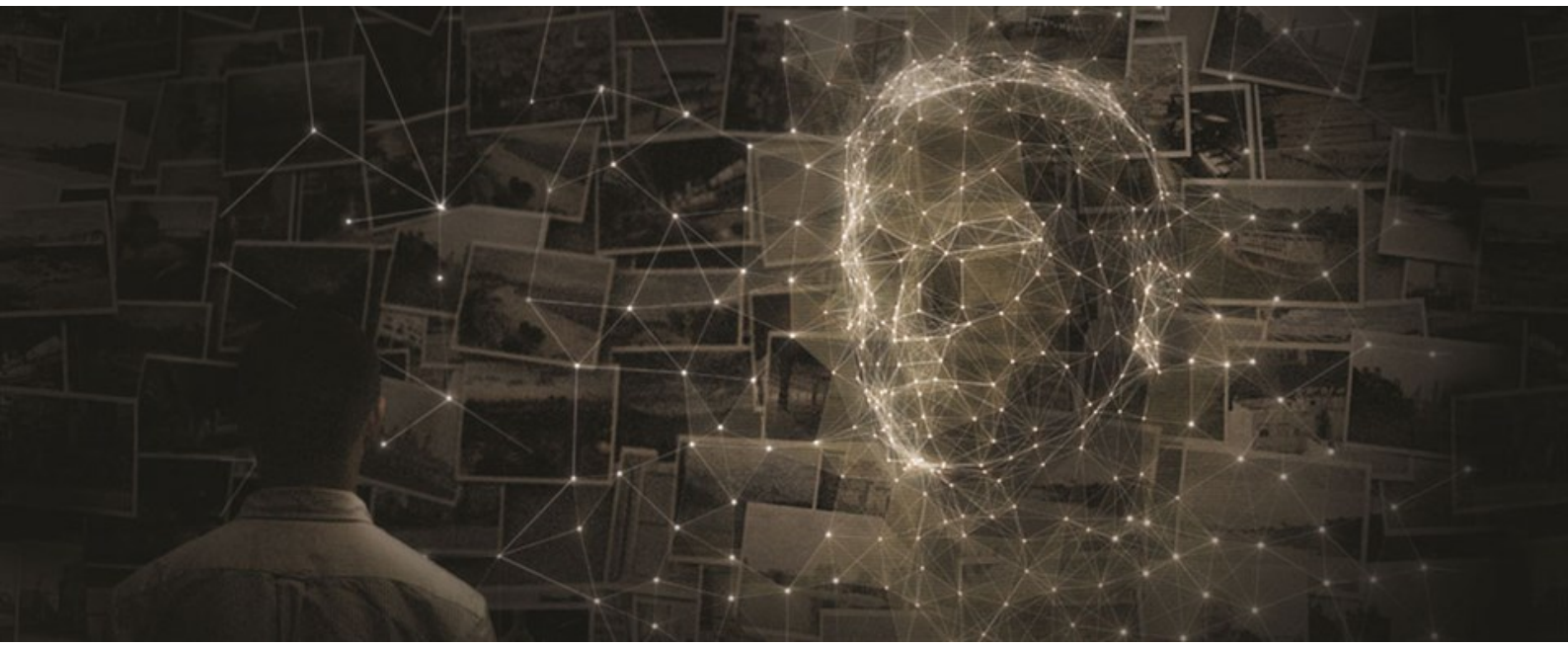


Финальный проект специализации Машинное обучение и анализ данных

тема: **Идентификация интернет-пользователей**

Ахременко В.В.

2021

Постановка задачи

Идентификация пользователя по его поведению в сети Интернет по последовательности из нескольких веб-сайтов, посещенных подряд одним и тем же человеком.

Данные

Для решения задачи предоставлены данные с прокси-серверов Университета Блеза Паскаля, их вид очень простой: ID пользователя, timestamp, посещенный веб-сайт.
Из данных выделены 3 подвыборки меньших размеров:
1. данные по 10 пользователям
2. данные по 150 пользователям
3. данные по пользователям, разбитые на 2 класса: 1- условный пользователь Alice, 0 — пользователь не Alice.

Данные по 10/150/3000 пользователям представлены как последовательность ID пользователя, timestamp, посещенный веб-сайт (веб-адрес).
Данные для идентификации Alice представлены в формате срезов из 10 сайтов и соответствующим им временным меткам. Так же длинна сессии ограничена 30 минутами

Предобработка данных

Для удобства все адреса сайтов были заменены на их номера в соответствии с частотой посещения.

Для анализа все посещенные пользователем сайты разбиты на сессии по 10 сайтов без перекрытия и ограничения по длительности сессии. В дальнейшем предполагалось длину сессии, ее максимальную длительность и перекрытие сделать гиперпараметрами при построении решения.

Сами номера сайтов не представляют интереса. Сформировали матрицу из признаков, где каждый признак — номер сайта, а его значение — количество раз, когда сайт встретился в сессии. Т.к. полученная матрица содержит в основном нули (в ряду/сессии максимальное кол-во не нулевых элементов — 10 из 48371 в выборке из 150 пользователей), для уменьшения объема преобразовали полученную матрицу в формат scipy.sparse.csr_matrix.

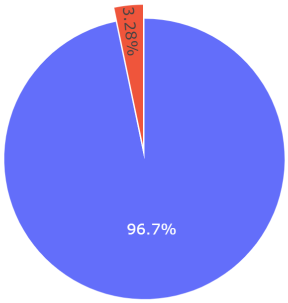
Так же выделены признаки:
- длительность сессии
- число уникальных посещенных сайтов в сессии
- время начала сессии
- день недели начала сессии

Первичный и визуальный анализ данных

Для анализа будем использовать выборки по 10 и 150 пользователях, т.к. оценивать возможности и работоспособность алгоритма следует на меньших объемах.

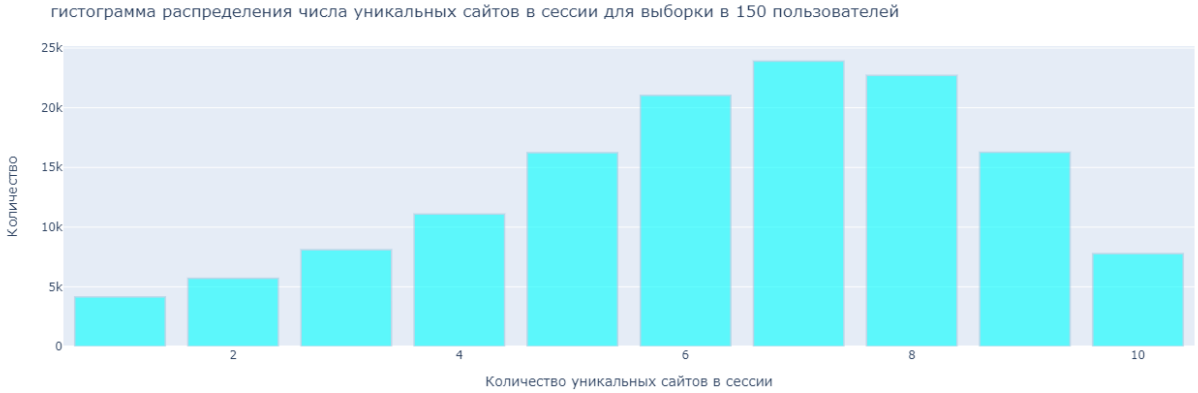
Данные представлены с 15 ноября 2013 по 28 мая 2014 с частичным перекрытием по пользователям.

Обращает на себя внимание малое количество сессий каждого класса в отношении ко всей выборке. Например по 150 пользователям максимальная доля одного пользователя в выборке — 3.28%



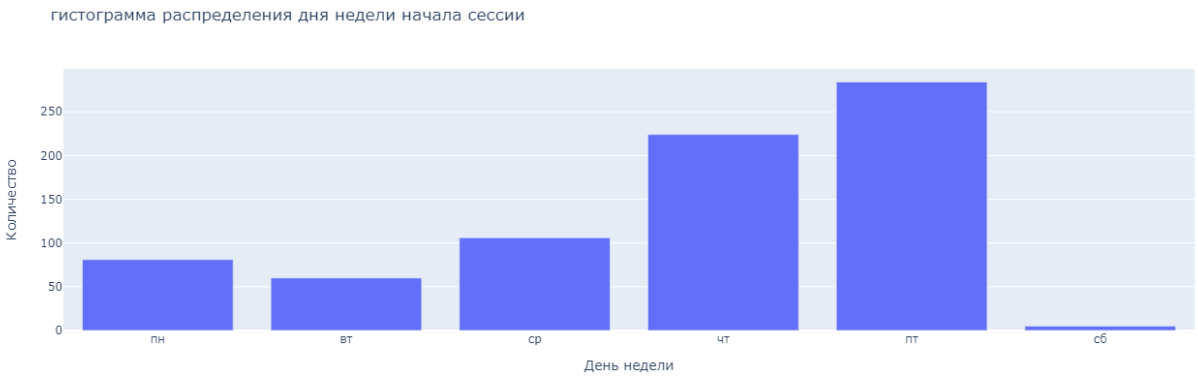
Пропусков нет.

Несмотря на визуальную схожесть с нормальным распределением гистограммы распределения числа уникальных сайтов для выборки 150 пользователей, Критерий Шапиро — Уилка отвергает данную гипотезу. Аналогично и для выборки в 10 пользователей.



Так же отвергаются гипотезы о нормальном распределении признаков время начала сессии и день недели начала сессии как для выборки из 10 пользователей, так и для выборки в 150 пользователей.

Визуальный анализ позволил выделить ряд паттернов в поведении пользователей. Например сессии в рабочие дни и почти полное отсутствие сессий на выходных.



Или число уникальных сайтов в сессии с пиком в 1/2.

Визуальный анализ подтвердил, что в поведении пользователей паттерны, которые возможно использовать в его идентификации.

Построение признаков

Следующим этапом предлагалось самостоятельно сконструировать признаки.

При построении признаков исходил из того, что необходимо идентифицировать 150 — 3000 пользователей. Следовательно, числовым и категориальным признакам отдается предпочтение при оценке трудозатрат на построение признака и ожидаемой отдачи по сравнению с бинарными.

Были созданы признаки: метрики длительности посещения первого сайта в сессии – mean и std. С идеей генерировать на их основе вероятность получить такое же значение для конкретного пользователя через расчет p-value для нормального распределения.

Так же, в виду проведенного нами визуального анализа данных, была сделана разбивка признаков на будний день и выходной.

Сравнение нескольких алгоритмов, оптимизация, выбор гиперпараметров

Сравнение алгоритмов было проведено по выборке из 10 пользователей для многоклассовой классификации с гиперпараметрами:
ширина окна = 10
длина сессии = 10
ограничения по максимальной длине сессии — нет.

Производилось сравнение следующих алгоритмов:
- KneighborsClassifier
- RandomForestClassifier
- LogisticRegression
- LinearSVC

с параметрами по-умолчанию.

Оценка производилась по кросс-валидации через StratifiedKFold 3-кратная, с перемешиванием по метрике accuracy.

Лучшие результаты были показаны LogisticRegression с метрикой 0.78 на отложенной выборке и LinearSVC с метрикой 0.77. После чего для двух лучших алгоритмов была произведена оптимизация параметров.

Для LogisticRegression оптимизировался параметра регуляризации C. Оптимизация производилась в 2 этапа:

- на 1м этапе через поиск по сетке с помощью LogisticRegressionCV и построение кривой обучения, отображающей как качество классификации зависит от одного из гиперпараметров алгоритма, для грубого приближения параметра регуляризации C выбирался уточненный диапазон для параметра регуляризации C
- на 2м этапе так же поиск по сетке с помощью LogisticRegressionCV и построение кривой обучение для уточненного приближения параметра регуляризации C выбирался лучший из опробованных параметров регуляризации C

С уточненным параметром регуляризации C для LogisticRegression метрика на отложенной выборке получилась 0.7798.

Для LinearSVC оптимизация проводилась так же в 2 этапа:

- на 1м этапе через поиск по сетке с помощью GridSearchCV и построение кривой обучение, отображающей как качество классификации зависит от одного из гиперпараметров алгоритма, для грубого приближения параметра регуляризации C выбирался уточненный диапазон для параметра регуляризации C
- на 2м этапе так же поиск по сетке с помощью GridSearchCV и построение кривой обучение для уточненного приближения параметра регуляризации C выбирался лучший из опробованных параметров регуляризации C

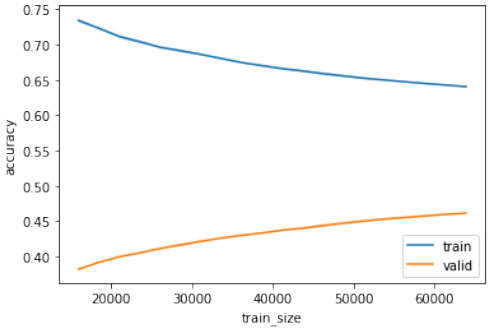
С уточненным параметром регуляризации C для LinearSVC метрика на отложенной выборке получилась 0.7817.

Затем с лучшим алгоритмом - LinearSVC - была произведена оценка гиперпараметров ширина окна, в диапазоне [10, 7, 5], и длина сессии, в диапазоне [15, 10, 7, 5] на выборке из 10 пользователей. Оценка максимальной длительности сессии не производилась.

Лучшими гиперпараметрами получились ширина окна = 5 и длинна сессии = 15 с значением метрики на отложенной выборке в 0.8764.

Оценка на выборке из 150 пользователей так же дала гиперпараметры ширина окна = 5 и длинна сессии = 15, но с существенно меньшим значением метрики — 0.6391.

В завершении данной секции была построена кривая обучения, отображающая, как качество классификации зависит от объема выборки.



Kaggle competition

В дополнение к вышеописанным было произведено исследование для варианта классификации one-vs-rest, когда пользователь сравнивается не с каждый пользователем, а против всех.

Для этого использовалась выборка Alice и, в дополнение, было принято участие в соревновании на Kaggle Inclass - Catch Me If You Can ("Alice"), Intruder Detection through Webpage Session Tracking. (https://www.kaggle.com/c/catch-me-if-you-can-intruder-detection-through-webpage-session-tracking2/overview)

Отличие этой выборки в том, что отсутствует выбор гиперпараметров. В данной выборке ширина окна = 10, длинна сессии = 10 и ограничение по длине сессии в 30 минут. Требуется определить сессии, которые соответствуют Alice.

Конечной целью являлось превышения значения метрики, заданной двумя безлайнами: первичной предобработкой данных, как мы делали ранее, и алгоритмами SGDClassifier (меньший результат) и LogisticRegression (больший результат).

Данные безлайны превышались с минимальным усилием использованием классификатора LogisticRegression с измененный балансом классов в 0.6 для Alice и 0.4 для остальных. Данное решение позволило получить 0.94474.

Так же было показано, что классификаторы на основе деревьев в моменте переобучаются на сформированной разряженной матрице и, следовательно, невозможности использовать их в дальнейшем.

Масштабирование

В завершении проекта был опробован модуль Vowpal Wabbit, показавший возможность масштабировать классификацию на десятки и сотни тысяч пользователей без серьезного наращивания инфрастуктуры.

Выводы

В результате проекта (исследования) была показана возможность идентификации пользователя по последовательности из посещенных им сайтов. Отражена возможность идентифицировать как всех пользователей одни алгоритмом, так и каждого пользователя своим алгоритмом.

В варианте идентификации всех пользователей одним классификатором показана сильно убывающая метрика в зависимости от количества пользователей, требуемых для идентификации.

В варианте идентификации одного пользователя одним классификатором удается получить сильно большую метрику, которая более стабильна. Из минусов данной стратегии — увеличение необходимых временных и вычислительных ресурсов при увеличении числа пользователей.

Оба варианта могут быть успешно использованы в соответствии с требуемой задачей.

Улучшение

Т.к. результаты получены почти без генерации признаков, основным доступным нам методом является генерация новых признаков.

Так же кажется возможным обучение различных классификаторов для идентификации пользователей для будней, выходных, рабочего времени, не рабочего времени или разделения признаков на такие группы. Данные варианты приведут к незначительному увеличению требуемых ресурсов, но, предположительно, дадут увеличение точности распознования.

Дополнительной возможностью увеличения точности классификации является получения дополнительных данных о пользователях, недоступные нам в данном проекте.