Санкт-Петербургский Государственный Университет

Факультет прикладной математики-процессов управления

Коган Даниил Вячеславович

Корреляция оттока клиентов оператора связи с предоставляемым качеством интернет-соединения

Выпускная квалификационная работа

Направление 01.04.02

Прикладная математика и информатика

Магистерская программа Математическое и информационное обеспечение экономической деятельности

Научный
руководитель
доктор физ.-мат. наук,
профессор
Прасолов А.В.

Санкт-Петербург 2019

Содержание

1 Введение

В данной работе рассматривается проблема оттока клиентов от оператора фиксированной связи, а также связь оттока с качеством предоставляемого оператором интернет-соединения.

Задача предсказания оттока активно решается в самых разных предметных областях, например, в банкинге, телекоммуникациях, розничной торговле. Необходимо успевать отслеживать переходы клиентов банка в другие банки или переходы клиентов одной телекоммуникационной компании в другие и вовремя проводить методики по удержанию таких клиентов.

Эта задача является экономически обоснованной, потому что для большинства сервисов справедливо следующее замечание: чем больше клиентов есть у сервиса, тем лучше он монетизируется и тем больше приносит прибыли. Следовательно, нам всегда выгодно иметь большое количество пользователей.

Достичь этой цели можно следующими способами: во-первых, количество пользователей может возрасти вследсвтие их активного привелчения, и мы имеем постоянный приток, во-вторых, пользовательская база может расти как следствие удержания старых пользователей, то есть, когда мы сокращаем отток пользовательской базы. Более того, когда мы удерживаем клиента, мы понимаем, какого именно пользователя мы удерживаем, здесь важно удерживать ключевых клиентов, которые приносят сервису много денег. С другой стороны, когда мы привлекаем одного нового пользователя, мы никогда не знаем, как сложится его взаимодействие с сервисом и перейдет ли он, скажем, в группу лояльных пользователей. Таким образом, задача удержания важна еще с этой точки зрения.

Кажется, что нам выгодно удерживать всех пользователей, которые у нас

есть. С другой стороны, процесс удержания — не бесплатный. Для того чтобы пользователя удержать, нужно в это инвестировать средства. Соответственно, удерживать абсолютно всех пользователей мы можем далеко не всегда.

Для того чтобы эффективно управлять процессом удержания, нам нужно уметь адресно удерживать тех пользователей, которые, с одной стороны, нам важны, а с другой стороны, тех пользователей, которые действительно склонны к оттоку. Плюс нужно учитывать тот факт, что процесс удержания происходит не мгновенно — нам нужно уметь выделять пользователей, склонных к оттоку, до того, как они приняли решение о том, чтобы от нас уйти, и непосредственно это сделали.

Причиной оттока могут являться различные факторы. Я предполагаю, что одним из значимых факторов оттока является недовольство абонентов качеством мобильного интернета. Интуитивно кажется, что наряду с качеством связи, на решение клиента о переходе от одного оператора к другому будет влиять и то, насколько они довольны предоставляемым мобильным интернетом, так как большинство абонентов пользуются интернетом на регулярной основе. Соответственно, недовольный абонент может начать поиски другого оператора, который лучше удовлетворяет его потребности и, возможно, найдет такого и перейдет к нему. Цель данной работы - выяснить, есть ли зависимость между оттоком клиентов у мобильного оператора с качеством, путем построения классификационной модели.

В рамках выполнения задачи построим классификационную модель с предсказанием оттока. Она будет подтверждать правильность сделанных нами выводов. Модель будем строить на данных, отражающих качество интернет-соединения. Дополнительной сложностью в данной задаче будет являться то, что абонент фиксированного оператора представляет из себя не одно устрой-

ство, а точку доступа, имеющую множество устройств в качестве постоянных пользователей сети. В работе будет представлен способ, с помощью которого можно будет определять принадлежит ли данное устройство хозяину точки доступа или нет (является ли оно постоянным пользователем). Это важно, потому что мы можем ошибочно полагать оттоком абонентов, у которых одиночные устройства сменили постоянную точку доступа, но другие при этом остались. Например, это может произойти, когда один из членов семьи переезжает, или когда устройство продают другому владельцу. Также, если бы мы не проводили никакую фильтрацию, то могли бы ошибочно полагать оттоком абонентов, у которых переехали все устройства и стали пользоваться новой сетью, а данная точка доступа бы при этом осталась. Такое может происходить, когда квартира сдается, и у нее периодически могут меняться владельцы. Вышеуказанные случае необходжимо обнаруживать и не допускать их попадания в ложный отток.

Сами операторы не имеют возможности построить такую модель в связи с неимением таких данных.

2 Список использованных источников

1. Cosimo Birtolo, Vincenzo Diessa, Diego De Chiara, and Pierluigi Ritrovato, Customer Churn Detection System: Identifying Customers Who Wish to Leave a Merchant, 2013

   Для решения проблемы оттока клиентов в этой статье предлагается система, способная определить отток по поведению клиента и помочь донести специальные предложения до таких клиентов.

2. URL: https://docs.microsoft.com/en-us/azure/machine-learning/studio/azure-ml-customer-churn-scenario

   В этой статье разобран пример построения системы определения оттока на Azure Machine Learning.

3. Matthew Kirk, Thoughtful Machine Learning with Python: A Test Driven Approach

   В этой книге описываются различные алгоритмы машинного обучения с пркатическими примерами использования.

4. Dipanjan Sarkar, Raghav Bali, Tushar Sharma, Practical Machine Learning with Python, 2018

   В данном пособии описаны алгоритмы моделей, а также процесс предобратботки данных для последующего их использования в анализе.

5. URL: http://xgboost.readthedocs.io/en/latest/model.html

   Подробное описание модели и документация к библиотеке XGBoost.

6. L. Mason, J. Baxter, P. Bartlett, M. Frean, Boosting Algorithms as Gradient Descent

В данной статье описываются методы бустинга, а также присутствуют теоремы, доказвающие их сходимость к оптимальным решениям.

7. Aleksandar J. Petkovski, Biljana L. Risteska Stojkoska, Kire V. Trivodaliev, and Slobodan A. Kalajdziski, Analysis of Churn Prediction: A Case Study onTelecommunication Services in Macedonia

   В статье были проанализированы данные об оттоке телекомуникационной индустрии, обозначены методологии предсказания оттока. Также была описана предобработка и очистка данных и проведен сравнительный анализ нескольких алгоритмов предсказания оттока. Данные рассматривались за 1 год по 22461 абоненту, из которых 2629 ушли в отток за это период.

8. Essam Shaaban, Yehia Helmy, Ayman Khedr, Mona Nasr, A Proposed Churn Prediction Model

   В данной статье были рассмотрены аналогичные с [7] методы предсказания оттока. Также был представлен полный список признаков, использованных в обучающей выборке.

9. Uri Laserson, Sandy Ryza, Sean Owen, Josh Wills, Advanced Analytics with Spark

   Примеры построения моделей с помощью Spark на языке Scala. Разбор библиотеки Spark MLlib.

10. Karau H., Warren H. High Performance Spark: Best Practices for Scaling and Optimizing Apache Spark

    Настройка и оптимизация кластера Spark.

11. Noel Welsh, Dave Gurnell, Essential Scala

Книга по программированию на Scala.

12. Paul Chiusano, Runar Bjarnason, Functional Programming in Scala Функицональное программирование в языке Scala.

3 Сбор и обработка данных и релизация модели оттока

3.1 Процесс решения задачи

Шаги для достижения цели:

1. Развертка вычислительного кластера Spark для обработки и хранения данных, которые будут использоваться для обучения модели. Настроить переток данных из Cassandra в Spark, используя Spark API для Scala. В качестве языка реализации я выбрал именно Scala (на ней написан и сам Spark) для удобства как написания комплексных процедур по обработке данных так и взаимодействия с хранилищами данных

2. После того как кластер настроен и готов к работе, и программы для выгрузки данных из Cassandra реализованы, необходимо составить набор данных для обучения и валидации модели. Набора данных будет три: тренировочный, валидационный и тестовый. На тренировочном будем настраивать внутренние параметры модели, на валидационном - гиперпараметры, а тестовый в обучении участвовать не будет - его мы будем использовать для оценки качества финальной модели.

3. Когда данные разделены на три выборки, их нужно разметить, зафиксировать факт оттока или его отсутствие, а именно каждой строке (вектору признаков) сопоставить метку ушел ли клиент или нет.

3.2 Разметка данных для модели

Теперь можно задаться вопросом о том, как же именно нужно разметить набор данных после его формирования. Сделаем разметку следующим образом: Итак, для каждого абонента у нас будет временной ряд, который будет отражать всю активность абонента, на протяжении всего периода пользования услугами конкретного оператора связи. Если быть точным, то у нас будет не идентификатор абонента, а уникальный идентификатор устройства, но это не меняет сути. Да, возможно, данное устройство перестанет пользоваться услугами оператора просто потому что оно было продано, но такие случаи нас не интересуют, потому что в таких ситуациях оператор не может ни на что повлиять. Поэтому в дальнейшем будем пренебрегать такими событиями.

1. Зафиксировать событие оттока, если оно есть. То есть прекратил ли абонент свою активность в рамках выбранного временного промежутка.

2. Сагрегируем все признаки за временной промежуток активности абонента, выбрав правила агрегации.

3. Разметить на "уходящий"/"в норме в зависимости от того, присутствует ли впереди на расстоянии N дней от границы активности абонента событие оттока. Здесь N - горизонт прогнозирования. В данной работе за горизонт прогнозирования примем величину, равную одному месяцу (31 день), в частности чтобы исключить случаи, когда абонент оформляет договор на тестовый период использования с другим оператором, но потом возвращается назад к своему (по истечении срока предоставления услуг, который обычно измеряется в месяцах).

4. Сбалансируем классы для решения в задачи классификации. То есть, сде-

лаем так, чтобы малое количество абонентов, уходящих в отток, которые содержатся в выборке минимально сказывалось на моделировании. О том, зачем нужно, чтобы в обучении классификатора присутствовало всех классов примерно в равной пропорции, поговорим позже.

Теперь данные размечены, и можно приступать к построению и настройке модели.

3.3 Алгоритм модели

Предстоит решить задачу нахождения функции, которая определяет в пространстве признаков правило разбиения их на два класса - абоненты, которые собираются уходить, и которые этого делать не планируют за фиксированный период времени. Сначала рассмотрим задачу попроще, а потом расширим ее. Будем решать задачу восстановления функции в общем контексте обучения с учителем. У нас будет набор пар признаков x и целевых переменных y, $(x_i, y_i)_{i=1,\dots,n}$, на котором мы будем восстанавливать зависимость вида $y = f(x)$. Восстанавливать будем приближением $\hat{f}(x)$, а приближение будем выбирать по функции потерь $L(y, f)$:

$$y \approx \hat{f}(x),$$

$$\hat{f}(x) = \arg\min_{f(x)} L(y, f(x))$$

Для решения задачи ограничимся семейством функций $f(x, \theta), \theta \in \mathbb{R}^d$. Тогда задача сводится к уже решаемой:

$$\hat{f}(x) = f(x, \hat{\theta})$$

$$\hat{\theta} = \arg\min_{\theta} L(y, f(x, \theta))$$

Расширим задачу, теперь она будет состоять в том, чтобы итеративно искать приближения $\hat{f}(x)$, где

$$\hat{f}(x) = \sum_{i=0}^{M} \hat{f}_i(x)$$

Здесь M - количество итераций. На каждой итерации мы будем строить модель по определенному правилу.

Таким образом мы хотим аппроксимировать исходную функцию $\hat{f}(x)$, в контексте нашей задачи являющуюся законом разделения классов уходящих и не уходящих абонентов в пространстве признаков. Чтобы решить задачу, ограничим поиск семейством функций $\hat{f}(x) = h(x, \theta)$. На каждой итерации нам также потребуется подбирать оптимальный коэффициент $\rho \in \mathbb{R}$. Для шага t задача выглядит следующим образом:

$$\hat{f}(x) = \sum_{i=0}^{t-1} \hat{f}_i(x),$$

$$(\rho_t, \theta_t) = \arg\min_{\rho,\theta} L(y, \hat{f}(x) + \rho h(x, \theta)),$$

$$\hat{f}_t(x) = \rho_t h(x, \theta_t)$$

Здесь $L(y, f(x, \theta))$ - функция потерь.

Теперь, зная выражение градиента функции потерь, мы можем обучать каждую последующую модель так, чтобы она двигалась в обратном от градиента направлении. То есть будем с помощью МНК выправлять предсказания по этим остаткам минимизируя квадрат разности между псевдо-остатками r и предсказаниями. Для шага t итоговая задача выглядит следующим образом:

$$\hat{f}(x) = \sum_{i=0}^{t-1} \hat{f}_i(x),$$

$$r_{it} = -\Big[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\Big]_{f(x)=\hat{f}(x)}, i = 1, \ldots, n,$$

$$\theta_t = \arg\min_{\theta} \sum_{i=1}^{n} (r_{it} - h(x_i, \theta))^2,$$

$$\rho_t = \arg\min_{\rho} \sum_{i=1}^{n} L(y_i, \hat{f}(x_i) + \rho h(x_i, \theta_t))$$

В качестве функции потерь для посторения каждой модели из ансамбля будем использовать логистическую функцию потерь (поскольку мы решаем задачу классификации):

$$L = \sum_i [y_i ln(p_i) + (1 - y_i) ln(1 - p_i)],$$

где y_i - метки классов в выборке, $p_i = \dfrac{1}{1 + e^{-\hat{y}_i}}$, а $\hat{y}_i$ - значения, предсказанные моделью.

Покажем, как мы будем считать производную от функции потерь.

$$L = \sum_i \left[y_i ln\Big(\frac{1}{1 + e^{-\hat{y}_i}}\Big) + (1 - y_i) ln\Big(\frac{e^{-\hat{y}_i}}{1 + e^{-\hat{y}_i}}\Big)\right]$$

Найдем $\dfrac{\partial L}{\partial \hat{y}_i}$:

$$\frac{\partial L}{\partial \hat{y}_i} = \frac{y_i - (1 - y_i)e^{\hat{y}_i}}{1 + e^{\hat{y}_i}}$$

Умножим на $\dfrac{e^{-\hat{y}_i}}{e^{-\hat{y}_i}}$:

$$\frac{\partial L}{\partial \hat{y}_i} = \frac{y_i e^{-\hat{y}_i} - 1 + y_i}{1 + e^{\hat{y}_i}} = \frac{y_i(1 + e^{-\hat{y}_i})}{1 + e^{-\hat{y}_i}} - \frac{1}{1 + e^{-\hat{y}_i}} = y_i - p_i$$

В качестве базовых моделей, которые будут формировать ансамбль, возьмем деревья решений CART. В алгоритме CART каждый узел дерева решений имеет двух потомков. На каждом шаге построения дерева правило, формируемое в узле, делит заданное множество примеров (обучающую выборку) на две части – часть, в которой выполняется правило и часть, в которой прави-

ло не выполняется. Для выбора оптимального правила используется функция оценки качества разбиения.

Оценочная функция, используемая алгоритмом CART, базируется на интуитивной идее уменьшения нечистоты (неопределённости) в узле. В алгоритме CART идея "нечистоты"формализована в индексе Gini. Если набор данных T содержит данные n классов, тогда индекс Gini определяется как:

$$Gini(T) = 1 - \sum_{i=1}^{n} p_i^2$$

где p_i – вероятность (относительная частота) класса i в T.

Если набор T разбивается на две части T_1 и T_2 с числом примеров в каждом N_1 и N_2 соответственно, тогда показатель качества разбиения будет равен:

$$Gini_{split}(T) = \frac{N_1}{N} Gini(T_1) * \frac{N_2}{N} Gini(T_2)$$

Наилучшим считается то разбиение, для которого $Gini_{split}(T)$ минимально. Обозначим N — число примеров в узле — предке, L, R — число примеров соответственно в левом и правом потомке, l_i и r_i — число экземпляров i-го класса в левом/правом потомке. Тогда качество разбиения оценивается по следующей формуле:

$$Gini_{split} = \frac{L}{N} * \left(1 - \sum_{i=1}^{n} \left(\frac{l_i}{L}\right)^2\right) + \frac{R}{N} * \left(1 - \sum_{i=1}^{n} \left(\frac{r_i}{R}\right)^2\right) \to min$$

Чтобы уменьшить объем вычислений формулу можно преобразовать:

$$Gini_{split} = \frac{1}{N}\left(L * \left(1 - \sum_{i=1}^{n} \left(\frac{l_i}{L}\right)^2\right) + R * \left(1 - \sum_{i=1}^{n} \left(\frac{r_i}{R}\right)^2\right)\right) \to min$$

Так как умножение на константу не играет роли при минимизации:

$$Gini_{split} = L - \frac{1}{L} * \sum_{i=1}^{n} l_i^2 + R - \frac{1}{R} * \sum_{i=1}^{n} r_i^2 \to min$$

В итоге, лучшим будет то разбиение, для которого величина максимальна. Таким образом, при построении «дерева решений» по методу CART ищется такой вариант ветвления, при котором максимально уменьшается значение показателя $Gini_{split}$

Для нашей модели мы будем сильно ограничивать глубину дерева, желаем получить на выходе ансамбль из неглубоких деревьев. То есть мы будем устанавливать лимит по глубине, после чего прекращать построение дерева.

Таким образом, на каждой итерации образуется новая модель (ее можно трактовать как направление наискорейшего спуска в классическом градиентном спуске). После чего мы прибавляем эту модель, умноженную на коэффициент, который минимизирует функцию потерь всего ансамбля. То есть, мы итеративно улучшаем нашу аппроксимирующую модель новыми.

3.4 Используемые инструменты

1. Cassandra: База данных, в которой хранится нужная для анализа информация. Cassandra - децентрализованное отказоустойчивое хранилище. Кластер состоит из узлов, по которым разбито содержимое базы. Преимущества кассандры:

   - быстрые запись и чтение
   - децентрализация и репликация данных, что влечет отказоустойчивость. Когда узел выходит из строя данные не теряются, потому что в кластере еще есть их копии Также можно обновлять БД без даунтайма (каждый узел отдельно),
   - размер кластера масштабируется по денежным затратам линейно

   На больших объемах данных она гораздно эффетивнее реляционных БД, так как, во-первых, для каждого запроса нам не нужен доступ ко всем узлам кластера, а только лишь к отедльным участкам некоторых узлов, во-вторых, мы можем взаимодействовать со всеми узлами параллельно. Запись и чтение происходят гораздо быстрее. Для составления модели нам нужно много данных, а также обучать модель на этих данных много раз для качественной ее настройки. Для этого настраиваем отдельно Spark-кластер, в который переместим необходимые данные из кассандры.

2. Spark - среда для реализации распределенной обработки данных. Преимущества:

   - Будем иметь более быстрый доступ к данным, храня их в оперативной памяти

- Хранение данных в виде Spark-абстракций (RDD, Dataframe, Dataset), с помощью которых сможем оперировать набором данных для обучения как единым логическим целым, независимо от того как эта коллекция данных разбросана по узлам кластера физически.
- Масштабируемый код. Одинаково пригоден для любых размеров данных
- Через Scala сможем наследовать классы абстракций и писать свои методы для их обработки, если такие понадобятся
- Библиотека Spark MLlib, API которой доступен для 4 популярных языков. В ней реализованы необходимые методы для анализа данных Таким образом мы сможем достаточно быстро обучать модель, а следовательно исполнить большее количество итераций для ее подробной настройки.

3. Для модели будем использовать XGBoost, его ключевые особенности:

- Регуляризация. Классическая реализация градиентного бустинга GBM (Gradient Boosting Machine) ее не содержит, в отличии от XGBoost. Регуляризация вводит штраф за сложность модели, потому что сложная модель склонна к нахождению не только общих закономерностей, присущих генеральной совокупности, из которой берется выборка, но и к запоминанию закономерностей, специфичных конкретно для данной обучающей выборки.
- Парраллелизация построения дерева. Обучение происходит гораздо быстрее, если распределить вычеслительную нагрузку на несколько ядер процессора.
- Гибкость в плане выбора функции потерь.

- Встроенные методы для обработки пропущенных значений.
- Прунинг деревьев. Заключается в искуственном ограничении количества узлов в деревьях для уменьшения склонности к переобучению.
- Встроенная кросс-валидация (скользящий контроль). Является процедурой эмпирического оценивания обобщающей способности алгоритмов. Фиксируется некоторое множество разбиений исходной выборки на две подвыборки: обучающую и контрольную. Для каждого разбиения выполняется настройка алгоритма по обучающей подвыборке, затем оценивается его средняя ошибка на объектах контрольной подвыборки. Оценкой скользящего контроля называется средняя по всем разбиениям величина ошибки на контрольных подвыборках.

  Если выборка независима, то средняя ошибка скользящего контроля даёт несмещённую оценку вероятности ошибки. Это выгодно отличает её от средней ошибки на обучающей выборке, которая может оказаться смещённой (оптимистически заниженной) оценкой вероятности ошибки, что связано с явлением переобучения.

3.5 Реализация определения оттока

Опишем реализацию определения оттока. Так как мы определеяем отток для фиксированного оператора, то будем это делать исходя из набора устройств, которые появлялись в домашней сети абонента. Здесь появляется несколько способов определения, о которых будет рассказано далее.

Среди всех устройств, побывавших в домашней сети конкретного абонента отфильтруем те, которые находятся в ней регулярно. Это можно сделать, например, исходя из общего числа дней активности. Основная идея заключается в том, что мы будем отслеживать устройства абонента через их сессии в сетях провайдеров сотового и фиксированного интернета. Теперь подробно рассмотрим данные, которые мы используем для определения оттока.

Для каждой сессии имеем множество различных данных о ней, из которых нам понадобятся следующие: временные координаты сессии, IP-адреса устройств, MAC-адреса точек доступа, соты, к которым устройство было прикреплено во время интернет-сессии.

С помощью IP-адресов мы как раз сможем определить, из сети какого оператора проходила сессия. Смена наиболее используемого провайдера будет являться основным индикатором перехода абонента от одного оператора к другому. Далее был построен алгоритм определения оттока, основывающийся исключительно на данных об IP. Он дал неудовлетворительный результат, потому что не включал переезды абонентов с места на место. Во-первых, необходимо было отсеять людей, снимающих квартиры и меняющих одно съемное жилье на другое, потому что по факту в квартире независимо от съемщика оставался один и тот же провайдер. Во-вторых, нужно отсеять в принципе переезжающих людей, потому что их нельзя отнести к контролируемому со стороны провайдера оттоку.

MAC-адреса точек доступа могли бы полностью решить проблему определения оттока, если бы у нас имелось достаточное количество данных о них. Сначала я пробовал составить алгоритм, основываясь именно на этих данных. Для каждой точки доступа составлялась статистика использования, и присваивался оператор, которым на данном MAC-адресе пользовались, а также фиксировалась смена оператора, если она была. Дальше, основываясь на наборе устойств, находившихся на точке доступа, устнавливалась принадлежность точки конкретному абоненту. Те точки, на которых происходила смена используемого оператора, относим к оттоку. В процессе оттока абонент может сменить MAC-адрес, поэтому необходимо отслеживать и это. Здесь приходится учитывать, были у абонента как минимум 2 точки доступа, непересекаемые по времени использования. Дополнительно нужно отслеживать, чтобы географические координаты этих точек совпадали. Иначе можем неправильно опознать случай, когда, например, абонент переехал из одной съемной квартиры в другую. При этом на старой квартире активность может прекратиться, потому что новый арендатор туда еще не заселился. Поэтому, конечно, такой случай оттоком не считается. В итоге такой подход оказался нежизнеспособным, потому что не покрывал значительную часть случаев оттока из-за отсутствующих данных.

Было решено вернуться к изначальному способу определения оттока, с добавлением в рассмотрение геолокации устройств абонента. В финальном варианте алгоритма для всех устройств как и прежде будем смотреть на подневную статистику интернет-сессий с разбивкой по операторам. Будем искать период стабильности, на протяжении которого устройство пользовалось услугами нашего оператора, а также период, на протяжении которого устройство стабильно пользовалось услугами другого оператора. В качестве периода

для фиксирования оттока будем брать месяц, чтобы убрать из рассмотрения случаи, когда люди пользовались пробным периодом у другого оператора. То есть, если устройство абонента стабильно пользовалось услугами нового оператора на протяжении месяца, при этом не меняя своего места проживания, то мы полагаем, что устройство ушло.

В следующем параграфе рассмотрим, как исходя из наших данных мы можем понять, сменилось ли место проживания абонента или нет.

3.6 Геолокация устройств абонента

Итак, для интернет-сессий, проходивших с фиксированных и мобильных операторов, имеем данные об активной и о соседних точках доступа, сигнал которых удалось поймать, а также о соте, к которой устройство было подключено на момент сессии, если это сотовое устройство. Для случаев, когда у нас есть информация о точках доступа, будем определять локацию по их положению, а местоположение сот будем использовать только в случае отсутсвия этих данных. Причина такого решения заключается в том, что WiFi-позиционирование работает точнее позиционирования по сотам. Для этого было проведено отдельное исследование, в котором эта гипотеза подтвердилась.

Таким образом, для каждого устройства имеется множество точек (сот или точек доступа WiFi), из которых необходимо выделить только те, которые входят в окресность места проживания владельца устройства.

Для этого я кластеризовал точки и в качестве домашней координаты выбрал центр кластера, к которому принадлежит наиболее популярная по количеству встреч в данных точка доступа (или сота). Для кластеризации выбрал алгоритм Mean Shift (Сдвиг среднего значения), так как кластеры в нем не захватывают точки, находящиеся дальше наперед заданного окна, потому что такие точки могут сильно смещать центр кластера.

После вычисления координат места жительства владельцев устройств мы ищем те устройства, которые мы подозреваем, что ушли в отток, и проверяем, что их место жительства не сменилось. Дополнительно смотрим, чтобы среди устрйоств данного абонента не было тех, который в отток не ушли. После чего формируем выборку ушедших в отток абонентов.

Вышеописанный подход дал результат, с которым стало возможно рабо-

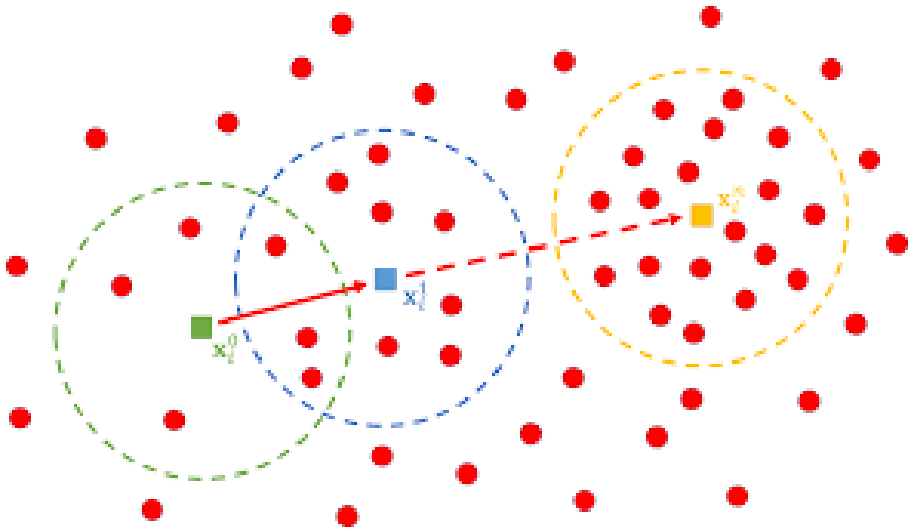

Рис. 1: Пример работы Mean Shift кластеризации

тать дальше, а именно размечать обучающую выборку и составлять признаки.

3.7 Подготовка данных

Теперь можно приступать к организации обучающей выборки. В нее будем включать параметры качества, такие как скорость и время первой буферизации во время видео, количество дропов (прерываний просмотра) за какой-либо промежуток времени и многие другие параметры качества. Также необходимо включить характеристики модели устройства, которой пользуется абонент, сделать завязку на географические координаты.

Опишем схему составления обучающей выборки. Будем собирать параметры качества по устройствам абонента.

Перейдем непосредственно к построению модели. Прежде всего мы начнем с построения набора данных для обучения и оценки качества модели.

Первое, о чем стоит упомянуть, это то, что мы можем работать с данными разного типа. Это могут быть как числовые данные, так и порядковые и категориальные. Порядковые данные также являются нечисловыми данными как и категориальные, но на них мы можем установить отношение порядка.

Важно понимать следующее: когда мы работаем с данными разного типа, нам нужно убедиться в том, что те инструменты, которые мы используем, мы используем правильно. То есть мы не используем категориальные данные в качестве числовых. Если мы все-таки используем номинальные или порядковые данные, мы их обрабатываем соответствующим образом.

Также в наших данных так или иначе часто будут присутствовать временные ряды, потому что когда мы говорим о поведении пользователей, мы говорим о том, как пользователи взаимодействуют с нашим сервисом в течение некоторого времени. Присутствующие в наших данных временные ряды мы заменим некоторыми характеристиками, которые мы из них можем извлечь. Поскольку объем данных составляет 3 месяца, то здесь особо не смыс-

ла искать какие-либо длительные тренды, вследствие которых клиент мог бы принимать решения о переходе. В данном варианте достаточно будет достать ключевые характеристики распределения показателя – например, его среднее, квантили распределения, а также доли наблюдений ниже порогового значения с подбором различных значений порогов. Таким образом, мы сможем подать модели информацию о распределении показателей качества.

Признаки качества будут описывать характеристики просмотра видео абонента – данные о средней скорости загрузки видео, его длительности, отдельно соберем такие характеристики касательно первой буферизации. Мы предполагаем, что очень важно, чтобы абонент не испытывал проблем при первой буферизации видео, потому что именно в этот период пользователь ждет начала проигрывания видео и в этот момент во многом формируется впечатление о качестве, что впоследствии становится одним из факторов перехода абонентов от одного оператора к другому.

Каждому измерению сопоставим качество, агрегированное за несколько разных по длительности временных промежутков. Например, конкретному моменту времени соотнесем усредненные метрики качества за неделю, месяц и так далее.

Добавим еще в признаки информацию о распределении величин. Можно создать несколько порогов для параметра и считать доли наблюдений ниже порогового значения или сохранять значения квантилей.

Далее заметим некоторые важные вещи о самой выборке. Часто целевой класс может составлять единицы или даже десятые доли процентов от всей выборки. Такая несбалансированность классов может негативно сказаться на модели классификации. Во-первых, может не хватить данных для настройки на целевой класс. Это может привести к тому, что объекты не будут клас-

сифицироваться к этому классу, потому что ошибка при их классификации может оказаться незначительной для модели. Такие проблемы часто возникают на практике. Важно заметить это в процессе обучения модели, и важно выбирать такие метрики, которые будут к этому чувствительны, и решать эту проблему в процессе построения модели.

Существует множество стратегий, предлагающих варианты борьбы с данной проблемой.

Идея одного из подходов заключается в том, чтобы переназначить веса объектам целевого класса так, чтобы скомпенсировать их малое количество. Таким образом, скомпенсируем количество объектов целевого класса их важностью в выборке. То есть, если мы имеем соотношение представителей классов, различающееся в десятки или сотни раз, изменим это соотношение так, чтобы оно стало примерно одинаковым. Под соотношением здесь имеется в виду следующее: суммируем все веса представителей первого класса и все веса представителей второго класса, а дольше считаем соотношение полученных величин. Тогда, если в первоначальном варианте все веса всех объектов выборки равняются единице, то есть до того момента, когда мы производим перевзвешивание, то соотношение как раз будет отражать дисбаланс классов в выборке. Теперь если мы увеличим веса у объектов целевого класса, то мы сможем склонить соотношение к более сбалансированному варианту. Также мы можем задавать веса объектам, исходя из стоимости ошибок классификации разного рода. В задаче оттока нам скорее важно найти клиентов, которые собираются уйти, чем клиентов, которые не собираются никуда уходить. Поэтому если мы ошибочно отнесем клиента, который уходить не собирается, к оттоку, это не так страшно, как если мы пропустим клиента, который от нас уходить собирается, потому что в этом случае мы, конечно же, не сможем его

удержать. Соответственно, исходя из этих соображений, если мы понимаем в цифрах, насколько стоимость этой ошибки больше, то полагаясь на это мы тоже можем устанавливать веса.

Следующая стратегия называется oversampling. Идея этого подхода следующая: если в нашей выборке не хватает представителей целевого класса,то есть представителей класса «отток», то сгенерируем больше объектов этого класса. Далее вопрос заключается в том, каак именно генерировать эти объекты. Один из подходов – создать дубликаты уже имеющихся представителей, что на самом деле равносильно методу переопределения весов представителей классов. Также можно сгенерировать новые объекты на основании имеющихся, немного изменив их значения и создать таким образом новых представителей класса «отток». Кроме того, мы можем генерировать новые объекты на основании нескольких объектов из класса «отток». То есть, скажем, взять группу некоторых объектов и каким-то образом их усреднить и построить объекты, похожие на несколько объектов целевого класса.

Другая стратегия — это стратегия undersampling, в некотором роде противоположная этой стратегии. В данном случае мы полагаем, что в выборке присутствует избыточное количество представителей нецелевого класса, от которых мы можем избавиться. Выберем объекты, которые мы хотим удалить из выборки. Очевидный вариант – убрать объекты нецелевого класса случайным образом. Также можно их кластеризовать и убрать какую-то долю представителей каждого кластера.

В нашей задаче наблюдается сильный дисбаланс классов - абонентов, ушедших в отток, гораздо меньше, чем абонентов, стабильно пользующихся услушами оператора. Воспользуемся здесь перераспределением весов представителей обоих классов. Выбранная нами реализация алгоритма обучения модели

имеет возможность указать эту величину в качестве параметра.

Обсудим метод оценки модели во время обучения - кросс-валидацию. При таком методе мы делим объекты выборки по непересекающимся группам пользователей. Те пользователи, которые участвуют в обучении, не участвуют в тестировании модели.

Теперь опишем подбор параметров. Помимо того, что подбор параметров нужно делать на основании кросс-валидации, важно всегда оставлять некоторую независимую выборку, которая никак не используется в процессе подбора параметров и других измерений в процессе оптимизации модели. Это необходимо для того, чтобы мы имели такой неприкосновенный инструмент, для того чтобы протестировать финальное решение. В тот момент, когда мы обучили модель практически на всех данных, нам важно понять, не допустили ли мы нигде никаких ошибок. Вот в этом случае всегда можно использовать ту выборку, которая в предыдущем анализе не участвовала никак.

3.8 Метрика оценки модели

Популярной метрикой в задаче предсказания оттока является лифт. Проранжируем по убыванию вероятности оттока, предсказанные моделью для всего множества клиентов из валидационной выборки. Далее возьмем некоторую квантиль от этого списка и посчитаем в новом списке долю клиентов, которые действительно ушли в отток и сравним ее с долей клиентов, которые бы ушли в отток, если бы мы взяли случайную выборку. Метрикой будет являться отношение этих двух чисел. То есть мы можем определить, насколько сильно будет отличаться список клиентов, предоставляемых моделью, от случайного выбора.

$$lift = \frac{predicted\ churn\ rate}{average\ churn\ rate}$$

Если варьировать квантиль множества выбираемых абонентов, то можно получить график с изображинием lift-кривой. Пример изображен на рис. 2.

Таким образом, если лифт равен двум, то это будет означать, что в выборке содержится в два раза больше абонентов, находящихся под угрозой оттока, чем в выборке, составленной случайно.

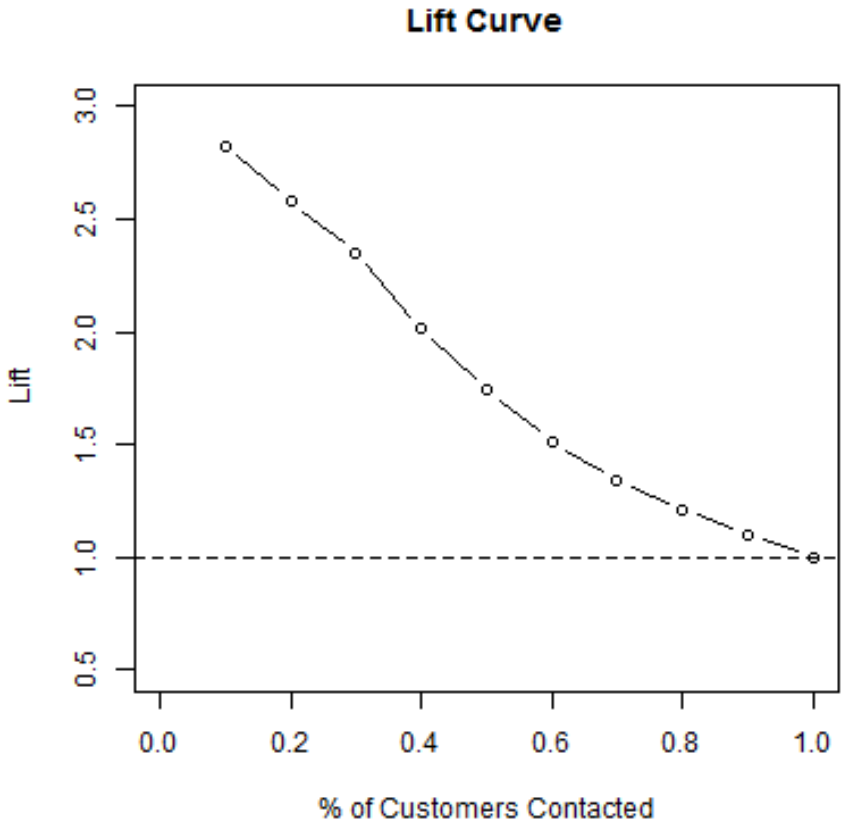

Рис. 2: Пример лифт-кривой

3.9 Построение модели

Будем оценивать модель по метрике лифт на верхней децили от тестовой выборки. То есть во сколько раз больше пользователей, ушедших в отток, окажется в верхней децили тестовой выборки, чем при случайном отборе.

Далее организуем в пространстве гиперпараметров поиск модели, оптимизирующей установленную нами метрику. То есть будем независимо обучать множество моделей и выбирать лучшую в контексте величины лифта.

Опишем параметры, которые мы будем оптимизировать:

- $n\_estimators$ - количество деревьев, учавствующих в ансамбле.
- max_depth - максимальная глубина дерева, учавствующего в ансамбле.

- *min_child_weight* - минимальный суммарный вес наблюдений, входящих в поддерево, который определяет, будет ли данное поддерево дальше делиться.

- *subsample* - доля наблюдений из обучающей выборки, используемых для построения данного дерева.

- *colsample_bytree* - доля признаков, используемых для построения данного дерева в ансамбле.

После обучения получаем модель (рис. 3).

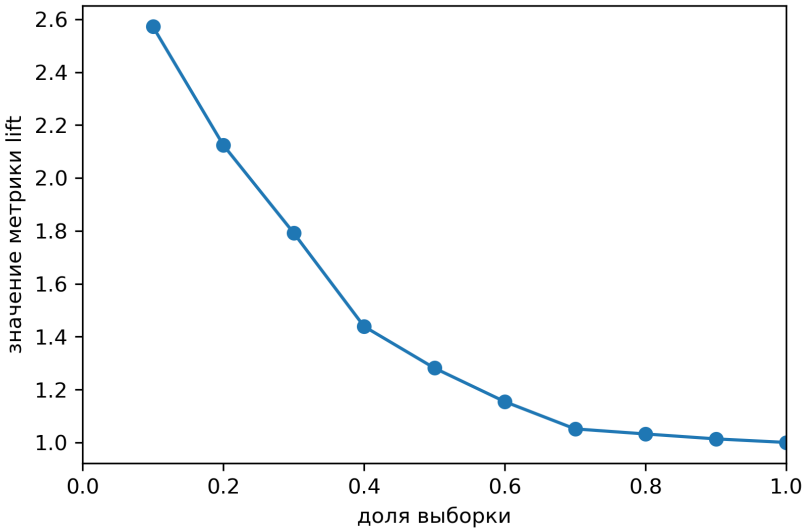

Рис. 3: Значения метрики итоговой модели

Таким образом, значение метрики на первой децили выборки составляет 2,57. То есть, в данной выборке будет содержаться в 2,57 раза больше пользователей, уходящих в отток, чем если бы мы брали пользователей случайным выбором. Отсюда мы можем сделать вывод о том, что качество интернет-соеднинения абонента является важным фактором в принятии решения о переходе от одного оператора к другому.

4 Заключение

При проведении анализа было поставлена гипотеза о том, что параметры качества интернет-соединения являются важным фактором, влияющим на отток клиента.

В данной работе был предложен уникальный подход к разметке данных в условиях предсказания оттока для фиксированного оператора, так как в такой постановке абонент представляет из себя не одно устройство, а точку доступа, имеющую множество устройств в качестве домашних. А именно, был предложен способ, включающий данные о геолокации устройств, полученные во время интернет-сессий.

Была построена модель, которая предсказывает отток и был получен результат, подтверждающий наличие связи между качеством интернет-сессий и желанием абонента перейти от одного оператора к другому. Данная работа имеет практическое применение для операторов связи для того, чтобы снижать количество абонентов, уходящих в отток, путем своевременного обнаружения клиентов с высокой предрасположенностью к смене оператора и их удержания. В свою очередь, снижение оттока ведет к снижению издержек оператора, так как удержание действующих абонентов обходится оператору дешевле, чем привлечение новых.