

Введение в искусственный интеллект. Машинное обучение

Тема: Решающие деревья. Случайный лес

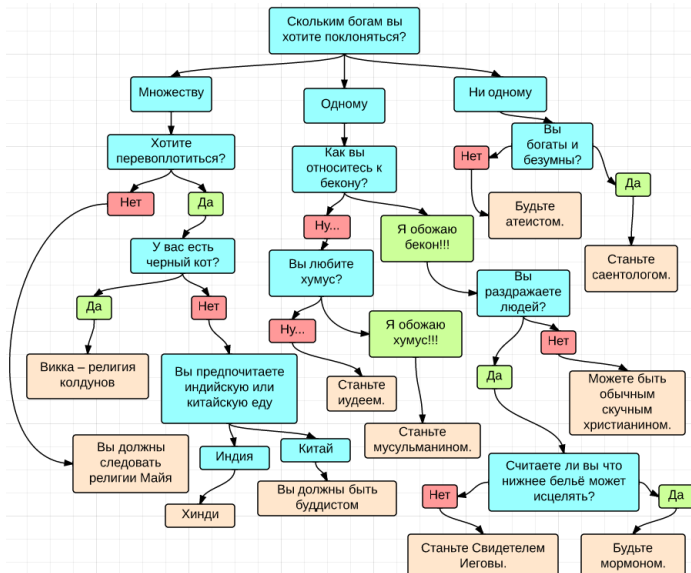
Бабин Д.Н., Иванов И.Е.

кафедра Математической Теории Интеллектуальных Систем



1. Случайные деревья
 - ID3 (Iterative Dichotomiser 3)
 - CART (Classification And Regression Tree)
2. Случайный лес

Примеры деревьев из жизни



КАК ВСЁ ПОЧИНИТЬ

ЭТО ДВИГАЕТСЯ?



Скрипт разговора оператора



- Инструкции любого call-центра

Примеры деревьев из жизни

- Инструкции любого call-центра
- Постановка медицинского диагноза

Примеры деревьев из жизни

- Инструкции любого call-центра
- Постановка медицинского диагноза
- Многие алгоритмы имеют древовидную структуру

Примеры деревьев из жизни

- Инструкции любого call-центра
- Постановка медицинского диагноза
- Многие алгоритмы имеют древовидную структуру

Достоинства

Главные достоинства дерева — интерпретируемость и простота

Определение

Решающее дерево — алгоритм машинного обучения, задающийся деревом со следующими свойствами:

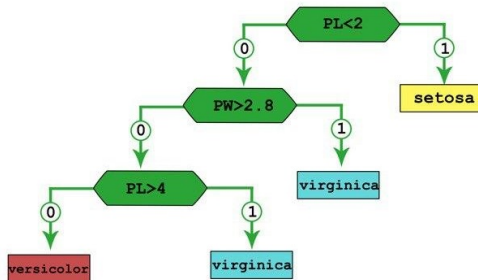
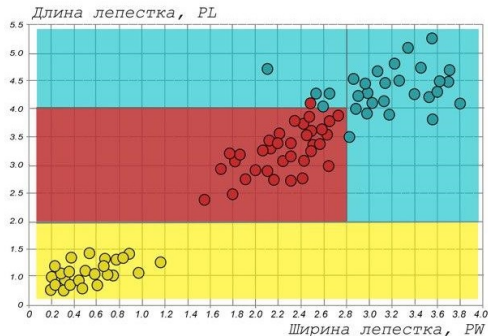
- 1 Выделена вершина — корень дерева
- 2 Листовой вершине (у которой нет потомков) соответствует некоторый ответ алгоритма $y \in Y$
- 3 Каждой внутренней вершине соответствует предикат. Каждому ребру из внутренней вершины соответствует некоторое значение предиката



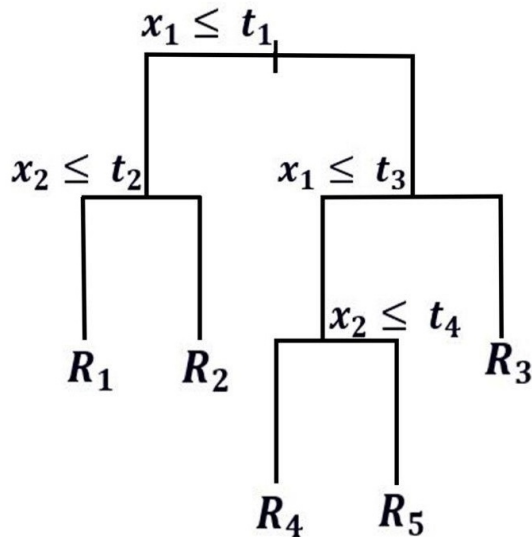
Пример решающего дерева для задачи классификации

Ирисы Фишера

Задача Фишера о классификации цветков ириса на 3 класса



Пример решающего дерева для задачи регрессии



Часто используемые виды предикатов

- Пороговое условие

Часто используемые виды предикатов

- Пороговое условие
- Конъюнкция пороговых условий

Часто используемые виды предикатов

- Пороговое условие
- Конъюнкция пороговых условий
- Синдром — выполнение какого-то минимального количества условий

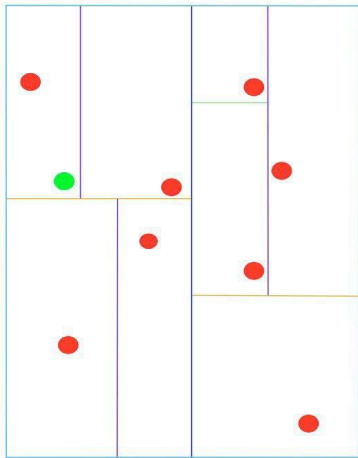
Часто используемые виды предикатов

- Пороговое условие
- Конъюнкция пороговых условий
- Синдром — выполнение какого-то минимального количества условий
- Полуплоскость — линейная пороговая функция

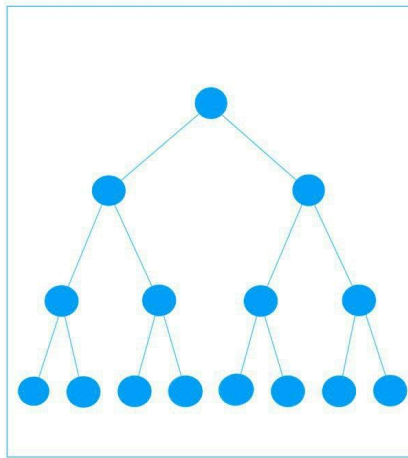
Часто используемые виды предикатов

- Пороговое условие
- Конъюнкция пороговых условий
- Синдром — выполнение какого-то минимального количества условий
- Полуплоскость — линейная пороговая функция
- Шар — пороговая функция близости

K-мерное дерево для kNN



Kd-Tree in 2D



Multiple Randomized Kd-Trees

Задача

Построение дерева наименьшего размера, которое не ошибается

Задача

Построение дерева наименьшего размера, которое не ошибается

Построение дерева определенного размера с минимальным количеством ошибок

Построение дерева

Задача

Построение дерева наименьшего размера, которое не ошибается

Построение дерева определенного размера с минимальным количеством ошибок

Проблема

Построение дерева наименьшего размера, которое не ошибается — NP - сложная задача

Построение дерева

Задача

Построение дерева наименьшего размера, которое не ошибается

Построение дерева определенного размера с минимальным количеством ошибок

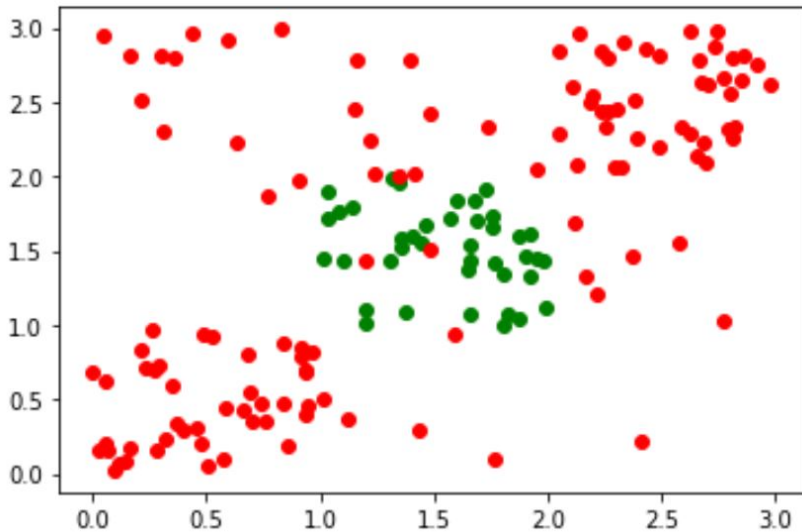
Проблема

Построение дерева наименьшего размера, которое не ошибается — NP - сложная задача

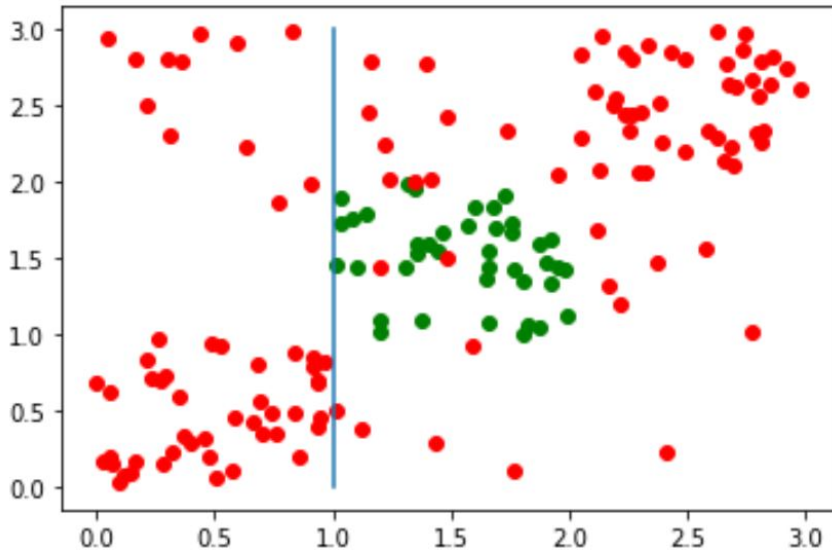
Решение

Итеративные жадные алгоритмы

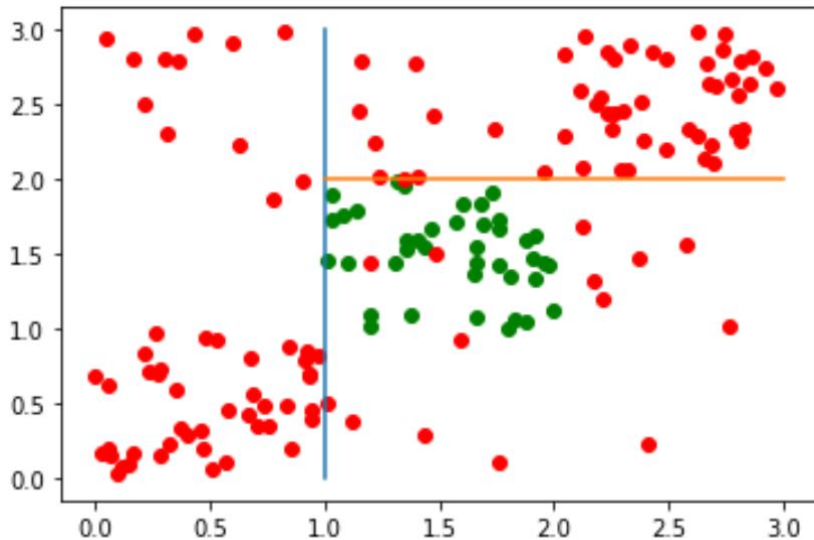
Пример построения дерева



Пример построения дерева



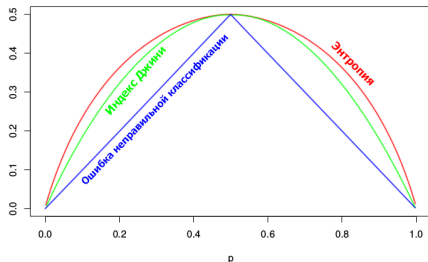
Пример построения дерева



Критерии информативности для задачи двухклассовой классификации

Пусть p — частота встречаемости одного из классов

| | |
|----------------------------|-------------------------------|
| Ошибка классификации | $1 - \max(p, 1 - p)$ |
| Индекс Джинни | $2p(1 - p)$ |
| Кросс-энтропийный критерий | $-p \log p - (1-p) \log(1-p)$ |

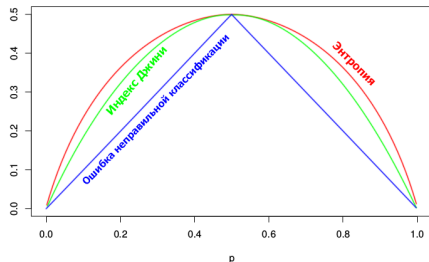


- Критерии довольно похоже
- Последние два критерия — дифференцируемые

Критерии информативности для задачи двухклассовой классификации

Пусть p — частота встречаемости одного из классов

| | |
|----------------------------|-------------------------------|
| Ошибка классификации | $1 - \max(p, 1 - p)$ |
| Индекс Джинни | $2p(1 - p)$ |
| Кросс-энтропийный критерий | $-p \log p - (1-p) \log(1-p)$ |



- Критерии довольно похоже
- Последние два критерия — дифференцируемые
- Есть по 400 наблюдений из двух классов. Вопрос: какое разделение лучше (300, 100) и (100, 300) против (200, 400) и (200, 0)?

Критерии информативности для задачи многоклассовой классификации

Определения

Пусть R_m — некоторая часть пространства, содержащая N_m объектов.

Пусть $\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} [y_i = k]$.

$k(m) = \arg \max_k \hat{p}_{mk}$ — мажоритарный класс

| | |
|----------------------------|--|
| Ошибка классификации | $\frac{1}{N_m} \sum_{i: x_i \in R_m} [y_i \neq k(m)] = 1 - \hat{p}_{mk(m)}$ |
| Индекс Джинни | $\sum_{k \neq k'} \hat{p}_{mk} \hat{p}_{mk'} = \sum_k \hat{p}_{mk} (1 - \hat{p}_{mk})$ |
| Кросс-энтропийный критерий | $-\sum_k \hat{p}_{mk} \log \hat{p}_{mk}$ |



Дано

U — множество объектов, \mathcal{B} — множество предикатов, $I(U)$ — критерий информативности.

Ветвление

Тогда для предиката β множество U разбивается на U_0 и U_1 .
Определим предикат с максимальной информативностью:

$$\beta = \arg \max_{\beta \in \mathcal{B}} I(\beta, U),$$

где $I(\beta, U) = \frac{|U_0|}{|U|} I(U_0) + \frac{|U_1|}{|U|} I(U_1)$.



Вход

U — обучающее множество объектов

\mathcal{B} — множество предикатов

def ID3(U):

- 1 Если все объекты из U лежат в одном классе, то отбить новый лист с меткой этого класса и выйти
- 2 Найти максимально информативный предикат $\beta = \arg \max_{\beta \in \mathcal{B}} I(\beta, U)$ и разбить выборку на 2 части $U = U_0 \cup U_1$
- 3 Если $U_0 = \emptyset$ или $U_1 = \emptyset$, то отбить лист с меткой мажоритарного класса и выйти
- 4 Запустить ID3 для U_0 и U_1 .



Преимущества и недостатки алгоритма ID3

Преимущества

Преимущества

- Простота реализации

Преимущества

- Простота реализации
- Хорошая интерпретируемость

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки
- Алгоритм легко поддаётся многочисленным усовершенствованиям

Преимущества и недостатки алгоритма ID3

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки
- Алгоритм легко поддаётся многочисленным усовершенствованиям

Недостатки

- Жадность. Локальный выбор оптимального предиката не является глобальным



Преимущества и недостатки алгоритма ID3

Преимущества

- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки
- Алгоритм легко поддаётся многочисленным усовершенствованиям

Недостатки

- Жадность. Локальный выбор оптимального предиката не является глобальным
- Большая вариация алгоритма, при небольших изменениях в данных структура дерева полностью меняется



Преимущества и недостатки алгоритма ID3

Преимущества

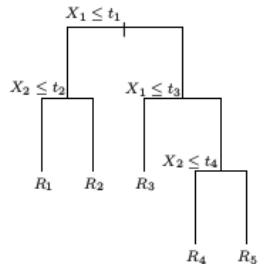
- Простота реализации
- Хорошая интерпретируемость
- Сложность алгоритма линейна по длине выборки
- Алгоритм легко поддаётся многочисленным усовершенствованиям

Недостатки

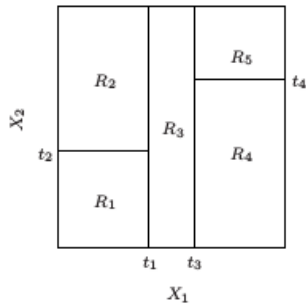
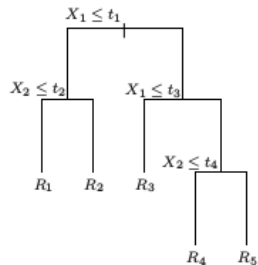
- Жадность. Локальный выбор оптимального предиката не является глобальным
- Большая вариация алгоритма, при небольших изменениях в данных структура дерева полностью меняется
- Алгоритм склонен к переобучения, так как усложняет структуру дерева



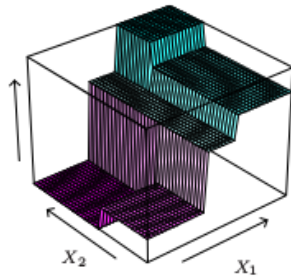
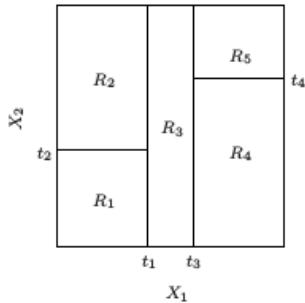
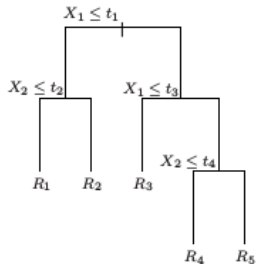
Алгоритм построения дерева для регрессии



Алгоритм построения дерева для регрессии

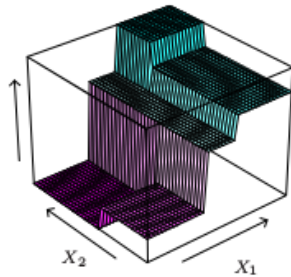
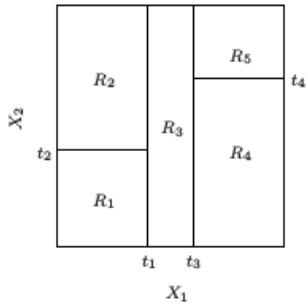
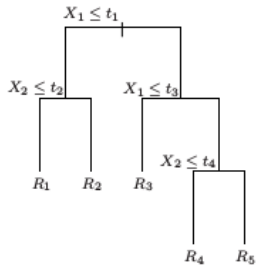


Алгоритм построения дерева для регрессии



$$f(x) = \sum_{m=1}^M c_m I(x \in R_m)$$

Алгоритм построения дерева для регрессии



$$f(x) = \sum_{m=1}^M c_m I(x \in R_m)$$
$$c_m = \text{ave}(y_i | x_i \in R_m)$$

Алгоритм построения дерева для регрессии

$$f(x) = \sum_{m=1}^M c_m I(x \in R_m)$$
$$c_m = \text{ave}(y_i | x_i \in R_m)$$

Ветвление

Пусть $R_1(j, s) = \{X | X_j \leq s\}$ и $R_2(j, s) = \{X | X_j > s\}$

Тогда на каждом шаге будем решать задачу минимизации

$$\min_{j,s} \left(\min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_2)^2 \right)$$

Проблема

Деревья обученные жадным алгоритмом склонны к переобучению

Проблема

Деревья обученные жадным алгоритмом склонны к переобучению

Решение

Добавление различных условий на сложность дерева, редукция решающих деревьев

Определения

Пусть T — некоторое поддереву дерева T_0 . Обозначим:

- $|T|$ — количество листов в T
- $N_m = \#\{x_i \in R_m\}$
- $c_m = \frac{1}{N_m} \sum_{x_i \in R_m} y_i$
- $Q_m(T) = \frac{1}{N_m} \sum_{x_i \in R_m} (y_i - c_m)^2$

Стратегия усечения дерева

$$C_\alpha(T) = \sum_{m=1}^{|T|} N_m Q_m(T) + \alpha |T|$$

- Максимальная глубина дерева (`max_depth`)

Регуляризация деревьев в scikit-learn

- Максимальная глубина дерева (`max_depth`)
- Минимальное число наблюдений для ветвления дерева (`min_samples_split`)



- Максимальная глубина дерева (`max_depth`)
- Минимальное число наблюдений для ветвления дерева (`min_samples_split`)
- Минимальное число наблюдения в листе (`min_samples_leaf`)



- Максимальная глубина дерева (`max_depth`)
- Минимальное число наблюдений для ветвления дерева (`min_samples_split`)
- Минимальное число наблюдения в листе (`min_samples_leaf`)
- Максимальное количество признаков, используемых для деления (`max_features`)



- Максимальная глубина дерева (`max_depth`)
- Минимальное число наблюдений для ветвления дерева (`min_samples_split`)
- Минимальное число наблюдения в листе (`min_samples_leaf`)
- Максимальное количество признаков, используемых для деления (`max_features`)
- Максимальное число листьев (`max_leaf_nodes`)



- Максимальная глубина дерева (`max_depth`)
- Минимальное число наблюдений для ветвления дерева (`min_samples_split`)
- Минимальное число наблюдения в листе (`min_samples_leaf`)
- Максимальное количество признаков, используемых для деления (`max_features`)
- Максимальное число листьев (`max_leaf_nodes`)
- Минимальное изменение критерия информативности (`min_impurity_decrease`)





Основная идея

Решающие деревья не устойчивы к небольшим изменениям данных. Поэтому при обучении на различных подвыборках решающие деревья будут ошибаться на разных объектах.

Основная идея

Решащие деревья не устойчивы к небольшим изменениям данных. Поэтому при обучении на различных подвыборках решающие деревья будут ошибаться на разных объектах.

Определение

- Обучение каждого решающего дерева происходит на сгенерированной случайной подвыборке с повторениями размера обучающей выборки
- Обучение происходит на случайной подвыборке признаков (их количество — входной параметр алгоритма, для классификации берут \sqrt{n} , для регрессии $\frac{n}{3}$)
- Дерево строится до полного исчерпания подвыборки и не подвергается процедуре прунинга
- Решающее правило $a(x) = \frac{1}{T} \sum_{t=1}^T a_t(x)$

Важность признаков

- При построении случайного леса считается количество вершин, в которых используется признак с весом (количество наблюдений в вершине)

Важность признаков

- При построении случайного леса считается количество вершин, в которых используется признак с весом (количество наблюдений в вершине)
- Признак тем важнее, чем большее разницу между метрикой качества на обычных данных и на данных, где этот признак случайно перемешан

Преимущества и недостатки случайного леса

Преимущества

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели
- Внутренняя оценка способности к обобщению

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели
- Внутренняя оценка способности к обобщению
- Высокая параллелизуемость и масштабируемость

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели
- Внутренняя оценка способности к обобщению
- Высокая параллелизуемость и масштабируемость

Недостатки

- Большой размер и сложность получающихся моделей

Преимущества и недостатки случайного леса

Преимущества

- Способность эффективно обрабатывать данные с большим числом признаков и классов
- Нечувствительность к масштабированию значений признаков
- Одинаково хорошо обрабатываются как непрерывные, так и дискретные признаки
- Существуют методы оценивания значимости отдельных признаков в модели
- Внутренняя оценка способности к обобщению
- Высокая параллелизуемость и масштабируемость

Недостатки

- Большой размер и сложность получающихся моделей
- Нулевая интерпретируемость — черный ящик

- Решающие деревья — хорошо интерпретируемый алгоритм машинного обучения

- Решающие деревья — хорошо интерпретируемый алгоритм машинного обучения
- На его основе строятся самые сильные модели машинного обучения (например, случайный лес)