УДК 004.622

Исаев Сергей Александрович
магистрант, факультета программной
инженерии и компьютерной техники,
Санкт-Петербургский национальный исследовательский
университет информационных технологий,
механики и оптики, Санкт-Петербург
ORCID ID: 0000-0002-6479-7587, email: 0994486@gmail.com
Isaev Sergey
master's student, faculty of Software Engineering and Computer Systems
Saint-Petersburg National Research University
of Information Technologies, Mechanics and Optics, St. Petersburg

ХАРАКТЕРИСТИКИ НЕАГРЕГИРОВАННОГО НАБОРА ДАННЫХ, ПРЕДОСТАВЛЯЕМОГО СЕРВИСОМ ВЕБ-АНАЛИТИКИ «ЯНДЕКС.МЕТРИКА»

Аннотация. В статье исследуются основные характеристики неагрегированных данных первичной аналитики сервиса «Яндекс.Метрика», возможность их использования совместно со средствами машинного обучения. Исследование показало, что неагрегированные данные, предоставляемые сервисом «Яндекс.Метрика», позволяют формировать объемные по своему признаковому пространству датасеты. Могут быть использованы совместно с алгоритмами машинного обучения, однако их применение возможно только после глубокой предварительной обработки. Результаты работы могут быть полезны при выборе системы веб-аналитики, также они помогут оценить трудоемкость подготовки неагрегированных данных для дальнейшего применения в задачах машинного обучения.

Ключевые слова: Яндекс метрика, веб-аналитика, неагрегированные данные, машинное обучение.

CHARACTERISTICS OF THE NON-AGGREGATED DATA SET PROVIDED BY THE WEB ANALYTICS SERVICE «YANDEX.METRICA»

Annotation. The article examines the main characteristics of non-aggregated data of primary analytics of the «Yandex.Metrica» service, the possibility of their use in conjunction with machine learning tools. The study showed that the non-aggregated data provided by the «Yandex.Metrica» service allows the formation of data sets that are voluminous in terms of their characteristic space. They can be used together with machine learning algorithms, however, their application is possible only after deep pre-processing. The results can be useful when choosing a web analytics system, they will also help to evaluate the complexity of preparing non-aggregated data for further use in machine learning tasks.

Keyword: Yandex. Metrica, web analytics, non-aggregated data, machine learning.

Введение

Специалисты видят задачу веб-аналитики в получении и графическом представлении т. н. веб-метрик: трафика, коэффициента конверсии, количества кликов, цены кликов, цены лида и продажи, предпочтений клиентов и т. д. Веб-аналитика применяется и для работ по оптимизации рекламных кампаний, повышению конверсии или увеличению количества заказов, уменьшению процента посетителей, совершивших отказ [3].

Развитие облачных технологий и соответствующей инфраструктуры дало возможность веб-аналитике, на базе собираемых метрик, подойти к решению задач с применением моделей машинного обучения.

Идея обработки веб-метрик с помощью современных средств машинного обучения или искусственных нейронных сетей (англ. Artificial Neural Networks, ANN) в прикладных задачах кажется весьма перспективной. К примеру, модуль ИСППР (англ. Intelligent Decision Support Systems, iDSS), в котором используется данный принцип, мог бы заострять внимание менеджера по продажам (или лица, принимающего решение, – ЛПР), на

перспективном клиенте. В качестве другого примера можно привести систему кастомизации контента на образовательных онлайн платформах. Учитывая тренд на развитие современных образовательных технологий EdTech (англ. Educational technology), это была бы очень востребованная задача [2].

На данный момент такие инструменты могут позволить себе крупные компании. Тем не менее уже довольно давно существуют доступные (условно бесплатные) и универсальные сервисы веб-аналитики от ведущих IT-компаний. Как правило, они предоставляют данные в агрегированном виде. Такой формат предполагает получение различных выборок из исходного набора в удобном для восприятия и анализа виде (таблицы, диаграммы и т. п.). Несмотря на очевидную простоту восприятия и наглядность агрегированных данных, их анализ часто носит поверхностный или ограниченный характер. Агрегированные данные тяжело использовать в системах с высокой степенью автоматизации и применять к ним алгоритмы ML или ANN.

Кроме отчетов на основе агрегированных данных, некоторые системы вебаналитики предоставляют доступ и к неагрегированным (сырым) данным первичной аналитики. Цель нашего исследования – получить представление о применимости сырых (неагрегированных) данных в задачах с использованием алгоритмов машинного обучения. Модели, полученные при помощи таких алгоритмов, могут применяться в инфраструктуре прикладных веб-приложений в качестве более доступной по стоимости замены системам аналитики корпоративного уровня.

Исследуемые данные собирались при помощи сервиса «Яндекс.Метрика» с мая 2014 года по декабрь 2019 года с официального веб-ресурса коммерческой организации, предоставляющей услуги в сфере дачного строительства, благоустройства и монтажа ограждений. Организация осуществляла услуги на территории одного из субъектов Российской Федерации – Ленинградской области[1].

В нашем исследовании мы использовали одну из самых популярных в мире систем аналитики от компании Яндекс – «Яндекс.Метрика» [8]. Принципы ее работы и основной функционал схожи с подобными системами от других компаний, а значит, результаты исследования могут иметь ценность для пользователей других систем.

Получение данных

Сервис «Яндекс.Метрика» предоставляет инструменты для получения как агрегируемых, так и неагрегируемых данных веб-аналитики. Агрегируемые отчеты формируются посредством выборки из исходного массива одного или нескольких многомерных наборов данных (гиперкуб или метакуб). К сожалению, такие обобщенные (агрегированные) отчеты ограничены функционалом сервиса. Если данных очень много, они могут быть подвергнуты семплированию и будут предоставлять усредненные метрики. Это сильно усложняет их эффективное применение совместно с алгоритмами машинного обучения и в глубоком обучении искусственных нейронных сетей.

Неагрегированные данные представляют собой записи об отдельных визитах или просмотрах, а также различную техническую информацию о визите – например, браузер и модель мобильного телефона [1]. Таким образом, сырые данные предоставляются ровно в том виде, в котором их извлек JavaScript дескриптор, внедренный на страницы сайта. Для выгрузки первичных данных используется новый программный интерфейс от компании Яндекс – Logs API. Он позволяет формировать запросы на выгрузку. Для работы с ними удобно применять специальные аналитические базы данных, такие как ClickHouse от

[1] Jupyter Notebook с программным кодом проекта, исходные данные и дополнительную информацию можно найти по адресу https://github.com/rukivbruki/Web_metrics_Analytics_using_machine_learning.

компании Яндекс или bigQuery от Google. В нашей работе мы пошли иным путем и использовали возможности языка Python и библиотек, написанных на этом языке.

Logs API позволяет получить данные двух видов – это данные о просмотрах (hits) и о визитах (visits). Для каждого набора данных предоставляется различная информация – в зависимости от переданного в запросе набора параметров. В рамках настоящей статьи мы исследовали данные о визитах, так как они, по сравнению с просмотрами, имеют более полную структуру и лучше отражают результаты исследования.

Для получения данных о визитах необходимо сделать запрос следующего вида: https://api-metrika.yandex.net/management/v1/counter/{counterId}/logrequests/evaluate и передать параметры:

?[date1=<string>]&[date2=<string>]&[fields=<string>]&[source=<log_request_source>] [4]

Формирование запроса на языке Python может выглядеть так, как показано на рисунке 1:

```
import requests

url_params = urllib.parse.urlencode(
    [('date1', START_DATE), ('date2', END_DATE), ('source', SOURCE),
     ('fields', ','.join(sorted(API_FIELDS, key=lambda s: s.lower())))]
)

url = f'{API_HOST}/management/v1/counter/{COUNTER_ID}/logrequests?' \
      + url_params

result = requests.post(url, headers={'Authorization': 'OAuth ' + TOKEN})
```

Рисунок 1. – Запрос к Logs API средствами языка программирования Python (v. 3.6)

Logs API не предоставляет данные сразу. Сначала делается основной запрос на их формирование. Затем запросу присваивается статус created. После постановки запроса в очередь и подготовки данных он приобретает статус processed, что означает готовность к выгрузке. Общая схема работы с Logs API изображена на рисунке 2:

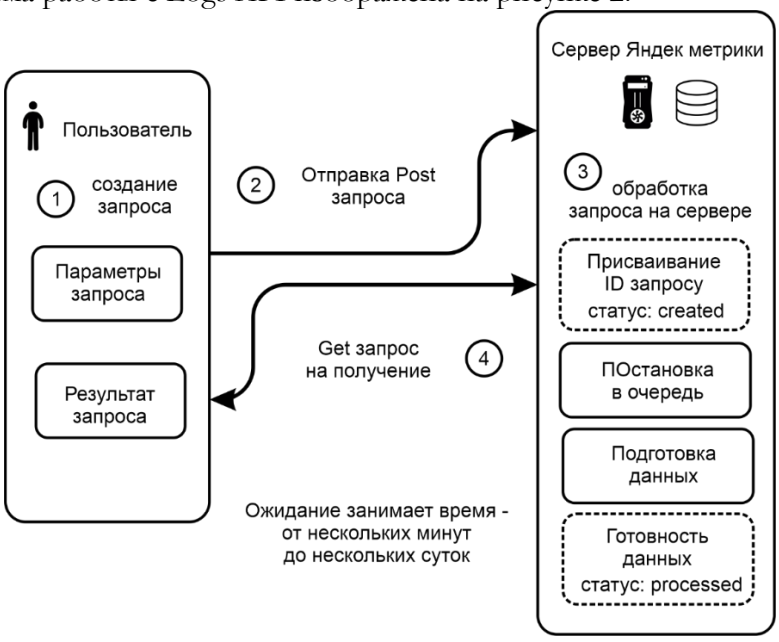

Рисунок 2. – Общая схема работы с Logs API

Описание исследуемых данных

Сервис «Яндекс.Метрика» можно гибко настроить на сбор необходимой информации, если передать ему подробные данные по рекламным кампаниям, электронной коммерции или вести собственную атрибуцию событий на сайте. Полнота предоставляемого отчета может отличаться в зависимости от настроек. При формировании

исследуемого набора данных применялись только стандартные настройки сервиса, а запрос на формирование данных запрашивал все поля1, доступные на данный момент.

После загрузки данных с сервера они были сконвертированы в файл формата CSV, объемом 192,5 мегабайта. Для обработки файла и перевода его в удобную для анализа структуру датафрейм (англ. Data Frame) нами были применены методы библиотек NumPy и Pandas. Ниже приведены основные характеристики полученного датафрейма:

– **Форма датафрейма:** 123394 строки, 134 колонки[2].

– **Колонки данных:** всего присутствует 134 колонки. Каждая колонка содержит значения одного признака: идентификатор визита, номер счетчика, просмотры, которые были в данном визите, дата и время визита, часовой пояс на компьютере посетителя и другие. С полным перечнем возможных параметров можно ознакомиться в документации [4] или на странице нашего Jupiter Notebook.

– **Формат дынных:** float64 – 11 колонок, int64 – 30 колонок и object – 93 колонки. Около 69 % всех данных о визитах представлены в виде категориальных значений (тип object в структуре данных Data Frame – это значения с использованием строковых объектов в Python). Оставшиеся значения датасета также можно рассматривать в качестве номинальных, т. е. определяя факт принадлежности к какой-то категории.

Например, рассмотрим поле ym:s:clientTimeZone – часовой пояс на компьютере посетителя. В исследуемом наборе данных значения поля имеют тип int64 и могут принимать следующие величины: 180, 420, 240, 300, 630, 330, 480, 60. 720, -1 и т. д. Очевидно, не имеет никого смысла оперировать этими значениями как числами или проводить с ними математические операции. Подобная логика применима практически ко всему признаковому пространству исследуемых нами данных.

Полнота данных

Так как в задачах машинного обучения признаки с отсутствующими значениями, равно как и с одним единственным значением, не имеют особого смысла, мы удалили из датафрейма все подобные столбцы. Ниже представлены результаты:

– **Форма датафрейма после удаления лишних столбцов:** 123394 строки, 78 колонок.

– **Осталось данных:** 57 %.

Объем (общее количество ячеек) сократился почти вдвое, 56 колонок было удалено. Однако после удаления лишних колонок осталось немало отсутствующих значений. На рисунке 4 показано их распределение по столбцам, в процентах от общего количества значений в каждом из них:

[2] Термин “колонка” или “столбец” используется, когда речь идет о структурах данных типа Data Frame и соответствует определению “поле” в терминологии баз данных или “признаку” в машинном обучении.

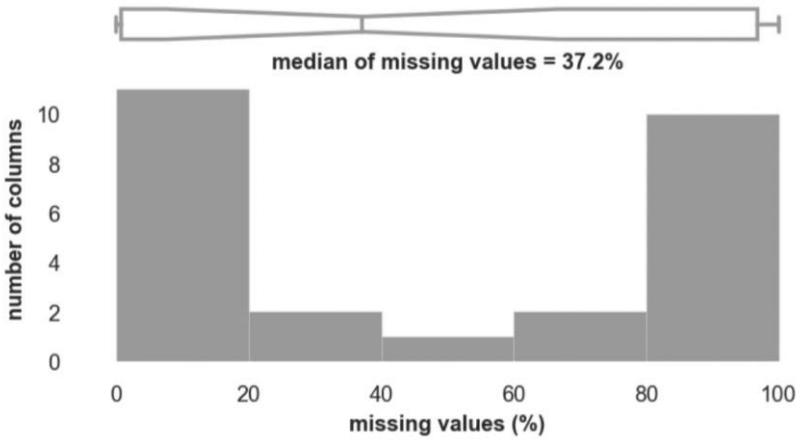

Рисунок 3. – Распределение пустых значений в исследуемом наборе данных

Видно, что данные, предоставляемые сервисом «Яндекс.Метрика», неоднородны и могут содержать большое количество пустых значений в пределах одного столбца (признака). Так, медиана доли отсутствующих значений составляет 37,2 %. Это обстоятельство делает затруднительным применение датасета во многих алгоритмах машинного обучения. Стоит позаботиться о дополнительной обработке исходных данных и предусмотреть заполнение пустых значений. **Интерпретируемость данных**

Предоставляемый сервисом «Яндекс.Метрика» набор данных интерпретируем. В зависимости от поставленных задач он может быть как сокращен, так и увеличен. В задачах машинного обучения это означает изменение признакового пространства. Рисунок 3 иллюстрирует подобный подход:

	ym:s:watchIDs	watchCounts
0	[2374542296120035191,237455688304060197 5,23746...	6
1	[18053553872993259407,18053565303429014250,180...	9
2	[5624597053286321553,5624657149761359519]	2
3	[1618959677078900425]	1
4	[2390111394919876378]	1
...	...	...

Рисунок 4. – Расширение признакового пространства за счет интерпретации существующих данных

Столбец ym:s:watchIDs содержит идентификаторы просмотров сайта для каждого посещения. На основе ym:s:watchIDs мы создали новый столбец watchCounts – количество просмотров в рамках одного посещения. Можно пойти еще дальше и, используя идентификатор просмотров, получить новые поля из другого набора данных, предоставляемого «Яндекс. Метрикой», – данных о просмотрах. Возможности подобных комбинаций практически неограниченны и позволяют сконструировать датасет с высокой степенью детализации.

Избыточность данных

Для быстрого визуального анализа данных на наличие мультиколлинеарности мы использовали Python библиотеку Seaborn. Она отображает корреляционную матрицу в виде тепловой карты. Чтобы данные отображались корректно, к ним был применен метод библиотеки Scikit-learn – featureHasher [7]. Этот метод преобразует последовательности номинальных признаков в разряженные матрицы, используя хэш-функцию для вычисления столбца матрицы соответствующего имени [5]. Результаты для первых 15 столбцов, полученные с использованием коэффициента корреляции Пирсона, можно наблюдать на рисунке 5:

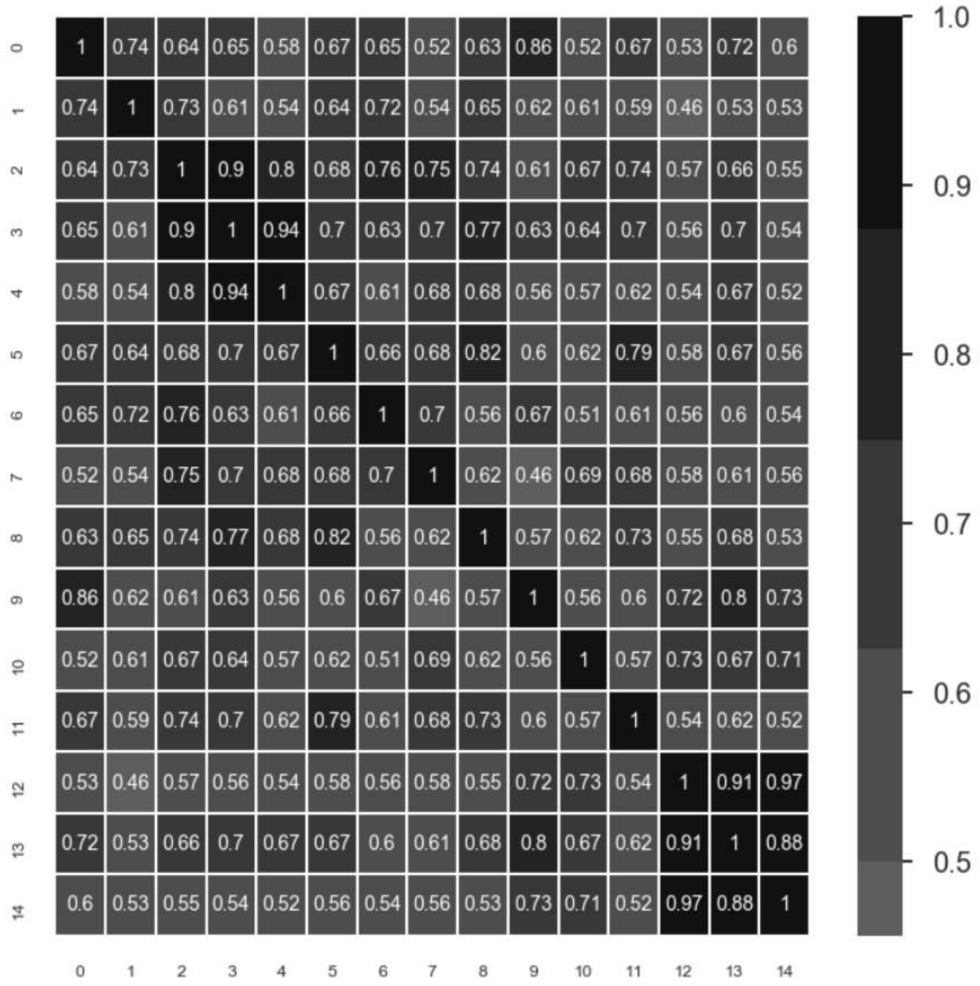

Рисунок 5. – Матрица корреляции признаков исследуемого набора данных

Полученная матрица корреляции визуализирует лишь около четверти исследуемых данных, тем не менее она наглядно свидетельствует о наличии тесной взаимосвязи между многими переменными. Чтобы проверить гипотезу о наличии мультиколлинеарности в данных, мы взяли пару случайных признаков и проверили их на наличие корреляции при помощи V – коэффициента Крамера, с поправкой на смещение [6]. Для значений ym:s:regionCountryID (index=14) и ym:s:regionCity (index=12) этот показатель составил 0,999, что свидетельствует о тесной связи. Общей рекомендацией для датафрейма с такими характеристиками будет применение к нему методов снижения размерности (англ. Dimensionality reduction).

Оптимизация данных

Большинство колонок в исследуемом нами наборе данных содержат небольшое количество уникальных значений – менее 50 %. Это дает возможность произвести их оптимизацию за счет преобразования данных датафрейма с типа Object в тип Category. Результат оптимизации представлен в табл. 1.

Таблица 1 –Показатель оптимизации при помощи изменения типа данных

Типы данных	**Использовано памяти**
Object	336,8 MB
Category	72,6 MB
Оптимизация: 78,4%	

Если применить эту оптимизацию к полям с высоким процентом уникальных значений (ym:s:visitID – 81 %, ym:s:watchIDs – 100 %, ym:s:dateTime – 95 %, ym:s:dateTimeUTC – 95 %, ym:s:clientID – 63 %), уровень использования памяти вырастет.

Заключение

Сервис «Яндекс.Метрика» позволяет получить неагрегированные данные при помощи интерфейса Logs API. Процесс получения данных состоит из нескольких этапов, по окончании можно запросить данные о просмотрах и визитах. Около 69 % всех данных о визитах представлены в виде категориальных значений, однако почти все значения можно рассматривать как номинальные, вне зависимости от исходного типа. Настроенный по умолчанию сервис «Яндекс.Метрика» формирует набор данных, который может содержать большое количество пустых или малозначительных с точки зрения машинного обучения признаков. После их удаления объем может сократиться почти в 2 раза. Оставшиеся данные неоднородны. Медиана доли отсутствующих значений по столбцам составляет 37,2 %. Необходимо дополнительно обработать такие поля перед применением алгоритмов машинного обучения и искусственных нейронных сетей. Предоставляемый «Яндекс.Метрикой» набор данных весьма интерпретируем. В зависимости от поставленных задач пользователю предоставляется широкая возможность для конструирования новых значений. Колонки в наборе данных, содержащие количество уникальных значений менее 50 %, могут быть эффективно оптимизированы за счет преобразования их типа с Object в тип Category.

Данные имеют признаки мультиколлинеарности – это обуславливает применение алгоритмов снижения размерности при последующей работе с ними.

Список литературы:

1. Блог Яндекс.Метрики: Logs API: выгружайте сырые данные из Метрики [Электронный ресурс]. Режим доступа: https://yandex.ru/blog/metrika/vygruzhayte-syrye-dannye-iz-metriki-cherez-logs-api (Дата обращения: 09.04.2020).
2. Интерфакс Академия – Исследование рынка цифровых образовательных технологий в сегменте взрослой аудитории (EdTech в дополнительном профессиональном образовании (ДПО) и дополнительном образовании (ДО) взрослых) [Электронный ресурс]. Режим доступа: https://academia.interfax.ru/ru/analytics/research/4257/ (Дата обращения: 09.04.2020).
3. Кошик, А. Веб-аналитика 2.0 на практике. Тонкости и лучшие методики / А. Кошик. Пер. с англ. – М.: Вильямс, 2011. – 528 с.: ил.
4. Оценка возможности создания запроса [Электронный ресурс]. Режим доступа: https://yandex.ru/dev/metrika/doc/api2/logs/queries/evaluate-docpage/ (Дата обращения: 09.04.2020).
5. Плас, Д. В. Python для сложных задач: наука о данных и машинное обучение. – СПб.: Питер, 2018. – 576 с.: ил.
6. Bergsma, Wicher A bias correction for Cramér's V and Tschuprow's T // Journal of the Korean Statistical Society, 2013. – Vol. 42, No. 3. – P. 323–328.
7. Sklearn.feature_extraction.FeatureHasher – scikit-learn 0.22.2 documentation [Электронный ресурс]. Режим доступа: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.FeatureHasher.html (Дата обращения: 09.04.2020).
8. Usage statistics of traffic analysis tools for websites [Электронный ресурс]. Режим доступа: https://w3techs.com/technologies/overview/traffic_analysis (Дата обращения: 09.04.2020).