



Практические задачи анализа данных

Лекция 7. Uplift моделирование

Московский авиационный институт
«МАИ»

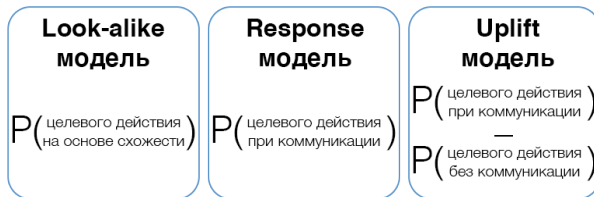
26 ноября 2021 г.

Задача продвижения продуктов за счет коммуникации с клиентом

Способы коммуникации

- SMS, push уведомления
- чат-бот в социальных сетях
- промо акции

Формирование сегментов для продвижения с помощью ML



- **Look-alike модель** оценивает вероятность того, что клиент выполнит целевое действие. Обучающей выборка — позитивные объекты (например, пользователи, установившие приложение) и случайные негативные объекты (сэмплирование небольшой подвыборки из всех остальных клиентов). Модель ищет клиентов, похожих на позитивные примеры.
- **Response модель** оценивает вероятность того, что клиент выполнит целевое действие при условии коммуникации. Обучающей выборка — данные, собранные после взаимодействия с клиентами.
- **Uplift модель** оценивает чистый эффект от коммуникации, пытаясь выбрать только тех клиентов, которые совершат целевое действие только после взаимодействия.

Разница реакций человека при наличии коммуникации и при ее отсутствии называется **causal effect**:

$$\tau_i = Y_i^1 - Y_i^0,$$

где Y_i^0 — реакция человека, если коммуникации не было,
 Y_i^1 — реакция человека, если коммуникация была

Зная признаковое описание i -го объекта X_i , можно ввести условный **усредненный эффект от воздействия** Conditional Average Treatment Effect (CATE):

$$\text{CATE}(x) = M[Y_i^1|X_i] - M[Y_i^0|X_i]$$

τ и CATE можно только оценить, так как одновременно не получится провести взаимодействие с объектом и не проводить

Оценка CATE или **Uplift** конкретного объекта:

$$\text{Uplift}(x) = M[Y_i | X_i = x, W_i = 1] - M[Y_i | X_i = x, W_i = 0],$$

где Y_i — наблюдаемая реакция клиента в результате маркетинговой кампании:

$$Y_i = W_i Y_i^1 + (1 - W_i) Y_i^0 = \begin{cases} Y_i^1 & \text{если } W_i = 1, \\ Y_i^0 & \text{если } W_i = 0. \end{cases}$$

$W_i = 1$, если объект попал в **целевую (treatment)** группу, в которой воздействие было произведено,

$W_i = 0$ — объект попал в **контрольную (control)** группу, где воздействия не было

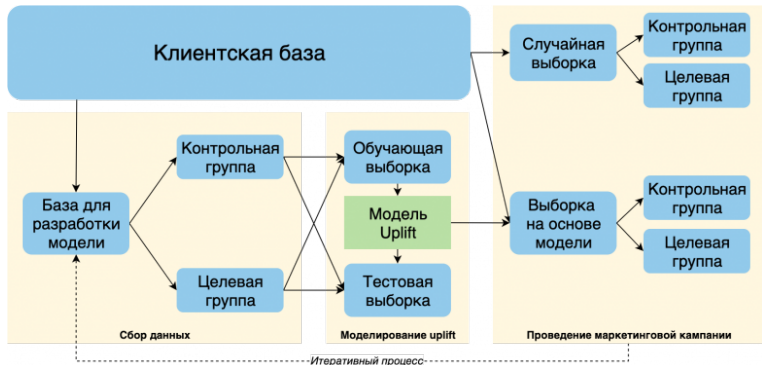
Далее $Y_i = 1$, если клиент совершил целевое действие и $Y_i = 0$ — не совершил

Когда используем Uplift

- Целевое действие выполняется клиентами с достаточно большой вероятностью без каких-либо коммуникаций. Например, мы рекламируем достаточно популярный продукт, но при этом не хотим тратить бюджет на клиентов, которые и без этого купят продукт
- Выполнено предположение об условной независимости (Conditional Independence Assumption — CIA): разделение на целевую и контрольную группу происходит случайно, а не в зависимости от значения какого-то признака

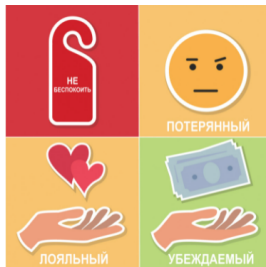
$$\text{CIA: } \{Y_i^0, Y_i^1\} \perp\!\!\!\perp W_i | X_i$$

- Условия проведения маркетинговой компании (проводится после прогноза Uplift для всех объектов) не должны существенно отличаться от условий проведения Uplift моделирования



Рекомендуется настраивать разработку Uplift модели и запуск кампании как итерационный процесс

Типы клиентов



- Не беспокоить — человек, который будет реагировать негативно, если с ним прокоммуницировать
- Потерянный — человек, который никогда не совершит целевое действие взаимодействие с такими создаёт дополнительные затраты
- Лояльный — человек, который будет реагировать положительно, несмотря ни на что, такие клиенты также впустую расходуют бюджет
- **Убеждаемый** — человек, который положительно реагирует на предложение, но при его отсутствии не выполнил бы целевого действия

Одна модель с признаком коммуникации

Treatment Dummy approach, Solo model approach, S-Learner

- Обучаемся сразу на обеих группах, величина W будет дополнительным признаком
- На тестовой выборке вероятности для каждого объекта вычисляем два раза, при двух значениях W :

$$p_1(x) = P(X = x | W = 1), \quad p_0(x) = P(X = x | W = 0)$$

- $\text{Uplift}(x) = p_1(x) - p_0(x)$, $x \in X_{\text{test}}$

Можно увеличить число признаков, добавив произведение признака на флаг взаимодействия: $x_{ij} \cdot w_i$

id	x_1	x_2	w	y		id	x_1	x_2	w	$x_1 w$	$x_2 w$	y
1	1	13	1	0	\Rightarrow	1	1	13	1	1	13	0
2	0	23	0	1		2	0	23	0	0	0	1
3	0	41	1	1		3	0	41	1	0	41	1
4	1	26	0	0		4	1	26	0	0	0	0

Две независимые модели

Two models approach, T-learner, difference two models

Отдельное моделирование условных вероятностей на целевой и контрольной группах

- Строим модель на целевой группе a_T
- Строим модель на контрольной группе a_C
- Строим прогнозы двух моделей на тестовой выборке и считаем

$$\text{Uplift}(x) = \text{Predict}_T(x) - \text{Predict}_C(x), \quad x \in X_{\text{test}}$$

Под $\text{Predict}(x)$ здесь понимается вероятность принадлежности x к классу $Y = 1$

Две зависимые модели (через данные)

Dependent Data Representation, Dependent Feature Representation

- Обучаем классификатор на контрольной группе и стоим прогнозы для объектов целевой группы

$$P_C = P(Y = 1 | X_{\text{control}}, W = 0)$$

- Обучаем второй классификатор на целевой группе, добавляя $P_C(X_{\text{treat}})$ в качестве признака

$$P_T = P(Y = 1 | X_{\text{treat}}, P_C(X_{\text{treat}}), W = 1)$$

- Вычисляем Uplift на тестовой выборке

$$\text{Uplift}(x) = P_T(x, P_C(x)) - P_C(x), \quad x \in X_{\text{test}}$$

Аналогично, можно сначала обучить классификатор P_T , а затем использовать его предсказания в качестве признака

Две зависимые модели (перекрестная зависимость)

X-learner

Применяется, если целевая группа достаточно маленькая

- Обучаем две модели: одну на контрольной группе a_C , другую — на целевой a_T
- Строим прогнозы меток классов, используя a_C на целевой группе, а a_T на контрольной

$$\hat{Y}_C = a_C(x), \quad x \in X_{\text{treat}}$$

$$\hat{Y}_T = a_T(x), \quad x \in X_{\text{control}}$$

- Вычисляем разницу между истинными метками и прогнозами моделей

$$\Delta_T = Y_{\text{treat}} - \hat{Y}_C, \quad \Delta_C = Y_{\text{control}} - \hat{Y}_T$$

- Обучаем две модели на преобразованных метках Δ_T и Δ_C :

$$a_T^{\text{new}}(X_{\text{treat}}, \Delta_T), \quad a_C^{\text{new}}(X_{\text{control}}, \Delta_C)$$

- Взвешенная с некоторым коэффициентом $g \in [0, 1]$ сумма предсказаний этих моделей и будет Uplift на тестовой выборке

$$\text{Uplift}(x) = (1-g) \cdot \text{Predict}_T^{\text{new}}(x) + g \cdot \text{Predict}_C^{\text{new}}(x), \quad x \in X_{\text{test}}$$

$g \rightarrow 1$, если размер целевой группы большой по сравнению с размером контрольной, и $g \rightarrow 0$, если наоборот

g можно рассматривать как функцию от объекта — $g(x)$

Трансформация классов

Class Transformation approach, Revert Label approach

Решаем задачу классификации с новой целевой переменной

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

или

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

id	x_1	x_2	w	y		id	x_1	x_2	z
1	1	13	1	0	\Rightarrow	1	1	13	0
2	0	23	0	1		2	0	23	0
3	0	41	1	1		3	0	41	1
4	1	26	0	0		4	1	26	1

Распишем подробнее, чему равна вероятность новой целевой переменной:

$$\begin{aligned} P(Z = 1|X = x) &= P_T(Y = 1|X = x) \cdot P(W = 1) + \\ &+ P_C(Y = 0|X = x) \cdot P(W = 0) \end{aligned}$$

Будем считать, что во время эксперимента контрольные и целевые группы были разделены в равных пропорциях, т.е. $P(W = 0) = P(W = 1) = \frac{1}{2}$. Тогда

$$\begin{aligned} 2P(Z = 1|X = x) &= P_T(Y = 1|X = x) + P_C(Y = 0|X = x) = \\ &= P_T(Y = 1|X = x) + (1 - P_C(Y = 1|X = x)) \end{aligned}$$

$$\begin{aligned} \text{Uplift}(x) &= P_T(Y = 1|X = x) - P_C(Y = 1|X = x) = \\ &= 2P(Z = 1|X = x) - 1 \end{aligned}$$

т.е. прогноз новой целевой переменной даёт нам значение Uplift

Трансформация классов (регрессия)

Transformed outcome

Новая целевая переменная

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

где $p = P(W_i = 1 | X = x)$ — вероятность отнесения к целевой группе

Величину p можно оценить как долю объектов с $W = 1$ в выборке или решить задачу классификации с целевой переменной W

id	x_1	x_2	w	y		id	x_1	x_2	z
1	1	13	1	0	$p = 0.5$ \Rightarrow	1	1	13	0
2	0	23	0	1		2	0	23	-2
3	0	41	1	1		3	0	41	2
4	1	26	0	0		4	1	26	0

Для новой целевой переменной Z обучаем модель регрессии с функционалом ошибки $MSE = \frac{1}{m} \sum_{i=1}^m (Z_i - \hat{Z}_i)^2$

Покажем, что условное мат. ожидание трансформированного таргета Z и есть желаемый causal effect

$$\tau_i(x) = M[Z_i | X_i = x]$$

Так как $Y_i = W_i Y_i^1 + (1 - W_i) Y_i^0$, то

$$Z_i = \frac{W_i}{p} Y_i^1 - \frac{1 - W_i}{(1 - p)} Y_i^0$$

Тогда

$$\begin{aligned} M[Z_i | X_i = x] &= M \left[\frac{W_i}{p} Y_i^1 | X_i = x \right] - M \left[\frac{1 - W_i}{1 - p} Y_i^0 | X_i = x \right] = \\ &= \frac{Y_i^1}{p} M[W_i | X_i = x] - \frac{Y_i^0}{1 - p} M[1 - W_i | X_i = x] \end{aligned}$$

При этом $M[W_i|X_i = x] = M[W_i] = P(W_i = 1) = p$

Тогда

$$M[Z_i|X_i = x] = \frac{Y_i^1}{p}p - Y_i^0(1 - p) = Y_i^1 - Y_i^0 = \tau_i(x)$$

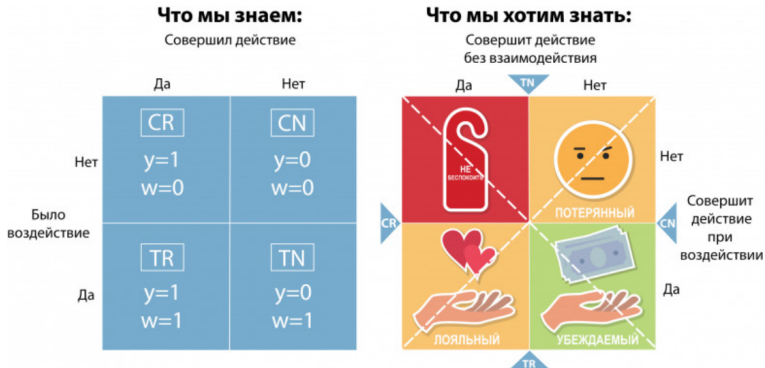
Многоклассовая модель

Generalized Lai Method

Разделим клиентов их на 4 непересекающихся класса

- Клиент выполнил целевое действие и мы с ним не взаимодействовали (Control Responder — **CR**):
 $Y_i = 1, W_i = 0$
- Клиент не выполнил целевое действие и мы с ним не взаимодействовали (Control Non-Responder — **CN**):
 $Y_i = 0, W_i = 0$
- Клиент выполнил целевое действие и мы с ним взаимодействовали (Treated Responder — **TR**):
 $Y_i = 1, W_i = 1$
- Клиент не выполнил целевое действие и мы с ним взаимодействовали (Treated Non-Responder — **TN**):
 $Y_i = 0, W_i = 1$

- среди CR могут быть как клиенты «не беспокоить» , так и лояльные клиенты
- среди CN могут быть как потерянные, так и убеждаемые типы клиентов
- среди TR могут быть как убеждаемые, так и лояльные типы клиентов
- среди TN могут быть как «не беспокоить», так и потерянные типы клиентов



id	x_1	x_2	w	y		id	x_1	x_2	z
1	1	13	1	0	$p=0.5$ \Rightarrow	1	1	13	TN
2	0	23	0	1		2	0	23	CR
3	0	41	1	1		3	0	41	TR
4	1	26	0	0		4	1	26	CN

$$\text{Uplift}(x) = [P(TR|X=x) + P(CN|X=x)] - \\ - [P(TN|X=x) + P(CR|X=x)]$$

Если выборки сильно не сбалансированы и r_C — доля клиентов в контрольной группе, тогда

$$\text{Uplift}(x) = \left[\frac{P(TR|X=x)}{1-r_C} + \frac{P(CN|X=x)}{r_C} \right] - \\ - \left[\frac{P(TN|X=x)}{1-r_C} + \frac{P(CR|X=x)}{r_C} \right]$$

Методы, основанные на деревьях

Decision trees for uplift modeling, Causal trees

Предыдущие методы имеют следующие недостатки:

- В методах с двумя моделями при расчете финального предсказания учитываются результаты двух моделей, а значит их ошибки могут суммироваться
- Если для обучения будут использоваться принципиально разные модели или природа данных целевой и контрольной групп будут 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

Построим деревья решений с специальным критерием разбиения

Дерево строится так, чтобы максимизировать расстояние (дивергенцию) между распределениями целевой переменной у контрольной и целевой групп

$$\text{Gain}_D = D_{\text{after split}}(P_T, P_C) - D_{\text{before split}}(P_T, P_C),$$

P_C , P_T — распределение целевой переменной в контрольной и целевой группах

$D(\cdot, \cdot)$ — расхождение между двумя распределениями

В качестве расхождения $Q = (q_1, \dots, q_m)$ и $P = (p_1, \dots, p_m)$ используется

- Дивергенция Кульбака—Лейблера

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

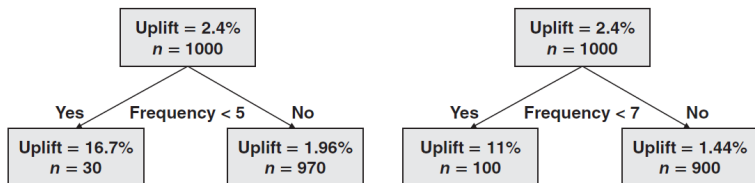
- Евклидово расстояние

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

- хи-квадрат

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

Также важное условие разбиения — это сведение к минимуму разницы между количеством объектов, попавших в левую и правую дочернюю вершину. Слева на картинке изображен пример плохого разбиения



Чтобы контролировать количество объектов в разбиении вычисляется взвешенная дивергенция

$$\tilde{D}(P_T, P_C) = \frac{m_L D(P_T(L), P_C(L)) + m_R D(P_T(R), P_C(R))}{m_L + m_R},$$

где n_L , n_R — количество объектов, попавших в левую и правую дочернюю вершину;

$P_T(L)$, $P_C(L)$, $P_T(R)$, $P_C(R)$ — распределения целевой переменной в целевой и контрольной группе для левой и правой дочерних вершин

После реализации дерева с новым критерием разбиения можно использовать этот алгоритм в качестве базового алгоритма в ансамблях

Как измерять качество Uplift моделей?

Использовать классические метрики качества для оценки обученной модели не получится, так как нет настоящих значений Uplift

Предсказанные значения Uplift сравнить не с чем

Вычисление метрик демонстрируется на данных хакатона BigTarget от Ленты и Microsoft (2020)

[Uplift metrics tutorial.ipynb](#)

uplift@k

Самая простая и понятная метрика — размер Uplift на первых k процентах вариационного ряда выборки

Мы хотим отобрать какое-то количество клиентов для взаимодействия. Пусть бюджет рассчитан на $k\%$ клиентов, тогда нам нужно оценить качество прогноза не на всей тестовой выборке, а только на объектах с наибольшим значением Uplift при отсечении по порогу в $k\%$

Для расчета uplift@k нужно отсортировать выборку по величине предсказанного Uplift и посмотреть разницу средних значений целевой переменной Y (response rate) в целевой и контрольной группах

$$\text{Uplift@k} = \text{response_rate@k}(X_{\text{treat}}) - \text{response_rate@k}(X_{\text{control}}),$$

где $\text{response_rate@k} = \text{mean}(Y@k)$,

$Y@k$ — значение целевой переменной на $k\%$

Теоретический Uplift@k принимает значения от -1 (когда в целевой группе все $Y = 0$, а в контрольной группе все $Y = 1$) и может достигать величины 1 (в целевой группе все $Y = 1$, в контрольной все $Y = 0$)

На практике Uplift@k принимает значения от 0 до 1, в зависимости от k , особенностей датасета и качества модели

Рассчитывать эту метрику можно двумя способами: сначала сортировать по предсказанному Uplift и далее считать разницу двух групп или сначала сортировать объекты из контрольной и целевой групп по отдельности

`y_val` — целевая переменная на валидации

`trmnt_val` — столбец флага коммуникации на валидации

`uplift_ct` — предсказанный Uplift методом ClassTransformation

```
from sklift.metrics import uplift_at_k

# k = 10%
k = 0.1

# strategy='overall' sort by uplift treatment and control together
uplift_overall = uplift_at_k(y_val, uplift_ct, trmnt_val, strategy='overall', k=k)

# strategy='by_group' sort by uplift treatment and control separately
uplift_bygroup = uplift_at_k(y_val, uplift_ct, trmnt_val, strategy='by_group', k=k)

print(f"uplift@{k * 100:.0f}%: {uplift_overall:.4f} (sort groups by uplift together)")
print(f"uplift@{k * 100:.0f}%: {uplift_bygroup:.4f} (sort groups by uplift separately)")

uplift@10%: 0.1467 (sort groups by uplift together)
uplift@10%: 0.1503 (sort groups by uplift separately)
```

Uplift by percentile (by decile, by bin)

Поведение метрики при разных значениях порога k в виде таблицы или графика

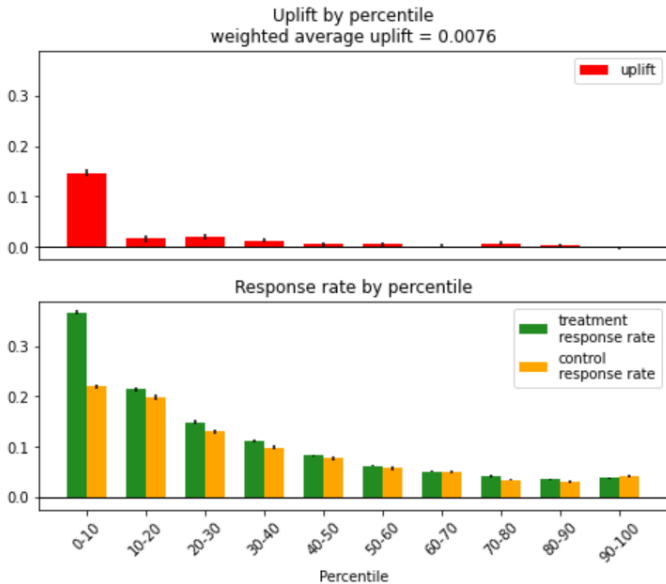
- Сортируем по предсказанному значению Uplift
- Делим отсортированные данные на бины
- В каждом бине отдельно оцениваем Uplift как разность между средними значениями целевой переменной в целевой и контрольной группах

```
from sklift.metrics import uplift_by_percentile

uplift_by_percentile(y_val, uplift_ct, trmnt_val,
                     strategy='overall',
                     total=True, std=True, bins=10)
```

	n_treatment	n_control	response_rate_treatment	response_rate_control	uplift
percentile					
0-10	12715	7896	0.366339	0.219605	0.146734
10-20	15560	5051	0.214267	0.197783	0.016485
20-30	15683	4928	0.149525	0.130682	0.018843
30-40	15675	4936	0.111388	0.098663	0.012725
40-50	15798	4813	0.082795	0.077914	0.004881
50-60	15776	4835	0.062500	0.057911	0.004589
60-70	15768	4843	0.051560	0.050176	0.001385
70-80	15793	4818	0.042171	0.034869	0.007301
80-90	15884	4727	0.035193	0.031521	0.003672
90-100	16116	4494	0.039030	0.041611	-0.002582
total	154768	51341	0.110126	0.102569	0.007557

n_treatment, n_control — размер целевой и контрольной группы



Weighted average Uplift (WAU)

— единое значение на всей тестовой выборке

Построим Uplift by bin и вычислим

$$\text{WAU} = \frac{\sum_{i=1}^k (n_T)_i \text{Uplift}_i}{\sum_{i=1}^k (n_T)_i},$$

где i — номер бина, k — число бинов, $(n_T)_i$ — размер целевой группы в i -м бине, Uplift_i — средняя величина Uplift в i -м бине

```
from sklift.metrics import weighted_average_uplift

uplift_full_data = weighted_average_uplift(y_val, uplift_ct, trmnt_val, bins=10)
print(f"average uplift on full data: {uplift_full_data:.4f}")
```

```
average uplift on full data: 0.0189
```

Uplift кривая (Uplift Curve)

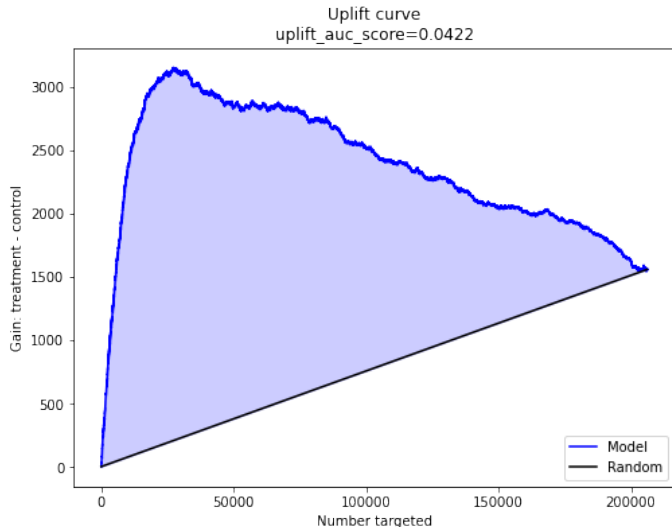
Uplift кривая совокупного прироста строится как функция от количества объектов, нарастающим итогом

Объекты выборки располагаются по возрастанию Uplift

$$UC(s) = \left(\frac{n_{T,Y=1}(s)}{n_T(s)} - \frac{n_{C,Y=1}(s)}{n_C(s)} \right) (n_T(s) + n_C(s)),$$

где s — накопившиеся число объектов, $n_T(s)$, $n_C(s)$ — число объектов целевой и контрольной группы среди s объектов
 $n_{T,Y=1}(s)$, $n_{C,Y=1}(s)$ — число объектов целевой и контрольной группы среди s объектов с меткой $Y = 1$

```
# only real  
# perfect=False  
plot_uplift_curve(y_val, uplift_ct, trmnt_val, perfect=False);
```



Qini кривая (Qini curve)

Если размер целевой группы n_T сильно больше, чем n_C , то Uplift кривая не оправданно завышает значение кривой Qini кривая аналогична Uplift кривой, но штрафует за большие значения n_T

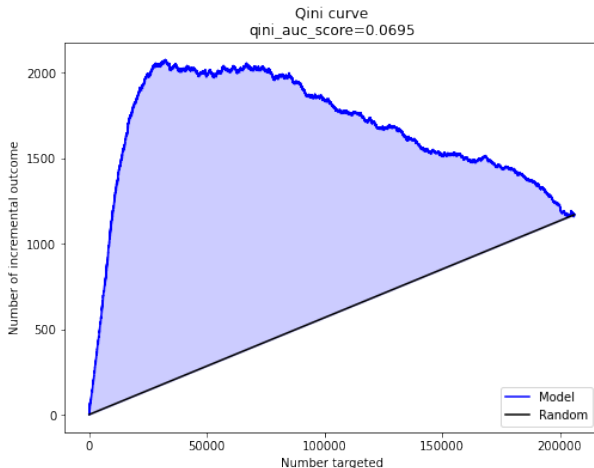
$$Qini(s) = n_{T,Y=1}(s) - \frac{n_{C,Y=1}(s)n_T(s)}{n_C(s)}$$

Кривые Uplift и Qini связаны соотношением

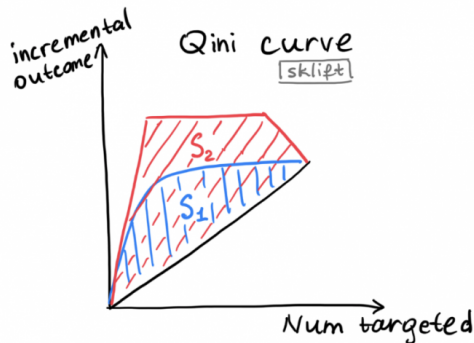
$$UC(s) = \frac{Qini(s)(n_T(s) + n_C(s))}{n_T(s)}$$

Когда контрольная и целевая группы сбалансированы, то Qini кривая будет в два раза ниже, чем Uplift

```
# no ideal Qini curve  
# only real Qini curve  
# perfect=False  
plot_qini_curve(y_val, uplift_ct, trmnt_val, perfect=False);
```

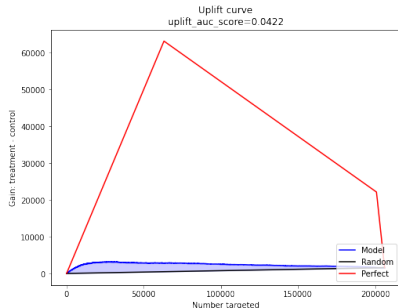
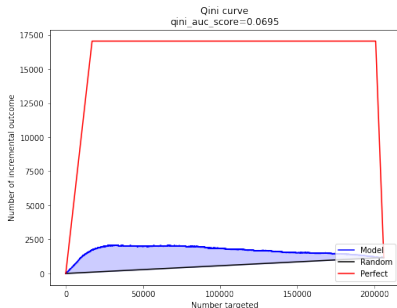


Коэффициент Qini или AUC Qini



$$\text{Qini coefficient} = \frac{S_1}{S_2}$$

Красным обозначена площадь под идеальной кривой



Для Uplift кривой коэффициент называется — Area under Uplift curve и считается аналогичным образом


```
from sklift.metrics import qini_auc_score

# AUQC = area under Qini curve = Qini coefficient
aucq = qini_auc_score(y_val, uplift_ct, trmnt_val)
print(f"Qini coefficient on full data: {aucq:.4f}")
```

Qini coefficient on full data: 0.0695

```
from sklift.metrics import uplift_auc_score

# AUUQ = area under uplift curve
auuc = uplift_auc_score(y_val, uplift_ct, trmnt_val)
print(f"Uplift auc score on full data: {auuc:.4f}")
```

Uplift auc score on full data: 0.0422

BigTarget = Lenta + Microsoft

<https://bigtarget.online/>

Данные <https://www.kaggle.com/mrmorj/bigtarget>

1-е место https://github.com/kurochkinSergei/bigtarget_barcode

2-е место https://github.com/DKozl50/pluh_sms

4-е место <https://github.com/zaikoval/bigtarget>

Лента

Данные

<https://www.notion.so/4b9a80e1e20e4726a025b3fc2c1ad14c>

Лучшее решение <https://youtu.be/gIHwnDH9PD8>

<https://github.com/anordertoreclaim/LentaHack>

Все решения

<https://drive.google.com/file/d/1fg6qPrm4ns1s5iobaBnnimiwgg1uyZz-/view>

Мегафон

Постановка задачи и данные <https://ods.ai/tracks/df21-megafon/competitions/megafon-df21-comp>

Решение от LightAutoML+обсуждение метрики (на уровне 16-го места)

<https://youtu.be/Gv4mhNhAqRs>

[LigthAtoML Uplift](#)

<https://youtu.be/ox9tkVgpQp4>

на уровне 2-го места

<https://github.com/wrapper228/Megafon-Uplift-Competition>

5-е место + EDA <https://www.kaggle.com/mrmorj/bigtarget-eda-uplift-model-insights-task>

7-е место https://github.com/denismashukov/megafon__uplift

Конкурс X5 Retail

Данные и постановка задачи

<https://ods.ai/competitions/x5-retailhero-uplift-modeling/data>

Постановка задачи и бизнес-мотивация

<https://youtu.be/yFQAIJBYXI0>

1-е место. Кратко и очень полезно: о датасете, проблемы валидации, специфичные значения параметров LightGBM, жадный отбор признаков

<https://youtu.be/fE9M4DcZvng>

2-е место https://github.com/kirrlx1994/Retail_hero

Датасеты

<https://www.uplift-modeling.com/en/latest/api/datasets/index.html>

Смежный вопрос A/B тестирование

<https://youtu.be/XWw4Wi6K0QU>

Лекция от В. Бабушкина Uplift моделирование (2 часа, Uplift с 34 мин)

https://youtu.be/u69k2d-7E_0

<https://www.packtpub.com/free-ebook/ai-crash-course/9781838645359>

<https://github.com/PacktPublishing/AI-Crash-Course/tree/master/Chapter%2010>

Краткая версия <https://youtu.be/yFQAIJBYXI0>

DataFest 2021 секция Маркетинга

<https://ods.ai/tracks/ml-in-marketing-df2021>

Uplift-моделирование. Сравнение различных подходов

https://youtu.be/5CeTziR6_kg

Uplift моделирование для задачи ранжирования клиентов по их чувствительности к изменению % ставки по кредиту (про задачу с 24:40)

https://youtu.be/Zivl9Y_8Cew

Рассылка персональных сообщений физ лицам клиентам банка

<https://youtu.be/UNegf9Rgpnw>

W. Verbeke, B. Baesens, C. Bravo. Profit-Driven Business Analytics: A Practitioner's Guide to Transforming Big Data into Added Value (Chapter 4). 2018. Wiley

Цикл статей на Хабр, в конце статей смотрите ссылки на литературу

https://habr.com/ru/company/ru_mts/blog/485980/

Курс от авторов на ODS

<https://ods.ai/tracks/uplift-modelling-course>

State-of-the-Art in Uplift Modeling

<https://doi.org/10.1198/016214504000001880>

Применение RL в задачах Uplift

[Optimising Individual Treatment Effect Using Bandits](#)

Библиотека (Uplift modeling)

<https://github.com/maks-sh/scikit-uplift>

Библиотека с деревьями (Uplift Modeling and Causal Inference)

<https://github.com/uber/causalml>

Пара глав-примеров с подробным описанием задачи

https://pbiecek.github.io/xai_stories/story-uplift-modelling.html

https://pbiecek.github.io/xai_stories/story-uplift-marketing1.html

Ноутбуки

Пример из LightAutoML [Ulift LAMA](#)

Примеры из библиотек scikit-uptlift и causalm1

Примеры разные

<https://www.steveklosterman.com/uplift-modeling/>

<https://www.kaggle.com/hughhuyton/criteo-uplift-modelling>

<https://www.kaggle.com/davinwijaya/why-you-should-start-using-uplift-modeling>

Пример по Лента+Microsoft

<https://www.kaggle.com/mrmorj/bigtarget-eda-uplift-model-insights-task>

Пример по данным X5 <https://www.kaggle.com/antoneshiev/ml2020-hw4>