -1
- 1000000
- -99999
- 999

Как понять что является пропуском?



Посмотрим на гистограмму.

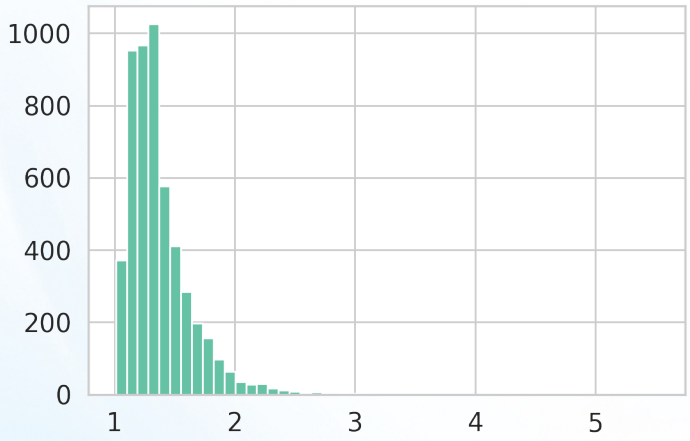
Все пропущенные значения заменены на -1.



А что произошло здесь?

Пропущенные значения заменены на среднее значение признака.

Как понять что является пропуском?



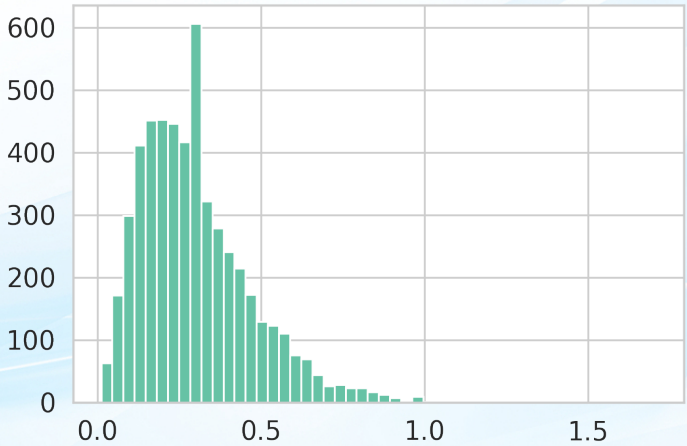
Что можно понять здесь?

Хмм, ничего не понятно...

Прологарифмируем

значения признака.

Теперь пропуски отчетливо видны.



Какие бывают пропуски?

Время	8:00	9:00	10:00	11:00	12:00
Температура возд.	21.4	22.1	**NaN**	24.2	25.5

Знаем: температура воздуха всегда есть :)

Возможные причины пропуска:

- Метеоролог был пьян.
  - События "метеоролог пьян" нет в датасете
    $\Longrightarrow$ *абсолютно случайный пропуск*.
  - Событие "метеоролог пьян" есть в датасете
    $\Longrightarrow$ *случайный пропуск*.
- Перегрелось оборудование
  $\Longrightarrow$ *неслучайный пропуск*.

Какие бывают пропуски?

- **Missing Completely at Random**

  Событие "признак пропущен" не зависит ни от других признаков, ни от значения пропущенного признака.

- **Missing at Random**

  Событие "признак пропущен" не зависит от значения пропущ. признака, но зависит от значения других признаков.

- **Missing not at Random**

  Событие "признак пропущен" зависит от значения пропущенного признака.

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 и KNN.
- Multiple Imputation — обучить несколько разных моделей

  предсказывать пропуски и усреднить их результаты.
- Использовать модели, умеющие работать с пропусками.

  Например, можно считать $X^T X$ и $X^T Y$ только по полным парам

$$\frac{1}{n}\left(X^T X\right)_{jk} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}\, x_{ik} \approx \frac{1}{n_{jk}}\sum_{i=1}^{n} x_{ij}\, x_{ik}\, I\{x_{ij} \text{ и } x_{ik} \text{ не пропущены}\},$$

$$\frac{1}{n}\left(X^T Y\right)_{jk} = \frac{1}{n}\sum_{i=1}^{n} x_{ij}\, y_i \approx \frac{1}{n_j}\sum_{i=1}^{n} x_{ij}\, y_i\, I\{x_{ij} \text{ не пропущено}\}.$$

где n_{jk} — число полных пар (x_{ij}, x_{ik}); n_j — число заполненных x_{ij}.

Что делать с пропусками? Неслучайные пропуски.

- Завести отдельный бинарный признак: $I\{x_j$ — пропущено$\}$.
- Для категориальных признаков завести новую категорию.
- Закодировать каким-то значением, не встречающимся в данных.

  Хорошо работает для моделей на основе деревьев
  т.к. позволяет сделать разделение на пропущенные и непропущенные.
- Использовать модели, умеющие работать с пропусками.

Можно ли их просто удалить?

Нет. Если NaN только для больших знач. T, то распр. будет другим.

Куда отнести Missing at Random?

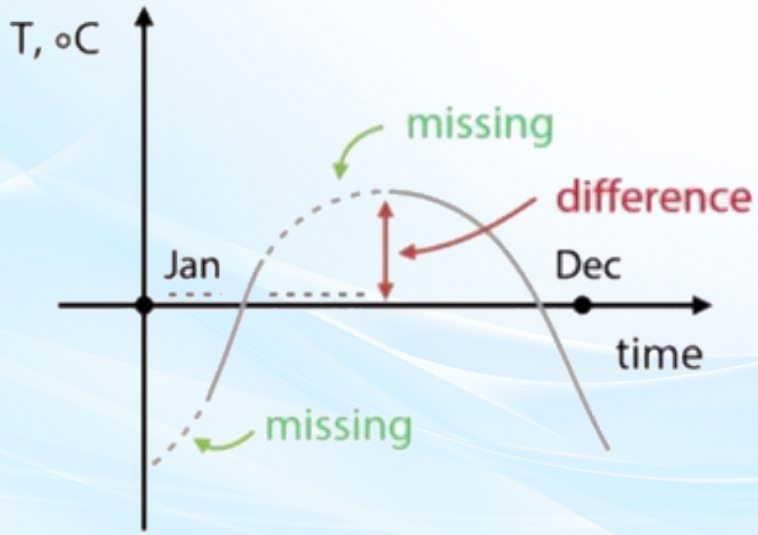
- Если мы изучаем природу, то пьянство метеоролога
  не должно на нее влиять. Можно считать случайным пропуском.
- Если мы изучаем метеоролога, то это неслучайный пропуск.

Заполнение пропусков

Замечание:

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

Пример 1:



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 2:

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 левое.

Закодируем категориальный признак средним значением вещ. признака.

При подсчете среднего использовали out-of-range значения :(

Решение:

Игнорировать пропущенные значения при подсчете средних и проч.

Решающие деревья

Пропуски в данных

Работа с категориальными признаками

Важность признаков

Стрижка деревьев

Плюсы и минусы деревьев

Простые методы обработки пропусков

В деревьях все типы пропусков обрабатываются одинаково.

Способы обработки пропусков:

1. Закодировать каким-то значением, не встречающимся в данных.
   Каким лучше?
   Заменить все пропуски в признаке x_j на $\max\limits_i x_{ij} + 1$.
   Тогда в дереве будет легко выбрать такое разбиение,
   что все объекты с пропущенным x_j пойдут в одно поддерево.
2. Отправлять объект в случайное поддерево.
3. Отправлять объект в оба поддерева и объединять результат.
   Разберем подробнее

Этап обучения

Деление вершины m

Пусть в вершине m оказалась выборка X_m.

$X_m^o(j)$ — объекты из X_m, для которых неизвестен x_j.

- При рассмотрении правила $I\{x_j < t\}$ используем приближение критерия информативности

$$Q(X_m, j, t) \approx \frac{|X_m \setminus X_m^o(j)|}{|X_m|} Q(X_m \setminus X_m^o(j), j, t).$$

- Если правило $I\{x_j < t\}$ оказалось оптимальным, то
  1. отправляем $X_m^o(j)$ в оба поддерева;
  2. оцениваем вероятности объекта пойти в поддерево:

$$\widehat{p}_{m\ell} = \frac{|X_\ell \setminus X_m^o(j)|}{|X_m \setminus X_m^o(j)|}, \quad \widehat{p}_{mr} = \frac{|X_r \setminus X_m^o(j)|}{|X_m \setminus X_m^o(j)|}.$$

Листья

Считаем оценки вероятностей классов стандартным способом.

Этап применения дерева

Пусть x_0 — новый объект. Нужно оценить его отклик Y_0.

Рассмотрим узел m с правилом $I\{x_j < t\}$.
ℓ и r — узлы левого и правого поддеревьев.
Обозначим "$x_0 \Rrightarrow m$" событие "x_0 попал в узел m".

- Значение x_{0j} известно $\Longrightarrow$ отправляем его в ℓ или r

$$\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow m) = \begin{cases} \widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow \ell), \\ \widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow r). \end{cases}$$

- Значение x_{0j} неизвестно $\Longrightarrow$ отправляем его в оба поддерева

$$\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow m) = \widehat{\mathsf{P}}_{x_0}(x_0 \Rrightarrow \ell \mid x_0 \Rrightarrow m)\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow \ell) +$$

$$+ \widehat{\mathsf{P}}_{x_0}(x_0 \Rrightarrow r \mid x_0 \Rrightarrow m)\,\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow r) =$$

$$= \widehat{p}_{m\ell}\,\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow \ell) + \widehat{p}_{mr}\,\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow r).$$

Итоговая оценка $\widehat{\mathsf{P}}_{x_0}(Y_0 = y) = \widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow root)$.

Этап применения дерева

- Значение x_{0j} неизвестно $\Longrightarrow$ отправляем его в оба поддерева

$$\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow m) = \widehat{\mathsf{P}}_{x_0}(x_0 \Rrightarrow \ell \mid x_0 \Rrightarrow m)\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow \ell) +$$

$$+ \widehat{\mathsf{P}}_{x_0}(x_0 \Rrightarrow r \mid x_0 \Rrightarrow m)\,\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow r) =$$

$$= \widehat{p}_{m\ell}\,\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow \ell) + \widehat{p}_{mr}\,\widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow r).$$

Итоговая оценка $\widehat{\mathsf{P}}_{x_0}(Y_0 = y) = \widehat{\mathsf{P}}_{x_0}(Y_0 = y \mid x_0 \Rrightarrow root)$.

Смысл операции: считаем оценки вероятностей классов в поддеревьях и усредняем их с весами, равными оценке вероятности попасть в конкретное поддерево.

Для регрессии нужно заменить условную вероятность на УМО. Т.е. считаем оценку отклика в поддеревьях и усредняем его с весами.

Решающие деревья

Пропуски в данных

Работа с категориальными признаками

Важность признаков

Стрижка деревьев

Плюсы и минусы деревьев

N-арные деревья

Для категориального признака x_j со значениями $\{c_1, \ldots, c_q\}$ разобъем вершину на q вершин по условиям $x_j = c_k$.

Функционал для такого разбиения:

$$Q(X_m, j) = \sum_{k=1}^{q} \frac{|X_k|}{|X_m|} H(X_k) \longrightarrow \min_j .$$

Минусы:

- Риск получения дерева с очень большим количеством листьев.
- Часто выбираются признаки с большим кол-вом значений, т.к. это сильно минимизирует $Q(X_m, j)$ и ведет к переобучению. Но это не всегда так для больших выборок.

Перевод категориальных в вещественные

Для категориального признака x_j со значениями $C = \{c_1, ..., c_q\}$ рассмотрим разбиения вида $I\{x_j \in B\}$, где $B \subset C$.

Таких разбиений $2^q \Rightarrow$ не можем перебрать все.

Оказывается, можно обойтись без такого перебора для задач бинарной классификации и регрессии.

Перевод в вещественные: бинарная классификация

Пусть $\mathcal{I}_m(j, c)$ — индексы объектов, попавших в вершину m и $x_j = c$.

Определим долю положительного класса среди них:

$$\widehat{p}_m(j, c) = \frac{1}{|\mathcal{I}_m(j, c)|} \sum_{i \in \mathcal{I}_m(j,c)} I\{Y_i = 1\}.$$

Заменим категорию c_k на ранг величины $\widehat{p}_m(j, c)$, то есть порядковый номер в упорядоченном наборе из $\widehat{p}_m(j, c_1), ..., \widehat{p}_m(j, c_q)$, и будем работать как с вещественной переменной в данной вершине.

Утверждение.
Поиск оптимального разбиения для энтропийного критерия и Джини найдет то же разбиение, что и при полном переборе.

Pattern recognition and neural networks, Ripley (1996, стр. 218).

Перевод в вещественные: регрессия

Пусть $\mathcal{I}_m(j, c)$ — индексы объектов, попавших в вершину m и $x_j = c$.

Определим среднее значение таргета среди них:

$$\widehat{y}_m(j, c) = \frac{1}{|\mathcal{I}_m(j, c)|} \sum_{i \in \mathcal{I}_m(j,c)} Y_i.$$

Заменим категорию c_k на ранг величины $\widehat{y}_m(j, c)$, то есть порядковый номер в упорядоченном наборе из $\widehat{y}_m(j, c_1), ..., \widehat{y}_m(j, c_q)$, и будем работать как с вещественной переменной в данной вершине.

Утверждение.
Поиск оптимального разбиения для критерия MSE найдет то же разбиение, что и при полном переборе.

Fisher, On Grouping for Maximum Homogeneity, 1958.

Решающие деревья

Пропуски в данных

Работа с категориальными признаками

Важность признаков

Стрижка деревьев

Плюсы и минусы деревьев

Постановка задачи

Задача:

Оценить степень влияния каждого признака на предсказание.

Иначе говоря, определить числа $\mathrm{I}_j \geqslant 0$, $\mathrm{I}_1 + \ldots + \mathrm{I}_d = 1$, такие что если $\mathrm{I}_{j_1} > \mathrm{I}_{j_2}$, то признак j_1 важнее признака j_2 для данной модели.

Замечание:

Данный функционал у моделей на основе решающих деревьев часто интерпретируют как степень влияния признаков на сам таргет.

Это не верно. Получаемые величины позволяют оценить, насколько полезен оказался каждый из признаков **для данной конкретной модели**.

Mean Decrease in Impurity (MDI)

Пусть X_m — подвыборка, дошедшая до узла m,
X_ℓ, X_r — подвыборки, идущие в левое и правое поддеревья,
H — выбранный критерий информативности.

При разбиении вершины m решается задача:

$$Q(X_m, j, t) = \frac{|X_\ell|}{|X_m|} H(X_\ell) + \frac{|X_r|}{|X_m|} H(X_r) \longrightarrow \min_{\ell, r}.$$

Уменьшение ошибки *относительно* вершины m составляет

$$H(X_m) - \frac{|X_\ell|}{|X_m|} H(X_\ell) - \frac{|X_r|}{|X_m|} H(X_r).$$

Общее уменьшение ошибки на этапе разбиения вершины m по признаку j и порогу t по отношению ко всей выборке:

$$\Delta \mathrm{I}_j^m = \frac{|X_m|}{|X|} H(X_m) - \frac{|X_\ell|}{|X|} H(X_\ell) - \frac{|X_r|}{|X|} H(X_r).$$

Пример

Пусть X_m — подвыборка, дошедшая до узла m,
X_ℓ, X_r — подвыборки, идущие в левое и правое поддеревья,
$H = MSE$ — выбранный критерий информативности.

При разбиении вершины m решается задача:

$$\frac{1}{|X_m|}\left\|Y_\ell - \widehat{Y}_\ell\right\|^2 + \frac{1}{|X_m|}\left\|Y_r - \widehat{Y}_r\right\|^2 \longrightarrow \min_{\ell,r}.$$

Уменьшение ошибки *относительно* вершины m составляет

$$\frac{1}{|X_m|}\left\|Y_m - \widehat{Y}_m\right\|^2 - \frac{1}{|X_m|}\left\|Y_\ell - \widehat{Y}_\ell\right\|^2 - \frac{1}{|X_m|}\left\|Y_r - \widehat{Y}_r\right\|^2.$$

Общее уменьшение ошибки на этапе разбиения вершины m
по признаку j и порогу t по отношению ко всей выборке:

$$\Delta \mathrm{I}_j^m = \frac{1}{|X|}\left\|Y_m - \widehat{Y}_m\right\|^2 - \frac{1}{|X|}\left\|Y_\ell - \widehat{Y}_\ell\right\|^2 - \frac{1}{|X|}\left\|Y_r - \widehat{Y}_r\right\|^2.$$

Mean Decrease in Impurity (MDI)

При построении дерева можем посчитать, какой вклад каждый признак вносит в уменьшение ошибки:

$$\Delta \mathrm{I}_j = \sum_m \Delta \mathrm{I}_j^m \cdot I \left\{ \begin{array}{c} \text{разбиение в вершине } m \\ \text{происходит по признаку } j \end{array} \right\}.$$

Отнормируем данные значения, получаем оценку важности:

$$\mathrm{I}_j = \frac{\Delta \mathrm{I}_j}{\sum\limits_{j=1}^{d} \Delta \mathrm{I}_j}$$

Случай леса.

Пусть $\mathscr{T}$ — набор деревьев в лесу.

$\mathrm{I}_j(T)$ — важность признака j для дерева T.

$$\mathrm{I}_j = \frac{1}{|\mathscr{T}|} \sum_{T \in \mathscr{T}} \mathrm{I}_j(T)$$

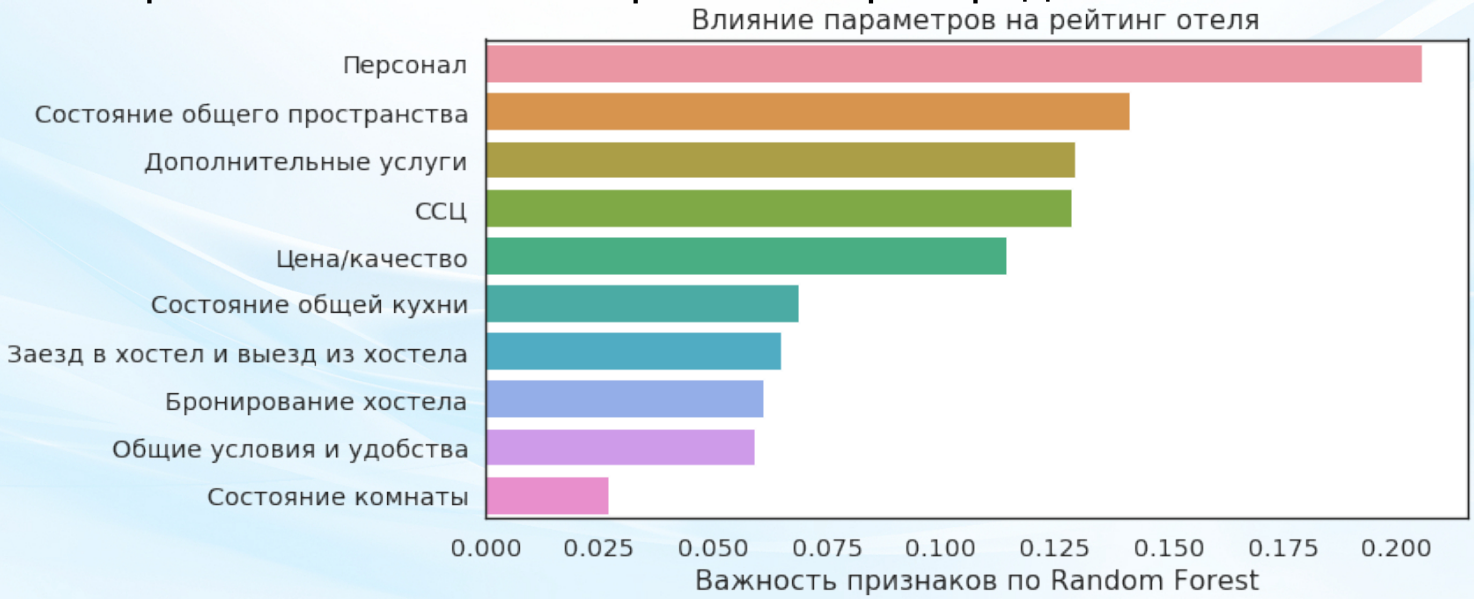
Mean Decrease in Impurity (MDI)

Плюсы:

- Поле `feature_importances_` в `sklearn` — важности признаков, посчитанные этим методом.
- Быстро считается, обучение происходит один раз.

Минусы:

- Важность признаков смещена в сторону признаков с большим количеством значений.
- Считается при использовании лишь обучающей выборки. Не смотрит на полезность признака при предсказании теста.



Решающие деревья

Пропуски в данных

Работа с категориальными признаками

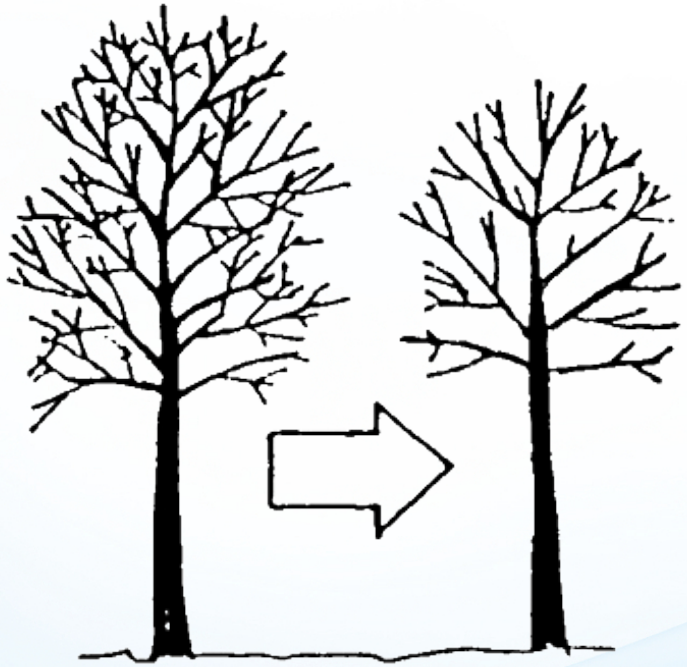
Важность признаков

Стрижка деревьев

Плюсы и минусы деревьев

Стрижка деревьев

Стрижка дерева является альтернативой критериям останова.



Сначала строим очень переобученное дерево, а затем оптимизируем его структуру для улучшения обобщающей способности.

Стрижка по валидационной выборке

X_{val} — валидационная выборка.

Рассмотрим вершину m.

Обозначим $X_{val,m}$ — объекты X_{val}, дошедшие до вершины m.

- Если $X_{val,m}$ — пустое.

  Тогда сделаем

  вершину m листовой.

- Если $X_{val,m}$ — непустое.

  Рассмотрим MSE / число ошибок классификации на $X_{val,m}$

  1. поддеревом из вершины m.
  2. поддеревом из вершины ℓ.
  3. поддеревом из вершины r.
  4. классом $c \in \mathcal{Y}$.

  В зависимости от того,

  что минимально

  1. ничего не делаем.
  2. удаляем поддерево из вершины r и заменяем вершину m на ℓ.
  3. симметрично п. 2.
  4. делаем вершину m листовой с меткой класса c.

Стрижка деревьев: Cost-Complexity Pruning

Пусть есть только обучающая выборка.

Пусть T — дерево, $\mathcal{L}(T)$ — множество листовых вершин,

$R(T)$ — исходная ошибка, считается *суммой* по листьям дерева T.

Введем новую ошибку: $R_\alpha(T) = R(T) + \alpha|\mathcal{L}(T)|$

Построим последовательность вложенных деревьев:

$$T_0 \supseteq T_1 \supseteq ... \supseteq T_q,$$

где T_0 — исходное дерево, а T_q — дерево только из корня.

Каждое следующее дерево получается выкидыванием некоторого поддерева T^m с корнем в вершине m из предыдущего:

$$T_{k+1} = T_k - T^m.$$

Стрижка деревьев: Cost-Complexity Pruning

Для нахождения оптимального T^m минимизируем функционал:

$$R_\alpha(T - T^m) - R_\alpha(T) \longrightarrow \min_t .$$

Распишем его

$$R_\alpha(T-T^m)-R_\alpha(T) = R(T-T^m)+\alpha|\mathcal{L}(T-T^m)|-R(T)-\alpha|\mathcal{L}(T)| =$$

$$= R(T)+R(m)-R(T^m)+\alpha\Big(|\mathcal{L}(T)|-|\mathcal{L}(T^m)|+1\Big)-R(T)-\alpha|\mathcal{L}(T)| =$$

$$= R(m) - R(T^m) + \alpha(1 - |\mathcal{L}(T^m)|)$$

В каком случае $R_\alpha(T - T^m) < R_\alpha(T)$?

Тогда, когда

$$\alpha > \frac{R(t) - R(T^m)}{|\mathcal{L}(T^m)| - 1}.$$

Стрижка деревьев: Cost-Complexity Pruning

В чем смысл α?

Число α регулирует компромисс между размером дерева и ошибкой на обучении. Чем больше α, тем меньшее дерево будет оптимальным. Для $\alpha = 0$ оптимальным будет само T.

Как найти оптимальную вершину m для отрезки?

- *Идея*: увеличиваем α с нуля и ищем m_0 — первую вершина, для которой $R_\alpha(T - T^{m_0}) = R_\alpha(T)$
- Если отрежем поддерево T^m, то для улучшения должно быть
$$\alpha > \frac{R(m) - R(T^m)}{|\mathcal{L}(T^m)| - 1} =: g_0(m)$$
- Тогда ищем вершину $m_0 = \arg\min_m g_0(m)$.
И положим $\alpha_1 = g_0(m_0)$.
- Определим $T_1 = T - T^{m_0}$ и проделаем ту же процедуру для T_1.

Стрижка деревьев: Cost-Complexity Pruning

Процедура

- Инициализация

  $T_0 = T$ — дерево, полученное при $\alpha_0 = 0$, т.е. исходное дерево.

- Итерации:

  Выбрать вершину $m_k \in T_k$ минимизирующую

$$g_k(m) = \frac{R(m) - R(T_k^m)}{|\mathcal{L}(T_k^m)| - 1}$$

  Положить $\alpha^{k+1} = g_k(m_k)$ и $T_{k+1} = T_k - T_k^{m_k}$

На выходе имеем посл-ти $T_0 \supseteq T_1 \supseteq ... \supseteq T_n$ и $\alpha_0 \leqslant \alpha_1 \leqslant ... \leqslant \alpha_n$.

Оптимальное α можно подобрать
по валидационной выборке или по кросс-валидации.

Pattern recognition and neural networks, Ripley

L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression

Решающие деревья

Пропуски в данных

Работа с категориальными признаками

Важность признаков

Стрижка деревьев

Плюсы и минусы деревьев

Плюсы и минусы деревьев

Плюсы

1. Интерпретируемая структура;
2. Восстанавливают сложные нелинейные зависимости;
3. Умеет обрабатывать категориальные признаки;
4. Умеет обрабатывать пропущенные значения;
5. Не требует нормализации и масштабирования признаков.

Минусы

1. Легко переобучаются: маленькое изменение выборки может сильно изменить дерево;
2. Требуют больше вычислений, чем линейные модели;
3. Решающее правило всегда параллельно осям признаков;
4. Плохо обрабатывают линейные зависимости;
5. Жадный выбор разбиения в вершине;
6. Чем дальше вершина от корня, тем меньше обучающая выборка.


ВСЁ!