Занятие 9. Решающие деревья

Гирдюк Дмитрий Викторович, Никольская Анастасия Николаевна

18 ноября 2023

СПбГУ, ПМ–ПУ, ДФС

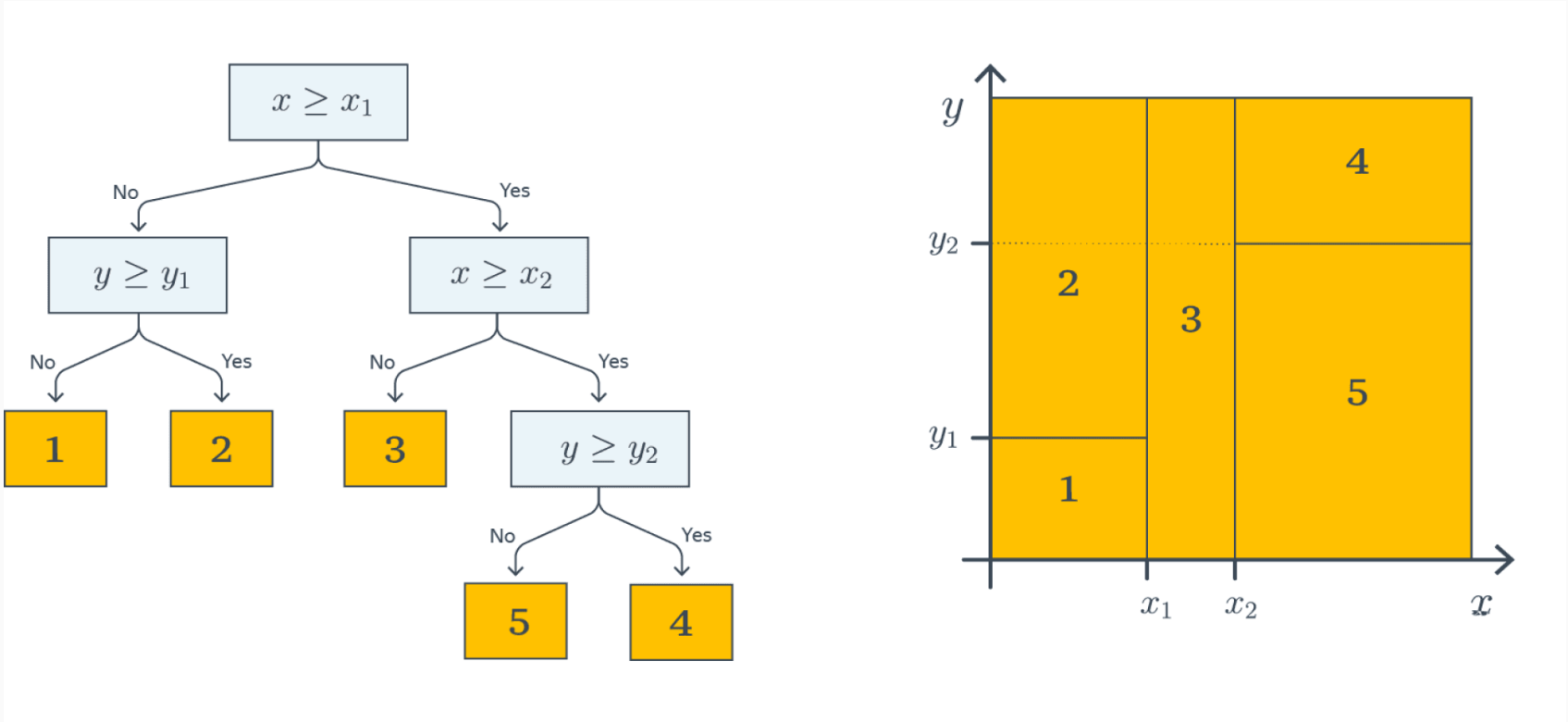
Задача обучения с учителем

- Постановка задачи обучения с учителем (supervised learning): необходимо предсказать значение целевой переменной $y \in Y$ объекта по набору его признаков $\boldsymbol{x} \in X$.
- Среда описывается совместным распределением $f_{X,Y}(\boldsymbol{x}, y)$, а выборкой из нее является набор пар $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{N}$.
- Результирующая модель возвращает значение y по признакам $\boldsymbol{x}$: $y = h(\boldsymbol{x}; \boldsymbol{\theta})$.
- Классификация:
  - $Y = \{0, 1\}$ – бинарная (binary)
  - $Y = \{1, 2, \ldots, K\}$ – многоклассовая (multiclass)
  - $Y = \{0, 1\}^K$ – многозначная (multi-label)
- Регрессия:
  - $Y = R$ – одномерная (ordinal)
  - $Y = R^K$ – многомерная (multiple)

Решающее дерево за минуту на одном слайде

- Решающее дерево – это непараметрический метод классификации и регрессии, представляющий собой способ построения решающих правил в иерархической структуре, состоящей из элементов двух типов – узлов и листьев.
- В узлах находятся решающие правила. В простейшем бинарном случае, в результате проверки правила множество примеров, попавших в узел, разбивается на два подмножества.
- К каждому подмножеству вновь применяется некоторое правило, и процедура рекурсивно повторяется, пока не будет достигнуто некоторое условие остановки алгоритма.
- В последнем узле проверка и разбиение не производится, и он объявляется листом.
- Лист определяет решение для каждого попавшего в него примера в зависимости от типа задачи.

Пример решающего дерева [1]



Определение решающего дерева

Пусть задано бинарное дерево (ациклический граф), в котором:

- Каждой внутренней вершине $v \in V_{in}$ приписано решающее правило (предикат) $Split_v(x, j, t) = [x_j \leqslant t]$, каждой листовой вершине $\widetilde{v} \in V_{leaf}$ приписан ответ/прогноз $Ans(\widetilde{v}) \in Y$.
- В ходе предсказания осуществляется проход по этому дереву к некоторому листу. Для каждого примера движение начинается из корня.
- Начинаем с корня дерева $v_0 \in V_{in}$. В очередной внутренней вершине v проход продолжится влево, если $Split_v = 0$, и вправо, если $Split_v = 1$. Процесс продолжается до момента, пока не будет достигнут некоторый лист $\widetilde{v}$ и будет получен ответ $Ans(\widetilde{v})$.

Замечания

- В общем случае, в листе ответ можно получать на основе некоторой функции (например, регрессии). Однако зачастую ответ $Ans(\widetilde{v})$ для объекта $\boldsymbol{x}^{(i)}$ является константой.
- Решающее правило не обязано быть именно таким: например, это могут быть и интервалы. Кроме того, сплит можно производить на $\geqslant 2$ ветви.
- Понятно, что способность к экстраполяции у решающего дерева за границы области значений обучающей выборки весьма слабая (чаще отсутствует вовсе).
- Дерево решений способно идеально приблизить обучающую выборку, отправив каждое наблюдение из нее в отдельный лист. Переобучение!

Построение решающего дерева i

- Как можно построить решающее дерево?
- По сути мы имеем дело с функцией

$$h(\boldsymbol{x}; \mathcal{D}) = \sum_{l=1}^{L} a_l \left[\boldsymbol{x} \in \widetilde{v}_l\right]$$

- Она не просто недифференцируема, ее структура зависит от обучающей выборки.
- Свести задачу построения дерева к задаче математической оптимизации можно (в теории она может быть переписана как задача смешанного целочисленного программирования), однако размер такой задачи будет внушительный.

Построение решающего дерева ii

- Достаточно очевидно, как можно построить решающее дерево, состоящее из одного решающего правила для случая непрерывного признака (что делать с категориальными признаками рассмотрим чуть позже): перебрать все уникальные значения признака в качестве границы t (или средние между значениями в их отсортированном ряду) и выбрать то, на котором достигается наименьшее значение лосса (его выбор также обсудим далее).
- Вопрос: какова алгоритмическая сложность построения такого правила?

Построение решающего дерева iii

- Увы, последовательно построив набор оптимальных (на этапе разбиения) решающих правил, мы вряд ли придем к оптимальному решающему дереву. Тут под оптимальным деревом понимается, конечно, не вырожденный случай, когда все объекты попадают в отдельные листы, а дерево оптимальное и при этом имеющее минимальную глубину.
- Задача построения такого дерева является NP-полной.
- Так что алгоритмы построения решающих деревьев являются жадными и напичканы всевозможными эвристиками для "поддержания формы" дерева.

Жадный алгоритм построения дерева

- Создаём вершину v для подвыборки $\mathcal{D}_v \in \mathcal{D}$, соответствующей текущей вершине. Изначально $\mathcal{D}_v = \mathcal{D}$.
- Если выполнен критерий остановки $Stop(\mathcal{D}_v)$, останавливаемся, объявляем эту вершину листом и присваиваем ей ответ $Ans(\mathcal{D}_v)$.
- Иначе: находим признак и предикат на нем $Split(\mathcal{D}_v)$, обеспечивающий наилучшее разбиение выборки по некоторому критерию $Gain(\mathcal{D}_v, Split_v)$.
- Рекурсивно повторяем процедуру для левой и правой ветви $\mathcal{D}_v^l$ и $\mathcal{D}_v^r$ до достижения критерия остановки.

Замечания по жадному алгоритму

- В разных алгоритмах применяются разные эвристики для “ранней остановки” или “отсечения”, чтобы избежать построения переобученного (слишком глубокого несимметричного) дерева.
- $Ans(\mathcal{D}_v)$ в случае задачи классификации – метка самого частого класса или оценка дискретного распределения вероятностей классов для объектов, попавших в этот лист; в случае задачи регрессии — среднее, медиана или другая статистика.
- В качестве критерия остановки может быть следующее: достижение максимального числа итераций, достижение минимального числа объектов в листе, значение функционала качества уменьшается меньше чем на заданный порог и т.д.
- Строгой теории, которая бы связывала оптимальность выбора разных вариантов критериев разбиения и разных метрик (классификации и регрессии) в общем случае нет.

Критерий разбиения

- Введем понятие *информативности* (англ. impurity).:

$$Q(\mathcal{D}_v) = \min_{c \in Y} \frac{1}{|\boldsymbol{X_v}|} \sum_{(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathcal{D}_v} \mathcal{L}(y^{(i)}, c)$$

- Тут в качестве c может быть как число из $\mathbb{R}$ для задачи регрессии, так и вектор вероятностей $c = (c_1, \ldots, c_K)$ в задаче классификации.
- Разность между взвешенной информативностью вершины-родителя $Q(\mathcal{D}_v)$ и суммарной информативностью ее двух потомков $Q(\mathcal{D}_v^l)$ и $Q(\mathcal{D}_v^r)$ есть величина неотрицательная, которая показывает, насколько лучше происходит предсказание в вершинах после сплита относительно исходной вершины:

$$Gain(\mathcal{D}_v, Split_v) = |\boldsymbol{X}_v| Q(\mathcal{D}_v) - |\boldsymbol{X}_v^l| Q(\mathcal{D}_v^l) - |\boldsymbol{X}_v^r| Q(\mathcal{D}_v^r)$$

- На его основе можно находить признак и соответствующий предикат для ветвления.

Ликбез: энтропия

- Энтропия Шеннона определяется для системы с N возможными состояниями как:

$$H = -\sum_{i=1}^{N} p_i \log_2 p_i, \tag{1}$$

  где p_i – вероятности нахождения системы в i-ом состоянии.

- Энтропия есть мера неопределенности системы. В нашем контексте – мера непредсказуемости реализации случайной величины.
- Энтропия максимальна при равномерном распределении классов и равна 0, если мы имеем дело с детерминированной величиной.

Информативность в задаче классификации: энтропия

- На основе энтропии: лучшим атрибутом разбиения будет тот, который обеспечит максимальное снижение энтропии результирующего подмножества относительно родительского.
- Выводится через минимизацию логарифма правдоподобия (занятие по логистической регрессии)

$$Q(\mathcal{D}_v) = \min_{\sum_k c_k = 1} \left(-\frac{1}{|\boldsymbol{X_v}|} \sum_{(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathcal{D}_v} \sum_{k=1}^{K} I\left[y^{(i)} = k\right] \log c_k \right)$$

$$Q(\mathcal{D}_v) = -\sum_{k=1}^{K} p_k \log p_k$$

где p_k есть доля попавших в вершину объектов класса k, т.е.

$$p_k = \frac{1}{|\boldsymbol{X_v}|} \sum_{(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathcal{D}_v} I\left[y^{(i)} = k\right]$$

Информативность в задаче классификации: критерий Джини

- В качестве альтернативы можно использовать критерий Джини (Gini impurity):

$$Q(\mathcal{D}_v) = \sum_{k=1}^{K} p_k(1 - p_k) = 1 - \sum_{k=1}^{K} p_k^2$$

- Интерпретация следующая: p_k есть оценка вероятности того, что наугад взятый объект в листе/поддереве относится к классу k, а $1 - p_k$ – оценка вероятности наблюдению оказаться неправильно классифицированным.
- Т.е. критерий Джини есть ожидаемая доля ошибок классификации.

Информативность в задаче классификации

- С вычислительной точки зрения Джини предпочтительнее: при вычислении энтропии нужен логарифм.

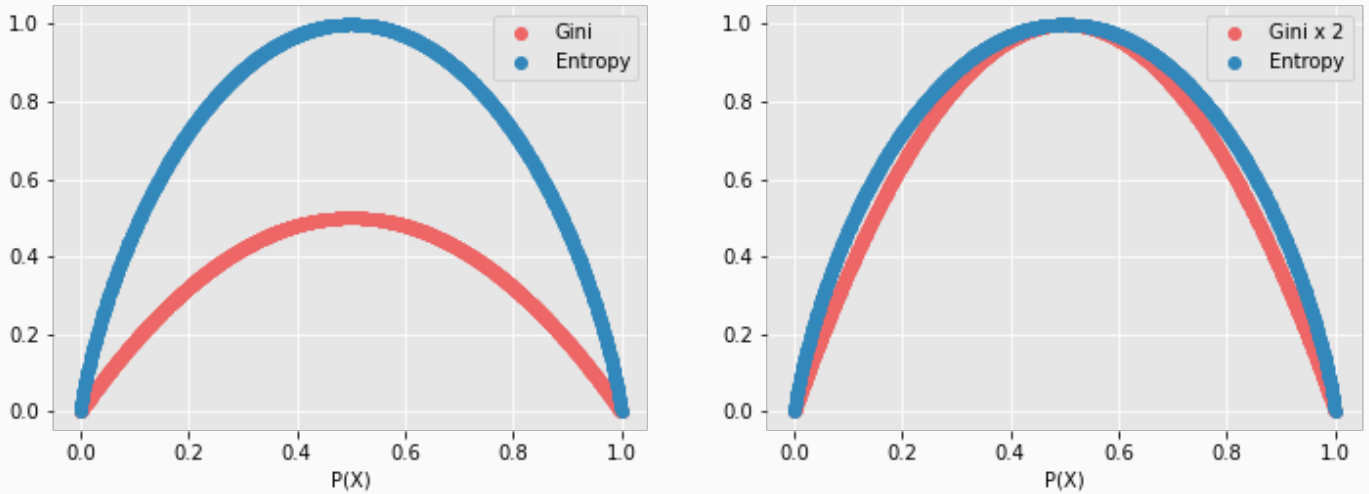


Рис. 1: Графики критериев [2]

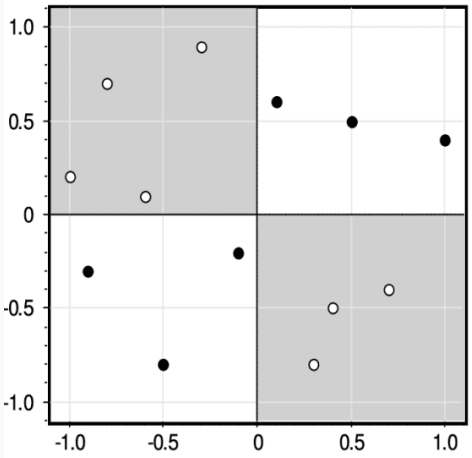
Информативность в задаче регрессии

- Для регрессии в качестве лосса могут быть использованы MSE или MAE.
- Тогда в качестве оптимального константного предсказания будут использоваться выборочное среднее и выборочная медиана, а информативность есть дисперсия и абсолютное отклонение от медианы соответственно

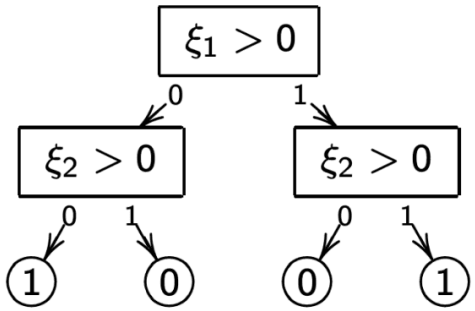
$$Q(\mathcal{D}_v) = \min_{c \in Y} \frac{1}{|\boldsymbol{X_v}|} \sum_{(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathcal{D}_v} (y^{(i)} - c)^2 =$$

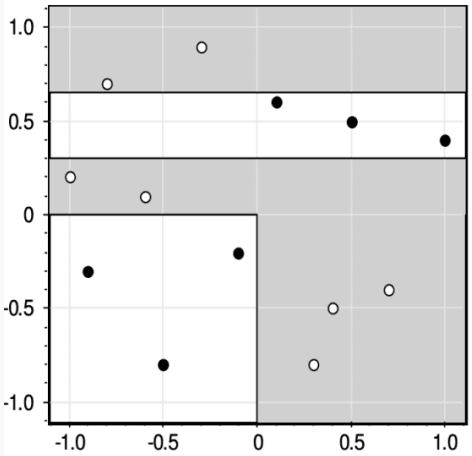
$$= \sum_{(\boldsymbol{x}^{(i)}, y^{(i)}) \in \mathcal{D}_v} \frac{(y^{(i)} - \overline{y})^2}{|\boldsymbol{X_v}|}$$

Пример неоптимальности жадного алгоритма: XOR [3]

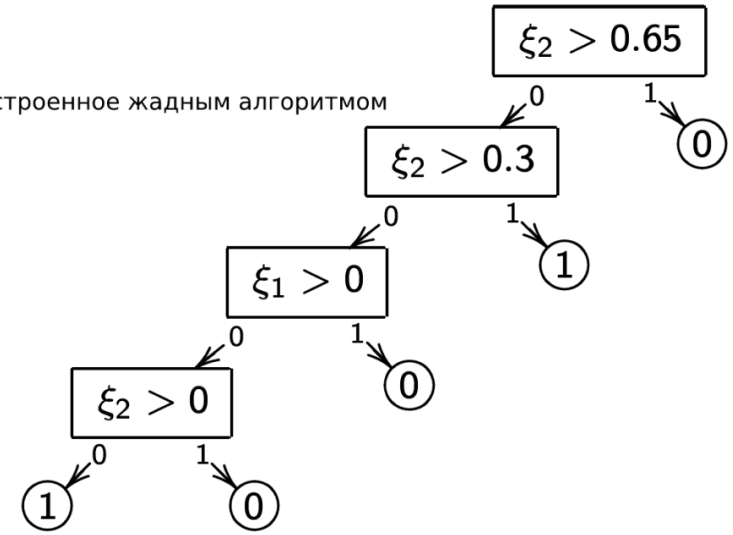


Оптимальное дерево





Дерево, построенное жадным алгоритмом



Пример решающего дерева на ирисах Фишера [4]

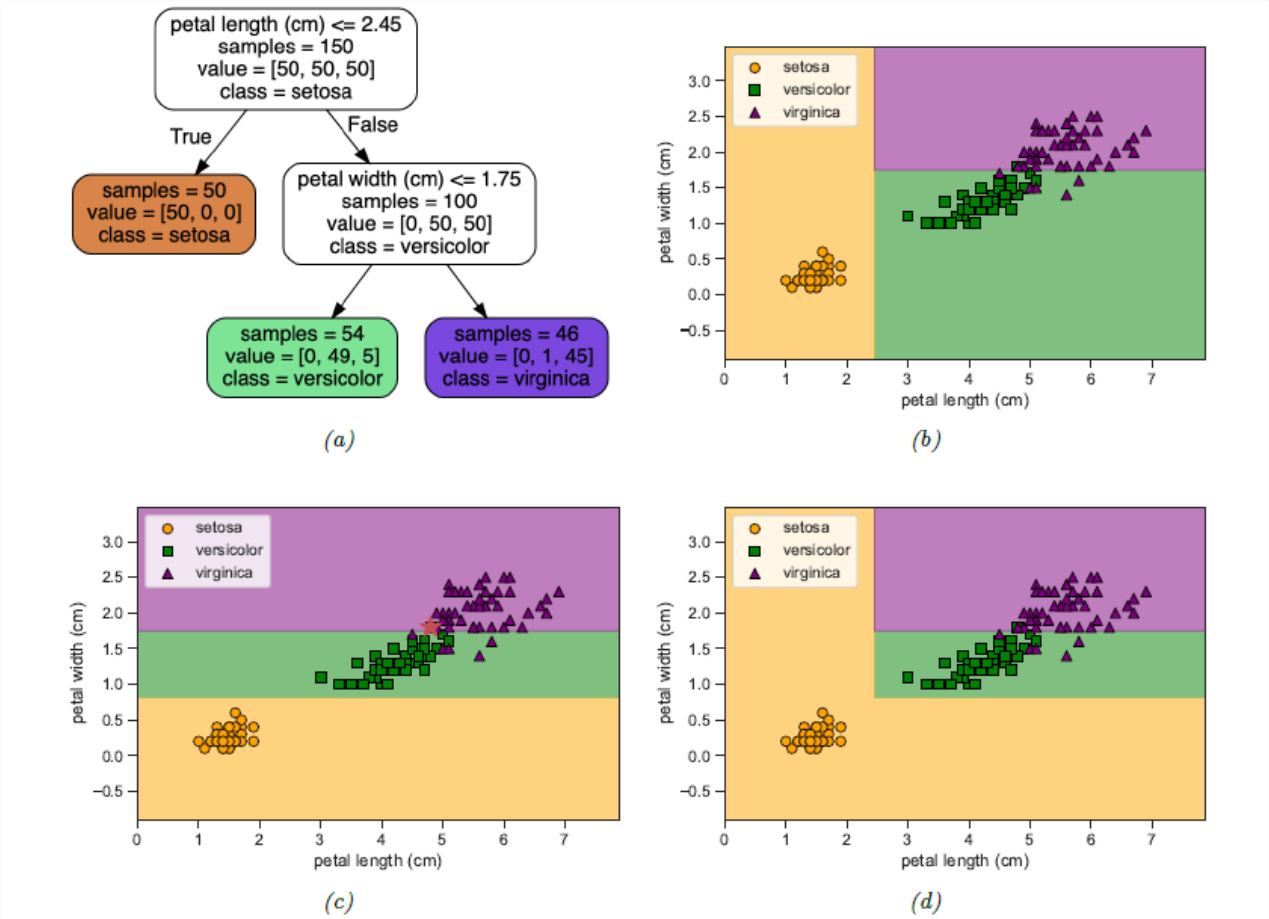


Figure 18.3: (a) A decision tree of depth 2 fit to the iris data, using just the petal length and petal width features. Leaf nodes are color coded according to the majority class. The number of training samples that pass from the root to each node is shown inside each box, as well as how many of these values fall into each class. This can be normalized to get a distribution over class labels for each node. (b) Decision surface induced by (a). (c) Fit to data where we omit a single data point (shown by red star). (d) Ensemble of the two models in (b) and (c). Generated by dtree_sensitivity.ipynb.

Категориальные признаки

- Ранее мы обсудили процесс нахождения оптимального сплита для непрерывного признака. Что делать с категориальными?
- Если категориальный признак принимает K значений, то простой перебор даст $2^{K-1} - 1$ уникальных сплитов. Немало!
- Для задачи бинарной классификации можно упорядочить категории по доле примеров класса 1 со значением c_k. И работать с трансформированным признаком аналогично непрерывному случаю.
- Для регрессии – по среднему значению таргета.

Пропущенные значения

- Интересной особенностью применения решающих деревьев является возможность работы с пропущенными значениями.
- На этапе обучения текущее подмножество выборки с пропущенными значениями по текущему признаку отправляется в обе ветви дерева. При этом запоминаются доли наблюдений в обеих ветвях.
- На этапе применения решающего дерева предсказание для объекта с пропущенным значением по текущему признаку ветвления считается на основе взвешенной (коэффициенты, что мы сохранили при обучении) суммы ответов, которые будут получены из каждой ветви дерева.

Регуляризация

- Чтобы дерево не переобучалось, ветвление обычно останавливают по одному из следующих критериев:
  - Ограничение по максимальной глубине дерева.
  - Ограничение на минимальное количество объектов в листе.
  - Ограничение на максимальное количество листьев в дереве.
  - Требование, чтобы функционал качества при делении текущей подвыборки на две улучшался не менее чем на $xx\%$.
- Отсечение ветвей также можно производить уже после того, как дерево было построено, используя результаты на отложенной выборке.

Преимущества и недостатки решающих деревьев

- $+$ Отличная интерпретируемость и простота использования.
- $+$ Минимальная предобработка: нормализацию делать не нужно, с пропущенными значениями работать умеют.
- $+$ (В теории) работа как с данными в интервальной шкале, так и с категориальными признаками.
- $-$ Жадная стратегия построения дерева часто дает на выходе очень сложную (глубокую) структуру дерева. Фактически, решающие деревья сильно склонны к переобучению.
- $-$ Высокая чувствительность к шуму, к выборке в целом, к выбранным критериям разбиения. Дисбаланс классов также мешает, его желательно контролировать.

Реализации решающих деревья

- Решающие деревья активно развивались с 80-х, существует немало реализаций: ID3 (классификация, минимизация энтропии); CART (первый вариант еще и для задачи регрессии; критерий Джини); C4.5, C5.0 (автор ID3, уже 2000-е, коммерческие реализации).
- В scikit-learn'е реализованы CART-деревья. Отдельные классы для регрессии и классификации. Нет поддержки категориальных признаков. Зато есть масса параметров, контролирующих структуру дерева.

Использованные источники i

1. **Глава по решающим деревьям из учебника по машинному обучению школы анализа данных yandex.** URL: https://academy.yandex.ru/handbook/ml/article/reshayushchiye-derevya.
2. **Decision Trees: Gini vs Entropy.** URL: https://quantdare.com/decision-trees-gini-vs-entropy/.
3. *Воронцов К.* **Презентация по логическим методам классификации из курса лекций Воронцова К.В.** URL: http://www.machinelearning.ru/wiki/images/archive/9/97/20140227072517!Voron-ML-Logic-slides.pdf.
4. *Murphy K. P.* **Probabilistic Machine Learning: An introduction.** MIT Press, 2022. URL: probml.ai.