**WEB MINING**

**SESSIONIZATION**

**16BCE1186**

**HARSH TRIPATHI**

**CODE:**

from csv import reader

from datetime import datetime

class Sessionize(object):

def \_\_init\_\_(self, filename, delta):

"""Function to initialize the different parameters of the object."""

self.delta = delta

csvfile = open(filename, 'r')

self.dataset\_ = list(reader(csvfile))

self.updateOrderingOfEntries()

def separateUsers(self):

"""Function to separate the server log entries based on the user

i.e. on the basis of the IP and user-agent."""

self.separate\_ = {}

for row in self.dataset\_:

if row[0] not in self.separate\_:

self.separate\_[row[0]] = []

self.separate\_[row[0]].append(row[1:])

self.updateTimestamp()

def updateTimestamp(self):

"""Function that updates the timestamp field in a format that

makes its processing by datetime module easy."""

for i in self.separate\_:

for j in self.separate\_[i]:

date\_time = j[0][1:-6]

j[0] = date\_time

def updateOrderingOfEntries(self):

"""Function to sort the entries in ascending order based on the

timestamp using Selection Sort."""

for i in range(len(self.dataset\_)):

min\_idx = i

t1 = datetime.strptime(self.dataset\_[i][1][1:-6],

"%d/%b/%Y:%H:%M:%S")

for j in range(i+1, len(self.dataset\_)):

t2 = datetime.strptime(self.dataset\_[j][1][1:-6],

"%d/%b/%Y:%H:%M:%S")

if t1 > t2:

min\_idx = j

self.dataset\_[i], self.dataset\_[min\_idx] = self.dataset\_[min\_idx], self.dataset\_[i]

def createSession(self):

"""Function to create session for each user based on the different

rules of sessionization."""

self.sessions\_ = {}

for i in self.separate\_:

if i not in self.sessions\_:

self.sessions\_[i] = []

for j in range(len(self.separate\_[i])):

temp = []

present = False

for l in self.sessions\_[i]:

if self.separate\_[i][j] in l:

present = True

if not present:

temp.append(self.separate\_[i][j])

for k in range(j + 1, len(self.separate\_[i])):

t1 = datetime.strptime(self.separate\_[i][j][0],

"%d/%b/%Y:%H:%M:%S")

t2 = datetime.strptime(self.separate\_[i][k][0],

"%d/%b/%Y:%H:%M:%S")

latest = max((t1, t2))

old = min((t1, t2))

difference = latest - old

if(difference.seconds <= self.delta):

temp.append(self.separate\_[i][k])

self.sessions\_[i].append(temp)

def printSessions(self):

"""Function to print the sessions per user."""

session\_id = 1

print('%s' % ('-' \* 93))

print('| {:^20} | {:^20} | {:^20} | {:^20} |'.format("Session Id",

"IP address", "Start Time", "End Time"))

print('%s' % ('-' \* 93))

for i in self.sessions\_:

for l in self.sessions\_[i]:

dates = []

for row in l:

dates.append(datetime.strptime(row[0],

"%d/%b/%Y:%H:%M:%S"))

print('| {:^20} | {:^20} | {:^20} | {:^20} |'.format(session\_id, i, str(min(dates)), str(max(dates))))

session\_id += 1

print('%s' % ('-' \* 93))

filename = input('Enter the name of the dataset: ')

delta = int(input('Enter delta value (minutes): '))

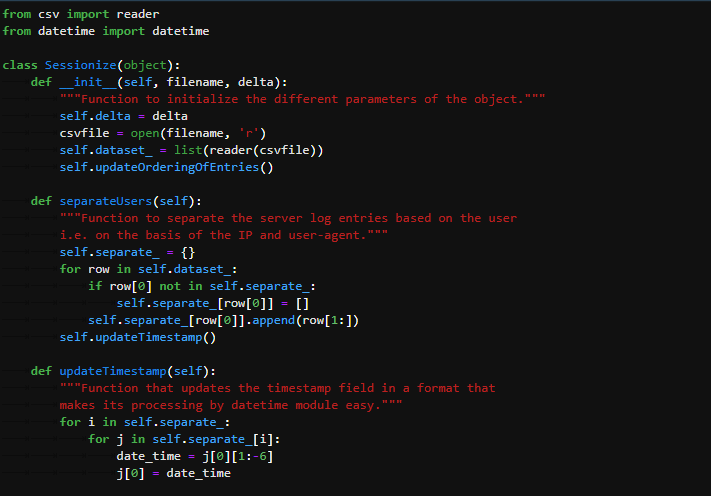
delta \*= 60

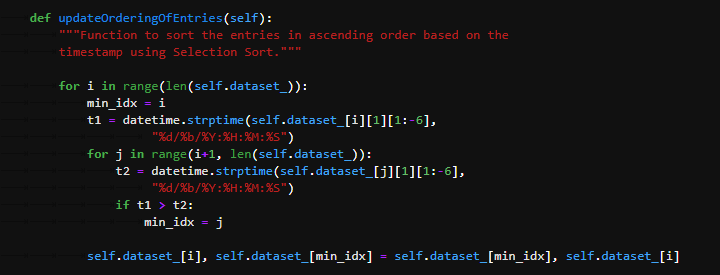
session\_create = Sessionize(filename, delta)

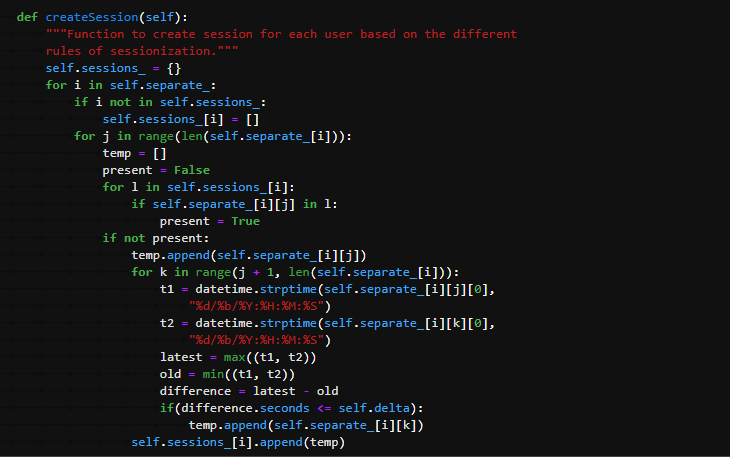
session\_create.separateUsers()

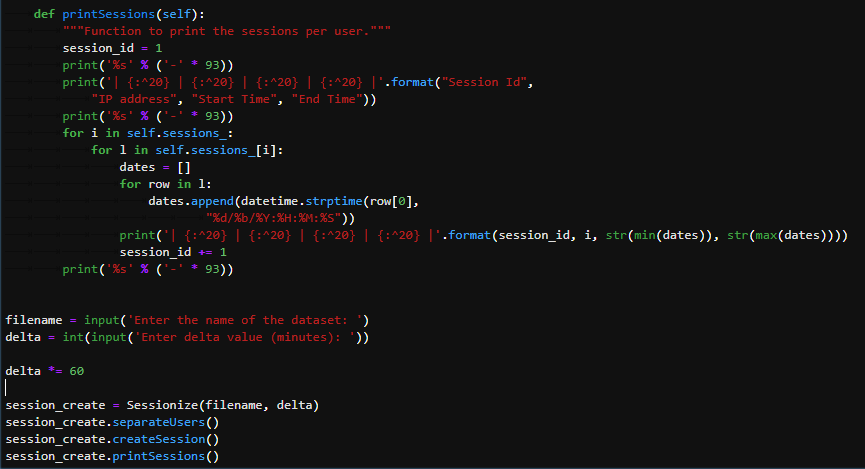
session\_create.createSession()

session\_create.printSessions()









**OUTPUT**

