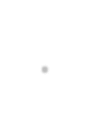
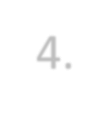
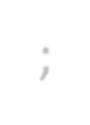
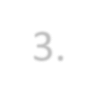
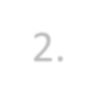
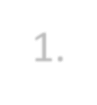
# BOOSTING

***Вопросы занятия***

1. Бустинг и его виды;
2. Реализация бустинга;

## 3. Особенности XGBoost;

4. Особенности CatBoost.



***В конце занятия научимся:***

− понимать как работает бустинг;

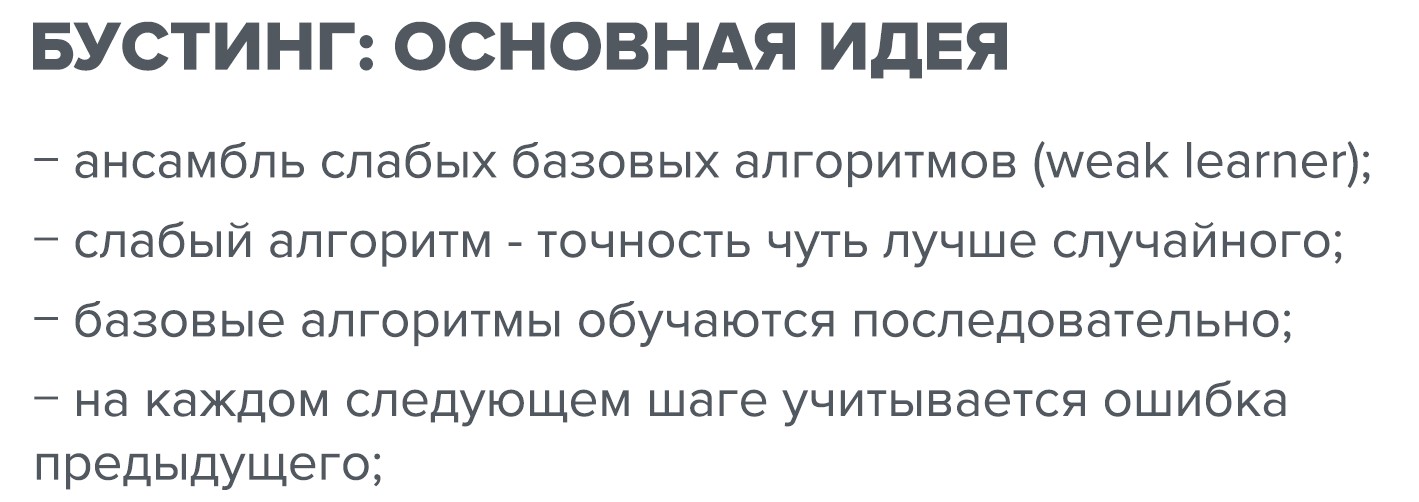
− применять нужный алгоритм бустинга на практике;

### − использовать XGBoost и настраивать его параметры

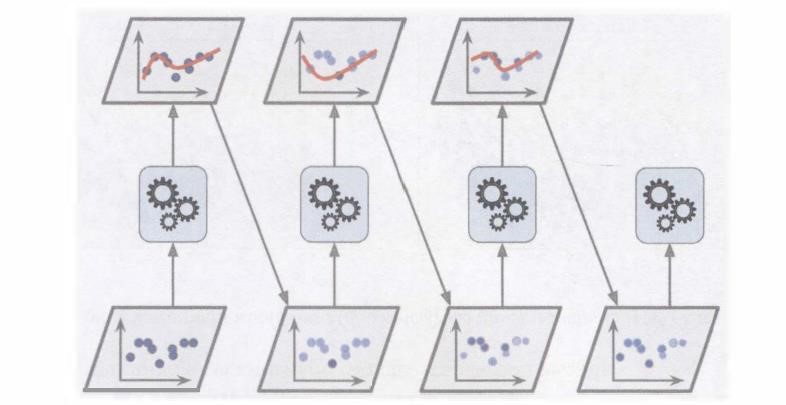
− использовать CatBoost и настраивать его параметры.

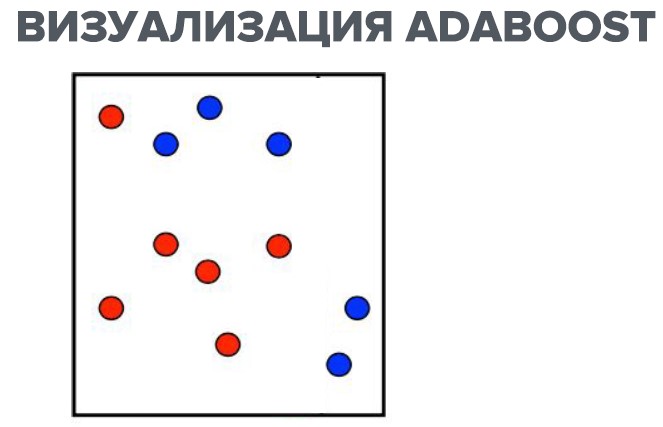
***BOOSTING и его виды***

*БУСТИНГ И ЕГО ВИДЫ*



Последовательное обучение в методе AdaBoost с обновлением весов образцов





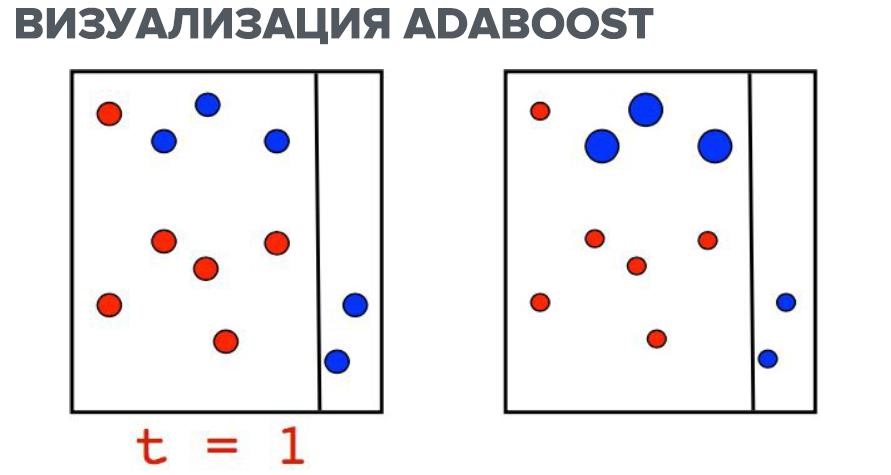
*\* https://habr.com/company/ods/blog/327250/*

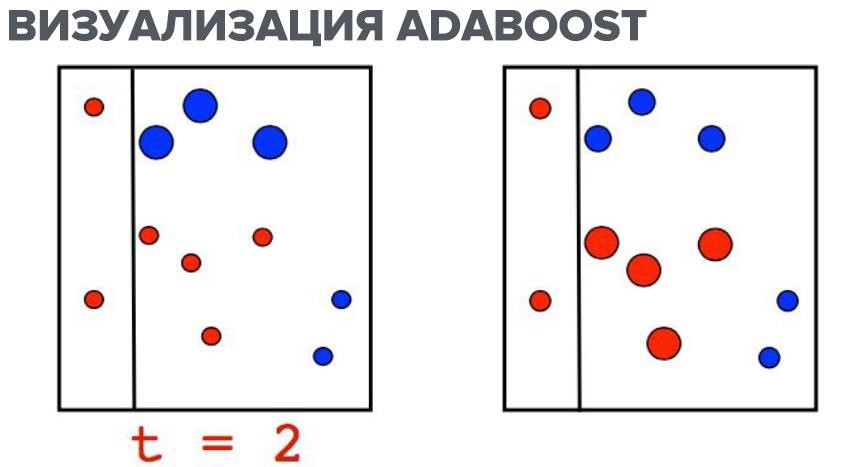


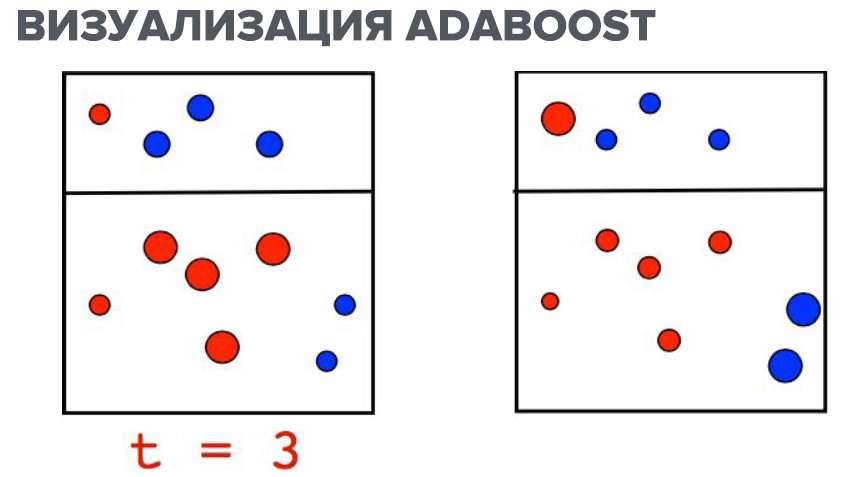
*(*

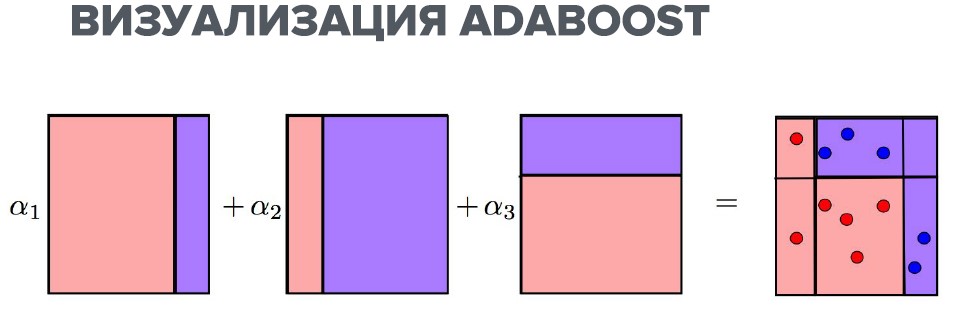
*подробная математика по ссылке*

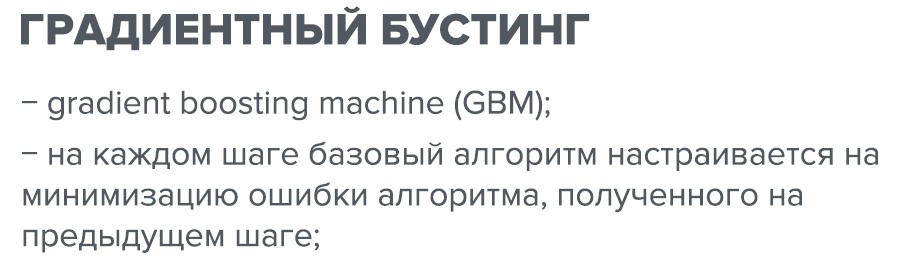
*)*



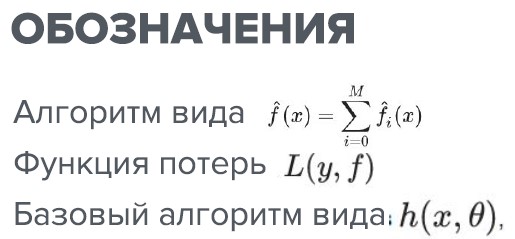


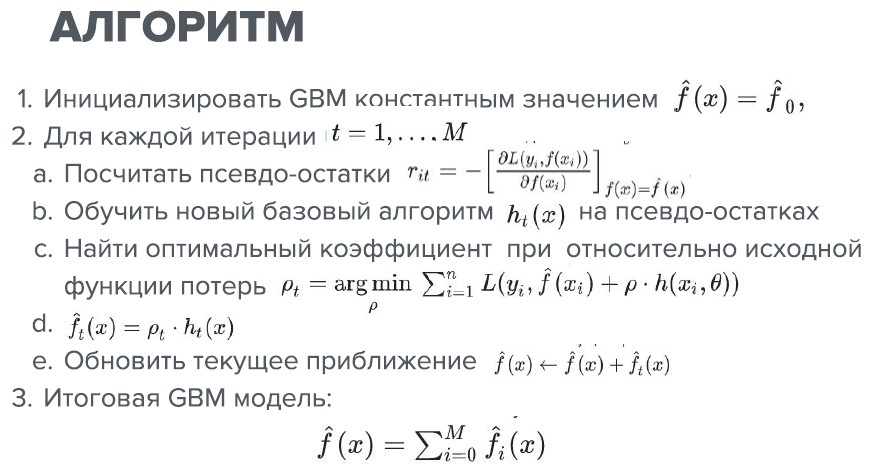


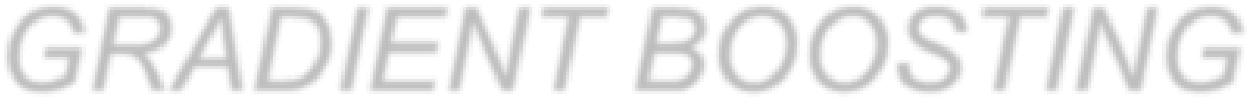
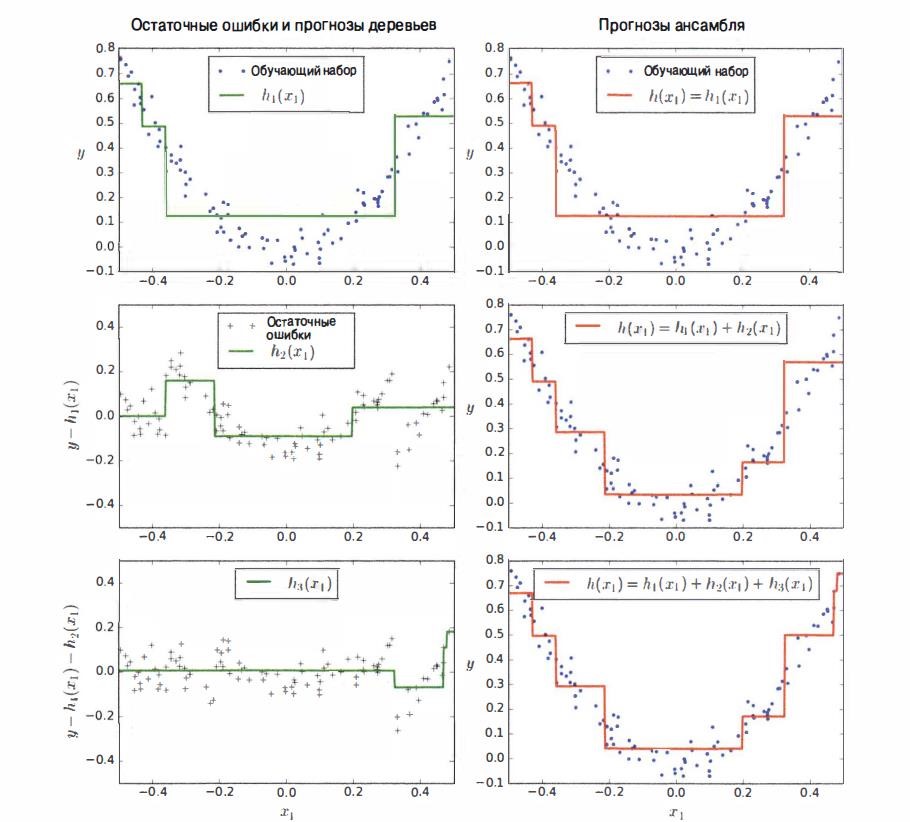






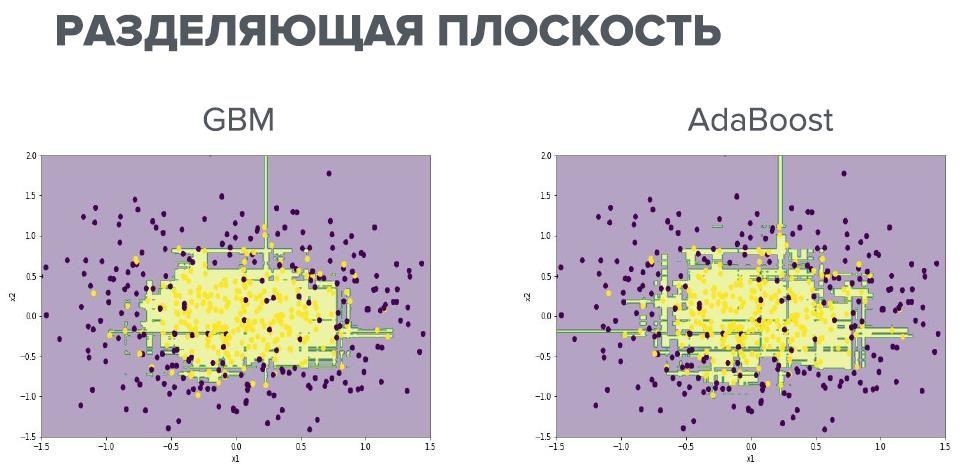


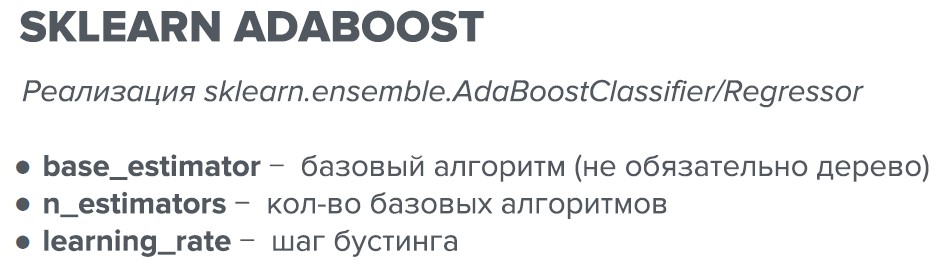


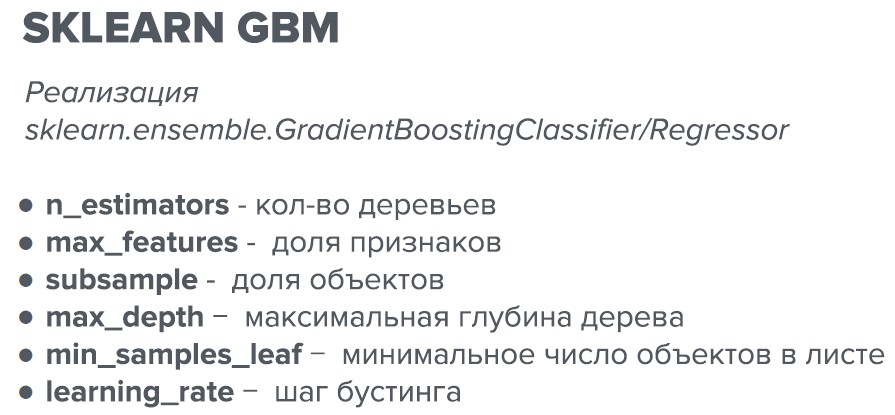


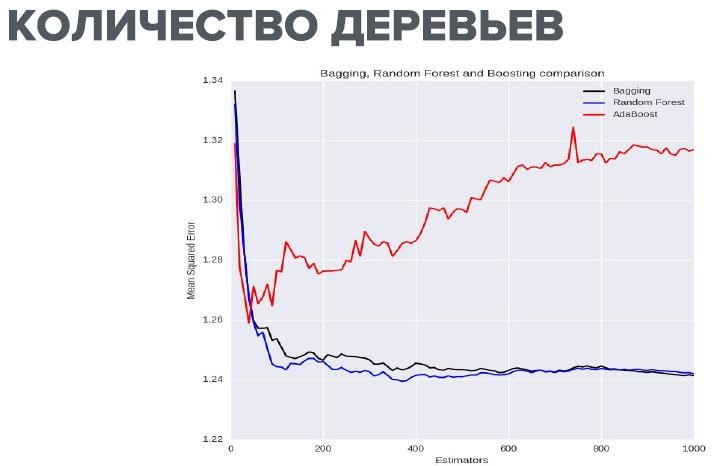
*GRADIENT BOOSTING*

*БУСТИНГ И ЕГО ВИДЫ*







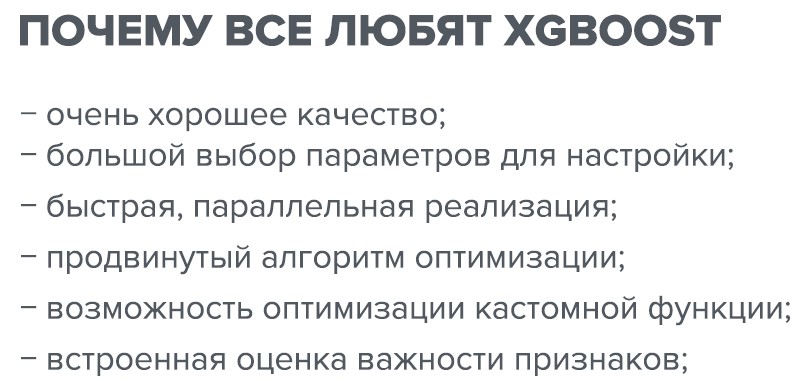


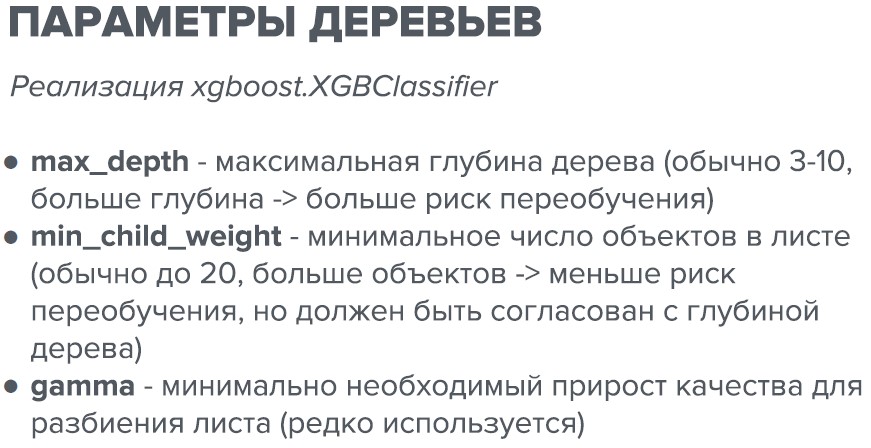


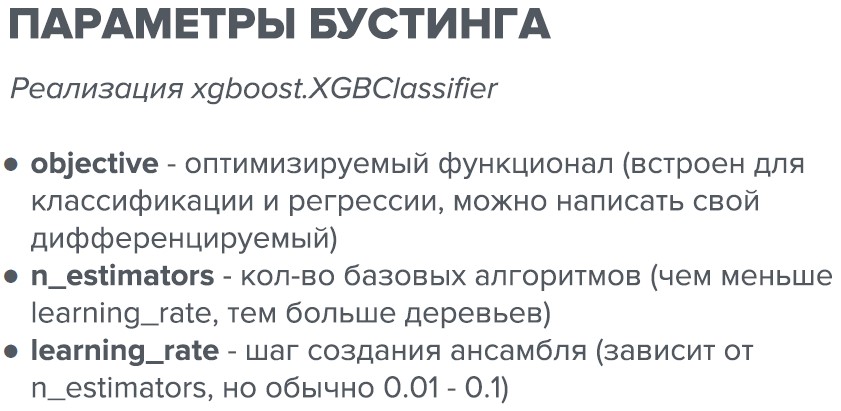
## ПРАКТИКА

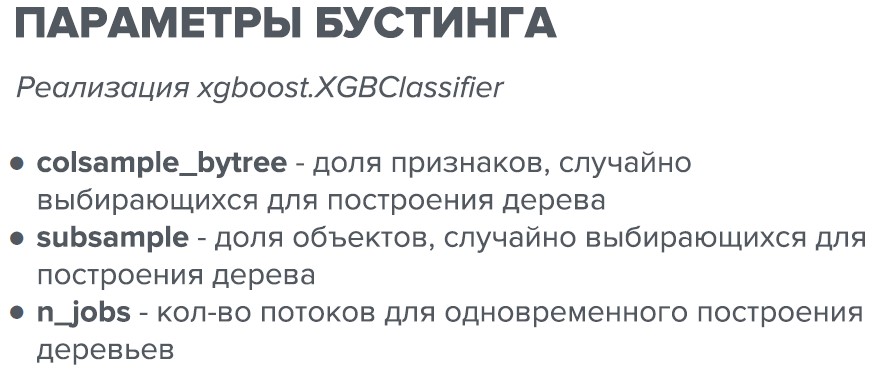
*XGBoost.ipynb*

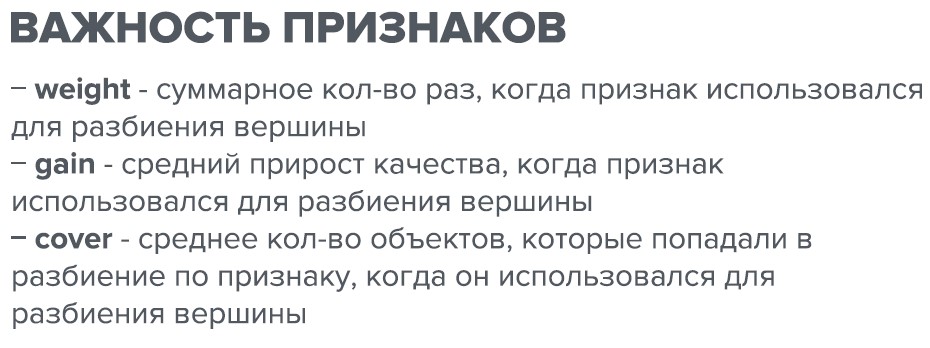














***ПРАКТИКА***



*XGBoost.ipynb*

*(*

*настройка параметров*

*)*

# CatBOOST

* CatBoost позволяет проводить обучение на нескольких GPU.



*CatBOOST*

*Преимущества использования CatBoost*



*:*

* 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.
* **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.
* **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.

•**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.

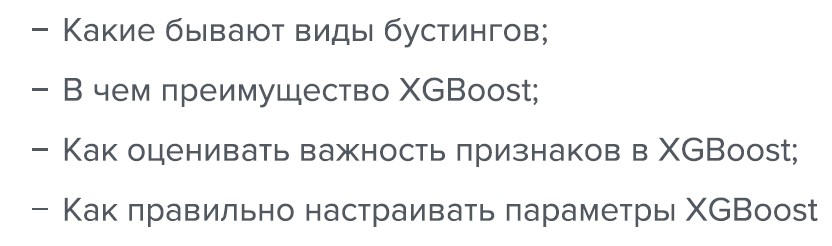
* **classes\_count**– количество классов для задач мультиклассификации.
* **task\_type** – используете вы CPU или GPU. По умолчанию стоит CPU.
* **devices**- идентификаторы устройств GPU, которые будут использоваться для обучения.
* **cat\_features**- массив с категориальными столбцами.
* **text\_features**- используется для объявления текстовых столбцов в задачах классификации.

## ПРАКТИКА

*XGBoost.ipynb*

*(CatBoost)*

*ЧТО МЫ СЕГОДНЯ УЗНАЛИ*



## BOOSTING