ADA MidSem P1

- Sampad Kumar Kar
- MCS202215

0. Imports

In [ ]:

```
import os, sys
import pandas as pd
import numpy as np
```

1. Data Loading

In [ ]:

```
data_path = os.path.join('data', 'raw', 'Dataset-Unicauca-Version2-87Atts.csv')

# read the data
df = pd.read_csv(data_path)
```

In [ ]:

```
df.head()
```

Out[ ]:

	Flow.ID	Source.IP	Source.Port	Destination.IP	Destination.Port	Protocol	Timestamp	Flow.Duration	Total.F
0	172.19.1.46-10.200.7.7-52422-3128-6	172.19.1.46	52422	10.200.7.7	3128	6	26/04/201711:11:17	45523	
1	172.19.1.46-10.200.7.7-52422-3128-6	10.200.7.7	3128	172.19.1.46	52422	6	26/04/201711:11:17	1	
2	10.200.7.217-50.31.185.39-38848-80-6	50.31.185.39	80	10.200.7.217	38848	6	26/04/201711:11:17	1	
3	10.200.7.217-50.31.185.39-38848-80-6	50.31.185.39	80	10.200.7.217	38848	6	26/04/201711:11:17	217	
4	192.168.72.43-10.200.7.7-55961-3128-6	192.168.72.43	55961	10.200.7.7	3128	6	26/04/201711:11:17	78068	

5 rows × 87 columns

In [ ]:

```
df.shape
```

Out[ ]:

```
(3577296, 87)
```

In [ ]:

```
df.columns
```

```
Out[ ]: Index(['Flow.ID', 'Source.IP', 'Source.Port', 'Destination.IP',
               'Destination.Port', 'Protocol', 'Timestamp', 'Flow.Duration',
               'Total.Fwd.Packets', 'Total.Backward.Packets',
               'Total.Length.of.Fwd.Packets', 'Total.Length.of.Bwd.Packets',
               'Fwd.Packet.Length.Max', 'Fwd.Packet.Length.Min',
               'Fwd.Packet.Length.Mean', 'Fwd.Packet.Length.Std',
               'Bwd.Packet.Length.Max', 'Bwd.Packet.Length.Min',
               'Bwd.Packet.Length.Mean', 'Bwd.Packet.Length.Std', 'Flow.Bytes.s',
               'Flow.Packets.s', 'Flow.IAT.Mean', 'Flow.IAT.Std', 'Flow.IAT.Max',
               'Flow.IAT.Min', 'Fwd.IAT.Total', 'Fwd.IAT.Mean', 'Fwd.IAT.Std',
               'Fwd.IAT.Max', 'Fwd.IAT.Min', 'Bwd.IAT.Total', 'Bwd.IAT.Mean',
               'Bwd.IAT.Std', 'Bwd.IAT.Max', 'Bwd.IAT.Min', 'Fwd.PSH.Flags',
               'Bwd.PSH.Flags', 'Fwd.URG.Flags', 'Bwd.URG.Flags', 'Fwd.Header.Length',
               'Bwd.Header.Length', 'Fwd.Packets.s', 'Bwd.Packets.s',
               'Min.Packet.Length', 'Max.Packet.Length', 'Packet.Length.Mean',
               'Packet.Length.Std', 'Packet.Length.Variance', 'FIN.Flag.Count',
               'SYN.Flag.Count', 'RST.Flag.Count', 'PSH.Flag.Count', 'ACK.Flag.Count',
               'URG.Flag.Count', 'CWE.Flag.Count', 'ECE.Flag.Count', 'Down.Up.Ratio',
               'Average.Packet.Size', 'Avg.Fwd.Segment.Size', 'Avg.Bwd.Segment.Size',
               'Fwd.Header.Length.1', 'Fwd.Avg.Bytes.Bulk', 'Fwd.Avg.Packets.Bulk',
               'Fwd.Avg.Bulk.Rate', 'Bwd.Avg.Bytes.Bulk', 'Bwd.Avg.Packets.Bulk',
               'Bwd.Avg.Bulk.Rate', 'Subflow.Fwd.Packets', 'Subflow.Fwd.Bytes',
               'Subflow.Bwd.Packets', 'Subflow.Bwd.Bytes', 'Init_Win_bytes_forward',
               'Init_Win_bytes_backward', 'act_data_pkt_fwd', 'min_seg_size_forward',
               'Active.Mean', 'Active.Std', 'Active.Max', 'Active.Min', 'Idle.Mean',
               'Idle.Std', 'Idle.Max', 'Idle.Min', 'Label', 'L7Protocol',
               'ProtocolName'],
              dtype='object')
```

Out of these the following columns are of interest:

- `Flow.ID` : This column likely represents a unique identifier for each network flow in the dataset. It is a nominal attribute that distinguishes different flows.
- `Source.IP` and `Destination.IP` : `Source.IP` contains the IP address of the source of the network flow, while `Destination.IP` contains the IP address of the destination. These are nominal attributes representing network addresses. ' `Source.Port` ' and `Destination.Port` : ' `Source.Port` ' and `Destination.Port` typically represent the source and destination port numbers associated with the network communication. Port numbers are used to identify specific services or processes on a device.
- `Protocol` : This column likely indicates the network protocol used for the communication in each flow. Common values include TCP, UDP, ICMP, etc. This is a categorical or nominal attribute.
- `Timestamp` : This is a date-type attribute and represents the time at which the network flow occurred. It provides the temporal aspect of the flow data.
- `Flow.Duration` : This represents the duration of the network flow in seconds. It measures how long the communication persisted between source and destination.
- `Flow.Bytes.s` : This is the flow rate in bytes per second, representing the data transfer rate of the flow. This could help in understanding the rate of data transmission for a specific flow.

2. Problems

2.1 Total no. of flows.

An unique flow is determined by an unique 6-tuple of the following attributes:

- `Source.IP`
- `Destination.IP`
- `Source.Port`
- `Destination.Port`
- `Protocol`
- `Timestamp`

```
In [ ]: columns_of_interest = ['Source.IP', 'Destination.IP', 'Source.Port', 'Destination.Port', 'Protocol', 'Timestamp'
        unique_flows = df[columns_of_interest].drop_duplicates()

        # extract the no. of unique flows
        num_unique_flows = unique_flows.shape[0]
        print("Total number of unique flows: ", num_unique_flows)
```

```
Total number of unique flows:  3141011
```

2.2 Total flow duration.

Total flow duration is just the sum of all the entries in the `Flow.Duration` column.

```
In [ ]: # total flow duration
        total_flow_duration = df['Flow.Duration'].sum()
        print("Total flow duration (in s): ", total_flow_duration)
```

```
Total flow duration (in s):  91015231179554
```

2.3 Total no. of bytes transferred.

Total bytes transferred per flow is the product of entries in `Flow.Bytes.s` and `Flow.Duration` respectively.

```
In [ ]: # add column named 'Flow.Transfer.Bytes' to the dataframe to store the total number of bytes transferred in each

        df['Flow.Transfer.Bytes'] = df['Flow.Bytes.s'] * df['Flow.Duration']

        # total number of bytes transferred
        total_bytes_transferred = df['Flow.Transfer.Bytes'].sum()
        print("Total number of bytes transferred: ", total_bytes_transferred)
```

```
Total number of bytes transferred:  4.696655193740449e+17
```

2.4 Big Flows.

We identify large flows in 3 ways:

- Big flows in terms of duration.
- Big flows in terms of bytes transferred.
- Big flows in terms of packets transferred.

2.4.1 Big flows in terms of duration.

```
In [ ]: # top 10 flows in terms of duration
        top_10_flows_duration = df.sort_values(by='Flow.Duration', ascending=False).head(10)
        top_10_flows_duration
```

Out[ ]:

	Flow.ID	Source.IP	Source.Port	Destination.IP	Destination.Port	Protocol	Timestamp	Flow.Durati
3566617	10.200.7.196-52.202.201.151-37047-443-6	52.202.201.151	443	10.200.7.196	37047	6	15/05/201705:28:00	1200000
2760107	192.168.220.5-10.200.7.5-1956-3128-6	10.200.7.5	3128	192.168.220.5	1956	6	11/05/201711:11:06	1200000
2340852	192.168.60.56-10.200.7.6-59217-3128-6	10.200.7.6	3128	192.168.60.56	59217	6	11/05/201709:40:01	1200000
2564368	179.1.4.237-10.200.7.195-443-46591-6	10.200.7.195	46591	179.1.4.237	443	6	11/05/201710:39:39	1200000
3248512	172.217.29.66-10.200.7.218-443-56678-6	10.200.7.218	56678	172.217.29.66	443	6	15/05/201711:15:22	1200000
567829	192.168.29.6-10.200.7.7-62740-3128-6	192.168.29.6	62740	10.200.7.7	3128	6	27/04/201708:34:19	1200000
3048052	192.168.41.3-10.200.7.4-60406-3128-6	192.168.41.3	60406	10.200.7.4	3128	6	11/05/201703:41:31	1200000
2406736	192.173.28.37-10.200.7.194-80-51948-6	10.200.7.194	51948	192.173.28.37	80	6	11/05/201710:06:02	1200000
981639	10.200.7.217-31.216.145.107-42426-80-6	31.216.145.107	80	10.200.7.217	42426	6	27/04/201710:38:51	1200000
2931821	192.168.29.5-10.200.7.5-54332-3128-6	10.200.7.5	3128	192.168.29.5	54332	6	11/05/201703:27:50	1200000

10 rows × 88 columns

2.4.2 Big flows in terms of bytes transferred.

In [ ]:

```
# top 10 flows in terms of bytes transferred
top_10_flows_bytes = df.sort_values(by='Flow.Transfer.Bytes', ascending=False).head(10)
top_10_flows_bytes
```

Out[ ]:

	Flow.ID	Source.IP	Source.Port	Destination.IP	Destination.Port	Protocol	Timestamp	Flow.Duratio
40680	192.168.180.51-10.200.7.4-57855-3128-6	192.168.180.51	57855	10.200.7.4	3128	6	26/04/201711:11:46	11926735
504485	185.181.102.34-10.200.7.218-443-50731-6	10.200.7.218	50731	185.181.102.34	443	6	27/04/201708:26:40	11993370
367385	185.181.102.39-10.200.7.218-443-53313-6	10.200.7.218	53313	185.181.102.39	443	6	27/04/201707:56:25	11997686
688385	192.168.150.16-10.200.7.4-49908-3128-6	192.168.150.16	49908	10.200.7.4	3128	6	27/04/201709:10:55	9595130
489249	185.181.102.39-10.200.7.217-443-45962-6	10.200.7.217	45962	185.181.102.39	443	6	27/04/201708:21:36	11999994
368971	185.181.102.39-10.200.7.218-443-58819-6	10.200.7.218	58819	185.181.102.39	443	6	27/04/201707:57:28	11380553
1766193	216.58.222.97-10.200.7.217-443-37798-6	10.200.7.217	37798	216.58.222.97	443	6	28/04/201710:09:35	10136153
541440	185.181.102.40-10.200.7.218-443-59509-6	185.181.102.40	443	10.200.7.218	59509	6	27/04/201708:21:36	11411403
1890687	192.168.90.91-10.200.7.7-56726-3128-6	192.168.90.91	56726	10.200.7.7	3128	6	28/04/201710:09:35	1014297
580761	192.168.72.31-10.200.7.8-56879-3128-6	192.168.72.31	56879	10.200.7.8	3128	6	27/04/201708:26:40	11999931

10 rows × 88 columns

2.4.3 Big flows in terms of packets transferred.

In [ ]:

```
# add column named 'Total.Flow.Packets' to the dataframe to store the total number of packets transferred in eac
df['Total.Flow.Packets'] = df['Flow.Packets.s'] * df['Flow.Duration']

# top 10 flows in terms of packets transferred
top_10_flows_packets = df.sort_values(by='Total.Flow.Packets', ascending=False).head(10)
top_10_flows_packets
```

Out[ ]:

	Flow.ID	Source.IP	Source.Port	Destination.IP	Destination.Port	Protocol	Timestamp	Flow.Durati
40680	192.168.180.51-10.200.7.4-57855-3128-6	192.168.180.51	57855	10.200.7.4	3128	6	26/04/201711:11:46	1192673
1321186	192.168.142.22-10.200.7.9-50359-3128-6	10.200.7.9	3128	192.168.142.22	50359	6	27/04/201704:55:11	1197357
1766193	216.58.222.97-10.200.7.217-443-37798-6	10.200.7.217	37798	216.58.222.97	443	6	28/04/201710:09:35	1013615
504485	185.181.102.34-10.200.7.218-443-50731-6	10.200.7.218	50731	185.181.102.34	443	6	27/04/201708:26:40	1199337
367385	185.181.102.39-10.200.7.218-443-53313-6	10.200.7.218	53313	185.181.102.39	443	6	27/04/201707:56:25	1199768
3319386	192.168.90.29-10.200.7.9-50081-3128-6	192.168.90.29	50081	10.200.7.9	3128	6	15/05/201711:11:01	1199623
2387032	192.168.90.86-10.200.7.5-50478-3128-6	192.168.90.86	50478	10.200.7.5	3128	6	11/05/201709:40:28	1199941
489249	185.181.102.39-10.200.7.217-443-45962-6	10.200.7.217	45962	185.181.102.39	443	6	27/04/201708:21:36	1199999
541440	185.181.102.40-10.200.7.218-443-59509-6	185.181.102.40	443	10.200.7.218	59509	6	27/04/201708:21:36	1141140
1890687	192.168.90.91-10.200.7.7-56726-3128-6	192.168.90.91	56726	10.200.7.7	3128	6	28/04/201710:09:35	1014297

10 rows × 89 columns