Машинное обучение

Лекция 14:
Деревья решений

Докладчик: Тима Бовт

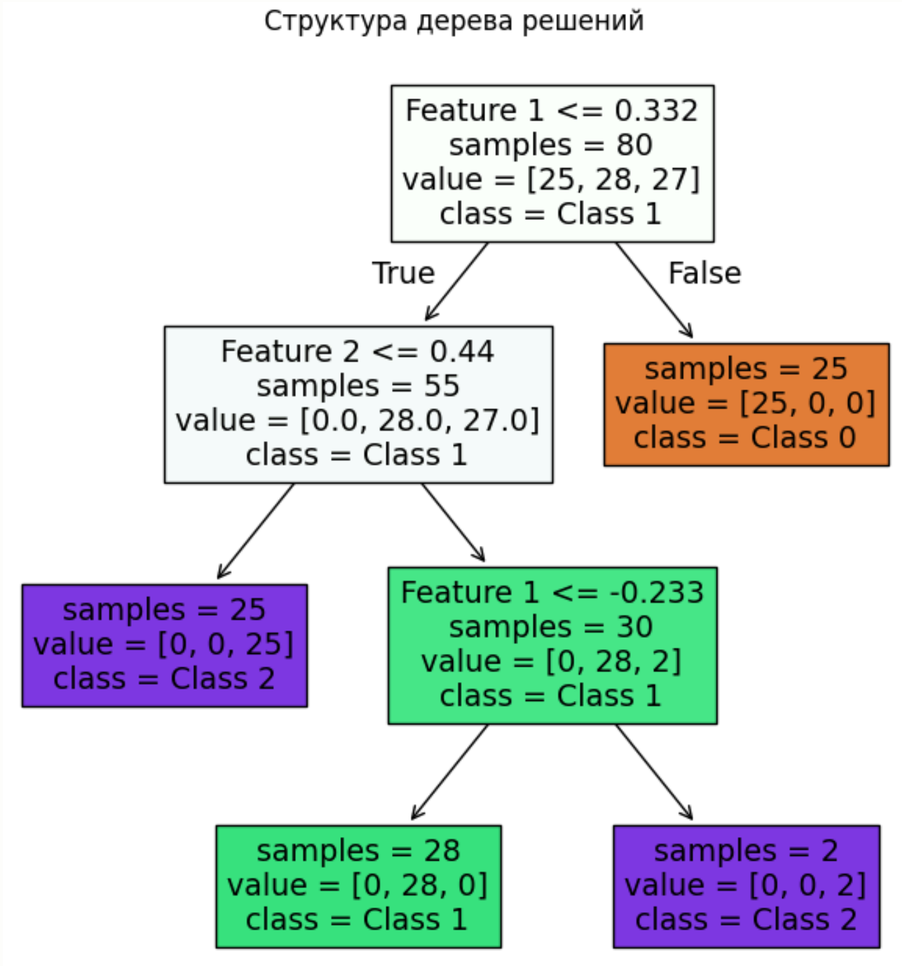
Что рассмотрим сегодня?

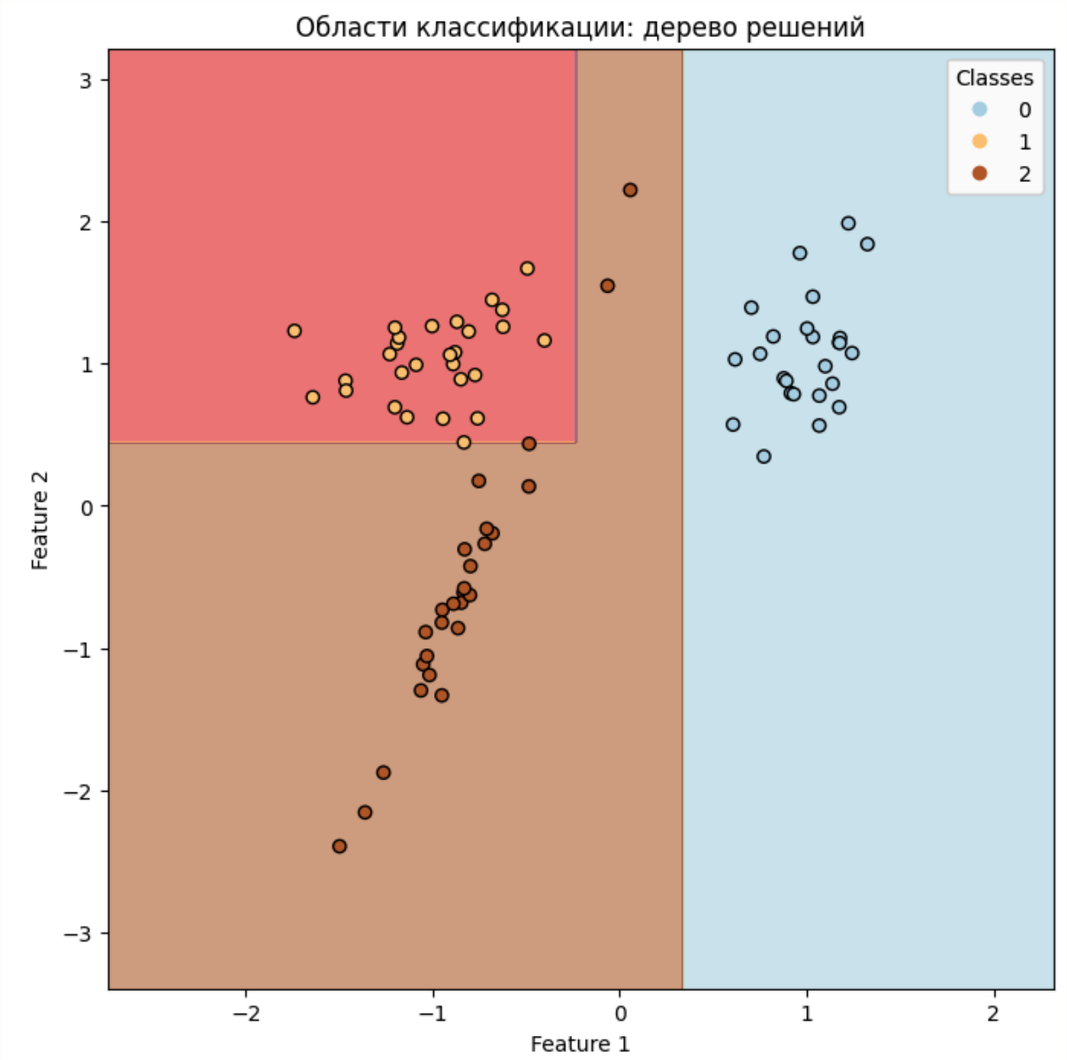
- Идея решающих деревьев
- Постановка задачи и принцип построения деревьев
- Жадный алгоритм построения решающего дерева
- Критерии ветвления для конкретных значений
- Регуляризация
- Работа с пропущенными значениями

Идея решающих деревьев

Решение задачи классификации

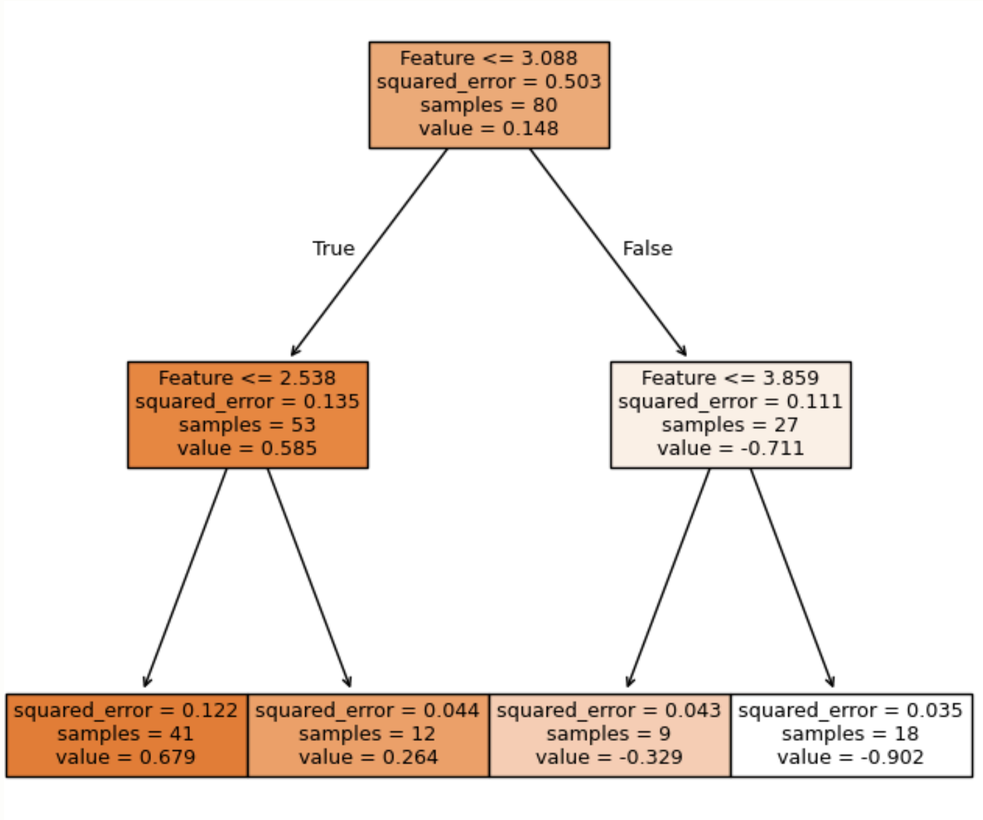
Решающее дерево – последовательность решающих правил (предикатов)

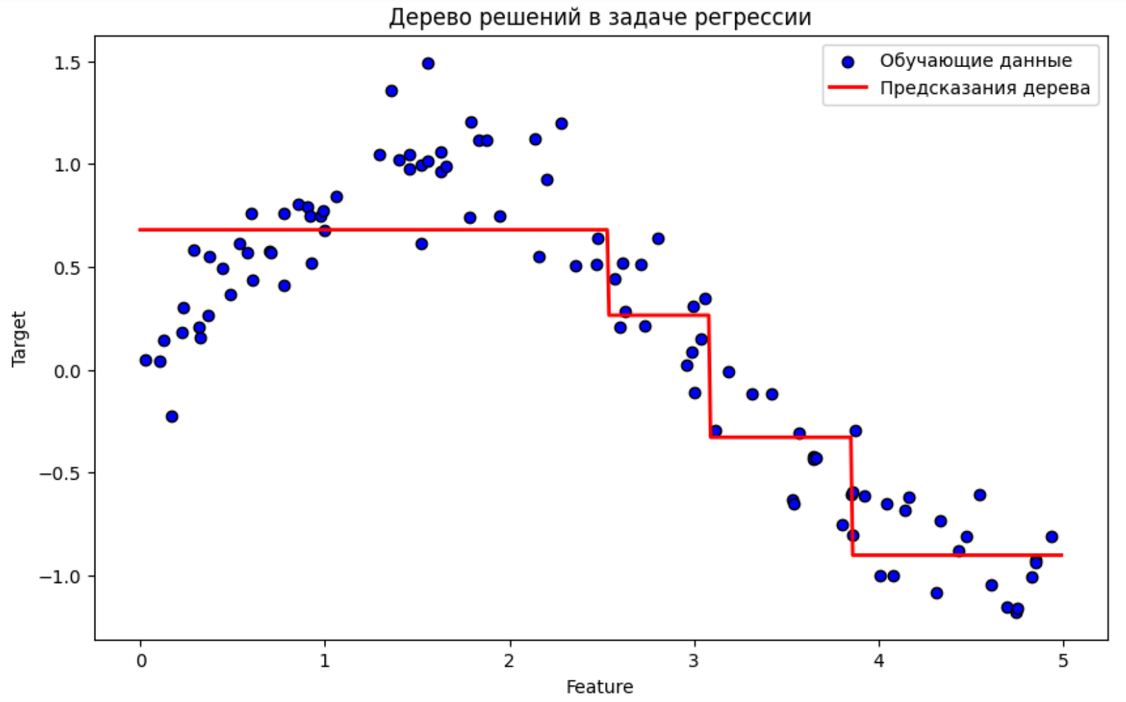




Решение задачи регрессии (глубина=2)

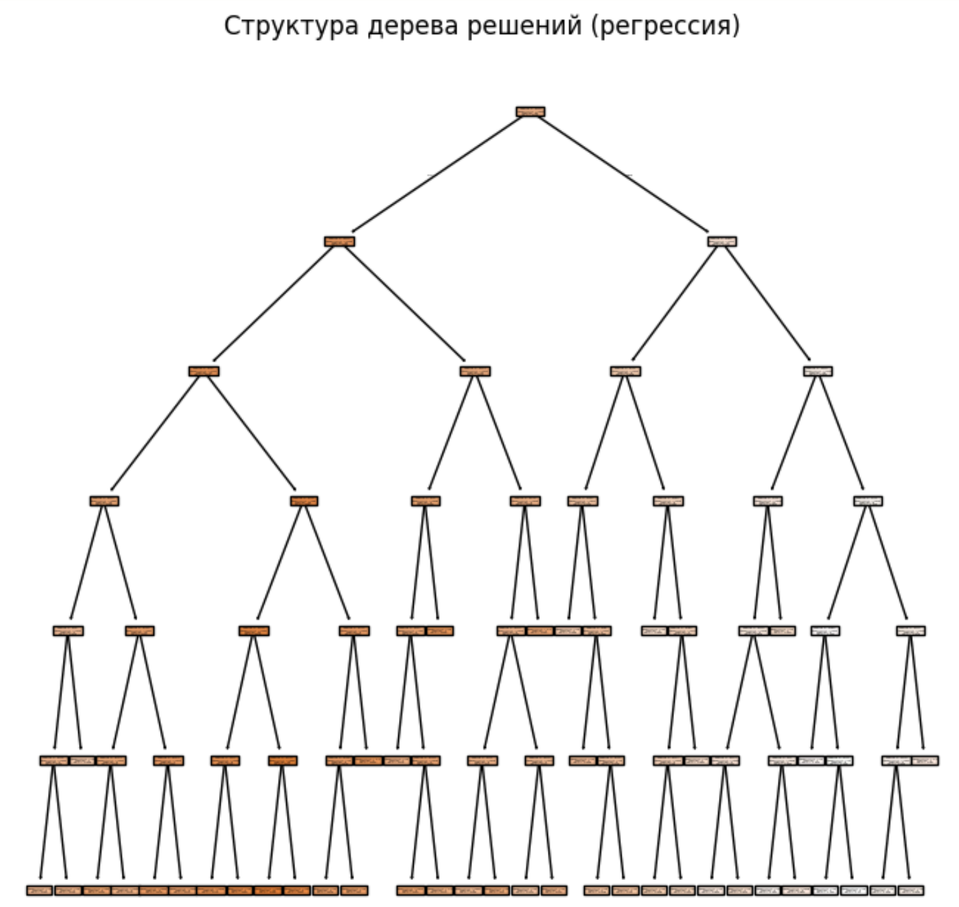
Решающее дерево – последовательность решающих правил (предикатов)

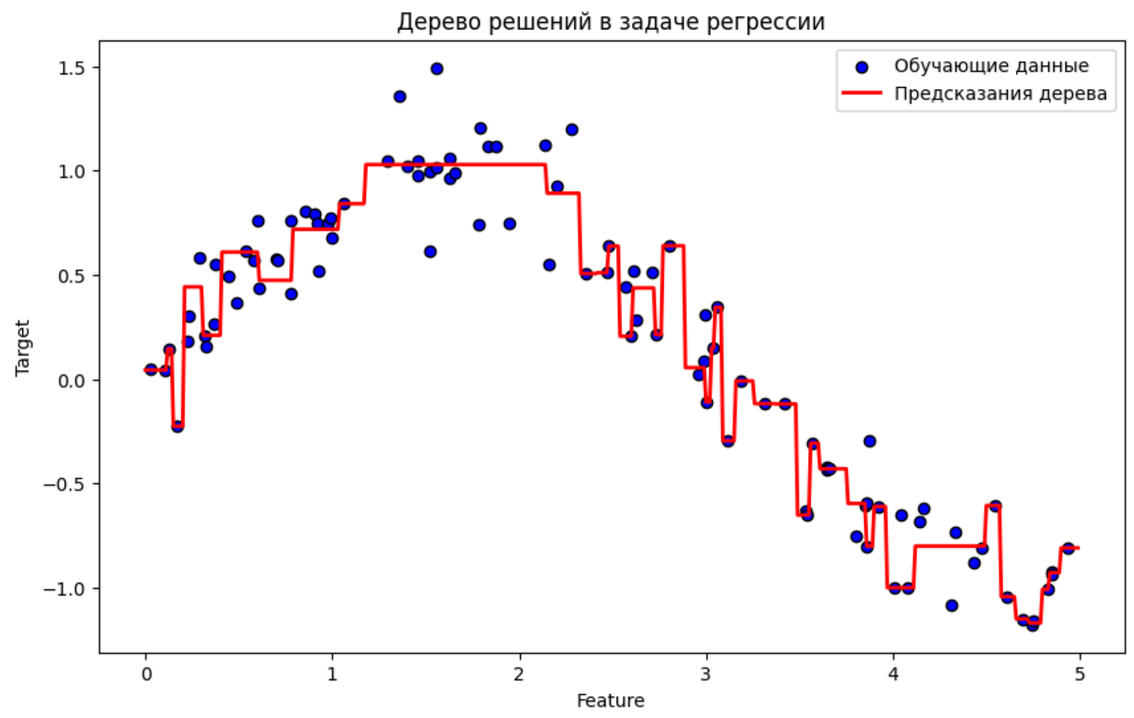




Решение задачи регрессии (глубина=6)

Решающее дерево – последовательность решающих правил (предикатов)





Постановка задачи
и принцип построения дерева

Принцип построения дерева

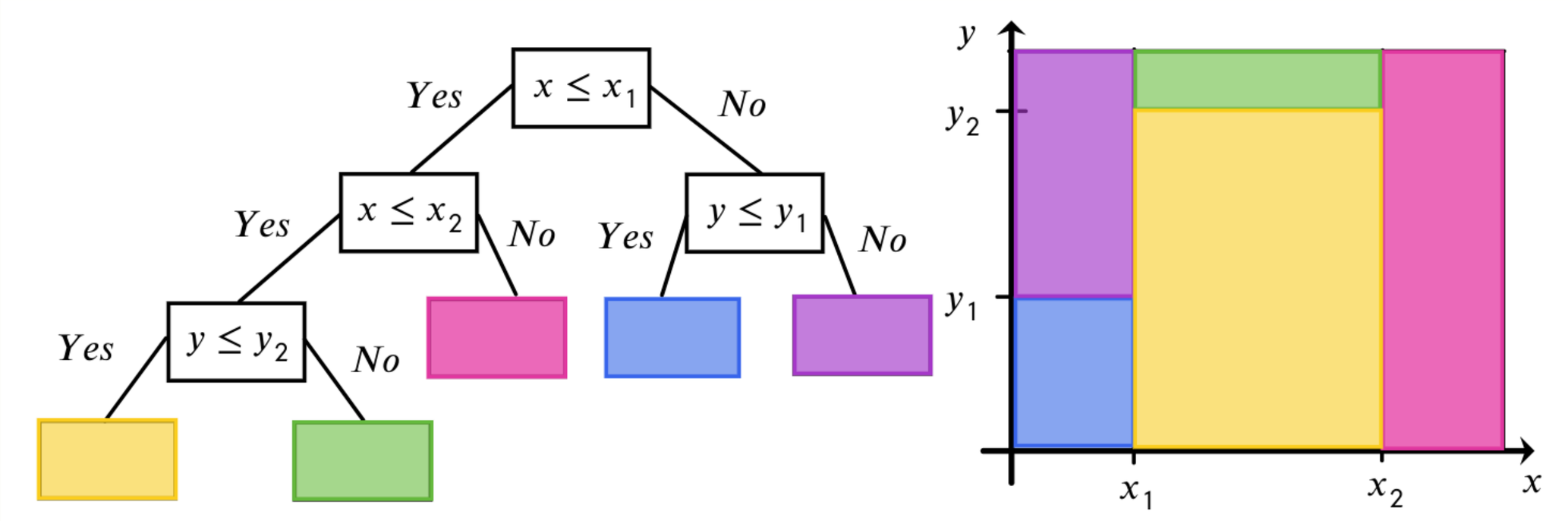
Пусть задана выборка $\mathbb{D} = (X|y)_{i=1}^{n}$, где $X \subseteq \mathbb{X}$, $y \subseteq \mathbb{Y}$, то есть

$$\mathbb{D} = \begin{pmatrix} x_{11} & \dots & x_{1m} & y_1 \\ \vdots & \ddots & \vdots & \vdots \\ x_{n1} & \dots & x_{nm} & y_n \end{pmatrix}.$$

Как было замечено ранее, мы можем решать как задачу регрессии, так и задачу классификации.

Принцип построения дерева

В каждом узле этого дерева находится предикат. Если предикат верен для текущего примера из выборки, мы переходим в левого потомка, если нет — в правого. Обычно предикаты — это **взятие порога** по значению какого-то признака. Таким образом, делится исходное пространство признаков. А значит дерево осуществляет **кусочно-постоянную аппроксимацию** реальной зависимости.



Принцип построения дерева

Пусть задано бинарное дерево, в котором:

- каждой внутренней вершине v приписан предикат $P(x) : \mathbb{X} \to \{0, 1\}$;
- каждой листовой вершине приписан прогноз $\hat{y} \in \mathbb{Y}$.

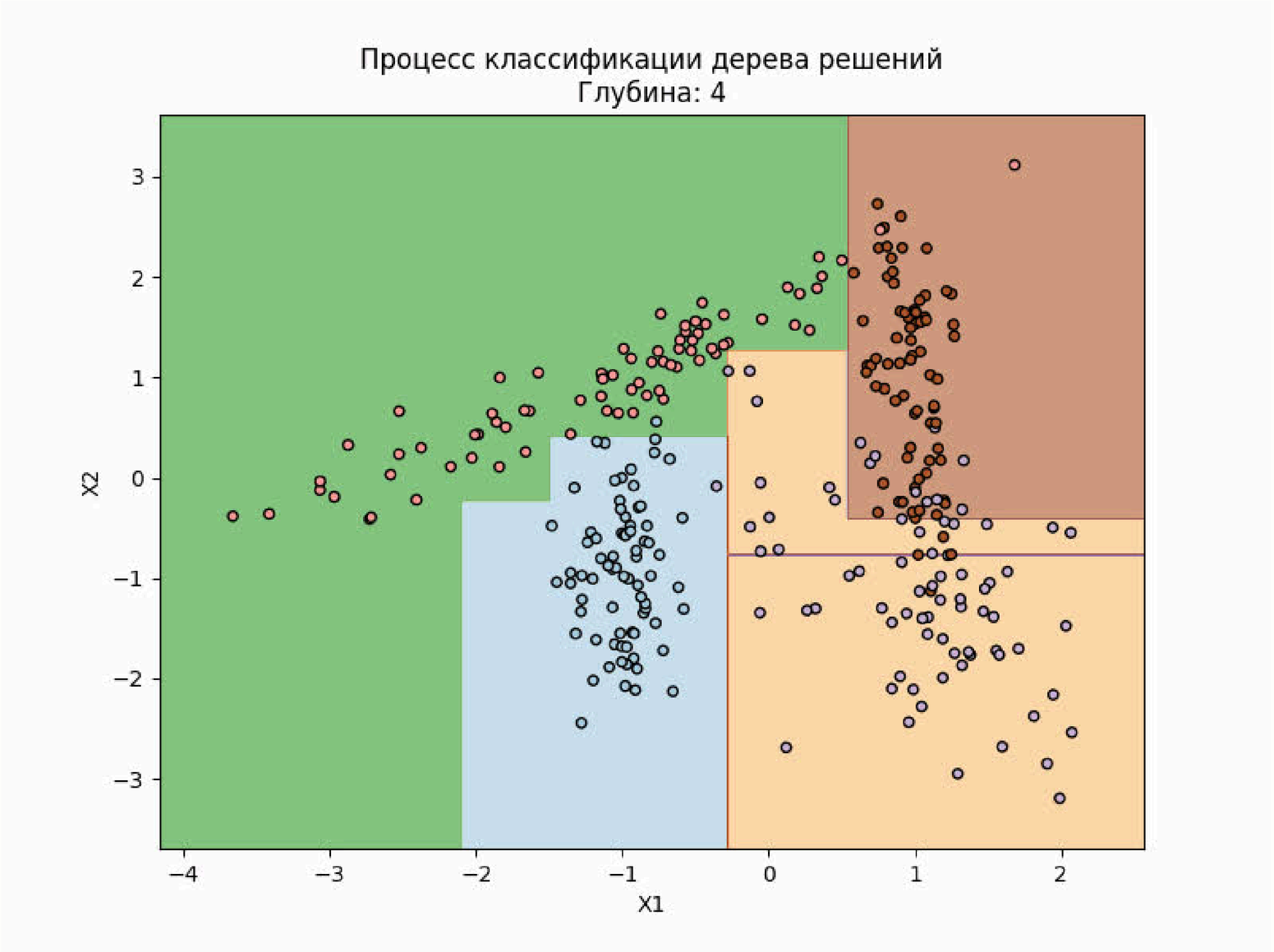
Чаще всего предикат определяется сравнением с порогом $t \in \mathbb{R}$ по некоторому признаку j:

$$P(x) = [x_j \leqslant t].$$

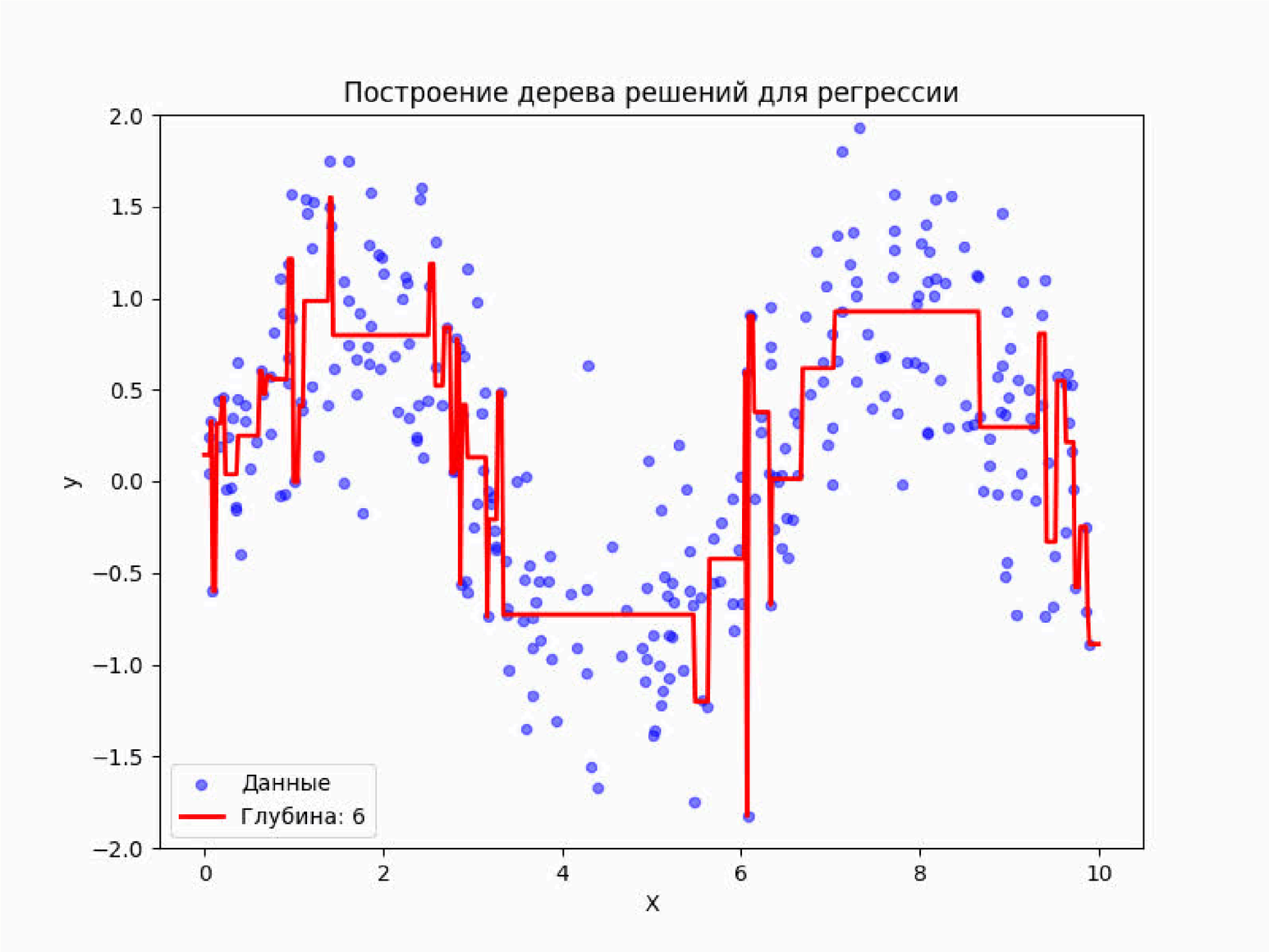
При проходе через узел дерева с данным предикатом объекты будут отправлены в правое поддерево, если значение j-го признака у них меньше либо равно t, и в левое — если больше.

При выполнении прогноза осуществляется проход по дереву к некоторому листу. Для каждого объекта выборки x движение начинается из корня. В очередной внутренней вершине v проход продолжится вправо, если $P(x) = 1$ и влево, если $P(x) = 0$. Проход продолжается до момента, пока не будет достигнут некоторый лист, и ответом алгоритма на объекте x считается прогноз $\hat{y}$, приписанный этому листу.

Принцип построения дерева

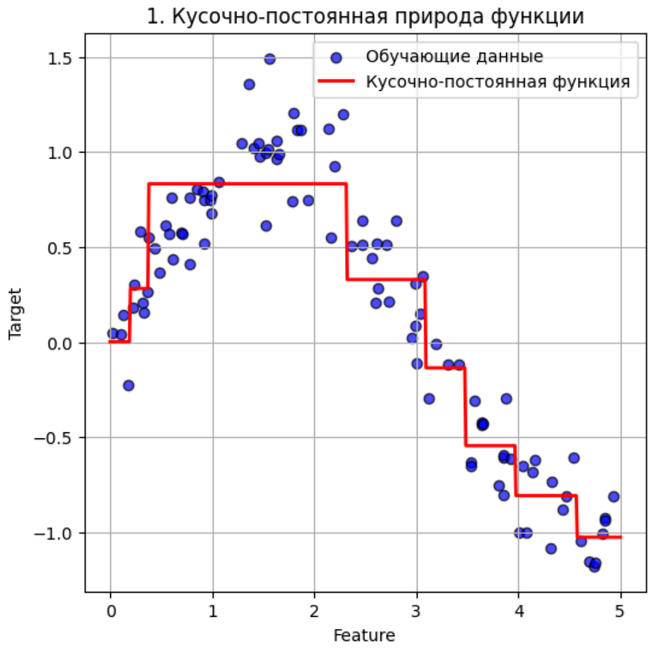


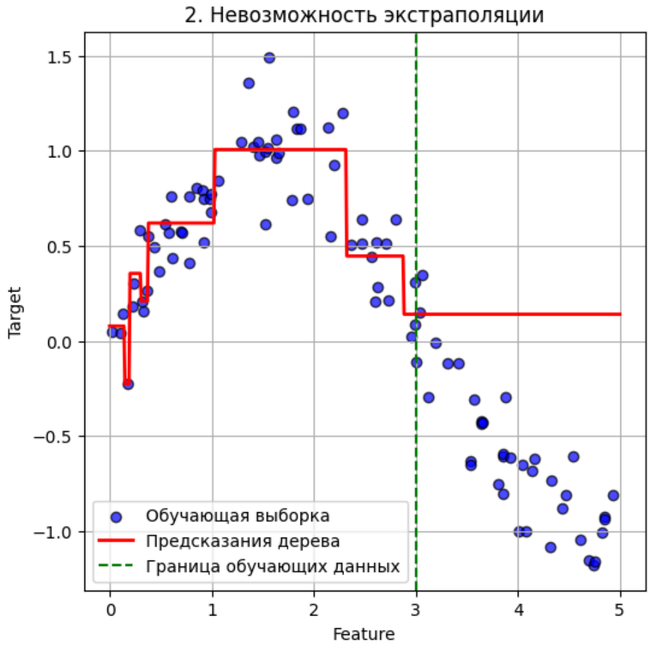
Принцип построения дерева

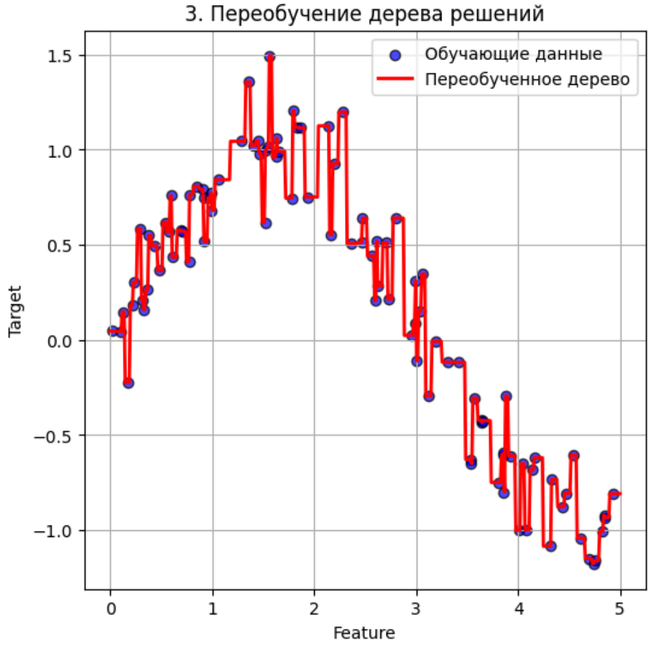


Свойства

- выученная функция — кусочно-постоянная, из-за чего производная равна нулю везде, где задана; следовательно, градиентные методы оптимизации работать не будут;
- дерево решений (в отличие от, например, линейной модели) не сможет экстраполировать зависимости за границы области значений обучающей выборки;
- дерево решений способно идеально приблизить обучающую выборку и ничего не выучить (то есть такой классификатор будет обладать низкой обобщающей способностью): для этого достаточно построить такое дерево, в каждый лист которого будет попадать только один объект.







Проблема построения оптимального дерева

Рассмотрим задачу построения **решающего пня** (дерева решений глубины 1). Индекс j пробегает значения $1, \ldots, m$. А для каждого j мы имеем n объектов, то есть n, вообще говоря, различных значений. Следовательно, для t существует $n - 1$ вариантов выбора порога. То есть для решающего пня мы можем построить $(n - 1) \times m$ предикатов.

Если задана некоторая функция потерь $L(f, X, y)$, то мы ищем решение

$$(j^*, t^*) = \arg\min_{j,t} L(P, X, y).$$

Для каждого из предикатов $P(x)$ нам нужно посчитать значение функции потерь на всей выборке, что, в свою очередь, занимает $O(n)$. Следовательно, сложность алгоритма $O(n^2 \times m)$. И это для одного решающего пня.

И если пытаться построить оптимальное с точки зрения качества на обучающей выборке дерево минимальной глубины (чтобы снизить переобучение), то поиск такого дерева — **NP-полная задача**.

Жадный алгоритм построения решающего дерева

Жадный алгоритм

Пусть $X' \subset X_{\text{train}}$ – это множество объектов, попавших в текущий лист. Жадный алгоритм можно описать так:

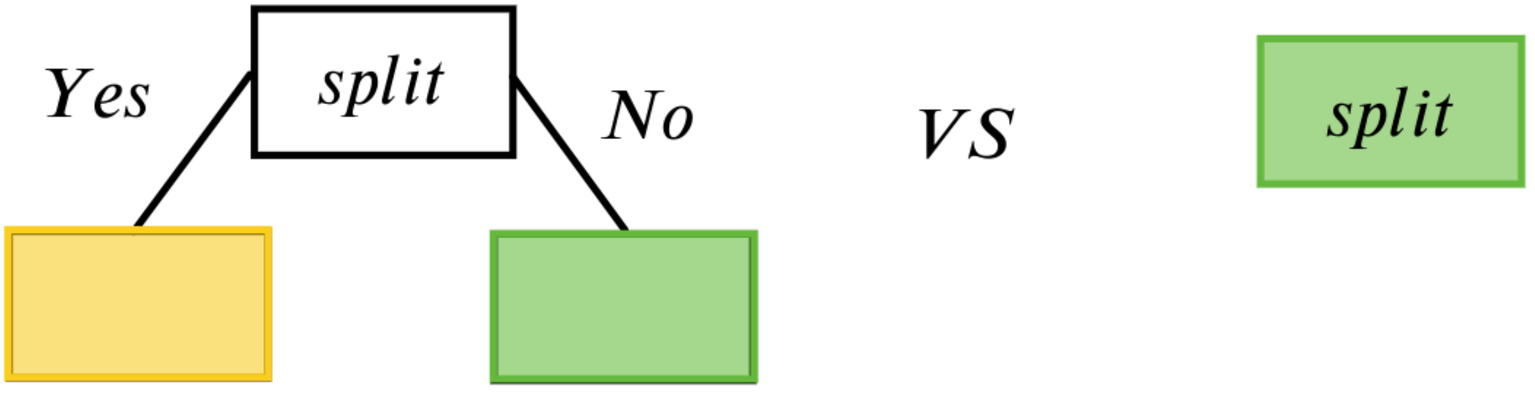
1. создаем вершину v;
2. если выполнен критерий остановки $\text{stop}(X')$, то останавливаемся, объявляем вершину листом и ставим ей в соответствие ответ $\text{ans}(X')$;
3. иначе находим предикат $P(x)$, который определит наилучшее разбиение текущего множества X' на X'_{left}, X'_{right}, максимизируя критерий ветвления $\text{split}(X', j, t)$;
4. повторяем шаги для X'_{left}, X'_{right}.

Теперь остановимся на всех вспомогательных функциях.

- **Критерий остановки** $\text{stop}(X')$ определяет, нужно ли продолжить ветвление или нет. Например, остановка только в том случае, если данные в листе однородные.
- **Ответ** $\text{ans}(X')$ определяет ответ для листа на основе попавших в него объектов:
  - для классификации – метка самого частого класса среди объектов листа;
  - регрессия – среднее, медиана или др. статистика по объектам листа.

Жадный алгоритм

- **Критерий ветвления** $\mathrm{split}(X', j, t)$ измеряет, на сколько хорошим будет предполагаемое разделение; чаще всего оценка производится по некоторой метрике качества дерева, если получившиеся два листа будут терминальными по сравнению с ситуацией, когда исходная вершина – лист.



Информативность

Пусть $\hat{y} \in \mathbb{R}$ – это ответы модели (для задачи регрессии или классификации). И пусть также задана некоторая функция потерь $L(y, \hat{y})$.

В момент поиска оптимального разделения $X' = X'_{\text{left}} \cup X'_{\text{right}}$ мы можем вычислить для $x \in X'$ такой таргет $\hat{y}$, который предсказала бы модель, будь текущая вершина терминальной. А следом можно также рассчитать и функционал L. Предсказанный таргет $\hat{y}$ должен минимизировать среднее качество функции потерь

$$\frac{1}{|X'|} \sum_{(x,y)\in(X'|y')} L(y, \hat{y}).$$

Это значение мы должны минимизировать. Следовательно, оптимальное значение обозначают

$$H(X') = \min_{\hat{y}\in Y} \frac{1}{|X'|} \sum_{(x,y)\in(X'|y')} L(y, \hat{y})$$

и называют **информативностью (impurity)**. Чем она ниже, тем лучше прогноз $\hat{y}$.

Информативность

Теперь рассмотрим решающий пень (дерево глубиной 1). Пусть произошло разделение $X' = X'_{\text{left}} \cup X'_{\text{right}}$, а $\hat{y}_{\text{left}}$, $\hat{y}_{\text{right}}$ – соответствующие предсказанные таргеты. Тогда функция потерь для всего пня равна

$$\frac{1}{|X'|}\left(\sum_{(x,y)\in(X'_{\text{left}}|y'_{\text{left}})} L(y,\hat{y}) + \sum_{(x,y)\in(X'_{\text{right}}|y'_{\text{right}})} L(y,\hat{y})\right) =$$

$$= \frac{1}{|X'|}\left(|X'_{\text{left}}| \cdot \frac{1}{|X'_{\text{left}}|} \sum_{(x,y)\in(X'_{\text{left}}|y'_{\text{left}})} L(y,\hat{y}) + |X'_{\text{right}}| \cdot \frac{1}{|X'_{\text{right}}|} \sum_{(x,y)\in(X'_{\text{right}}|y'_{\text{right}})} L(y,\hat{y})\right).$$

Тогда информативность равна

$$\frac{|X'_{\text{left}}|}{|X'|} \cdot H(X'_{\text{left}}) + \frac{|X'_{\text{right}}|}{|X'|} \cdot H(X'_{\text{right}}).$$

Итоговая формула критерия ветвления

В итоге для принятия решения мы сравниваем информативности

$$H(X') \text{ и } \frac{|X'_{\text{left}}|}{|X'|} \cdot H(X'_{\text{left}}) + \frac{|X'_{\text{right}}|}{|X'|} \cdot H(X'_{\text{right}})$$

и выбираем тот вариант, при котором информативность ниже.

Домножив все на $|X'|$ получим формальное определение критерия ветвления

$$\text{split}(X', j, t) = |X'| \cdot H(X') - |X'_{\text{left}}| \cdot H(X'_{\text{left}}) + |X'_{\text{right}}| \cdot H(X'_{\text{right}}) \geqslant 0.$$

Чем лучше предполагаемое разделение, тем больше эта величина.

Критерии ветвления для конкретных задач

Критерий втвления для MSE

Рассмотрим задачу регрессии с функцией потерь MSE

$$L(y, \hat{y}) = (y - \hat{y})^2.$$

Информативность текущего листа определяется формулой

$$H(X') = \min_{\hat{y} \in Y} \frac{1}{|X'|} \sum_{(x,y) \in (X'|y')} (y - \hat{y})^2$$

Оптимальным предсказанием для задачи минимизации MSE является среднее значение

$$\hat{y}^* = \frac{1}{|X'|} \sum y_i.$$

тогда формула информативности

$$H(X') = \frac{1}{|X'|} \sum_{(x,y) \in (X'|y')} (y - \hat{y}^*)^2$$

Получается, что оценка значения в каждом листе — это среднее, а выбирать сплиты надо так, чтобы сумма дисперсий в листьях была как можно меньше.

Критерий втвления для missclassification loss

Рассмотрим задачу классификации на K классов **с missclassification loss**

$$L(y, \hat{y}) = [y \neq \hat{y}].$$

Пусть предсказание в листе – это один конкретный класс. Тогда информативность имеет вид

$$H(X') = \min_{\hat{y} \in Y} \frac{1}{|X'|} \sum_{(x,y) \in (X'|y')} [y \neq \hat{y}].$$

Обозначим p_k – доля объектов класса $k \in \{1, \ldots, K\}$ в текущей вершине

$$p_k = \frac{1}{|X'|} \sum_{(x,y) \in (X'|y')} [y = k].$$

Оптимальным предсказанием в листе будет наиболее частый класс $k^* \in \{1, \ldots, K\}$, а тогда информативность

$$H(X') = \frac{1}{|X'|} \sum_{(x,y) \in (X'|y')} [y \neq k^*] = 1 - p_{k^*}.$$

Критерий втвления для энтропии. Метрика Бриера

Для задач с энтропией можно доказать, что информативность задается

$$H(X') = -\sum_{k=1}^{K} p_k \log p_k.$$

Теперь рассмотрим задачу классификации, но пусть предсказание модели – это распределение вероятностей классов $C_1, \ldots, C_K$.

Определим **метрику Бриера**

$$BS = \frac{1}{N} \sum_{i=1}^{N} (f_i - o_i)^2,$$

где f_i – спрогнозированная вероятность события, o_i – реальное значение события: 0 если событие не произошло; 1, если событие произошло. То есть под метрикой Бриера понимается MSE от вероятности.

Критерий информативности Джини

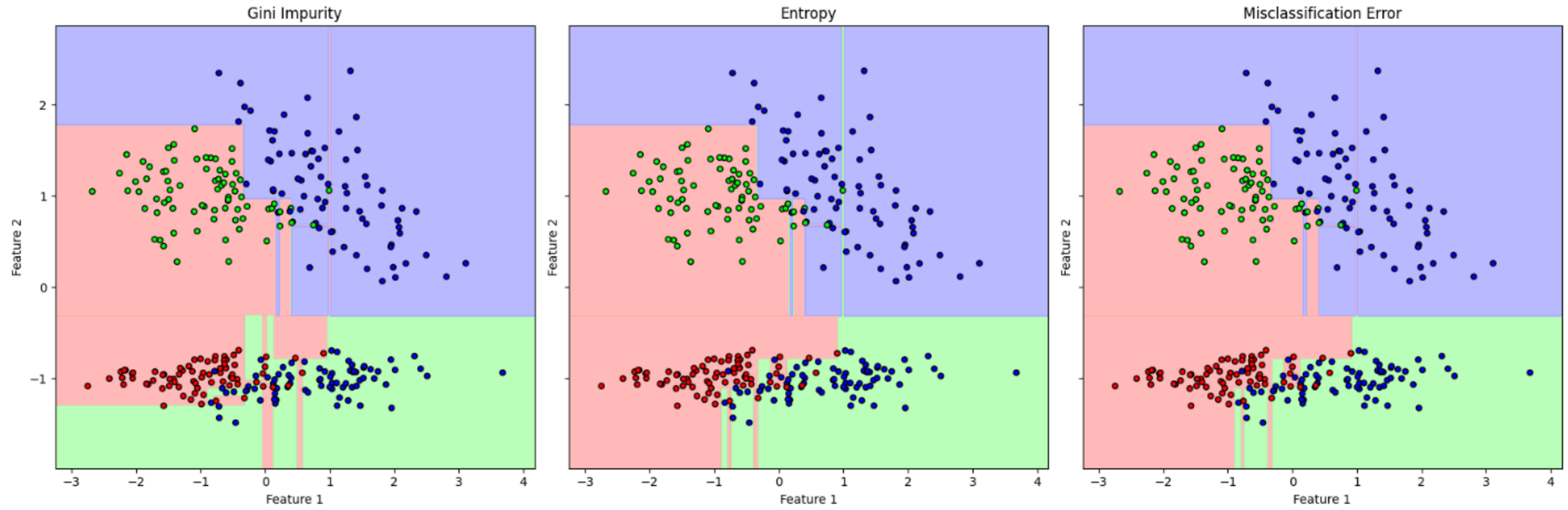
Тогда можно определить информативность, используя метрику Бриера

$$H(X') = \min_{\hat{y} \in Y} \frac{1}{|X'|} \sum_{(x,y) \in (X'|y')} \sum_{k=1}^{K} (C_k - [y \neq k])^2.$$

Можно также доказать, что оптимальное значение этой метрики достигается на векторе частот классов $(p_1, \ldots, p_K)$. Тогда, если подставить это в выражение, то получится **критерий информативности Джини**

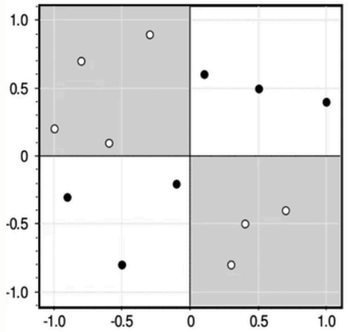
$$H(X') = \sum_{k=1}^{K} p_k(1 - p_k).$$

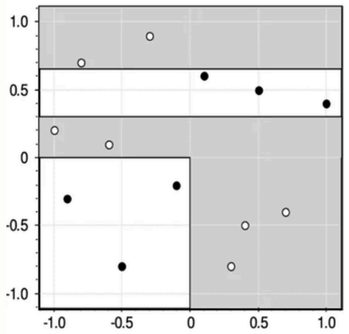
Критерии информативности для задачи классификации



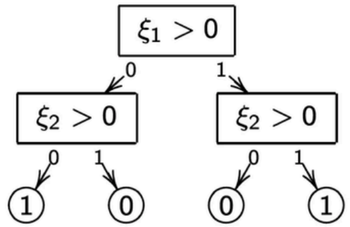
Неоптимальность критериев

Рассмотрим задачу XOR

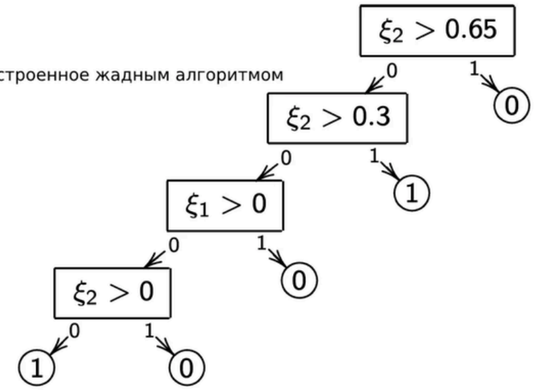




Оптимальное дерево



Дерево, построенное жадным алгоритмом



Жадный алгоритм не дает оптимального решения задачи XOR.

Также бывают ситуации, когда оптимальное с точки зрения выбранной метрики дерево вы получите с критерием ветвления, построенным по другой метрике (например, Джини для misclassification error).

https://education.yandex.ru/handbook/ml/article/reshayushchiye-derevya

Регуляризация

Pruning

Деревья легко переобучаются, если процесс ветвления не остановить вовремя. Таким образом, возникают определенные критерии для остановки ветвления

- ограничение по максимальной глубине дерева;
- ограничение на минимальное количество объектов в листе;
- ограничение на максимальное количество листьев в дереве;
- требование, чтобы функционал разбиения $\text{split}(X', j, t)$ при делении текущей подвыборки на две улучшался не менее чем на s процентов.

Существуют два варианта, когда в проверяются эти критерии:

- можно проверять прямо во время построения дерева, такой способ называется **pre-pruning**;
- можно построить дерево жадно без ограничений, а затем провести стрижку (pruning), то есть удалить некоторые вершины из дерева так, чтобы итоговое качество не сильно ухудшилось, но дерево начало подходить под условия регуляризации, такой способ называется **post-pruning**.

Работа с пропущенными значениями

Работа с пропущенными значениями

Одним из преимуществ дерева является возможность работать с пропущенными значениями.

Рассмотрим этап обучения. При выборе разбиения объекты с пропущенным значением некоторого признака игнорируются. После выбора разбиения, эти объекты отправляются в оба поддерева с соответствующими весами: $\frac{|X'_{\text{left}}|}{|X'|}$ для левого поддерева и $\frac{|X'_{\text{right}}|}{|X'|}$ для правого.

Рассмотрим этап предсказания. Пусть в вершину, где идет разбиение по j-ому признаку приходит объект с пропущенным значением этого признака. В таком случае он отправляется в каждую из дальнейших вектор и там получает соответствующие предсказания $\hat{y}_{\text{left}}$, $\hat{y}_{\text{right}}$. Эти предсказания усредняются с весами

$$\hat{y} = \frac{|X'_{\text{left}}|}{|X'|} \cdot \hat{y}_{\text{left}} + \frac{|X'_{\text{right}}|}{|X'|} \cdot \hat{y}_{\text{right}}.$$

Для задачи регрессии это сразу даст нам таргет, а в задаче бинарной классификации – оценку вероятности класса 1.

Плюсы и минусы деревьев решений

Плюсы деревьев решений

- Дерево решений легко визуализировать и объяснить, особенно для небольших деревьев.
- Не требуется нормализация или стандартизация признаков, так как дерево работает с исходными значениями.
- Дерево решений может обрабатывать как числовые, так и категориальные признаки без преобразования (например, в one-hot encoding).
- Дерево решений не предполагает линейности данных или нормального распределения.

Минусы деревьев решений

- Деревья решений подвержены переобучению, особенно если дерево становится слишком глубоким.
- Небольшие изменения в данных могут привести к построению совершенно другого дерева, так как алгоритм жадно выбирает локально оптимальные разбиения.
- Дерево решений плохо работает с коррелированными признакам, так как оно выбирает только один из них на каждом уровне разбиения.
- Отдельное дерево решений обычно менее точно, чем ансамблевые методы.