

Classification Tree (CART) в Python

Версия 2 30.04.2020

Classification and regression trees CART.
деревья классификации и регрессии

*Breiman, Leo; Friedman, J. H.; Olshen, R. A.; Stone, C. J.
(1984). Classification and regression trees.*

В Python чаще всего используют процедуры

```
sklearn.tree.DecisionTreeClassifier  
sklearn.tree.DecisionTreeRegressor
```

В R чаще всего используют пакеты rpart, party и randomForest.

Термин CART защищен патентом!

Желаете написать процедуру с таким названием, платите деньги. Под другими названиями можно.

Деревья классификации — слабый (weak) метод классификации (регрессии).

Но комбинируя деревья можно получить сильные классификаторы. Такие как случайный лес (Random Forest), gradient boosting machine и XGBoost.

Поэтому без изучения деревьев классификации и регрессии не обойтись.

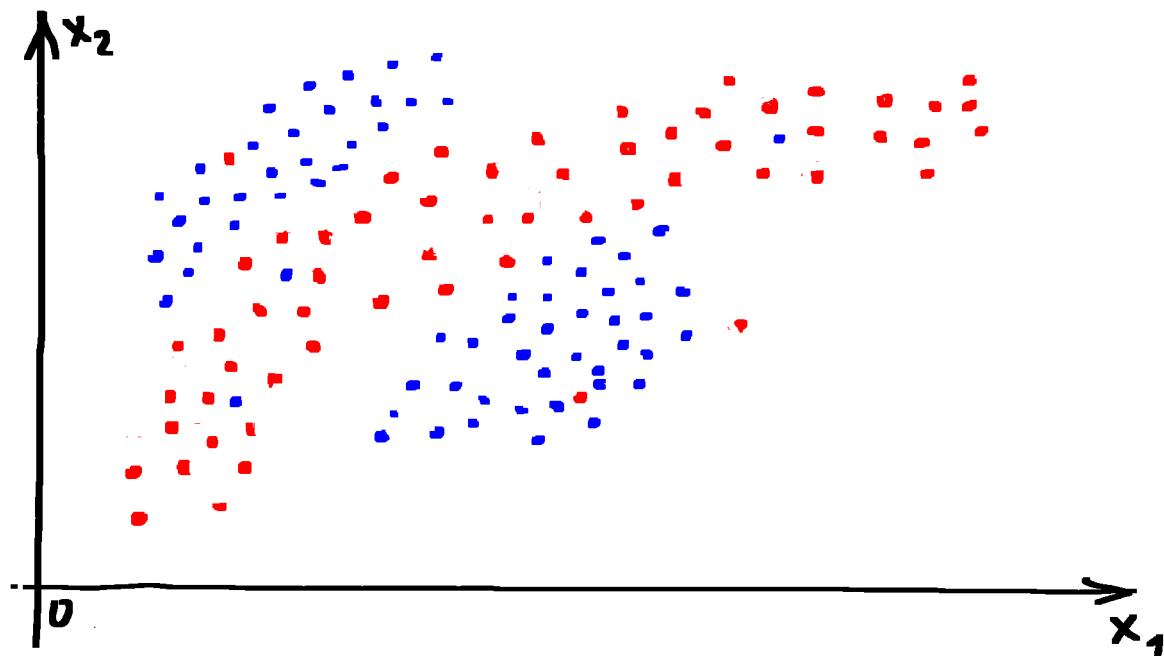
Нейронные сети состоят из нейронов.

Случайный лес (Random Forest) и XGBoost состоят из деревьев.

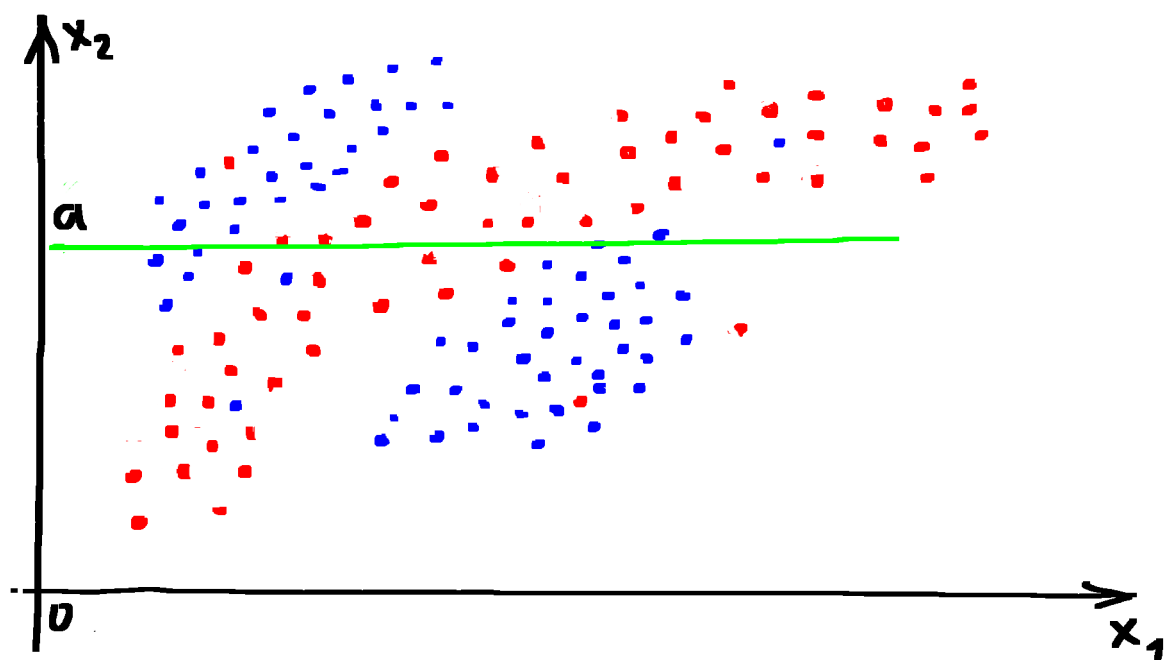
Кроме того, деревья легко интерпретируются. В некоторых ситуациях ради понимания можно отказаться от максимального качества метода.

Алгоритм построения дерева классификации на примере.

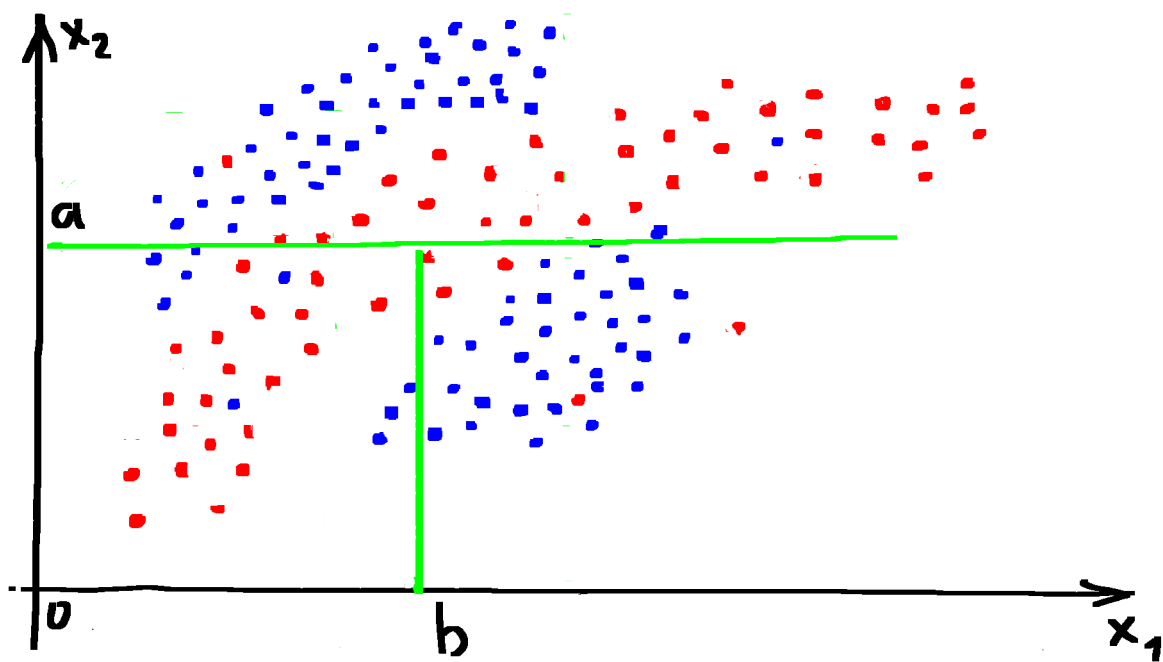
Первый вариант представления дерева классификации. Графический.



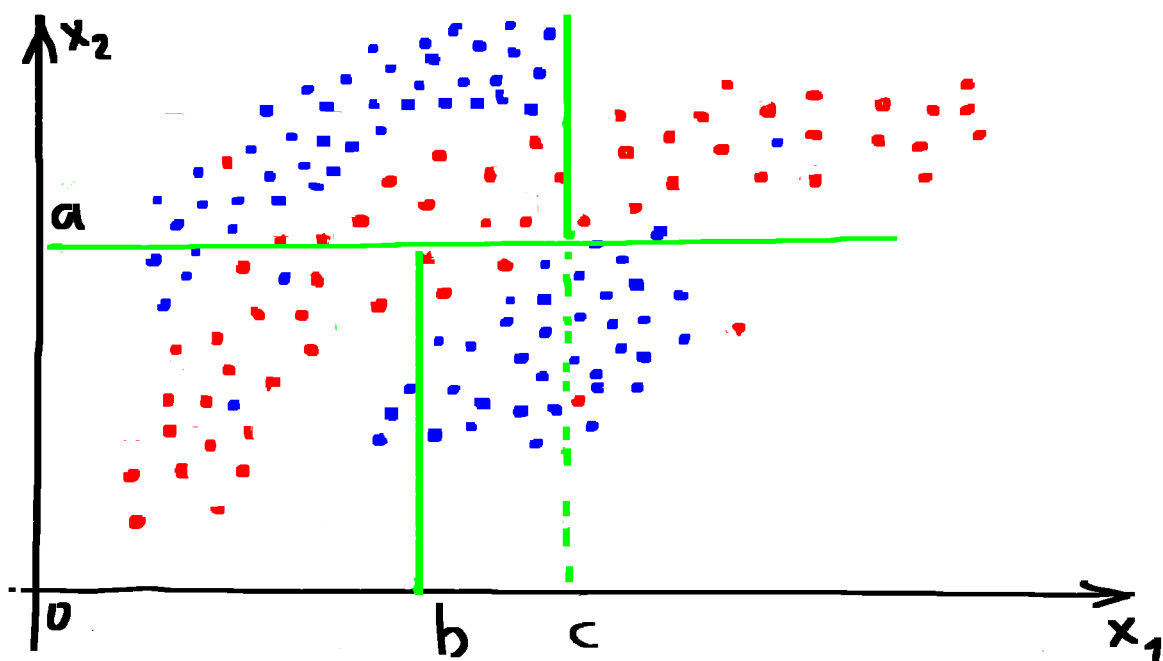
Шаг 0. Исходные данные, два класса



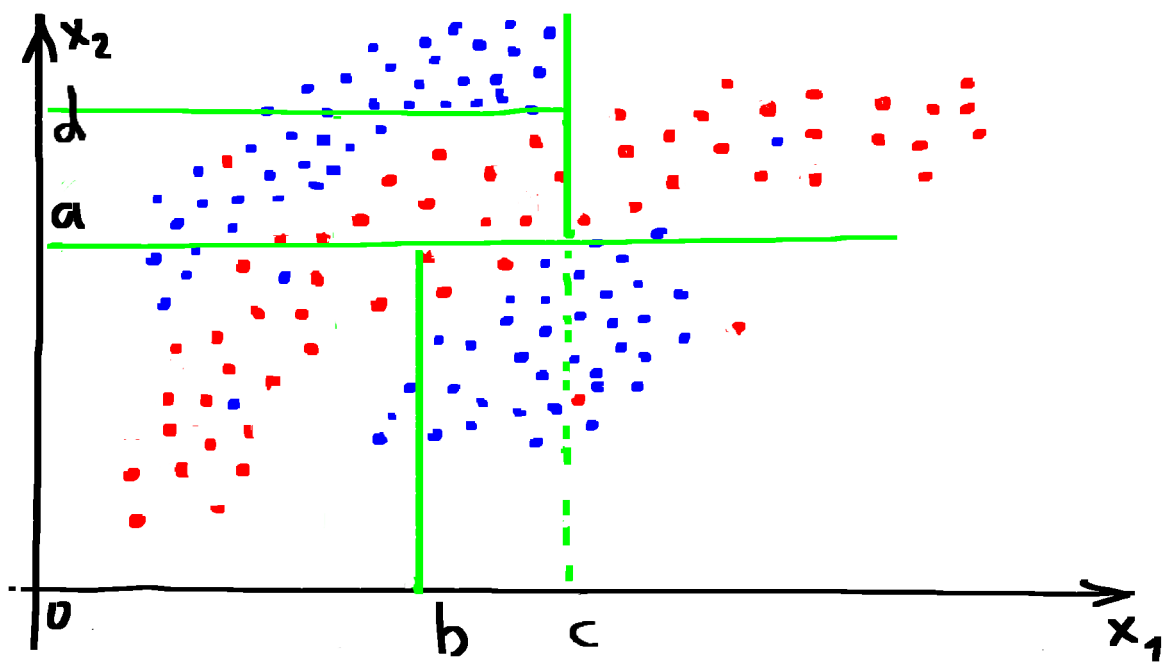
Шаг 1. Первое разделение (но процедура выберет другую)



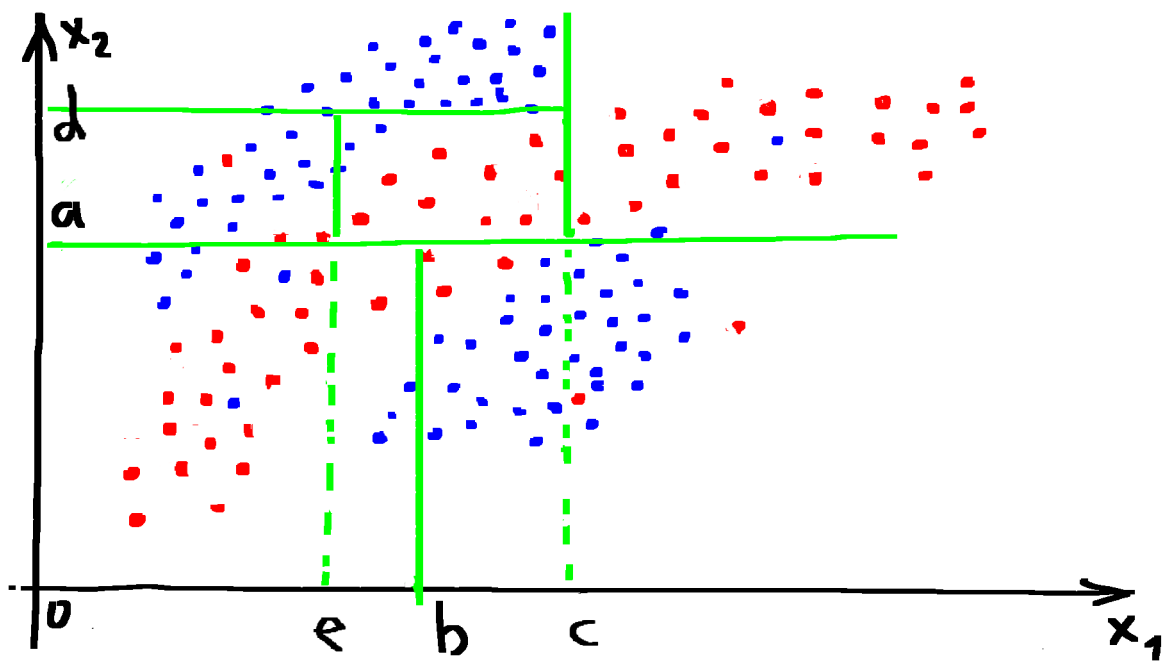
Шаг 2. Второе разделение. Делится только нижняя полуплоскость.



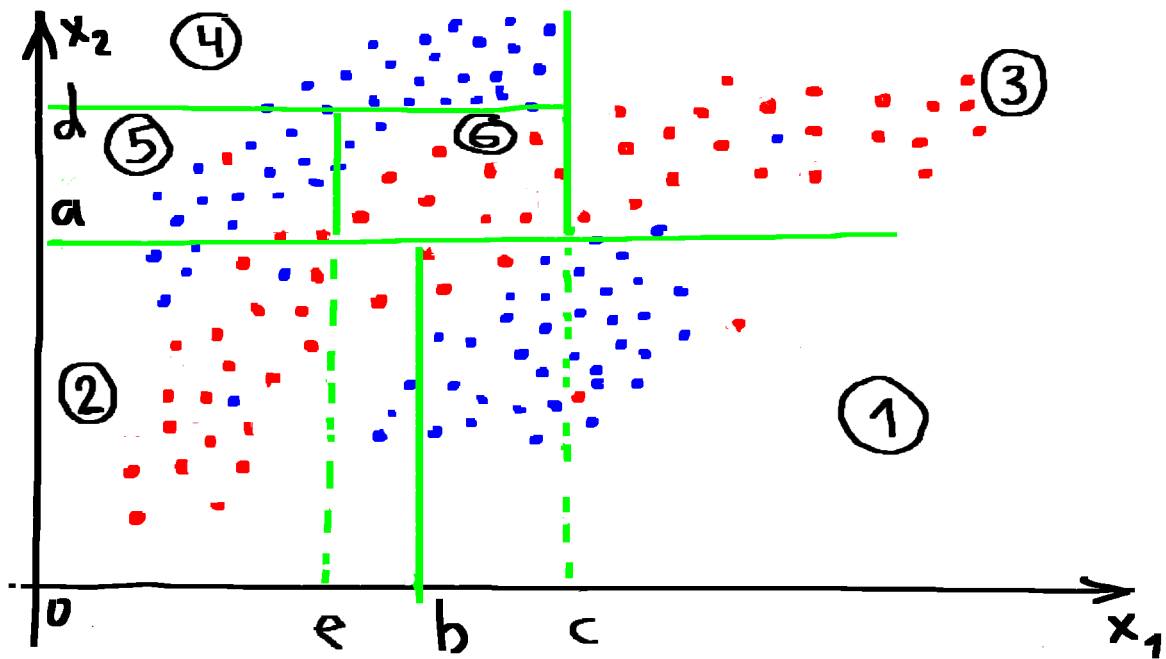
Шаг 3. Третье разделение. Делится только верхняя полуплоскость.



Шаг 4. Четвертое разделение.



Шаг 5. Пятое разделение. Пожалуй, дальнейшие разделения уже не нужны. (Почему?)



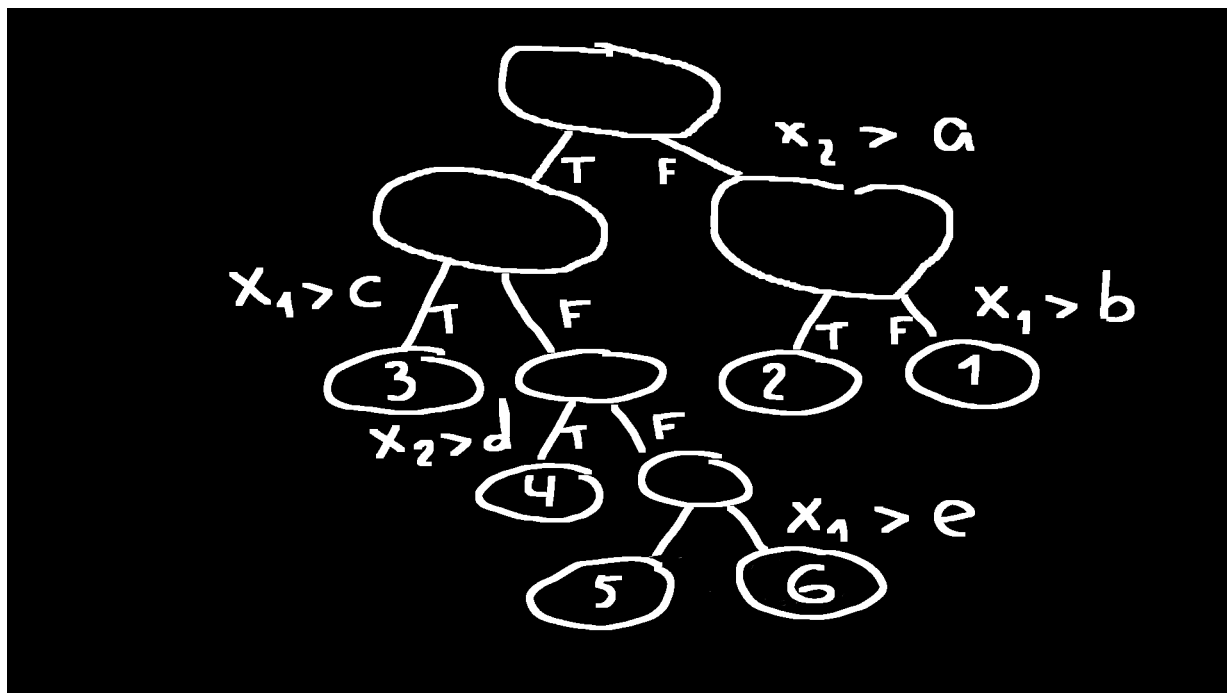
Шаг 6. Для дальнейшего удобно обозначить каждый прямоугольник.

Второй вариант представления дерева классификации. В виде логических правил.

- 1 $(x_1 > b) \cap (x_2 < a) \Rightarrow \text{синий класс}$
- 2 $(x_1 < b) \cap (x_2 < a) \Rightarrow \text{красный класс}$
- 3 $(x_1 > c) \cap (x_2 > a) \Rightarrow \text{красный класс}$
- 4 $(x_1 < c) \cap (x_2 > d) \Rightarrow \text{красный класс}$
- 5 $(x_1 < e) \cap (x_2 > a) \cap (x_2 < d) \Rightarrow \text{красный класс}$

Задача: Правило для шестого прямоугольника записать самостоятельно!

Третий вариант представления дерева классификации. Из которого становится понятно, причем тут деревья...



Основные понятия

узел (node)
 родительский (parent node)
 потомок (child node)
 расщепление узла (node splitting)
 конечный узел (лист дерева) (final node)
 пороговое значение (threshold)

Узел — подмножество объектов из обучающей выборки.

Алгоритм построения дерева классификации

Impurity measures (Меры загрязненности)

Обозначения

Пусть в задаче k классов.

Обозначим p_j - вероятность принадлежать классу j .

На практике p_j - доля объектов класса j в узле.

Энтропия (entropy)

$$H_1 = - \sum_{j=1}^k p_j \cdot \log_2 p_j$$

Индекс Джини (Gini index)

$$H_2 = 1 - \sum_{j=1}^k p_j^2 = \sum_{j=1}^k p_j \cdot (1 - p_j)$$

Ошибки классификации (classification error)

$$H_3 = 1 - \max(p_j)$$

Когда узел родитель разделяется на 2 узла потомка.

Тогда очищение (увеличение чистоты узлов) измеряется посредством

$$\Delta H = H_{\text{родителя}} - \left(\frac{n_{\text{левый}}}{n_{\text{родителя}}} \cdot H_{\text{левый}} + \frac{n_{\text{правый}}}{n_{\text{родителя}}} \cdot H_{\text{правый}} \right)$$

CART строит только бинарные деревья (что значит бинарное дерево)

Кроме CART используется много других способов строить деревья классификации... Два популярных метода

C4.5 — алгоритм для построения деревьев решений, разработанный Квинланом (John Ross Quinlan).

C4.5 улучшает алгоритм **ID3** того же автора.

Метод когда-то был самым популярным, но автор сделал процедуру платной...

С того момента и по сегодняшний день самый популярный метод CART. Но тучи сгущаются. Фармацевтам и банкирам не нравится нестабильность деревьев (подробнее ниже)...

CHAID (Chi-square automatic interaction detection)

Метод допускает не бинарные расщепления, это хорошо. Расщепления часто оказываются несбалансированными (много узлов с малым числом элементов). Это плохо.

oblique trees

пространство разделяется произвольными гиперплоскостями.

Снято требование, чтобы гиперплоскость была перпендикулярна одной из осей координат

Пакет R, реализующий процедуру oblique.tree был медленным и глючным, когда мы тестировали его 10 лет назад...

Пакет Python, реализующий процедуру oblique.tree отсутствует...

oblivious trees

В пределах одного слоя дерева все расщепления производятся по одной переменной.

Пакет R, реализующий процедуру: отсутствует (?)

Говорят, раньше на их основе был реализован в Matrixnet...

Говорят, теперь поиск Яндекса теперь опирается на Deep Learning...

Обрезание деревьев (pruning)

уменьшаем размер дерева и препятствуем переподгонке.

first train the Decision Tree without restrictions, then pruning (deleting) unnecessary nodes.

A node whose children are all leaf nodes is considered unnecessary if the purity improvement it provides is not statistically significant.

Standard statistical tests, such as the χ^2 test, are used to estimate the probability that the improvement is purely the result of chance (which is called the null hypothesis). If this probability, called the p-value, is higher than a given threshold (typically 5%, controlled by a hyperparameter), then the node is considered unnecessary and its children are deleted. The pruning continues until all unnecessary nodes have been pruned.

Проблемы

1 Несбалансированные классы в выборке. Надо балансировать...

Достоинства

0. Результаты работы метода хорошо интерпретируются.

1. Метод CART не требует никаких предположений о вероятностном распределении переменных. Вообще никаких предположений нет! Если предположений нет, то их нельзя нарушить...

2. Метод CART самостоятельно отбирает информативные переменные. В частности, если переменная не используется при построении дерева, то она неинформативна. Хотя все не так просто.

3. Дерево строится быстро. Особенно при сравнении с другими подходами.

Но есть исключение. Если предиктор измерен в номинальной шкале и может принимать в узле k значений, то этот узел можно разбить на две части $2^{(k-1)} - 1$ способами.

Поэтому большие значения k недопустимы на практике. Если переменные измерены в порядковой или количественной шкале, то ограничений нет.

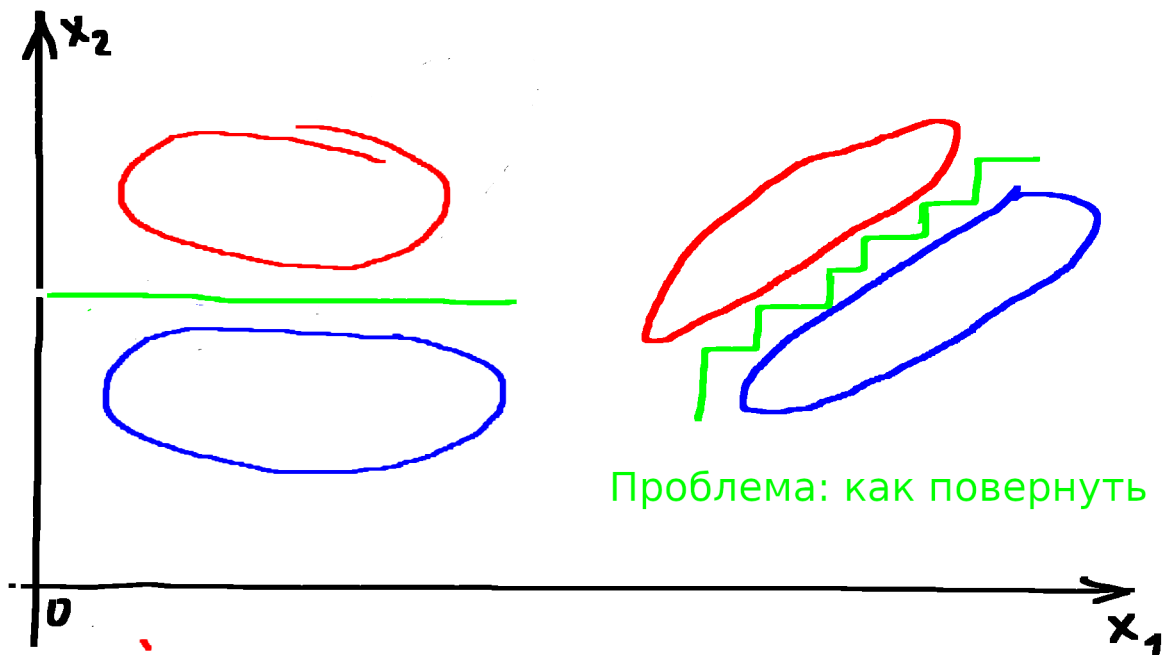
Построение дерева можно еще ускорить за счет binning(?), другое название гистограмма. Но это ускорение скорее важно для комбинаций деревьев (XGBoost, Light GBM), когда строятся сотни или тысячи деревьев.

4. Метод устойчив (robust) к монотонным преобразованиям предикторов. Как следствие, стандартизация предикторов не нужна.
5. Метод устойчив (robust) к выбросам.

Недостатки

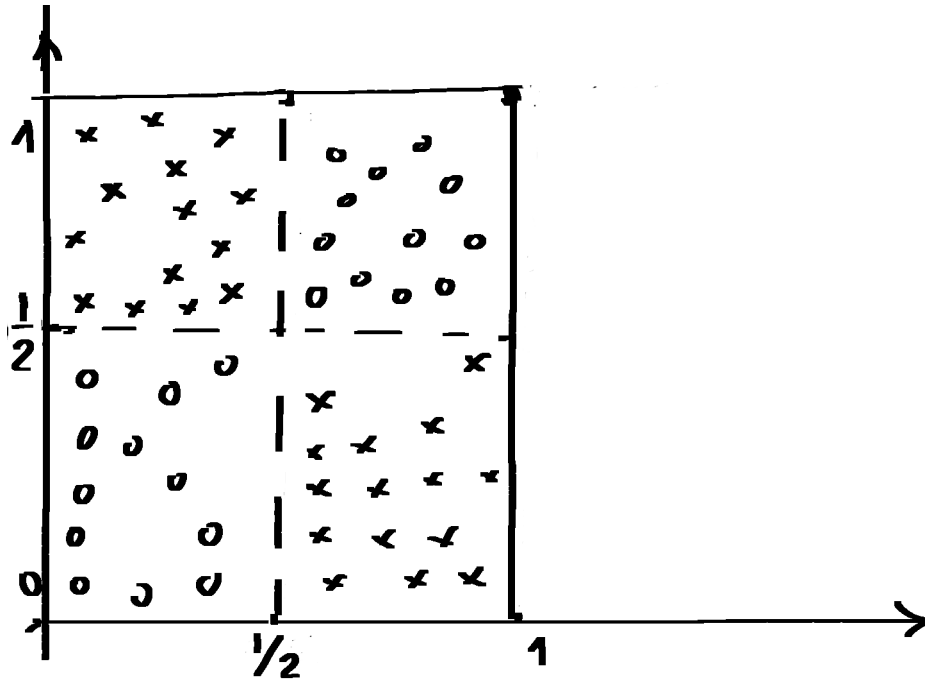
1. CART Нестабилен к изменениям в обучающей выборке. Деревья, построенные на разных выборках могут внешне сильно отличаться (дерево может увеличиваться, уменьшаться, включать другие предикторы и т.д..) Нестабильность осложняет валидацию модели. Хотя качество этих разных моделей будет примерно одинаковым. Поэтому в некоторых ситуациях CART “запрещен”
Как такое может случаться? Когда два расщепления очищают почти одинаково. Для одной выборки лучше окажется одно расщепление, для другой — второе.

2. CART Неустойчив к поворотам данных



3. Процедуры питона `DecisionTreeRegressor` не умеют работать с пропусками (используйте `gpart` из R)

4. CART жадный алгоритм. Но нахождение оптимального дерева является NP-полной задачей: его построение требует $O(\exp(m))$ времени, где m – число узлов. Заметим, что при анализе данных используют множество жадных алгоритмов, у которых хорошая репутация. Например кластеризация методом к-средних (k-means)



Технические детали

1. Как изменить алгоритм, чтобы метод определял не код класса, а вероятность принадлежать классу?

2. Зачем в питоновских реализациях CART задают зерно датчика случайных чисел?

Во первых, может быть так, что два расщепления одинаково уменьшают загрязненность. Тогда выбираем расщепление «подбрасывая монетку»

Расщепление делаем случайным. См. Параметр процедуры

splitter : string, optional (default="best")

*The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the **best random split**.*

Моя рекомендация: используйте значение опции best. Если обучается долго, предварительно проведите binning.

Regression Tree (CART) в Python

Деревья регрессии — приближение кривой кусочно постоянной функцией

Обозначим

m – номер узла

N_m - число наблюдений в узле m

y_i - значения отклика для наблюдений, попавших в узел

Новому наблюдению, попавшему в узел, сопоставляется не класс, а значение отклика

$$\bar{y}_m = \frac{1}{N_m} \cdot \sum_{i \in \text{узел}_m} y_i$$

Impurity measures (Меры загрязненности) в задачах регрессии

$$H_2(\text{узел}_m) = \frac{1}{N_m} \cdot \sum_{i \in \text{узел}_m} (y_i - \bar{y}_m)^2$$

$$H_1(\text{узел}_m) = \frac{1}{N_m} \cdot \sum_{i \in \text{узел}_m} |y_i - \bar{y}_m|$$