Содержание

ВВЕДЕНИЕ

В данной выпускной квалификационной работе рассматривается проблема ранжирования клиентов для осуществления коммуникации самым убеждаемым клиентам, которые без той самой коммуникации не совершат целевое действие.

В данной работе решается проблема прогноза инкрементального отклика клиента при планировании коммуникаций с помощью UpLift моделирования методами машинного обучения, где на основании полученного значения будет происходить ранжирование клиентов от самых убеждаемых к самым неприкасаемым, для повышения эффективности коммуникации при сохранении объемов затрат на ее проведение.

Результаты данной работы будут использованы в отделе управления взаимоотношений с клиентами в ретейл компании косметики и парфюмерии.

Появление данной задачи обусловлено желанием проводить нативную коммуникацию только с теми людьми, которым это нужно, чтобы не тратить денежный ресурс в пустую на тех, кому коммуникация не нужна или даже вызовет негативные эмоции и заставит уйти к конкуренту.

Объектом исследования являются клиенты ретейл сети косметики и парфюмерии, которых мы хотим ранжировать для выделения наиболее убеждаемых

Предметом исследования выступает сравнение различных алгоритмов ранжирования методами машинного обучения на двух различных источниках данных.

Цель данной работы – исследование подходов к разработке UpLift моделирования методами машинного обучения для планирования

проведения рекламной кампании.

Данная работа развивает описанные в [1] идеи прогнозированию эффекта от коммуникации для каждого клиента при планировании рекламной кампании. С помощью показателей качества обучения из [2] и [3] удалось определить наилучший алгоритм для Uplift моделирования из описанных в [1], [4].

ОСНОВНАЯ ЧАСТЬ

1 ТЕОРЕТИЧЕСКАЯ ЧАСТЬ

1.1 Функционалы качества прогноза моделей

1.1.1 UpLift на k – процентах выборки

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.

Самая простая и интуитивно понятная метрика, описанная в [2], особенно для применения в бизнесе и для интерпретации – UpLift на k – процентах выборки.

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.

Определяется формулой (1):

$$UpLift_{K\%} = CR_{K\%}\left(X_{target}\right) - CR_{K\%}\left(X_{control}\right), \quad (1)$$

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%.

1.1.2 Кривая UpLift

Далее определим кривую, которая строится как функция с нарастающим итогом, где для каждой точки задается соответствующий UpLift.

Определяется формулой (2):

$$UC(t) = \left(\frac{N_{target,Y=1}(t)}{N_{target,Y=0,1}(t)} - \frac{N_{control,Y=1}(t)}{N_{control,Y=0,1}(t)}\right) * \left(N_{target,Y=0,1}(t) + N_{control,Y=0,1}(t)\right), \quad (2)$$

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 - control

Так как данный показатель относительный, он может ввести в заблуждение при интерпретации, а также не будет отражать действительность при неравных пропорция target и control. Поэтому далее опишем более интерпретируемый показатель.

Пример кривой UpLift на рисунке (рисунок 1.15).

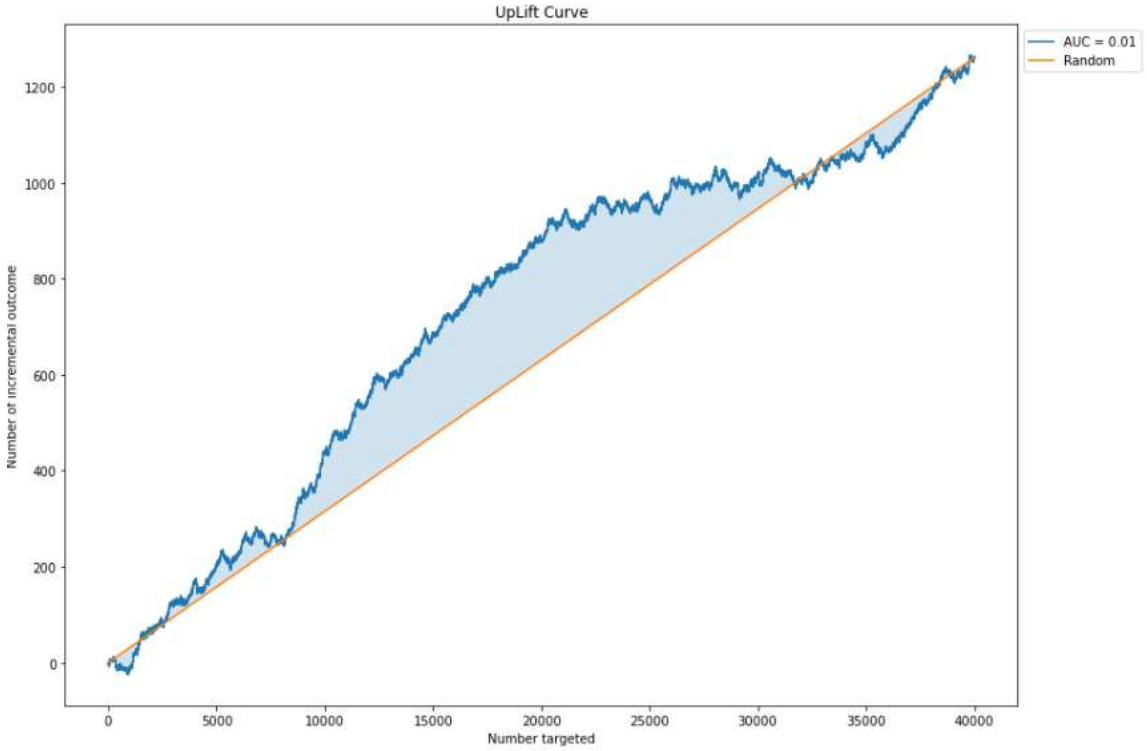

Рисунок 1.15. – Пример кривой UpLift

1.1.3 Кривая QINI

Следующую функцию, описанную в [2], можно выразить через UpLift кривую и получим определение в формуле (3):

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

$$= \left(\frac{N_{target,Y=1}(t)}{N_{target,Y=0,1}(t)} - \frac{N_{control,Y=1}(t)}{N_{control,Y=0,1}(t)}\right) * N_{target,Y=0,1}(t) = \quad (3)$$

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

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.

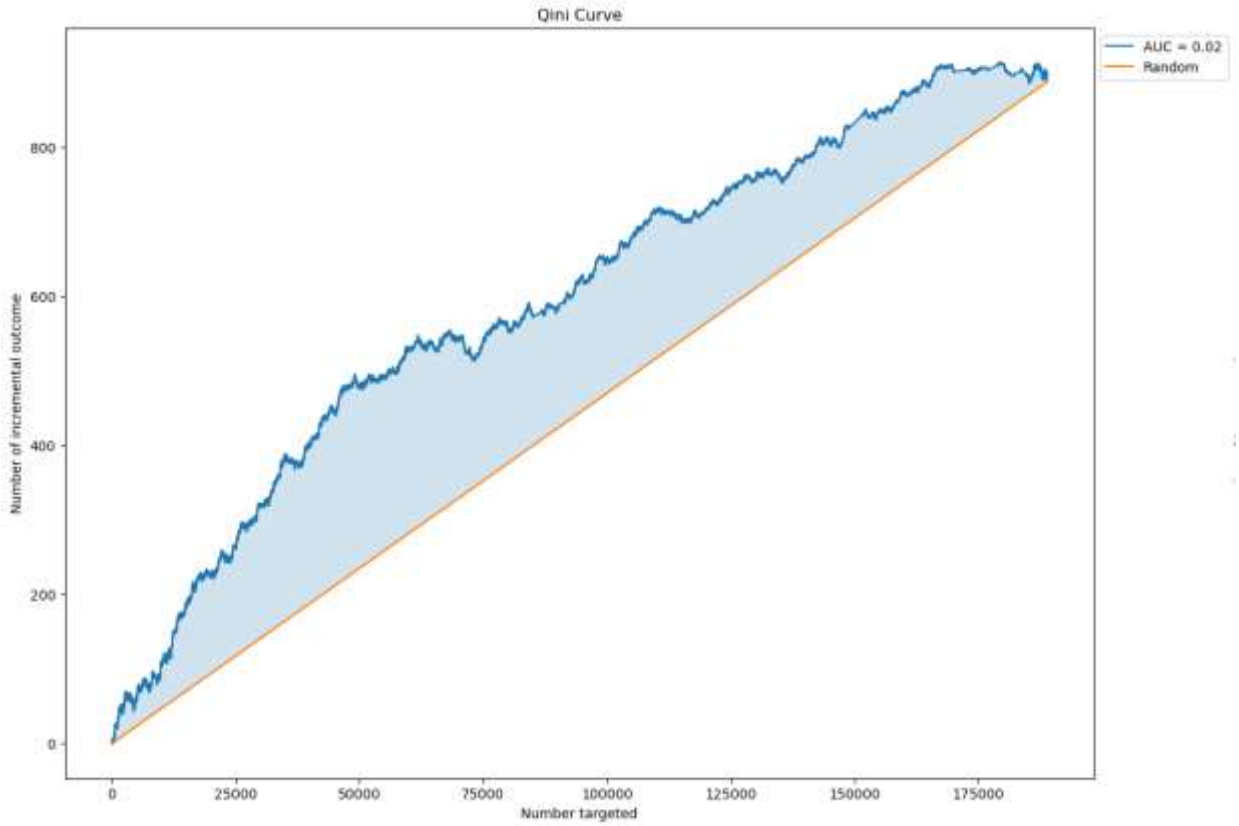

Рисунок 1.16 – Приме кривой QINI

1.2 UpLift моделирование методами машинного обучения

1.2.1 Постановка задачи UpLift

Суть UpLift моделирования в том, чтобы определить, на каких клиентов коммуникация сработает, а на каких нет. Воспользовавшись [1], определим базовые понятия.

Эффект от коммуникации определим как casual effect:

$$\tau_i = Y_i^1 - Y_i^0, \quad (4)$$

где Y_i^1- реакция i – го человека, если коммуникация была, Y_i^0- реакция, если коммуникации не было.

Зная признаковое описание i – го объекта X, можно ввести условный усредненный эффект от воздействия Conditional Average

Effect (CATE):

$$CATE(x) = M[Y_i^1 | X_i] - M[Y_i^0 | X_i] \quad (5)$$

Casual effect и CATE можно только оценить, так как одновременно невозможно провести коммуникацию с человеком и не провести. Оценка CATE и является UpLift. Тогда для конкретного объекта он имеет следующее определение:

$$UpLift(x) = M[Y_i | X_i = x, W_i = 1] - M[Y_i | X_i = x, W_i = 0], \quad (6)$$

Где Y_i – наблюдаемая реакция клиента в результате маркетинговой кампании:

$$Y_i = W_I Y_i^1 + (1 - W_i) Y_i^0 = \begin{cases} Y_i^1, \text{если } W_I = 1 \\ Y_i^0, \text{если } W_i = 0 \end{cases} \quad (7)$$

$W_i = 1$, если объект попал в целевую (threatment) группу, в которой была коммуникация,

$W_i = 0$, если объект попал в контрольную (control) группу, в которой коммуникации не было,

$Y_i = 1$, если объект совершил целевое действие,

$Y_i = 0$, если объект не совершил целевое действие

1.2.2 Метод UpLift моделирования с одной независимой моделью

Данный вариант решения из [1] использует переменную W как признак. Тогда обучающий набор данных имеет вид, приведенных в таблице 1.1.

Таблица 1.1 - Пример обучающего набора данных

Обучающие признаки				Целевая
X11	…	X1n	W1	Y1
X21	…	X2n	W2	Y2
……				…
Xm1	…	Xmn	Wm	Ym

С помощью логистической регрессии или подобной модели классификации обучаем модель на данных и после обучения находим

разность вероятностей на тестовой выборке, где в переменной W задаем везде единицы – будто бы была коммуникация, и на той же выборке обрабатываем данные, где в переменной W задаем нули – будто бы единицы не было. Тогда Uplift будет иметь вид:

$$UpLift = P\left(\begin{bmatrix} x_1^1 & \cdots & x_1^n & 1 \\ \vdots & \ddots & \vdots & \\ x_m^1 & \cdots & x_m^n & 1 \end{bmatrix}\right) - P\left(\begin{bmatrix} x_1^1 & \cdots & x_1^n & 0 \\ \vdots & \ddots & \vdots & \\ x_m^1 & \cdots & x_m^n & 0 \end{bmatrix}\right), \quad (8)$$

где P – вероятность целевого действия

1.2.3 Метод UpLift моделирования с двумя независимыми моделями

Второй подход из [1] требует уже обучения двух моделей, одна модель для экспериментальной группы – $P[Y|X = x, W = 1]$, где была коммуникация, вторая модель для контрольной группы $P[Y|X = x, W = 0]$ где коммуникации не было. После обучение моделей на тренировочных выборках, совершается обработка тестовой выборки для каждой модели и за UpLift берется так же разность двух вероятностей:

$$UpLift = P_1\left(\begin{bmatrix} x_1^1 & \cdots & x_1^n & 1 \\ \vdots & \ddots & \vdots & \\ x_m^1 & \cdots & x_m^n & 1 \end{bmatrix}\right) - P_2\left(\begin{bmatrix} x_1^1 & \cdots & x_1^n & 0 \\ \vdots & \ddots & \vdots & \\ x_m^1 & \cdots & x_m^n & 0 \end{bmatrix}\right), \quad (9)$$

где P_1 – вероятность целевого действия первой модели, а P_2 – вероятность целевого действия первой модели

1.2.4 Метод трансформации класса (задача классификации)

В данном методе из [1] мы вернемся снова к единой модели, но теперь преобразуем коммуникационную переменную и целевую переменную в одну следующим образом:

$$Z_i = Y_i * W_i + (1 - Y_i)(1 - W_i), \quad (10)$$

где Y_i-целевая переменная,W_i -коммуникационная переменная.

Тогда трансформированный класс будет иметь следующие

значения:

$$Z_i = \begin{cases} 1 \text{ при } W_i = 1;\ Y_i = 1 \\ 0 \text{ при } W_i = 0;\ Y_i = 1 \\ 0 \text{ при } W_i = 1;\ Y_i = 0 \\ 1 \text{ при } W_i = 0;\ Y_i = 0 \end{cases} \quad (11)$$

Тогда UpLift будет определяться следующим образом по формуле (12):

$$UpLift = P\left(\begin{bmatrix} x_1^1 & \cdots & x_1^{\mathrm{n}} \\ \vdots & \ddots & \vdots \\ x_m^1 & \cdots & x_{\mathrm{m}}^{\mathrm{n}} \end{bmatrix}\right), \quad (12)$$

где P – вероятность выполнения закодированного целевого действия

1.2.5 Метод трансформации класса (задача регрессии)

В данном методе мы преобразуем коммуникационную переменную и целевую переменную в одну следующим образом:

$$Z_i = Y_i * \frac{W_i - p}{p*(1-p)}, \quad (13)$$

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

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

$$Z_i = \begin{cases} 2, \text{при } W_i = 1;\ Y_i = 1 \\ 0, \text{при } W_i = 0{,}1;\ Y_i = 1 \\ -2, \text{при } W_i = 0; Y_i = 1 \end{cases} \quad (14)$$

Тогда UpLift будет определяться следующим образом по формуле (15):

$$UpLift = R\left(\begin{bmatrix} x_1^1 & \cdots & x_1^{\mathrm{n}} \\ \vdots & \ddots & \vdots \\ x_m^1 & \cdots & x_{\mathrm{m}}^{\mathrm{n}} \end{bmatrix}\right), \quad (15)$$

где R – регрессионное значение закодированного целевого действия.

1.3 Модели машинного обучения

1.3.1 Логистическая регрессия

Логистическая регрессия — это метод машинного обучения, который используется для предсказания вероятности отнесения объекта к определенному классу.

Он основан на логистической функции (16):

$$P(x, w) = \frac{1}{1+e^{-(w*x)}} \quad (16)$$

, которая принимает на вход линейную комбинацию признаков — x объекта и выдает вероятность его принадлежности к классу — $P(x)$, а w – параметры логистической регрессии .

Логистическая регрессия широко используется в задачах классификации, например, для определения того, является ли электронное письмо спамом или нет, или для диагностики заболеваний на основе медицинских данных.

В данном случае, логистическая функция будет использоваться для прогноза вероятности $P(x, w)$ – где в нашем случае это вероятность целевого действия при параметрах x = {X, T}, где X – это параметры объекта, а T – признак воздействия на объект.

Так как логистическая регрессия – модель машинного обучения с учителем, нужно оценивать качество прогноза и корректировать параметры модели для нахождения оптимального набора весов, при котором вероятность целевого действия будет иметь оценку, наиболее приближенную к истинному значению.

Функция потерь для логистической регрессии называется логистической функцией потерь (log loss) или кросс-энтропией (cross-entropy loss). Она измеряет разницу между предсказанными вероятностями классов и фактическими метками классов. Логистическая функция потерь минимизируется при обучении

модели. Формула логистической функции потерь выглядит так (17):

$$L\big(y, P(x,w)\big) = -y * log(P(x,w)) - (1 - y) * log(1 - P(x,w)) \quad (17)$$

, где y - фактическая метка класса (0 или 1), $P(x,w)$ – вероятность совершения целевого действия.

Далее, для минимизации функции ошибки логистической регрессии, определим метод оптимизации.

При обучении логистической регрессии часто используется метод градиентного спуска для оптимизации параметров модели. Он позволяет находить минимум функции потерь путем итеративного изменения параметров модели в направлении антиградиента функции потерь.

Существуют различные варианты метода градиентного спуска, такие как стохастический градиентный спуск (SGD), мини – пакетный градиентный спуск (mini – batch GD) и т.д.

SGD является наиболее распространенным методом оптимизации для логистической регрессии, так как он работает быстрее и требует меньше вычислительных ресурсов, чем другие методы оптимизации. В SGD параметры модели обновляются на каждом шаге для каждого объекта в обучающей выборке, что позволяет быстрее сойтись к оптимальному решению.

Метод оптимизации SGD (стохастический градиентный спуск) работает следующим образом:

1. Инициализируются параметры модели случайными значениями.

2. Выбирается случайный объект из обучающей выборки.

3. Вычисляется градиент функции потерь по параметрам модели на основе выбранного объекта.

4. Обновляются параметры модели в направлении антиградиента функции потерь с помощью формулы (18):

$$w = w - h * \nabla L(y, P(x,w)) \quad (18)$$

, где w – вектор параметров модели, h – шаг изменения весов

(learning rate), $\nabla L(y, P(x, w))$ – градиент функции потерь.

5. Повторяются шаги 2 – 4 для каждого объекта в обучающей выборке.

6. Повторяются шаги 2 – 5 до тех пор, пока не будет достигнут критерий остановки, например, определенное количество эпох обучения или достижение минимального значения функции потерь.

SGD обновляет параметры модели на каждом шаге, что позволяет быстрее сойтись к оптимальному решению.

В методе стохастического градиентного спуска, градиент $\nabla L(y, P(x, w))$ находится следующим образом – он определяется как вектор частных производных функции потерь по каждому параметру модели. Градиент показывает направление наискорейшего убывания функции потерь и используется для обновления параметров модели в каждой итерации оптимизации.

Для каждого объекта в обучающей выборке градиент вычисляется по формуле (19):

$$\nabla L(y, x, w) = x * (P(x, w) - y) \quad (19)$$

, где x - вектор признаков объекта.

Таким образом, для каждого объекта в обучающей выборке вычисляется градиент функции потерь, и параметры модели обновляются в направлении антиградиента.

1.3.2 Линейная регрессия

Так как для моделирования UpLift используется и метод трансформации класса с переходом к задаче регрессии – рассмотрим один из основных и известных методом машинного обучения – линейная регрессия.

Линейная регрессия - это статистический метод, используемый для оценки связи между непрерывными переменными. Это модель, которая пытается установить линейную зависимость между

зависимой переменной (таргетом) и одной или несколькими независимыми переменными (факторами).

Линейная регрессия предполагает, что зависимость между переменными может быть описана линейной функцией. Линейная функция представляет собой уравнение прямой линии в двумерном пространстве, которая может быть расширена на более высокие размерности.

Для построения модели линейной регрессии необходимо определить коэффициенты, которые лучше всего соответствуют уравнению линейной функции, связывающей зависимую и независимые переменные. Эти коэффициенты могут быть определены с помощью метода наименьших квадратов (МНК), который минимизирует сумму квадратов расстояний между фактическими значениями таргета и предсказанными значениями.

Линейная регрессия широко используется в различных областях, включая экономику, финансы, бизнес, медицину, науку о материалах и многие другие.

Регрессионная модель представляет собой следующее (20):

$$\tilde{y} = F(x, w) + \varepsilon \quad (20)$$

, где $w = \{w_0, w_1, \dots, w_k)$ – вектор весов модели, ε – случайная составляющая модели, а $F(x, w)$ определяется как (21):

$$F(x, w) = w_0 + w_1 * x_1 + w_2 * x_2 + \cdots + w_k * x_k \quad (21)$$

, где k – количество факторов линейной регресии, а x_i – факторы модели.

Наиболее распространенной функцией потерь для линейной регрессии является среднеквадратичная ошибка (Mean Squared Error, MSE), которая определяется следующим образом (22):

$$MSE = \frac{1}{N} * \sum_{i=1}^{N} (y_i - \tilde{y}_i)^2 \quad (22)$$

, где N - количество наблюдений, y_i - фактическое значение таргета

для i-го наблюдения, $\tilde{y}_i$- предсказанное значение таргета для i-го наблюдения.

Для оптимизации параметров модели линейной регрессии с использованием функции потерь в виде среднеквадратичной ошибки (MSE), рассмотрим уже упомянутый выше метод стохастического градиентного спуска, который работает следующим образом:

1. Инициализация весов модели случайными значениями или нулями.

2. Выбор случайного объекта из обучающего набора данных или пройти по всем объектам с перемешиванием.

3. Вычисление линейной комбинации признаков с использованием текущих весов (23):

$$\tilde{y} = w_0 + w_1 * x_1 + w_2 * x_2 + \ldots + w_k * x_k \quad (23)$$

4. Вычисление ошибки (разницы между предсказанным и истинным значением) для одного объекта (24):

$$error = \tilde{y}_i - y_i \quad (24)$$

5. Вычисление градиента функции потерь (среднеквадратичной ошибки) на основе одного объекта для каждого веса (25):

$$\nabla w_j = 2 * error * x_{ij} \quad (25)$$

, где j = 0, 1, ..., k (k - количество признаков)

6. Обновление весов с использованием вычисленного градиента (26):

$$b_j = b_j - h * \nabla b_j \quad (26)$$

, где h – шаг изменения параметров модели (скорость обучения).

7. Повторение шагов 2-6 заданное количество эпох (итераций по набору данных) или до сходимости.

1.3.3 Дерево решений

Алгоритм машинного обучения для построения дерева решений основан на рекурсивном разделении данных на подмножества, основываясь на значениях признаков. Ветвление происходит по определенным критериям, обычно направленным на уменьшение неоднородности данных в подмножествах. Вот общий алгоритм для построения дерева решений:

1. Инициализируем корневой узел, в котором содержатся все обучающие данные.

2. Определяем лучшее разделение данных на подмножества по индексу Джини.

3. Создаем новый узел для каждого подмножества. Если все объекты в подмножестве принадлежат одному классу или доле объектов одного класса превышает заданный порог, сделайте этот узел листовым и присвойте ему класс на основе большинства объектов в подмножестве.

4. Если узел не является листовым, повторите шаги 2 и 3 для каждого из подмножеств рекурсивно, чтобы расти вниз по каждой ветви.

5. (Опционально) Проведем "обрезку" дерева, чтобы уменьшить его сложность, удалив или сокращая ветви с наименьшим влиянием на качество дерева и недостаточным количеством узлов.

6. Если операция "обрезки" не была выполнена, то остановка происходит, когда все решения будут в листьях или если дерево достигло максимальной глубины.

Как только дерево решений полностью построено на обучающих данных, его можно использовать для классификации или регрессии на новых данных.

Добавлю, что данный алгоритм применим как для задачи классификации, так и для задачи регрессии.

Индекс Джини (или коэффициент Джини) — это мера неоднородности или степень разделения классов в задаче классификации.

Индекс Джини для задачи классификации вычисляется следующим образом (27):

$$Gini = 1 - \sum_{i=1}^{N} p_i^2 \quad (27)$$

, где p_i — это вероятность принадлежности каждого объекта в разделенном подмножестве к i-му классу из N – классов. Вероятности рассчитываются как количество объектов определенного класса, деленное на общее количество объектов в подгруппе.

Чем ниже значение индекса Джини, тем более "чистым" (с меньшей смешанностью классов) является подмножество. Значение индекса Джини лежит в диапазоне от 0 до 1, где 0 соответствует идеальному разделению (когда все объекты в подгруппе относятся к одному классу), а 1 — максимальной смешанности классов.

При построении дерева решений алгоритм будет рассматривать каждый возможный способ разделения данных по признакам и выбирать тот, который даёт наибольшее уменьшение индекса Джини.

Пример. Допустим, у нас есть подмножество данных с 100 объектами, из которых 40 объектов принадлежат к классу A, а 60 объектов к классу B. Чтобы вычислить индекс Джини для данного подмножества, сначала найдем вероятности:

$$P(A) = \frac{40}{100} = 0.4;\ P(B) = \frac{60}{100} = 0.6$$

Тогда индекс Джини будет равен:

$$Gini = 1 - (0.4^2 + 0.6^2) = 1 - (0.16 + 0.36) = 0.48$$

Критерий разбиения, используемый для построения модели дерева решений для задачи регрессии, обычно основывается на среднеквадратичном отклонении (MSE) предсказаний. Этот критерий

измеряет расстояние между реальными и предсказанными значениями целевой переменной на каждом разделении и используется для выбора наилучшего разделения в каждом узле дерева.

Формула для вычисления критерия разбиения MSE при построении дерева решений для задачи регрессии выглядит следующим образом, как уже было описано в формуле (22):

$$MSE = \frac{1}{n} * \sum_{i=1}^{n}(y_i - \tilde{y}_i)^2 \quad (22)$$

, где n - количество объектов в узле, y_i - истинное значение целевой переменной для i-го объекта, $\tilde{y}_i$ - предсказанное значение моделью для i-го объекта

Смысл этой формулы заключается в том, что мы находим среднее значение квадратов ошибок предсказаний модели на каждом разделении, что позволяет нам выбрать наилучшее разделение, для которого этот показатель будет минимальным.

При выборе наилучшего разбиения для дерева решений используется алгоритм рекурсивного разбиения. Алгоритм сначала находит лучший признак и порог разбиения, который минимизирует MSE для текущего узла. Затем данные разбиваются на две части и алгоритм продолжает работу рекурсивно для обоих под-узлов, до тех пор, пока не будет достигнут критерий останова.

2 ПРАКТИЧЕСКАЯ ЧАСТЬ

2.1 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 [6] с разбиением выборки на 5 долей. По итогу кросс валидации будут браться средние показатели качества обучения, на основе которых и будет сравнение. Иллюстрация работы кросс валидации на рисунке ниже (рисунок 2.1).

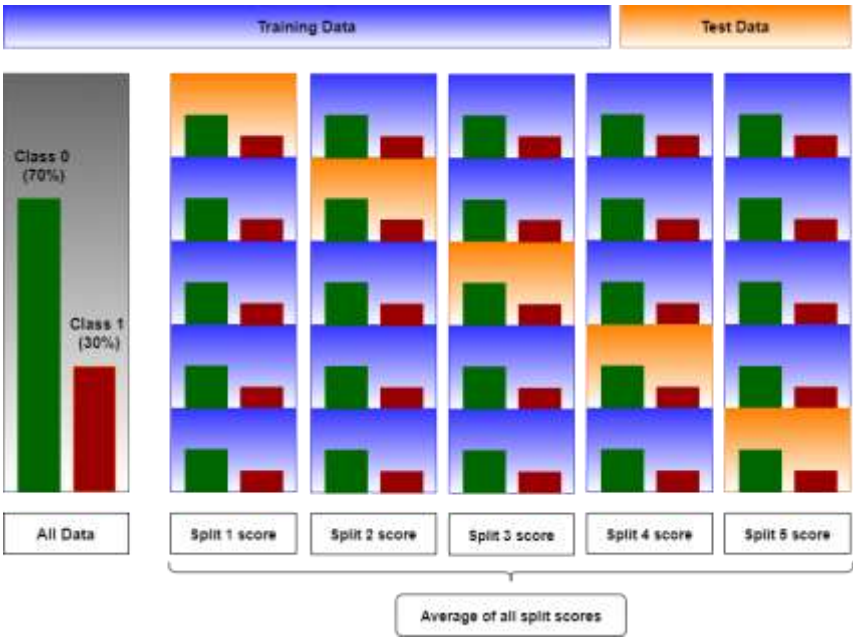

Рисунок 2.1 - Схема кросс валидации

2.8 Результаты численного эксперимента

Проведя череду экспериментов, стоит посмотреть на все результаты разом и выделить лучшее решение для данных X5-Retail (рисунок 2.14).

Номер	Структура	WAU	UpLift на k%	Qini curve AUC	UpLift curve AUC
1	Базовое решение	0,033	0,034	0,000	0,000
2	Решение с одной моделью	0,033	0,032	0,000	0,000
3	Решение с двумя независимыми моделями	0,033	0,053	0,010	0,012
4	Трансформация класса в задачу регрессии	0,033	0,044	0,006	0,006
5	Трансформация класса в задачу регрессии с поиском лучшей модели	0,035	0,070	0,024	0,034

Рисунок 2.14 - Сравнительные результаты целевых показателей качества обучения

Проведя череду экспериментов, стоит посмотреть на все результаты разом и выделить лучшее решение для собственных данных (рисунок 2.15).

Номер	Структура	$UpLift_{30\%}$	Qini curve AUC	UpLift curve AUC
1	Базовое решение	0,0073	0,0016	0,0004
2	Решение с одной моделью	0,0158	0,0223	0,0055
3	Решение с двумя независимыми моделями	0,0144	0,0167	0,0042
4	Трансформация класса в задачу классификации	0,0124	0,0081	0,0022
5	Трансформация класса в задачу регрессии	0,0138	0,0155	0,0038
6	Решение с одной моделью с поиском лучшей модели	0,0233	0,0543	0,0136
7	Трансформация класса в задачу регрессии с поиском лучшей модели	0,0179	0,0314	0,0077

Рисунок 2.15 - Сравнительные результаты целевых показателей качества обучения

Стоит заметить, что в зависимости от данных, при одних и тех же подходах машинного обучения, наилучший результат дают совершенно разные модели

Как можно заметить, для наших данных по всем показателям (рисунок 2.15) лучшая модель для наших данных – это метод моделирования с помощью одной модели – стека из ансамблей моделей классификации под номером 6.

Далее найдем экономическую выгоду нашей модели с помощью показателя $UpLift_{30\%}$, т.к. он отражает номинальный прирост доли клиентов с покупкой вы выборке реципиентов. Пусть в среднем, клиент, совершивший покупку, принесет 2 500 руб. выручки.

Изначально в нашем эксперименте участвовало 473 861 клиентов с отправкой СМС, что естественно не весь объем имеющейся базы и даже не 10% от нее. Тогда представим, что это 30% от имеющей базы для простоты интерпретации.

Из этих 473 тыс. реципиентов, покупку совершило 34 тыс., т.е. вероятность покупки примерно 0.0718 вне зависимости от объема выборки (при ее уменьшении). Наша наилучшая модель дает прирост в 0.0233. Тогда вероятность покупки с применением UpLift модели составила бы 0.0951, далее найдем экономический прирост: 0.0233 *

473861 * 2500 = 27 602 403 руб.

Таким образом, при сохранении объема расходов на отправку СМС, применение UpLift моделирования в нашем случае принесет 27.6 млн руб. дополнительной выручки при выборке в 473 861 реципиентов

ЗАКЛЮЧЕНИЕ

В данной работе предлагается обзор подходов к UpLift моделированию методами машинного обучения.

Были выбраны и описаны структуры с одной моделью машинного обучения, с двумя независимыми моделями машинного обучения и два вида трансформации класса для обучения одной модели машинного обучения классификации и регрессии.

Численные результаты эксперимента показали, что наилучшего UpLift по показателям качества обучения можно добиться с помощью автоматического подбора моделей задачи классификации и последующим применением ее в алгоритме с одной независимой моделью.

Найденный алгоритм, возможно, будет наилучшим только для рассматриваемых в задаче данных, так как в зависимости от скрытой природы зависимостей обучающих признаков, различные структуры могут показывать наилучшие результаты на одних данных и наихудшие на других.

В работе приведены обзоры на различные способы решения проблемы и полученные результаты в перспективе могут быть аналогичны и для остальной клиентской базы ретейл компании косметики и парфюмерии.

Дальнейшая работа в аспирантуре по данной теме нацелена на преобразование описанных выше методов машинного обучения под задачу UpLift, где параметр воздействия будет выступать уже не в роли обучающего признака, а в виде целевой переменной, наряду с признаком выполнения целевого действия.

Подходы к трансформации функции активации и целевой функции для обучения описанных методов будут взяты из нескольких статей, будут написаны вычислительные модули и с помощью обучения на реальных или иных других данных в открытом источнике. С помощью сравнения, будет наглядно показана практическая польза такого подхода, как например в [5].

Помимо трансформации уже стандартных моделей машинного

обучения, будет разработан подход к оценке не UpLift как инкремента вероятности позитивного воздействия на клиента, а будет рассмотрен подход к финансовой оценке, как описано в [6].

ЛИТЕРАТУРА

[1] Gutierrez P., G´erardy J. Causal Inference and Uplift Modeling A review of the literature // PMLR – 2016 - URL: https://proceedings.mlr.press/v67/gutierrez17a/gutierrez17a.pdf

[2] Weijia Zhang, Jiuyong Li, Lin Liu A unified survey of treatment effect heterogeneity modelling and uplift modelling // arXiv – 2021 – URL: https://arxiv.org/pdf/2007.12769

[3] Devriendt F., Guns T., Verbeke W. LEARNING TO RANK FOR UPLIFT MODELING // arXiv – 2020 – URL: https://arxiv.org/pdf/2002.05897

[4] Nyberg O., Kussmierczyk T., Klami A. Uplift Modeling with High Class Imbalance // PMLR – 2021 – URL: https://proceedings.mlr.press/v157/nyberg21a/nyberg21a.pdf

[5] Mouloud Belbahri, Alejandro Murua, Olivier Gandouet, Vahid Partovi Nia. A TWIN NEURAL MODEL FOR UPLIFT // arXiv – 2021 – URL: https://arxiv.org/pdf/2105.05146

[6] Robin Gubela, Stefan Lessmann, Johannes Haupt, Annika Baumann, Tillmann Radmer, Fabian Gebert. Revenue Uplift Modeling // ResearchGate 2017 – URL: https://www.researchgate.net/publication/321729653

[7] RF – сегментация – URL: https://www.moengage.com/blog/rfm-analysis-using-rfm-segments/

[8] Гудфеллоу Я., Бенджио И., Курвилль А. Глубокое обучениХ`е // пер. с анг. А. А. Слинкина. – 2-е изд., испр. – М.: ДМК Пресс – 2018. – 652

[9] Туториал по uplift моделированию. Часть 1 – URL: https://habr.com/ru/companies/ru_mts/articles/485980/

[10] Курс по uplift – URL: https://ods.ai/tracks/uplift-modelling-course

[11] Введение в Uplift – URL: https://newtechaudit.ru/vvedenie-v-uplift-modelirovanie/

[12] Продвинутые методы Uplift-моделирования – URL: https://habr.com/ru/companies/glowbyte/articles/686398/

[13] Ян Лекун. Как учится машина. Революция в области нейронных сетей и глубокого обучения // пер. с анг. Е. Арсеновой - © ООО «Альпина ПРО» - 2021

[14] Mouloud Belbahri, Alejandro Murua, Olivier Gandouet, and Vahid Partovi Nia. Qini-based uplift regression // arXiv – 2019 – URL: arXiv:1911.12474, 2019.

[15] StefanWager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests // arXiv – 2015 – URL: arXiv:1510.04342, 2015.