In [39]:
```
import pandas as pd
import numpy as np
```

In [2]:
```
pip install ucimlrepo
```

```
Collecting ucimlrepo
  Downloading ucimlrepo-0.0.6-py3-none-any.whl (8.0 kB)
Installing collected packages: ucimlrepo
Successfully installed ucimlrepo-0.0.6
Note: you may need to restart the kernel to use updated packages.
```

In [8]:
```
from ucimlrepo import fetch_ucirepo
import pandas as pd
import numpy as np
import matplotlib as plt

# fetch dataset
rt_iot2022 = fetch_ucirepo(id=942)

# data (as pandas dataframes)
X = rt_iot2022.data.features
y = rt_iot2022.data.targets

# metadata
print(rt_iot2022.metadata)

# variable information
print(rt_iot2022.variables)
```

```
{'uci_id': 942, 'name': 'RT-IoT2022 ', 'repository_url': 'https://archive.ics.uci.ed
u/dataset/942/rt-iot2022', 'data_url': 'https://archive.ics.uci.edu/static/public/94
2/data.csv', 'abstract': 'The RT-IoT2022, a proprietary dataset derived from a real-t
ime IoT infrastructure, is introduced as a comprehensive resource integrating a diver
se range of IoT devices and sophisticated network attack methodologies. This dataset
encompasses both normal and adversarial network behaviours, providing a general repre
sentation of real-world scenarios.\nIncorporating data from IoT devices such as Thing
Speak-LED, Wipro-Bulb, and MQTT-Temp, as well as simulated attack scenarios involving
Brute-Force SSH attacks, DDoS attacks using Hping and Slowloris, and Nmap patterns, R
T-IoT2022 offers a detailed perspective on the complex nature of network traffic. The
bidirectional attributes of network traffic are meticulously captured using the Zeek
network monitoring tool and the Flowmeter plugin. Researchers can leverage the RT-IoT
2022 dataset to advance the capabilities of Intrusion Detection Systems (IDS), foster
ing the development of robust and adaptive security solutions for real-time IoT netwo
rks. ', 'area': 'Engineering', 'tasks': ['Classification', 'Regression', 'Clusterin
g'], 'characteristics': ['Tabular', 'Sequential', 'Multivariate'], 'num_instances': 1
23117, 'num_features': 83, 'feature_types': ['Real', 'Categorical'], 'demographics':
[], 'target_col': ['Attack_type'], 'index_col': ['id'], 'has_missing_values': 'no',
'missing_values_symbol': None, 'year_of_dataset_creation': 2023, 'last_updated': 'Fri
Mar 08 2024', 'dataset_doi': '10.24432/C5P338', 'creators': ['B. S.', 'Rohini Nagapad
ma'], 'intro_paper': {'title': 'Quantized autoencoder (QAE) intrusion detection syste
m for anomaly detection in resource-constrained IoT devices using RT-IoT2022 datase
t', 'authors': 'B. S. Sharmila, Rohini Nagapadma', 'published_in': 'Cybersecurity',
'year': 2023, 'url': 'https://www.semanticscholar.org/paper/753f6ede01b4acaa325e302c3
8f1e0c1ade74f5b', 'doi': None}, 'additional_info': {'summary': None, 'purpose': None,
'funded_by': None, 'instances_represent': None, 'recommended_data_splits': None, 'sen
sitive_data': None, 'preprocessing_description': None, 'variable_info': 'Column Detai
ls:\nid.orig_p\nid.resp_p\nproto\nservice\nflow_duration\nfwd_pkts_tot\nbwd_pkts_tot
\nfwd_data_pkts_tot\nbwd_data_pkts_tot\nfwd_pkts_per_sec\nbwd_pkts_per_sec\nflow_pkts
_per_sec\ndown_up_ratio\nfwd_header_size_tot\nfwd_header_size_min\nfwd_header_size_ma
x\nbwd_header_size_tot\nbwd_header_size_min\nbwd_header_size_max\nflow_FIN_flag_count
\nflow_SYN_flag_count\nflow_RST_flag_count\nfwd_PSH_flag_count\nbwd_PSH_flag_count\nf
low_ACK_flag_count\nfwd_URG_flag_count\nbwd_URG_flag_count\nflow_CWR_flag_count\nflow
_ECE_flag_count\nfwd_pkts_payload.min\nfwd_pkts_payload.max\nfwd_pkts_payload.tot\nfw
d_pkts_payload.avg\nfwd_pkts_payload.std\nbwd_pkts_payload.min\nbwd_pkts_payload.max
\nbwd_pkts_payload.tot\nbwd_pkts_payload.avg\nbwd_pkts_payload.std\nflow_pkts_payloa
d.min\nflow_pkts_payload.max\nflow_pkts_payload.tot\nflow_pkts_payload.avg\nflow_pkts
_payload.std\nfwd_iat.min\nfwd_iat.max\nfwd_iat.tot\nfwd_iat.avg\nfwd_iat.std\nbwd_ia
t.min\nbwd_iat.max\nbwd_iat.tot\nbwd_iat.avg\nbwd_iat.std\nflow_iat.min\nflow_iat.max
\nflow_iat.tot\nflow_iat.avg\nflow_iat.std\npayload_bytes_per_second\nfwd_subflow_pkt
s\nbwd_subflow_pkts\nfwd_subflow_bytes\nbwd_subflow_bytes\nfwd_bulk_bytes\nbwd_bulk_b
ytes\nfwd_bulk_packets\nbwd_bulk_packets\nfwd_bulk_rate\nbwd_bulk_rate\nactive.min\na
ctive.max\nactive.tot\nactive.avg\nactive.std\nidle.min\nidle.max\nidle.tot\nidle.avg
\nidle.std\nfwd_init_window_size\nbwd_init_window_size\nfwd_last_window_size\nAttack_
type', 'citation': None}}
                    name     role         type demographic description units  \
0              id.orig_p  Feature      Integer        None        None  None
1              id.resp_p  Feature      Integer        None        None  None
2                  proto  Feature  Categorical        None        None  None
3                service  Feature   Continuous        None        None  None
4          flow_duration  Feature   Continuous        None        None  None
..                   ...      ...          ...         ...         ...   ...
80  fwd_init_window_size  Feature      Integer        None        None  None
81  bwd_init_window_size  Feature      Integer        None        None  None
82  fwd_last_window_size  Feature      Integer        None        None  None
83           Attack_type   Target  Categorical        None        None  None
84                    id       ID      Integer        None        None  None

   missing_values
0              no
```

```
1               no
2               no
3               no
4               no
..             ...
80              no
81              no
82              no
83              no
84              no

[85 rows x 7 columns]
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Cell In[8], line 19
     16 # variable information
     17 print(rt_iot2022.variables)
---> 19 pd.concat([x,y])

NameError: name 'x' is not defined
```

In [10]:
```
pd.concat([X,y]) #merge the two dataframes (X,y) into one dataframe
```

Out[10]:

	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot	bwd_pkts_tot	fwd_data_pkts
0	38667.0	1883.0	tcp	mqtt	32.011598	9.0	5.0	
1	51143.0	1883.0	tcp	mqtt	31.883584	9.0	5.0	
2	44761.0	1883.0	tcp	mqtt	32.124053	9.0	5.0	
3	60893.0	1883.0	tcp	mqtt	31.961063	9.0	5.0	
4	51087.0	1883.0	tcp	mqtt	31.902362	9.0	5.0	
...	...	...	...	...	...	...	...	...
123112	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
123113	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
123114	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
123115	NaN	NaN	NaN	NaN	NaN	NaN	NaN	
123116	NaN	NaN	NaN	NaN	NaN	NaN	NaN	

246234 rows × 84 columns

In [30]:
```
df = pd.concat([X,y], axis=1) #concatinated the NaN values column wise (axis=1)
df
```

Out[30]:

	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot	bwd_pkts_tot	fwd_data_pkts
0	38667	1883	tcp	mqtt	32.011598	9	5	
1	51143	1883	tcp	mqtt	31.883584	9	5	
2	44761	1883	tcp	mqtt	32.124053	9	5	
3	60893	1883	tcp	mqtt	31.961063	9	5	
4	51087	1883	tcp	mqtt	31.902362	9	5	
...	...	...	...	...	...	...	...	...
123112	59247	63331	tcp	-	0.000006	1	1	
123113	59247	64623	tcp	-	0.000007	1	1	
123114	59247	64680	tcp	-	0.000006	1	1	
123115	59247	65000	tcp	-	0.000006	1	1	
123116	59247	65129	tcp	-	0.000006	1	1	

123117 rows × 84 columns

In [15]:
```
df.dtypes #we want to change the datatypes of proto, service, and Attack_type to numer
```

Out[15]:
```
id.orig_p                int64
id.resp_p                int64
proto                   object
service                 object
flow_duration          float64
                         ...
idle.std               float64
fwd_init_window_size     int64
bwd_init_window_size     int64
fwd_last_window_size     int64
Attack_type             object
Length: 84, dtype: object
```

In [35]:
```
df['proto'].unique() #we can see that these are the objects we want to replace as nume
```

Out[35]:
```
array(['tcp', 'udp', 'icmp'], dtype=object)
```

In [36]:
```
df['service'].unique() #we can see that these are the objects we want to replace as nu
```

Out[36]:
```
array(['mqtt', '-', 'http', 'dns', 'ntp', 'ssl', 'dhcp', 'irc', 'ssh',
       'radius'], dtype=object)
```

In [37]:
```
df['Attack_type'].unique() #we can see that these are the objects we want to replace a
```

Out[37]:
```
array(['MQTT_Publish', 'Thing_Speak', 'Wipro_bulb', 'ARP_poisioning',
       'DDOS_Slowloris', 'DOS_SYN_Hping', 'Metasploit_Brute_Force_SSH',
       'NMAP_FIN_SCAN', 'NMAP_OS_DETECTION', 'NMAP_TCP_scan',
       'NMAP_UDP_SCAN', 'NMAP_XMAS_TREE_SCAN'], dtype=object)
```

In [ ]:
```

```

In [16]:
```
df.describe()
```

Out[16]:

	id.orig_p	id.resp_p	flow_duration	fwd_pkts_tot	bwd_pkts_tot	fwd_data_pkts_tot
count	123117.000000	123117.000000	123117.000000	123117.000000	123117.000000	123117.000000
mean	34639.258738	1014.305092	3.809566	2.268826	1.909509	1.471218
std	19070.620354	5256.371994	130.005408	22.336565	33.018311	19.635196
min	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	17702.000000	21.000000	0.000001	1.000000	1.000000	1.000000
50%	37221.000000	21.000000	0.000004	1.000000	1.000000	1.000000
75%	50971.000000	21.000000	0.000005	1.000000	1.000000	1.000000
max	65535.000000	65389.000000	21728.335580	4345.000000	10112.000000	4345.000000

8 rows × 81 columns

In [17]:
```
df.head()
```

Out[17]:

	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot	bwd_pkts_tot	fwd_data_pkts_tot
0	38667	1883	tcp	mqtt	32.011598	9	5	3
1	51143	1883	tcp	mqtt	31.883584	9	5	3
2	44761	1883	tcp	mqtt	32.124053	9	5	3
3	60893	1883	tcp	mqtt	31.961063	9	5	3
4	51087	1883	tcp	mqtt	31.902362	9	5	3

5 rows × 84 columns

In [18]:
```
df.tail()
```

Out[18]:

	id.orig_p	id.resp_p	proto	service	flow_duration	fwd_pkts_tot	bwd_pkts_tot	fwd_data_pkts
123112	59247	63331	tcp	-	0.000006	1	1	
123113	59247	64623	tcp	-	0.000007	1	1	
123114	59247	64680	tcp	-	0.000006	1	1	
123115	59247	65000	tcp	-	0.000006	1	1	
123116	59247	65129	tcp	-	0.000006	1	1	

5 rows × 84 columns

Conclusion

In this quiz, the dataset came with missing values (NaN), what I did was concatenate the two dataframes (X, y) into a single dataframe column wise. Furthermore, after concatenating the two

seperate dataframes, we now check the dtypes of every coloumn. We were tasked to change the dtypes of a column that is labeled as object. To do that it's either we create a function on our own or use the lambda function. However, I wasn't able to convert the said columns into a numeric datatype. I should practice more of my OOP so that next quiz I would complete all of my tasks