# Проект "Идентификация пользователей по посещенным веб-страницам". Финальный отчет.

В этом проекте на протяжении нескольких недель изучались пользовательские данные по посещенным сайтам, и на их основе делались выводы о некоторых особенностях поведения пользователей в интернете, позволяющие идентифицировать их.

Исходные данные имели следующий вид:

In [1]:

```
#импортируем нужные модули
from __future__ import division, print_function
import warnings
warnings.filterwarnings('ignore')
from glob import glob
import os
import pickle
from tqdm import tqdm_notebook
import numpy as np
import pandas as pd
from scipy.sparse import csr_matrix
```

In [2]:

```
PATH_TO_DATA1 = '/home/satanklaus/Python&ML/Python coursera/6 course/capstone_user_iden
tification'
user31_data = pd.read_csv(os.path.join(PATH_TO_DATA1,
                                       '10users/user0031.csv'))
#Посмотрим на данные одного пользователя
user31_data.head()
```

Out[2]:

| | timestamp | site |
|---|---|---|
| **0** | 2013-11-15 08:12:07 | fpdownload2.macromedia.com |
| **1** | 2013-11-15 08:12:17 | laposte.net |
| **2** | 2013-11-15 08:12:17 | www.laposte.net |
| **3** | 2013-11-15 08:12:17 | www.google.com |
| **4** | 2013-11-15 08:12:18 | www.laposte.net |

Как видно, нам доступны данные о посещенной вебстранице и времени ее посещения. Были реализованы функции, объединяющие данные из файлов в общую выборку по сессиям, для которых мы указывали различные параметры:

- длина сессий по количеству сайтов (отрезки по 5,7,10 и 15 сайтов);
- скользящее окно для длины сессии (сессии могут перекрываться);
- некоторые из функций переводили сессионную статистику пользователей в разреженный формат sparse matrix;
- подготовка различных признаков на основе имеющихся данных.

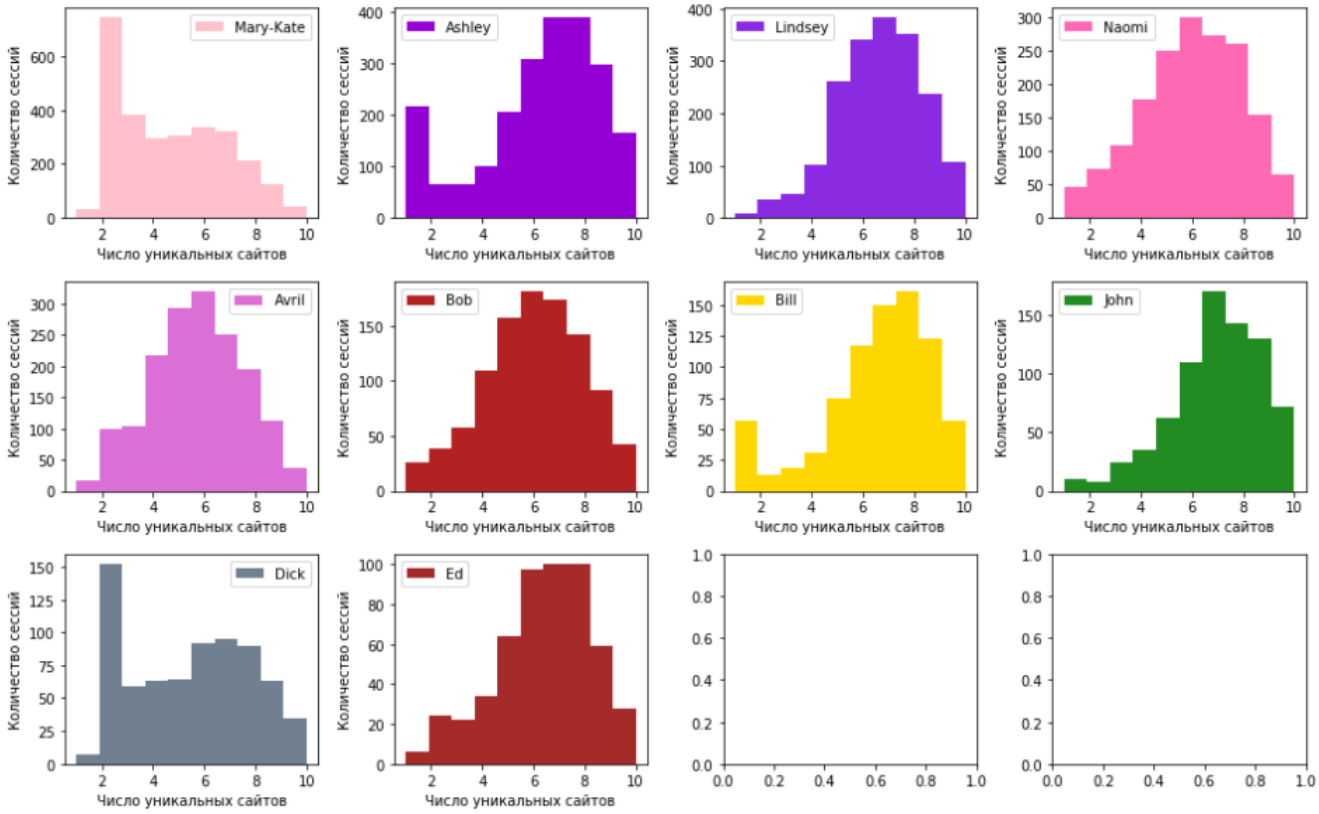
Вот таким образом выглядел один результатов реализации функции для данных 10 пользователей:

| | site1 | site2 | site3 | site4 | site5 | site6 | site7 | site8 | site9 | site10 | session_timespan | #unique_sites | start_hour | day_of_week | target |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 2 | 3 | 3 | 6 | 38 | 13 | 3 | 6 | 2 | 19 | 6 | 6 | 16 | 6 | 237 |
| **1** | 38 | 3 | 19 | 38 | 38 | 13 | 13 | 38 | 1 | 8 | 3 | 6 | 16 | 6 | 237 |
| **2** | 3 | 6 | 1 | 38 | 15 | 15 | 38 | 3 | 38 | 2 | 5 | 6 | 16 | 6 | 237 |
| **3** | 38 | 1601 | 19 | 13 | 38 | 19 | 19 | 13 | 19 | 14 | 11 | 5 | 16 | 6 | 237 |
| **4** | 38 | 1 | 13 | 38 | 19 | 3 | 13 | 859 | 138 | 13 | 3 | 7 | 16 | 6 | 237 |

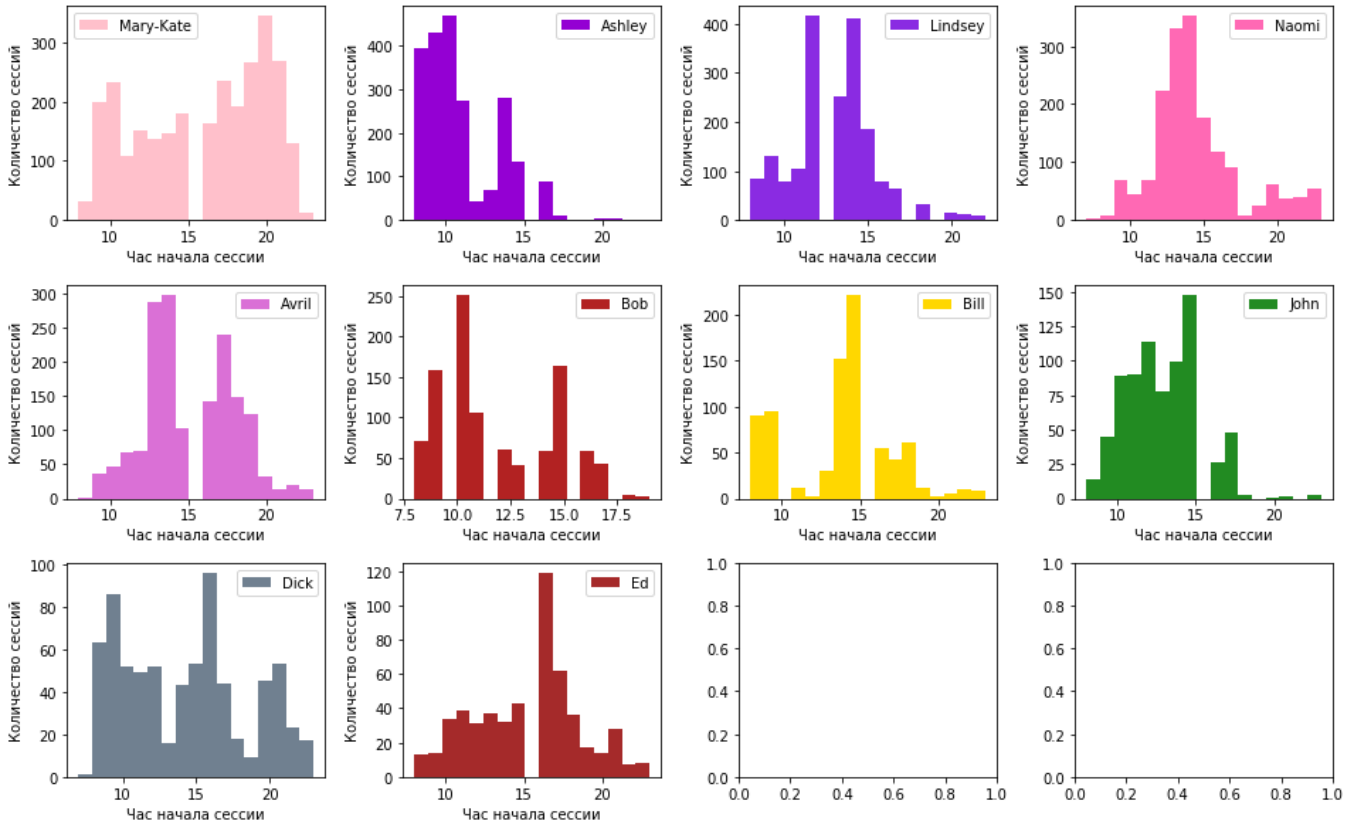
1) site1...site10 - посещенные номера сайтов в сессии (был составлен словарь для всех посещенных сайтов 10 пользователей, каждому сайту присвоен свой номер);
2) session_timespan - длина сессии в секундах;
3) #unique_sites - числой различных сайтов в сессии;
4) start hour - час начала сессии;
5) day_of_week - день недели начала сессии;
6) target - id пользователя.

Используя данных 10 пользователей (каждому было дано условное имя), был проведен визуальный анализ полученных признаков для выявления особенностей поведения пользователей:

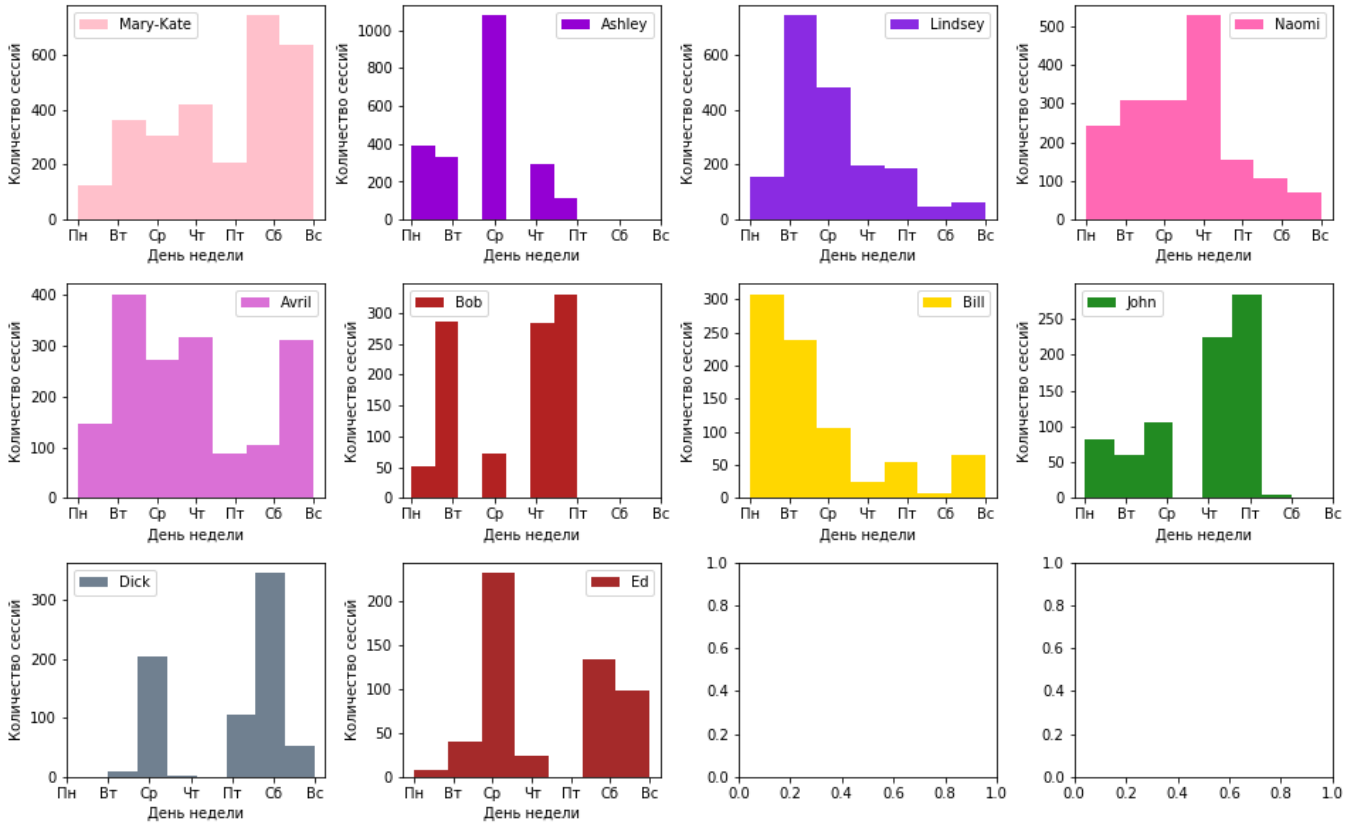
*Распределение числа уникальных сайтов по пользователям*




*Распределение часа начала сессии по пользователям*



# *Распределение дней недели по пользователям*

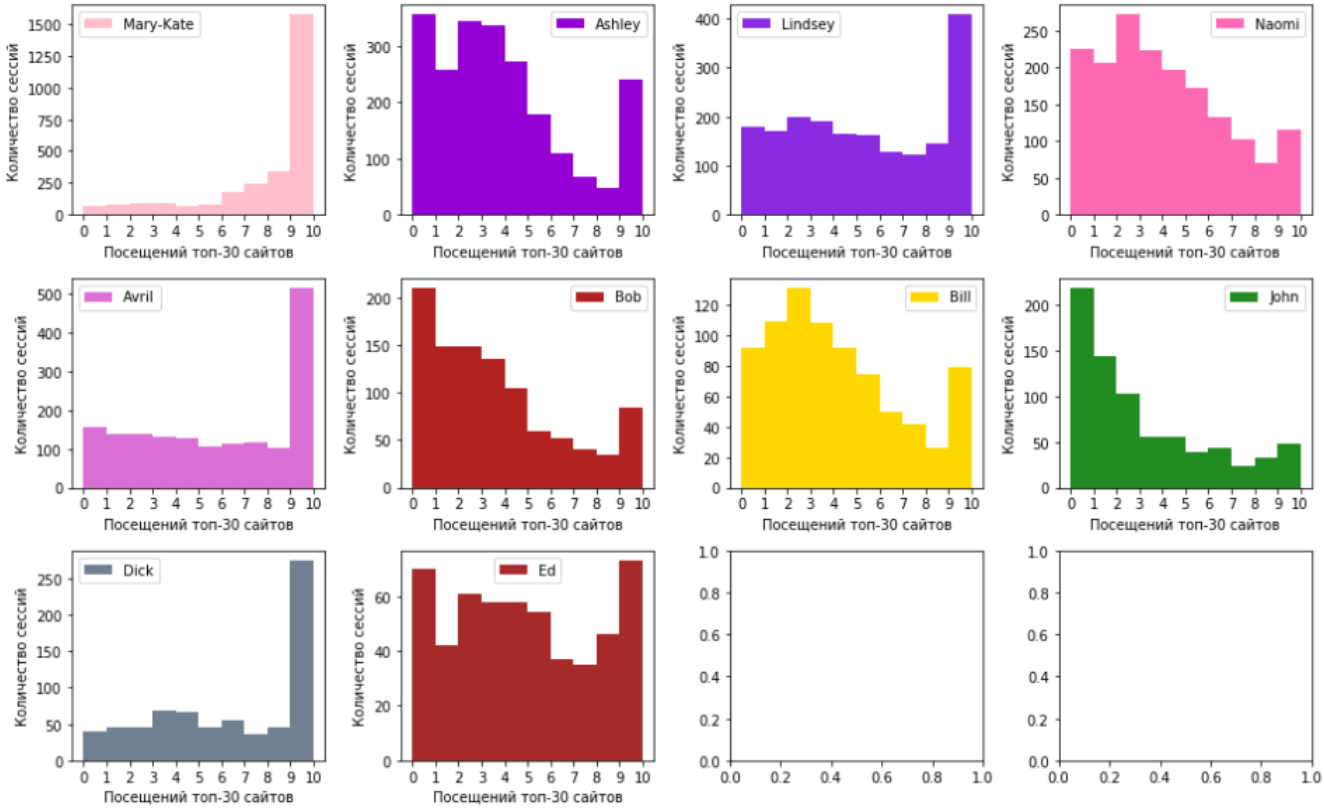


На основании графиков были сделаны следующие выводы:

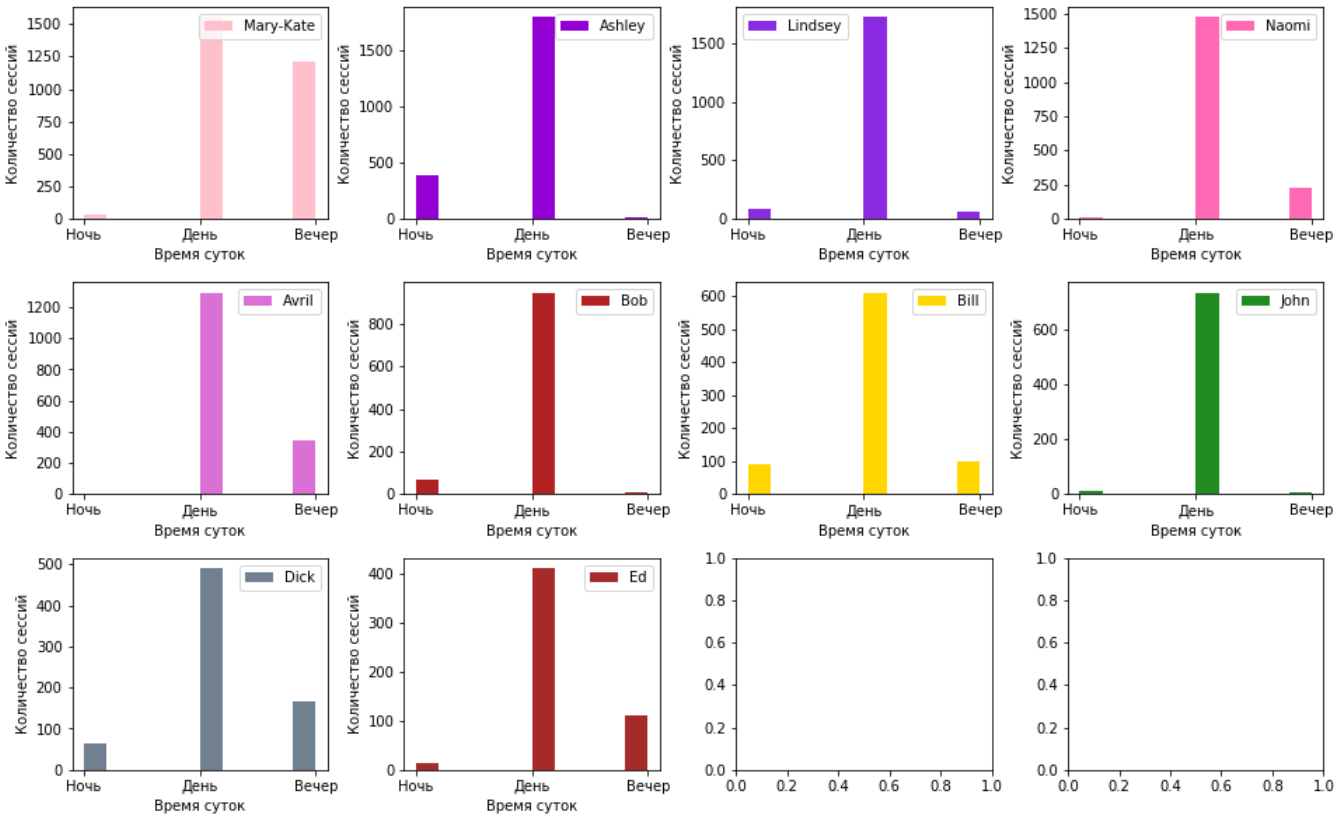
1. Mary-Kate - работает преимущественно с небольшим числом сайтов в сессии, в вечернее время и по субботам и воскресеньям.
2. Ashley - работает преимущественно с большим (7-8) количеством сайтов, иногда только с 1 сайтом, исключительно в рабочие дни и в первой половине дня.
3. Lindsey - работает преимущественно с большим (7-8) количеством сайтов, в основном в рабочие дни в середине дня.
4. Naomi - работает преимущественно с большим (7-8) количеством сайтов,в основном в пн-чт в середине дня.
5. Avril - работает преимущественно со средним (5-6) количеством сайтов, по рабочим и выходным в середине дня.
6. Bob - работает преимущественно со средним (5-8) количеством сайтов, исключительно по рабочим дням в первой половине дня
7. Bill - работает преимущественно с большим (6-9) количеством сайтов, в основном в начале рабочей недели в первой половине дня
8. John - работает преимущественно с большим (7-9) количеством сайтов, практически только по рабочим дням в конце рабочей недели в первой половине дня
9. Dick - работает преимущественно с большим (6-8) количеством сайтов, иногда только с 2 сайтами, в основном по ср и сб равномерно в течение дня
10. Ed - работает преимущественно с большим (6-8) количеством сайтов, в основном по ср, сб и вс в равномерно с преобладанием второй половины дня.

Также были построены дополнительные признаки и проведен аналогичный визуальный анализ:

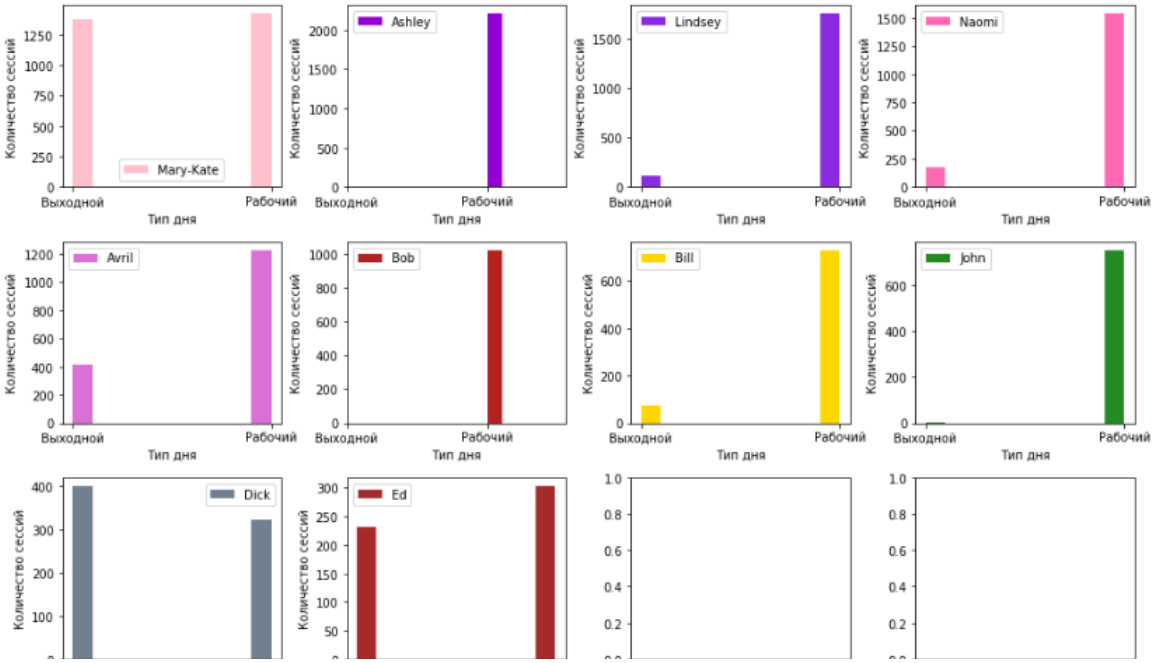
*Распределение посещений топ-30 сайтов по пользователям*




*Распределение сессий по времени суток по пользователям*



*Распределение сессий по рабочим/выходным дням по пользователям*




На основании графиков можно сделать вывод, что все пользователи имеют свои особенности поведения в интернете, на основе которых их можно идентифицировать.

Следующим этапом проекта было участие в [соревновании (https://inclass.kaggle.com/c/catch-me-if-you-can-intruder-detection-through-webpage-session-tracking2)](), в котором необходимо было среди всех сессий различных пользователей научиться определять сессии одного из них, которому дали имя Alice.

In [39]:

```
#загрузим данные соревнования и проведем небольшие преобразования
PATH_TO_DATA = '/home/satanklaus/Python&ML/Python coursera/6 course/5 week/competition
 data'
SITE_COL = ['site%d' % i for i in range(1, 11)]
TIME_COL = ['time%d' % i for i in range(1, 11)]
train_df = pd.read_csv(os.path.join(PATH_TO_DATA, 'train_sessions.csv'), index_col='ses
sion_id', parse_dates=TIME_COL)
test_df = pd.read_csv(os.path.join(PATH_TO_DATA, 'test_sessions.csv'), index_col='sessi
on_id',parse_dates=TIME_COL)
test_df['month'] = test_df['time1'].dt.month
test_df['year'] = test_df['time1'].dt.year
test_df['day'] = test_df['time1'].dt.day
train_df['month'] = train_df['time1'].dt.month
train_df['year'] = train_df['time1'].dt.year
train_df['day'] = train_df['time1'].dt.day
```

In [43]:

```
#Рассмотрим распределение дней по месяцам в обучающей выборке
for i in range(1,13):
    print('month = ', i ,'\n', train_df[train_df['year']==2013][train_df['month']==i].d
ay.unique())
```

```
month =  1
 [12]
month =  2
 [12]
month =  3
 [12]
month =  4
 [12]
month =  5
 [12]
month =  6
 [12]
month =  7
 [12]
month =  8
 [12]
month =  9
 [12]
month =  10
 [12]
month =  11
 [21 29 25 27 30 12 20 26 15 19 22 24 28 18 17 23 16]
month =  12
 [16 12 17 18 19 13 20 14 15 21 24 28 23 22 26 27 31]
```

In [44]:

```
#и распределение дней по месяцам в обучающей выборке
for i in range(1,11):
    print('month = ', i ,'\n', test_df[test_df['month']==i].day.unique())
```

```
month =  1
 []
month =  2
 []
month =  3
 []
month =  4
 []
month =  5
 [16 21 25  2 27  3 19  4 28 20 13 14  5 15 23 24 22 26 18 17  1]
month =  6
 [5 2 1 4]
month =  7
 [3 1 2 4 5]
month =  8
 [4 1 3 2 5]
month =  9
 [4 1 3 2 5]
month =  10
 [4 2 5 1 3]
```

Как видно, распределение дней в некоторые месяца очень подозрительно и наводит на мысль, о том, что парсер собирал данные с ошибкой, и необходимо поменять в датах сессий дни и месяцы местами для дней<=12.

Это позволит нам корректно отсортировать данные сессий по времени и использовать кроссвалидацию по времени,а также заняться предварительным анализом данных для настройки признаков для модели.

В обучающей выборке объекты класса Alice имеют метку '1'.

In [46]:

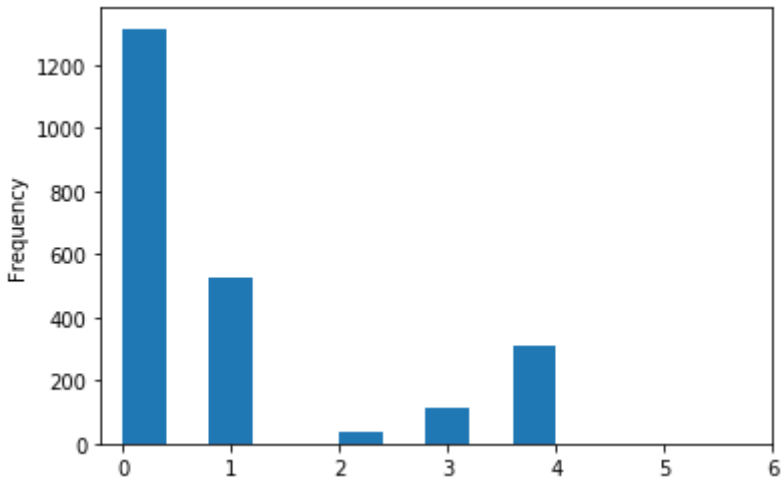
```
sum(train_df.target),train_df.shape[0]
```

Out[46]:

```
(2297, 253561)
```

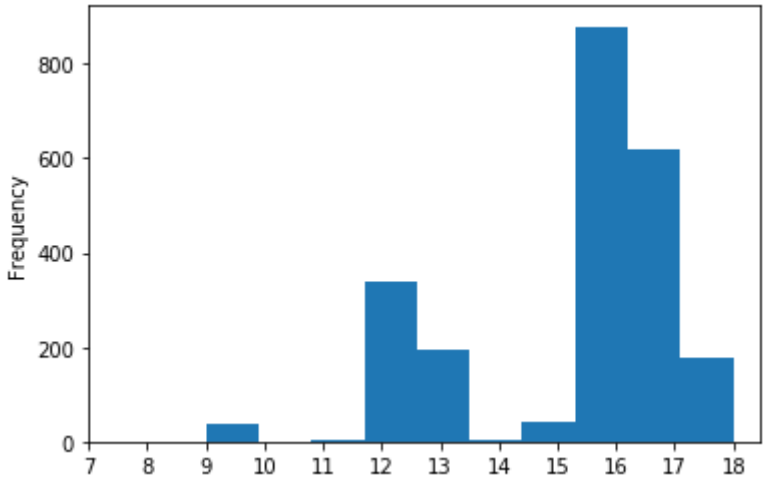
Выборка не сбалансирована по классам, следовательно метрика accuracy будет непоказательна; будем использовать показатель roc-auc.

Посмотрим на дни недели активности Alice (0 - пн, 6 - вскр):




Добавим к разреженной матрице посещенных сайтов по сессиям первый бинарный признак: равен ли день недели 0,1,3,4.

Посмотрим на часы активности Alice:




Добавим следующий признак: равен ли час начала сессии 12,13,15...18.
Последним признаком в модель для обучения добавим отмасштабированную при помощи StandardScaler длину сессии.
Обучим на полученных данных логистическую регрессию с параметрами по умолчанию:

```
logit = LogisticRegression(C=1, random_state=17, solver='liblinear')
time_split = TimeSeriesSplit(n_splits=10)
cv_scores8 = cross_val_score(logit, X_trans_train, y, cv=time_split,
                             scoring='roc_auc', n_jobs=-1)
cv_scores8, cv_scores8.mean(), cv_scores8.std()
```

```
(array([0.77375168, 0.95897322, 0.96273524, 0.9102824 , 0.95382424,
        0.98500259, 0.92092357, 0.9716455 , 0.98363604, 0.9664274 ]),
 0.9387201887976296,
 0.05963751927888849)
```

Данная модель получила следующие показатели на лидерборде:

| Submission and Description | Public Score |
|---|---|
| fe_an_5th.csv<br>7 days ago by Satanklaus | 0.95659 |

что, является вполне приемлемым показателем в рамках прохождения курса.

Что можно улучшить в дальнейшем для увеличения скора:

1. Настройка параметров модели (например, коэффициента регуляризации C);
2. Добавление признаков к модели, основанных на посещении популярных для Alice сайтов;
3. Использование "стакинга".

## Это соревнование, анализ его данных, построение "правильных" признаков для модели являлось наиболее ценной практической частью данного проекта.

# Результаты проделанной в проекте работы:

1. Изучены методы анализа данных применительно к статистике поведения пользователей в Интернете;
2. Получено понимание важности предварительного визуального анализа данных для успешности настройки модели;
3. Изучены некоторые технические особенности работы с данными, имеющими временной характер;
4. Данная модель и принципы, заложенные в ее основу, могут быть использованы для выявления несанкционированного доступа к компьютеру/аккаунтам пользователя и своевременного выявления мошенника.