IT-Academy

BOOSTING

КУХАЛЬСКИЙ НИКОЛАЙ ГЕННАДЬЕВИЧ

Вопросы занятия

1. Бустинг и его виды;
2. Реализация бустинга;
3. Особенности XGBoost;
4. Особенности CatBoost.

В конце занятия научимся:

- понимать как работает бустинг;
- применять нужный алгоритм бустинга на практике;
- использовать XGBoost и настраивать его параметры
- использовать CatBoost и настраивать его параметры.

IT-Academy

BOOSTING
и его виды

IT-Academy

БУСТИНГ: ОСНОВНАЯ ИДЕЯ

– ансамбль слабых базовых алгоритмов (weak learner);

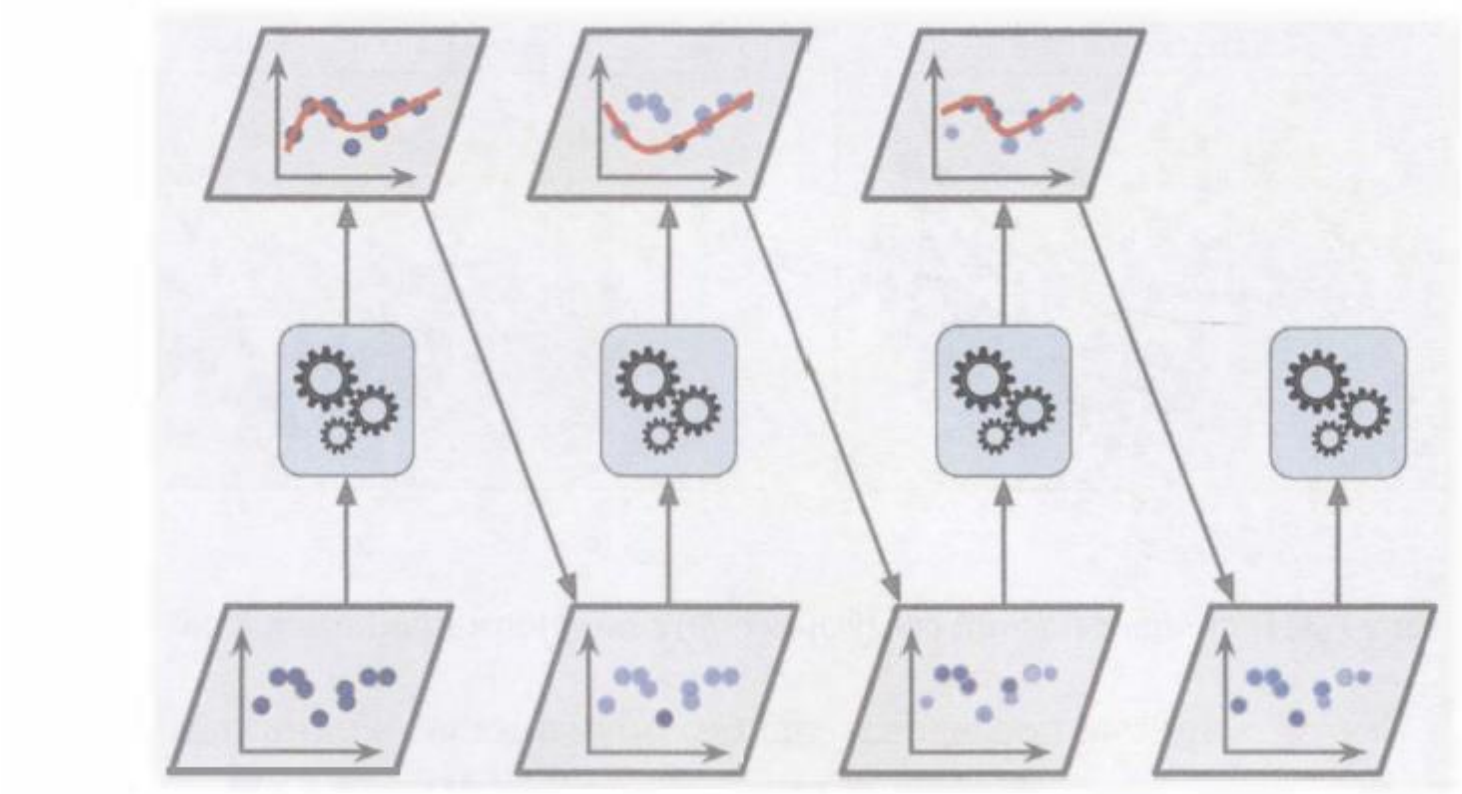
– слабый алгоритм - точность чуть лучше случайного;

– базовые алгоритмы обучаются последовательно;

– на каждом следующем шаге учитывается ошибка предыдущего;

ADABOOST

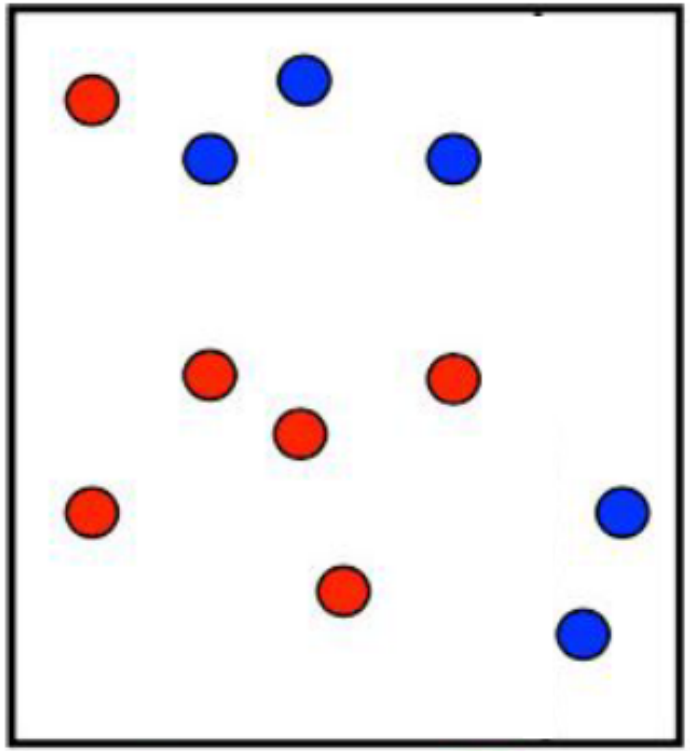
Последовательное обучение в методе AdaBoost с обновлением весов образцов



ADABOOST

IT-Academy

ВИЗУАЛИЗАЦИЯ ADABOOST



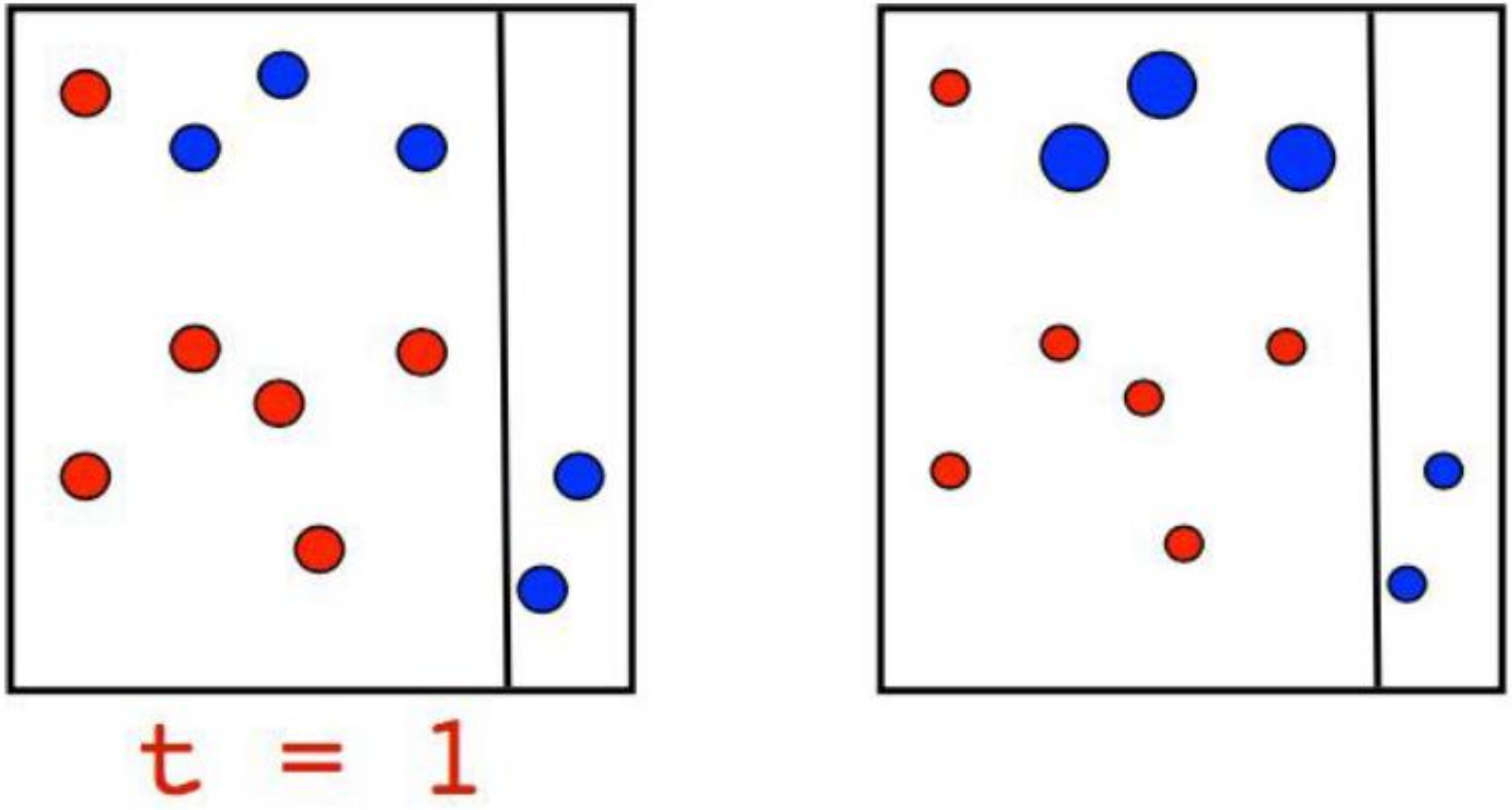
* https://habr.com/company/ods/blog/327250/

(подробная математика по ссылке)

ADABOOST

IT-Academy

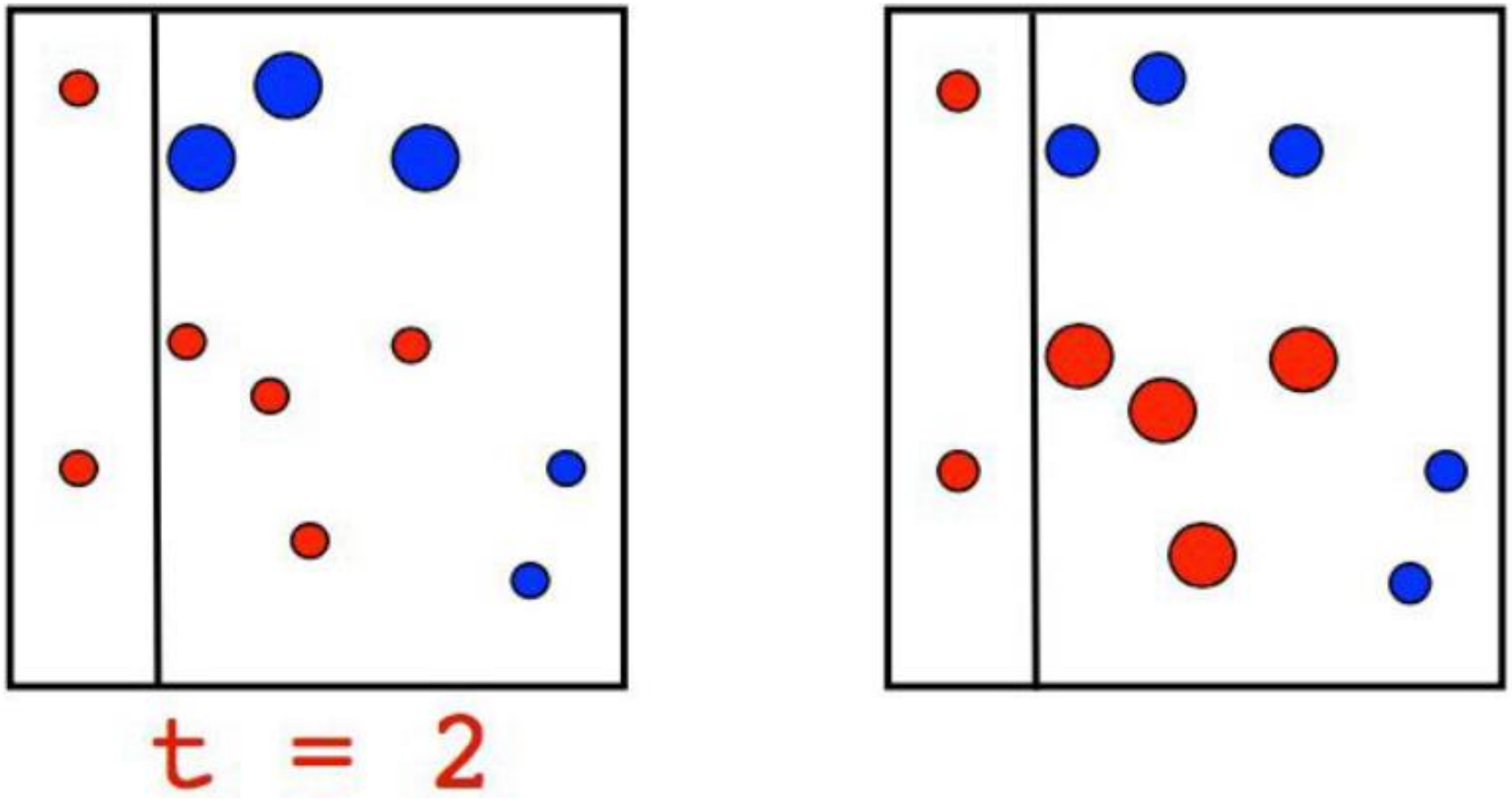
ВИЗУАЛИЗАЦИЯ ADABOOST



ADABOOST

IT-Academy

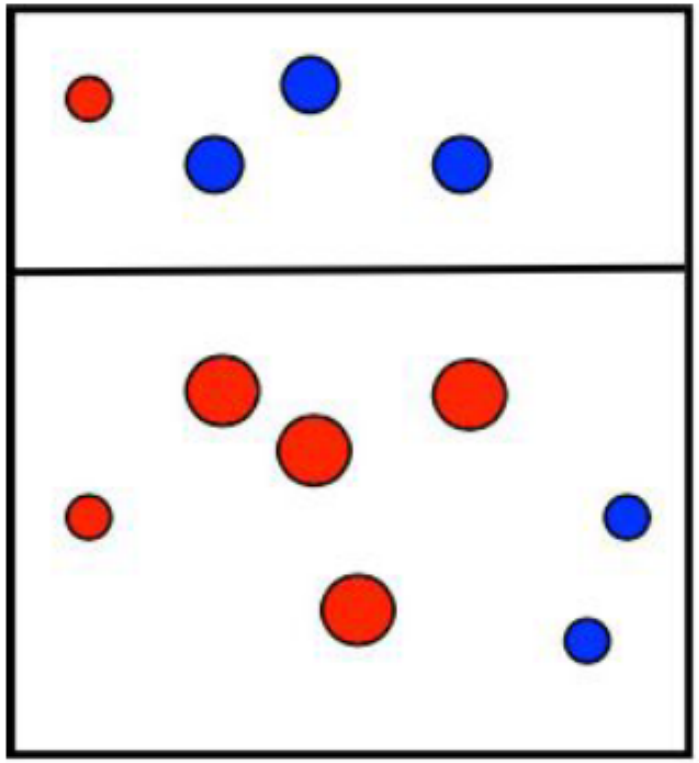
ВИЗУАЛИЗАЦИЯ ADABOOST



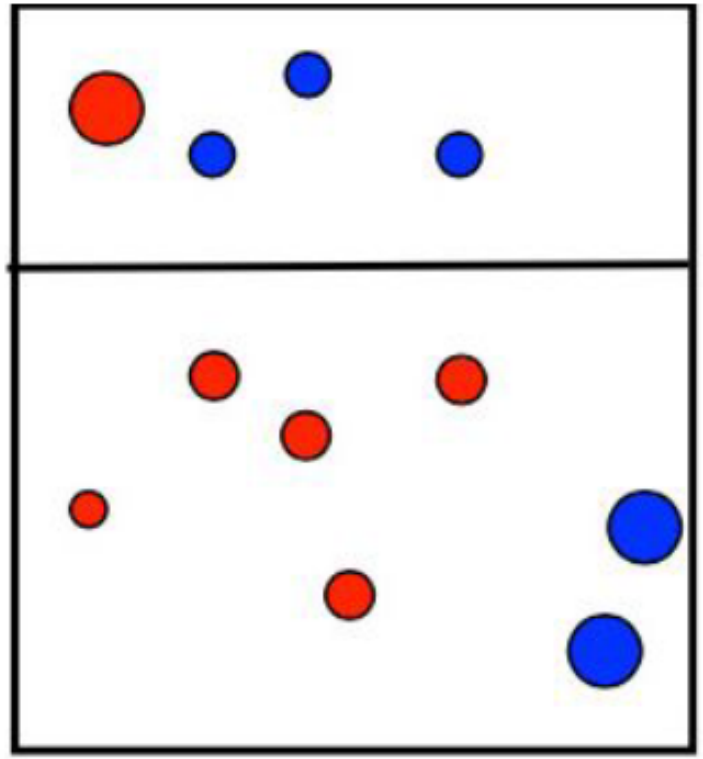
ADABOOST

IT-Academy

ВИЗУАЛИЗАЦИЯ ADABOOST

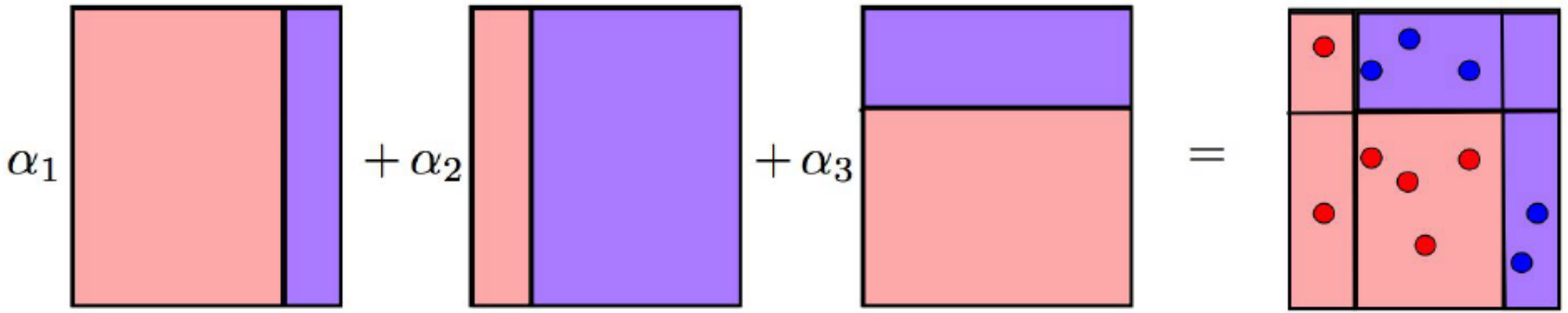


t = 3



IT-Academy

ВИЗУАЛИЗАЦИЯ ADABOOST



IT-Academy

ГРАДИЕНТНЫЙ БУСТИНГ

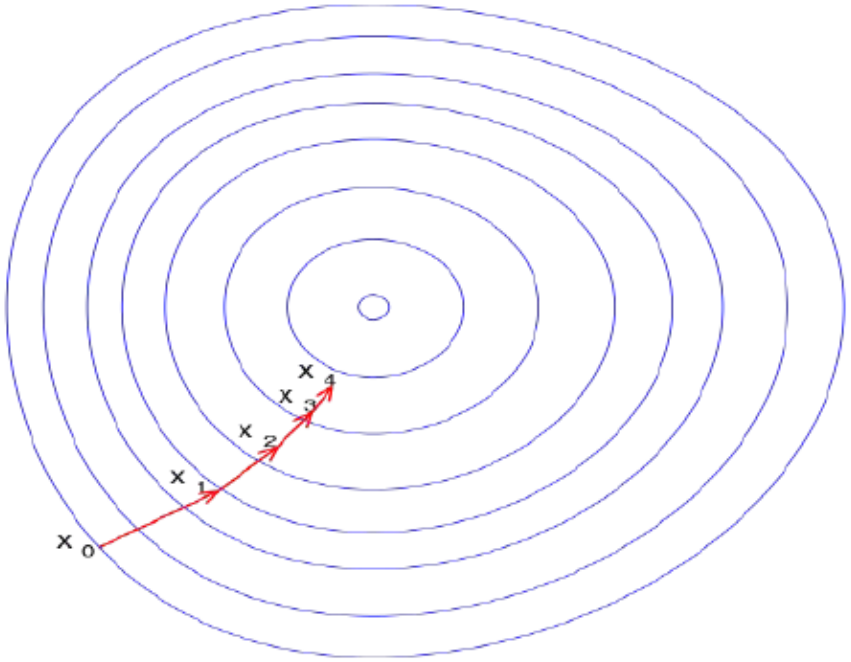
– gradient boosting machine (GBM);

– на каждом шаге базовый алгоритм настраивается на минимизацию ошибки алгоритма, полученного на предыдущем шаге;

GRADIENT BOOSTING

IT-Academy

ГРАДИЕНТНЫЙ СПУСК



$$\theta_j \leftarrow \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

IT-Academy

ОБОЗНАЧЕНИЯ

Алгоритм вида $\hat{f}(x) = \sum_{i=0}^{M} \hat{f}_i(x)$

Функция потерь $L(y, f)$

Базовый алгоритм вида: $h(x, \theta)$,

GRADIENT BOOSTING

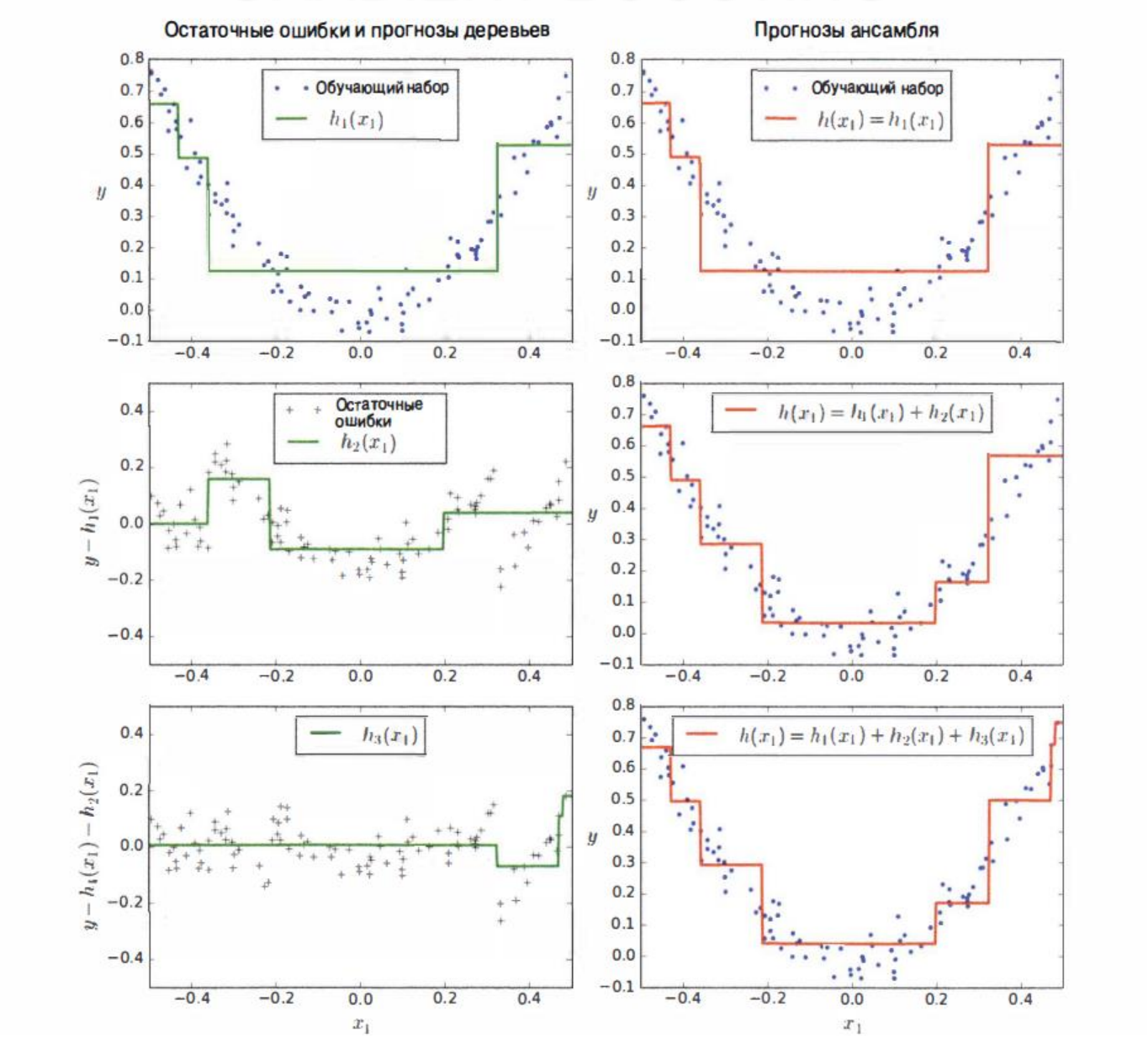
IT-Academy

АЛГОРИТМ

1. Инициализировать GBM константным значением $\hat{f}(x) = \hat{f}_0$,
2. Для каждой итерации $t = 1, \ldots, M$
   a. Посчитать псевдо-остатки $r_{it} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x)=\hat{f}(x)}$
   b. Обучить новый базовый алгоритм $h_t(x)$ на псевдо-остатках
   c. Найти оптимальный коэффициент при относительно исходной функции потерь $\rho_t = \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i) + \rho \cdot h(x_i, \theta))$
   d. $\hat{f}_t(x) = \rho_t \cdot h_t(x)$
   e. Обновить текущее приближение $\hat{f}(x) \leftarrow \hat{f}(x) + \hat{f}_t(x)$
3. Итоговая GBM модель:

$$\hat{f}(x) = \sum_{i=0}^{M} \hat{f}_i(x)$$

GRADIENT BOOSTING



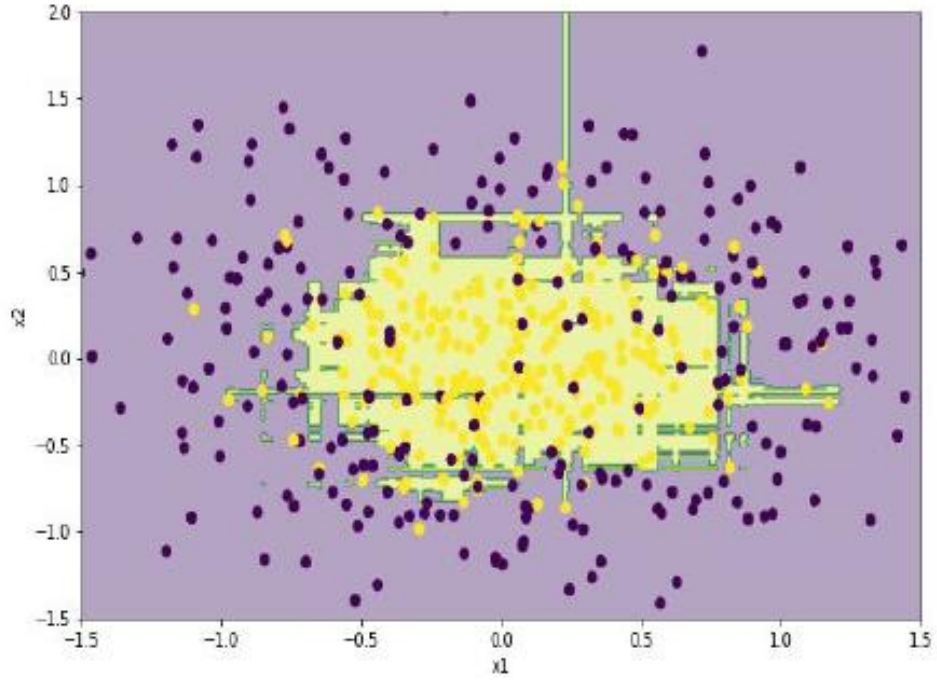
IT-Academy

БУСТИНГ И ЕГО ВИДЫ

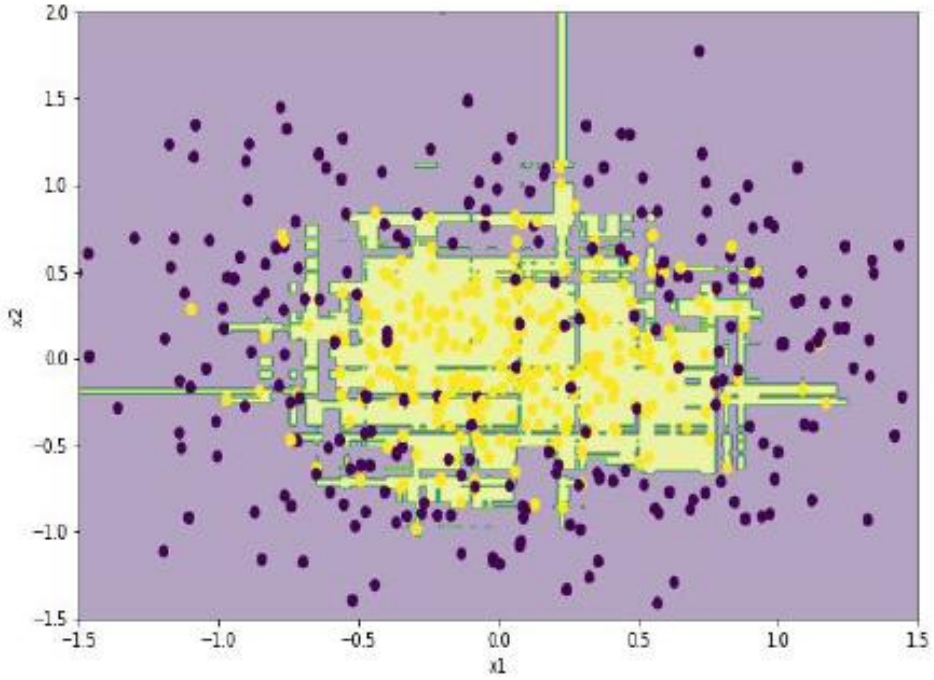
IT-Academy

РАЗДЕЛЯЮЩАЯ ПЛОСКОСТЬ

GBM



AdaBoost



РЕАЛИЗАЦИЯ БУСТИНГА

IT-Academy

SKLEARN ADABOOST

Реализация sklearn.ensemble.AdaBoostClassifier/Regressor

- **base_estimator** – базовый алгоритм (не обязательно дерево)
- **n_estimators** – кол-во базовых алгоритмов
- **learning_rate** – шаг бустинга

РЕАЛИЗАЦИЯ БУСТИНГА

IT-Academy

SKLEARN GBM

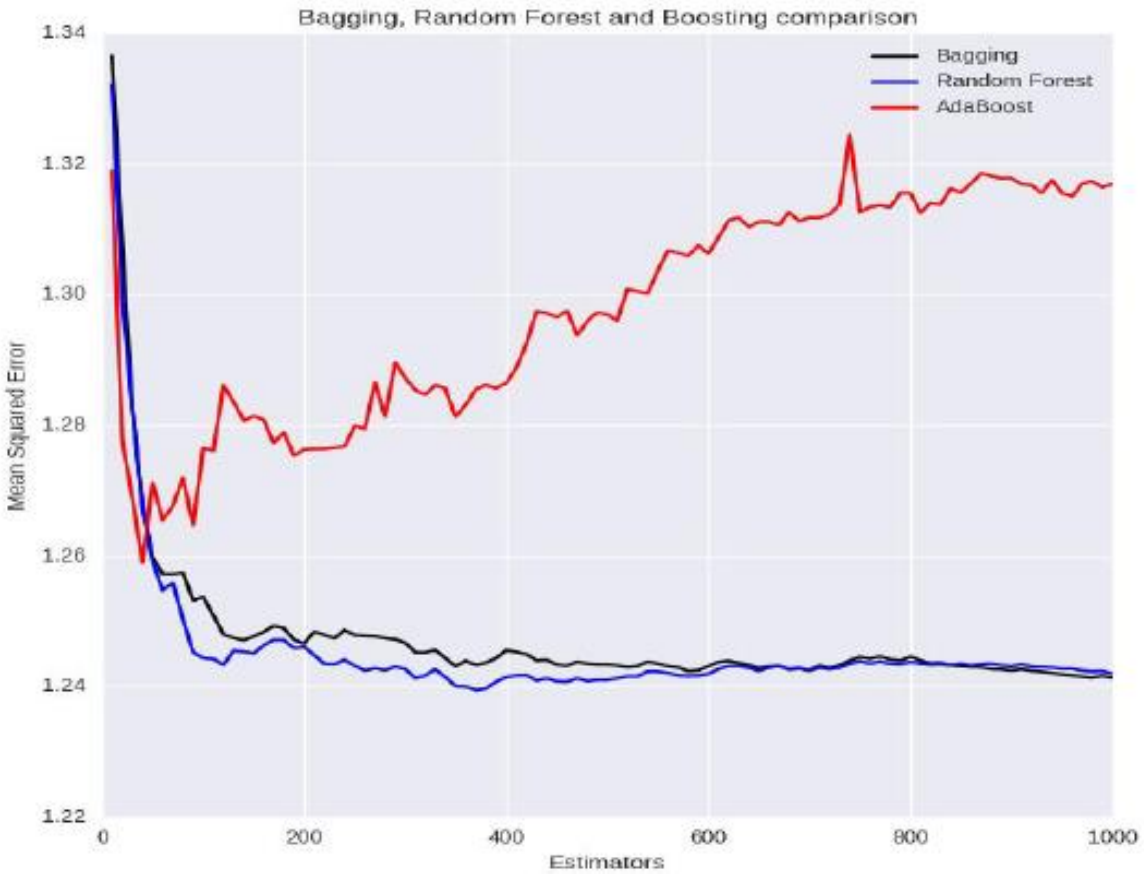
Реализация
sklearn.ensemble.GradientBoostingClassifier/Regressor

- **n_estimators** - кол-во деревьев
- **max_features** - доля признаков
- **subsample** - доля объектов
- **max_depth** – максимальная глубина дерева
- **min_samples_leaf** – минимальное число объектов в листе
- **learning_rate** – шаг бустинга

РЕАЛИЗАЦИЯ БУСТИНГА

IT-Academy

КОЛИЧЕСТВО ДЕРЕВЬЕВ



РЕАЛИЗАЦИЯ БУСТИНГА

IT-Academy

ПРОДВИНУТЫЕ РЕАЛИЗАЦИИ

	XGBoost	LightGBM	CatBoost
Разработчик	• независимый разработчик	• Microsoft	• Yandex
Плюсы	• скорость • качество • интерфейс	• скорость (самый быстрый)	• работа с категориальными переменными
Минусы		• недостаточно гибкий • плохой интерфейс	

IT-Academy

ПРАКТИКА

XGBoost.ipynb

IT-Academy

ПОЧЕМУ ВСЕ ЛЮБЯТ XGBOOST

- очень хорошее качество;
- большой выбор параметров для настройки;
- быстрая, параллельная реализация;
- продвинутый алгоритм оптимизации;
- возможность оптимизации кастомной функции;
- встроенная оценка важности признаков;

IT-Academy

ПАРАМЕТРЫ ДЕРЕВЬЕВ

Реализация xgboost.XGBClassifier

- **max_depth** - максимальная глубина дерева (обычно 3-10, больше глубина -> больше риск переобучения)
- **min_child_weight** - минимальное число объектов в листе (обычно до 20, больше объектов -> меньше риск переобучения, но должен быть согласован с глубиной дерева)
- **gamma** - минимально необходимый прирост качества для разбиения листа (редко используется)

IT-Academy

ПАРАМЕТРЫ БУСТИНГА

Реализация xgboost.XGBClassifier

- **objective** - оптимизируемый функционал (встроен для классификации и регрессии, можно написать свой дифференцируемый)
- **n_estimators** - кол-во базовых алгоритмов (чем меньше learning_rate, тем больше деревьев)
- **learning_rate** - шаг создания ансамбля (зависит от n_estimators, но обычно 0.01 - 0.1)

IT-Academy

ПАРАМЕТРЫ БУСТИНГА

Реализация xgboost.XGBClassifier

- **colsample_bytree** - доля признаков, случайно выбирающихся для построения дерева
- **subsample** - доля объектов, случайно выбирающихся для построения дерева
- **n_jobs** - кол-во потоков для одновременного построения деревьев

IT-Academy

ВАЖНОСТЬ ПРИЗНАКОВ

– **weight** - суммарное кол-во раз, когда признак использовался для разбиения вершины
– **gain** - средний прирост качества, когда признак использовался для разбиения вершины
– **cover** - среднее кол-во объектов, которые попадали в разбиение по признаку, когда он использовался для разбиения вершины

IT-Academy

ПРАКТИКА

XGBoost.ipynb
(настройка параметров)

IT-Academy

CatBOOST

CatBOOST

IT-Academy

Преимущества использования CatBoost:

- CatBoost позволяет проводить обучение на нескольких GPU.
- Библиотека позволяет получить отличные результаты с параметрами по умолчанию, что сокращает время, необходимое для настройки гиперпараметров.
- Обеспечивает повышенную точность за счет уменьшения переобучения.
- Возможность быстрого предсказания с применением модели CatBoost.
- Умеет под капотом обрабатывать пропущенные значения.
- Может использоваться для регрессионных и классификационных задач.

IT-Academy

Параметры обучения

- **loss_function или objective** – показатель, используемый для обучения. Есть регрессионные показатели, такие как среднеквадратичная ошибка для регрессии и logloss для классификации.
- **eval_metric** – метрика, используемая для обнаружения переобучения.
- **Iterations** – максимальное количество построенных деревьев, по умолчанию 1000. Альтернативные названия num_boost_round, n_estimators и num_trees.
- **learning_rate или eta** – скорость обучения, которая определяет насколько быстро или медленно модель будет учиться. Значение по умолчанию обычно равно 0.03.
- **random_seed или random_state**– случайное зерно, используемое для обучения.
- **l2_leaf_reg или reg_lambda** – коэффициент при члене регуляризации L2 функции потерь. Значение по умолчанию – 3.0.
- **bootstrap_type** – определяет метод сэмплинга весов объектов, например это может быть Байес, Бернулли, многомерная случайная величина или Пуассон.
- **depth** = глубина дерева.

IT-Academy

Параметры обучения

- **grow_policy** – определяет, как будет применяться жадный алгоритм поиска. Может стоять в значении SymmetricTree, Depthwise или Lossguide. По умолчанию SymmetricTree. В SymmetricTree дерево строится уровень за уровнем, пока не достигнет необходимой глубины. На каждом шаге листья с предыдущего дерева разделяются с тем же условием. При выборе параметра Depthwise дерево строится шаг за шагом, пока не достигнет необходимой глубины. Листья разделяются с использованием условия, которое приводит к лучшему уменьшению потерь. В Lossguide дерево строится по листьям до тех пор, пока не будет достигнуто заданное количество листьев. На каждом шаге разделяется нетерминальный лист с лучшим уменьшением потерь.

- **min_data_in_leaf или min_child_samples** – это минимальное количество обучающих сэмплов в листе. Этот параметр используется только с политиками роста Lossguide и Depthwise.

- **max_leaves или num_leaves** – этот параметр используется только с политикой Lossguide и определяет количество листьев в дереве.

- **ignored_features** — указывает на признаки, которые нужно игнорировать в процессе обучения.

- **nan_mode** – метод работы с пропущенными значениями. Параметры Forbidden, Min и Max. При использовании Forbidden наличие пропущенных значений вызовет ошибку. При использовании параметра Min пропущенные значения будут приняты за минимальные значения для данного признака. В Max пропущенные значения будут приняты как максимальные значения для данного признака.

IT-Academy

Параметры обучения

•**leaf_estimation_backtracking** – тип бэктрекинга, использующийся при градиентном спуске. По умолчанию используется AnyImprovement. AnyImprovement уменьшает шаг спуска до того, как значение функции потерь будет меньшим, чем оно было на последней итерации.

•**boosting_type** - схема бустинга. Она может быть простой для классической схемы градиентного бустинга или упорядоченной, что обеспечит лучшее качество на небольших наборах данных.

•**score_function** – тип оценки, используемой для выбора следующего разбиения при построении дерева. Cosine используется по умолчанию. Другие доступные варианты L2, NewtonL2 и NewtonCosine.

•**early_stopping_rounds** - если стоит True, устанавливает тип детектора переобучения в Iter и останавливает обучение, когда достигается оптимальное значение.

IT-Academy

Параметры обучения

- **classes_count** – количество классов для задач мультиклассификации.
- **task_type** – используете вы CPU или GPU. По умолчанию стоит CPU.
- **devices** - идентификаторы устройств GPU, которые будут использоваться для обучения.
- **cat_features** - массив с категориальными столбцами.
- **text_features** - используется для объявления текстовых столбцов в задачах классификации.

IT-Academy

ПРАКТИКА

XGBoost.ipynb
(CatBoost)

ЧТО МЫ СЕГОДНЯ УЗНАЛИ

IT-Academy

- Какие бывают виды бустингов;
- В чем преимущество XGBoost;
- Как оценивать важность признаков в XGBoost;
- Как правильно настраивать параметры XGBoost

IT-Academy

BOOSTING

КУХАЛЬСКИЙ НИКОЛАЙ ГЕННАДЬЕВИЧ