МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное
учреждение высшего образования
«Московский авиационный институт
(национальный исследовательский университет)» (МАИ)

УТВЕРЖДАЮ
Заведующий кафедрой
«Теория вероятностей и компьютерное моделирование»
д.ф.-м.н., профессор

_____ А.И. Кибзун
«07» июня 2022 г.

ОТЧЕТ
О НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

по теме:
Исследование задачи прогнозирования оттока клиентов

Научный руководитель

к.ф.-м.н., доцент _____ Платонов Е.Н.

Исполнитель

магистр группы М8О-101М-21 _____ Фейзуллин К.М.

Москва 2022

Реферат

Отчет 24 с., 19 рис., 1 табл., 1 источн.

Объектом исследования являются задача прогнозирования оттока клиентов

Цель работы – постановка задачи и исследование методов решения.

В результате работы определены методы решения задачи бинарной классификации оттока покупателей. Дальнейшее исследование может включать в себя исследование решений задачи UpLift моделирования и сравнительное исследование решений задач.

Оглавление

Введение

В данной научно-исследовательской работе проводится исследование возможных подходов к решению задачи прогнозирования оттока клиентов с помощью методов машинного обучения.

С ростом глобализации и цифровизации появилась возможность работать с потребительскими данными, активно взаимодействовать с потребителями путем разных акций, особых предложений. Чтобы клиент не забывал о поставщике потребительских услуг, производитель может напомнить о себе посредством коммуникации.

Но стоит взять во внимание, что каждая коммуникация стоит денег. Если клиентская база составляет 1 тыс. клиентов, то прислать всем SMS стоит не дорого. Но если увеличить масштаб базы до миллиона или нескольких миллионов, то слепая коммуникация со всеми подряд станет очень дорогой. Даже если у компании большой оборот выручки, каждая такая коммуникация будет ощутимо сказываться на общем бюджете.

Поэтому коммуникацию можно использовать гораздо более оптимальным способом. Например, совершать коммуникацию с потенциально ушедшим пользователем.

Однако с ростом клиентской базы даже выборочная коммуникация с потенциально потерянными клиентами будет затратной и следующей задачей является прогнозирование, повлияет ли коммуникация на пользователя.

Основная часть отчета о НИР

В данной работе производится первичный анализ методом машинного обучения для UpLift моделирования, определяются возможные подходы к решению задачи и основные этапы работы, проводится анализ реализованных методов решения.

Определение метрик для оценки качества UpLift моделирования

Так как задача UpLift представляет собой задачу оценки (скор балл) эффекта от коммуникации на реципиента, то нет и истинных ответов. Получается, что не удастся использовать классические метрики, такие как Accuracy и PR AUC, основанные на матрице ошибок, для классификации или среднеквадратичная ошибка для задачи регрессии при трансформации классов.

UpLift на первых k – процентах выборки

Самая простая и интуитивно понятная метрика, особенно для применения в бизнесе и для интерпретации.

Допустим, что на коммуникации в компании имеется скромный бюджет, который может обеспечить связь всего с 30% клиентской базы для побуждения к целевому действию. Тогда целью UpLift моделирования будет найти такой алгоритм, который лучше всех максимизирует эффект от коммуникаций на первых 30% клиентов.

Чтобы получить значение этой метрики, нужно ранжировать результат прогноза по убыванию, чтобы отобрать клиентов, на которых коммуникация оказывает наибольший эффект. Далее берется разница между конверсией целевой группы, с которой осуществлялась коммуникация, и конверсией контрольной группы, которая осталась без коммуникации.

Формула имеет следующий вид:

$$UpLift_{K\%} = CR_{K\%}\left(X_{target}\right) - CR_{K\%}\left(X_{control}\right),$$

$$\text{где } CR_{K\%} = \frac{\text{Отклик}_{K\%}}{\text{Размер выборки}_{K\%}}.$$

Как и сам UpLift, $UpLift_{K\%}$ имеет область значений [-1, 1].

Причем, данную метрику можно рассчитать двумя способами, в зависимости от ранжирования по прогнозу UpLift:

- Сортировка происходит по прогнозу и далее берется разность рабочей и контрольной группы.
- Сортировка происходит внутри каждой группы обособленно и далее берется разность.

Второй вариант имеет более практическое применение, так для оценки эффективности от коммуникаций при рекламных кампаниях, при планировании проведения мероприятий, образуются две однородные выборки – рабочая и тестовая группа.

Для дальнейшего исследования будем оценивать метрику при k = 30%.

UpLift по процентилям

Данная метрика представляется в виде таблицы или графика для общего понимания качества работы модели при разных долях выборки K%, где для каждого K% определяется $CR_{K\%}\left(X_{target}\right)$и $CR_{K\%}\left(X_{control}\right)$. Пример отображения на рисунке 1.

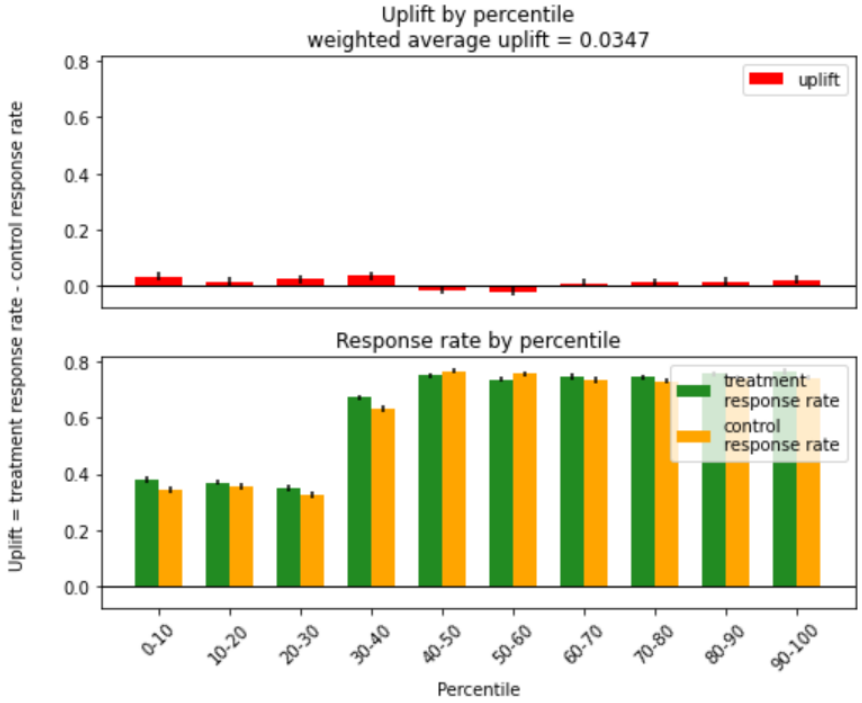

Рисунок 1

Алгоритм расчета схож с предыдущей метрикой:

1. Выборка сортируется по прогнозу UpLift.
2. Отсортированные данные делятся на интервалы – обычно берется 10 интервалов.
3. Для каждого интервала оцениваем $CR_{K\%}\left(X_{target}\right)$ и $CR_{K\%}\left(X_{control}\right)$ и берем разность.

Средний взвешенный UpLift (Weighted Average UpLift)

Данная метрика представляет собой оценку UpLift по всей выборки и определяется следующий образом:

$$WAU = \frac{\sum_{i=1}^{k} N_i * UpLift_i}{\sum_{i=1}^{k} N_i},$$

где N_i – размер рабочей выборки на i – м интервале,

$UpLift_i = CR$ целевой группы на i – м интервале.

UpLift кривая (UpLift Curve)

Данная кривая строится как функция с нарастающим итогом, где для каждой точки задается соответствующий UpLift.

Определяется следующим образом:

$$UC(t) = \left(\frac{N_{target,Y=1}(t)}{N_{target,Y=0,1}(t)} - \frac{N_{control,Y=1}(t)}{N_{control,Y=0,1}(t)}\right) * \left(N_{target,Y=0,1}(t) + N_{control,Y=0,1}(t)\right),$$ где

$N_{target,Y=0,1}(t)$ – размер всей рабочей группы при всей выборке выборки размера t ,

$N_{target,Y=1}(t)$ – размер рабочей группы , совершившей целевое действие, при всей выборке размера t.

Аналогично и для контрольной группы.

Пример данной кривой на рисунке 2.

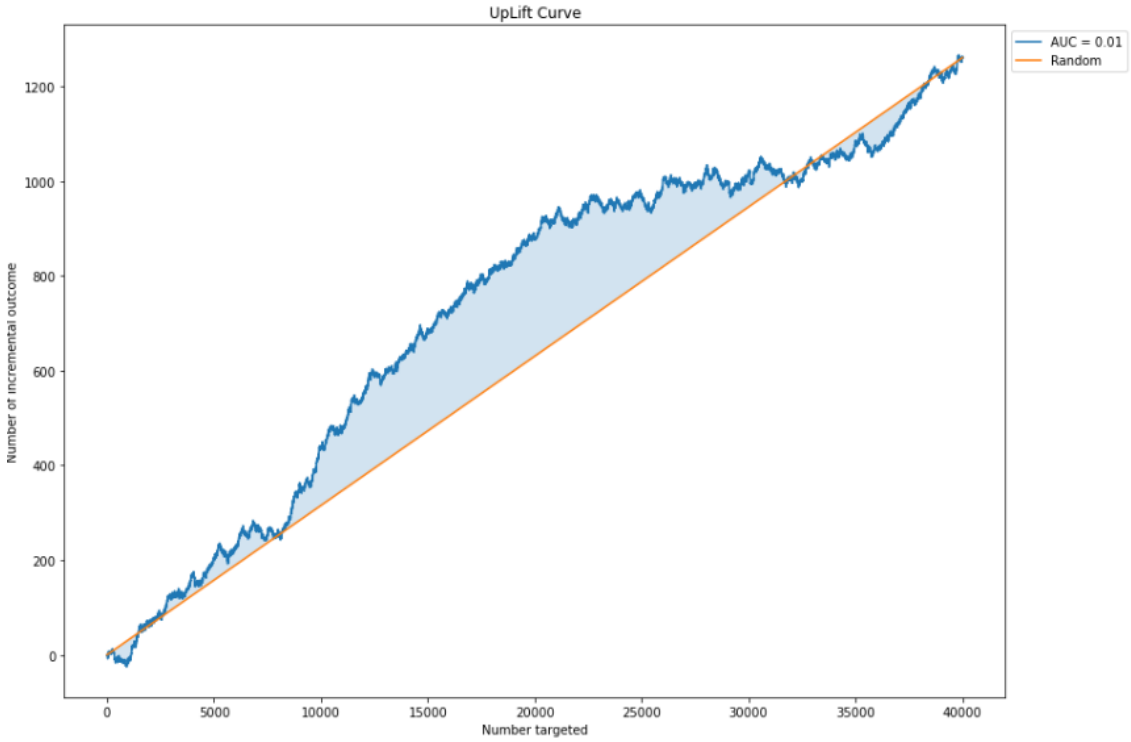

Рисунок 2

Qini кривая

Данную функцию можно выразить через UpLift кривую следующим образом:

$$Qini(t) = UC(t) * \frac{N_{target,Y=0,1}(t)}{\left(N_{target,Y=0,1}(t)+N_{control,Y=0,1}(t)\right)} =$$

$$= \left(\frac{N_{target,Y=1}(t)}{N_{target,Y=0,1}(t)} - \frac{N_{control,Y=1}(t)}{N_{control,Y=0,1}(t)} \right) * N_{target,Y=0,1}(t) =$$

$$= N_{target,Y=1}(t) - N_{control,Y=1}(t) * \frac{N_{target,Y=0,1}(t)}{N_{control,Y=0,1}(t)}$$

Данная кривая будет полезна в тех случаях, когда рабочая группа кратно превышает размер контрольной группы, с чем можно столкнуться во время исследования модели при внедрении в бизнес, когда у компании есть бюджет на произведение коммуникаций со всей клиентской базой и чтобы не упускать потенциальный доход, контрольная группа выделяется как можно меньше.

Таким образом будет получено инкрементальный эффект от коммуникаций в единицах измерения одного клиента.

Источник данных

За источник данных было взято уже завершенное соревнование по UpLift моделированию от российской мега-корпорации X5 Retail Group (ныне X5 Group) на платформе Open Data Science (ODS)[1]. Этот набор данных имеет преимущество над ныне существующими в открытом доступе благодаря тому, что это фактически моментальный снимок базы данных компании, во временном интервале за четыре месяца, хранящий в себе транзакции клиентов за соответствующий период, их обезличенные анкетные данные, обезличенный продуктовый справочник с данными по каждому товару сети.

Данное преимущество позволяет самому смоделировать и выделить важные признаки, и получить релевантный опыт работы с живыми, а не синтетическими или уже агрегированными данными.

Опишем набор данных детальнее. Он состоит из:

- Общей информации о клиентах:

	client_id	first_issue_date	first_redeem_date	age	gender
1	000012768d	2017-08-05 15:40:48.0000000	2018-01-04 19:30:07.0000000	45	U
2	000036f903	2017-04-10 13:54:23.0000000	2017-04-23 12:37:56.0000000	72	F
3	000048b7a6	2018-12-15 13:33:11.0000000	1900-01-01 00:00:00.0000000	68	F
4	000073194a	2017-05-23 12:56:14.0000000	2017-11-24 11:18:01.0000000	60	F
5	00007c7133	2017-05-22 16:17:08.0000000	2018-12-31 17:17:33.0000000	67	U

Рисунок 3

- Общая информация о товарах на складе:

	product_id	level_1	level_2	level_3	level_4	segment_id	brand_id	vendor_id	netto	is_own_trademark	is_alcohol
1	0003020d3c	c3d3a8e8c6	c2a3ea8d5e	b7cda0ec0c	6376f2a852	123	394a54a7c1	9eaff48661	0,4	0	0
2	0003870676	e344ab2e71	52f13dac0c	d3cfe81323	6dc544533f	105	acd3dd483f	10486c3cf0	0,68	0	0
3	0003ceaf69	c3d3a8e8c6	f2333c90fb	419bc5b424	f6148afbc0	271	f597581079	764e660dda	0,5	0	0
4	000701e093	ec62ce61e3	4202626fcb	88a515c084	48cf3d488f	172	54a90fe769	03c2d70bad	0,112	0	0
5	0007149564	e344ab2e71	52f13dac0c	d3cfe81323	6dc544533f	105	63417fe1f3	f329130198	0,6	0	0

Рисунок 4

[1] https://ods.ai/competitions/x5-retailhero-uplift-modeling

- История покупок клиента до коммуникаций:

	client_id	transaction_id	TRANSDATE	regular_points_received	express_points_received	regular_points_spent	express_points_spent	AMOUNT	store_id	product_id	QUANTITY	trn_sum_from_iss	trn_sum_from_red
1	2f4980b9bc	837832dee3	2019-03-06	14,1	0	0	0	1926,68	e87d6aefdc	4009f09b04	1	5	0
2	2f4980b9bc	837832dee3	2019-03-06	14,1	0	0	0	1926,68	e87d6aefdc	f848f9b373	1	38	0
3	2f4980b9bc	837832dee3	2019-03-06	14,1	0	0	0	1926,68	e87d6aefdc	34dd2b6d85	1	100	0
4	2f4980b9bc	837832dee3	2019-03-06	14,1	0	0	0	1926,68	e87d6aefdc	769a8a92bd	1	51	0
5	2f4980b9bc	1c5d5f6b57	2019-03-09	15,1	0	0	0	2029,89	e87d6aefdc	55e6ba317a	1	184	0

Рисунок 5

- Целевые переменные для обучения:

client_id	treatment_flg	target
000012768d	0	1
000036f903	1	1
00010925a5	1	1
0001f552b0	1	1
00020e7b18	1	1

Рисунок 6

- Данные для теста:

client_id	treatment_flg	target
fffe0abb97	0	0
fffe0ed719	0	1
fffea1204c	0	1
fffeca6d22	1	0
fffff6ce77	0	1

Рисунок 7

Анализ и агрегирование данных

Так как данные для UpLift моделирования составляют 4 Гб. в формате csv, что достаточно много для табличных данных самом экономном формате, то было решено взаимодействовать с ними через реляционный язык запросов SQL. Для этого был развернут локальный SQL Server на СУБД MSSQL и с помощью SQL Management Studio были загружены табличные данные.

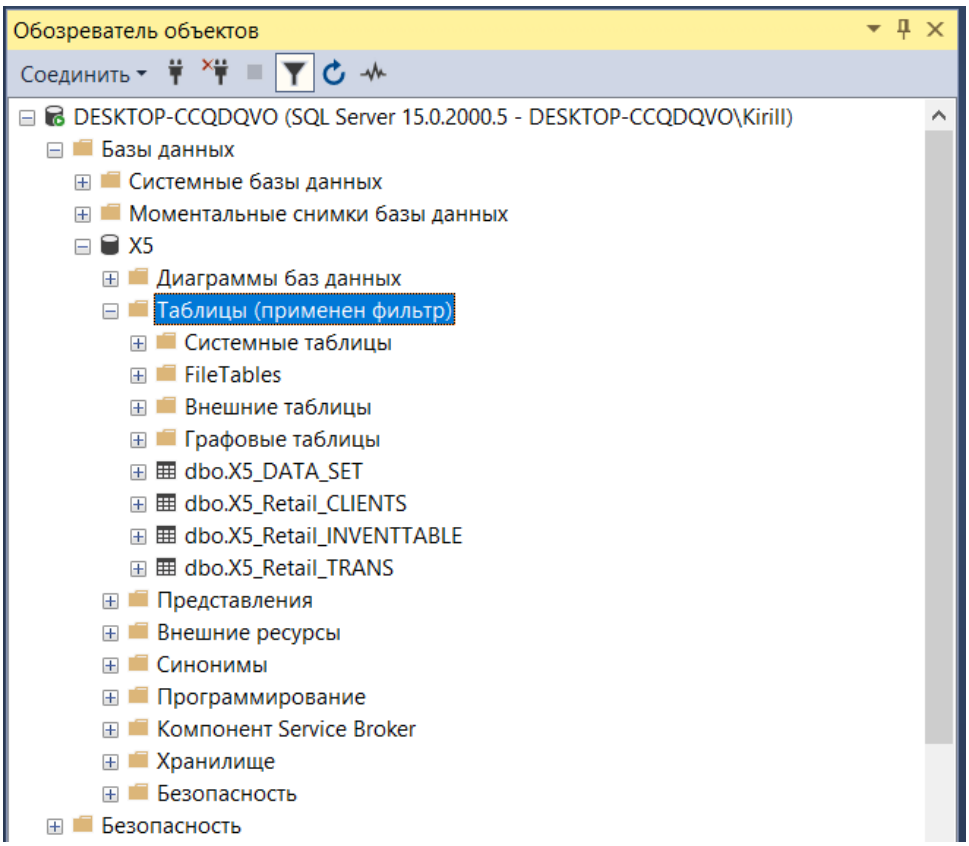

Рисунок 8

Через транзакции были выделены наиболее часто покупаемые товары для агрегации их в признаки.

Таким образом были выделены наиболее продаваемые:

- Уровни в иерархии товаров – рисунок 9.

	level	level_name	qty	Number
1	1	e344ab2e71	34006656	1
2	1	c3d3a8e8c6	21443244	2
3	1	ec62ce61e3	1649063	3
4	1		8084	4
5	2	ad2b2e17d2	10020239	1
6	2	ed2ad1797c	9353348	2
7	2	034aca0659	5971432	3
8	2	703f4b6eb0	5597036	4
9	2	1d2939ba1d	5003892	5
10	3	ca69ed9de2	4889143	1
11	3	b4f6ca38b2	2701359	2
12	3	334b74af37	2349777	3
13	3	e33cc0b2a4	2043846	4
14	3	453baf42c5	1811526	5
15	4	efbcf6d00c	2490149	1
16	4	146717c1b2	2240059	2
17	4	5330a84194	1587696	3
18	4	47fc199714	1454836	4

Рисунок 9

- Бренды – рисунок 10.

	brand	qty	Number
1	4da2dc345f	5815721	1
2	ab230258e9	2886997	2
3		2795792	3
4	037a833d06	2596996	4
5	8281de6bcb	2423379	5

Рисунок 10

- Поставщики – рисунок 11.

	vendor	qty	Number
1	43acd80c1a	8622311	1
2	e6af81215a	5688651	2
3	6bc8b3c476	2682034	3
4	63243765ed	1903282	4
5	bf8fc0055c	1584141	5

Рисунок 11

- Сегменты товаров – рисунок 12.

	segment	qty	Number
1	105	2776209	1
2	230	2701359	2
3	18	2345695	3
4	1	1941509	4
5	9	1824000	5

Рисунок 12

Для моделирования основных обучающих признаков был использован принцип RFM - сегментации[2]. То есть, по покупкам клиентов были определены следующие параметры:

- Частота покупок – количество покупок за расчетный период.
- Период с момента последней покупки.
- Сумма товарооборота с клиента за расчетный период - в нашем случае возьмем средний чек, так как это стратифицировать клиентов явным образом.

Также была собрана статистика по доле алкогольных товаров в чеке, доля внутренних брендов, среднее время между покупками и сопутствующая статистика по уровням товаров, брендам, поставщикам и сегментам. Вдобавок к этому были учтены и анкетные данные.

Таким образом было получено пространство из 30-ти обучающих признаков.

[2] RFM – сегментация // https://www.moengage.com/blog/rfm-analysis-using-rfm-segments/

Реализация UpLift моделирования методами машинного обучения

Базовая модель

Перед проведением экспериментов следует определить базовую модель, от функционала качества которой нужно будет отталкиваться. Так как базовая модель предполагает слепое прогнозирование без обработки пространства признаков, в нашем случае подойдет равномерная случайная величина, распределенная от -1 до 1.

По итогам такого моделирования получаем следующие значения метрик:

- WAU = 0.0332
- $UpLift_{30\%}$= 0.029
- Qini curve AUC = 0
- UpLift curve AUC = 0.

Экспериментальная установка

Исследование методов UpLift моделирования с помощью машинного обучения реализовано на высокоуровневом языке программирования Python, с использованием библиотек scikit-learn, scikit-uplift, CatBoost.

Для сравнения методов моделирования используется модель градиентного бустинга с базовыми параметрами, реализованный в библиотеке CatBoost.

Чтобы избежать ложных выводов по результатам работы модели на тестовом множестве, в исследовании используется кросс валидация с разбиением выборки на 5 долей. По итогу кросс валидации будет браться средняя по метрикам качества, на основе которых и будет сравнение. Иллюстрация работы кросс валидации на рисунке 13.

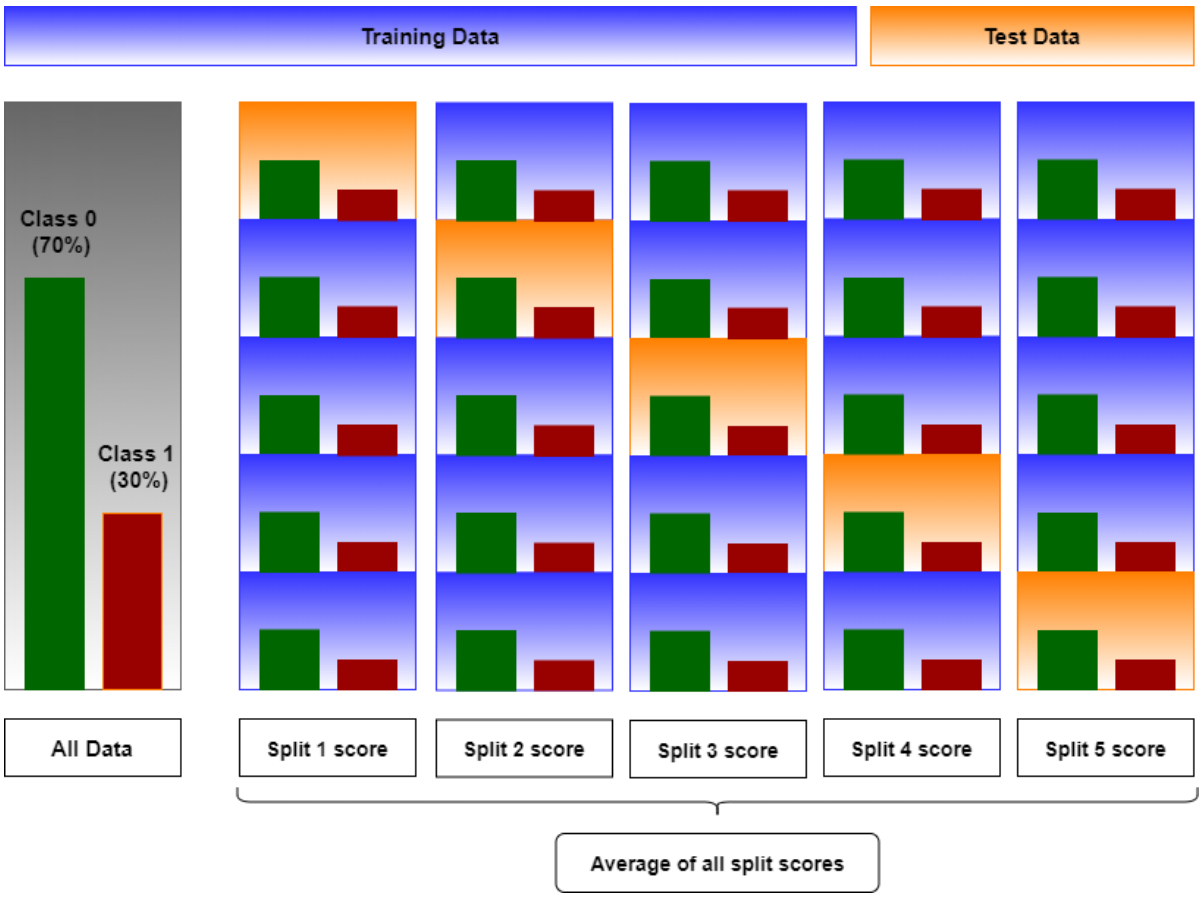

Рисунок 13. Схема кросс валидации

Моделирование с одной моделью

Самое простое и понятное решение. На тренировочной выборке обучаем любую модель бинарной классификации по всем обучающим признакам, включая коммуникационную переменную.

Далее для тестовой выборки задаем коммуникационную переменную равную 1 и определяем прогноз вероятности, что объект совершит целевое действие.

Далее для тестовой выборки задаем коммуникационную переменную равную 0 и снова определяем прогноз вероятности, что объект совершит целевое действие.

После этого берется разность вероятностей при наличии коммуникации и при отсутствии, что и будет значением UpLift.

По итогам моделирования получены следующие усредненные метрики:

- WAU = 0.0332
- $UpLift_{30\%}$= 0.0296
- Qini curve AUC = 0
- UpLift curve AUC = 0.

Также стоит добавить, что как на рисунках 14 и 15, для каждого разбиении фактически отсутствует инкрементальный эффект.

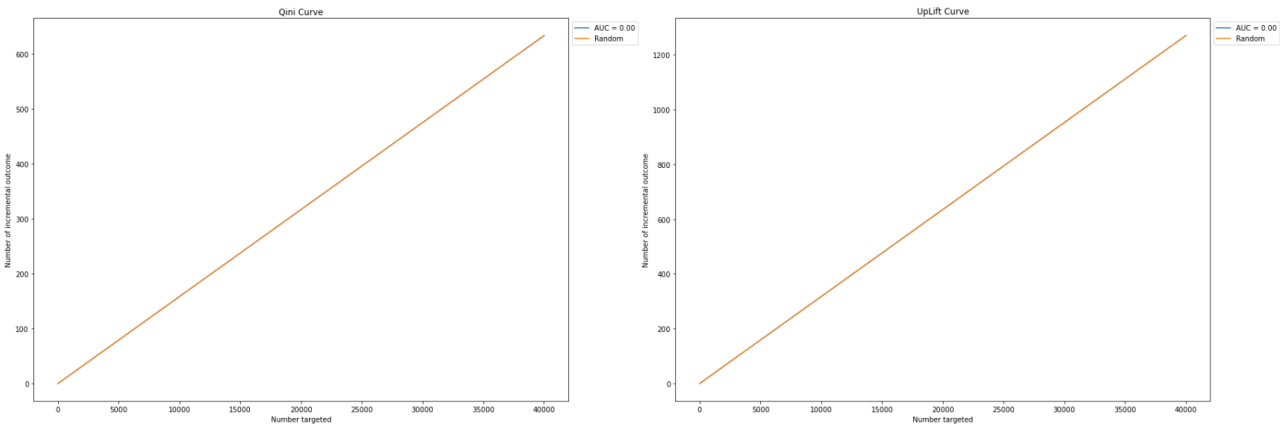

Рисунок 14

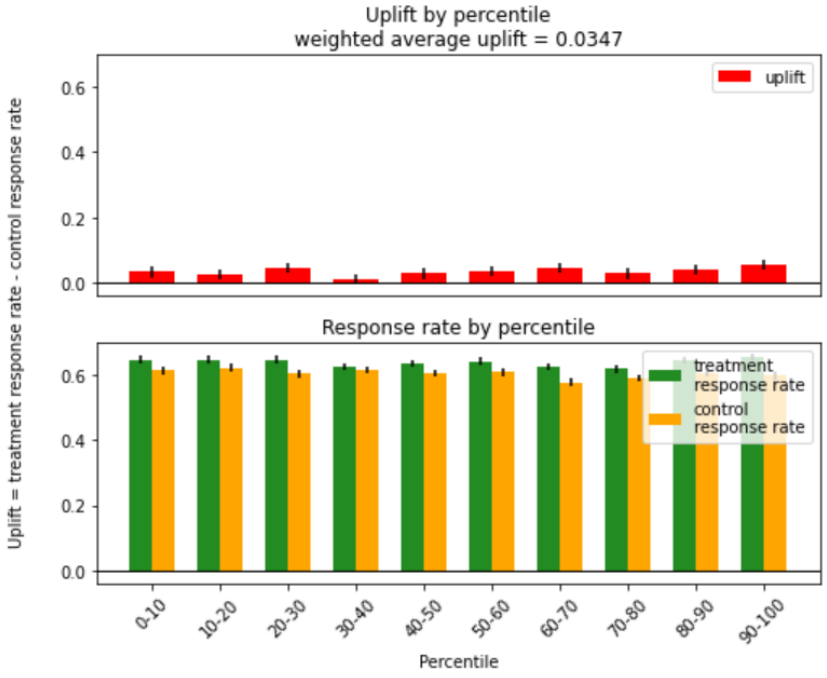

Рисунок 14

Моделирование с двумя независимыми моделями

Метод представляет собой обучение двух независимых моделей на тренировочных данных, где одна модель обучается на целевой группе, а вторая обучается на контрольной. Далее на тестовых данных прогнозируется вероятность выполнения целевого действия для одной и для второй модели и берется их разность.

Но тут сразу возникает нюанс, что при отсутствии равного объема целевой и контрольной группы, модели не будут иметь одинаковую полноту обучения. Но в нашем случае этого происходить не будет, так как рабочая и тестовая группа равного объема.

По итогам моделирования получены следующие усредненные метрики:

- WAU = 0.0327
- $UpLift_{30\%}$= 0.0471
- Qini curve AUC = 0.006
- UpLift curve AUC = 0.01

По итогу кросс валидации имеются два типа событий:

- Когда uplift позволяет получить инкрементальный эффект, как на рисунке 15.

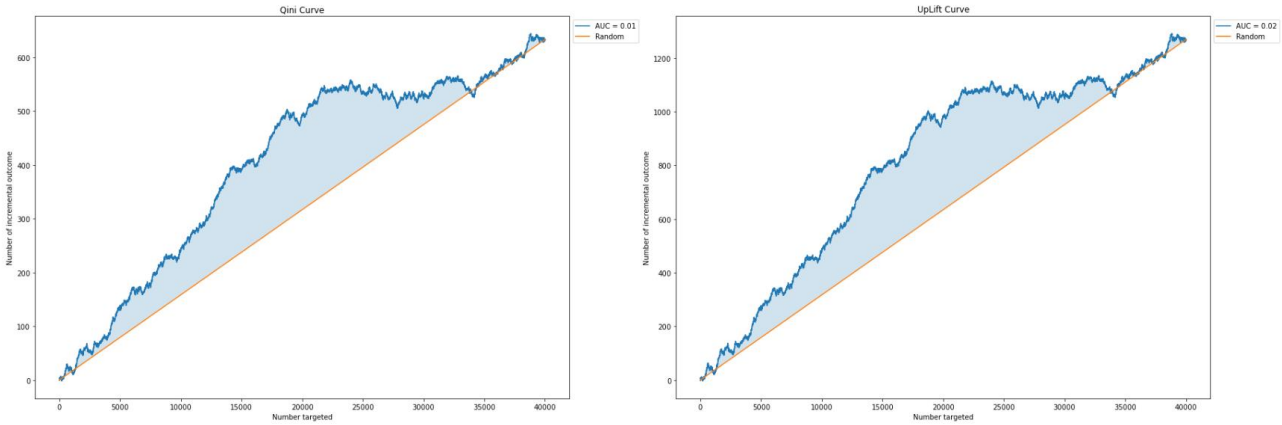

Рисунок 15

- Когда uplift позволяет получить инкрементальный эффект с переменным успехом, как на рисунке 16.

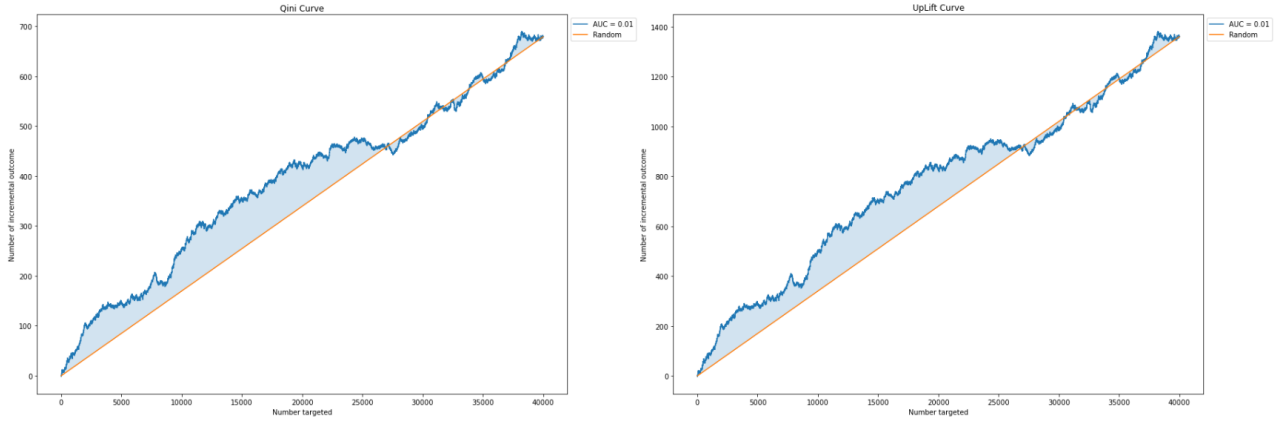

Рисунок 16

Так же стоит добавить, что после каждой итерации обучения разбивка конверсий по процентилям имеет следующий вид, как на рисунке 17. Что говорит об ухудшении результатов по сравнению с предыдущим экспериментом, так как при уменьшении размера выборки падает и конверсия, несмотря на увеличивающий UpLift.

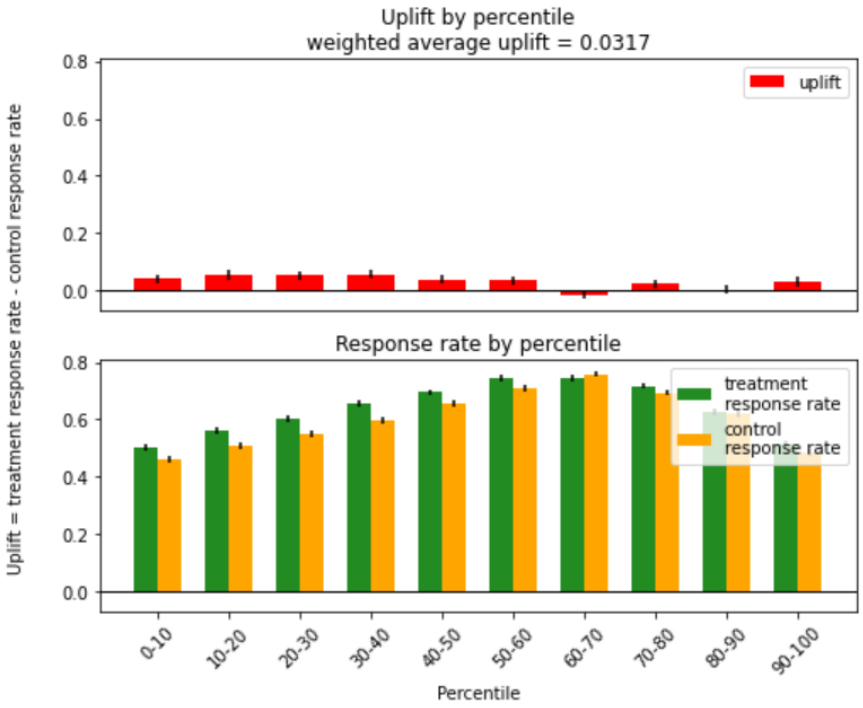

Рисунок 17

Метод трансформации класса

В данном методе мы вернемся снова к единой модели, но теперь преобразуем коммуникационную переменную и целевую переменную в одну следующим образом:

$Z_i = Y_i * \frac{W_i - p}{p*(1-p)}$, где $Y_i -$ целевая переменная, $W_i -$ коммуникационная переменная, $p = P(W = 1) = \frac{N_{target}}{N} -$ таким образом, получаем вероятность принадлежности объекта к целевой группе.

В нашем случае, $p = 0.5$. Тогда трансформированный класс будет иметь следующие значения:

$$Z_i = \begin{cases} 2, \text{при } W_i = 1;\ Y_i = 1 \\ 0, \text{при } W_i = 0{,}1;\ Y_i = 1 \\ -2, \text{при } W_i = 0; Y_i = 1 \end{cases}$$

Далее произведем переход к задаче регрессии для однозначной интерпретации прогноза.

По результатам моделирования были получены следующие усредненные результаты:

- WAU = 0.0331
- $UpLift_{30\%}$= 0.0401
- Qini curve AUC = 0.004
- UpLift curve AUC = 0.004

Несмотря на не лучшие значения усредненных метрик, распределение конверсий в зависимости от объема выборки, как на рисунке 18, говорит о том, что модель не уменьшает явно конверсию при уменьшении объема выборки, приближаясь в этом плане к результату первой модели. Но рисунок 19 говорит о присутствии инкремента, чего уже в первой модели не наблюдалось.

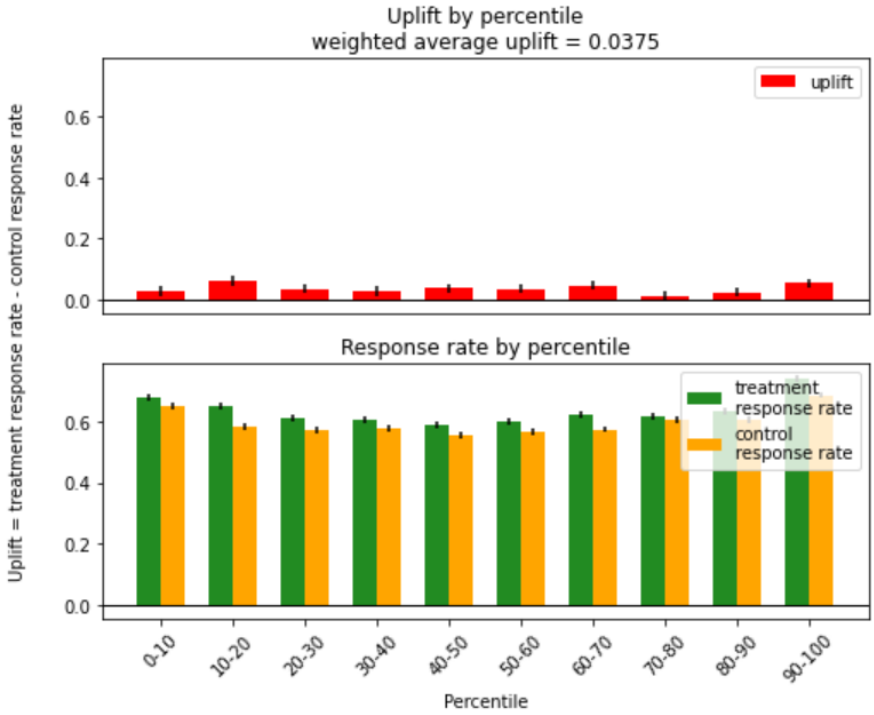

Рисунок 18

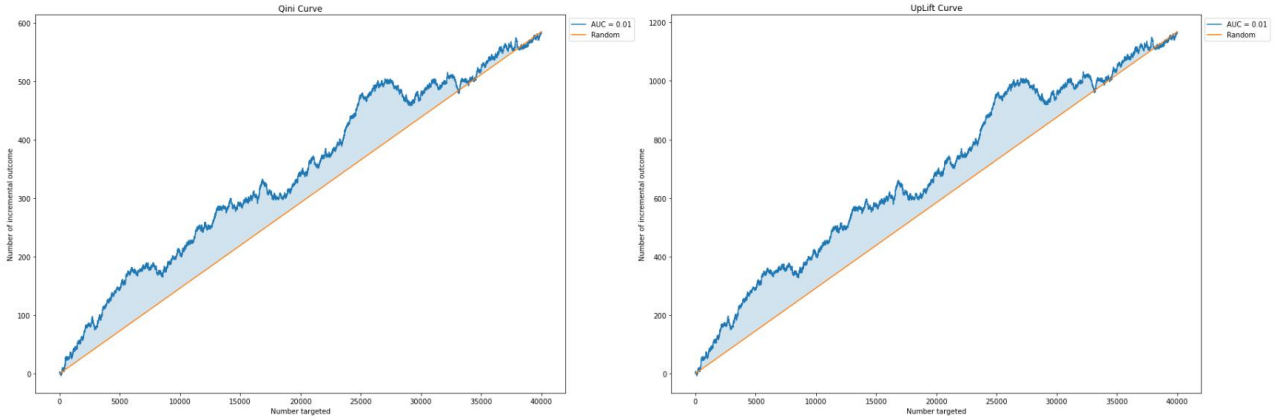

Рисунок 19

Исследований архитектур моделей машинного обучения

Так как по результатам подходов наилучшие имеет метод трансформации классов с переходом к задаче регрессии, то возникает вопрос – какая модель позволяет получить наилучший результат для нашей задачи.

Если считать, что наши целевые переменные достоверные, то косвенно оценивать качество моделей для сравнения можно и с помощью среднеквадратичной ошибки. Ведь та модель, которая лучше всего обучиться на тренировочных данных и тестовых данных и должна потенциально иметь наилучший UpLift на практике.

Сравнение структур моделей будет происходить с помощью библиотеки evalml, которая содержит внутри себя уже весь реализованный функционал.

По итогам поиска по 11-ти моделям, наилучшие показатели имеет уже использованный ранее градиентный бустинг из библиотеки Яндекс CatBoost. Лучшие результаты в таблице 1.

pipeline_name	validation_score	percent_better_baseline
CatBoost Regressor w/ Imputer + Select Columns Transformer	1,576280549	0,27%
Extra Trees Regressor w/ Imputer + Select Columns Transformer	1,576300785	0,14%
Elastic Net Regressor w/ Imputer + Standard Scaler + Select Columns Transformer	1,576303973	0,12%
Mean Baseline Regression Pipeline	1,576322675	0,00%

Таблица 1

Заключение

В данной работе были исследованы методы моделирования UpLift с помощью машинного обучения на исходных данных от X5 Retail Group, выложенных в открытый доступ.

В работе были рассмотрены метрики оценивания качества прогноза UpLift при алгоритме с одной моделью, при алгоритме с двумя независимыми моделями и при работе с одной моделью после трансформации классов и перехода к задаче регресии.

По итогам моделирования с данными обучающими признаками, лучшее качество имеет метод трансформации классов.

После определения метода было решено найти наилучшую структуру модели с помощью AutoML конвейров. В результате чего выяснилось, что с данными признаками лучшей моделью является градиентный бустинг в библиотеке CatBoost от компании Яндекс.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Источник данных // https://ods.ai/competitions/x5-retailhero-uplift-modeling

2. RF – сегментация // https://www.moengage.com/blog/rfm-analysis-using-rfm-segments/

_____ _____

Дата Подпись