Случайный лес — это классификатор, состоящий из классификаторов с древовидной структурой {h (x, Ѳk), k = 1, ...}, где {Ѳk} - независимые одинаково распределенные случайные векторы (тренировочные данные). Решение принимается на основе голосования, где каждое дерево дает единичный голос за самый популярный класс на входе x.

Основная схема построения:

* Повторяется k раз:
  + Для каждого дерева извлекается новый обучающий набор случайным образом с возвратом из данного, и подпространство признаков (использование бутстрэпа и случайного выбора признаков, что позволяет снизить коррелированность между деревьями и избежать переобучения)
  + По выбранной тренировочной выборке строится дерево решений (всего k штук) (в данном случае не происходит усечения деревьев)
* Усреднение предсказания (для задачи регрессии) или голосование (для классификации)

Оut-of-bag:

Каждое дерево в случайном лесе обучается по подмножеству объектов. Это значит, что те объекты, которые не вошли в тренировочную выборку, по сути являются контрольными для данного дерева. Значит, мы можем для каждого объекта найти деревья, которые были обучены без него (около 1/3 примеров остаются вне тренировочной выборки), и вычислить по их ответам ошибку. Этот метод оценки ошибки для случайного дерева называется «out-of-bag».

Основные параметры алгоритма:

* F - number of features – число признаков, по которым ищется разбиение (max\_features) (Причем различие между регрессией и классификацией состоит в том, что корреляция увеличивается довольно медленно по мере увеличения числа используемых признаков. Для классификации обычно используется max\_features = sqrt(M), а для регрессии M/3, где M - количество признаков)
* Кол-во деревьев в ансамбле (n\_estimators)
* L - количества переменных, которые необходимо объединить в случайную линейную комбинацию (для увеличения F при небольшом количестве входных данных)

Основные достоинства Случайного дерева:

* Хорошая точность (как у Adaboost) (т. к. деревья в ансамбле слабо коррелируемы)
* Устойчив к выбросу и шуму
* Быстрый (за счет обучения каждого дерева на части данных)
* Легкость организации параллельных вычислений
* Не переобучается
* Снимается задача усечения полного дерева решений

Недостатки:

* Не интерпретируемые модели
* Плохо работает на разреженных признаках
* Большой размер