In [1]:
```
import pandas as pd
import numpy as np
train = pd.read_csv('UNSW_NB15_training-set.csv')
test = pd.read_csv('UNSW_NB15_testing-set.csv')
train.shape, test.shape
```

Out[1]: ((175341, 45), (82332, 45))

In [2]:
```
train.head(175341)
```

Out[2]:

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_
0	1	0.121478	tcp	-	FIN	6	4	258	172	74.087490	...	1	1	
1	2	0.649902	tcp	-	FIN	14	38	734	42014	78.473372	...	1	2	
2	3	1.623129	tcp	-	FIN	8	16	364	13186	14.170161	...	1	3	
3	4	1.681642	tcp	ftp	FIN	12	12	628	770	13.677108	...	1	3	
4	5	0.449454	tcp	-	FIN	10	6	534	268	33.373826	...	1	40	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
175336	175337	0.000009	udp	dns	INT	2	0	114	0	111111.107200	...	13	24	
175337	175338	0.505762	tcp	-	FIN	10	8	620	354	33.612649	...	1	2	
175338	175339	0.000009	udp	dns	INT	2	0	114	0	111111.107200	...	3	13	
175339	175340	0.000009	udp	dns	INT	2	0	114	0	111111.107200	...	14	30	
175340	175341	0.000009	udp	dns	INT	2	0	114	0	111111.107200	...	16	30	

175341 rows × 45 columns

In [3]:
```
test.head(82332)
```

Out[3]:

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_sport_ltm	ct_dst_src_ltm	is_ftp_l
0	1	0.000011	udp	-	INT	2	0	496	0	90909.090200	...	1	2	
1	2	0.000008	udp	-	INT	2	0	1762	0	125000.000300	...	1	2	
2	3	0.000005	udp	-	INT	2	0	1068	0	200000.005100	...	1	3	
3	4	0.000006	udp	-	INT	2	0	900	0	166666.660800	...	1	3	
4	5	0.000010	udp	-	INT	2	0	2126	0	100000.002500	...	1	3	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
82327	82328	0.000005	udp	-	INT	2	0	104	0	200000.005100	...	1	2	
82328	82329	1.106101	tcp	-	FIN	20	8	18062	354	24.410067	...	1	1	
82329	82330	0.000000	arp	-	INT	1	0	46	0	0.000000	...	1	1	
82330	82331	0.000000	arp	-	INT	1	0	46	0	0.000000	...	1	1	
82331	82332	0.000009	udp	-	INT	2	0	104	0	111111.107200	...	1	1	

82332 rows × 45 columns

In [4]:
```
train.dtypes
```

Out[4]:
```
id                   int64
dur                float64
proto               object
service             object
state               object
spkts                int64
dpkts                int64
sbytes               int64
dbytes               int64
rate               float64
sttl                 int64
dttl                 int64
sload              float64
dload              float64
sloss                int64
dloss                int64
sinpkt             float64
dinpkt             float64
sjit               float64
djit               float64
swin                 int64
stcpb                int64
dtcpb                int64
dwin                 int64
tcprtt             float64
synack             float64
ackdat             float64
smean                int64
dmean                int64
trans_depth          int64
response_body_len    int64
ct_srv_src           int64
ct_state_ttl         int64
ct_dst_ltm           int64
ct_src_dport_ltm     int64
ct_dst_sport_ltm     int64
ct_dst_src_ltm       int64
is_ftp_login         int64
ct_ftp_cmd           int64
ct_flw_http_mthd     int64
ct_src_ltm           int64
ct_srv_dst           int64
is_sm_ips_ports      int64
attack_cat          object
label                int64
dtype: object
```

In [5]:
```
test.dtypes
```

Out[5]:
```
id                   int64
dur                float64
proto               object
service             object
state               object
spkts                int64
dpkts                int64
sbytes               int64
dbytes               int64
rate               float64
sttl                 int64
dttl                 int64
sload              float64
dload              float64
sloss                int64
dloss                int64
sinpkt             float64
dinpkt             float64
sjit               float64
djit               float64
swin                 int64
stcpb                int64
dtcpb                int64
dwin                 int64
tcprtt             float64
synack             float64
ackdat             float64
smean                int64
dmean                int64
trans_depth          int64
response_body_len    int64
ct_srv_src           int64
ct_state_ttl         int64
ct_dst_ltm           int64
ct_src_dport_ltm     int64
ct_dst_sport_ltm     int64
ct_dst_src_ltm       int64
is_ftp_login         int64
ct_ftp_cmd           int64
ct_flw_http_mthd     int64
ct_src_ltm           int64
ct_srv_dst           int64
is_sm_ips_ports      int64
attack_cat          object
label                int64
dtype: object
```

In [6]:
```
#Combine into file - 'concat_data':
train['source']= 'train'
test['source'] = 'test'
concat_data=pd.concat([train, test],ignore_index=True)
concat_data.shape
```

Out[6]:
```
(257673, 46)
```

In [7]:
```
concat_data.head(257673)
```

Out[7]:

	id	dur	proto	service	state	spkts	dpkts	sbytes	dbytes	rate	...	ct_dst_src_ltm	is_ftp_login	ct_ftp_cmd
0	1	0.121478	tcp	-	FIN	6	4	258	172	74.087490	...	1	0	0
1	2	0.649902	tcp	-	FIN	14	38	734	42014	78.473372	...	2	0	0
2	3	1.623129	tcp	-	FIN	8	16	364	13186	14.170161	...	3	0	0
3	4	1.681642	tcp	ftp	FIN	12	12	628	770	13.677108	...	3	1	1
4	5	0.449454	tcp	-	FIN	10	6	534	268	33.373826	...	40	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
257668	82328	0.000005	udp	-	INT	2	0	104	0	200000.005100	...	2	0	0
257669	82329	1.106101	tcp	-	FIN	20	8	18062	354	24.410067	...	1	0	0
257670	82330	0.000000	arp	-	INT	1	0	46	0	0.000000	...	1	0	0
257671	82331	0.000000	arp	-	INT	1	0	46	0	0.000000	...	1	0	0
257672	82332	0.000009	udp	-	INT	2	0	104	0	111111.107200	...	1	0	0

257673 rows × 46 columns

In [8]:
```
# Find the number of missing data
concat_data.apply(lambda x: sum(x.isnull()))
```

Out[8]:
```
id                  0
dur                 0
proto               0
service             0
state               0
spkts               0
dpkts               0
sbytes              0
dbytes              0
rate                0
sttl                0
dttl                0
sload               0
dload               0
sloss               0
dloss               0
sinpkt              0
dinpkt              0
sjit                0
djit                0
swin                0
stcpb               0
dtcpb               0
dwin                0
tcprtt              0
synack              0
ackdat              0
smean               0
dmean               0
trans_depth         0
response_body_len   0
ct_srv_src          0
ct_state_ttl        0
ct_dst_ltm          0
ct_src_dport_ltm    0
ct_dst_sport_ltm    0
ct_dst_src_ltm      0
is_ftp_login        0
ct_ftp_cmd          0
ct_flw_http_mthd    0
ct_src_ltm          0
ct_srv_dst          0
is_sm_ips_ports     0
attack_cat          0
label               0
source              0
dtype: int64
```

In [9]:
```
var = ['proto','service','state']
for v in var:
    print ('\nFrequency count for variable %s'%v)
    print (concat_data[v].value_counts())
```

```
Frequency count for variable proto
tcp     123041
udp      92701
unas     15599
arp       3846
ospf      3271
         ...
rdp        131
crtp       131
igmp        48
icmp        15
rtp          1
Name: proto, Length: 133, dtype: int64

Frequency count for variable service
-           141321
dns          68661
http         27011
smtp          6909
ftp-data      5391
ftp           4980
pop3          1528
ssh           1506
dhcp           120
snmp           109
ssl             86
irc             30
radius          21
Name: service, dtype: int64

Frequency count for variable state
FIN    117164
INT    116438
CON     20134
REQ      3833
RST        84
ECO        12
ACC         4
PAR         1
URN         1
no          1
CLO         1
Name: state, dtype: int64
```

In [10]:
```
var_to_encode = ['proto','service','state']
concat_data = pd.get_dummies(concat_data, columns=var_to_encode)
concat_data.columns
```

Out[10]:
```
Index(['id', 'dur', 'spkts', 'dpkts', 'sbytes', 'dbytes', 'rate', 'sttl',
       'dttl', 'sload',
       ...
       'state_CLO', 'state_CON', 'state_ECO', 'state_FIN', 'state_INT',
       'state_PAR', 'state_REQ', 'state_RST', 'state_URN', 'state_no'],
      dtype='object', length=200)
```

In [11]:
```
concat_data.head(257673)
```

Out[11]:

	id	dur	spkts	dpkts	sbytes	dbytes	rate	sttl	dttl	sload	...	state_CLO	state_CON	state_ECO	s
0	1	0.121478	6	4	258	172	74.087490	252	254	1.415894e+04	...	0	0	0	
1	2	0.649902	14	38	734	42014	78.473372	62	252	8.395112e+03	...	0	0	0	
2	3	1.623129	8	16	364	13186	14.170161	62	252	1.572272e+03	...	0	0	0	
3	4	1.681642	12	12	628	770	13.677108	62	252	2.740179e+03	...	0	0	0	
4	5	0.449454	10	6	534	268	33.373826	254	252	8.561499e+03	...	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
257668	82328	0.000005	2	0	104	0	200000.005100	254	0	8.320000e+07	...	0	0	0	
257669	82329	1.106101	20	8	18062	354	24.410067	254	252	1.241044e+05	...	0	0	0	
257670	82330	0.000000	1	0	46	0	0.000000	0	0	0.000000e+00	...	0	0	0	
257671	82331	0.000000	1	0	46	0	0.000000	0	0	0.000000e+00	...	0	0	0	
257672	82332	0.000009	2	0	104	0	111111.107200	254	0	4.622222e+07	...	0	0	0	

257673 rows × 200 columns

In [12]:
```
train = concat_data.loc[concat_data['source']=='train']
test = concat_data.loc[concat_data['source']=='test']
```

In [13]:
```
train.drop('source',axis=1,inplace=True)
test.drop('source',axis=1,inplace=True)
```

```
C:\Users\admin\anaconda3\lib\site-packages\pandas\core\frame.py:4308: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy (https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy)
  return super().drop(
```

In [14]:
```
train.to_csv('UNSW_NB15_ohe_training_set.csv',index=False)
test.to_csv('UNSW_NB15_ohe_testing_set.csv',index=False)
```

In [ ]:
```

```