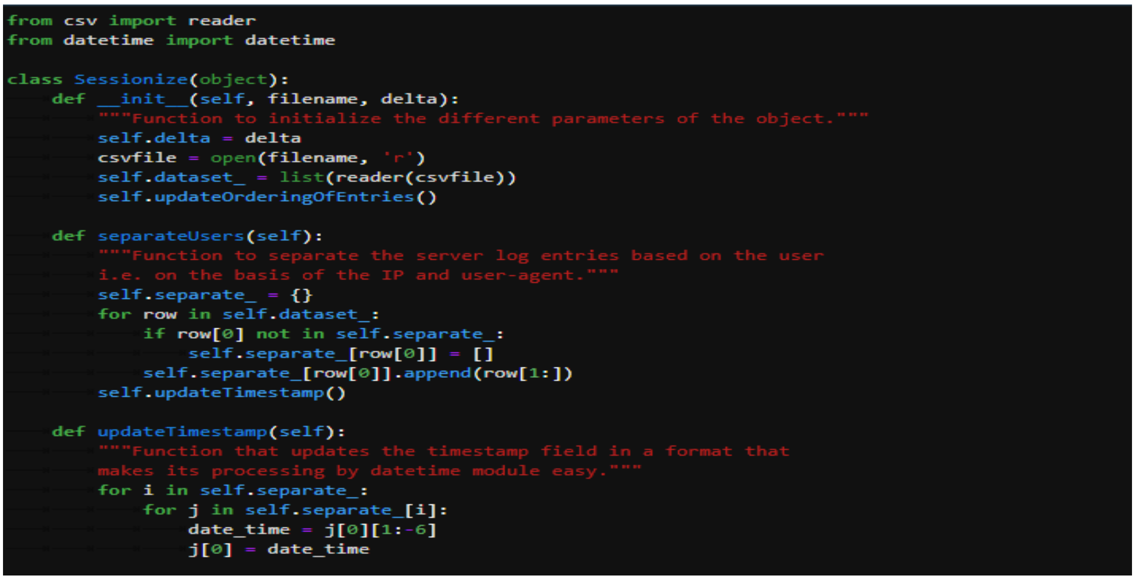
WEB MINING LAB -> SESSIONIZATION

16BCE1184
SOUMOK DUTTA

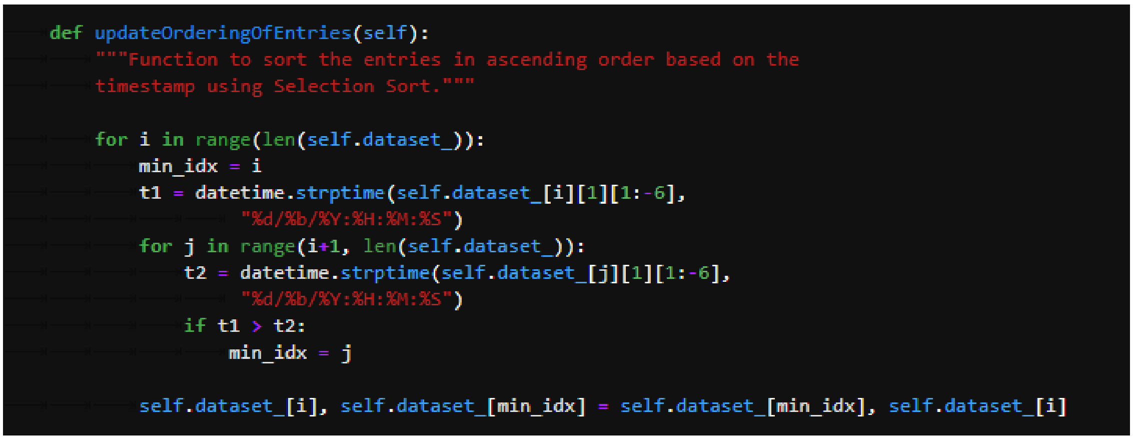
CODE:

```
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
```

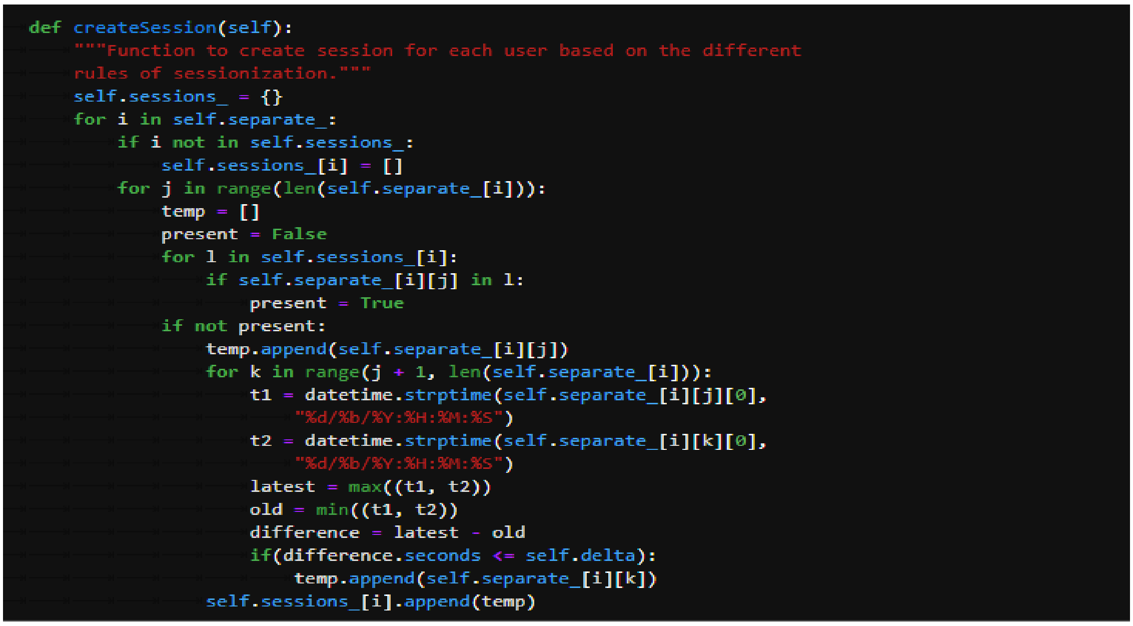
```
            t1 = datetime.strptime(self.dataset_[i][1][1:-6],
                    "%d/%b/%Y:%H:%M:%S")
            for j in range(i+1, len(self.dataset_)):
                t2 = datetime.strptime(self.dataset_[j][1][1:-6],
                    "%d/%b/%Y:%H:%M:%S")
                if t1 > t2:
                    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()
```



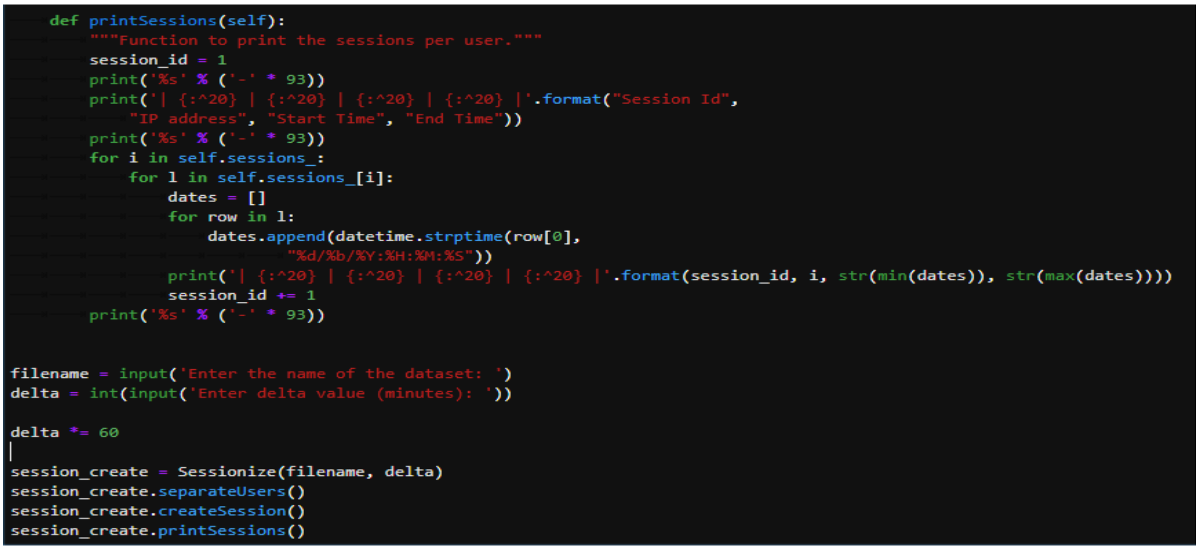
```python
from csv import reader
from datetime import datetime

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
```



```python
    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]
```

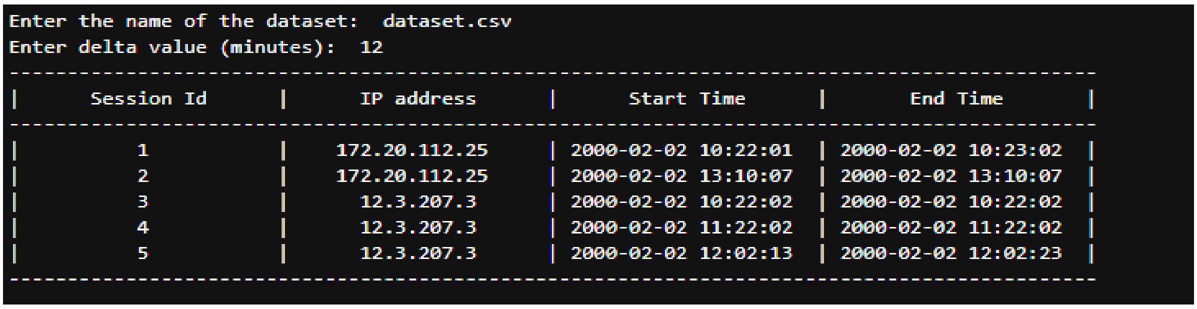


```python
    def createSession(self):
        """Function to create session for each user based on the different
        rules of sessionization."""
        self.sessions_ = {}
        for i in self.separate_:
            if i not in self.sessions_:
                self.sessions_[i] = []
            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)
```

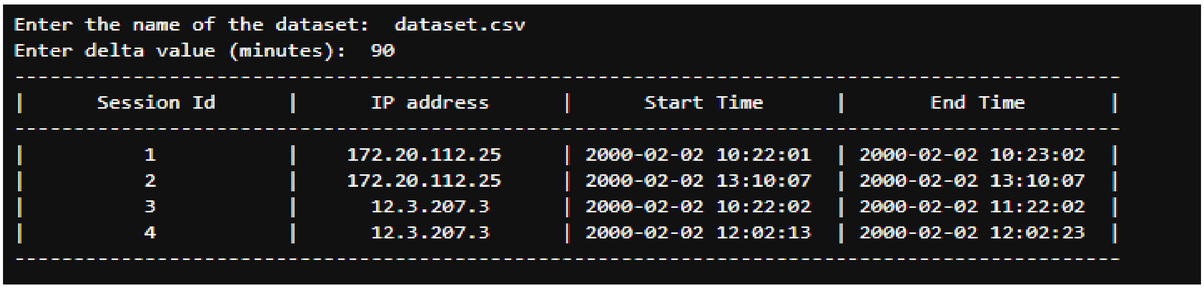


```
    def printSessions(self):
        """Function to print the sessions per user."""
        session_id = 1
        print('%s' % ('-' * 93))
        print('| {:^20} | {:^20} | {:^20} | {:^20} |'.format("Session Id",
            "IP address", "Start Time", "End Time"))
        print('%s' % ('-' * 93))
        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()
```

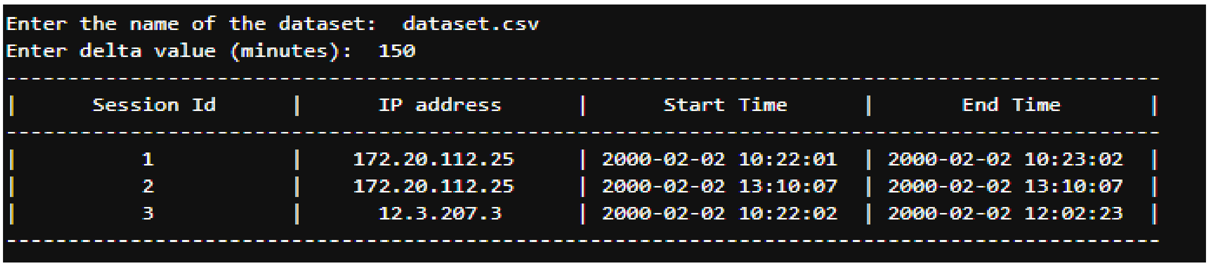
OUTPUT



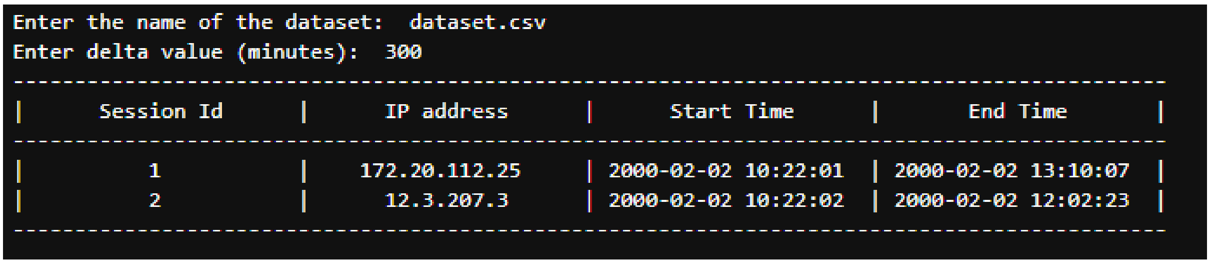
```
Enter the name of the dataset:  dataset.csv
Enter delta value (minutes):  12
---------------------------------------------------------------------------------------------
|      Session Id      |      IP address      |      Start Time      |       End Time       |
---------------------------------------------------------------------------------------------
|          1           |    172.20.112.25     | 2000-02-02 10:22:01  | 2000-02-02 10:23:02  |
|          2           |    172.20.112.25     | 2000-02-02 13:10:07  | 2000-02-02 13:10:07  |
|          3           |      12.3.207.3      | 2000-02-02 10:22:02  | 2000-02-02 10:22:02  |
|          4           |      12.3.207.3      | 2000-02-02 11:22:02  | 2000-02-02 11:22:02  |
|          5           |      12.3.207.3      | 2000-02-02 12:02:13  | 2000-02-02 12:02:23  |
---------------------------------------------------------------------------------------------
```



```
Enter the name of the dataset:  dataset.csv
Enter delta value (minutes):  90
---------------------------------------------------------------------------------------------
|      Session Id      |      IP address      |      Start Time      |       End Time       |
---------------------------------------------------------------------------------------------
|          1           |    172.20.112.25     | 2000-02-02 10:22:01  | 2000-02-02 10:23:02  |
|          2           |    172.20.112.25     | 2000-02-02 13:10:07  | 2000-02-02 13:10:07  |
|          3           |      12.3.207.3      | 2000-02-02 10:22:02  | 2000-02-02 11:22:02  |
|          4           |      12.3.207.3      | 2000-02-02 12:02:13  | 2000-02-02 12:02:23  |
---------------------------------------------------------------------------------------------
```



```
Enter the name of the dataset:  dataset.csv
Enter delta value (minutes):  150
---------------------------------------------------------------------------------------------
|      Session Id      |      IP address      |      Start Time      |       End Time       |
---------------------------------------------------------------------------------------------
|          1           |    172.20.112.25     | 2000-02-02 10:22:01  | 2000-02-02 10:23:02  |
|          2           |    172.20.112.25     | 2000-02-02 13:10:07  | 2000-02-02 13:10:07  |
|          3           |      12.3.207.3      | 2000-02-02 10:22:02  | 2000-02-02 12:02:23  |
---------------------------------------------------------------------------------------------
```



```
Enter the name of the dataset:  dataset.csv
Enter delta value (minutes):  300
---------------------------------------------------------------------------------------------
|      Session Id      |      IP address      |      Start Time      |       End Time       |
---------------------------------------------------------------------------------------------
|          1           |    172.20.112.25     | 2000-02-02 10:22:01  | 2000-02-02 13:10:07  |
|          2           |      12.3.207.3      | 2000-02-02 10:22:02  | 2000-02-02 12:02:23  |
---------------------------------------------------------------------------------------------
```