```
import pandas as pd
```

```
df=pd.read_csv("/content/data.txt")
df.shape
```

```
(670, 43)
```

```
df.head()
```

	0	tcp	private	REJ	0.1	0.2	0.3	0.4	0.5	0.6	...	0.04.1	0.06.1	0.00.3	0.00.4	0.00.5	0.00.6	1.00.2	1.00.3	nep
0	0	tcp	private	REJ	0	0	0	0	0	0	...	0.00	0.06	0.00	0.00	0.00	0.0	1.00	1.00	nep
1	2	tcp	ftp_data	SF	12983	0	0	0	0	0	...	0.61	0.04	0.61	0.02	0.00	0.0	0.00	0.00	nc
2	0	icmp	eco_i	SF	20	0	0	0	0	0	...	1.00	0.00	1.00	0.28	0.00	0.0	0.00	0.00	
3	1	tcp	telnet	RSTO	0	15	0	0	0	0	...	0.31	0.17	0.03	0.02	0.00	0.0	0.83	0.71	m
4	0	tcp	http	SF	267	14515	0	0	0	0	...	1.00	0.00	0.01	0.03	0.01	0.0	0.00	0.00	nc

5 rows × 43 columns

```
columns = (['duration','protocol_type','service','flag','src_bytes','dst_bytes','land','wrong_fragment','urgent','hot','num_failed_logins
df.columns = columns
```

```
df.head()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	dst
0	0	tcp	private	REJ	0	0	0	0	0	0	...	0.00	
1	2	tcp	ftp_data	SF	12983	0	0	0	0	0	...	0.61	
2	0	icmp	eco_i	SF	20	0	0	0	0	0	...	1.00	
3	1	tcp	telnet	RSTO	0	15	0	0	0	0	...	0.31	
4	0	tcp	http	SF	267	14515	0	0	0	0	...	1.00	

5 rows × 43 columns

```
df.tail()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate	d
665	0	tcp	private	RSTO	0	0	0	0	0	0	...	0.08	
666	0	tcp	private	S0	0	0	0	0	0	0	...	0.05	
667	0	tcp	http	SF	275	5223	0	0	0	0	...	1.00	
668	0	tcp	http	SF	323	541	0	0	0	0	...	1.00	
669	4	tcp	pop_3	SF	32	93	0	0	0	0	...	0.53	

5 rows × 43 columns

```
df.shape
```

```
(670, 43)
```

```
df.describe()
```

	duration	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	num_failed_logins	logged_in	num_comprc
count	670.000000	670.000000	670.000000	670.0	670.000000	670.0	670.000000	670.000000	670.000000	670.0
mean	337.798507	4582.792537	3790.956716	0.0	0.008955	0.0	0.098507	0.026866	0.431343	0.9
std	2494.048923	26811.366961	36073.041746	0.0	0.133630	0.0	0.554232	0.161812	0.495634	23.6
min	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.000000	0.0
25%	0.000000	0.000000	0.000000	0.0	0.000000	0.0	0.000000	0.000000	0.000000	0.0
50%	0.000000	52.000000	45.500000	0.0	0.000000	0.0	0.000000	0.000000	0.000000	0.0
75%	0.000000	275.750000	696.250000	0.0	0.000000	0.0	0.000000	0.000000	1.000000	0.0
max	54451.000000	283618.000000	834163.000000	0.0	3.000000	0.0	7.000000	1.000000	1.000000	611.0

8 rows × 39 columns

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 670 entries, 0 to 669
Data columns (total 43 columns):
 #   Column   Non-Null Count  Dtype  
---  ------   --------------  -----  
 0   0        670 non-null    int64  
 1   tcp      670 non-null    object 
 2   private  670 non-null    object 
 3   REJ      670 non-null    object 
 4   0.1      670 non-null    int64  
 5   0.2      670 non-null    int64  
 6   0.3      670 non-null    int64  
 7   0.4      670 non-null    int64  
 8   0.5      670 non-null    int64  
 9   0.6      670 non-null    int64  
 10  0.7      670 non-null    int64  
 11  0.8      670 non-null    int64  
 12  0.9      670 non-null    int64  
 13  0.10     670 non-null    int64  
 14  0.11     670 non-null    int64  
 15  0.12     670 non-null    int64  
 16  0.13     670 non-null    int64  
 17  0.14     670 non-null    int64  
 18  0.15     670 non-null    int64  
 19  0.16     670 non-null    int64  
 20  0.17     670 non-null    int64  
 21  0.18     670 non-null    int64  
 22  229      670 non-null    int64  
 23  10       670 non-null    int64  
 24  0.00     670 non-null    float64
 25  0.00.1   670 non-null    float64
 26  1.00     670 non-null    float64
 27  1.00.1   670 non-null    float64
 28  0.04     670 non-null    float64
 29  0.06     670 non-null    float64
 30  0.00.2   670 non-null    float64
 31  255      670 non-null    int64  
 32  10.1     670 non-null    int64  
 33  0.04.1   670 non-null    float64
 34  0.06.1   670 non-null    float64
 35  0.00.3   670 non-null    float64
 36  0.00.4   670 non-null    float64
 37  0.00.5   670 non-null    float64
 38  0.00.6   670 non-null    float64
 39  1.00.2   670 non-null    float64
 40  1.00.3   669 non-null    float64
 41  neptune  669 non-null    object 
 42  21       669 non-null    float64
dtypes: float64(16), int64(23), object(4)
memory usage: 225.2+ KB
```

```
df.isna().sum()
```

	0
0	0
tcp	0
private	0
REJ	0
0.1	0
0.2	0
0.3	0
0.4	0
0.5	0
0.6	0
0.7	0
0.8	0
0.9	0
0.10	0
0.11	0
0.12	0
0.13	0
0.14	0
0.15	0
0.16	0
0.17	0
0.18	0
229	0
10	0
0.00	0
0.00.1	0
1.00	0
1.00.1	0
0.04	0
0.06	0
0.00.2	0
255	0
10.1	0
0.04.1	0
0.06.1	0
0.00.3	0
0.00.4	0
0.00.5	0
0.00.6	0
1.00.2	0
1.00.3	1
neptune	1
21	1

dtype: int64

```
df.isnull()
```

	duration	protocol_type	service	flag	src_bytes	dst_bytes	land	wrong_fragment	urgent	hot	...	dst_host_same_srv_rate
0	False	False	False	False	False	False	False	False	False	False	...	False
1	False	False	False	False	False	False	False	False	False	False	...	False
2	False	False	False	False	False	False	False	False	False	False	...	False
3	False	False	False	False	False	False	False	False	False	False	...	False
4	False	False	False	False	False	False	False	False	False	False	...	False
...	...	...	...	...	...	...	...	...	...	...	...	...
665	False	False	False	False	False	False	False	False	False	False	...	False
666	False	False	False	False	False	False	False	False	False	False	...	False
667	False	False	False	False	False	False	False	False	False	False	...	False
668	False	False	False	False	False	False	False	False	False	False	...	False
669	False	False	False	False	False	False	False	False	False	False	...	False

670 rows × 43 columns

```
df.columns.isnull
```

pandas.core.indexes.base.Index.isna
def isna() -> npt.NDArray[np.bool_]

Detect missing values.

Return a boolean same-sized object indicating if the values are NA.
NA values, such as ``None``, :attr:`numpy.NaN` or :attr:`pd.NaT`, get
mapped to ``True`` values.
Everything else get mapped to ``False`` values. Characters such as

```
integer_columns_df = df.select_dtypes(include='int64')
print(integer_columns_df)
```

```
     0    0.1    0.2  0.3  0.4  0.5  0.6  0.7  0.8  0.9  ...  0.13  0.14  \
0    0      0      0    0    0    0    0    0    0    0  ...     0     0
1    2  12983      0    0    0    0    0    0    0    0  ...     0     0
2    0     20      0    0    0    0    0    0    0    0  ...     0     0
3    1      0     15    0    0    0    0    0    0    0  ...     0     0
4    0    267  14515    0    0    0    0    0    1    0  ...     0     0
..  ..    ...    ...  ...  ...  ...  ...  ...  ...  ...  ...   ...   ...
665  0      0      0    0    0    0    0    0    0    0  ...     0     0
666  0      0      0    0    0    0    0    0    0    0  ...     0     0
667  0    275   5223    0    0    0    0    0    1    0  ...     0     0
668  0    323    541    0    0    0    0    0    1    0  ...     0     0
669  4     32     93    0    0    0    0    0    1    0  ...     0     0

     0.15  0.16  0.17  0.18  229  10  255  10.1
0       0     0     0     0  136   1  255     1
1       0     0     0     0    1   1  134    86
2       0     0     0     0    1  65    3    57
3       0     0     0     0    1   8   29    86
4       0     0     0     0    4   4  155   255
..    ...   ...   ...   ...  ...  ..  ...   ...
665     0     0     0     0  260  20  255    20
666     0     0     0     0  290  13  255    13
667     0     0     0     0    1   1   22   255
668     0     0     0     0    1   4    9   255
669     0     0     0     0    2   1  255   136

[670 rows x 23 columns]
```

```
integer_columns_df.mean()
```

	0
0	337.798507
0.1	4582.792537
0.2	3790.956716
0.3	0.000000
0.4	0.008955
0.5	0.000000
0.6	0.098507
0.7	0.026866
0.8	0.431343
0.9	0.928358
0.10	0.002985
0.11	0.002985
0.12	1.022388
0.13	0.008955
0.14	0.001493
0.15	0.007463
0.17	0.000000

```
integer_columns_df.median()
```

	0
0	0.0
0.1	52.0
0.2	45.5
0.3	0.0
0.4	0.0
0.5	0.0
0.6	0.0
0.7	0.0
0.8	0.0
0.9	0.0
0.10	0.0
0.11	0.0
0.12	0.0
0.13	0.0
0.14	0.0
0.15	0.0
0.16	0.0
0.17	0.0
0.18	0.0
229	9.0
10	6.0
255	255.0
10.1	165.0

dtype: float64