Теория uplift-моделирования

Введение

Uplift-моделирование — это подход в машинном обучении, который позволяет оценить **прирост эффекта** от определенного воздействия (например, рекламы или маркетинговой кампании). В отличие от классических моделей классификации, uplift-модель предсказывает не вероятность целевого события, а разницу между вероятностями события в двух группах:

- **Treatment-группа** — получившие воздействие (например, видели рекламу).
- **Control-группа** — не получившие воздействие (не видели рекламу).

Основные определения

Response Rate (Отклик) **Response rate** — это доля клиентов, которые выполнили целевое действие (например, покупку) в каждой группе. Для $k\%$ топовых клиентов response rate считается как среднее значение целевого признака Y:

$$\text{response\_rate@k} = \frac{1}{N_k} \sum_{i=1}^{N_k} Y_i, \tag{1}$$

где:

- N_k — количество клиентов в топ $k\%$ выборки;
- Y_i — целевой признак для клиента i (например, $Y = 1$, если клиент совершил покупку, и $Y = 0$, если нет).

Uplift Uplift на уровне $k\%$ определяется как разница между откликом в treatment-группе и откликом в control-группе:

$$\text{uplift@k} = \text{response\_rate@k}_{(\text{treatment})} - \text{response\_rate@k}_{(\text{control})}. \tag{2}$$

Итоговое определение Uplift-модель предсказывает разницу вероятностей между группами:

$$U(X) = P(Y = 1|X, T = 1) - P(Y = 1|X, T = 0), \tag{3}$$

где:

- $U(X)$ — uplift для клиента с признаками X;
- $P(Y = 1|X, T = 1)$ — вероятность отклика для treatment-группы;
- $P(Y = 1|X, T = 0)$ — вероятность отклика для control-группы.

Пример расчета uplift

Рассмотрим следующий пример:

- В **treatment-группе** (100 человек) целевое действие выполнили 30 человек: $\text{response\_rate}_{\text{treatment}} = 30/100 = 0.3\,(30\%)$.
- В **control-группе** (100 человек) целевое действие выполнили 20 человек: $\text{response\_rate}_{\text{control}} = 20/100 = 0.2\,(20\%)$.

Тогда uplift будет равен:

$$\text{uplift} = 0.3 - 0.2 = 0.1\,(10\%). \tag{4}$$

Это означает, что воздействие увеличило вероятность выполнения целевого действия на 10%.

Кривая Qini и метрика Qini Score

Кривая Qini показывает накопленный uplift по топовым $k\%$ клиентов. Qini Score определяется как площадь под кривой Qini:

$$\text{Qini Score} = \int_0^1 \text{uplift}(k)\,dk. \tag{5}$$

—

Вывод

Uplift-моделирование позволяет оптимизировать воздействие на клиентов, сфокусировав ресурсы на тех, кто с наибольшей вероятностью отреагирует. Методы моделирования включают двухмодельный подход, одиночную модель с измененным таргетом и анализ кривой Qini для оценки качества модели.

Описание датасетов для построения моделей

1. Датасет `clients`

Датасет содержит информацию о клиентах и включает следующие поля:

- **client_id** — ID клиента, внешний ключ к датасетам `purchases` и `uplift_train`.
- **first_issue_date** — дата и время, когда клиент впервые был зарегистрирован в системе или получил свою карту лояльности.
- **first_redeem_date** — дата и время, когда клиент впервые совершил покупку или использовал карту лояльности (например, для начисления или списания бонусов).

- **age** — возраст клиента.
- **gender** — пол клиента.

После проведения ряда проверок, мы заметили, что существуют очень странные клиенты, например у некоторых есть отрицательный возраст, а кто-то живет уже 1901 год (Рис.1). Скорее всего это связано с какой-то технической неполадкой. В общем и целом, доля этих клиентов маленькая, она составляет порядка 0,3% и ввиду того, что просто удалить клиентов мы не хотим мы заменили странные на наш взгляд возраста медианным значением. Также

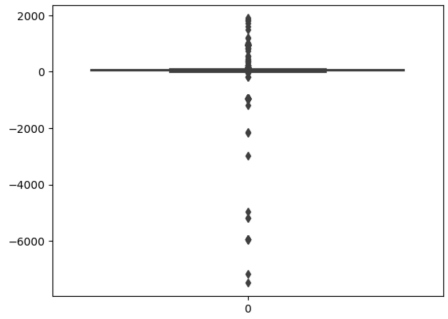

Рис. 1: Боксплот возраста.

	client_id	first_issue_date	first_redeem_date	age	gender
168731	6c43d328e8	2017-07-24	2019-02-27	-7491	U
91960	3b030e7fa6	2018-11-16	2018-12-07	-7158	U
210260	86d1dc8761	2018-10-15	2019-01-26	-5973	U
215345	8a0e463a19	2017-05-04	2017-06-13	-5953	U
177265	71c2137d2d	2018-11-25	2019-03-22	-5949	U

Рис. 2: Топ 5 с минимальным возрастом.

	client_id	first_issue_date	first_redeem_date	age	gender
141178	5a9420d3ea	2017-09-09	2018-03-08	1901	U
172772	6edee96676	2017-11-16	2018-10-18	1852	U
128323	5252823991	2018-10-30	2019-02-22	1841	U
184781	768f4e55ac	2018-10-31	2019-07-19	1800	U
29948	1350cceaf5	2018-11-17	2019-05-16	1717	U

Рис. 3: Топ 5 с максимальным возрастом.

мы заметили, что есть пропуски в дате совершения покупки 9% (35469 клиентов) видимо, не все пользователи успели совершить покупки/ или это техническая ошибка. Дропать их нельзя, так как это наши потенциальные клиенты - если предложить им скидку, может, они и начнут пользоваться нашей услугой. Парадокс заключается в том, что несмотря на то, что у клиентов стоит проупск в дате совершения покупки, они что-то приобретали. Скорее всего, пропуски связаны с каким-то техническим сбоем. Мы решили заполнить пропуски следующим образом: посмотрели сколько в среднем времени проходит между тем, когда клиент зарегестрировался в системе и совершил первую покупку/воспользовался картой лояльности, получили 180, и эту величину прибавим к `first_issue_date` пользователей, у которых `first_redeem_date` пустое. Таким образом, мы заполнили пропуски в колонке, где отражается дата первой покупки.

2. Датасет `products`

Датасет содержит информацию о товарах и их характеристиках. Поля включают:

- **product_id** — ID товара, внешний ключ к датасету `purchases`.
- **level_1** — верхний уровень категории товара (например, Продукты питания, Напитки, Косметика).
- **level_2** — подкатегория верхнего уровня (например, Молочные продукты, Алкогольные напитки, Шампуни).
- **level_3** — более детализированная категория (например, Йогурты, Пиво, Мужские шампуни).
- **level_4** — самая детализированная категория товара (например, Греческий йогурт, Светлое пиво, Шампунь с кератином).
- **segment_id** — идентификатор сегмента товара (например, премиум-класс, товары для детей, товары для дома).
- **brand_id** — идентификатор бренда товара.
- **vendor_id** — ID поставщика товара.
- **netto** — вес или объем товара.
- **is_own_trademark** — бинарный признак (0/1), указывающий, является ли товар собственным брендом компании (1 — собственный, 0 — сторонний).
- **is_alcohol** — бинарный признак (0/1), указывающий, относится ли товар к алкогольной продукции (1 — да, 0 — нет).

3. Датасет `purchases`

Датасет содержит информацию о транзакциях клиентов и включает следующие поля:

- **client_id** — ID клиента, внешний ключ к датасетам `clients` и `uplift_train`.
- **transaction_id** — ID покупки.
- **transaction_datetime** — дата и время совершения транзакции.
- **regular_points_received** — количество регулярных бонусных баллов, начисленных клиенту за эту покупку.
- **express_points_received** — количество экспресс-бонусных баллов, начисленных клиенту (например, в рамках акции).
- **regular_points_spent** — количество регулярных бонусных баллов, списанных в этой покупке.
- **express_points_spent** — количество экспресс-бонусных баллов, списанных в этой покупке.
- **purchase_sum** — итоговая сумма покупки.
- **store_id** — ID магазина, в котором была совершена покупка.
- **product_id** — ID товара, купленного в рамках покупки.
- **product_quantity** — количество единиц товара в данной покупке.
- **trn_sum_from_iss** — бонусы, начисленные на эту покупку и сразу использованные для оплаты.
- **trn_sum_from_red** — бонусы, накопленные ранее и использованные для этой транзакции.

В этом датасете возникла проблема с колонкой trn_sum_from_red. В ней пустные значения составляют порядка 93%, поэтому было принято решение ее дропнуть.

4. Датасет `uplift_train`

Датасет содержит информацию для построения uplift-модели на основе A/B теста:

- **client_id** — ID клиента, внешний ключ к датасетам `clients` и `purchases`.
- **treatment_flg** — бинарный признак, указывающий на группу клиента в A/B тесте:
  - 1 — тестовая группа (treatment).

  - 0 — контрольная группа (control).

- **target** — бинарный признак, указывающий на результат воздействия на пользователя:
  - 1 — клиент совершил покупку (целевое действие).
  - 0 — клиент не совершил покупку.

Связи между датасетами

- `clients.client_id` связывает датасет `clients` с `purchases` и `uplift_train`.
- `products.product_id` связывает датасет `products` с `purchases`.
- `uplift_train.client_id` связывает `uplift_train` с `clients` и `purchases`.

Feature Engineering

В общем и целом, обучаться наша модель будет на данных 400162 клиентов и для каждого клиента мы разработали фичи. Пройдемся по порядку.

1. Информация о клиенте

Эти признаки описывают основные характеристики клиентов, такие как возраст, пол и даты их первой активности:

- **age** — возраст клиента. Используется для анализа демографического профиля и выявления возрастных групп с разной активностью.
- **gender** — пол клиента. Помогает понять различия в поведении между мужчинами и женщинами.
- **first_issue_year**, **first_issue_month**, **first_issue_day** — дата регистрации клиента. Используется для анализа влияния времени регистрации на покупательскую активность.
- **first_redeem_year**, **first_redeem_month**, **first_redeem_day** — дата первой покупки. Помогает понять, сколько времени клиенту потребовалось для первой транзакции.
- **issue_redeem_days_diff** — разница в днях между регистрацией и первой покупкой. Указывает на скорость вовлечения клиента.

2. Финансовая активность

Эти признаки описывают финансовую активность клиента:

- **total_purchase_sum** — общая сумма всех покупок клиента. Оценивает общий вклад клиента.
- **total_purchase_count** — общее количество покупок клиента. Анализирует активность клиента.
- **avg_purchase_sum** — средняя стоимость одной покупки. Позволяет понять средний размер корзины клиента.
- **max_purchase_sum**, **min_purchase_sum**, **std_purchase_sum** — максимальная, минимальная сумма покупки и стандартное отклонение. Оценивают вариативность покупок клиента.
- **purchase_frequency_weekly**, **purchase_frequency_monthly** — частота покупок клиента в неделю и месяц. Показывает регулярность покупок.
- **avg_days_between_purchases**, **max_days_between_purchases**, **min_days_between_purchases** — среднее, максимальное и минимальное количество дней между покупками. Оценивают лояльность и вовлечённость клиента.
- **time_span_days** — количество дней между первой и последней транзакцией. Анализирует активность клиента за весь период.

3. Активность по дням недели

Признаки, отражающие активность клиента в зависимости от дня недели:

- **purchases_weekday_0**, ..., **purchases_weekday_6** — количество покупок, совершённых в каждый день недели (понедельник, вторник, ...).
- **purchase_sum_weekday_0**, ..., **purchase_sum_weekday_6** — сумма покупок, совершённых в каждый день недели.
- **weekday_ratio_0**, ..., **weekday_ratio_6** — доля покупок, совершённых в определённый день недели от общего количества покупок.
- **favorite_weekday** — день недели, когда чаще всего совершаются покупки.
- **favorite_weekday_sum** — день недели с максимальной суммой покупок.

4. Активность по месяцам

Признаки, отражающие активность клиента в зависимости от месяца:

- **purchases_month_1**, ..., **purchases_month_12** — количество покупок, совершённых в каждом месяце.
- **purchase_sum_month_1**, ..., **purchase_sum_month_12** — сумма покупок, совершённых в каждом месяце.
- **month_ratio_1**, ..., **month_ratio_12** — доля покупок, совершённых в каждом месяце от общего количества покупок.
- **favorite_month** — месяц, когда чаще всего совершаются покупки.
- **favorite_month_sum** — месяц с максимальной суммой покупок.

5. Продукты и бренды

Признаки, связанные с продуктами и брендами:

- **unique_products_purchased** — количество уникальных продуктов, купленных клиентом.
- **favorite_product_id** — ID продукта, который чаще всего покупался.
- **favorite_brand_id** — ID бренда, чьи продукты чаще всего покупались.
- **favorite_segment_id** — ID сегмента, чьи продукты чаще всего покупались.
- **product_diversity** — доля уникальных продуктов в общем количестве покупок.
- **avg_product_quantity** — среднее количество единиц товара в одной покупке.
- **max_product_quantity** — максимальное количество единиц товара в одной покупке.
- **avg_product_netto** — средняя масса/объем товаров в покупках.

6. Специфические покупки

Признаки, связанные с определёнными категориями продуктов:

- **alcohol_purchase_sum**, **alcohol_purchase_count** — общая сумма и количество покупок алкогольной продукции.
- **own_trademark_sum**, **own_trademark_count** — общая сумма и количество покупок продуктов собственной марки.

- **alcohol_ratio** — доля покупок алкогольной продукции в общем количестве покупок.
- **own_trademark_ratio** — доля покупок продуктов собственной марки в общем количестве покупок.

7. Магазины и суммы выше порогов

Признаки, связанные с магазинами и крупными покупками:

- **favorite_store_id** — ID магазина, где чаще всего совершались покупки.
- **purchases_above_threshold_500**, **purchases_above_threshold_750**, **purchases_above_threshold_1000** — количество покупок, сумма которых превышает 500, 750 и 1000 соответственно.
- **sum_above_threshold_500**, **sum_above_threshold_750**, **sum_above_threshold_1000** — сумма покупок, превышающая 500, 750 и 1000 соответственно. По желаню сумму можно изменить в коде.

8. Динамика покупок

Признаки, описывающие динамику покупок клиента:

- **purchase_growth_rate** — темп роста покупок клиента за последние 3 месяца.

Uplift модели

1) Treatment Dummy approach

Самое простое и интуитивное решение заключается в том, что модель обучается одновременно на двух группах. При этом бинарный флаг коммуникации выступает в качестве дополнительного признака. Каждый объект из тестовой выборки скорим дважды: с флагом коммуникации, равным 1, и с флагом, равным 0. Вычитая вероятности по каждому наблюдению, получим искомый uplift.

Для решения любых задач классификации и регрессии мы использовали библиотеку CatBoost.

Была построена модель CatBoostClassifier, которая предсказывала целевое действие (target). В виду того, что перед нами не стояла цель получить наилучший результат с точки зрения подгонки под тестовые данные, то не использовались методы подбора гиперпараметров через

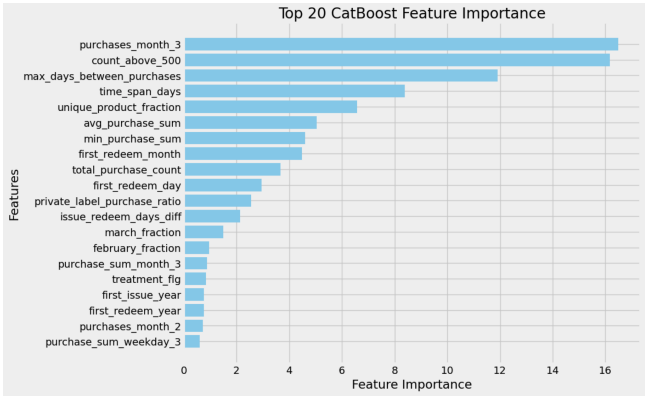

Рис. 4: CatBoost top 20 features

GridSearch или Optuna, а параметры подбирались используя встроенные в CatBoost графики LogLoss и AUC на трейне и валидации, именно по ним оценивалась модель (во избежании Overfitting).

uplift рассчитывался как разность уверенности классификатора в выставленном классе для оценок, полученных при использованиии treatment flg = 1 (для всех юзеров) и treatment flg = 0 (для всех юзеров)

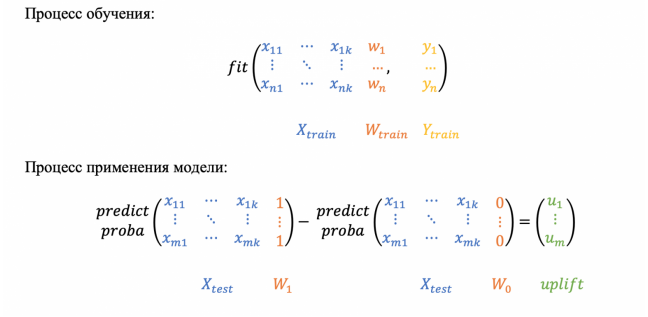

Рис. 5: Treatment Dummy approach

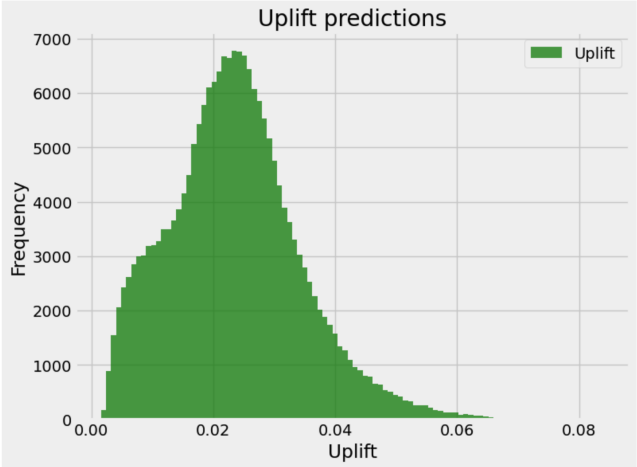

Рис. 6: Treatment Dummy approach results

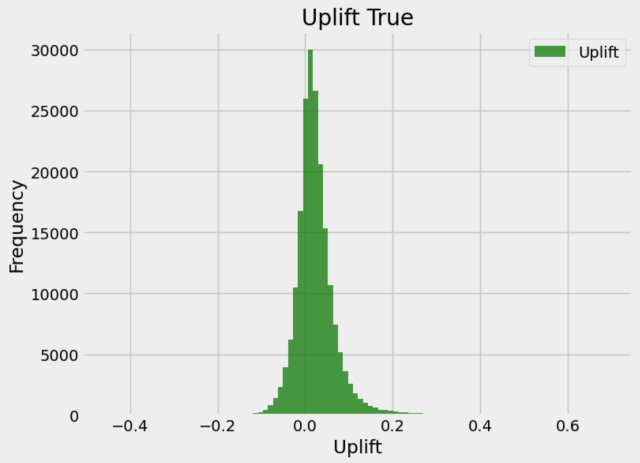

Рис. 7: True Results (Uplift)

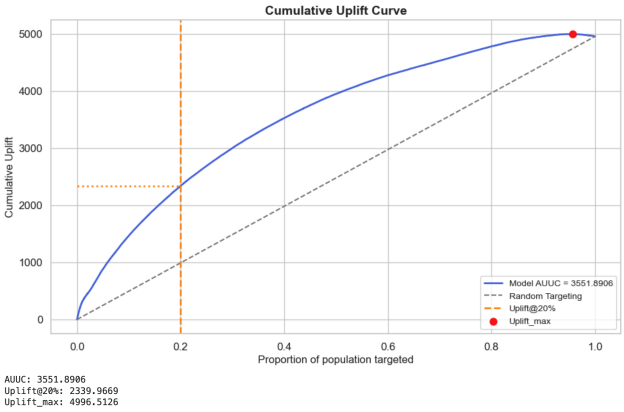

Рис. 8: AUUC Curve

- **Ось X – Proportion of population targeted** Доля целевой популяции, на которую направлено воздействие (от 0 до 100%).
- **Ось Y – Cumulative Uplift** Накопленный uplift – суммарный прирост эффекта по мере привлечения большего количества клиентов из топовой части предсказаний модели.
- **Синяя линия – Модельный uplift** Это накопленный uplift, который предсказывает модель. Чем выше кривая, тем эффективнее модель при фокусировании на наиболее перспективных клиентах.
- **Серая пунктирная линия – Random Targeting** Линия показывает результат случайного таргетирования и служит базовым ориентиром.
- **Оранжевая пунктирная линия – Uplift@20%** Отмечает значение uplift при таргетировании на **топ 20%** клиентов.
- **Красная точка – Uplift max** Красная точка на графике показывает максимальный накопленный uplift, достигнутый моделью.
- **AUUC – Area Under the Uplift Curve** Площадь под кривой накопленного uplift. Чем больше эта площадь, тем лучше модель. Она показывает, насколько эффективно модель приоритизирует клиентов с максимальным потенциалом прироста.

2) Two Models approach

Подход с двумя моделями один из самых популярных и достаточно часто встречается в статьях. Метод заключается в отдельном моделировании двух условных вероятностей на целевой и контрольной группах, а именно:

1. Строится первая модель, оценивающая вероятность выполнения целевого действия среди клиентов, с которыми мы взаимодействовали.
2. Строится вторая модель, оценивающая ту же вероятность, но среди клиентов, с которыми мы не производили коммуникацию.
3. Затем для каждого клиента рассчитывается разность оценок вероятностей двух моделей.

В нашем случае используется **CatBoost** для построения двух отдельных моделей. Одна модель обучается на группе с treatment_flg = 1, а другая на группе с treatment_flg = 0. Флаг коммуникации не используется как объясняющая переменная.

Для каждого пользователя из выборочной группы делаются два предсказания:

$$\text{predict\_proba}_{\text{treatment}} = P(Y = 1 \mid X, \text{treatment\_flg} = 1),$$
$$\text{predict\_proba}_{\text{control}} = P(Y = 1 \mid X, \text{treatment\_flg} = 0).$$

Затем рассчитывается uplift для каждого пользователя как разность вероятностей:

$$\text{uplift}(X) = \text{predict\_proba}_{\text{treatment}} - \text{predict\_proba}_{\text{control}}.$$

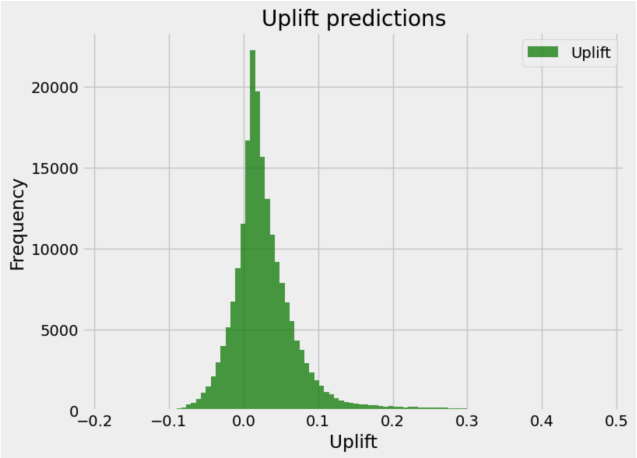

Рис. 9: Two Models approach

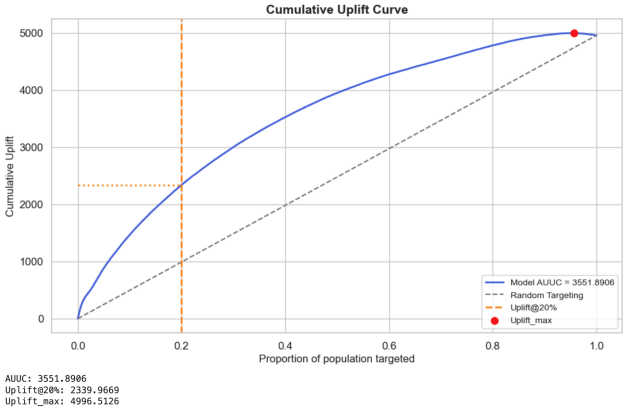

Рис. 10: Two Models approach AUUC

Как можно заметить на кривой Model AUUC появилась точка максимума - почему же так? Да все просто: модель прогнозирует пользователей, на которых лучше не воздействовать, так как они откажутся от совершения целевого действия (в прошой модели все uplift значения были > 0, поэтому было выгодно рассылать смс всем пользователям).

3) Class Transformation approach (Classification)

Достаточно интересный и математически подтверждённый подход к построению модели, который заключается в прогнозировании немного изменённой целевой переменной:

$$Z_i = Y_i \cdot W_i + (1 - Y_i) \cdot (1 - W_i)$$

где:

- Z_i — новая целевая переменная i-го клиента,
- Y_i — целевая переменная i-го клиента,
- $W_i = \{0, 1\}$ — бинарный флаг коммуникации:
  - при $W_i = 1$ — i-й клиент попал в целевую (*treatment*) группу, где была коммуникация;
  - при $W_i = 0$ — i-й клиент попал в контрольную (*control*) группу, где не было коммуникации.

Другими словами, новый класс равен 1, если мы знаем, что на конкретном наблюдении результат при взаимодействии был бы таким же или лучше, как и в контрольной группе, если бы мы могли знать результат в обеих группах:

$$Z_i = \begin{cases} 1, & \text{если } W_i = 1 \text{ и } Y_i = 1, \\ 1, & \text{если } W_i = 0 \text{ и } Y_i = 0, \\ 0, & \text{в остальных случаях.} \end{cases}$$

Таким образом, увеличив вдвое прогноз нового таргета и вычтя из него единицу, мы получим значение самого uplift, т.е.

$$\textit{uplift} = 2 \cdot P(Z = 1) - 1$$

Исходя из допущения, описанного выше:

$$P(W = 1) = P(W = 0) = \frac{1}{2},$$

данный подход следует использовать только в случаях, когда количество клиентов, с которыми мы прокоммуницировали, равно количеству клиентов, с которыми коммуникации не было. В нашей выборке группы поделены поровну, поэтому мы можем воспользоваться этим допущением.

По итогу мы снова обучаем CatBoost классификатор, делаем предикты, получаем predict proba и рассчитываем uplift.

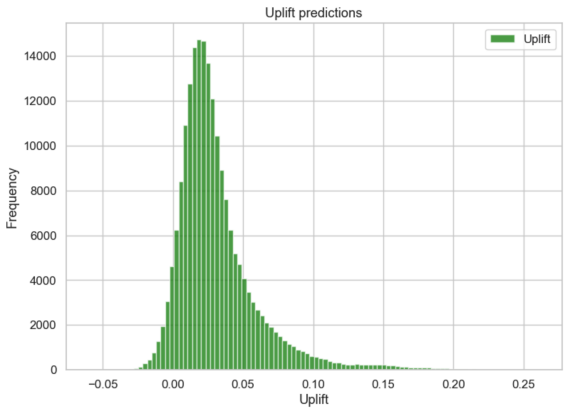

Рис. 11: Class Transformation approach (Classification)

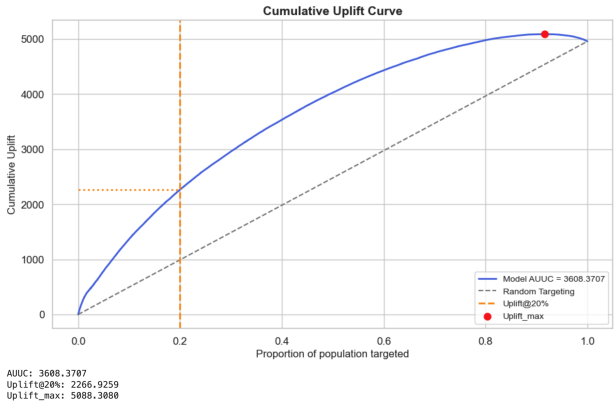

Рис. 12: Class Transformation approach (Classification) AUUC

4) Class Transformation approach (Regression)

На предыдущий тип трансформации классов накладываются серьёзные ограничения: целевая переменная Y_i может быть только бинарной, а контрольная и целевая группы должны быть распределены в равных пропорциях. Давайте рассмотрим более общий подход из, не имеющий таких ограничений.

Трансформируем исходную целевую переменную Y_i по следующей формуле:

$$Z_i = Y_i \frac{W_i - p}{p(1-p)},$$

где:

- Z_i — новая целевая переменная для i-го клиента,
- W_i — флаг коммуникации для i-го клиента,
- $p*$ — propensity score или вероятность отнесения к целевой группе: $p = P(W_i = 1 \mid X_i = x)$.

Здесь важно отметить, что можно оценить p как долю объектов с $W = 1$ в выборке. Или воспользоваться способом, в котором предлагается оценить p как функцию от X, обучив классификатор на имеющихся данных $X = x$, а в качестве целевой переменной взяв вектор флага коммуникации W.

После применения формулы получаем новую целевую переменную Z_i и можем обучить модель регрессии с функционалом ошибки:

$$MSE = \frac{1}{n} \sum_{i=0}^{n} (Z_i - \hat{Z}_i)^2.$$

Так как именно при применении MSE предсказания модели являются условным математическим ожиданием целевой переменной.

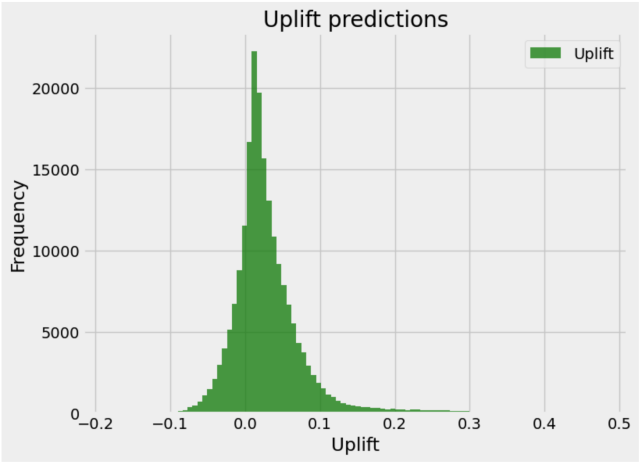

Рис. 13: Class Transformation approach (Regression)

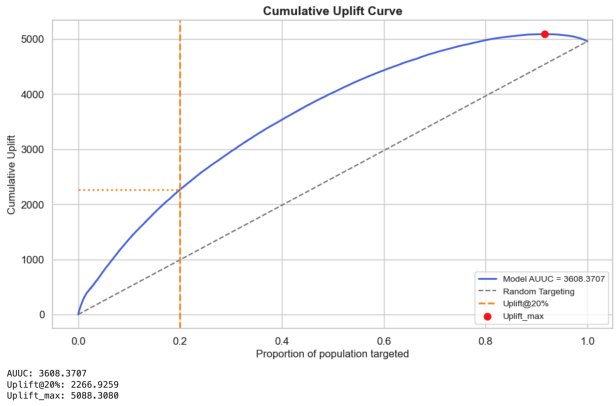

Рис. 14: Class Transformation approach (Regression) AUUC

5) Causal trees

Стоит отметить, что предыдущие методы имеют следующие недостатки:

- В методах с двумя моделями при расчёте финального предсказания учитываются результаты двух моделей, а значит их ошибки суммируются;
- Если для обучения будут использоваться принципиально разные модели или природа данных целевой и контрольной групп будет сильно отличаться, то может потребоваться калибровка предсказаний моделей;
- Так как во многих методах uplift прогнозируется косвенно, модели могут пропускать слабые различия между целевой и контрольной группами.

Хочется взять хорошо зарекомендовавший себя метод и изменить его так, чтобы непосредственно оптимизировать uplift. Например, использовать деревья решений с другим критерием разбиения. Дерево строится так, чтобы максимизировать расстояние (дивергенцию) между распределениями целевой переменной у *контрольной* и *целевой* групп. Формально для каждого разбиения это можно записать так:

$$Gain_D = D_{\text{after split}}(P^T, P^C) - D_{\text{before split}}(P^T, P^C),$$

где P^C, P^T — распределения целевой переменной в *контрольной* и *целевой* группах, D — дивергенция (расхождение) между двумя распределениями.

Есть несколько видов дивергенции D, которые используют для решения этой задачи:

- **Дивергенция Кульбака–Лейблера (Kullback-Leibler divergence):**

$$KL(P, Q) = \sum_i p_i \log \frac{p_i}{q_i}.$$

- **Евклидово расстояние (Euclidean distance):**

$$E(P, Q) = \sum_i (p_i - q_i)^2.$$

- **Дивергенция хи-квадрат (Chi-squared divergence):**

$$\chi^2(P, Q) = \sum_i \frac{(p_i - q_i)^2}{q_i}.$$

Где распределения представлены как $Q = (q_1, \ldots, q_n)$ и $P = (p_1, \ldots, p_n)$.

Если получается так, что в вершине при разбиении остаются объекты одной группы (контрольной или целевой), то дивергенция сводится к стандартному для деревьев критерию:

- KL-дивергенция — к энтропийному критерию,
- Евклидово расстояние и хи-квадрат — к критерию Джини.

Мы использовали библиотеку от Uber - Causalml А именно класс UpliftRandomForestClassifier, где по умолчанию установлена Дивергенция Кульбака–Лейблера.

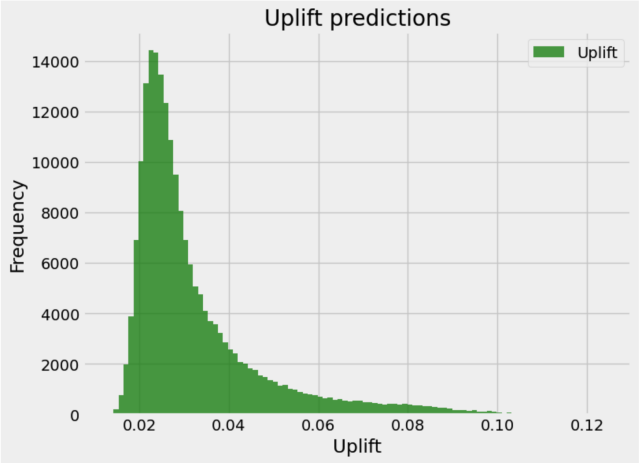

Рис. 15: Causal trees

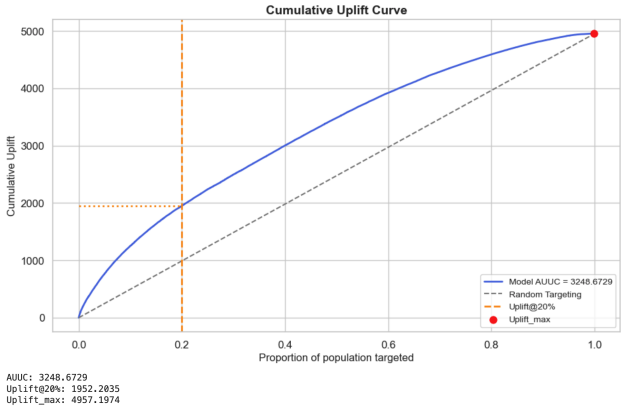

Рис. 16: Causal trees AUUC