

ICT607: Artificial Intelligence for Cybersecurity

March 27, 2023

Lab 5: Preprocessing dataset – intrusion detection

Data preprocessing is the process of preparing raw data for analysis by applying various techniques to make the data more useful and meaningful. It involves transforming data into a format that can be easily analyzed by machine learning algorithms.

In this laboratory, we will preprocess the KDD Cup 1999 dataset, which is a widely used dataset for evaluating intrusion detection systems. It contains a sample of network traffic data from the DARPA 1998 Intrusion Detection System Evaluation, which aimed to evaluate the ability of intrusion detection systems to detect various types of network attacks.

The dataset includes network connections, each represented by 41 features. The connections are classified into one of five categories: normal, DoS, probe, R2L (unauthorized access from a remote machine), and U2R (unauthorized access to local superuser privileges).

The 41 features include basic features such as duration, protocol type, and service, as well as more detailed features such as number of failed login attempts, number of root accesses, and number of file creations. Some features are continuous, while others are categorical or binary.

1 Load and organise the dataset

Download the dataset from [Kaggle](https://www.kaggle.com/datasets/galaxyh/kdd-cup-1999-data). Unzip and upload *kddcup.data_10_percent.gz* file into */content* folder of your colab session.

```
[ ]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

```
[ ]: # feature_names found from https://www.kaggle.com/datasets/galaxyh/
↳kdd-cup-1999-data?resource=download&select=kddcup.names
feature_names=['duration',
'protocol_type',
'service',
'flag',
'src_bytes',
'dst_bytes',
```

```

'land',
'wrong_fragment',
'urgent',
'hot',
'num_failed_logins',
'logged_in',
'num_compromised',
'root_shell',
'su_attempted',
'num_root',
'num_file_creations',
'num_shells',
'num_access_files',
'num_outbound_cmds',
'is_host_login',
'is_guest_login',
'count',
'srv_count',
'serror_rate',
'srv_error_rate',
'rerror_rate',
'srv_rerror_rate',
'same_srv_rate',
'diff_srv_rate',
'srv_diff_host_rate',
'dst_host_count',
'dst_host_srv_count',
'dst_host_same_srv_rate',
'dst_host_diff_srv_rate',
'dst_host_same_src_port_rate',
'dst_host_srv_diff_host_rate',
'dst_host_serror_rate',
'dst_host_srv_serror_rate',
'dst_host_rerror_rate',
'dst_host_srv_rerror_rate',
'target']

```

```
[ ]: len(feature_names)
```

```
[ ]: 42
```

```

[ ]: # attack_types found from https://www.kaggle.com/datasets/galaxyh/
    ↪kdd-cup-1999-data?resource=download&select=training_attack_types
    # dot (.) added with each type to match the dataset
attack_types = {
    'normal.': 'normal',
    'back.': 'dos',

```

```

'buffer_overflow.': 'u2r',
'ftp_write.': 'r2l',
'guess_passwd.': 'r2l',
'imap.': 'r2l',
'ipsweep.': 'probe',
'land.': 'dos',
'loadmodule.': 'u2r',
'multihop.': 'r2l',
'neptune.': 'dos',
'nmap.': 'probe',
'perl.': 'u2r',
'phf.': 'r2l',
'pod.': 'dos',
'portsweep.': 'probe',
'rootkit.': 'u2r',
'satan.': 'probe',
'smurf.': 'dos',
'spy.': 'r2l',
'teardrop.': 'dos',
'warezclient.': 'r2l',
'warezmaster.': 'r2l',
}

```

```

[ ]: # Load the dataset
df = pd.read_csv("kddcup.data_10_percent.gz", names=feature_names)

```

```

[ ]: df

```

```

[ ]:
      duration  protocol_type  service  flag  src_bytes  dst_bytes  land  \
0           0            tcp    http    SF         181        5450     0
1           0            tcp    http    SF         239         486     0
2           0            tcp    http    SF         235        1337     0
3           0            tcp    http    SF         219        1337     0
4           0            tcp    http    SF         217        2032     0
...         ...            ...      ...    ...         ...         ...
494016        0            tcp    http    SF         310        1881     0
494017        0            tcp    http    SF         282        2286     0
494018        0            tcp    http    SF         203        1200     0
494019        0            tcp    http    SF         291        1200     0
494020        0            tcp    http    SF         219        1234     0

      wrong_fragment  urgent  hot  ...  dst_host_srv_count  \
0           0         0    0  ...              9
1           0         0    0  ...             19
2           0         0    0  ...             29
3           0         0    0  ...             39
4           0         0    0  ...             49

```

...
494016	0	0	0	...	255
494017	0	0	0	...	255
494018	0	0	0	...	255
494019	0	0	0	...	255
494020	0	0	0	...	255

	dst_host_same_srv_rate	dst_host_diff_srv_rate	\
0	1.0	0.0	
1	1.0	0.0	
2	1.0	0.0	
3	1.0	0.0	
4	1.0	0.0	
...	
494016	1.0	0.0	
494017	1.0	0.0	
494018	1.0	0.0	
494019	1.0	0.0	
494020	1.0	0.0	

	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	\
0	0.11	0.00	
1	0.05	0.00	
2	0.03	0.00	
3	0.03	0.00	
4	0.02	0.00	
...	
494016	0.01	0.05	
494017	0.17	0.05	
494018	0.06	0.05	
494019	0.04	0.05	
494020	0.17	0.05	

	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	\
0	0.00	0.00	0.0	
1	0.00	0.00	0.0	
2	0.00	0.00	0.0	
3	0.00	0.00	0.0	
4	0.00	0.00	0.0	
...	
494016	0.00	0.01	0.0	
494017	0.00	0.01	0.0	
494018	0.06	0.01	0.0	
494019	0.04	0.01	0.0	
494020	0.00	0.01	0.0	

dst_host_srv_rerror_rate	target
--------------------------	--------

```

0          0.0  normal.
1          0.0  normal.
2          0.0  normal.
3          0.0  normal.
4          0.0  normal.
...
494016          0.0  normal.
494017          0.0  normal.
494018          0.0  normal.
494019          0.0  normal.
494020          0.0  normal.

```

[494021 rows x 42 columns]

```
[ ]: df['target'].value_counts()
```

```

[ ]: smurf.          280790
      neptune.       107201
      normal.        97278
      back.          2203
      satan.         1589
      ipsweep.       1247
      portsweep.     1040
      warezclient.   1020
      teardrop.      979
      pod.           264
      nmap.          231
      guess_passwd.  53
      buffer_overflow. 30
      land.          21
      warezmaster.   20
      imap.          12
      rootkit.        10
      loadmodule.     9
      ftp_write.      8
      multihop.       7
      phf.            4
      perl.           3
      spy.            2
      Name: target, dtype: int64

```

```
[ ]: attack_types['back.']
```

```
[ ]: 'dos'
```

```

[ ]: # adding attack_type column
      df['attack_type'] = df.target.apply(lambda r:attack_types[r])

```

```
[ ]: df
```

```
[ ]:
      duration protocol_type service flag  src_bytes  dst_bytes  land  \
0           0           tcp    http   SF        181       5450     0
1           0           tcp    http   SF        239        486     0
2           0           tcp    http   SF        235       1337     0
3           0           tcp    http   SF        219       1337     0
4           0           tcp    http   SF        217       2032     0
...         ...         ...    ...   ...         ...         ...
494016      0           tcp    http   SF        310       1881     0
494017      0           tcp    http   SF        282       2286     0
494018      0           tcp    http   SF        203       1200     0
494019      0           tcp    http   SF        291       1200     0
494020      0           tcp    http   SF        219       1234     0
```

```

      wrong_fragment  urgent  hot  ...  dst_host_same_srv_rate  \
0                  0        0    0  ...                      1.0
1                  0        0    0  ...                      1.0
2                  0        0    0  ...                      1.0
3                  0        0    0  ...                      1.0
4                  0        0    0  ...                      1.0
...               ...      ...  ...  ...                      ...
494016             0        0    0  ...                      1.0
494017             0        0    0  ...                      1.0
494018             0        0    0  ...                      1.0
494019             0        0    0  ...                      1.0
494020             0        0    0  ...                      1.0
```

```

      dst_host_diff_srv_rate  dst_host_same_src_port_rate  \
0                          0.0                          0.11
1                          0.0                          0.05
2                          0.0                          0.03
3                          0.0                          0.03
4                          0.0                          0.02
...                         ...                          ...
494016                     0.0                          0.01
494017                     0.0                          0.17
494018                     0.0                          0.06
494019                     0.0                          0.04
494020                     0.0                          0.17
```

```

      dst_host_srv_diff_host_rate  dst_host_serror_rate  \
0                               0.00                     0.00
1                               0.00                     0.00
2                               0.00                     0.00
3                               0.00                     0.00
4                               0.00                     0.00
```

```

...
494016          0.05          0.00
494017          0.05          0.00
494018          0.05          0.06
494019          0.05          0.04
494020          0.05          0.00

```

```

          dst_host_srv_serror_rate  dst_host_rerror_rate  \
0                0.00                0.0
1                0.00                0.0
2                0.00                0.0
3                0.00                0.0
4                0.00                0.0

```

```

...
494016          0.01          0.0
494017          0.01          0.0
494018          0.01          0.0
494019          0.01          0.0
494020          0.01          0.0

```

```

          dst_host_srv_rerror_rate  target  attack_type
0                0.0  normal.      normal
1                0.0  normal.      normal
2                0.0  normal.      normal
3                0.0  normal.      normal
4                0.0  normal.      normal

```

```

...
494016          0.0  normal.      normal
494017          0.0  normal.      normal
494018          0.0  normal.      normal
494019          0.0  normal.      normal
494020          0.0  normal.      normal

```

[494021 rows x 43 columns]

```
[ ]: df.drop(['target'],axis=1,inplace=True)
```

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

```
[ ]:
duration  protocol_type  service  flag  src_bytes  dst_bytes  land  \
0         0            tcp    http    SF        181        5450    0
1         0            tcp    http    SF        239         486    0
2         0            tcp    http    SF        235        1337    0
3         0            tcp    http    SF        219        1337    0
4         0            tcp    http    SF        217        2032    0

```

```

wrong_fragment  urgent  hot  ...  dst_host_srv_count  \

```

0	0	0	0	...	9
1	0	0	0	...	19
2	0	0	0	...	29
3	0	0	0	...	39
4	0	0	0	...	49

	dst_host_same_srv_rate	dst_host_diff_srv_rate	\
0	1.0	0.0	
1	1.0	0.0	
2	1.0	0.0	
3	1.0	0.0	
4	1.0	0.0	

	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	\
0	0.11	0.0	
1	0.05	0.0	
2	0.03	0.0	
3	0.03	0.0	
4	0.02	0.0	

	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	\
0	0.0	0.0	0.0	
1	0.0	0.0	0.0	
2	0.0	0.0	0.0	
3	0.0	0.0	0.0	
4	0.0	0.0	0.0	

	dst_host_srv_rerror_rate	attack_type
0	0.0	normal
1	0.0	normal
2	0.0	normal
3	0.0	normal
4	0.0	normal

[5 rows x 42 columns]

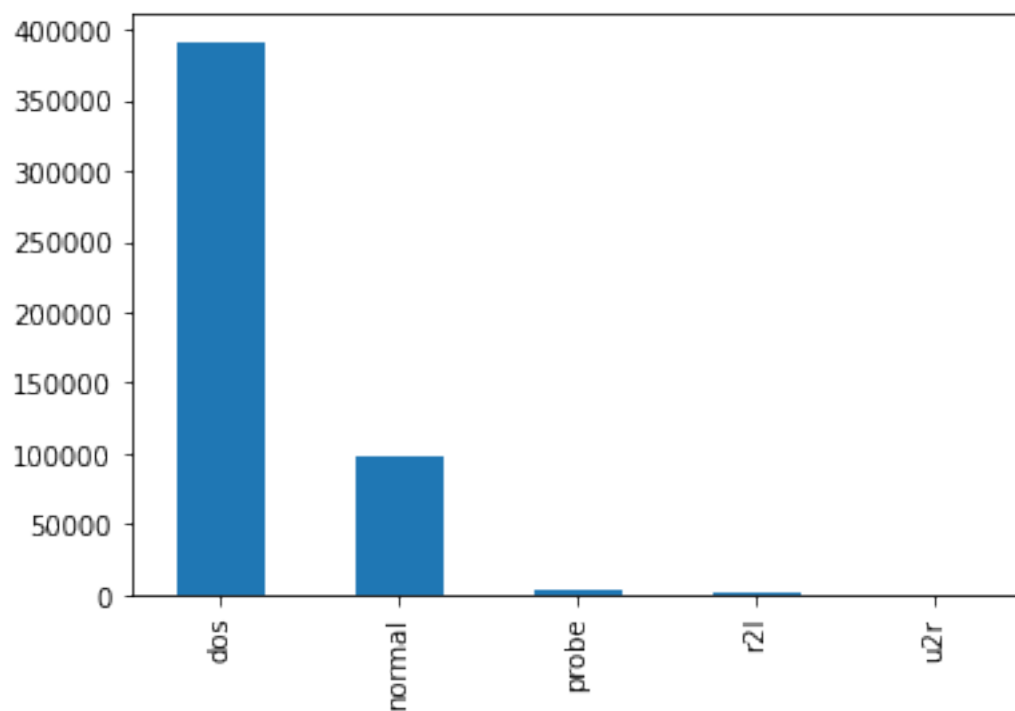
```
[ ]: df['attack_type'].value_counts()
```

```
[ ]: dos      391458
normal    97278
probe      4107
r2l        1126
u2r         52
Name: attack_type, dtype: int64
```

```
[ ]: # Visualisation
df['attack_type'].value_counts().plot(kind="bar")
```

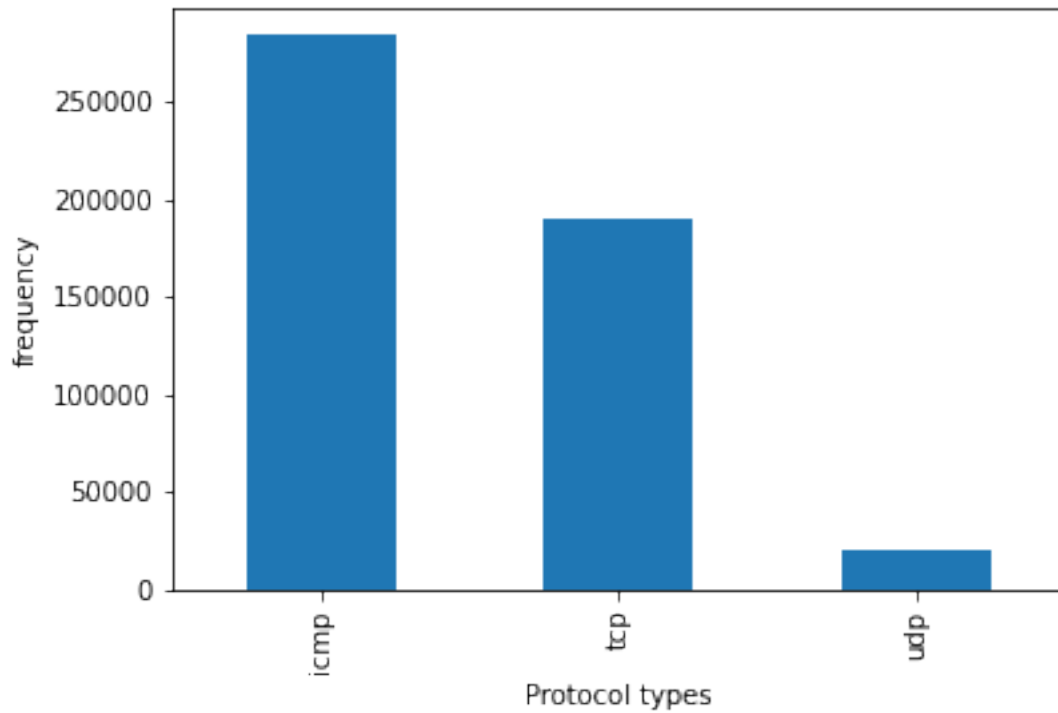


```
[ ]: <Axes: >
```



```
[ ]: df['protocol_type'].value_counts().plot(kind="bar",xlabel="Protocol_
↪types",ylabel="frequency")
```

```
[ ]: <Axes: xlabel='Protocol types', ylabel='frequency'>
```



2 Removing NA values and constant features

```
[ ]: df.isna().any() # checking NA values
```

```
[ ]: duration           False
      protocol_type     False
      service           False
      flag               False
      src_bytes          False
      dst_bytes          False
      land               False
      wrong_fragment     False
      urgent             False
      hot                False
      num_failed_logins   False
      logged_in          False
      num_compromised     False
      root_shell         False
      su_attempted       False
      num_root           False
      num_file_creations  False
      num_shells         False
```

```

num_access_files      False
num_outbound_cmds     False
is_host_login         False
is_guest_login        False
count                 False
srv_count              False
serror_rate           False
srv_serror_rate       False
rerror_rate           False
srv_rerror_rate       False
same_srv_rate         False
diff_srv_rate         False
srv_diff_host_rate    False
dst_host_count        False
dst_host_srv_count    False
dst_host_same_srv_rate False
dst_host_diff_srv_rate False
dst_host_same_src_port_rate False
dst_host_srv_diff_host_rate False
dst_host_serror_rate  False
dst_host_srv_serror_rate False
dst_host_rerror_rate  False
dst_host_srv_rerror_rate False
attack_type           False
dtype: bool

```

```
[ ]: # df.dropna('columns',inplace=True)
```

```
[ ]: df = df[[col for col in df if df[col].nunique(>1)]] # keep columns where there
↳are more than 1 unique values
```

```
[ ]: df
```

```
[ ]:
      duration protocol_type service flag  src_bytes  dst_bytes  land  \
0           0           tcp    http   SF         181        5450    0
1           0           tcp    http   SF         239         486    0
2           0           tcp    http   SF         235        1337    0
3           0           tcp    http   SF         219        1337    0
4           0           tcp    http   SF         217        2032    0
...         ...           ...     ...   ...         ...         ...
494016       0           tcp    http   SF         310        1881    0
494017       0           tcp    http   SF         282        2286    0
494018       0           tcp    http   SF         203        1200    0
494019       0           tcp    http   SF         291        1200    0
494020       0           tcp    http   SF         219        1234    0

```

```
wrong_fragment  urgent  hot  ...  dst_host_srv_count  \
```

0	0	0	0	...	9
1	0	0	0	...	19
2	0	0	0	...	29
3	0	0	0	...	39
4	0	0	0	...	49
...
494016	0	0	0	...	255
494017	0	0	0	...	255
494018	0	0	0	...	255
494019	0	0	0	...	255
494020	0	0	0	...	255

	dst_host_same_srv_rate	dst_host_diff_srv_rate	\
0	1.0	0.0	
1	1.0	0.0	
2	1.0	0.0	
3	1.0	0.0	
4	1.0	0.0	
...	
494016	1.0	0.0	
494017	1.0	0.0	
494018	1.0	0.0	
494019	1.0	0.0	
494020	1.0	0.0	

	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	\
0	0.11	0.00	
1	0.05	0.00	
2	0.03	0.00	
3	0.03	0.00	
4	0.02	0.00	
...	
494016	0.01	0.05	
494017	0.17	0.05	
494018	0.06	0.05	
494019	0.04	0.05	
494020	0.17	0.05	

	dst_host_serror_rate	dst_host_srv_serror_rate	dst_host_rerror_rate	\
0	0.00	0.00	0.0	
1	0.00	0.00	0.0	
2	0.00	0.00	0.0	
3	0.00	0.00	0.0	
4	0.00	0.00	0.0	
...	
494016	0.00	0.01	0.0	
494017	0.00	0.01	0.0	

494018	0.06	0.01	0.0
494019	0.04	0.01	0.0
494020	0.00	0.01	0.0

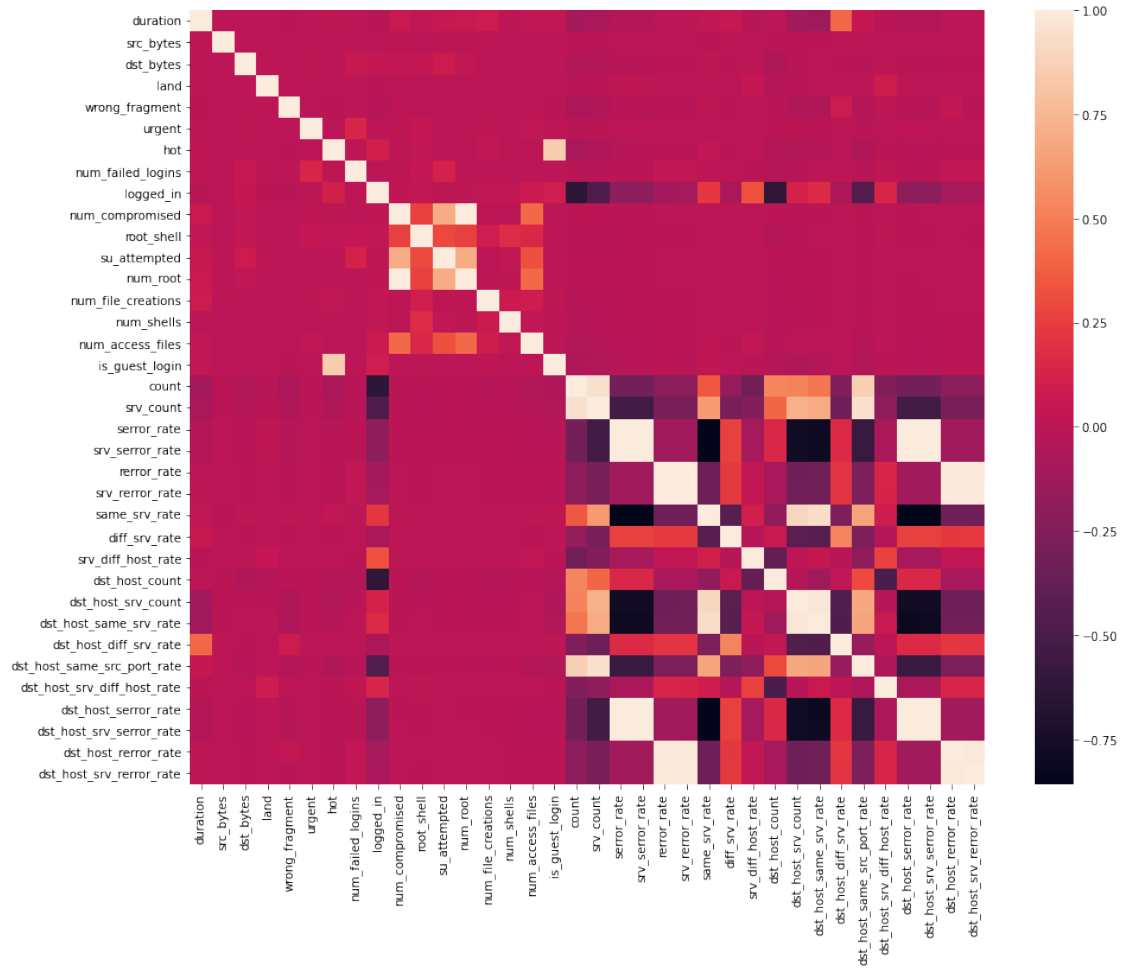
	dst_host_srv_rerror_rate	attack_type
0	0.0	normal
1	0.0	normal
2	0.0	normal
3	0.0	normal
4	0.0	normal
...
494016	0.0	normal
494017	0.0	normal
494018	0.0	normal
494019	0.0	normal
494020	0.0	normal

[494021 rows x 40 columns]

3 Removing highly-correlated features

```
[ ]: corr = df.corr() # Pearson correlation coefficient between the columns
plt.figure(figsize=(15,12)) # creates a new figure with a specified size of 15
    ↪ inches by 12 inches.
sns.heatmap(corr) # creates a heatmap of the correlation matrix using seaborn
```

```
[ ]: <Axes: >
```



```
[ ]: # num_root is highly correlated with num_compromised and should be ignored for
      ↪ analysis.
df.drop('num_root',axis = 1,inplace = True) # axis=1 specifies that we want to
      ↪ drop the columns axis

# srv_serror_rate is highly correlated with error_rate and should be ignored
      ↪ for analysis.
df.drop('srv_serror_rate',axis = 1,inplace = True)

# srv_error_rate is highly correlated with error_rate and should be ignored
      ↪ for analysis.
df.drop('srv_error_rate',axis = 1, inplace=True)

# dst_host_srv_error_rate is highly correlated with srv_error_rate and should
      ↪ be ignored for analysis.
df.drop('dst_host_srv_error_rate',axis = 1, inplace=True)
```

```

# dst_host_error_rate is highly correlated with error_rate and should be
↳ ignored for analysis.
df.drop('dst_host_error_rate',axis = 1, inplace=True)

# dst_host_error_rate is highly correlated with srv_error_rate and should be
↳ ignored for analysis.
df.drop('dst_host_error_rate',axis = 1, inplace=True)

# dst_host_srv_error_rate is highly correlated with error_rate and should be
↳ ignored for analysis.
df.drop('dst_host_srv_error_rate',axis = 1, inplace=True)

# dst_host_same_srv_rate is highly correlated with dst_host_srv_count and
↳ should be ignored for analysis.
df.drop('dst_host_same_srv_rate',axis = 1, inplace=True)

# srv_count is highly correlated with count and should be ignored for analysis.
df.drop('srv_count',axis = 1, inplace=True)

```

```

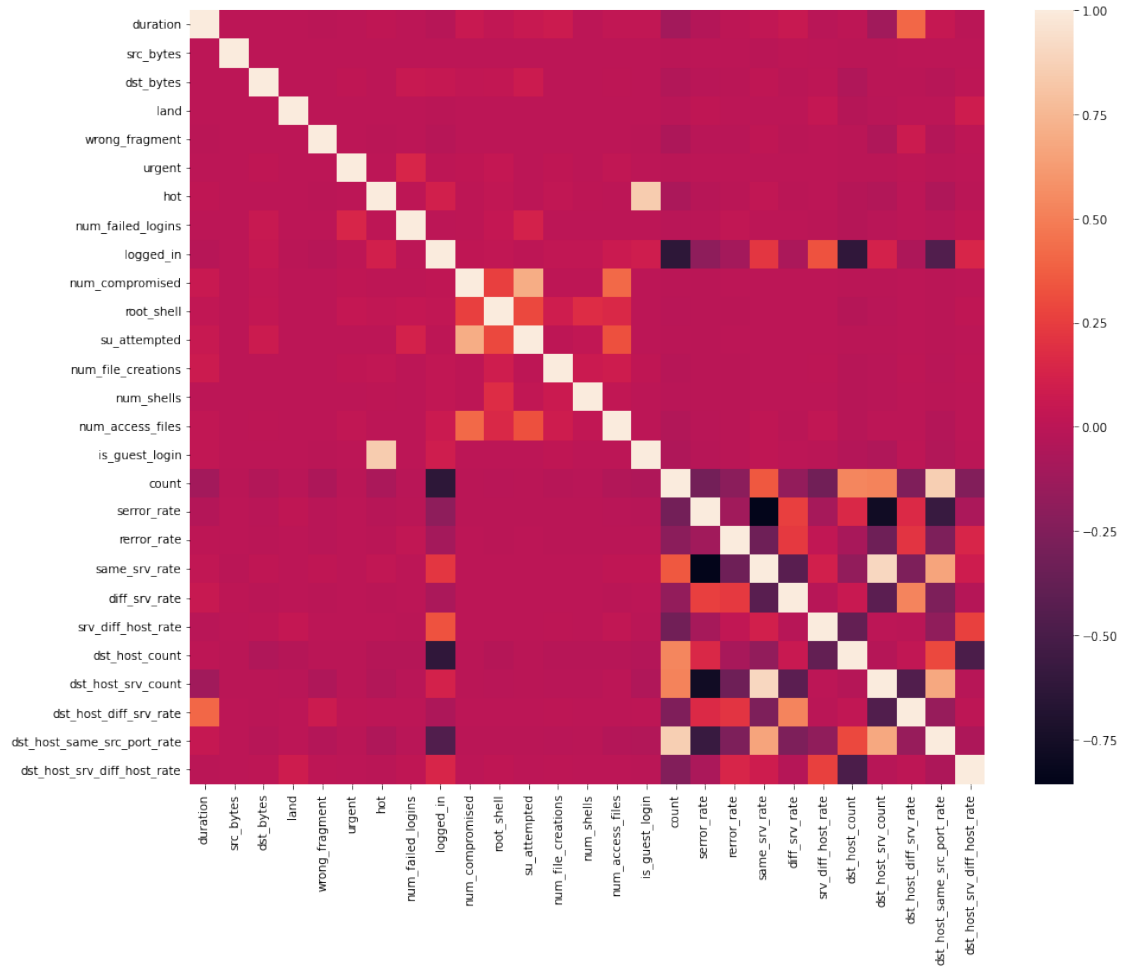
[ ]: corr = df.corr() # Pearson correlation coefficient between the columns
plt.figure(figsize=(15,12)) # creates a new figure with a specified size of 15
↳ inches by 12 inches.
sns.heatmap(corr) # creates a heatmap of the correlation matrix using seaborn

```

```

[ ]: <Axes: >

```



4 Label encoding the features

```
[ ]: df
```

```
[ ]:
duration protocol_type service flag src_bytes dst_bytes land \
0          0          tcp    http   SF      181      5450    0
1          0          tcp    http   SF      239       486    0
2          0          tcp    http   SF      235     1337    0
3          0          tcp    http   SF      219     1337    0
4          0          tcp    http   SF      217     2032    0
...      ...      ...      ...      ...      ...      ...
494016      0          tcp    http   SF      310     1881    0
494017      0          tcp    http   SF      282     2286    0
494018      0          tcp    http   SF      203     1200    0
494019      0          tcp    http   SF      291     1200    0
494020      0          tcp    http   SF      219     1234    0
```


	wrong_fragment	urgent	hot	...	rerror_rate	same_srv_rate	\
0	0	0	0	...	0.0	1.0	
1	0	0	0	...	0.0	1.0	
2	0	0	0	...	0.0	1.0	
3	0	0	0	...	0.0	1.0	
4	0	0	0	...	0.0	1.0	
...	
494016	0	0	0	...	0.0	1.0	
494017	0	0	0	...	0.0	1.0	
494018	0	0	0	...	0.0	1.0	
494019	0	0	0	...	0.0	1.0	
494020	0	0	0	...	0.0	1.0	

	diff_srv_rate	srv_diff_host_rate	dst_host_count	dst_host_srv_count	\
0	0.0		9	9	
1	0.0		19	19	
2	0.0		29	29	
3	0.0		39	39	
4	0.0		49	49	
...	
494016	0.0	0.40	86	255	
494017	0.0	0.00	6	255	
494018	0.0	0.17	16	255	
494019	0.0	0.17	26	255	
494020	0.0	0.14	6	255	

	dst_host_diff_srv_rate	dst_host_same_src_port_rate	\
0	0.0	0.11	
1	0.0	0.05	
2	0.0	0.03	
3	0.0	0.03	
4	0.0	0.02	
...	
494016	0.0	0.01	
494017	0.0	0.17	
494018	0.0	0.06	
494019	0.0	0.04	
494020	0.0	0.17	

	dst_host_srv_diff_host_rate	attack_type
0	0.00	normal
1	0.00	normal
2	0.00	normal
3	0.00	normal
4	0.00	normal
...

494016	0.05	normal
494017	0.05	normal
494018	0.05	normal
494019	0.05	normal
494020	0.05	normal

[494021 rows x 31 columns]

```
[ ]: df['protocol_type'].value_counts()
```

```
[ ]: icmp    283602
      tcp     190065
      udp     20354
      Name: protocol_type, dtype: int64
```

```
[ ]: pmap = {"icmp":0,"tcp":1,"udp":2}
      df['protocol_type'] = df['protocol_type'].map(pmap)
```

```
[ ]: df['protocol_type'].value_counts()
```

```
[ ]: 0    283602
      1    190065
      2     20354
      Name: protocol_type, dtype: int64
```

```
[ ]: #flag feature mapping
      fmap = {'SF':0,'SO':1,'REJ':2,'RSTR':3,'RSTO':4,'SH':5 , 'S1':6 , 'S2':7,'RSTOS0':
             ↪8,'S3':9 , 'OTH':10}
      df['flag'] = df['flag'].map(fmap)
```

```
[ ]: #attack type feature mapping
      amap = {'dos':0,'normal':1,'probe':2,'r2l':3,'u2r':4}
      df['attack_type'] = df['attack_type'].map(amap)
```

```
[ ]: df["service"].value_counts()
```

```
[ ]: ecr_i      281400
      private   110893
      http      64293
      smtp      9723
      other     7237
      ...
      X11        11
      tim_i       7
      pm_dump     1
      tftp_u      1
      red_i       1
```

Name: service, Length: 66, dtype: int64

```
[ ]: df.drop('service',axis = 1,inplace= True)
```

```
[ ]: df
```

```
[ ]:
```

	duration	protocol_type	flag	src_bytes	dst_bytes	land	\
0	0	1	0	181	5450	0	
1	0	1	0	239	486	0	
2	0	1	0	235	1337	0	
3	0	1	0	219	1337	0	
4	0	1	0	217	2032	0	
...	
494016	0	1	0	310	1881	0	
494017	0	1	0	282	2286	0	
494018	0	1	0	203	1200	0	
494019	0	1	0	291	1200	0	
494020	0	1	0	219	1234	0	

	wrong_fragment	urgent	hot	num_failed_logins	...	rerror_rate	\
0	0	0	0	0	...	0.0	
1	0	0	0	0	...	0.0	
2	0	0	0	0	...	0.0	
3	0	0	0	0	...	0.0	
4	0	0	0	0	...	0.0	
...	
494016	0	0	0	0	...	0.0	
494017	0	0	0	0	...	0.0	
494018	0	0	0	0	...	0.0	
494019	0	0	0	0	...	0.0	
494020	0	0	0	0	...	0.0	

	same_srv_rate	diff_srv_rate	srv_diff_host_rate	dst_host_count	\
0	1.0	0.0	0.00	9	
1	1.0	0.0	0.00	19	
2	1.0	0.0	0.00	29	
3	1.0	0.0	0.00	39	
4	1.0	0.0	0.00	49	
...	
494016	1.0	0.0	0.40	86	
494017	1.0	0.0	0.00	6	
494018	1.0	0.0	0.17	16	
494019	1.0	0.0	0.17	26	
494020	1.0	0.0	0.14	6	

	dst_host_srv_count	dst_host_diff_srv_rate	\
0	9	0.0	

1	19	0.0
2	29	0.0
3	39	0.0
4	49	0.0
...
494016	255	0.0
494017	255	0.0
494018	255	0.0
494019	255	0.0
494020	255	0.0

	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate	attack_type
0	0.11	0.00	1
1	0.05	0.00	1
2	0.03	0.00	1
3	0.03	0.00	1
4	0.02	0.00	1
...
494016	0.01	0.05	1
494017	0.17	0.05	1
494018	0.06	0.05	1
494019	0.04	0.05	1
494020	0.17	0.05	1

[494021 rows x 30 columns]

5 Scaling

```
[ ]: from sklearn.preprocessing import MinMaxScaler
```

```
[ ]: y = df[['attack_type']]
      X = df.drop(['attack_type'],axis=1)
```

```
[ ]: y
```

```
[ ]:      attack_type
0          1
1          1
2          1
3          1
4          1
...      ...
494016      1
494017      1
494018      1
494019      1
```

494020

1

[494021 rows x 1 columns]

[]: X

```
[ ]:
      duration  protocol_type  flag  src_bytes  dst_bytes  land  \
0           0           1      0      181      5450      0
1           0           1      0      239       486      0
2           0           1      0      235      1337      0
3           0           1      0      219      1337      0
4           0           1      0      217      2032      0
...
494016       0           1      0      310      1881      0
494017       0           1      0      282      2286      0
494018       0           1      0      203      1200      0
494019       0           1      0      291      1200      0
494020       0           1      0      219      1234      0

      wrong_fragment  urgent  hot  num_failed_logins  ...  serror_rate  \
0                  0       0   0                  0  ...      0.00
1                  0       0   0                  0  ...      0.00
2                  0       0   0                  0  ...      0.00
3                  0       0   0                  0  ...      0.00
4                  0       0   0                  0  ...      0.00
...
494016              0       0   0                  0  ...      0.00
494017              0       0   0                  0  ...      0.00
494018              0       0   0                  0  ...      0.17
494019              0       0   0                  0  ...      0.00
494020              0       0   0                  0  ...      0.00

      rerror_rate  same_srv_rate  diff_srv_rate  srv_diff_host_rate  \
0              0.0              1.0            0.0                0.00
1              0.0              1.0            0.0                0.00
2              0.0              1.0            0.0                0.00
3              0.0              1.0            0.0                0.00
4              0.0              1.0            0.0                0.00
...
494016          0.0              1.0            0.0                0.40
494017          0.0              1.0            0.0                0.00
494018          0.0              1.0            0.0                0.17
494019          0.0              1.0            0.0                0.17
494020          0.0              1.0            0.0                0.14

      dst_host_count  dst_host_srv_count  dst_host_diff_srv_rate  \
0                  9                  9                0.0
```

1	19	19	0.0
2	29	29	0.0
3	39	39	0.0
4	49	49	0.0
...
494016	86	255	0.0
494017	6	255	0.0
494018	16	255	0.0
494019	26	255	0.0
494020	6	255	0.0

	dst_host_same_src_port_rate	dst_host_srv_diff_host_rate
0	0.11	0.00
1	0.05	0.00
2	0.03	0.00
3	0.03	0.00
4	0.02	0.00
...
494016	0.01	0.05
494017	0.17	0.05
494018	0.06	0.05
494019	0.04	0.05
494020	0.17	0.05

[494021 rows x 29 columns]

```
[ ]: scaler = MinMaxScaler() # MinMaxScaler scales the data to a fixed range (by default, between 0 and 1) by subtracting the minimum value and dividing by the range of the data.
X = scaler.fit_transform(X)
```

```
[ ]: X
```

```
[ ]: array([[0. , 0.5 , 0. , ..., 0. , 0.11, 0. ],
           [0. , 0.5 , 0. , ..., 0. , 0.05, 0. ],
           [0. , 0.5 , 0. , ..., 0. , 0.03, 0. ],
           ...,
           [0. , 0.5 , 0. , ..., 0. , 0.06, 0.05],
           [0. , 0.5 , 0. , ..., 0. , 0.04, 0.05],
           [0. , 0.5 , 0. , ..., 0. , 0.17, 0.05]])
```

6 Train-test split

```
[ ]: from sklearn.model_selection import train_test_split
```

```
[ ]: # Split test and train data
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.33,
↳random_state=42)
print(X_train.shape, X_test.shape)
print(y_train.shape, y_test.shape)
```

```
(330994, 29) (163027, 29)
(330994, 1) (163027, 1)
```

```
[ ]: X_train.shape[1]
```

```
[ ]: 29
```

NB: Parts of this program is taken and improved from <https://www.kaggle.com/code/iamyajat/intrusion-detection-system-using-neural-networks>, which has been released under the Apache 2.0 open source license

7 Practice task

Visualise at least 10 features of the KDD Cup dataset using different types of plots (such as bar, histogram, line, etc.).