Web Mining – CSE 3024

Sessionization using Python

By Abhijeet Ambadekar (16BCE1156)

Q1. Following is a set of web usage logs from an organization. Assume the following rules.

a) The time heuristic h1 for a session duration = 30 minutes.
b) The time heuristic h2 for the average duration for the user visit of a page = 1 minute.
c) Include a page in a session if the page that refers is in that session.
d) During a session, the IP address, browser and OS is the same.

You need to printout a table containing session number, IP address, session start time and session end time.

Q2. Use the same program to find do sessionization of a log file that is downloaded from web that is uploaded into moodle today and give it in same format as Q1.

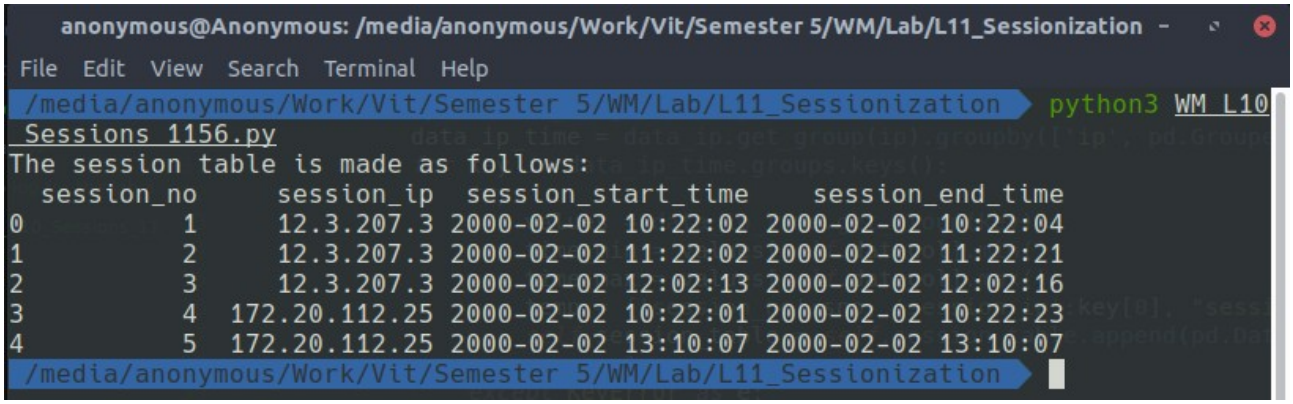
Program Code:

```
import pandas as pd

import datetime


pd.set_option('display.max_columns', None)



class Sessionizer():

      def __init__(self, h1, h2):

            self.h1 = h1

            self.h2 = h2

            self.session_table = pd.DataFrame(columns = ["session_no",
'session_ip', 'session_start_time', 'session_end_time'])

            self.session_table.set_index("session_no")


      def load(self, data, datecol=None):

            if datecol:

                  self.datecol = datecol

            self.data = data



      def parse(self, datecol=None, datetime_format=None):

            # if not (datecol and datetime_format):

                  # datecol = self.datecol

            self.data[datecol] = [datetime.datetime.strptime(dt,
datetime_format) for dt in self.data[datecol]]



      def sessionize(self):

            data_ip = self.data.groupby('ip')

            sno = 1

            for ip in data_ip.groups.keys():
```

```
                  data_ip_time = data_ip.get_group(ip).groupby(['ip',
pd.Grouper(key=self.datecol, freq= str(self.h1)+"Min")])
                  for key in data_ip_time.groups.keys():
                        try:
                              values = data_ip_time.get_group(key)
                              time_min = values[self.datecol].min()
                              time_max = values[self.datecol].max()
                              temp = {'session_no':sno, 'session_ip':key[0],
"session_start_time":time_min, "session_end_time":time_max}
                              self.session_table =
self.session_table.append(pd.DataFrame(temp.copy(), index=[0]), ignore_index =
True, sort=False)
                              sno += 1
                        except KeyError as e:
                              print(e)



      def print_details(self):
            print("The session table is made as follows: ")
            print(self.session_table)


data = pd.read_csv("data.csv")
datecol = "timestamp"
datetime_format = "%d/%b/%Y:%H:%M:%S"



ss = Sessionizer(30, 1)
ss.load(data, datecol)
ss.parse(datecol, datetime_format)
ss.sessionize()
ss.print_details()
```

Output:

Q1.



Q2.

<<Insert output after receiving the moodle log file >>