

Идентификация пользователей по посещенным веб-страницам

Специализация "Машинное обучение и анализ данных"

Выполнил Русланцев Андрей

О проекте

В этом проекте решалась задача идентификации пользователя по его поведению в сети Интернет. Это сложная и интересная задача на стыке анализа данных и поведенческой психологии. В качестве примера, компания Яндекс решает задачу идентификации взломщика почтового ящика по его поведению. В двух словах, взломщик будет себя вести не так, как владелец ящика: он может не удалять сообщения сразу по прочтении, как это делал хозяин, он будет по-другому ставить флажки сообщениям и даже по-своему двигать мышкой. Тогда такого злоумышленника можно идентифицировать и "выкинуть" из почтового ящика, предложив хозяину войти по SMS-коду. Этот пилотный проект описан в [статье](#) на Хабрахабре. Похожие вещи делаются, например, в Google Analytics и описываются в научных статьях, найти можно многое по фразам "Traversal Pattern Mining" и "Sequential Pattern Mining".

Задача курса

По последовательности из нескольких веб-сайтов, посещенных подряд один и тем же человеком, мы будем идентифицировать этого человека. Идея такая: пользователи Интернета по-разному переходят по ссылкам, и это может помогать их идентифицировать (кто-то сначала в почту, потом про футбол почитать, затем новости, контакт, потом наконец - работать, кто-то - сразу работать).

План проекта

1 неделя. Подготовка данных к анализу и построению моделей. Первая часть проекта посвящена подготовке данных для дальнейшего описательного анализа и построения прогнозных моделей.

2 неделя. Подготовка и первичный анализ данных. На второй неделе мы продолжим подготавливать данные для дальнейшего анализа и построения прогнозных моделей. Сделаем длину сессии параметром, и потом при обучении прогнозных моделей выберем лучшую длину сессии. Также мы статистически проверим первые гипотезы, связанные с нашими наблюдениями.

3 неделя. Визуальный анализ данных и построение признаков. На 3 неделе мы займемся визуальным анализом данных и построением признаков.

4 неделя. Сравнение алгоритмов классификации. Тут мы наконец подойдем к обучению моделей классификации, сравним на кросс-валидации несколько алгоритмов. Также для выбранного алгоритма построим кривые валидации.

5 неделя. Соревнование Kaggle Inclass по идентификации пользователей. Здесь мы попробуем классификатор Scikit-learn SGDClassifier, который работает намного быстрее на больших выборках

6 неделя. Vowpal Wabbit. На этой неделе мы познакомимся с популярной библиотекой Vowpal Wabbit и попробуем ее на данных по веб-сессиям.

7 неделя. Оформление финального проекта.

Данные

Будем использовать данные из [статьи](#) "A Tool for Classification of Sequential Data". Данные пришли с прокси-серверов Университета Блеза Паскаля и имеют очень простой вид. Для каждого пользователя заведен csv-файл с названием user****.csv (где вместо звездочек - 4 цифры, соответствующие ID пользователя), а в нем посещения сайтов записаны в следующем формате:

Скачать исходные данные можно по [ссылке](#) в статье, там же описание. Данные устроены следующим образом:

- В каталоге 10users лежат 10 csv-файлов с названием вида "user[USER_ID].csv", где [USER_ID] - ID пользователя;
- Аналогично для каталога 150users - там 150 файлов;
- В 3users - игрушечный пример из 3 файлов, это для отладки кода предобработки.

timestamp	site
2013-11-15 08:12:07	google.com
2013-11-15 08:12:38	youtube.com
2013-11-15 08:12:58	github.com

1. Подготовка данных

Первая часть проекта посвящена подготовке данных для дальнейшего описательного анализа и построения прогнозных моделей. Написан код для предобработки данных (исходно посещенные веб-сайты указаны для каждого пользователя в отдельном файле) и формирования единой обучающей выборки. Также в этой части использован разреженный формат данных (матрицы `Scipy.sparse`), который хорошо подходит для данной задачи.

Часть 1. Подготовка обучающей выборки

Реализована функция `prepare_train_set`, которая принимает на вход путь к каталогу с csv-файлами `path_to_csv_files` и параметр `session_length` - длину сессии, а возвращает 2 объекта: `DataFrame`, в котором строки соответствуют уникальным сессиям из `session_length` сайтов, `session_length` столбцов - индексам этих `session_length` сайтов и последний столбец - ID пользователя и частотный словарь сайтов вида `{'site_string': [site_id, site_freq]}`, например для недавнего игрушечного примера это будет `{'vk.com': (1, 2), 'google.com': (2, 2), 'yandex.ru': (3, 3), 'facebook.com': (4, 1)}`

Часть 2. Работа с разреженным форматом данных

Использована идея мешка слов по отношению к посещенным сайтам. Созданы новые матрицы, в которых строкам будут соответствовать сессии из 10 сайтов, а столбцам - индексы сайтов. На пересечении строки i и столбца j будет стоять число n_{ij} - сколько раз сайт j встретился в сессии номер i . Делать это будем с помощью разреженных матриц `Scipy - csr matrix`.

Реализована функция, преобразующая данные по собранным сессиям в разреженный формат.

2. Подготовка и первичный анализ данных

Сделаем число сайтов в сессии параметром и применим идею скользящего окна - сессии будут перекрываться. Реализуем функцию, которая возвращает 2 объекта: разреженную матрицу X_{sparse} (двухмерная `Scipy.sparse.csr_matrix`), в которой строки соответствуют сессиям из `session_length` сайтов, а `max(site_id)` столбцов - количеству посещений `site_id` в сессии и вектор `y` (Numpy array) "ответов" в виде ID пользователей, которым принадлежат сессии из X_{sparse} .

Применим функцию к исходным данным и проведем анализ информации.

Посчитаем распределение числа уникальных сайтов в каждой сессии из 10 посещенных подряд сайтов (рис. 1).

Проверим с помощью QQ-плота и критерия Шапиро-Уилка, что эта величина распределена нормально (рис. 2).

p -value очень близко к 0, уверенно отвергаем гипотезу о нормальном распределении. По графику тоже наблюдаем очень тяжелые хвосты распределения.

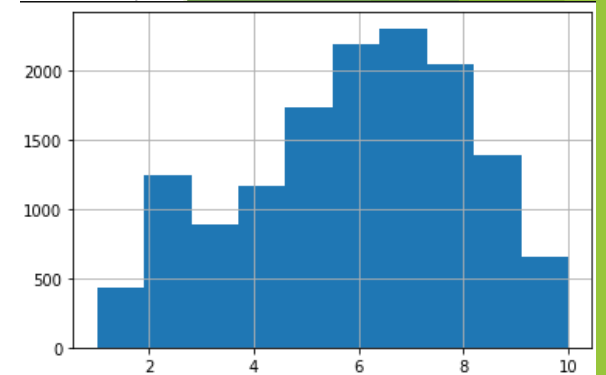


Рис. 1. Количество уникальных сайтов в сессиях длиной 10

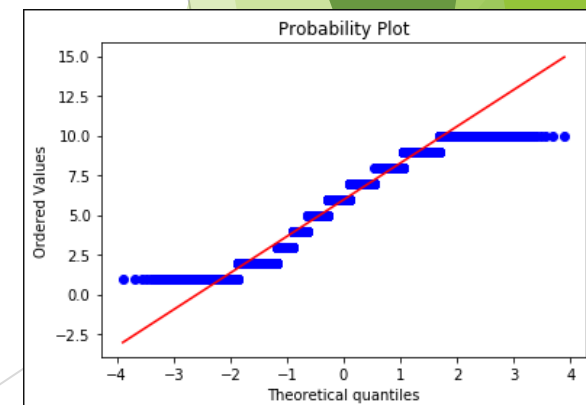


Рис. 2. QQ-plot для распределения количества уникальных сайтов

3. Визуальный анализ данных

Создадим следующие признаки:

`session_timespan` - продолжительность сессии (разница между максимальным и минимальным временем посещения сайтов в сессии, в секундах)

`#unique_sites` - число уникальных сайтов в сессии

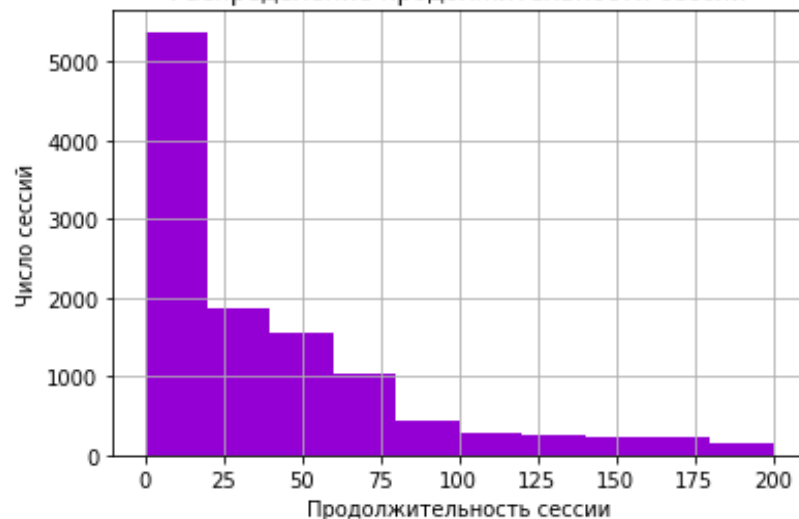
`start_hour` - час начала сессии (то есть час в записи минимального timestamp среди десяти)

`day_of_week` - день недели (то есть день недели в записи минимального timestamp среди десяти)

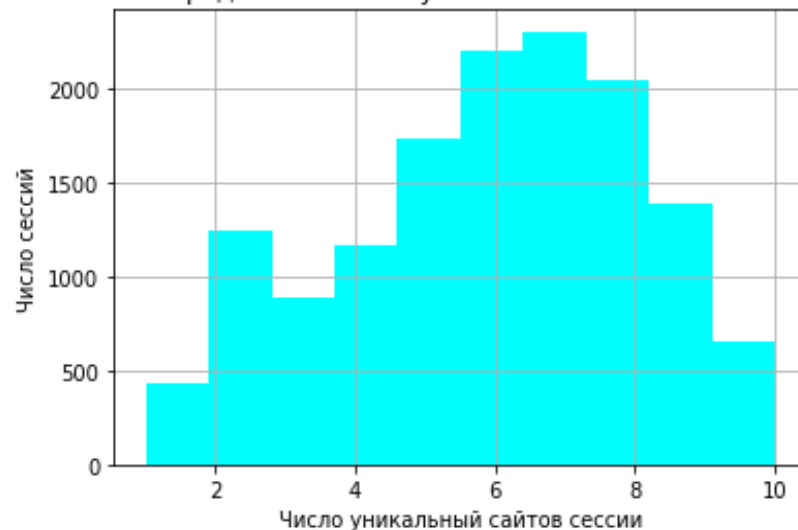
Проведем визуальный анализ данных.

3. Визуальный анализ данных

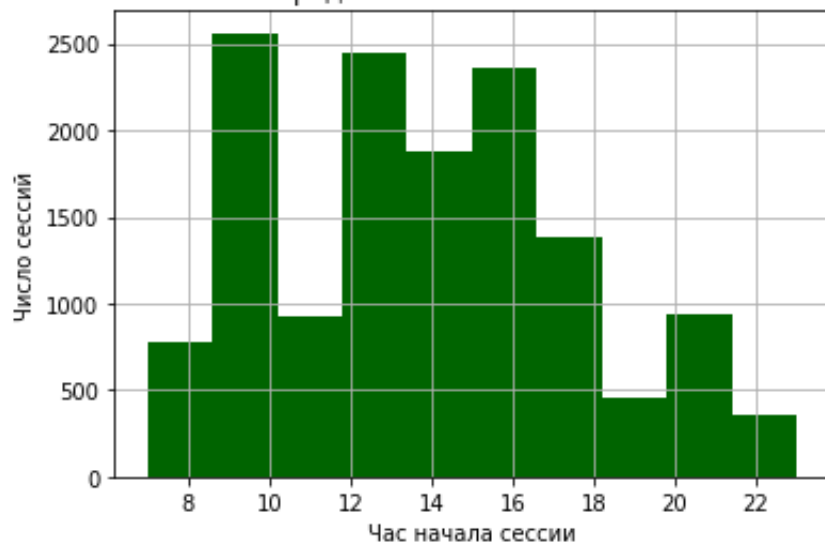
Распределение продолжительности сессий



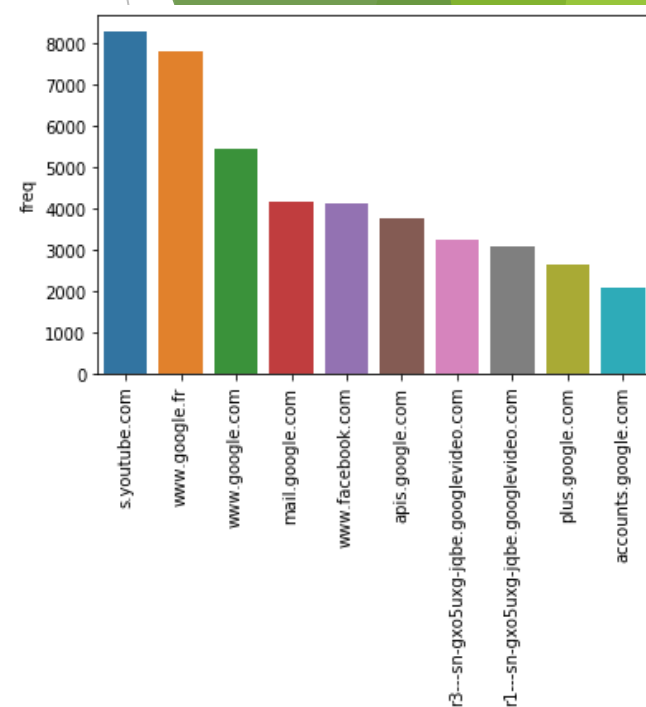
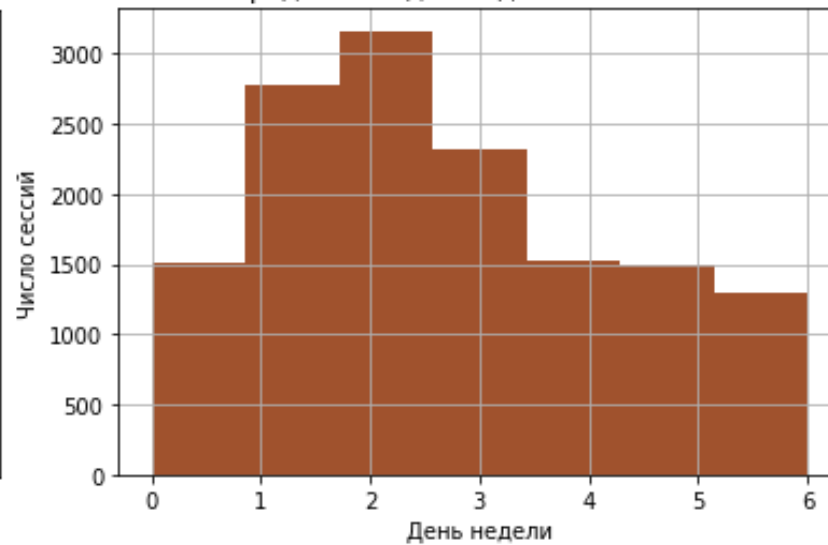
Распределение числа уникальных сайтов в сессии



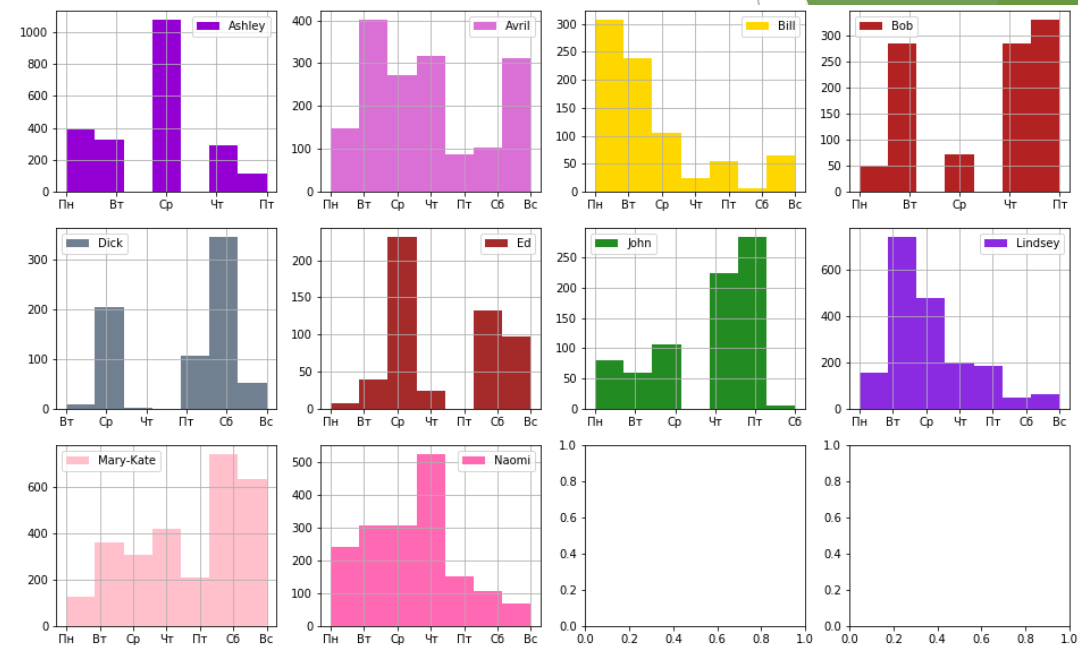
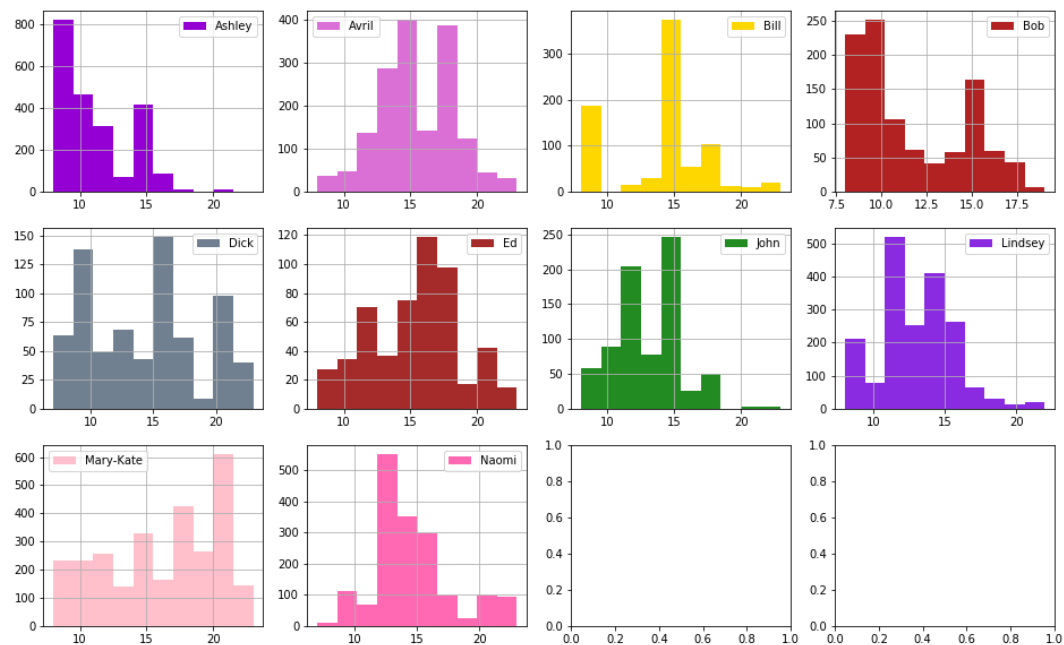
Распределение часа начала сессии



Распределение дня недели начала сессии



3. Визуальный анализ данных



3. Визуальный анализ данных

Сделаем предварительные выводы про каждого пользователя по построенным графикам.

- У всех пользователей преобладают короткие сессии
- Ashley начинает сессии утром (примерно в 9-10 часов), второй пик приходится примерно на 15 часов. Основная активность приходится на среду, на выходных нет посещений сайтов. Обычно посещает либо один сайт за сессию, либо примерно 7-8 (бимодальное распределение)
- Avril, в основном, заходит в середине дня, очень редко утром или вечером. Посещает за сессию в среднем 6 сайтов.
- Bill чаще всего заходит днем (в 15 часов), или утром (реже). Активность максимальная в понедельник, к воскресенью уменьшается. Посещает либо один сайт, либо примерно 8.
- Bob активен по будням, в основном в пн, чт и пт, большая активность с утра. Не посещает сайты в выходные дни. В среднем посещает 6 сайтов за сессию.
- Dick активен в течение всего дня, больше всего сб и ср. Сессии часто состоят из 2 уникальных сайтов.
- Ed более активен в середине дня. Активен в течение всей недели, но более всего в ср, сб и вс. Сессии по 6-8 уникальных сайтов
- John наиболее активен в 12-15 часов. Наибольшая активность в будни, особенно в чт и пт. Сессии преимущественно по 7-8 сайтов.
- Lindsey наиболее активна в 11-16 часов. Сидит в интернете всю неделю, но наибольшая активность во вт и ср. Преимущественно 7-8 сайтов в сессии.
- Mary-Kate активна в течение всего дня, к вечеру активность увеличивается. Активна в течение всей недели, более всего - в выходные. Чаще встречаются сессии из 2 уникальных сайтов
- Naomi активна в течение дня но больше всего в середине. Активна всю неделю, наиболее активна в среду. Сессии в среднем состоят из 6-8 уникальных сайтов.

4. Сравнение алгоритмов классификации

Разобьем выборку на 2 части. На одной будем проводить кросс-валидацию, на второй - оценивать модель, обученную после кросс-валидации. Зададим заранее тип кросс-валидации: 3-кратная, с перемешиванием.

Обучим KNeighborsClassifier со 100 ближайшими соседями и посмотрим на долю правильных ответов на 3-кратной кросс-валидации по выборке (X_train, y_train) и отдельно на выборке (X_valid, y_valid). Доли правильных ответов для KNeighborsClassifier на кросс-валидации и отложенной выборке равны 0.563 и 0.587 соответственно.

Обучим случайный лес (RandomForestClassifier) из 100 деревьев. Посмотрим на OOB-оценку и на долю правильных ответов на выборке (X_valid, y_valid). Они равны 0.724 и 0.731.

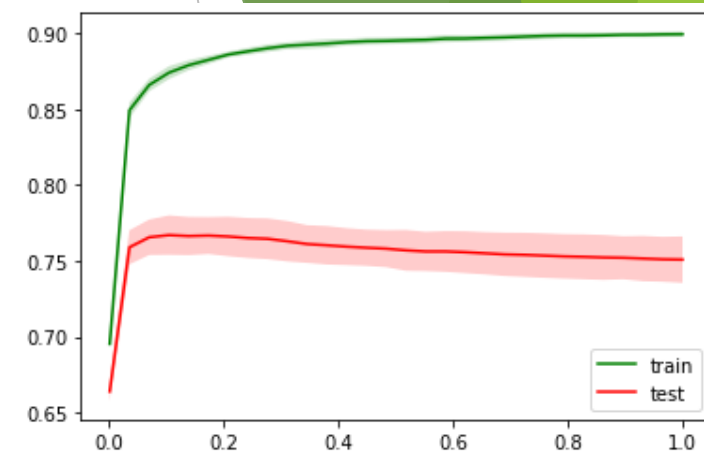
Обучим логистическую регрессию (LogisticRegression) с параметром C по умолчанию. Доли правильных ответов на кросс-валидации и на выборке (X_valid, y_valid) равны 0.761 и 0.777.

Предварительно можно видеть, что логистическая регрессия показывает лучший результат при заданных параметрах.

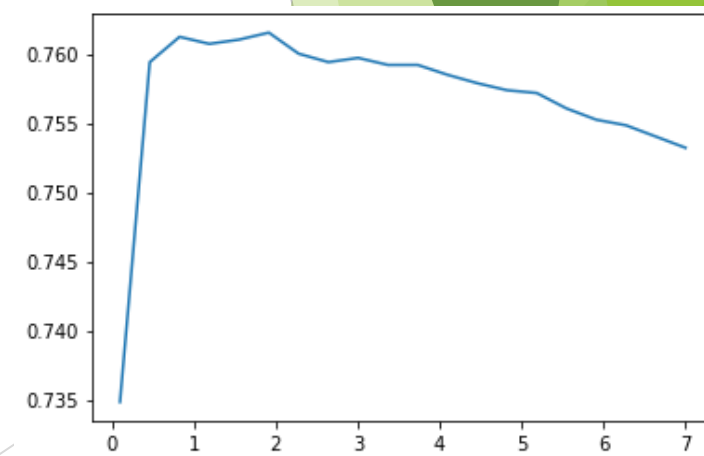
С помощью LogisticRegressionCV подберем параметр C для LogisticRegression. Оно равно 1.916 и доля правильных ответов на кросс-валидации равна 0.762. Доля правильных ответов на выборке (X_valid, y_valid) равна 0.779.

Обучим линейный SVM (LinearSVC) с параметром C=1. Доли правильных ответов на кросс-валидации и отложенной выборке равны 0.751 и 0.777.

С помощью GridSearchCV подберем параметр C для SVM. Найденное значение равно 0.104, доля правильных ответов на кросс-валидации 0.767. Доля правильных ответов на выборке (X_valid, y_valid) равна 0.781.



Кривая валидации для LinearSVC







Зависимость доли правильных ответов на кросс валидации от параметра C для логистической регрессии

5. Соревнование Kaggle Inclass по идентификации пользователей

Мы познакомимся с данными [соревнования](#) Kaggle по идентификации пользователей и сделаем в нем первые посылки. В обучающей выборке видим следующие признаки: - sitei - индекс i-го посещенного сайта в сессии - timei - время посещения i-го сайта в сессии - user_id - ID пользователя. Сессии пользователей выделены так, что они не могут быть длиннее получаса или 10 сайтов. То есть сессия считается оконченной либо когда пользователь посетил 10 сайтов подряд, либо когда сессия заняла по времени более 30 минут.

Для первого прогноза будем использовать только индексы посещенных сайтов. Создадим разреженные матрицы аналогично тому, как мы это делали ранее. Используем SGDClassifier с логистической функцией потерь. Обучим модель на выборке (X_train, y_train). Посчитаем ROC AUC на отложенной выборке. Он равен 0.934.

Сделаем прогноз для тестовой выборки. Бейзлайн "SGDClassifier" на лидерборде побит.

3610	Manmmu Marovek		0.91646	1	9d
3611	[YDF & MIPT] Andrei Ruslantsev		0.91646	1	~10s
Your First Entry ↑ Welcome to the leaderboard!					
3612	[YDF & MIPT] Sergey Dyachenko		0.91646	4	10mo
3613	[YDF & MIPT] Coursera_Annen...		0.91646	2	8mo

5. Соревнование Kaggle Inclass по идентификации пользователей

Добавим признаки из ноутбука для третьей недели, попробуем оптимизировать гиперпараметры.




Обучим логистическую регрессию.

ROC AUC на отложенной выборке равен 0.979, что выше, чем в предыдущем случае.

Обучим классификатор на всех данных. Сделаем прогноз для тестовой выборки и отправим его.

Бенчмарк "Logit +3 features" со скором 0.92784 побит.

Результат можно еще улучшить, применяя другие модели, придумывая дополнительные признаки и т.п. Как минимум стоит попробовать бустинг или случайный лес.

2708	Дмитрий Белов		0.93445	3	1y
2709	[YDF & MIPT] Andrei Ruslantsev		0.93440	3	2m
Your Best Entry ↑					
Your submission scored 0.93440, which is an improvement of your previous score of 0.91646. Great job!					
2710	Ihar Malkevich		0.93438	22	3y

6. Vowpal wabbit

Исходные данные все те же самые, но выделено 400 пользователей, и решается задача их идентификации.

Vowpal Wabbit любит, чтоб метки классов были распределены от 1 до K , где K - число классов в задаче классификации (в нашем случае - 400). Поэтому пришлось применить LabelEncoder, да еще и +1 потом добавить (LabelEncoder переводит метки в диапазон от 0 до $K-1$).

Были обучены модели Vowpal Wabbit, LogisticRegression и SGDClassifier на обучающей выборке и сделаны прогнозы на тестовой выборке. На публичной части тестовой выборки получились следующие доли правильных ответов:

- 0.194 для Vowpal Wabbit
- 0.188 для SGDClassifier
- 0.199 для Логистической регрессии

Выводы по курсу

В рамках выполнения проекта по идентификации интернет-пользователей были изучены и использованы такие вещи как

- разреженные матрицы
- визуальный анализ данных
- поиск гиперпараметров по заданной сетке
- частотные словари
- мешок слов

В результате проведен визуальный анализ данных, построены модели классификаторов, выбрана лучшая из них, подобраны оптимальные параметры для модели и произведена оценка качества выбранной модели.