ML Professional

Gradient boosting

otus.ru

Проверить, идет ли запись

Меня хорошо видно && слышно?

Ставим “+”, если все хорошо “-”,
если есть проблемы

Тема вебинара

ML Professional
Gradient boosting

Андрей Канашов

Team Lead Data Scientist в ПИК

- Ценообразование и тарификация
- Рекомендательные системы
- Прогнозирование ключевых метрик
- Анализ клиентского поведения

Правила вебинара

Активно
участвуем

Off-topic обсуждаем
в учебной группе
#ML-2024-07

Задаем вопрос
в чат

Вопросы вижу в чате,
могу ответить не сразу

Условные обозначения

Индивидуально

Время, необходимое
на активность

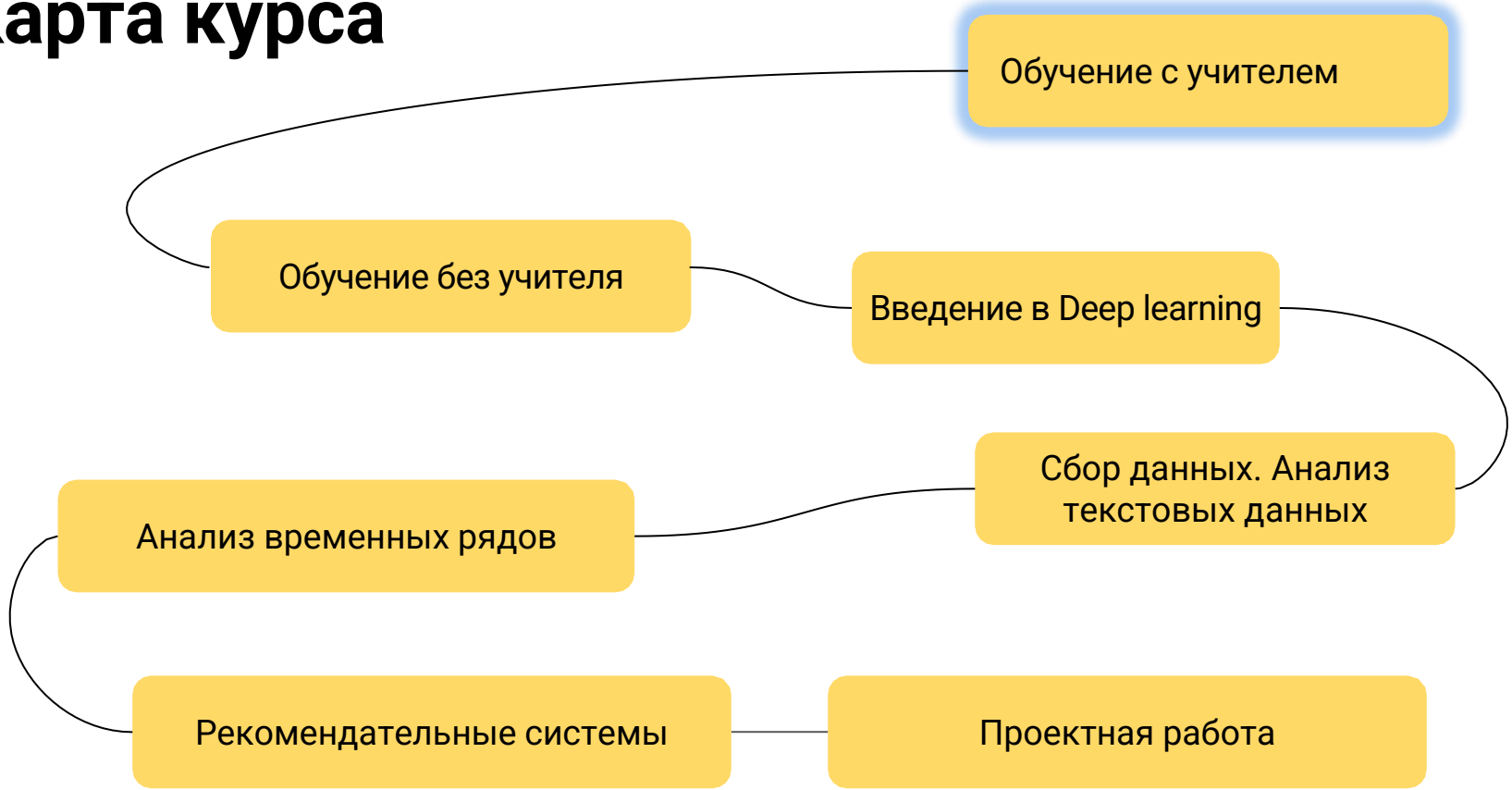
Пишем в чат

Говорим голосом

Документ

Ответьте себе или
задайте вопрос

Карта курса


Обучение с учителем
Обучение без учителя
Введение в Deep learning
Сбор данных. Анализ текстовых данных
Анализ временных рядов
Рекомендательные системы
Проектная работа

Маршрут вебинара

- Идея
- Последовательное улучшение
- Градиентный бустинг над деревьями
- Гиперпараметры
- LightGBM | XGBoost | CatBoost
- Практика применения

Цели занятия

Что вы сможете

1. Рассмотреть метод градиентного бустинга
2. Узнать, как применять современные библиотеки

Смысл

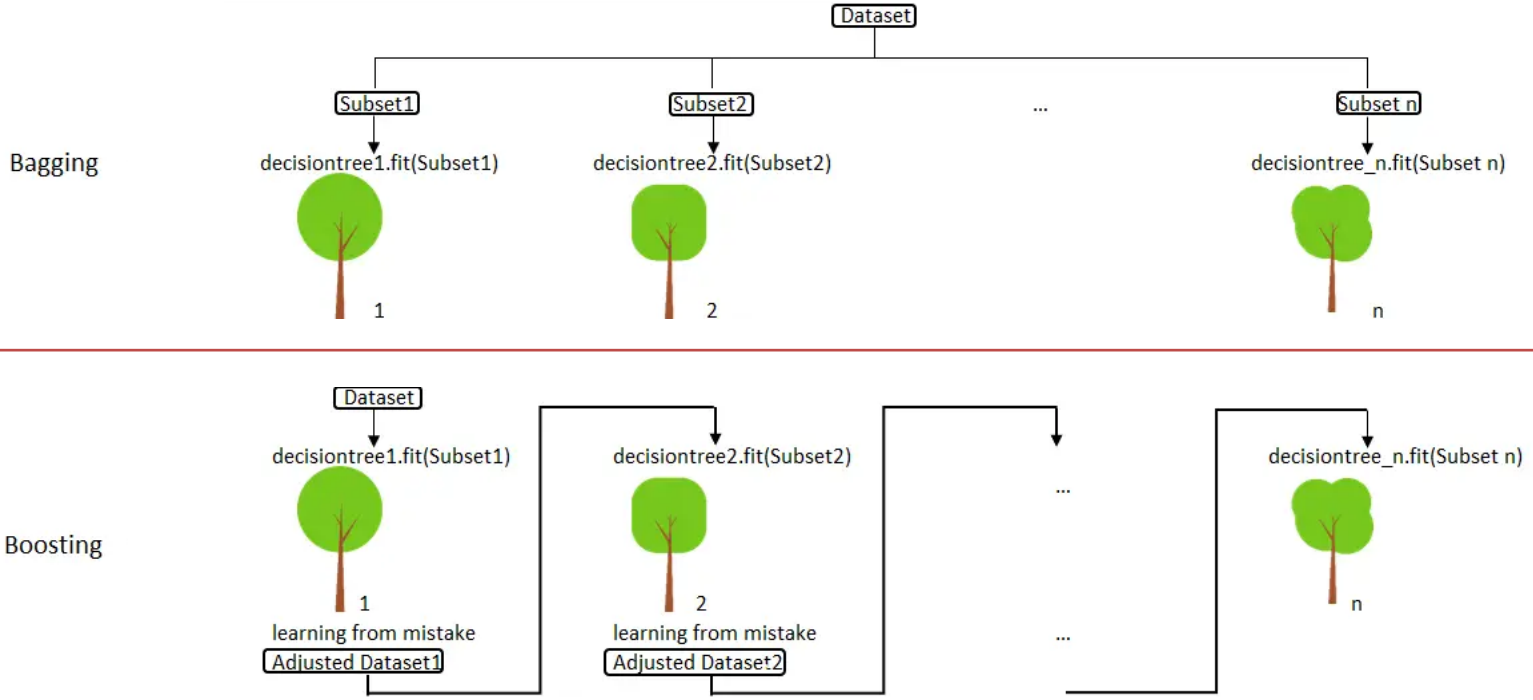
Зачем вам это уметь

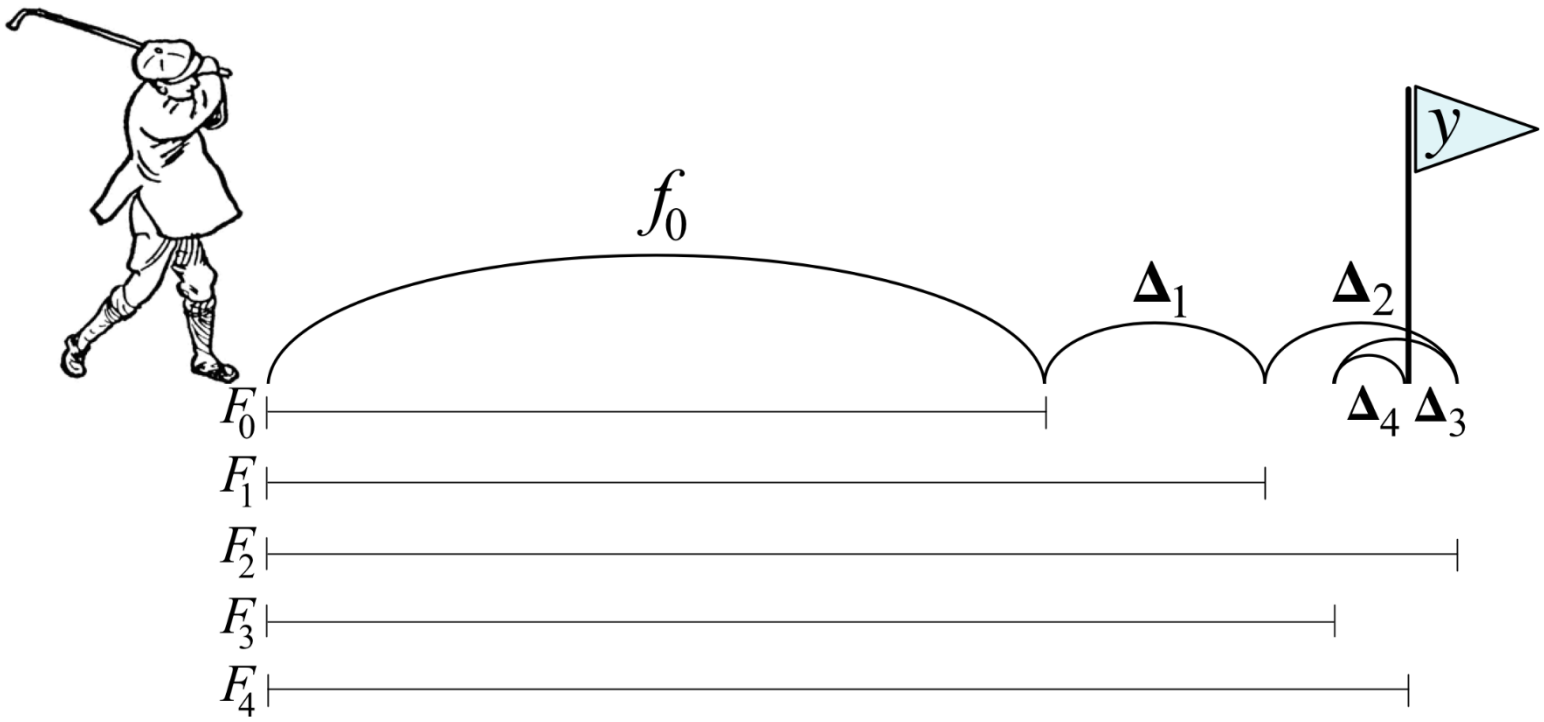
1. Понимать как применять методы градиентного бустинга
2. Знать особенности применения

Gradient boosting

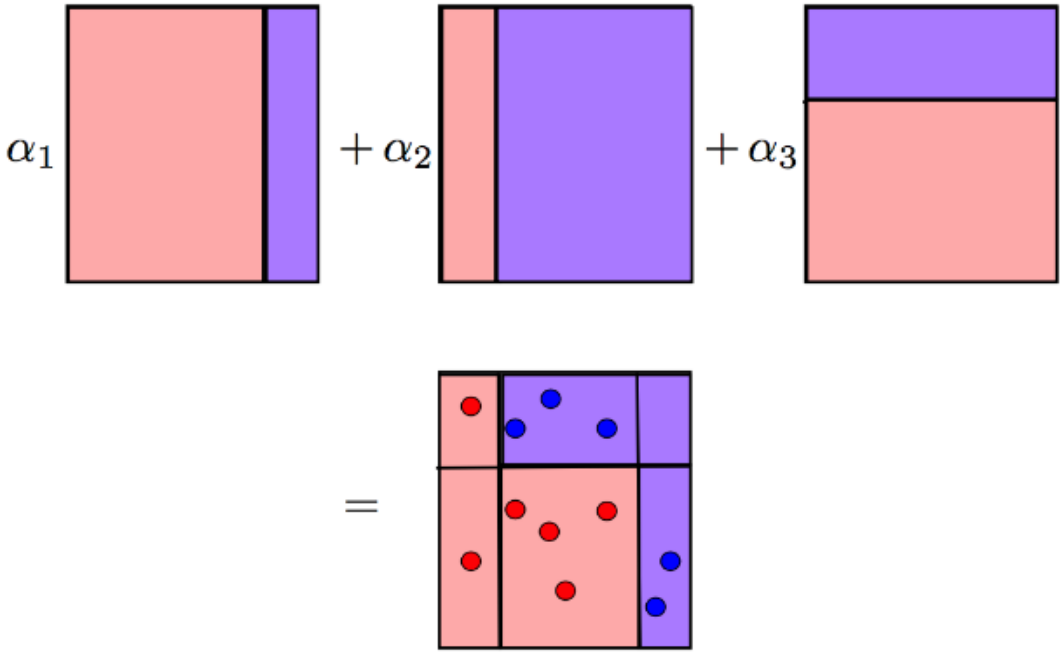
идея

Bagging vs Boosting




f_0
Δ_1
Δ_2
y
Δ_4
Δ_3
F_0
F_1
F_2
F_3
F_4

AdaBoost – Последовательное улучшение



Алгоритм - AdaBoost

- $a(x) = b_1(x) + b_2(x) + \cdots + b_k(x)$ – последовательное приближение решения, $\mathcal{L}(y, x)$- функция потерь
- Построим **первое** приближение: $b_1(x) = \underset{b \in \mathcal{B}}{\operatorname{argmin}} \mathcal{L}\big(y, b(x)\big)$ **на исходных** данных и вычислим ошибки: $s_i^1 = y_i - \lambda b_1(x);$
- Построим **второе** приближение **на ошибках первого**: $b_2(x) = \underset{b \in \mathcal{B}}{\operatorname{argmin}} \mathcal{L}\big(s^1, b(x)\big)$ и вычислим его ошибки: $s_i^2 = s_i^1 - \lambda b_2(x);$
- …
- Приближение n на ошибках $n-1$: $b_n(x) = \underset{b \in \mathcal{B}}{\operatorname{argmin}} \mathcal{L}\big(s^{n-1}, b(x)\big)$, $s \leftarrow s^{n-1} - \lambda b_n;$

Особенности бустинга

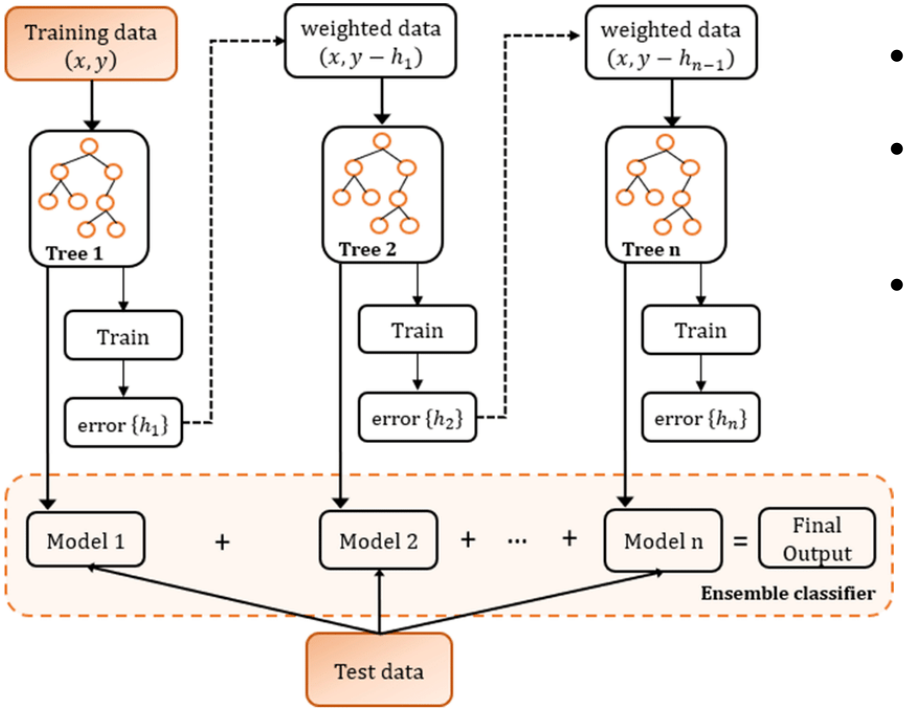
Бустинг - итерационный алгоритм, реализующий «сильный» классификатор, который позволяет добиться произвольно малой ошибки обучения на основе композиции «слабых» классификаторов, каждый из которых немного лучше, чем просто угадывание.

Базовые классификаторы должны быть слабыми, из сильных хорошую композицию не построить («бритва Оккама»)

- сильный классификатор, давая нулевую ошибку на обучающих данных, не адаптируется и композиция будет состоять из одного классификатора
- один, даже сильный, классификатор может дать «плохое» предсказание на данных тестирования, давая «хорошие» результаты на обучающих данных.

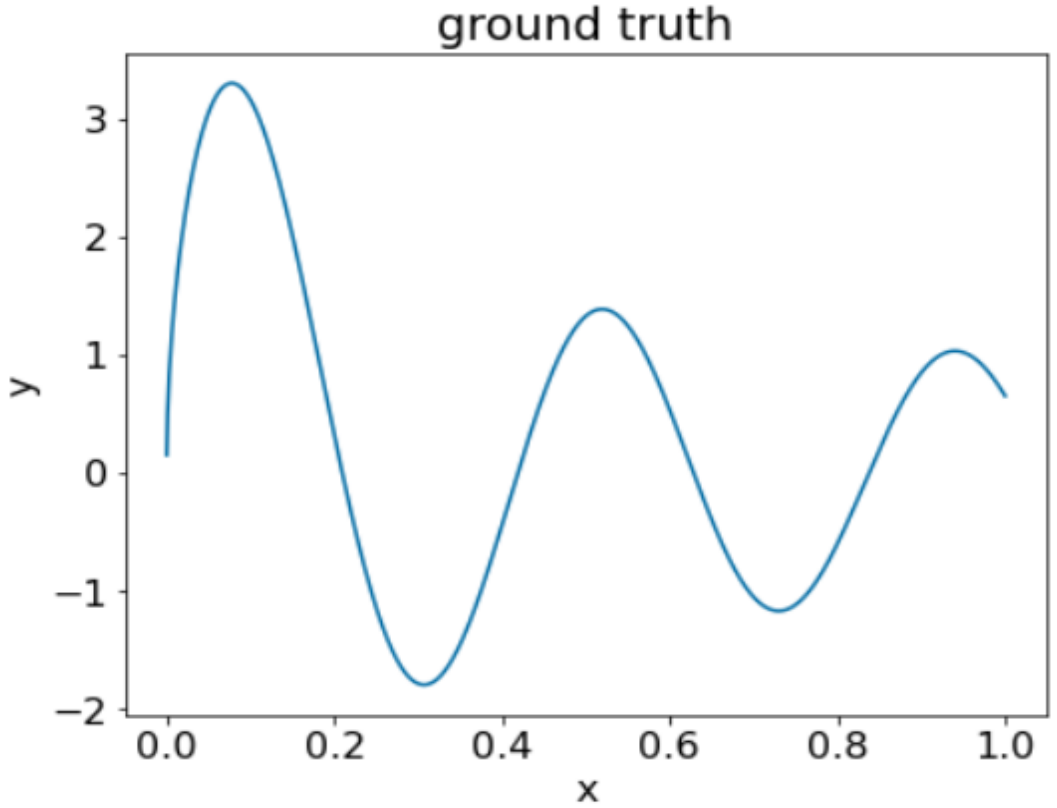
Gradient boosting

Градиентный бустинг над решающими деревьями



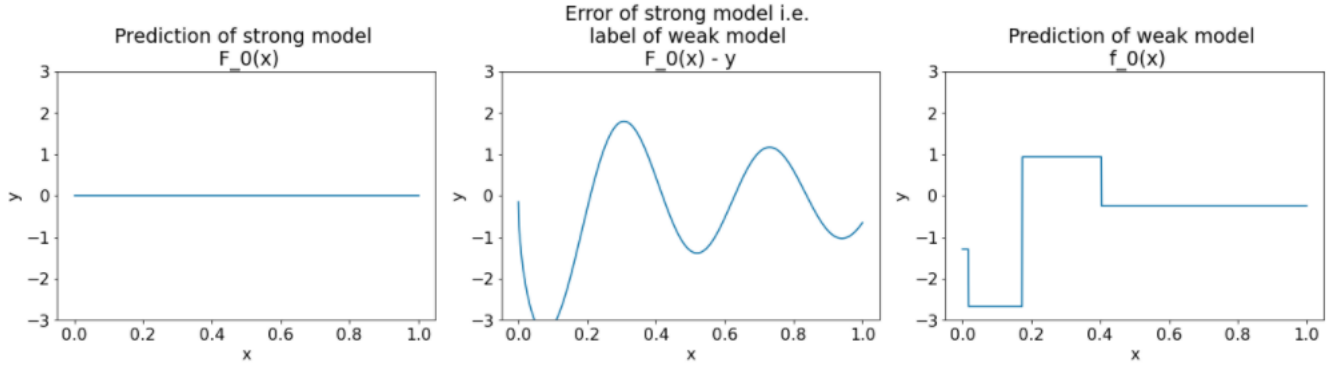
- Строим алгоритмы последовательно
- Каждый следующий строится на ошибках предыдущего
- Решение принимается методом взвешенного голосования

Пример работы градиентного бустинга

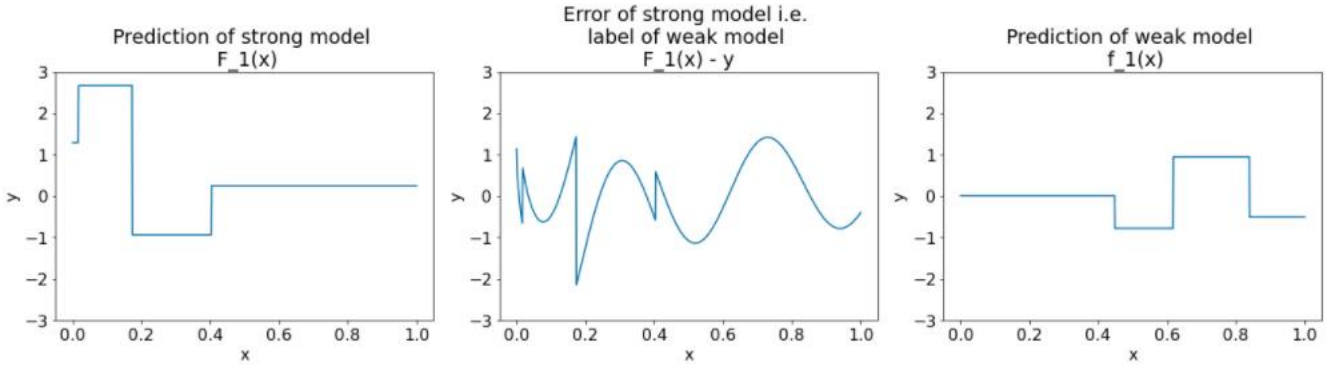


Пример работы градиентного бустинга

Первый шаг

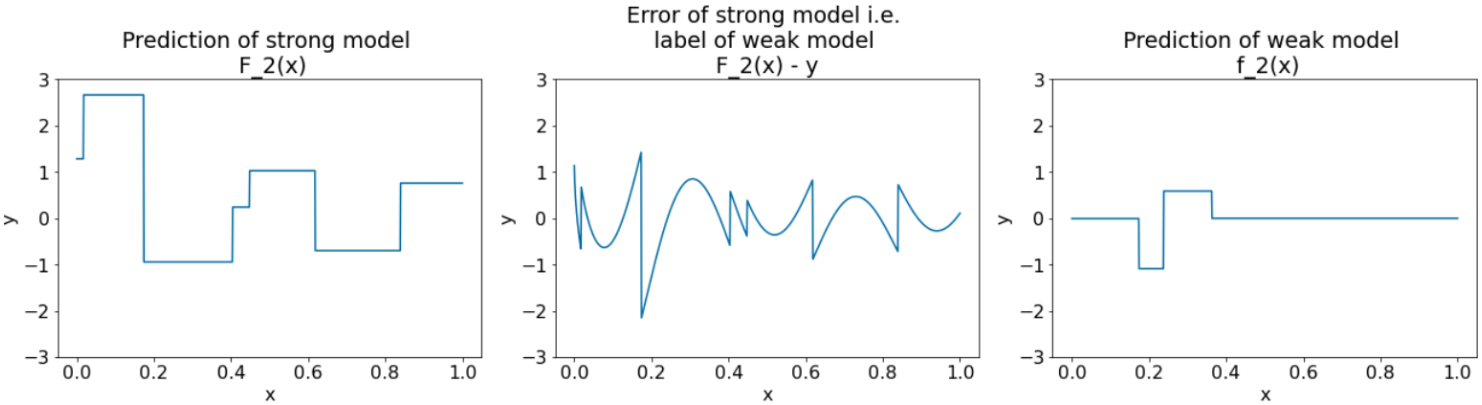


Второй шаг



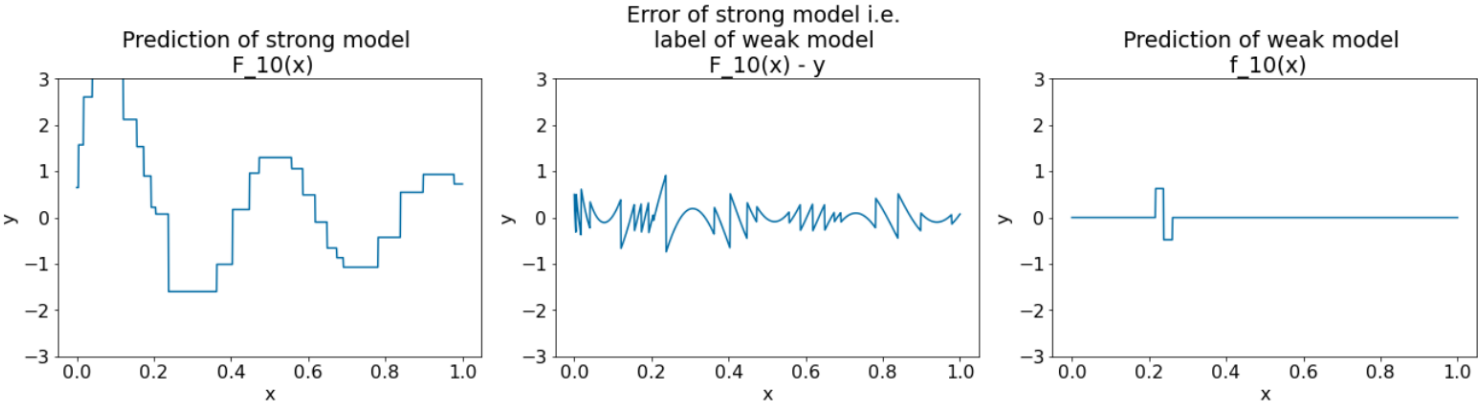
Пример работы градиентного бустинга

Третий шаг



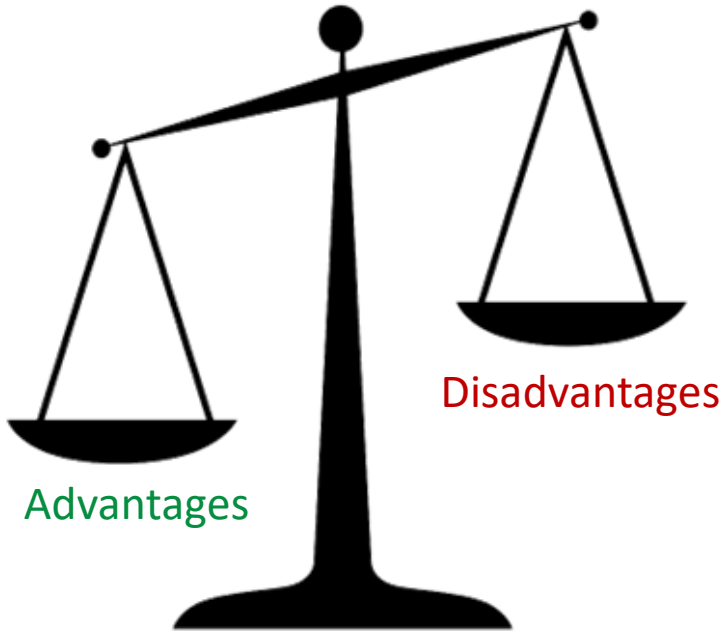
...

11-й шаг



AdaBoost – достоинства и недостатки

- Эффективный с вычислительной точки зрения
- Позволяет решать сложные задачи, которые плохо решаются отдельными алгоритмами
- Простой с точки зрения программирования Только один параметр настройки - число итераций
- Не требует априорной информации о слабом классификаторе
- Обеспечивает во многих случаях высокую точность прогнозирования
- Прост для модификаций



- Слишком эффективные или сложные классификаторы могут привести к переобучению
- Чрезмерная чувствительность к выбросам
- Громоздкие композиции из сотен алгоритмов не интерпретируемы
- Требуются достаточно большие обучающие выборки
- Не удается строить короткие композиции из «сильных» алгоритмов типа SVM (только длинные из слабых)
- Слишком “слабые” слабые классификаторы плохо работают

Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

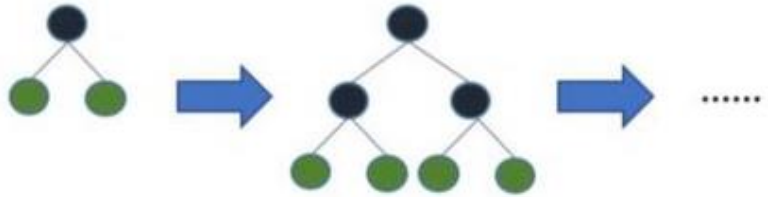
Практика

Современные реализации

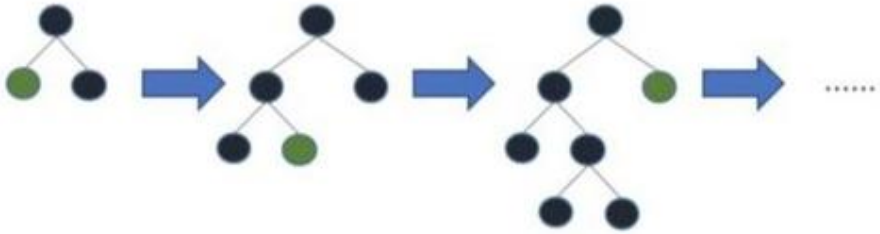
Реализации бустинга

XGBoost & LightGBM

dmlc
XGBoost

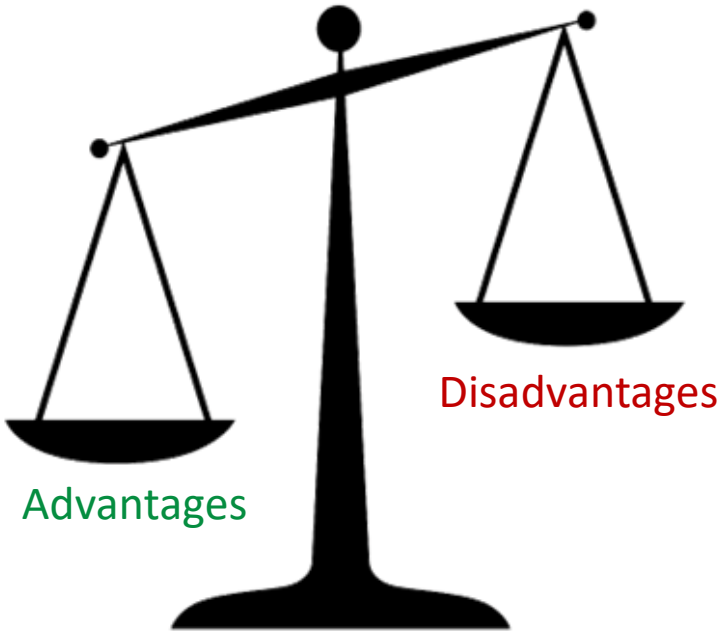


LightGBM



Gradient Boosting – достоинства и недостатки

- **Высокая точность**: Один из лучших результатов среди классических алгоритмов ML
- **Гибкость**: Может использовать различные функции потерь и легко адаптируется к задачам регрессии, классификации, ранжирования.
- **Обработка пропущенных данных**: Модели градиентного бустинга могут эффективно работать с пропущенными значениями в данных.



- **Длительное время обучения.**
- **Переобучение**: Если не контролировать глубину деревьев и количество итераций, модель может легко переобучиться.
- **Чувствительность к гиперпараметрам**: Модель очень чувствительна к настройке параметров, таких как learning_rate и n_estimators.
- **Проблемы с интерпретацией**: Модели градиентного бустинга часто трудно интерпретировать по сравнению с простыми моделями.

Вопросы?

Ставим “+”,
если вопросы есть

Ставим “–”,
если вопросов нет

Список материалов для изучения

Градиентный бустинг
https://education.yandex.ru/handbook/ml/article/gradientnyj-busting

Рефлексия

Заполните, пожалуйста, опрос о занятии по ссылке в чате

Спасибо за внимание!

Приходите на следующие вебинары

21.08 – Метод опорных векторов

Андрей Канашов

Team Lead Data Scientist в ПИК

- Ценообразование и тарификация
- Рекомендательные системы
- Прогнозирование ключевых метрик
- Анализ клиентского поведения