**CatBoost**: Основное отличие CatBoost — это встроенная обработка категориальных признаков. Большинство других библиотек требуют явной предобработки категориальных данных (например, через one-hot encoding или label encoding), что может привести к потере информации или неэффективной обработке. CatBoost использует метод счётчиков (mean encoding) и позволяет обучаться на данных в исходном виде без необходимости явного кодирования категорий. Это упрощает процесс и снижает вероятность переобучения.

<https://translated.turbopages.org/proxy_u/en-ru.ru.299530dc-6717d570-5d9aa0ef-74722d776562/https/www.geeksforgeeks.org/categorical-encoding-with-catboost-encoder/>

<https://habr.com/ru/companies/otus/articles/778714/>

<https://diapazon-plus.ru/b/kak-catboost-vzaimodejstvuet-s-kategorialnymi-priznakami-i-obespechivaet-maksimalnuyu-tochnost-modelirovaniya#metod-raboty-catboost-s-kategorialnymi-priznakami>

<https://habr.com/ru/companies/ods/articles/327250/>

-

**CatBoost**: Использует уникальный способ борьбы с переобучением во время кодирования категориальных признаков, применяя стохастическое кодирование. Это помогает уменьшить смещение модели и предотвращает информацию о целевой переменной во время кодирования признаков, что способствует более стабильному обучению.

-

**CatBoost**: Хорошо работает с разреженными и несбалансированными данными. Благодаря продуманным алгоритмам работы с категориальными признаками и применению счётчиков, модель может эффективно обрабатывать такие данные без явной балансировки или ручной настройки.

-

CatBoost: Использует обратное boosting-обучение, которое обучает деревья не по порядку добавления, а с учётом ошибок предыдущих шагов и на всём множестве данных. Это делает модель менее склонной к переобучению и более устойчивой к случайным шумам в данных.

XGBoost и LightGBM: У этих алгоритмов отсутствует подобный механизм. Переобучение предотвращается в основном за счёт настройки гиперпараметров, таких как регуляризация, глубина деревьев, размер подвыборок.

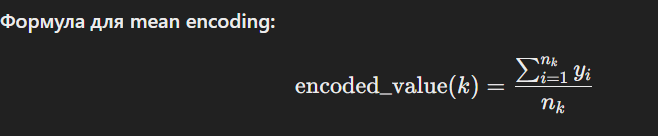
---------

**1. Что такое mean encoding?**

**Mean encoding (среднее кодирование)** — это техника преобразования категориальных признаков в числовые значения на основе статистики целевой переменной. Основная идея — заменить категорию на её среднее значение целевой переменной (например, для задачи классификации это может быть вероятность класса 1).

**Как работает mean encoding:**

* Для каждой категории вычисляется среднее значение целевой переменной по всем примерам с данной категорией.
* Например, в задаче бинарной классификации категория может быть заменена на вероятность того, что она относится к классу 1. Если категория "A" встречается 10 раз, и в 7 случаях целевая переменная равна 1, то для этой категории среднее значение будет 7/10=0.7.
* Эту технику также можно адаптировать для задач регрессии, где категорию заменяют на среднее значение целевой переменной для каждой категории.

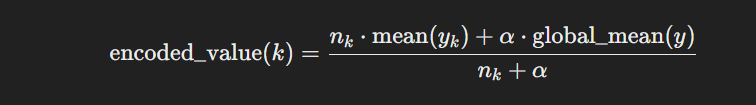


где:

* k — категория,
* yi​ — значение целевой переменной для i-го примера,
* nk — количество примеров с категорией k.

**Проблемы mean encoding:**

* **Переобучение**: Если категориальные признаки имеют много уникальных значений, эта техника может привести к переобучению, так как модель начинает слишком сильно зависеть от этих средних значений.
* **Устранение переобучения**: Для борьбы с этим часто применяют сглаживание — добавление значения, которое является глобальным средним для всех категорий. Формула сглаживания:

где α— гиперпараметр, который контролирует степень сглаживания.

-

### 3. ****Что такое стохастическое кодирование?****

**Стохастическое кодирование (Stochastic encoding)** — это модификация mean encoding, которая вводит случайность в процесс кодирования категориальных признаков для уменьшения переобучения. В традиционном mean encoding при кодировании каждой категории используется вся информация о данных, что может привести к утечке информации о целевой переменной.

**Как работает стохастическое кодирование:**

* Для того чтобы избежать утечки информации и переобучения, mean encoding для каждой категории вычисляется на подвыборке данных, исключая текущий пример.
* Это позволяет избежать использования информации из самого примера при кодировании категории, тем самым снижая вероятность переобучения.

**Пример:** Для каждого примера в данных вычисляется mean encoding на всех данных, кроме текущего примера. Это добавляет случайность в процесс кодирования и делает модель более устойчивой.

-

### 4. ****Что за счётчики, которые помогают работать с разреженными и несбалансированными данными?****

**Счётчики** в контексте градиентного бустинга — это статистики, которые вычисляются на категориальных признаках для улучшения их обработки. В CatBoost они называются target statistics, и их основная цель — обеспечить эффективную обработку категориальных признаков и предотвратить переобучение.

**Как работают счётчики:**

* Для каждого категориального признака вычисляется не только его среднее значение, но и другие статистики, такие как количество встречаемых примеров с данной категорией.
* Эти статистики используются для регулирования вклада категориальных признаков в модель и снижения их влияния на переобучение.

**Пример:** Категория, которая встречается редко (разреженные данные), будет иметь меньший вес при обучении, что уменьшает её влияние на конечную модель. Это особенно важно для несбалансированных данных, когда одни категории появляются чаще, чем другие.

-

### 5. ****Что такое обратное boosting-обучение?****

**Обратное boosting-обучение (Ordered boosting)** — это особый метод обучения в CatBoost, который изменяет порядок, в котором обучаются деревья, чтобы избежать переобучения. Обычный градиентный бустинг добавляет деревья в модель по порядку, минимизируя ошибку, сделанную предыдущими деревьями. Однако это может привести к тому, что каждое следующее дерево слишком сильно подстраивается под текущие данные.

**Основная идея:**

* В обратном бустинге деревья обучаются на подмножествах данных, причём каждое дерево обучается только на тех данных, которые оно ещё не видело. Это помогает предотвратить переобучение, так как каждое новое дерево строится, не используя информацию о предыдущих деревьях напрямую.

-

### 6. ****Как подбираются гиперпараметры? Почему CatBoost хорошо работает из "коробки"?****

**Гиперпараметры в CatBoost** подбираются автоматически с помощью встроенных механизмов, которые регулируют важные аспекты модели, такие как глубина деревьев, скорость обучения, регуляризация и параметры для работы с категориальными признаками.

**Почему CatBoost хорошо работает из "коробки":**

* **Автоматическая обработка категориальных признаков**: CatBoost автоматически кодирует категориальные признаки и подбирает параметры для этого процесса, что существенно упрощает подготовку данных.
* **Сглаживание и счётчики**: Эти механизмы позволяют CatBoost лучше работать с разреженными и несбалансированными данными, автоматически регулируя вклады редких категорий и предотвращая переобучение.
* **Обратное boosting-обучение**: Этот подход к обучению моделей делает процесс устойчивым к переобучению без необходимости точной настройки гиперпараметров.
* **Быстрая адаптация**: Благодаря встроенной оптимизации на уровне работы с GPU и многопоточности, модель может эффективно использовать ресурсы для ускорения обучения и улучшения точности.