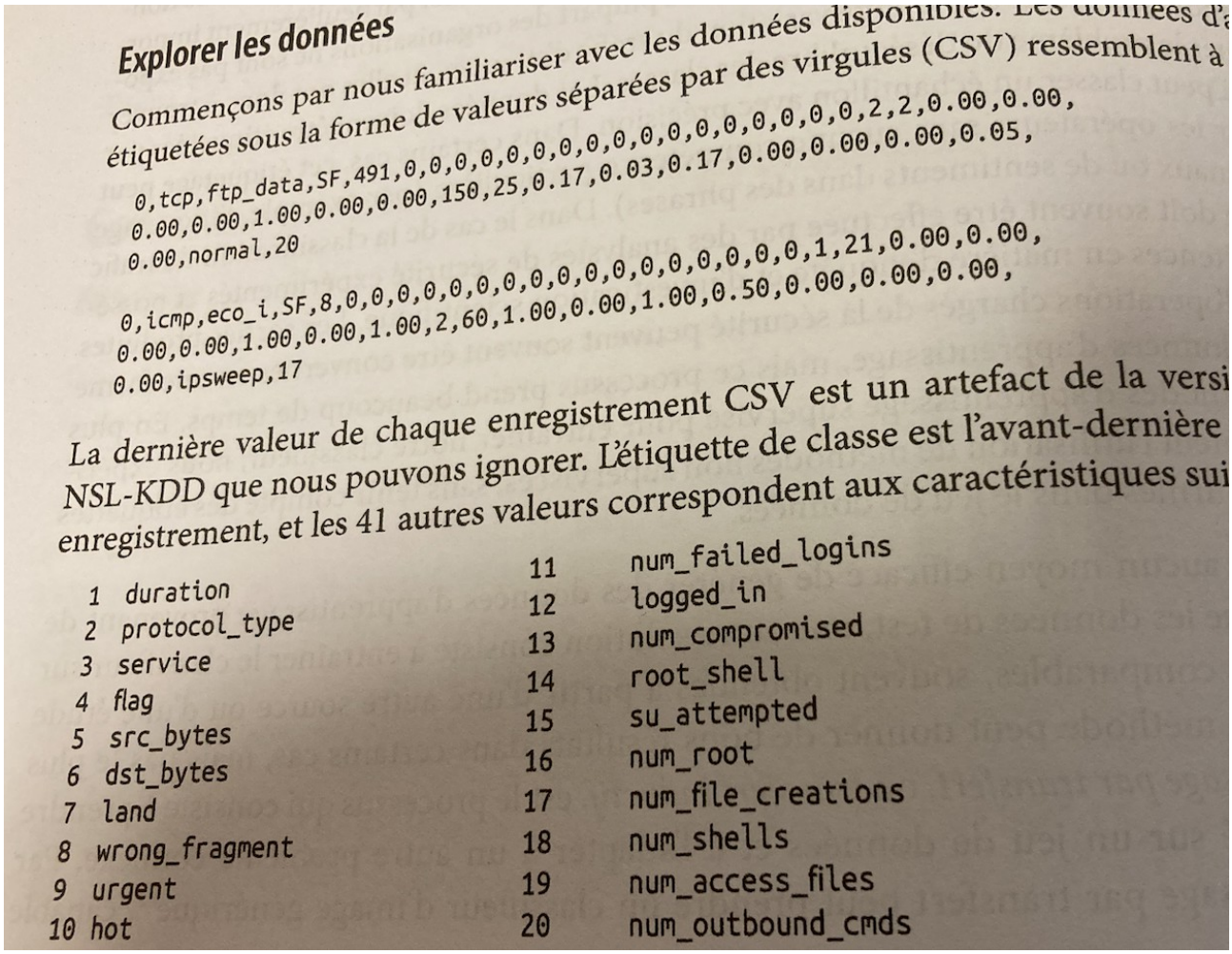
TP4 : Construire un modèle prédictif pour classifier les attaques réseau

Dans ce TP vous allez construire un classifieur d'attaques réseau en utilisant l'apprentissage automatique.

La tâche consiste à utiliser le dataset fourni constituant des attaques réseau. Le but est de concevoir un classifieur général qui catégorise chaque ligne du fichier dans l'une des cinq classes suivantes : *benign, dos, r2l, u2r et probe (Cinq classes).*

Le jeu de données contient des échantillons étiquetés par des attaques spécifiques. Chaque attaque appartient à une des cinq catégories (classes) définies ci dessus. La description du contenu des données est mentionné ci dessous :



Explorer les données

Commençons par nous familiariser avec les données disponibles. Les données d'
étiquetées sous la forme de valeurs séparées par des virgules (CSV) ressemblent à

```
0,tcp,ftp_data,SF,491,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,0.00,0.00,
0.00,0.00,1.00,0.00,0.00,150,25,0.17,0.03,0.17,0.00,0.00,0.00,0.05,
0.00,normal,20

0,icmp,eco_i,SF,8,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,21,0.00,0.00,
0.00,0.00,1.00,0.00,1.00,2,60,1.00,0.00,1.00,0.50,0.00,0.00,0.00,
0.00,ipsweep,17
```

La dernière valeur de chaque enregistrement CSV est un artefact de la versi
NSL-KDD que nous pouvons ignorer. L'étiquette de classe est l'avant-dernière
enregistrement, et les 41 autres valeurs correspondent aux caractéristiques sui

```
 1  duration
 2  protocol_type
 3  service
 4  flag
 5  src_bytes
 6  dst_bytes
 7  land
 8  wrong_fragment
 9  urgent
10  hot
11  num_failed_logins
12  logged_in
13  num_compromised
14  root_shell
15  su_attempted
16  num_root
17  num_file_creations
18  num_shells
19  num_access_files
20  num_outbound_cmds
```

```
21 is_host_login
22 is_guest_login
23 count
24 srv_count
25 serror_rate
26 srv_serror_rate
27 rerror_rate
28 srv_rerror_rate
29 same_srv_rate
30 diff_srv_rate
31 srv_diff_host_rate
32 dst_host_count
33 dst_host_srv_count
34 dst_host_same_srv_rate
35 dst_host_diff_srv_rate
36 dst_host_same_src_port_rate
37 dst_host_srv_diff_host_rate
38 dst_host_serror_rate
39 dst_host_srv_serror_rate
40 dst_host_rerror_rate
41 dst_host_srv_rerror_rate
```

Voici ce que nous trouvons en inspectant le contenu de categor

```
{
    'benign': ['normal'],
    'probe': ['nmap', 'ipsweep', 'portsweep', 'satan',
              'mscan', 'saint', 'worm'],
    'r2l': ['ftp_write', 'guess_passwd', 'snmpguess',
            'imap', 'spy', 'warezclient', 'warezmaster',
            'multihop', 'phf', 'imap', 'named', 'sendmail',
            'xlock', 'xsnoop', 'worm'],
    'u2r': ['ps', 'buffer_overflow', 'perl', 'rootkit',
            'loadmodule', 'xterm', 'sqlattack', 'httptunnel'],
    'dos': ['apache2', 'back', 'mailbomb', 'processtable',
            'snmpgetattack', 'teardrop', 'smurf', 'land',
            'neptune', 'pod', 'udpstorm']
}
```

Travail à faire

1 – Modèle d'entrainement par apprentissage supervisé

. Explorez les données et visulisez la proportion de données par classe d'attaque

. Construire un modèle (classifieur) sur la base du fichier de données fourni à l'aide d'un réseau de neurone puis calculez les scores du modèle obtenus sur le fichier de données de test fourni.

2 – Modèle d'entrainement par apprentissage non supervisé

. On se propose d'utliser la méthode K_Means afin de classer les données du fichier fournis en **Cinq classes.**
. Mesurez le score de clustering en comparant la classe déduite par K_Means avec la classe associé dans le fichier fourni (étiquette)