Мы будем решать похожую задачу: по последовательности из нескольких веб-сайтов, посещенных подряд одним 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, если это возможно).

Будем использовать данные из [статьи](http://ceur-ws.org/Vol-1703/paper12.pdf) "A Tool for Classification of Sequential Data". И хотя мы не можем рекомендовать эту статью (описанные методы далеки от state-of-the-art, лучше обращаться к [книге](http://www.charuaggarwal.net/freqbook.pdf) "Frequent Pattern Mining" и последним статьям с ICDM), но данные там собраны аккуратно и представляют интерес.

Имеются данные с прокси-серверов Университета Блеза Паскаля, их вид очень простой: ID пользователя, timestamp, посещенный веб-сайт.

Скачать исходные данные можно по ссылке в статье (там же описание), для этого задания хватит данных не по всем 3000 пользователям, а по 10 и 150. [Ссылка](https://drive.google.com/open?id=11AqEDEITiodB8fcB8IZvp_5odfZuqAsS) на архив *capstone\_user\_identification.zip* (~7 Mb, в развернутом виде ~60 Mb).

В ходе выполнения проекта вас ожидает 4 задания типа Programming Assignment, посвященных предобработке данных, первичному анализу, визуальному анализу данных, сравнению моделей классификации и настройке выбранной модели и изучению ее переобучения. Также у вас будет 3 взаимно оцениваемых задания (Peer Review) – по визуализации данных (в том числе со свеже созданными признаками), по оценке результатов участия в [соревновании](https://inclass.kaggle.com/c/catch-me-if-you-can-intruder-detection-through-webpage-session-tracking2) Kaggle Inclass и по всему проекту в целом.

В ходе проекта мы будем работать с библиотекой Vowpal Wabbit. Если будут проблемы с ее установкой, можно воспользоваться Docker-образом, например, тем, что описан в [Wiki](https://github.com/Yorko/mlcourse_open/wiki/%D0%9F%D0%9E-%D0%B4%D0%BB%D1%8F-%D0%BF%D1%80%D0%BE%D1%85%D0%BE%D0%B6%D0%B4%D0%B5%D0%BD%D0%B8%D1%8F-%D0%BA%D1%83%D1%80%D1%81%D0%B0-%D0%B8-Docker) репозитория [открытого курса](https://github.com/Yorko/mlcourse_open) OpenDataScience по машинному обучению.

План проекта такой:

**1 неделя.**Подготовка данных к анализу и построению моделей. Programming Assignment

Первая часть проекта посвящена подготовке данных для дальнейшего описательного анализа и построения прогнозных моделей. Надо будет написать код для предобработки данных (исходно посещенные веб-сайты указаны для каждого пользователя в отдельном файле) и формирования единой обучающей выборки. Также в этой части мы познакомимся с разреженным форматом данных (матрицы Scipy.sparse), который хорошо подходит для данной задачи.

* Подготовка обучающей выборки
* Работа с разреженным форматом данных

**2 неделя.**Подготовка и первичный анализ данных. Programming Assignment

На второй неделе мы продолжим подготовливать данные для дальнейшего анализа и построения прогнозных моделей. Конкретно, раньше мы определили что сессия – это последовательность из 10 посещенных пользователем сайтов, теперь сделаем длину сессии параметром, и потом при обучении прогнозных моделей выберем лучшую длину сессии. Также мы познакомимся с предобработанными данными и статистически проверим первые гипотезы, связанные с нашими наблюдениями.

* Подготовка нескольких обучающих выборок для сравнения
* Первичный анализ данных, проверка гипотез

**3 неделя.**Визуальный анализ данных построение признаков. Peer-Review

На 3 неделе мы займемся визуальным анализом данных и построением признаков. Сначала мы вместе построим и проанализируем несколько признаков, потом Вы сможете сами придумать и описать различные признаки. Задание имеет вид Peer-Review, так что творчество здесь активно приветствуется. Если задействуете IPython-виджеты, библиотеку Plotly, анимации и прочий интерактив, всем от этого будет только лучше.

* Визуальный анализ данных
* Построение признаков

**4 неделя**. Сравнение алгоритмов классификации. Programming Assignment

Тут мы наконец подойдем к обучению моделей классификации, сравним на кросс-валидации несколько алгоритмов, разберемся, какие параметры длины сессии (session\_length и window\_size) лучше использовать. Также для выбранного алгоритма построим кривые валидации (как качество классификации зависит от одного из гиперпараметров алгоритма) и кривые обучения (как качество классификации зависит от объема выборки).

* Сравнение нескольких алгоритмов на сессиях из 10 сайтов
* Выбор параметров – длины сессии и ширины окна
* Идентификация конкретного пользователя и кривые обучения

**5 неделя.**СоревнованиеKaggle Inclass по идентификации пользователей. Peer-Review

Здесь мы вспомним про концепцию стохастического градиентного спуска и попробуем классификатор Scikit-learn SGDClassifier, который работает намного быстрее на больших выборках, чем алгоритмы, которые мы тестировали на 4 неделе. Также мы познакомимся с данными [соревнования](https://inclass.kaggle.com/c/catch-me-if-you-can-intruder-detection-through-webpage-session-tracking2) Kaggle по идентификации пользователей и сделаем в нем первые посылки. По итогам этой недели дополнительные баллы получат те, кто побьет указанные в соревновании бенчмарки.

**6 неделя**. Vowpal Wabbit. Tutorial + Programming Assignment

На этой неделе мы познакомимся с популярной библиотекой Vowpal Wabbit и попробуем ее на данных по веб-сессиям. Знакомиться будем на данных Scikit-learn по новостям, сначала в режиме бинарной классификации, затем – в многоклассовом режиме. Затем будем классифицировать рецензии к фильмам с сайта IMDB. Наконец, применим Vowpal Wabbit к данным по веб-сессиям. Материала немало, но Vowpal Wabbit того стоит!

* Статья про Vowpal Wabbit
* Применение Vowpal Wabbit к данным по посещению сайтов

**7 неделя**. Оформление финального проекта. Peer-Review

В самом конце Вас ожидает взаимная проверка финальных версий проекта. Здесь можно будет разгуляться, поскольку свобода творчества есть на каждом этапе проекта: можно использовать все исходные данные по 3000 пользователям, можно создавать свои интересные признаки, строить красивые картинки, использовать свои модели или ансамбли моделей и делать выводы. Поэтому совет такой: по мере выполнения заданий параллельно копируйте код и описание в .ipynb-файл проекта или описывайте результаты по ходу в текстовом редакторе.