Искусственный Интеллект

Лекция 7: Деревья решений. Ансамбли моделей

Мартынюк Полина Антоновна

telegram: @PAMartynyuk

email: pa-martynyuk@yandex.ru

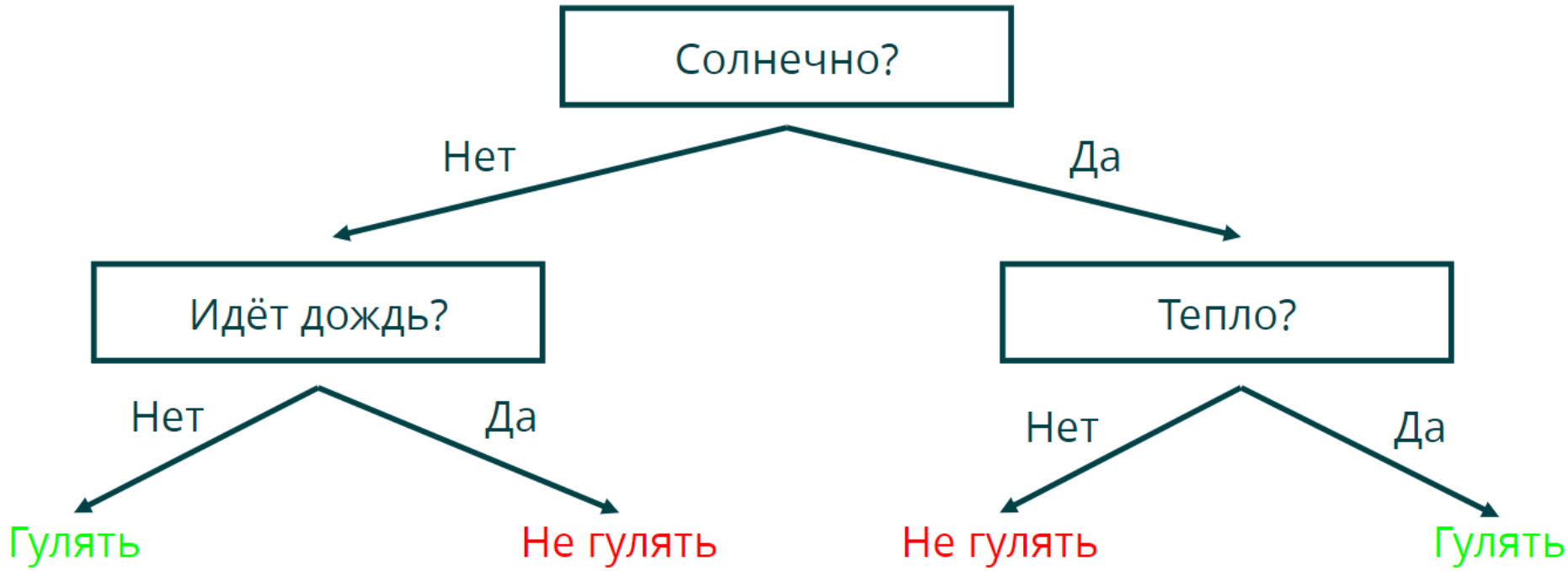
Проблема

- Умеем обучать модель для решения задачи классификации
- Умеем обучать модель для решения задачи регрессии

Проблемы внедрения в бизнес:

- А как работает модель?
- Модель предсказуема?
- Где гарантия того, что модель не наделает ошибок?

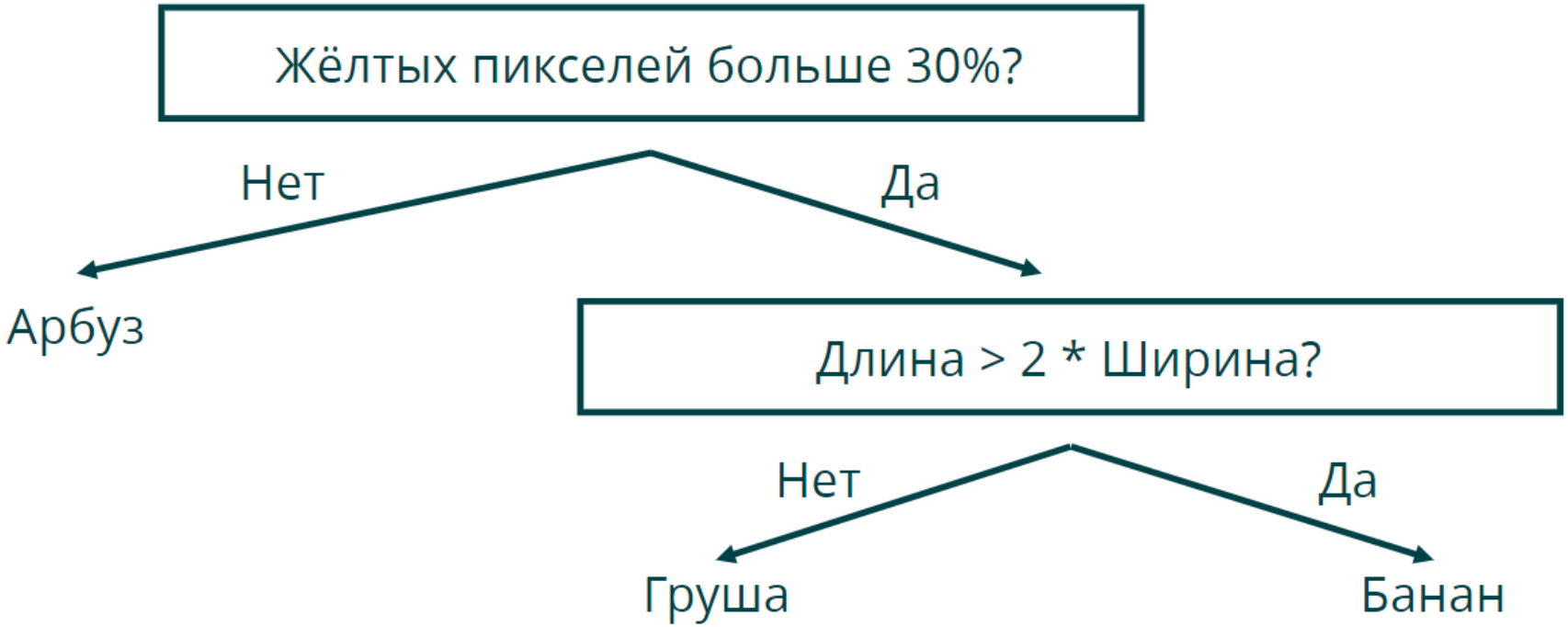
Деревья решений



В чем измерить «солнечность»? Что значит «тепло»?

Деревья решений

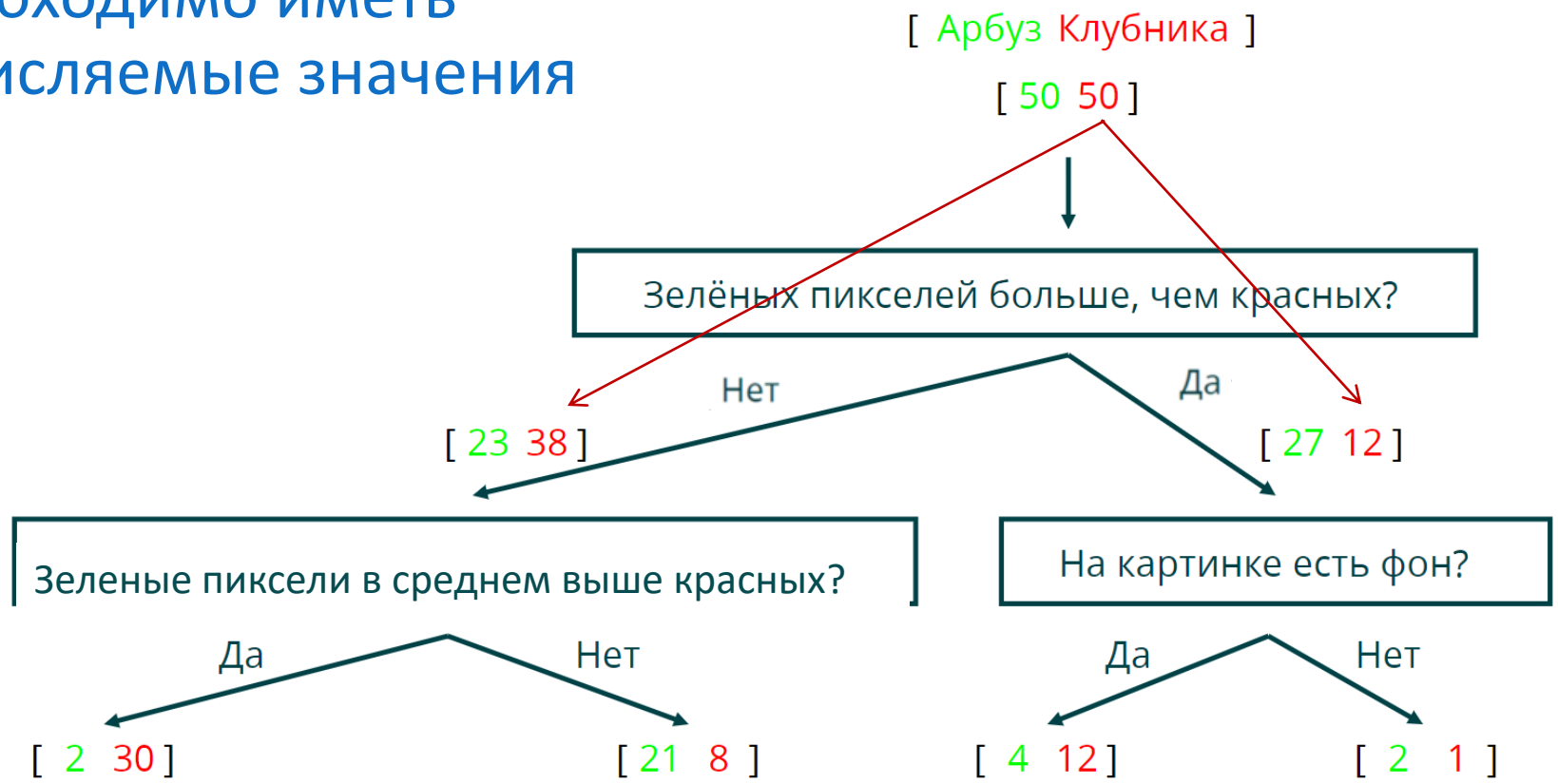
- Необходимо иметь исчисляемые значения



Условия представимы в виде математических формул → математическая модель
Глубина дерева = 2
Как еще можно оценить дерево?

Деревья решений

- Необходимо иметь исчисляемые значения



Проблема – в условиях должны быть задействованы фичи
(условия по фичам – унифицированные условия)

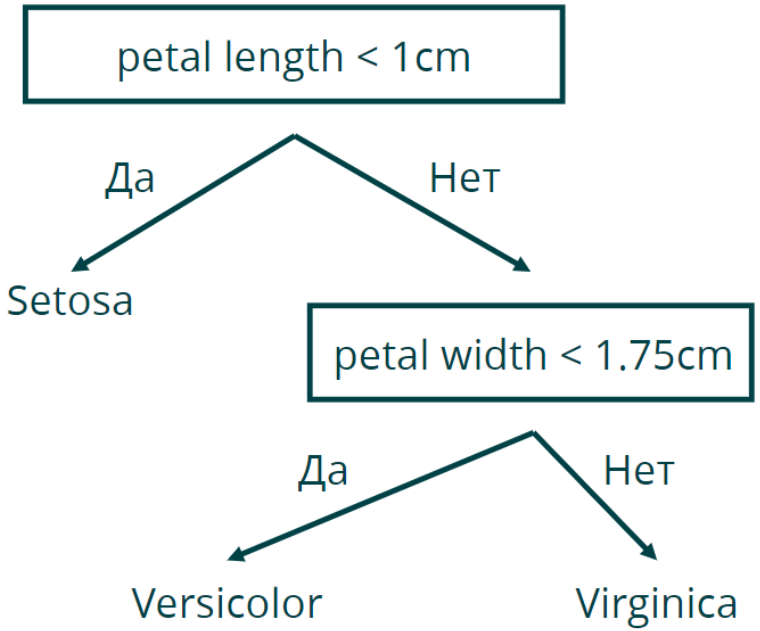
Деревья решений

Формат условий

- 1 условие – 1 фича (1 признак)

$$[x_i < c]$$

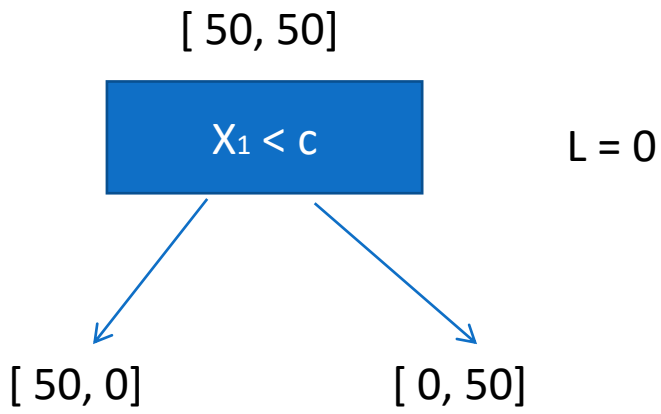
petal length	petal width	sepal length	sepal width	target
				virginica
				versicolor
				setosa
				virginica
				virginica
				setosa
				versicolor



Проблема – как подобрать фичи и условия?

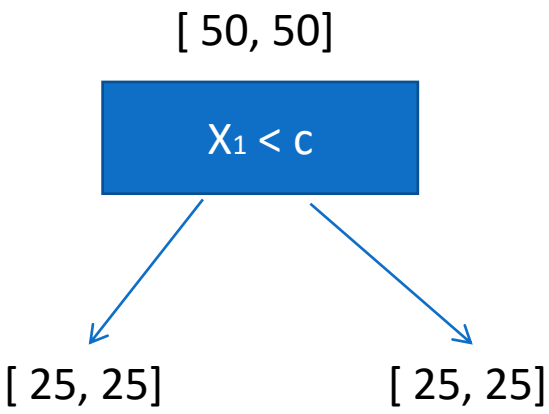
Деревья решений

Идеальный вариант



L = 0

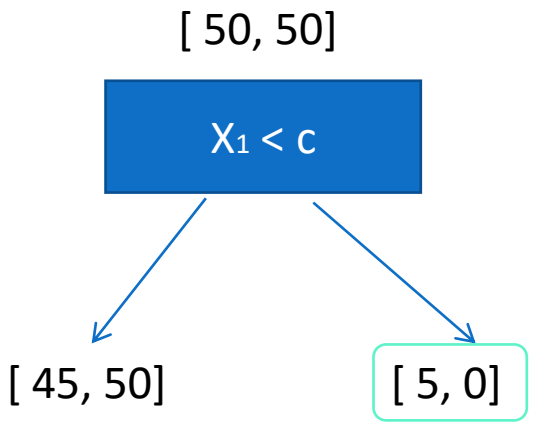
Наихудший вариант



L = max

Деревья решений

Очень специфичное условие – разделение хорошее, но мала доля отделенных элементов класса → надо следить, какую долю наблюдений отсекает условие



Деревья решений

Функция ошибки для дерева решений

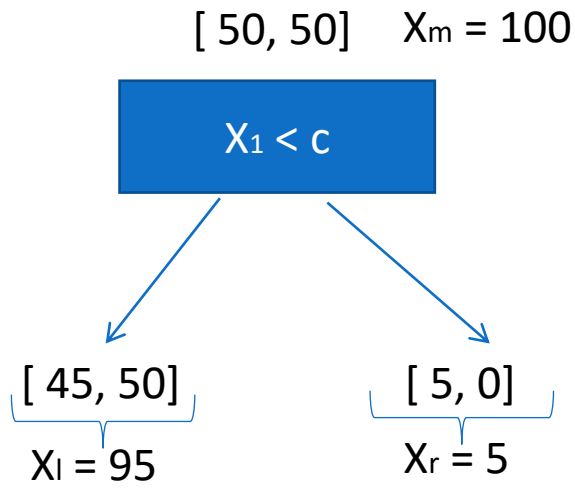
Ошибка слева
(критерий информативности)

$$L(X_m, i, c) = I_{left} + I_{right}$$

$$L(X_m, i, c) = \frac{X_l}{X_m} I_{left} + \frac{X_r}{X_m} I_{right}$$

0.95 0.05

$$L(X_m, i, c) = \frac{X_l}{X_m} I(X_l) + \frac{X_r}{X_m} I(X_r)$$

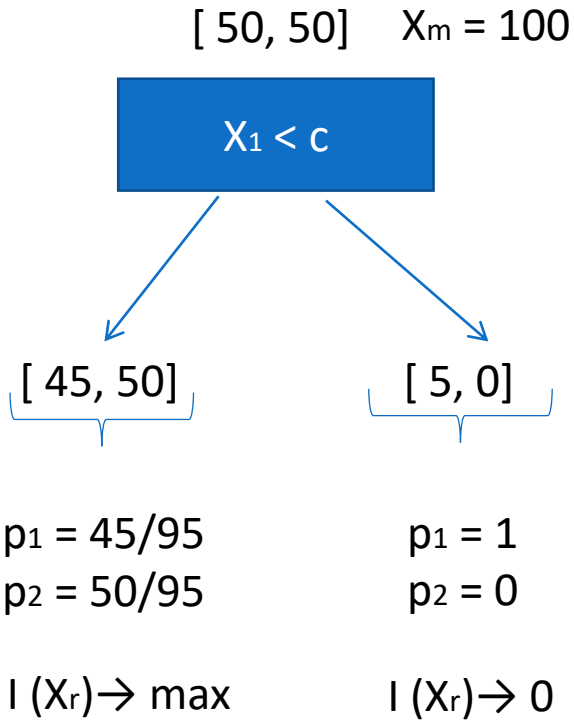


Деревья решений

Критерий информативности Джини

$$I(X) = \sum_k p_k(1 - p_k)$$

$$p_k = \frac{1}{X} \sum_i [y_i = k]$$

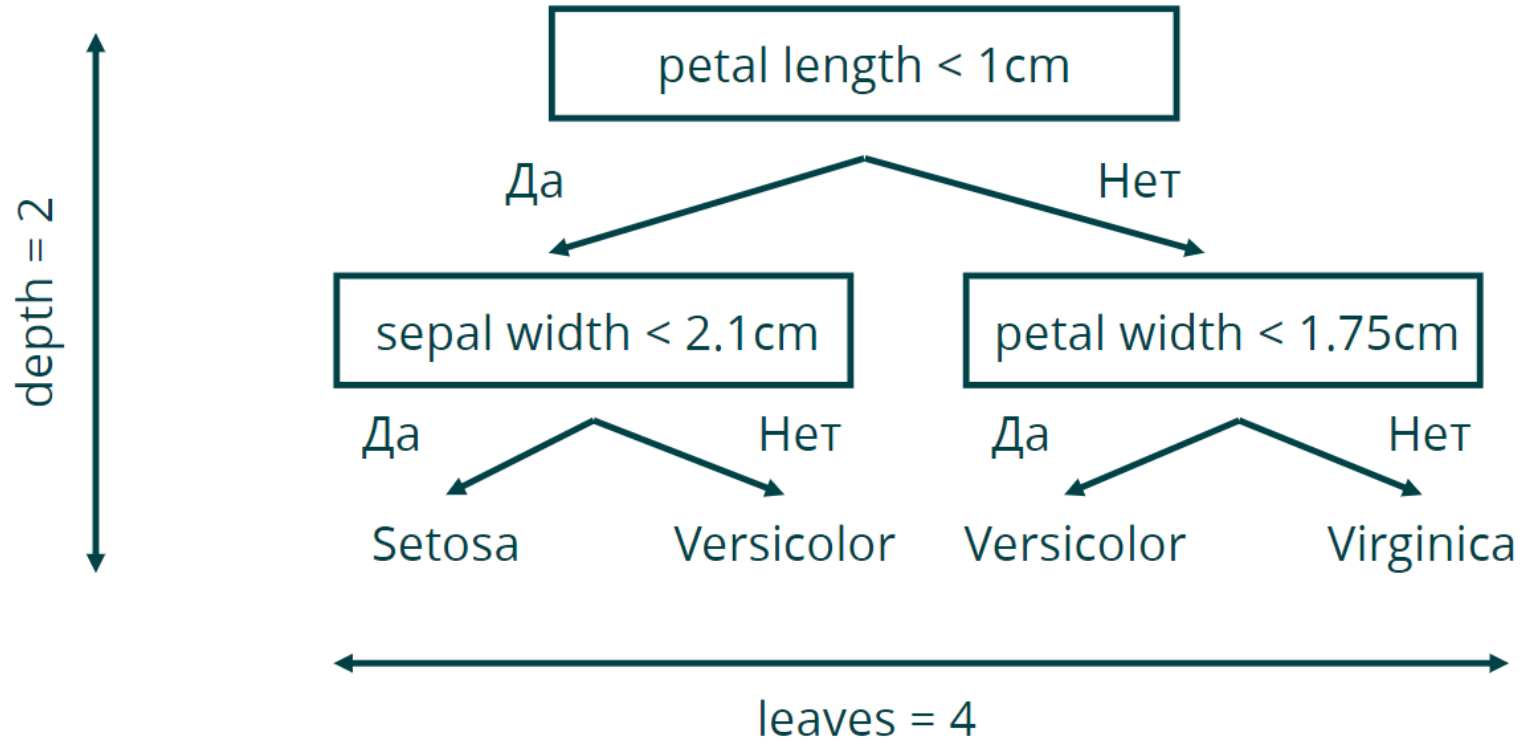


Другой критерий - энтропия

$$I(X) = -\sum_k p_k \ln(p_k)$$

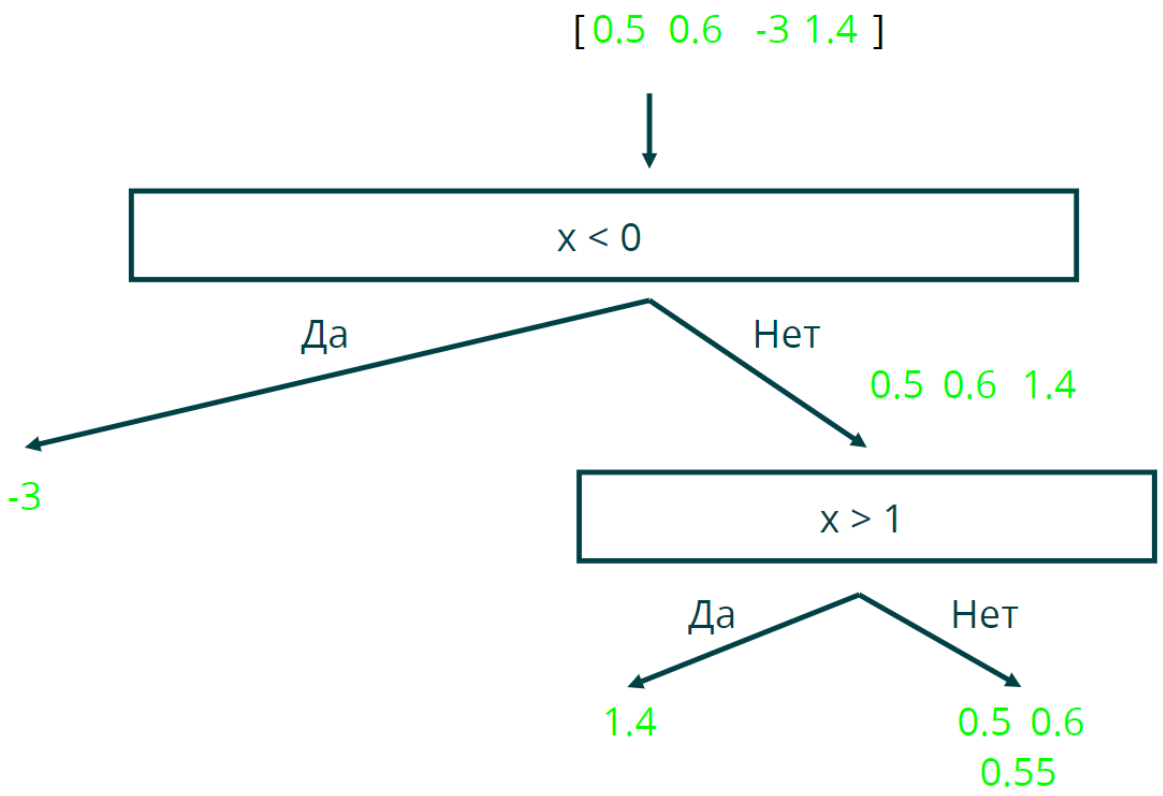
Деревья решений

Параметры дерева решений

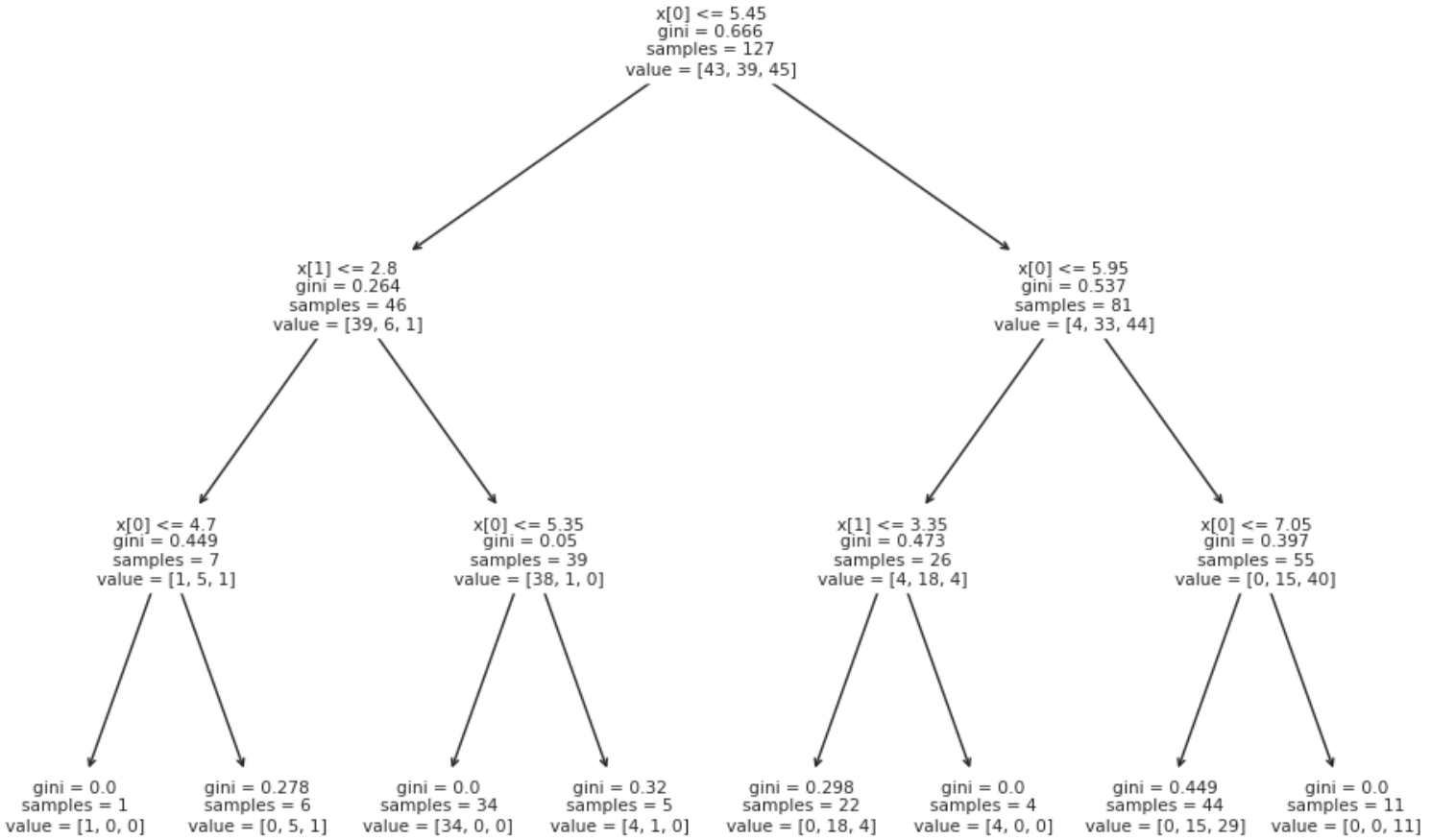


Деревья решений

Дерево решений для регрессии



Деревья решений



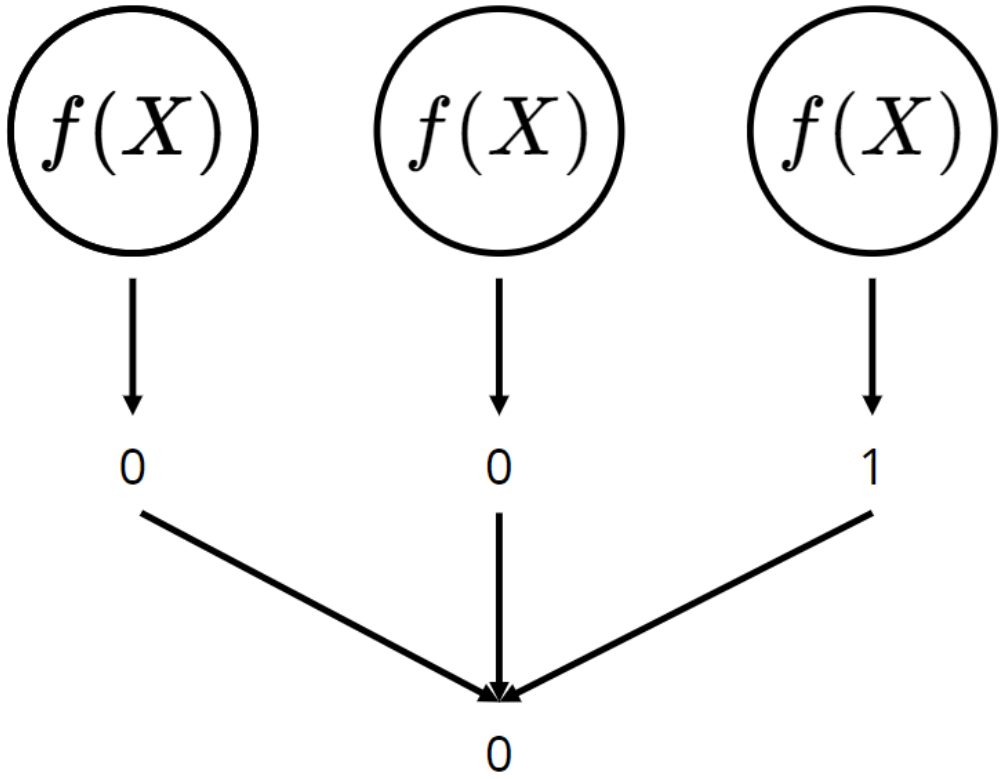
https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

Деревья решений vs. нейронные сети

- Деревья можно (*очень осторожно*) назвать упрощённым вариантом нейронных сетей
- Обе модели способны выявлять **сложные нелинейные зависимости**
- Модели, основанные на **деревьях** - **детерминированные** (дают точный ответ), **нейронные сети** построены на **вероятностях** и дают вероятностный ответ
- Модели, основанные на деревьях, **объясняют** свой результат. Нейронные сети зачастую не дают понятной человеку информации о способе принятия решения

Ансамбли моделей

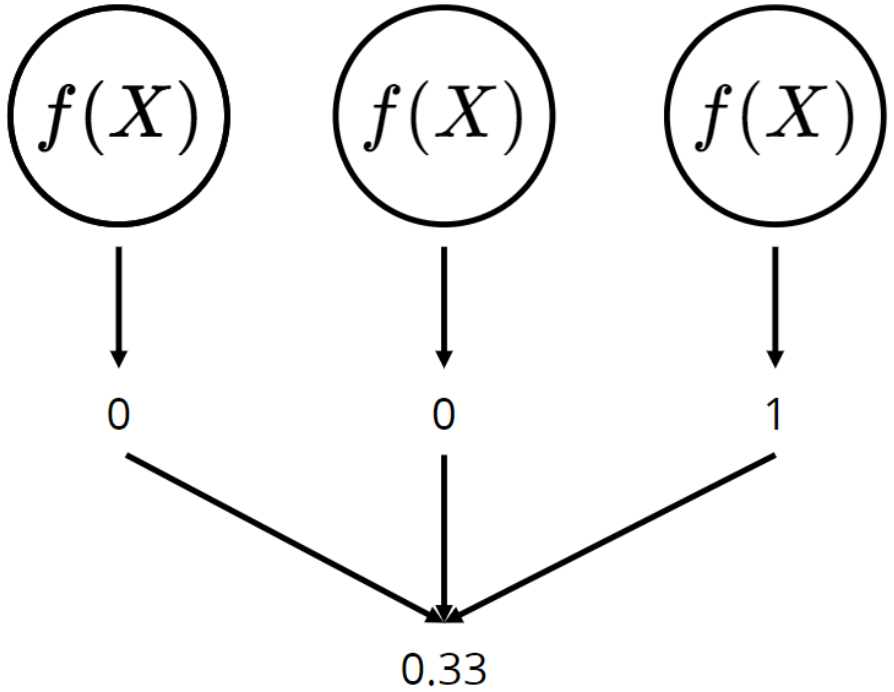
- **Max Voting**



Минус – нет степени уверенности в ответе

Ансамбли моделей

- **Averaging**



$$f_1(X), f_2(X), ..., f_n(X)$$

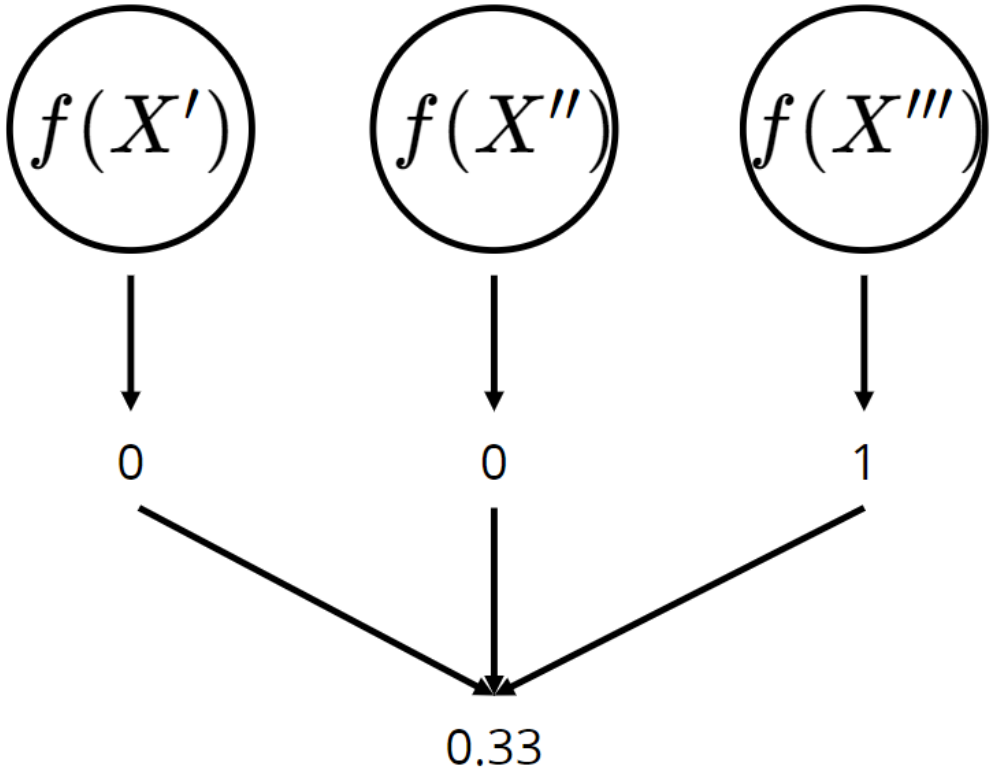
$$f(X) = \frac{\sum_i f_i(X)}{n}$$

Минус – если все модели обучены на одинаковом датасете, они дадут одинаковый результат

Ансамбли моделей

- **Bagging**

Bootstrap AGGregation

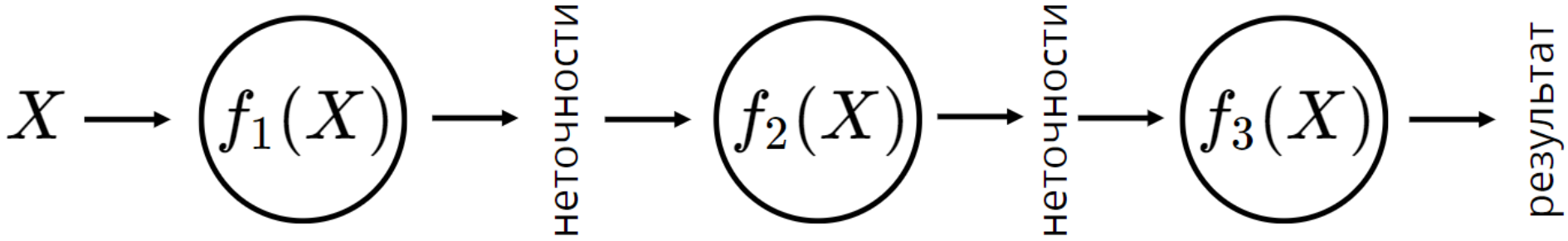


Наиболее известная модель – Случайный лес (Random Forest)

Ансамбли моделей

- **Boosting**

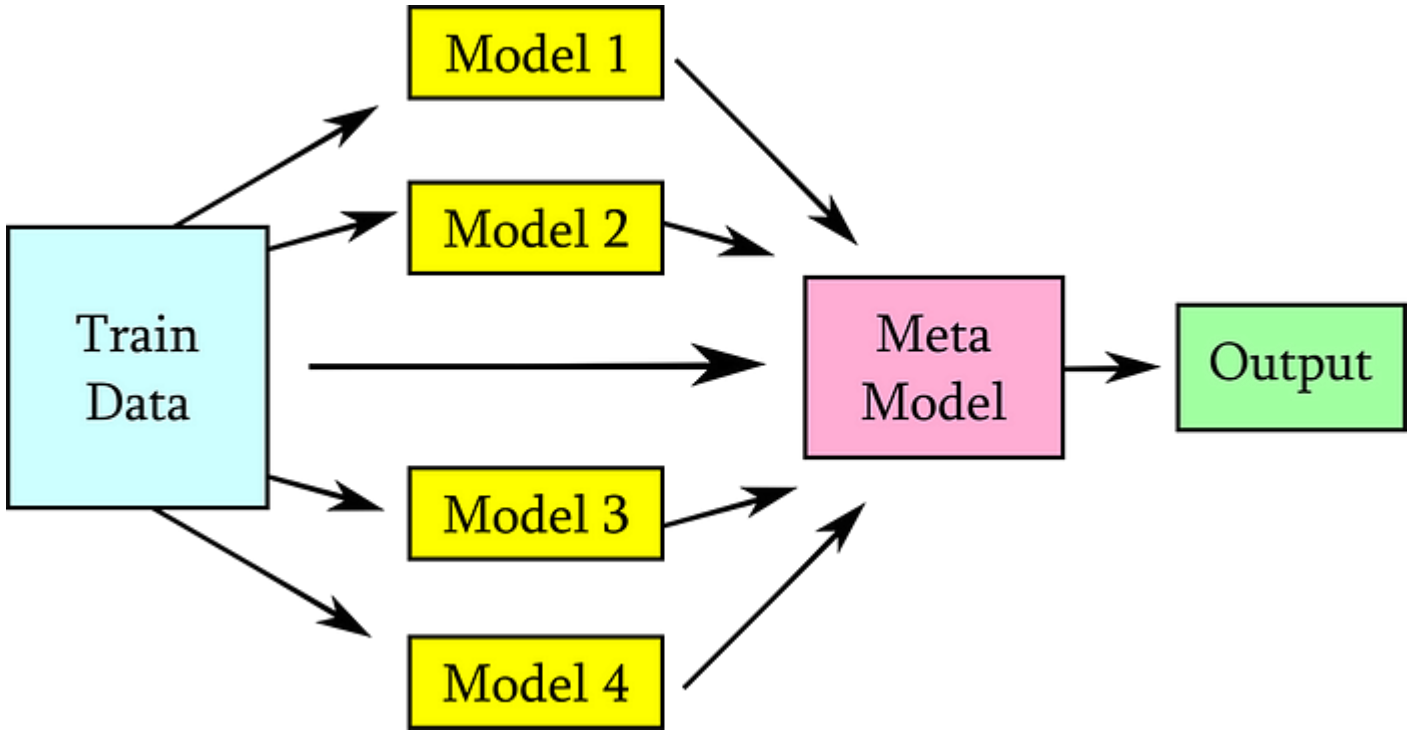
  Последовательный набор моделей, последующие работают с ошибками предыдущей модели



Каждая следующая модель предсказывает то, как надо поправить предыдущую

Ансамбли моделей

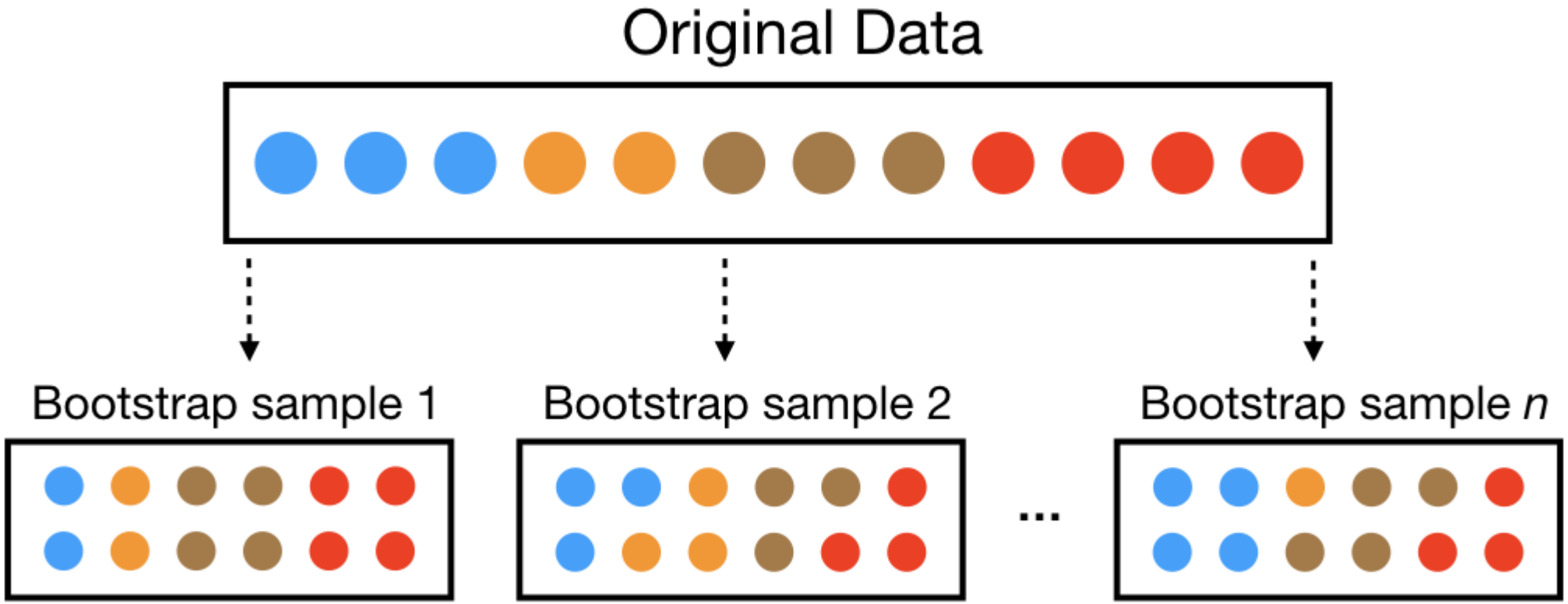
- **Stacking**



Мета-модель обучается на выходных данных предыдущих моделей

Случайный лес (Random Forest)

- **Bootstrap**



Bootstrap

Для каждого дерева выбираем экземпляры данных с возвращением

$$\{x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}\} \rightarrow \{x^{(1)}, x^{(3)}, x^{(4)}\}$$

плохой вариант - требуется дополнительный гиперпараметр

$$\{x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)}, x^{(5)}\} \rightarrow \{x^{(1)}, x^{(3)}, x^{(4)}, x^{(1)}, x^{(4)}\}$$

Bootstrap

height	color	gender	weight
1.6	blue	m	88
1.6	green	f	76
1.5	blue	f	56
1.8	red	m	73
1.5	green	m	77
1.4	blue	f	57

height	color	gender	weight
1.6	blue	m	88
1.8	red	m	73
1.5	blue	f	56
1.8	red	m	73
1.5	green	m	77
1.5	blue	f	56

Случайный лес (Random Forest)

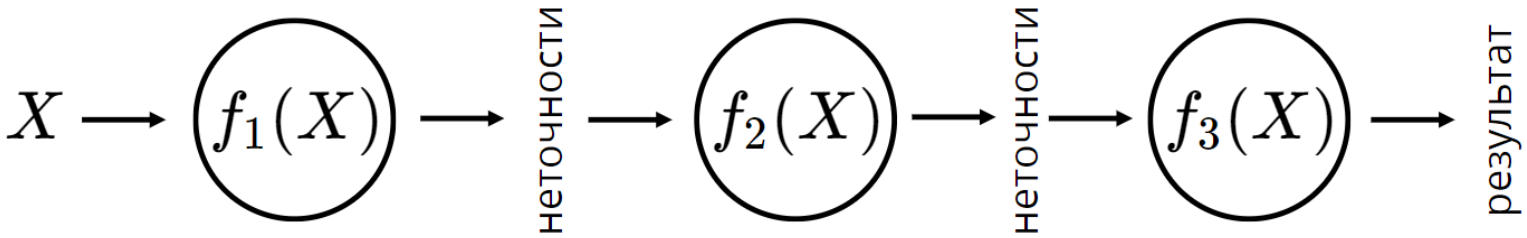
- Всё, что относится к деревьям, можно сказать и про случайный лес
- Случайный лес гораздо менее требователен к вычислительным ресурсам, чем нейронная сеть
- Случайный лес требует гораздо меньше данных для выхода на хорошие значения качества

Преимущества:

- Можно распараллелить
- Показывает важность каждой фичи

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Boosting

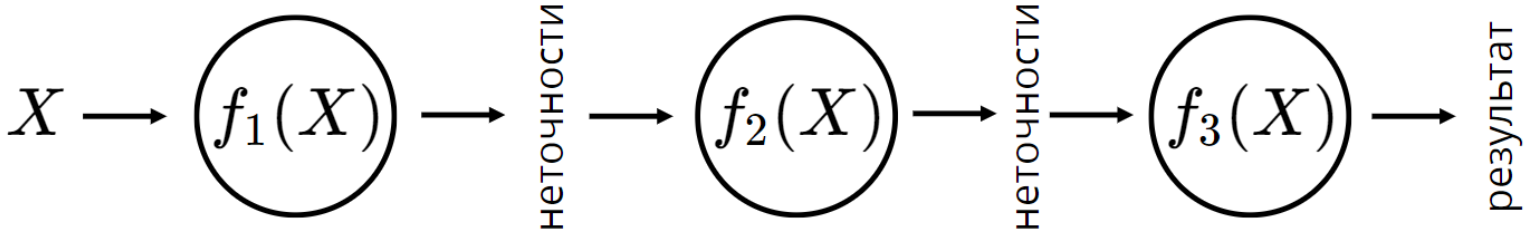


target

height	color	gender	weight
1.6	blue	m	88
1.6	green	f	76
1.5	blue	f	56
1.8	red	m	73
1.5	green	m	77
1.4	blue	f	57

Средний вес: 71.2

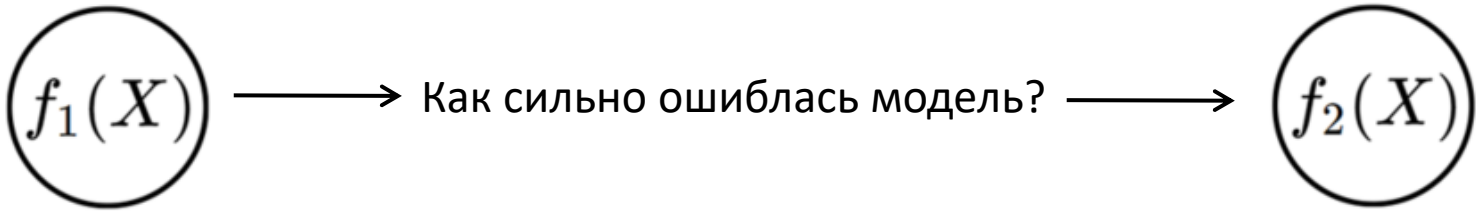
Boosting



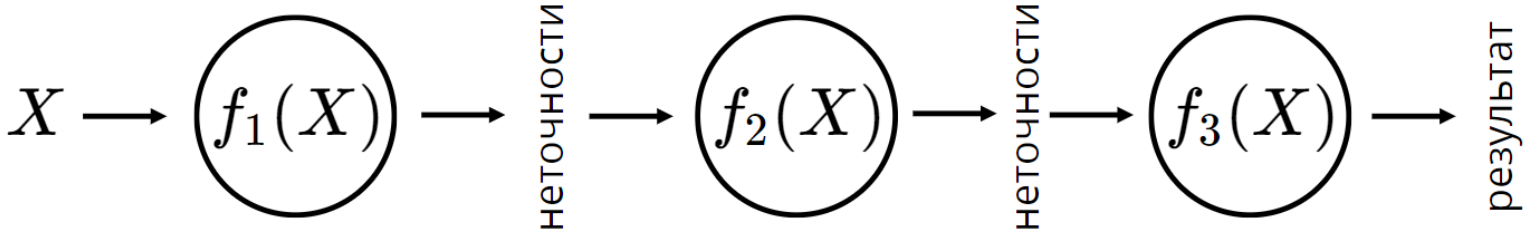
target

height	color	gender	weight	residuals
1.6	blue	m	88	16.8
1.6	green	f	76	4.8
1.5	blue	f	56	-15.2
1.8	red	m	73	1.8
1.5	green	m	77	5.8
1.4	blue	f	57	-14.2

Средний вес: 71.2



Boosting



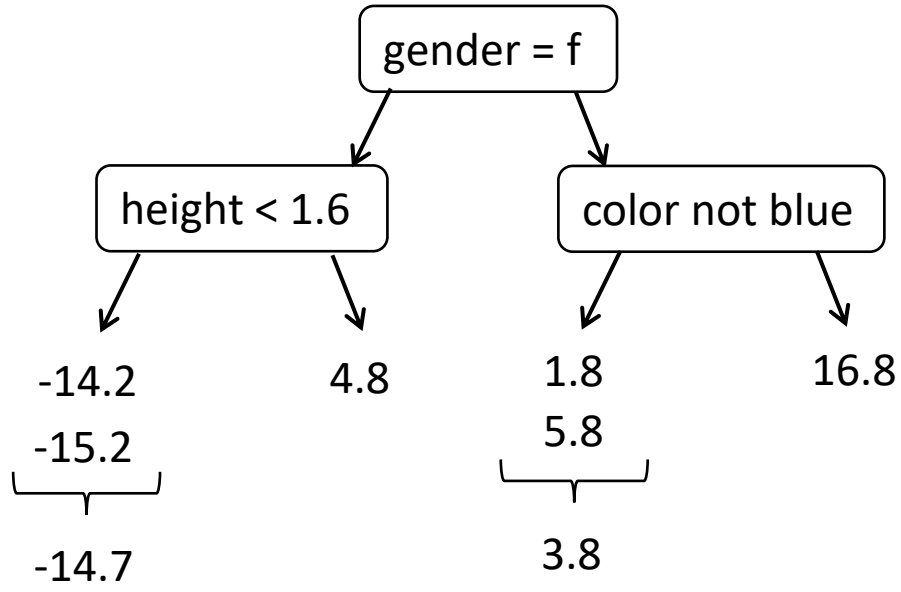
target

height	color	gender	weight	residuals
1.6	blue	m	88	16.8
1.6	green	f	76	4.8
1.5	blue	f	56	-15.2
1.8	red	m	73	1.8
1.5	green	m	77	5.8
1.4	blue	f	57	-14.2

Средний вес: 71.2

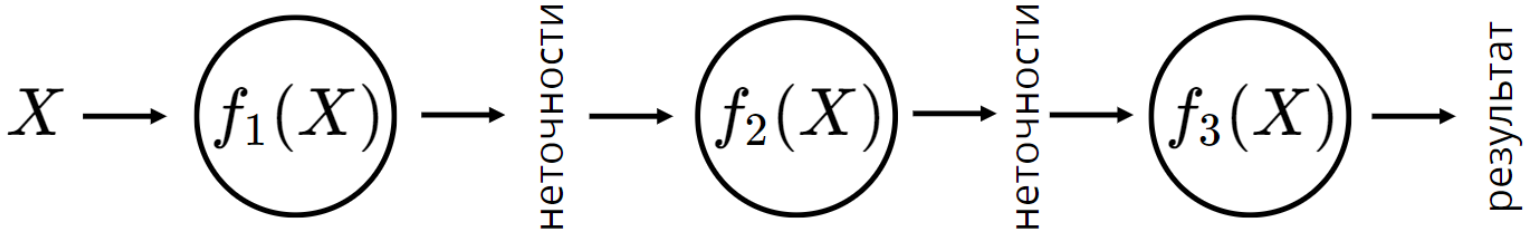
$f_1(X)$

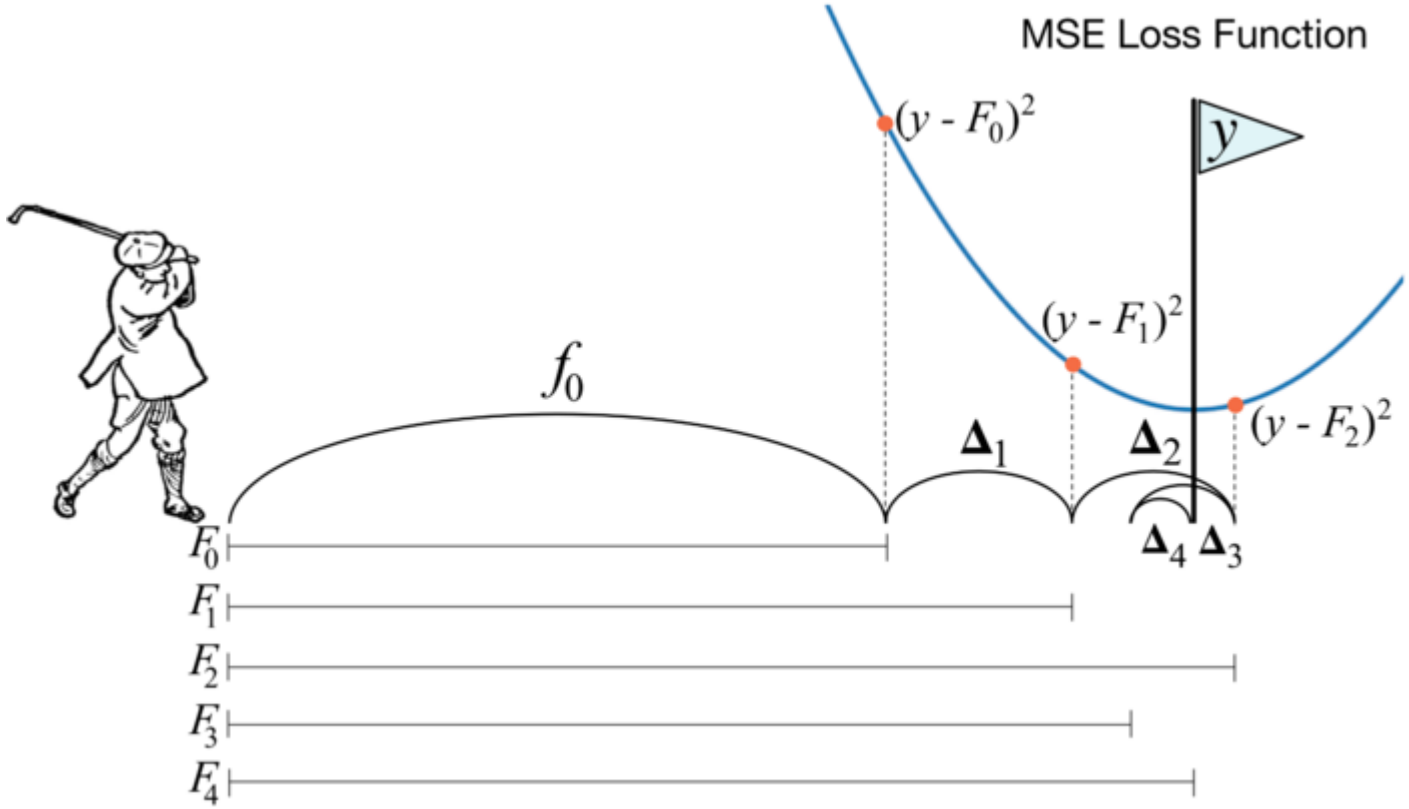
$f_2(X)$



71.2 + 0.1 (f2) + 0.1 (f2) + …

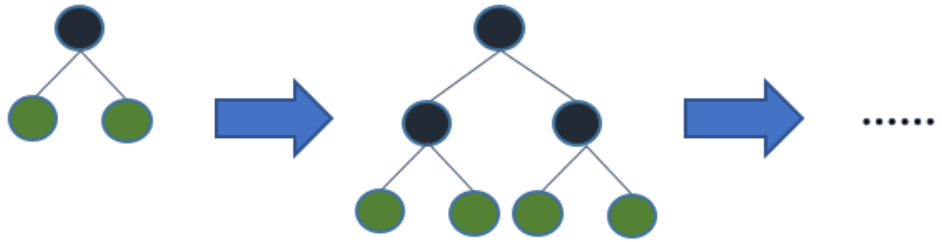
Boosting


X
$f_1(X)$
неточности
$f_2(X)$
неточности
$f_3(X)$
результат


MSE Loss Function
$(y - F_0)^2$
y
$(y - F_1)^2$
$(y - F_2)^2$
f_0
Δ_1
Δ_2
Δ_4 Δ_3
F_0
F_1
F_2
F_3
F_4

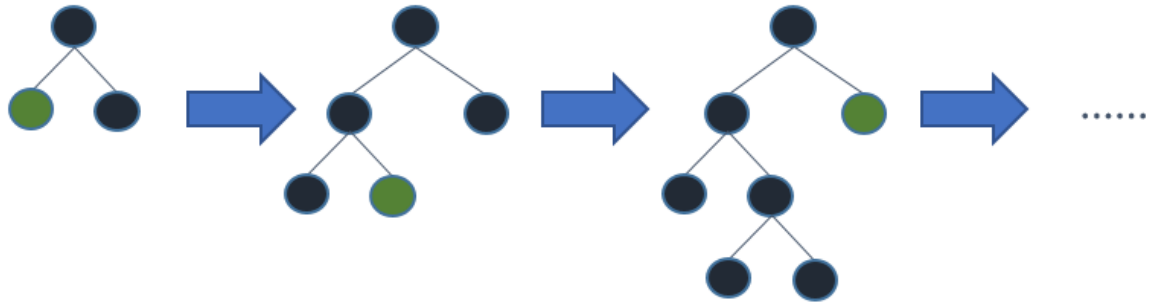
Boosting

XGBoost



Level-wise tree growth

LightGBM



Leaf-wise tree growth

https://neptune.ai/blog/xgboost-vs-lightgbm

Выводы

- Дерево решений обучается слой за слоем с помощью индекса Джини или энтропии и позволяет визуализировать данные
- Ансамбли моделей позволяют с помощью большого количества простых моделей находить сложные нелинейные зависимости
- Случайный лес параллельно объединяет деревья решений с помощью механизма бутстрапинг
- Бустинг объединяет модели в цепочку последовательных исправлений ошибок предыдущей моделей

Спасибо за внимание!

Конец Лекции 7