Занятие № 9

Ансамблирование моделей

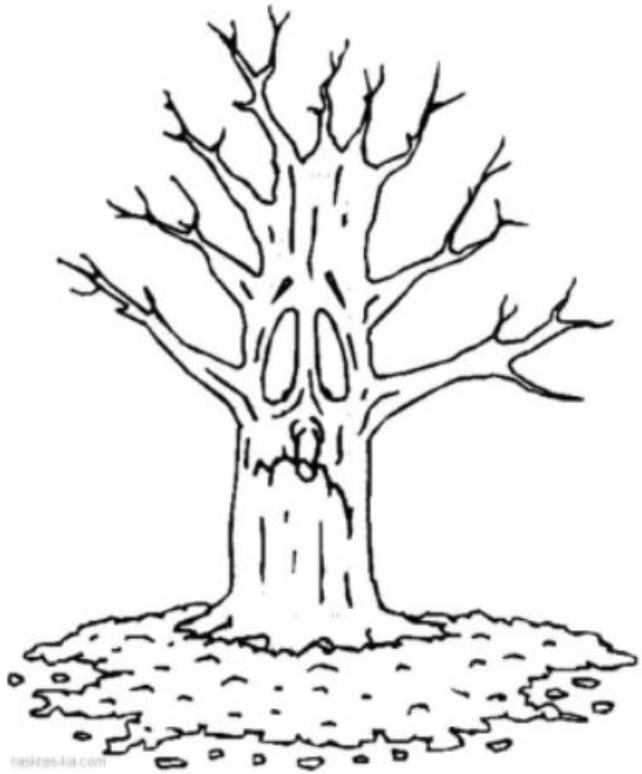
Содержание

Введение


Требуемый заем обеспечен
Нет
Да
Отказать в выдаче кредита
Наличие недвижимости
Нет
Да
Образование
Наличие страховки
специальное
среднее
высшее
Нет
Да
...
...
...
...
Выдать кредит

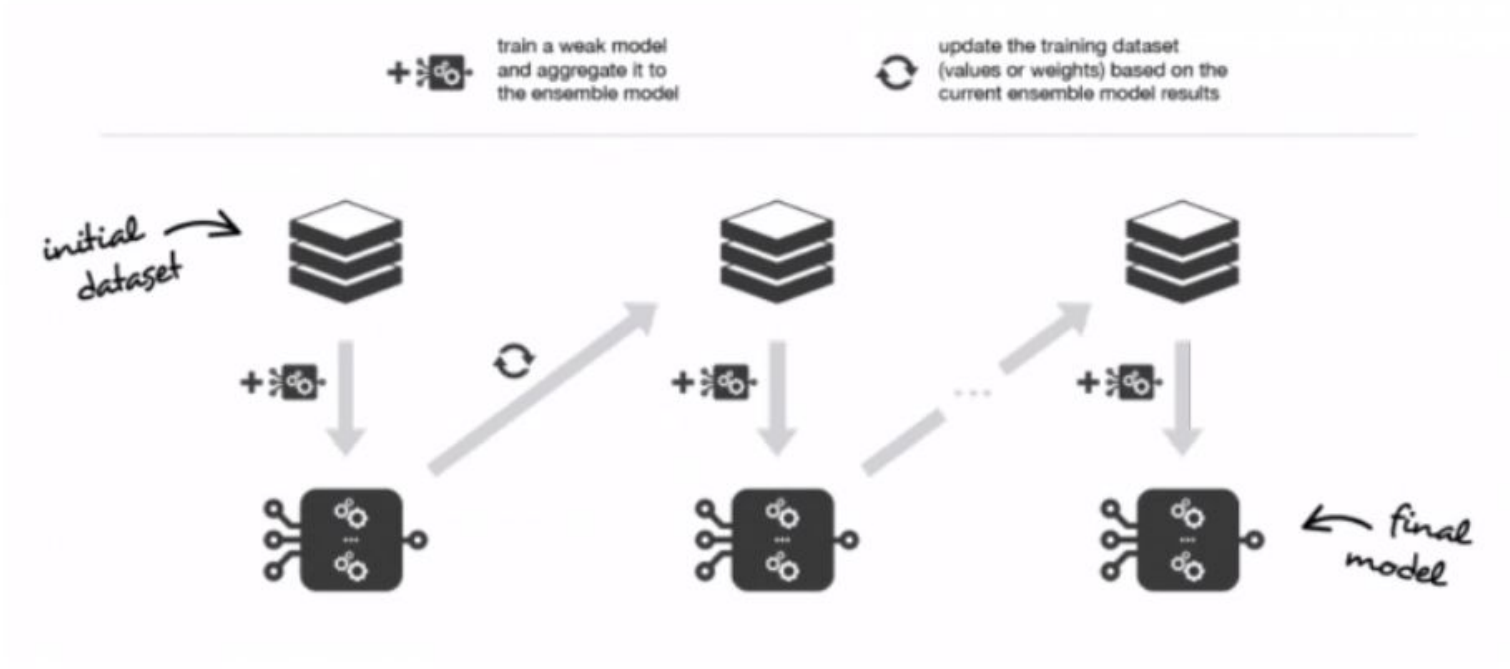
Что не так с деревьями?

1. Острая проблема переобучения
2. Неустойчивость
3. Не учитывает нелинейные зависимости или даже простые линейные, которые идут не по осям координат (f.e., представьте дерево для классификатора вида y>x)
4. Чувствителен к несбалансированным классам
5. Хорошо интерполирует, плохо экстраполирует



Звать ансамбль!

Звать ансамбль!

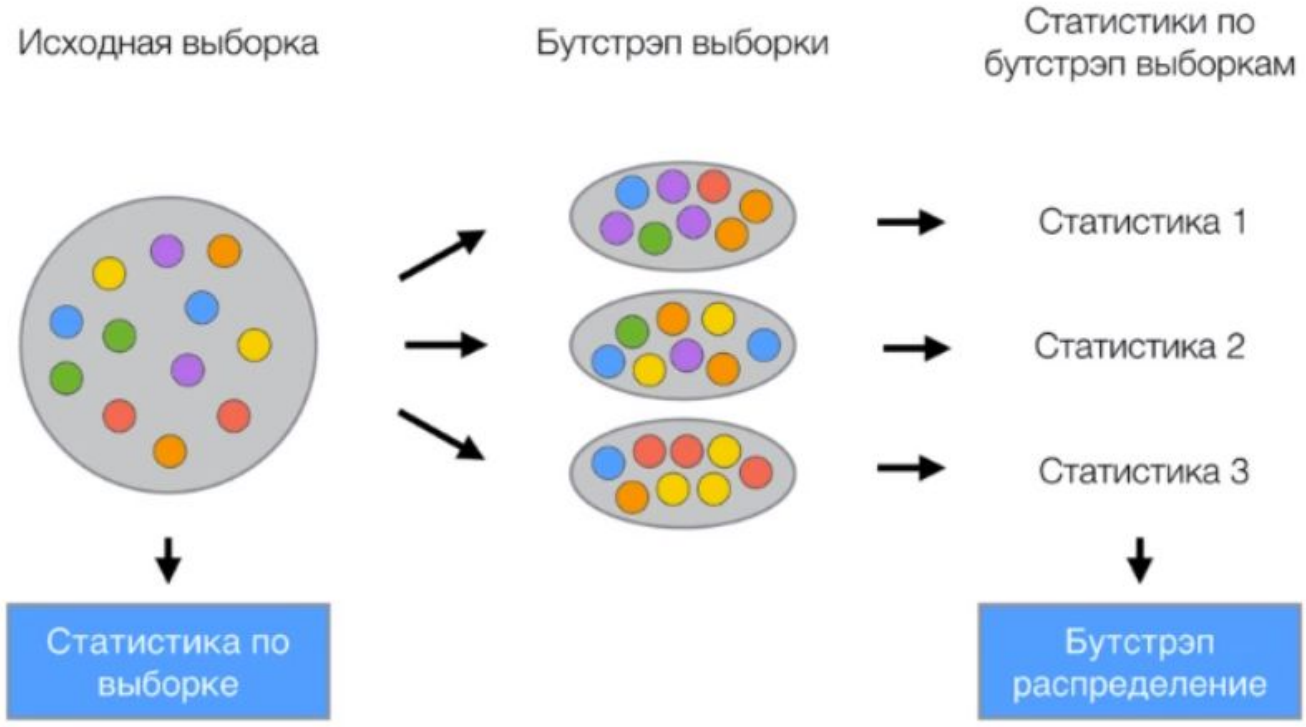


«Мудрость толпы»

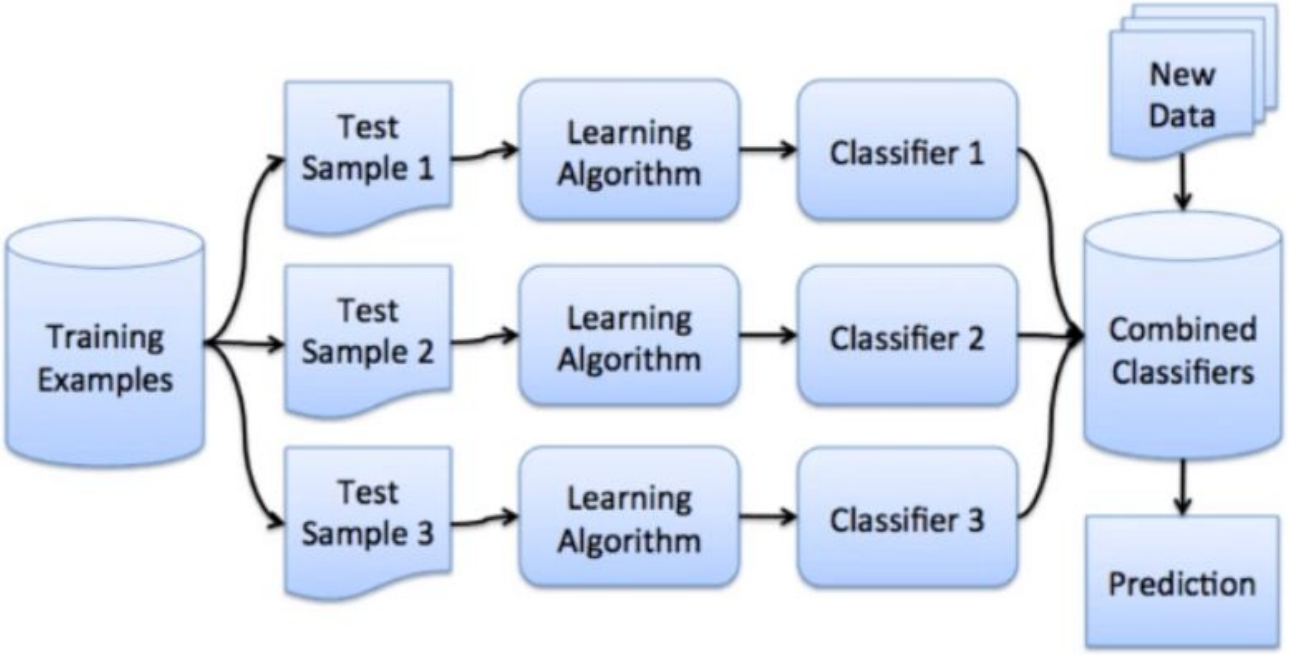
Фрэнсис Гальтон «Мудрость толпы»

Фрэнсис Гальтон в 1906 году посетил рынок, где проводилась некая лотерея для крестьян.
Их собралось около 800 человек, и они пытались угадать вес быка, который стоял перед ними. Бык весил 1198 фунтов. Ни один крестьянин не угадал точный вес быка, но если посчитать среднее от их предсказаний, то получим 1197 фунтов.
Эту идею уменьшения ошибки применили и в машинном обучении.

Бутстреп


Исходная выборка
Бутстрэп выборки
Статистики по бутстрэп выборкам
Статистика 1
Статистика 2
Статистика 3
Статистика по выборке
Бутстрэп распределение

Бэгинг


Training Examples
Test Sample 1
Test Sample 2
Test Sample 3
Learning Algorithm
Learning Algorithm
Learning Algorithm
Classifier 1
Classifier 2
Classifier 3
New Data
Combined Classifiers
Prediction

Бэгинг

$$\varepsilon_i(x) = b_i(x) - y(x), i = 1, \ldots, n$$

$$E_x(b_i(x) - y(x))^2 = E_x\varepsilon_i^2(x).$$

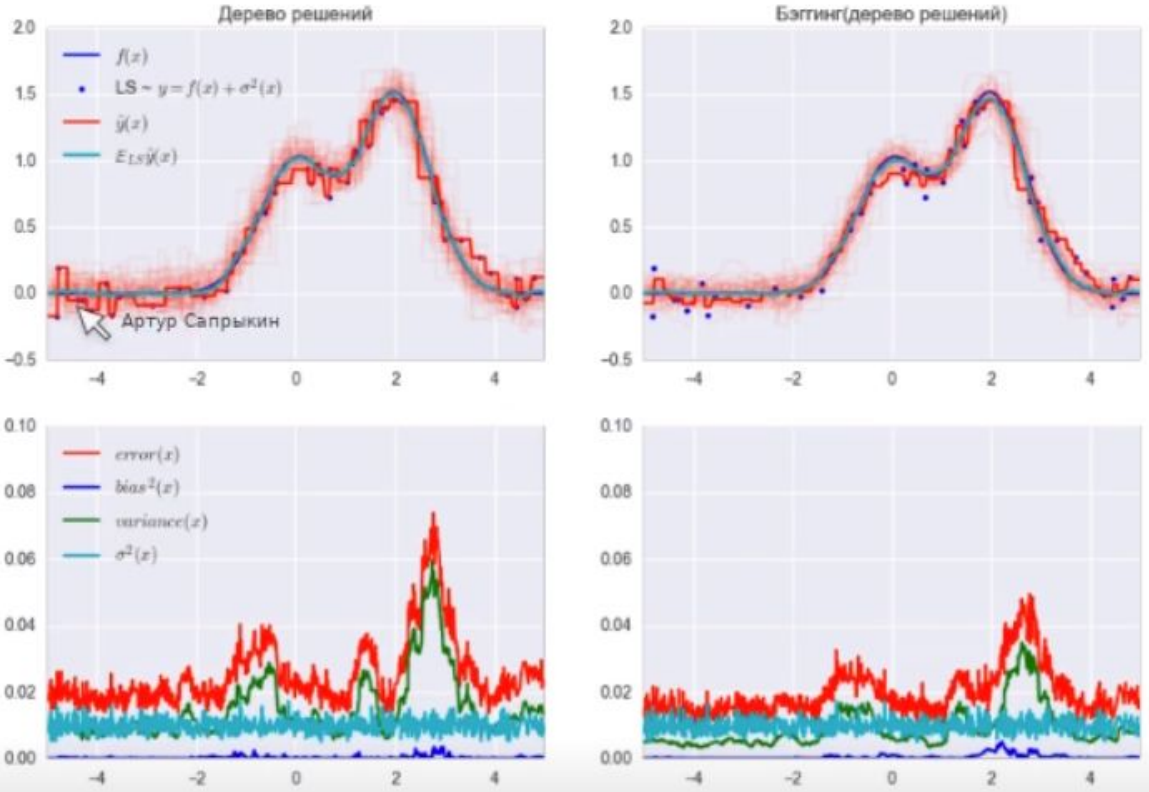
$$E_1 = \frac{1}{n}E_x\sum_{i=1}^{n}\varepsilon_i^2(x)$$

Бэгинг

$$a(x) = \frac{1}{n} \sum_{i=1}^{n} b_i(x)$$

$$\begin{aligned} E_n &= E_x \left(\tfrac{1}{n} \textstyle\sum_{i=1}^{n} b_i(x) - y(x) \right)^2 \\ &= E_x \left(\tfrac{1}{n} \textstyle\sum_{i=1}^{n} \varepsilon_i \right)^2 \\ &= \tfrac{1}{n^2} E_x \left(\textstyle\sum_{i=1}^{n} \varepsilon_i^2(x) + \sum_{i \neq j} \varepsilon_i(x) \varepsilon_j(x) \right) \\ &= \tfrac{1}{n} E_1 \end{aligned}$$

Бэгинг


Дерево решений
Бэггинг(дерево решений)
$f(x)$
LS ~ $y = f(x) + \sigma^2(x)$
$\hat{y}(x)$
$\mathcal{E}_{LS}\hat{y}(x)$
Артур Сапрыкин
$error(x)$
$bias^2(x)$
$variance(x)$
$\sigma^2(x)$

Бэгинг

from sklearn.ensemble import BaggingRegressor

base_estimator object or None – модель регрессии из sklearn (по умолчанию деревья решений)
n_estimators – количество моделей
max_samples int or float, optional (default=1.0) Количество сэмплов для обучения
max_features int or float, optional (default=1.0) Количество признаков для обучения

Случайный лес

Алгоритм построения случайного леса, состоящего из **N** деревьев, выглядит следующим образом:

Для каждого **n=1,...,N**:

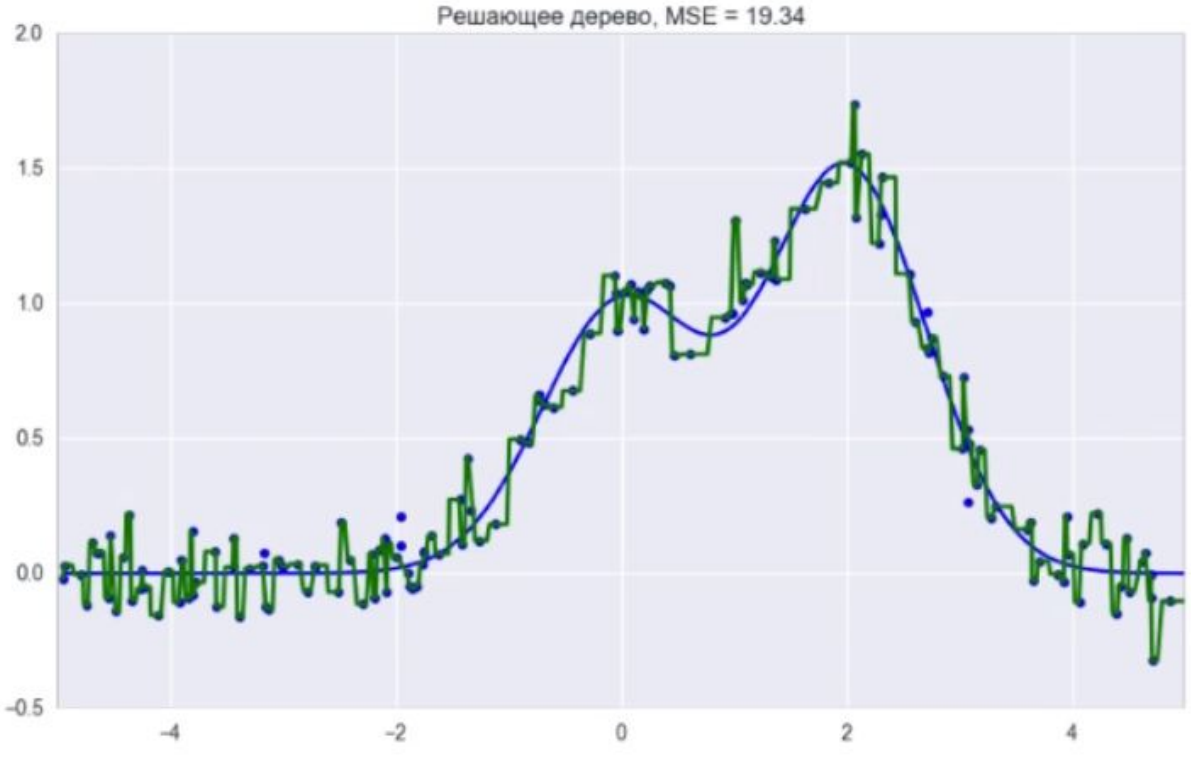
Сгенерировать выборку **Xn** с помощью бутстрэпа;
Построить решающее дерево **bn** по выборке **Xn**:
— по заданному критерию мы выбираем лучший признак, делаем разбиение в дереве по нему и так до исчерпания выборки
— дерево строится, пока в каждом листе не более **nmin** объектов или пока не достигнем определенной высоты дерева
— при каждом разбиении сначала выбирается **m** случайных признаков из **n** исходных,
и оптимальное разделение выборки ищется только среди них.

Случайный лес

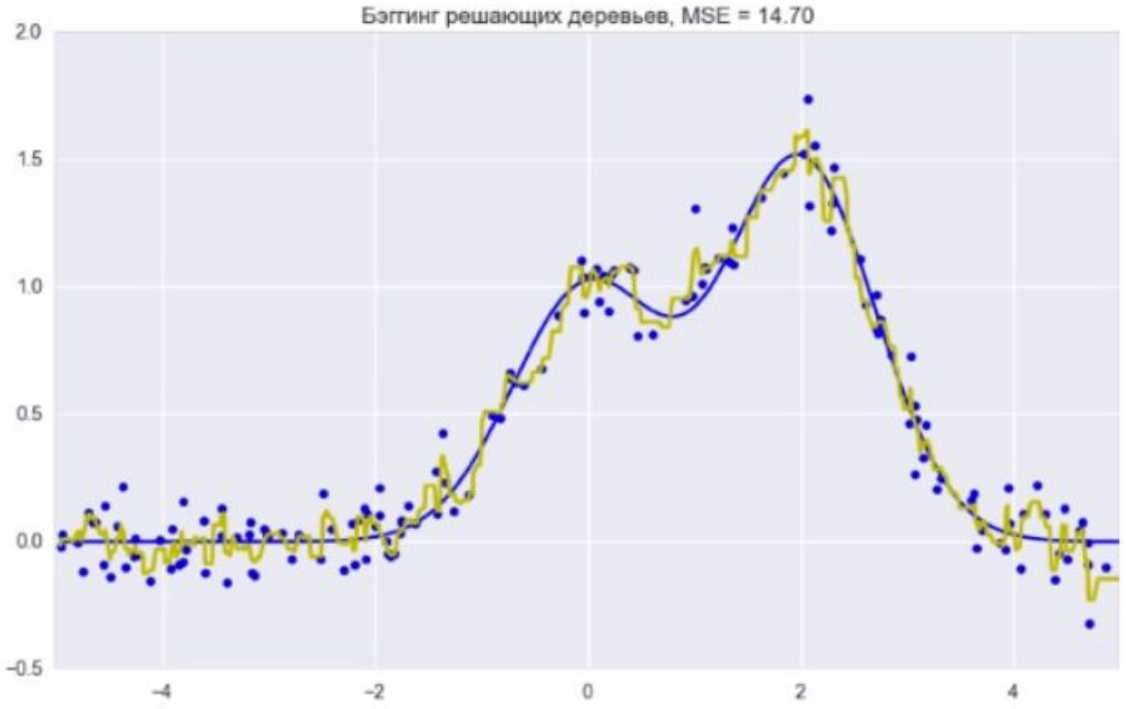
from sklearn.ensemble import DecisionTreeRegressor

criterion – метод оценки ошибки (MSE по умолчанию)
splitter str – метод разделения ”better” или “random”
n_estimators – количество моделей
max_deph – максимальная глубина деревьев
max_samples int or float, optional (default=1.0) Количество сэмплов для обучения
max_features int or float, optional (default=1.0) Количество признаков для обучения

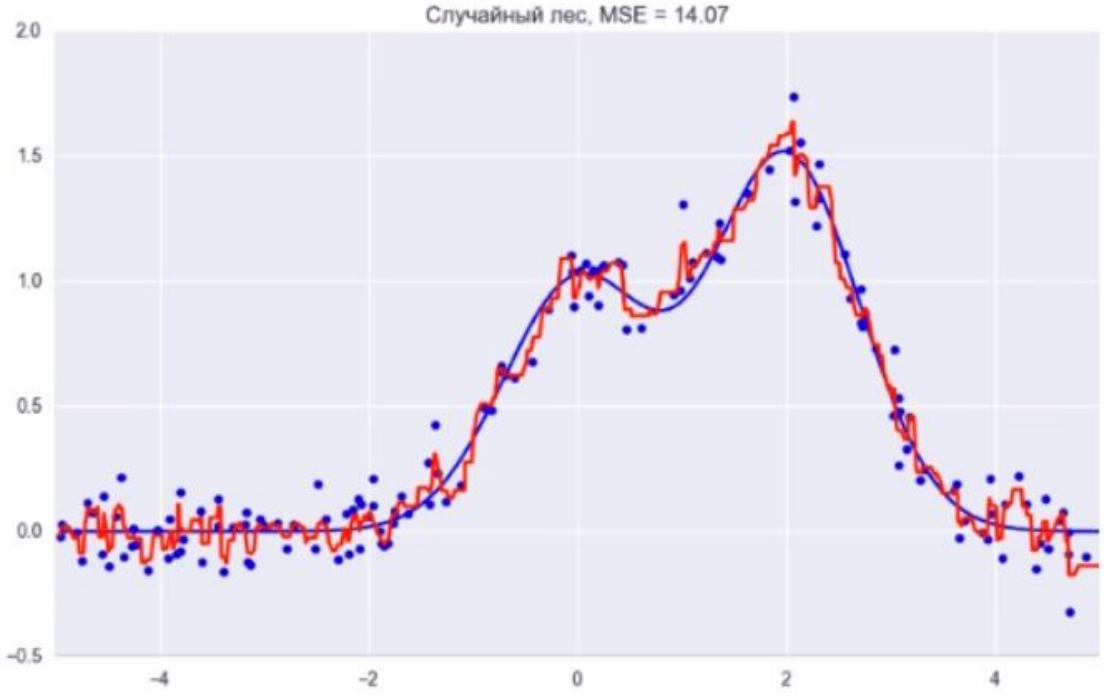
Случайный лес



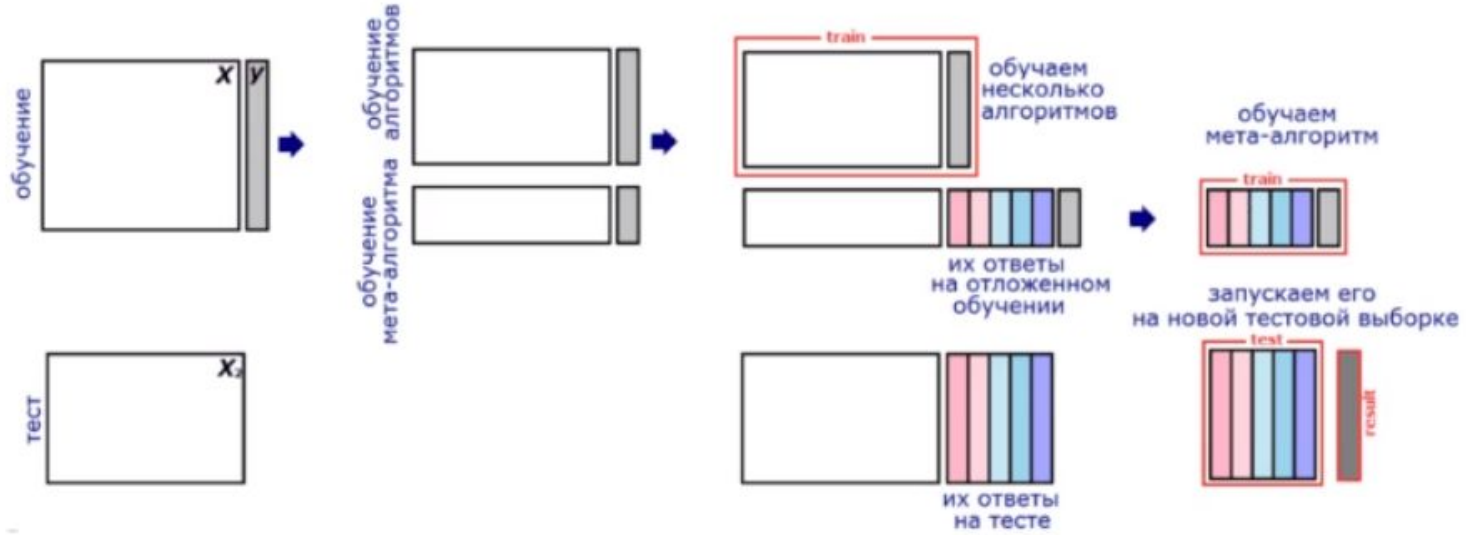
Случайный лес



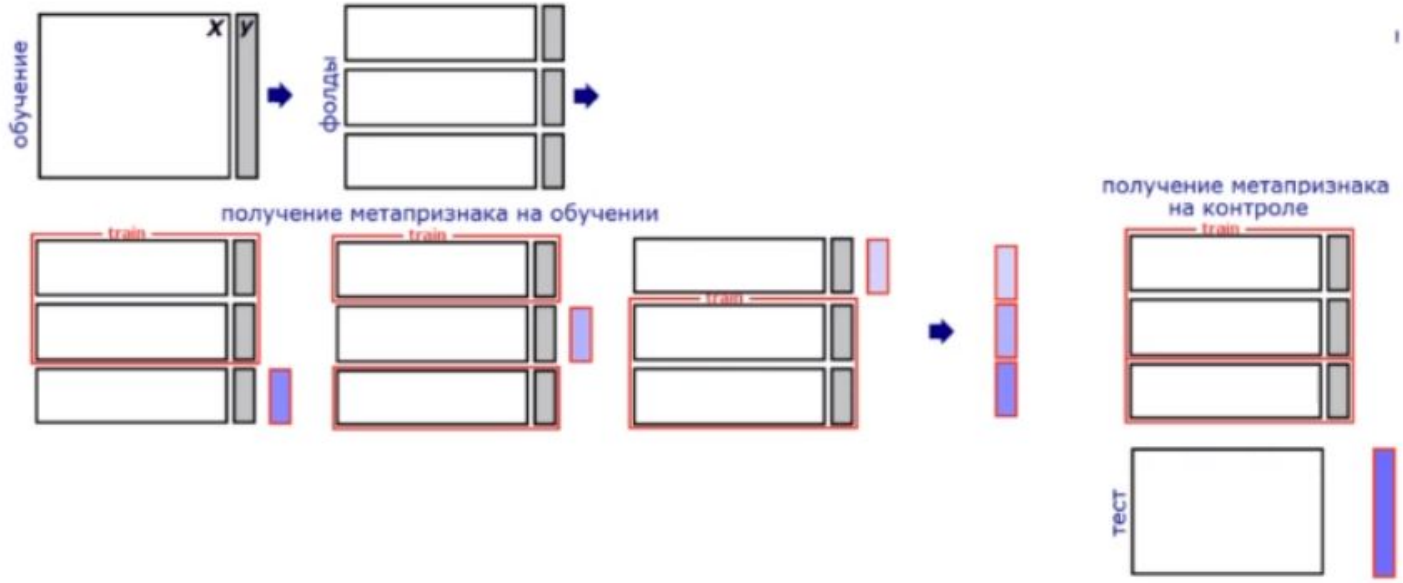
Случайный лес



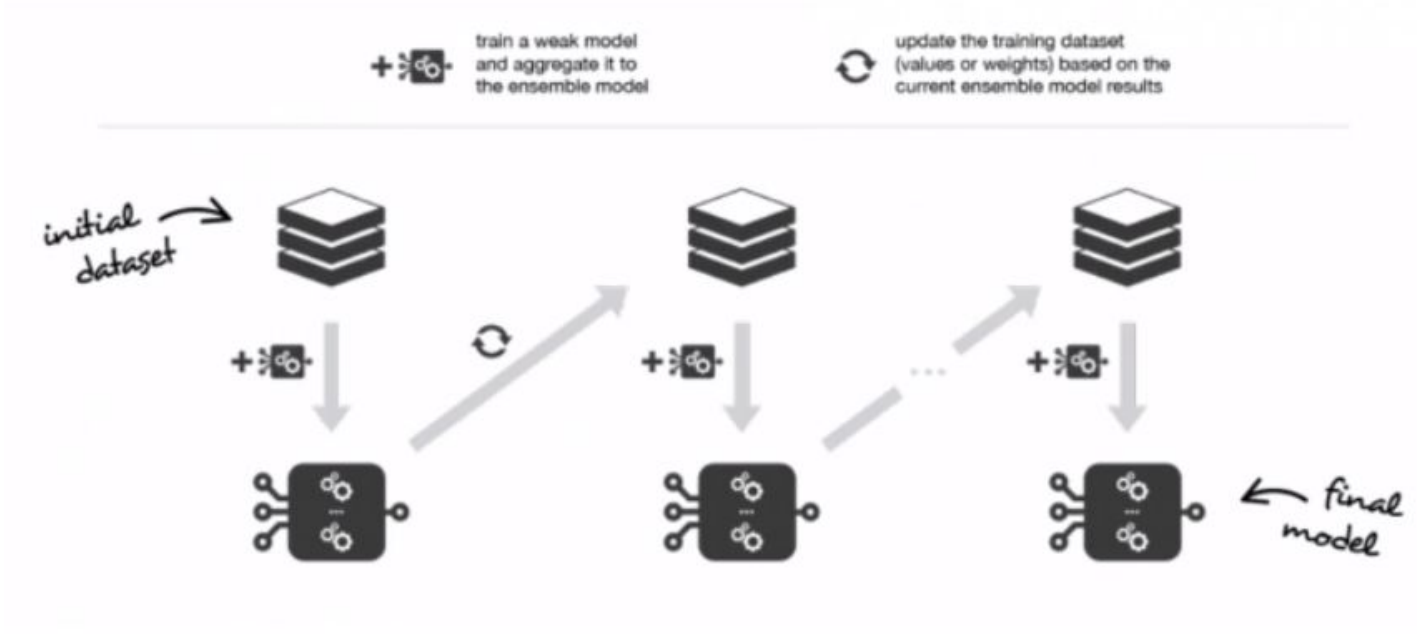
Блендинг


обучение
X
y
тест
X₂
обучение алгоритмов
обучение мета-алгоритма
train
обучаем несколько алгоритмов
их ответы на отложенном обучении
их ответы на тесте
обучаем мета-алгоритм
train
запускаем его на новой тестовой выборке
test
result

Стекинг


обучение
X
y
фолды
получение метапризнака на обучении
train
train
train
получение метапризнака
на контроле
train
тест

Бустинг


train a weak model
and aggregate it to
the ensemble model
update the training dataset
(values or weights) based on the
current ensemble model results
initial
dataset
final
model