

Санкт-Петербургский государственный университет

*Королихин Владимир Игоревич*

Выпускная квалификационная работа

Применение машинного обучения для  
улучшения модели трат автоматических  
стратегий назначения ставок в рекламных  
технологиях

Уровень образования: магистратура

Направление *02.04.03 «Математическое обеспечение и администрирование  
информационных систем»*

Основная образовательная программа *ВМ.5665.2022 «Математическое обеспечение и  
администрирование информационных систем»*

Профиль *Информационно-аналитические системы*

Научный руководитель:  
к.ф.-м.н., доц. Е.Г.Михайлова

Рецензент:  
старший разработчик ООО "Яндекс.Технологии" В.Ю. Каманов

Санкт-Петербург  
2024

Saint Petersburg State University

*Vladimir Korolikhin*

Master's Thesis

Application of machine learning to enhance  
automatic bid strategy models in advertising  
technologies

Education level: master

Speciality *02.04.03 "Software and Administration of Information Systems"*

Programme *BM.5665.2022 "Software and Administration of Information Systems"*

Scientific supervisor:  
C.Sc., docent E.G. Mikhailova

Reviewer:  
Senior Developer at LLC Yandex Technologies V.Y. Kamanov

Saint Petersburg  
2024

# Оглавление

<b>Введение</b>	<b>4</b>
<b>1. Постановка задачи</b>	<b>6</b>
<b>2. Обзор</b>	<b>7</b>
2.1. Процесс запуска рекламной кампании . . . . .	7
2.2. Обзор видов автостратегий . . . . .	8
2.3. Описание текущего аналитического подхода расчета ставок	9
2.4. Недостатки аналитических формул . . . . .	11
<b>3. Сбор данных</b>	<b>13</b>
3.1. Общий пайплайн . . . . .	13
3.2. Признаки по методу Холта - Винтерса . . . . .	14
3.3. Дополнительные эвристики . . . . .	15
3.4. Реализация системы сбора и обработки данных для обу- чения . . . . .	16
<b>4. Обучение моделей</b>	<b>20</b>
4.1. Выбор функции потерь . . . . .	20
4.2. Разделение таргета . . . . .	22
4.3. Обучение модели . . . . .	25
4.4. Метрики качества и замеры . . . . .	25
<b>5. Апробация модели</b>	<b>27</b>
5.1. Эксперименты в автобюджете . . . . .	27
5.2. Онлайн метрики и замеры . . . . .	27
<b>6. Результаты</b>	<b>31</b>
<b>Список литературы</b>	<b>32</b>

# Введение

В прошлом создание контекстной рекламы требовало от рекламодателя сложной ручной настройки. Для максимизации своих бизнес-показателей он должен был тщательно выбирать поисковые запросы, релевантные объявлению, а затем самостоятельно выставить ставки для участия в рекламном аукционе (алгоритм, решающий какую рекламу показать). Однако в последние годы произошел сдвиг в сторону автоматизации и упрощения работы с рекламой с использованием методов машинного обучения. Это привело к разработке автотаргетинга, который автоматически подбирает поисковые запросы для таргетирования рекламы, и автоматических стратегий, позволяющих рекламодателям доверить процесс эффективного расходования своих рекламных бюджетов рекламной площадке. В данной работе рассматриваются способы улучшения автоматических стратегий в Яндекс Директе [8]. Это сервис для размещения объявлений контекстной рекламы на Яндексе и на сайтах-партнерах его рекламной сети.

Рекламодатель может выбирать различные виды автостратегий и их параметры исходя из целей бизнеса и знания стоимости собственных продуктов или определенных событий. С технической точки зрения он накладывает ограничения на некоторую функцию, которую необходимо оптимизировать. Ее результатом является набор управляющих воздействий, использующихся непосредственно в аукционе. Кроме выдерживания ограничений, такая функция должна быть адаптивна к внезапным изменениям в ее параметрах. Например, рекламодатель может изменить среднюю цену за клик. Также необходимо учитывать ситуацию холодного старта - когда кампания только запустилась и данных, чтобы предугадать ее поведение в аукционе недостаточно. При такой постановке задачи точные аналитические формулы начинают работать не очень хорошо. Для решения этой проблемы лучше всего подходит машинное обучение, которое позволяет эффективно находить сложные и не поддающиеся формализации структурные закономерности входных данных. Таким образом, исследование способов перехода от ана-

литического подхода решения задачи к моделям машинного обучения является актуальной задачей.

Можно доказать, что оптимальной моделью трат, максимизирующей конверсии при ограничениях на недельный бюджет или среднюю цену конверсии является такая, в которой ставка остается постоянной на протяжении всей недели. Вследствие этого возникает задача минимизации дисперсии ставок. Кроме того, ставки рекламных кампаний подчиняются степенному закону, что означает с одной стороны большое количество маленьких кампаний и малое количество больших, а с другой разброс целевой переменной по абсолютной величине. Такие ограничения сужают набор функций потерь и приемочных метрик. Само значение ставки вычисляется на основе оцененных коэффициентов, которые посчитаны по накопленной статистике из прошлого или же характеристик рекламной кампании в случае отсутствия такой статистики. Таким коэффициентом может выступать, например, временной таргетинг.

В данной работе рассматривается подход по переходу от аналитических формул в алгоритмах выставления ставок к алгоритмам машинного обучения, которые бы могли оценивать отдельные коэффициенты этих формул. Они позволяют адаптировать рекламную кампанию к изменениям в аукционе, более точно проводить аналитику кампании и улучшить модель трат, приводя ее к оптимальному состоянию.

# 1. Постановка задачи

Целью данной работы является адаптация текущей аналитической формулы вычисления коэффициентов, участвующей в формулах управляющих воздействий автобюджета, на модель машинного обучения. Для достижения цели были поставлены следующие задачи:

1. Сделать обзор модели трат биддера автобюджета.
2. Реализовать сбор пула для обучения модели.
3. Подобрать приемочные метрики, функцию потерь и обучить модель.
4. Провести эксперименты, проанализировать результаты и внедрить полученную модель.

## 2. Обзор

В этом разделе рассматривается работа автостратегий, возможные способы прогнозирования ставок, а также принцип работы рекламы в Яндекс Директе.

### 2.1. Процесс запуска рекламной кампании

Чтобы запустить рекламу, рекламодатель должен сначала создать набор рекламных материалов, которые будут рассказывать потенциальным клиентам о преимуществах продуктов или услуг. Материалы включают в себя рекламные тексты, изображения или видео, а также набор ссылок, которые будут вести по рекламным банерам. Следующим шагом рекламодатель должен задать таргетинг - набор критериев, которые позволят определить аудиторию, на которую будет рассчитана реклама. Существует также автотаргетинг, который автоматически подбирает запросы пользователя под заданные рекламные банеры. По данному набору ограничений решается попадет ли реклама в рекламный хит. Хит это процесс загрузки страницы с рекламными банерами. В случае положительного решения реклама начинает участие в аукционе, который в этом хите проводится. В аукционе рекламные кампании конкурируют за показ на банере, а торгуются между собой при помощи ставок [2]. Алгоритм, который выбирает ставку в каждом хите называется биддинг. В данной работе рассматривается только этап биддинга. Будем считать, что рекламные материалы нам заданы, а на таргетинг повлиять никак не можем. Единственное чем мы можем управлять это ставками, меняя которые можем добиваться эффективности рекламы.

Исторически самым первым способом биддинга являлось ручное управление ставками, которое перекладывало на рекламодателя необходимость задавать ставку для каждой ключевой фразы. В зависимости от ключевой фразы, по которой произошел таргетинг выставлялась ставка с которой реклама заходит в аукцион.

В последние годы произошел переход в сторону автоматизации выставления ставок, которые называются автостратегиями. Рекламодатель

тель с помощью ограничений ставит оптимизационную задачу на алгоритм автостратегии, которые в свою очередь считают оптимальную ставку. Например рекламодатель может поставить задачу максимизировать конверсии при заданной средней цене конверсии и ограничения на недельный бюджет.

## **2.2. Обзор видов автостратегий**

### **2.2.1. Автостратегии, основанные на правилах**

Автостратегии, основанные на правилах похожи на стратегии с ручным управлением ставок, но используют дополнительный набор правил или эвристик, выставленных рекламодателем, которые определяют, как следует корректировать ставки.

Такие автостратегии часто используются в ситуациях, когда существуют конкретные ограничения или требования, которые должны быть выполнены. Например, рекламодатель может хотеть, чтобы его реклама показывалась только в определенное время суток или в определенные дни недели. Более сложные правила могут включать калибровку ставки на определенную величину, если показатель кликабельности по его объявлению падает ниже определенного порога.

Одним из преимуществ автостратегий, основанных на правилах, является то, что они часто относительно просты в реализации и могут быть настроены в соответствии с конкретными потребностями рекламодателя. Однако они могут не учитывать такие факторы как поведение пользователей, конкуренция и меняющиеся рыночные условия.

Автостратегии, основанные на правилах наиболее эффективны при использовании в сочетании с другими алгоритмами [9], в частности с машинным обучением.

### **2.2.2. Автостратегии, использующие машинное обучение**

Автостратегии также могут использовать машинное обучение. Такие алгоритмы учатся по историческим данным и корректируют ставки



в режиме реального времени на основе различных факторов, таких как размещение рекламы или таргетинг на аудиторию. Например в статье [1] изучен метод прогнозирования поведения ставок с использованием градиентного бустинга на решающих деревьях. Для последующих корректировок авторы используют линейную регрессию.

В работе [2] авторы использовали обучение с подкреплением для обучения модели автоматического назначения ставок, которая показала хорошие результаты на A/B тестировании.

К сожалению, переиспользовать какие-либо вышеперечисленные подходы не представляется возможным из-за сложившихся внутри компании процессов по обучению моделей, в частности используется Catboost [3]. С другой стороны, в данной работе не ставится цель предсказывать ставки, а лишь коэффициенты в формулах. Конечно, в будущем планируется перейти на полностью машинно-ориентированный подход, но это более сложная задача.

## 2.3. Описание текущего аналитического подхода расчета ставок

Рассмотрим, какой будет являться оптимальная ставка при следующих условиях:

### **Теорема.**

*Условие.* Если выполнено хотя бы одно из следующих ограничений:

- Задана средняя цена конверсии.
- Задано ограничение на период работы рекламной кампании.

*Утверждение.* Тогда оптимальной будет такая ставка, которая остается постоянной на протяжении всей работы рекламной кампании.

### **Доказательство.**

Для простоты, в качестве периода возьмем неделю.

На рисунке изображены хиты, которые произойдут в течение недели. Ширина столбика это количество конверсий, а высота это цена кон-

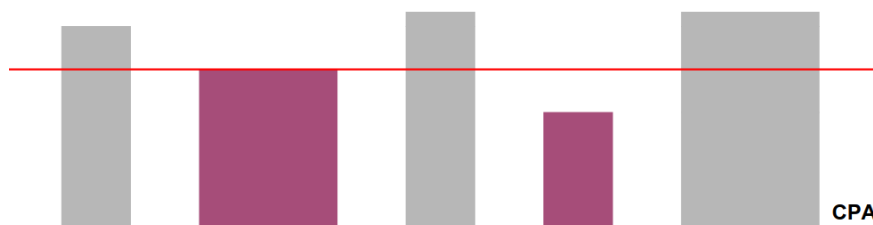


Рис. 1: количество конверсий и цена конверсии

версии. Таким образом график в терминах CPA, т.е. Cost per action - средняя цена конверсии.

Для начала покажем какой будет искомая ставка. Постепенно поднимая линию выше, мы будем покупать все более дорогие конверсии. Вследствие этого вырастет бюджет, который мы за неделю потратили и средняя цена конверсии. Таким образом, в какой-то момент мы достигнем одного из ограничений. Ставка, которая при этом получится и будет оптимальной.

Предположим существует другой алгоритм, который смог купить больше конверсий, чем наш. Значит он должен был купить какое-то количество конверсий над чертой, так как все что ниже купил наш алгоритм. Возможно он отказался от части конверсий, которые купил наш алгоритм, однако ему нужно их будет докупить из тех, что находятся над чертой и соответственно дороже. Следовательно вырастет либо потраченный недельный бюджет, либо средняя цена конверсии, что противоречит условию.

□

Примечание: в доказательстве алгоритма мы пользуемся предположением, что нам заранее известно ожидаемое количество конверсий в каждом хите.

В Яндекс Директе существует большое количество автостратегий. Можно максимизировать конверсии/установки/клики/показы с различными видами ограничений на среднюю цену события (например конверсий) и/или на недельный/периодный бюджет с оплатой за показы/клики/конверсии. Все перечисленные автостратегии объединяет общая схема подсчета ставок. Для определенности в работе рассматривается оп-

тимизация конверсий с ограничением на недельный бюджет.

Рассмотрим текущий аналитический подход расчета ставки для данного случая:

- Пусть  $T(t)$  - средняя интенсивность трафика, не зависящая от заказа, а зависящая от количества пользователей в интернете в данный момент времени. Например, ночью этот параметр будет меньше, чем днем.

$$\text{При этом: } \int_{week} T(t) dt = 1$$

- Тогда интенсивность трат в текущий момент времени ( $EventCost$ ) можно связать со ставкой на заказ ( $Bid$ ), где  $C$  - некоторая константа, зависящая от заказа, а  $P \geq 1$  - степень к ставке, которая позволяет захватывать более дорогие места показа, чтобы быстрее прийти к стационарному состоянию:

$$EventCost(t) = C \cdot Bid(t)^P \cdot T(t) \quad (1)$$

- Проинтегрировав формулу (1), можно выразить ставку на заказ на оставшийся период работы кампании:

$$Bid = \left( \frac{WeekLimit - SpentCost(now)}{C(now) \cdot \int_{now}^{week} T(t) dt} \right)^{\frac{1}{P}}, \quad (2)$$

где  $WeekLimit$  - ограничение на недельный бюджет, а  $SpentCost$  - траты к текущему моменту времени

- Аналогично можно выразить и константу  $C$ :

$$C(now) = \frac{\int_{now-week}^{week} EventCost(t) dt}{\int_{now-week}^{week} Bid(t)^P \cdot T(t) dt} \quad (3)$$

## 2.4. Недостатки аналитических формул

Ранее было показано, что в идеальном случае ставка на заказ должна быть постоянной на протяжении всей работы кампании. К сожалению, в реальном мире такого добиться очень сложно из-за множества проблем. Например:

- События связанные, с резким увеличением потока клиентов. Например, черная пятница.

- Изменения рекламодателем настроек кампании.
- Различные внутренние поломки алгоритмов рекламы

Для вычисления ставки, исходя из формулы (2) нужно знать  $C(now)$ , которая оценивается по историческим данным согласно формуле (3). Из этого вытекает несколько недостатков текущего подхода:

1. В случае сезонных скачков (например, черной пятницы) изменится и объем трафика, приходящий по данному заказу, что повлияет на ставки. Вернуться после таких изменений к прежним значениям может быть проблематично
2. В случае старта кампании в формуле (3) возникает неопределенность  $\frac{0}{0}$ , решить которую помогают прайорные значения в числителе и знаменателе. По мере получения статистики прайоры уменьшаются, однако точность  $C$  на старте кампании остается под сомнением.

Решить данную проблему позволит машинное обучение, которое сможет эффективно использовать имеющиеся характеристики кампании для предсказания начальных значений  $C$ . А в будущем и накопленную статистику для адаптивной реакции на сезонные изменения.

## 3. Сбор данных

В данном разделе описывается процесс сбора обучающего датасета, обзор основных признаков и выделение наиболее полезных из них.

### 3.1. Общий пайплайн

Для успешного обучения модели машинного обучения необходимо правильно организовать процесс сбора датасета. Для этого используются логи данных по всем рекламным кампаниям. Они являются источником информации о событиях, которые происходят во время показа рекламы пользователю и содержат разнообразные параметры, такие как затраты рекламодателя за клик пользователя по его баннеру, время, прошедшее с момента последнего показа и другие данные. Имеющиеся логи за определенный промежуток агрегируются по часам. Для этого используются ключевые атрибуты - номер кампании (OrderID) и час (Hour). Такой подход позволяет объединить данные для каждой кампании в пределах отдельных часовых интервалов. Агрегация сокращает количество обрабатываемых строк в последующем пайплайне. Это позволяет ускорить и оптимизировать процесс обработки данных. Назовем первый этап предобработки `Collect logs`.

Следующий этап сбора данных представляет собой программу редьюсера (`Make samples`), которая принимает логи данных по ключу OrderID и суммирует статистику в отсортированном по Hour порядке. В результате, накапливается кумулятивная сумма. В первом подходе было решено использовать два вида признаков: счетчиковые и оценочные. Счетчиковые признаки применяются к значениям, которые потенциально отражают активность рекламной кампании, такие как количество кликов или показов. Оценочные признаки, в свою очередь, используются для оценки целевой формулы с использованием аналитической формулы, которая была описана в прошлых разделах. Данные признаки можно посчитать с помощью кумулятивных сумм, взяв разность суммы за текущий момент времени и сумму, отстающую на заданное количество часов. В целях учета различных временных интервалов, про-

цесс формирования признаков осуществляется для разных окон, таких как 6 часов, 1 день, 3 дня и 7 дней. Это позволяет модели машинного обучения анализировать работу рекламной кампании в разные периоды времени и тем самым лучше оценить активность и эффективность ее расходов.

Последним этапом является добавление признаков (Prepare samples), которые не нуждаются в обработке редьюсером к данным, которые получаются на выходе из редьюсера. Это такие данные как заложенный недельный бюджет, количество оставшихся дней работы кампании и другие.

### 3.2. Признаки по методу Холта - Винтерса

Метод Холта-Винтерса [7] является расширением экспоненциального сглаживания для прогнозирования временных рядов, которые содержат сезонный компонент. В нашей задаче параметр  $C$  в силу несовершенства системы все же имеет определенный тренд, который можно аналитически прогнозировать и использовать как признак для модели.

Существует также множество других методов прогнозирования временных рядов, например ARIMA [6], STL [5], метод Калмана [4]. Однако, из-за ограниченного количества готовых решений и необходимости глубокого понимания алгоритма для написания собственной реализации было решено остановиться на методе Холта-Винтерса.

В его основе лежит идея, что будущее значение временного ряда можно оценить, используя три основных компонента временного ряда: уровень (level), тренд (trend) и сезонность (seasonality). Используем аддитивный вариант метода, в предположении, что амплитуда сезонных колебаний остается относительно постоянной во времени.

- Уровень ( $L_t$ ):

$$L_t = \alpha(y_t - S_{t-s}) + (1 - \alpha)(L_{t-1} + T_{t-1})$$

, где  $S_{t-s}$  — значение сезонного компонента в момент времени  $t - s$ ,  $s$  — длина сезонного периода,  $\alpha$  — коэффициент сглаживания для уровня

$(0 < \alpha \leq 1)$ .

- Тренд ( $T_t$ ):

$$T_t = \beta(L_t - L_{t-1}) + (1 - \beta)T_{t-1}$$

, где  $\beta$  — коэффициент сглаживания для тренда ( $0 < \beta \leq 1$ ).

- Сезонность ( $S_t$ ):

$$S_t = \gamma(y_t - L_t) + (1 - \gamma)S_{t-s}$$

, где  $\gamma$  — коэффициент сглаживания для сезонности ( $0 < \gamma \leq 1$ ).

- Прогноз:

$$\hat{y}_{t+m} = L_t + mT_t + S_{t-s+1+(m-1) \bmod s}$$

Данный алгоритм можно легко встроить в программу редьюсера, описанную выше. Используя все доступные данные, будем предсказывать аналитическое значение  $C$  на разные промежутки времени вперед и используем как признак для модели.

### 3.3. Дополнительные эвристики

Опишем некоторые эвристики, которые могут улучшить качество обучающего датасета:

- Так как подсчет признаков на этапе применения модели предполагает использование того же самого пайплайна, данные могут приходить с задержками или же наоборот слишком быстро. Это зависит от текущих вычислительных ресурсов, которые не постоянны. В связи с этим в программу редьюсера была добавлена возможность задавать случайный отступ признаков от таргета. Таким образом, эмулируется непостоянство времени доезда данных.
- У всех рекламных кампаний разные открукты, а потому могут возникать случаи, когда за окно, в котором считается таргет, может попадать минимальное количество статистики, например, к ним было всего несколько кликов. Такие примеры будут скорее вносить шум в данные и их лучше не включать в обучение.

- Использование информации о потенциальной возможности работы кампании в заданные часы может позволить модели быстрее реагировать на падение откруток. Таким образом, в случае отсутствия статистики в определенные часы, в пул добавляются примеры с нулевыми статистиками по кликам, показам, деньгам. Благодаря этому модель будет видеть падение откруток на кампании в случаях, когда этого не ожидалось.
- При помощи встроенной в CatBoost возможности определения наиболее значимых признаков, на основании того, какие из них оказывают больший вклад в уменьшение лосса, выбирались направления для дальнейшего инжиниринга новых признаков и подбора прайорных значений. После нескольких итераций модели на вход подавалось около 400 признаков

### 3.4. Реализация системы сбора и обработки данных для обучения

На рис. 2 изображена диаграмма компонент предложенного решения по сбору и подготовке данных для обучения моделей. Серым цветом выделены компоненты, реализованные в работе. Их описание было дано выше, теперь подробнее опишем реализацию.

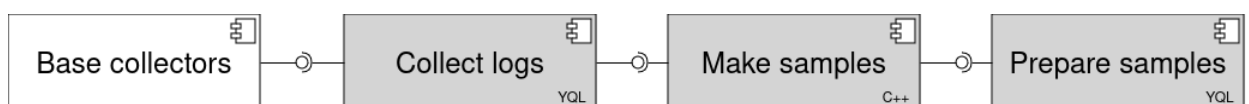


Рис. 2: Диаграмма компонент системы сбора и предобработки данных

#### 3.4.1. Компонента Collect logs

Здесь происходит первоначальная обработка данных из различных коллекторов - Base collectors. Base collectors это набор статических и динамических таблиц, содержащих информацию, например, о кликах, показах, ставках, настройках кампаний, рабочих часах. Эти данные лежат в разных таблицах, имеют разные типы данных, единицы измере-



ния. Необходимо их агрегировать и преобразовать в удобный формат для последующего расчета признаков.

Для реализации компоненты был выбран язык YQL <sup>1</sup>. Это универсальный, декларативный, основанный на SQL язык запросов. Он используется для запуска MapReduce задач в системе хранения и обработки больших данных YTsaurus <sup>2</sup>. Данная платформа является стандартом в компании.

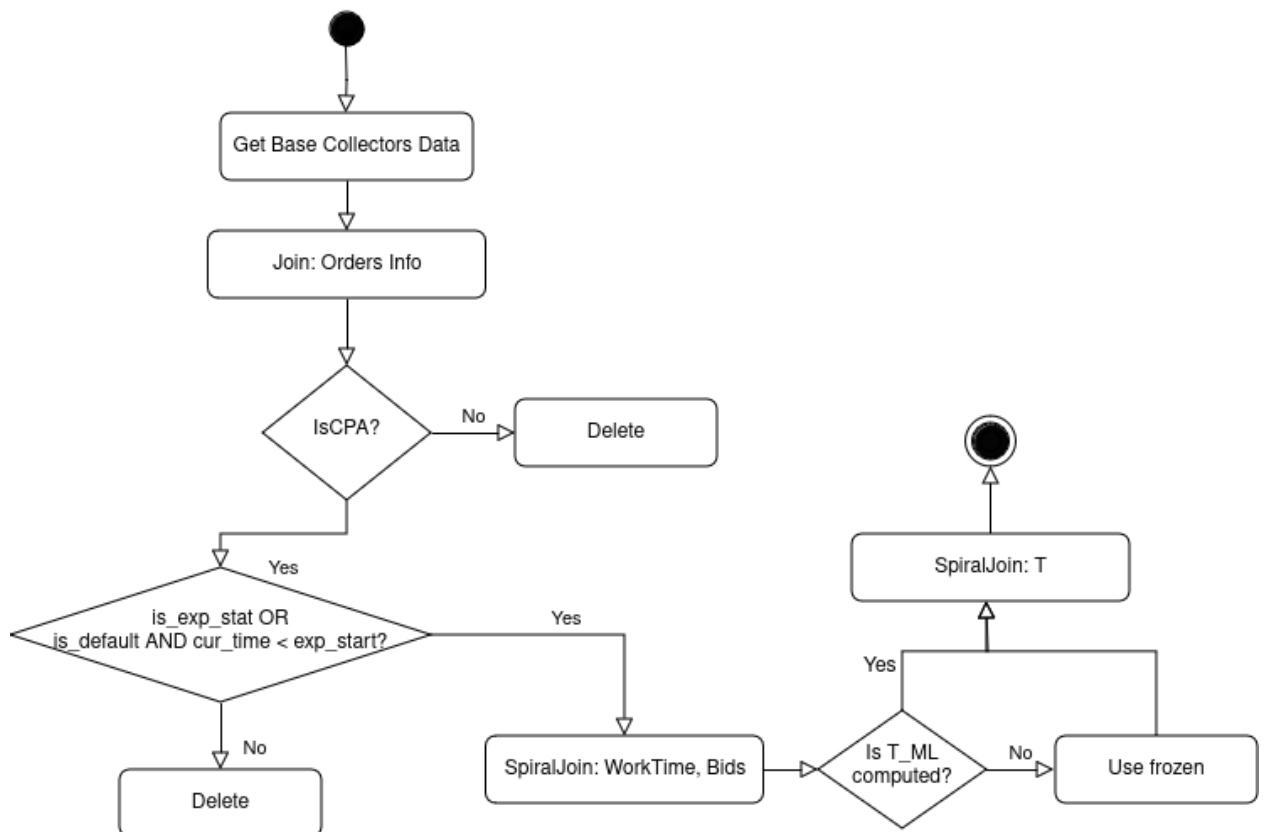


Рис. 3: Диаграмма компонент системы сбора и предобработки данных

На рис. 3 изображена диаграмма активностей описанной компоненты. Поля из логов базовых коллекторов преобразуются к нужному типу и домножаются на корректирующие коэффициенты. К ним добавляются поля-характеристики заказов. Так как в работе рассматриваются кампании с оптимизацией конверсий, то далее делается фильтрация по данному условию. После этого необходимо выделить статистику относящуюся к текущему эксперименту (подробнее про эксперименты бу-

<sup>1</sup><https://ytsaurus.tech/docs/ru/yql/index>, дата обращения: 11.05.2024.

<sup>2</sup><https://ytsaurus.tech/docs/ru>, дата обращения: 11.05.2024.

дет рассказно в части про апробацию), а в случае, когда эксперимент только недавно запустился взять статистику из дефолта. Далее необходимо выполнить спиральный джойн логов, содержащих рабочие часы кампании и ставки. Следующим этапом добавляется  $T$ , участвующая в формуле ставки. В зависимости от ее наличия могут браться зафиксированные константные значения.

### 3.4.2. Компонента Make samples

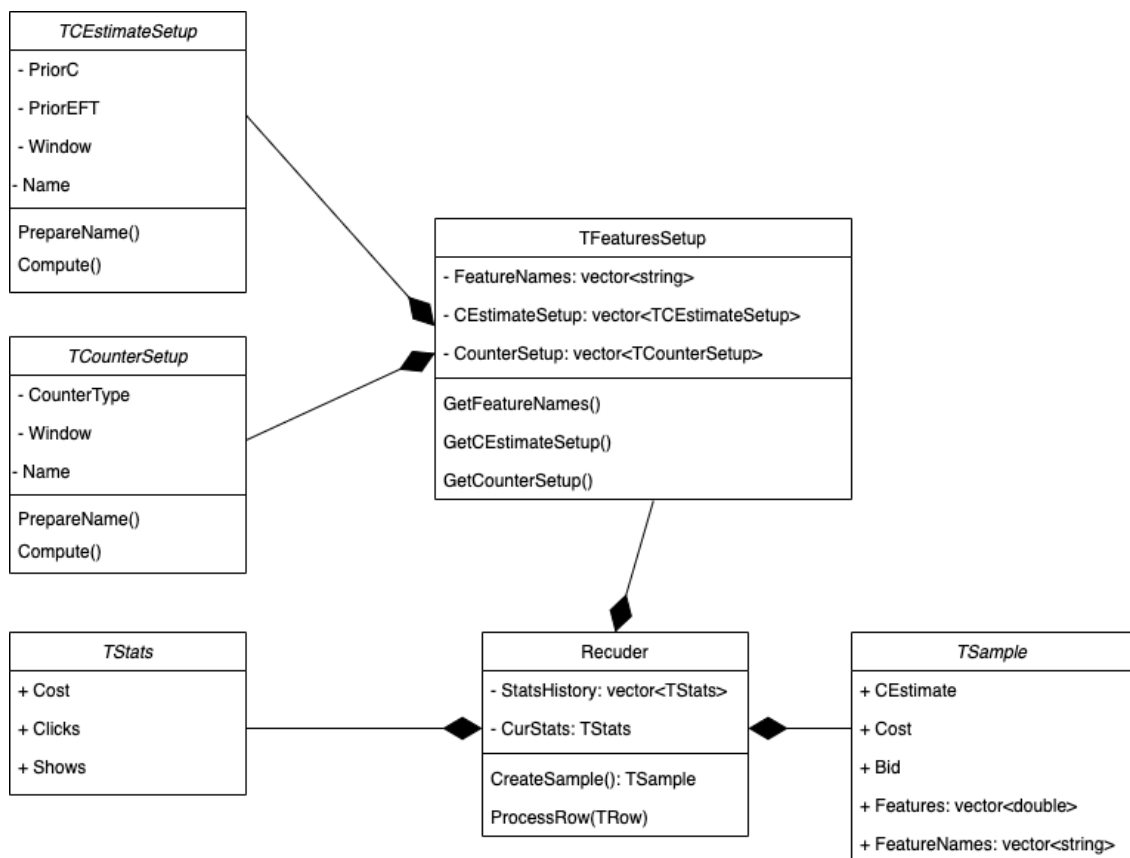


Рис. 4: Диаграмма классов компоненты make samples

На рис. 4 изображена диаграмма классов данной компоненты. После предобработки на стадии Collect logs логи поступают, агрегированные по часам в редьюсер, который по ключу OrderID в порядке Hour поступления данных TRow агрегирует префиксные суммы в CurStats. Это суммы по кликам, показам, деньгам и т.д. Впоследствии CurStats сохраняется в StatsHistory. Статистика для нового семпла TSample получается вычитанием текущей последней накопленной статистики из про-

шлой, информация о которой хранится в StatsHistory. TSample содержит оценочные признаки и счетчиковые, а также таргет CEstimate. Для его оценки используется класс TFeaturesSetup, который создает набор отдельных конфигураций для разного вида признаков. Эти конфигурации содержат окно Window, за которое нужно посчитать признак, прайорные значения, в случае, если статистики не будет и сумма будет нулевой.

### **3.4.3. Компонента Prepare samples**

На последней стадии происходит джойн тех логов, что не были добавлены ранее, последующая фильтрация семплов, а также добавление продовой оценки  $C$ , для того, чтобы сравнить ее качество с прогнозом новой модели.

## 4. Обучение моделей

В данном разделе обосновывается выбор функции потерь, приемочных метрик, а также основные результаты обучения моделей.

### 4.1. Выбор функции потерь

Перед обзором функций потерь отметим некоторые свойства нашего датасета:

- На практике, таргет, на который обучается модель, в каждом часу оказывается довольно шумный. Однако, если рассматривать кампании в масштабе недели, то дисперсии остается уже не так много.
- Наша цель научиться предсказывать параметр  $C$  таким образом, чтобы до конца недели его не нужно было значительно корректировать. То есть нам необходимо оценивать матожидание расхода кампании, чтобы корректно прогнозировать темп откруток и попадать в заданный рекламодателем бюджет.
- Датасет представлен рекламными кампаниями разных размеров: для крупных клиентов характерна более высокая интенсивность расходов по сравнению с небольшими. Из чего следует большой разброс значений  $C$ . Необходимо учитывать значительную разницу в абсолютных значениях таргета при выборе функции потерь, чтобы крупные кампании не забирали всю ошибку на себя.

Так как предполагается использовать CatBoost в качестве модели машинного обучения, рассмотрим некоторые функции потерь, которые поддерживаются в Catboost:

#### 1. Арифметические:

(a) **Средняя квадратичная ошибка (MSE):**

$$\text{MSE} = \frac{\sum_{i=1}^N w_i \cdot (t_i - p_i)^2}{\sum_{i=1}^N w_i}$$

(b) **Корень из средней квадратичной ошибки (RMSE):**

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^N w_i \cdot (t_i - p_i)^2}{\sum_{i=1}^N w_i}}$$

Данные функции потерь могут плохо решать поставленную задачу, так как нам нужно штрафовать модель за ошибки одинаково ”в X раз”, а не за ошибки ”на X”. Несмотря на то, что у рекламных кампаний разный масштаб, они все одинаково важны.

2. **MAPE:**

$$\text{MAPE} = \frac{\sum_{i=1}^N w_i \frac{|a_i - t_i|}{\max(1, |t_i|)}}{\sum_{i=1}^N w_i}$$

Рассмотрим пример, когда у нас есть кампания, правильный прогноз которой 0.1, а в датасете 9 раз встречается  $t=0$  и один раз  $t=1$ . В этом случае оптимальный прогноз будет равен нулю, что не интуитивно.

3. **Log-loss:**

$$\text{Log-loss} = \frac{\sum_{i=1}^N -w_i \cdot (t_i \log(p_i) + (1 - t_i) \log(1 - p_i))}{\sum_{i=1}^N w_i}$$

Метрика подходит для задачи, когда таргет находится в пределах  $[0, 1]$ . В нашем случае это не так. Однако мы можем этого достичь,

разделив его на максимальную величину, предварительно убрав потенциальные выбросы. Для этого предлагается убрать данные больше 95-го перцентиля.

#### 4. RMSLE:

$$\text{RMSLE} = \sqrt{\frac{1}{\sum_{i=1}^n w_i} \cdot \sum_{i=1}^n w_i \cdot (\log(p_i + 1) - \log(t_i + 1))^2}$$

Такая метрика также подходит, ее преимущество в том, что таргет не нужно нормализовывать.

CatBoost основан на методе градиентного бустинга над деревьями решений. В листьях деревьев, построенных с помощью CatBoost, для предсказания используется значение, основанное на сумме корректировок предыдущих моделей (деревьев), с добавлением вычисленного значения в текущем листе. Это значение может представлять собой среднее значение целевой переменной для регрессии или вероятности классов для задач классификации. В идеале хочется, чтобы в каждый лист попадали похожие рекламные кампании, что подразумевает схожесть их таргетов. Усреднение значений в листе будет хорошим свойством, что даст более устойчивое и корректное решение для каждого отдельного примера. В случае log-loss такое достигается, чего не скажешь про арифметические метрики и rmsle.

## 4.2. Разделение таргета

В прошлом разделе было показано, что log-loss может быть хорошей функцией потерь. Однако, использование его в текущем виде имеет ряд проблем:

1. Слагаемое  $-p_i \log x_i$  в данной функции потерь все еще может быть чувствительно к масштабу в случае отсутствия нормировки.
2. Предварительная нормализация таргета на большую константу,

для приведения его к промежутку  $[0, 1]$  может приводить к необходимости выучивания катбустом более сложных зависимостей.

## 1. Стандартный алгоритм обучения

Рассмотрим один обучающий семпл, его таргет обозначим как  $p$ . Для того, чтобы использовать log-loss можно разделить этот семпл на два с соответствующими весами. Получим точно такую же оптимизационную задачу:

$$\begin{cases} t = 1, & w = p \\ t = 0, & w = 1 - p \end{cases}$$

где  $t$  это бинарный таргет, а  $w$  вес этого семпла.

$$\text{LogLoss} = \frac{-\sum_{i=1}^n (p_i \log(x_i)) - \sum_{i=1}^n (1 - p_i) \log(1 - x_i)}{\sum_{i=1}^n p_i + \sum_{i=1}^n (1 - p_i)}$$

Найдем оптимальный константный прогноз.

$$\frac{-n \log(c) \sum_{i=1}^n p_i - n \log(1 - c) \sum_{i=1}^n (1 - p_i)}{\sum_{i=1}^n p_i + \sum_{i=1}^n (1 - p_i)}$$

Пусть  $a = \sum_{i=1}^n p_i$  и  $b = \sum_{i=1}^n (1 - p_i)$

Тогда

$$\frac{-n(\log(c)a + \log(1 - c)b)}{a + b} \rightarrow \min$$

Оптимум будет:

$$c = \frac{a}{a + b} = \frac{a}{n}$$

Запишем формулу перевода логита в искомую величину от  $[0, 1]$ :

$$x = \text{logit} \left( \frac{a}{n} \right) = \log \left( \frac{a}{n} \right) - \log \left( 1 - \frac{a}{n} \right) = \log \left( \frac{a}{n - a} \right)$$

Итоговое преобразование:  $\text{sigmoid}(x) = \frac{a}{n}$

В данном случае знаменатель в формуле  $\text{LogLoss}$  константный, а в числителе присутствует упомянутая проблема с масштабом.

## 2. Преобразование весов

Покажем другой способ обучения, где оптимальный константный прогноз остается тем же, однако меняется оптимизационная задача.

$$\begin{cases} t = 1, & w = p \\ t = 0, & w = 1 \end{cases}$$

аналогично,  $t$  это бинарный таргет, а  $w$  вес этого семла.

$$\frac{-\sum_{i=1}^n (p_i \log x_i) - \sum_{i=1}^n \log(1 - x_i)}{\sum_{i=1}^n p_i + n}$$

Найдем оптимальный константный прогноз.

$$\frac{-n \log(c) \sum_{i=1}^n p_i - n \log(1 - c)}{\sum_{i=1}^n p_i + n}$$

$$-\frac{n(\log(c)a + \log(1 - c)n)}{a + n}$$

$$-(\log(c)a + \log(1 - c)n) \rightarrow \min$$

Оптимум будет:

$$c = \frac{a}{a + n}$$

Запишем формулу перевода логита в искомую величину от  $[0, 1]$ :

$$x = \text{logit} \left( \frac{a}{a + n} \right) = \log(a) - \log \left( 1 - \frac{a}{a + n} \right)$$

Тогда

$$x = \log \left( \frac{a}{n} \right)$$

Итоговое преобразование:  $\exp(x) = \frac{a}{n}$

Таким образом, при модификации весов в знаменателе появилась сумма, которая будет нормировать ошибку. Кроме того, в случае экспоненты, чтобы  $p$  уменьшилось в  $N$  раз, нужно уменьшить значение ответа катбуста на  $\log(N)$ , а это проще когда мы поделим исходный



таргет на большое число  $N$ . Однако, когда  $p$  это сигмоида от ответа катбуста, то для уменьшения значения  $p$  на каждом семпле в  $N$  раз, катбусту нужно будет сделать нетривиальные преобразования с исходным своим прогнозом. Поэтому второй вариант обучения предпочтительнее.

### 4.3. Обучение модели

Так как модель должна уметь пресказывать правильную интенсивность трат, то эти предсказания должны соответствовать деньгам, которые кампания потратила за промежуток времени. Таким образом, в качестве весов модели можно использовать множитель из формулы ожидаемых денег:  $EventCost(t) = C \cdot Bid(t)^P \cdot T(t)$ , то есть вес каждого семпла при обучении будет  $Bid(t)^P \cdot T(t)$ , где данные параметры считаются в редьюсере по заданным окнам.

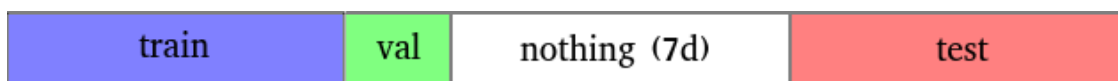


Рис. 5: Схема обучения модели

На рисунке 5 схематично обозначено расположение датасетов во времени. Выборка была разбита на 3 части: тренировочную (train), валидационную (val) и тестовую (test). Поскольку как таргет, так и некоторые признаки рассчитываются за некоторый период, то они могут не очень сильно меняться со временем. В связи с этим, было бы не совсем корректно тестировать модель на данных, которые близки по времени к тем, на которых она училась. Для этого тестовая выборка идет на 7 дней позже валидационной. Такой промежуток был выбран, так как признаки содержат статистику не больше чем за 7 дней последних дней.

### 4.4. Метрики качества и замеры

В прошлых разделах рассматривались недостатки функций ошибок: они чувствительны к масштабу. Похожие рассуждения можно прове-

сти относительно  $r^2$ ,  $SMAPE$ . Поэтому в качестве метрики качества решено использовать метрику  $LLp$ :

$$best\_const = \frac{\sum_{i=0}^N w_i t_i}{\sum_{i=0}^N w_i}$$

$$LL\_best\_const = \frac{\sum_{i=0}^N -w_i(t_i \log(best\_const) + (1-t_i) \log(1-best\_const))}{\sum_{i=0}^N w_i}$$

$$LLp = \frac{LL\_best\_const - LL}{best\_const}$$

Она имеет следующие преимущества:

1. Возможность понять лучше ли модель константного прогноза
2. Возможность сравнивать модели между собой
3. Нет зависимости от масштаба

Сравнение по метрике  $llp$  делалось с аналитической моделью, где в качестве таргетов использовались исходная  $C$  и  $EventCost$ . Таким образом, проверяется что мы не только хорошо предсказываем интенсивность трат, но и умеем оценивать деньги, которые с помощью этой интенсивности будут получены.

В лучшей итерации удалось получить  $+5\%$  по метрике  $llp$  для датасета  $C$  и  $+6\%$  по  $llp$  для датасета по  $Cost$ .

Таким образом, общий пайплайн сбора, обучения и замеров можно изобразить в виде схемы 6:

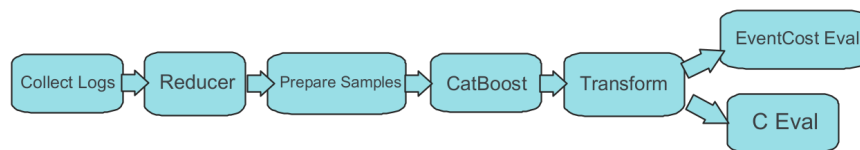


Рис. 6: Общий пайплайн обучения

Данная схема поддержана в виде конфига, написанного на языке YAML. Конфиг описывает набор параметров для обучения, стадии по обработке, обучения и замеров. Конфиг собирается в граф, наподобие MLFlow и позволяет визуальнo отследить все процессы.

## 5. Апробация модели

В данном разделе описывается принцип работы экспериментов автобюджета, приемочные метрики в онлайн, а также выводы из проведенного эксперимента с новой моделью.

### 5.1. Эксперименты в автобюджете

Для оценки качества модели в онлайн был запущен эксперимент на 10-процентном траффике. Принцип работы экспериментов следующий:

1. Пользователь заходит на сайт, начинается процесс отбора рекламы. По идентификатору пользователя хеш функцией определяется номер выборки (`testid`).
2. Исходя из того, в какую выборку попал пользователь, для него участвует определенный набор кампаний в аукционе (эти рекламные кампании прошли фильтрации).
3. Рассмотрим этот набор рекламодателей. Для них автобюджет предсказывает параметр  $C$  и выставляет ставки. То есть существует массив для каждого заказа:  $[(testid, \text{ставка}), \dots]$ . Ставка в разных `testid` разная.

Таким образом, получаем изолированные экспериментальные выборки. Каждая из них выдерживает один и тот же набор ограничений, которые задал рекламодатель.

Для проведения эксперимента использовалась одна полная календарная неделя. Так как рекламодатели имеют возможность задать недельные бюджеты и часы показов в течение недели, замеры на таком промежутке должны показать реакцию модели во всех возможных условиях.

### 5.2. Онлайн метрики и замеры

В качестве метрики приемки используется:

$$\text{OrdCost}(\text{testid}) = \sum_{\text{OrderID}} \text{Actions}(\text{OrderID}, \text{testid}) \cdot \text{AvgCost}(\text{OrderID}),$$

где  $\text{AvgCost}(\text{OrderID}) = \frac{\sum_{\text{click} \in \text{last\_month}} \text{Cost}(\text{OrderID}, \text{click})}{\sum_{\text{click} \in \text{last\_month}} \text{Actions}(\text{OrderID}, \text{click})}$

Это оценка ожидаемых будущих денег в эксперименте через количество конверсий. То есть взвешиваем конверсии средним CPA (Cost Per Actions). Метрика учитывает полезность внедрения для рекламодателя и компании: в случае когда цена на трафик выросла, а конверсий больше не стало, метрика в сравнении с контролем не будет меняться. Хотя компания зарабатывает, рекламодателю нет пользы от того, что теперь он платит больше за те же самые конверсии. Если же конверсии стали дороже и их стало больше, то метрика уже будет положительной. Однако, в ней есть и недостаток. Можно получить положительные значения, занижая при этом CPA и просадить деньги в момент выкатки новой модели в продакшн. Здесь можно сделать предположение о том, что улучшение траффика для рекламодателя побудит его повысить недельные бюджеты, так как теперь конверсии для него стали дешевле. Однако, такое внедрение скорее рискованное.

Относительные метрики считаются следующим образом:

$\text{OrdCost}_r = \frac{\text{OrdCost}_{\text{exp}}}{\text{OrdCost}_{\text{control}}} \times 100$  - описанная ранее оценка ожидаемых будущих денег.

$\text{Cost}_r = \frac{\text{Cost}_{\text{exp}}}{\text{Cost}_{\text{control}}} \times 100$  - отношение заработанных денег в эксперименте к контролю.

В таб. 1 приведены полученные значения. Стандартное отклонение считалось бутстрепом: на каждой итерации, каждой строчке лога, содержащей клик назначался случайный вес. Далее производилась агрегация логов с соответствующим весом. Для набора полученных значений бралось стандартное отклонение.

$\text{Cost}_r$	$\text{OrdCost}_r$
$99.63 \pm 0.41$	$101.78 \pm 2.06$

Таблица 1: Замеры по основным метрикам

Из замеров видно, что данная модель скорее полезна для рекламодателей, так как при улучшенном качестве траффика получаем серые деньги.

Приведем также несколько графиков, показывающих предсказания новой модели в сравнении со старой аналитической формулой.

На рис. 7 изображены три графика коэффициента  $C$  за две недели с грануляцией в час для одной из крупных кампаний: *prod* - текущая продовая формула, *ml* - предсказания модели, *target* - оценка интенсивности трат, посчитанная за день. Из графика видно, что модель ближе к среднему значению таргета, при этом более ровная и стационарная.

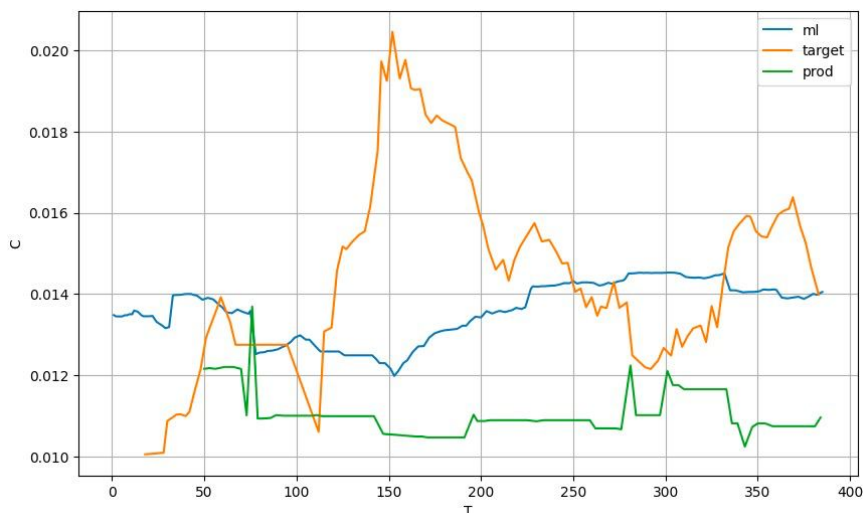


Рис. 7: Предсказание модели на одной из крупных кампаний

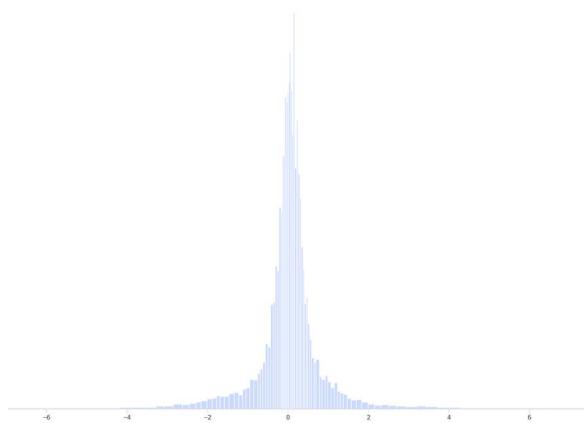


Рис. 8: Среднее отношение для кампаний с большим количеством кликов

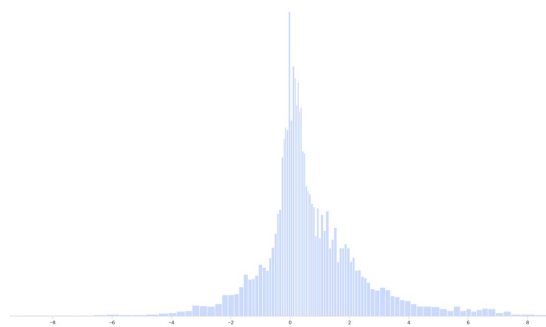


Рис. 9: Среднее отношение для кампаний с небольшим количеством кликов

На рис. 8 и 9 изображены гистограммы средних отношений аналитического расчета к предсказанию модели для кампаний с большим

количеством кликов ( $> 30$ ) и небольшим ( $\leq 30$ ) соответственно. Видно, что для высокоинтенсивных кампаний прогноз находится вокруг нуля, а с малым количеством кликов наоборот - смещен вправо, что логично, так как на них, возможно, ставка недостаточно высока и новая модель, занижая параметр  $C$  ее растит.

## 6. Результаты

В ходе работы получены следующие результаты:

1. Выполнен обзор предметной области, описан процесс заведения рекламной кампании. Также рассмотрена работа автостратегий и способы прогнозирования ставок на примере Яндекс Директа. Описаны преимущества автостратегий над стратегиями с ручным управлением ставками.
2. Спроектирован и реализован пайплайн по сбору и обработке данных для обучения модели. Реализация выполнена в виде отдельных программных модулей, которые позволяют сохранять промежуточные результаты, а также легко добавлять новую функциональность за счет разделения на отдельные компоненты. Также подготовлен конфиг для запуска пайплайна с заданными аргументами, что позволит воспроизводить результаты через время и переобучать модель «по кнопке».
3. Подобраны метрики качества, а также разработана схема обучения модели машинного обучения. В результате обучения Catboost были получены положительные результаты по двум основным метрикам  $\text{Lpr}$ : +6% на костовом датасете и +5% по  $C$ .
4. Выполнена апробация модели посредством ставочного эксперимента. В ходе аналитики удалось получить ожидаемый эффект от внедрения: более стационарная оценка коэффициента  $C$ , улучшение работы на холодном старте, а также повышение качества трафика для рекламодателей.

Таким образом, выполнена адаптация текущей аналитической формулы вычисления коэффициентов в формуле ставки на модель машинного обучения. Модель успешно внедрена и приносит ожидаемую прибыль.

## Список литературы

- [1] Li Wei, Cui Ying Grace, Zhang Ruofei, and Mao Jianchang. Bid landscape forecasting in online advertising. — 2012. — Apr. 5. — US Patent App. 12/895,773.
- [2] Amin Kareem, Kearns Michael, Key Peter, and Schwaighofer Anton. Budget optimization for sponsored search: Censored learning in mdps // arXiv preprint arXiv:1210.4847. — 2012.
- [3] Catboost documentaion. — <https://catboost.ai/>. — Accessed: 2024-03-01.
- [4] Kalman Rudolph Emil. A New Approach to Linear Filtering and Prediction Problems // Journal of Basic Engineering. — 1960. — Vol. 82, no. 1. — P. 35–45.
- [5] Cleveland Robert B, Cleveland William S, McRae Jean E, and Terpenning Irma. STL: A Seasonal-Trend Decomposition Procedure Based on Loess // Journal of Official Statistics. — 1990. — Vol. 6, no. 1. — P. 3–73.
- [6] Box George EP, Jenkins Gwilym M, Reinsel Gregory C, and Ljung Greta M. Time Series Analysis: Forecasting and Control. — John Wiley & Sons, 2015.
- [7] Winters Peter R. Forecasting Sales by Exponentially Weighted Moving Averages // [Management Science](#). — 1960. — Vol. 6, no. 3. — P. 324–342.
- [8] Yandex Direct: Advertising Service. — <https://direct.yandex.ru/>. — Accessed: 2024-03-01.
- [9] Berg Jordan, Greenwald Amy, Naroditskiy Victor, and Sodomka Eric. A first approach to autonomous bidding in ad auctions // Workshop on Trading Agent Design and Analysis at the 11th ACM Conference on Electronics Commerce. — 2010.