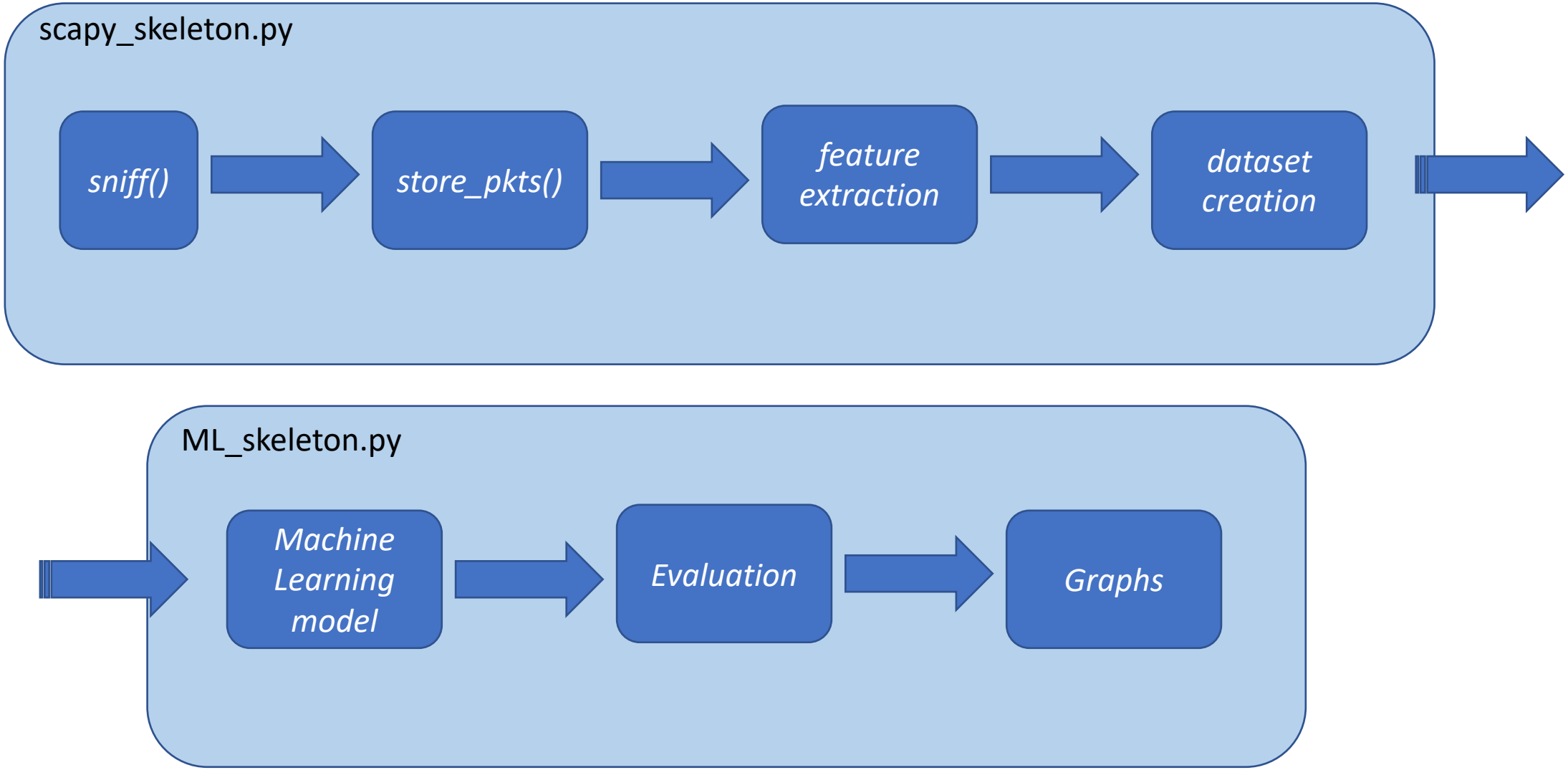
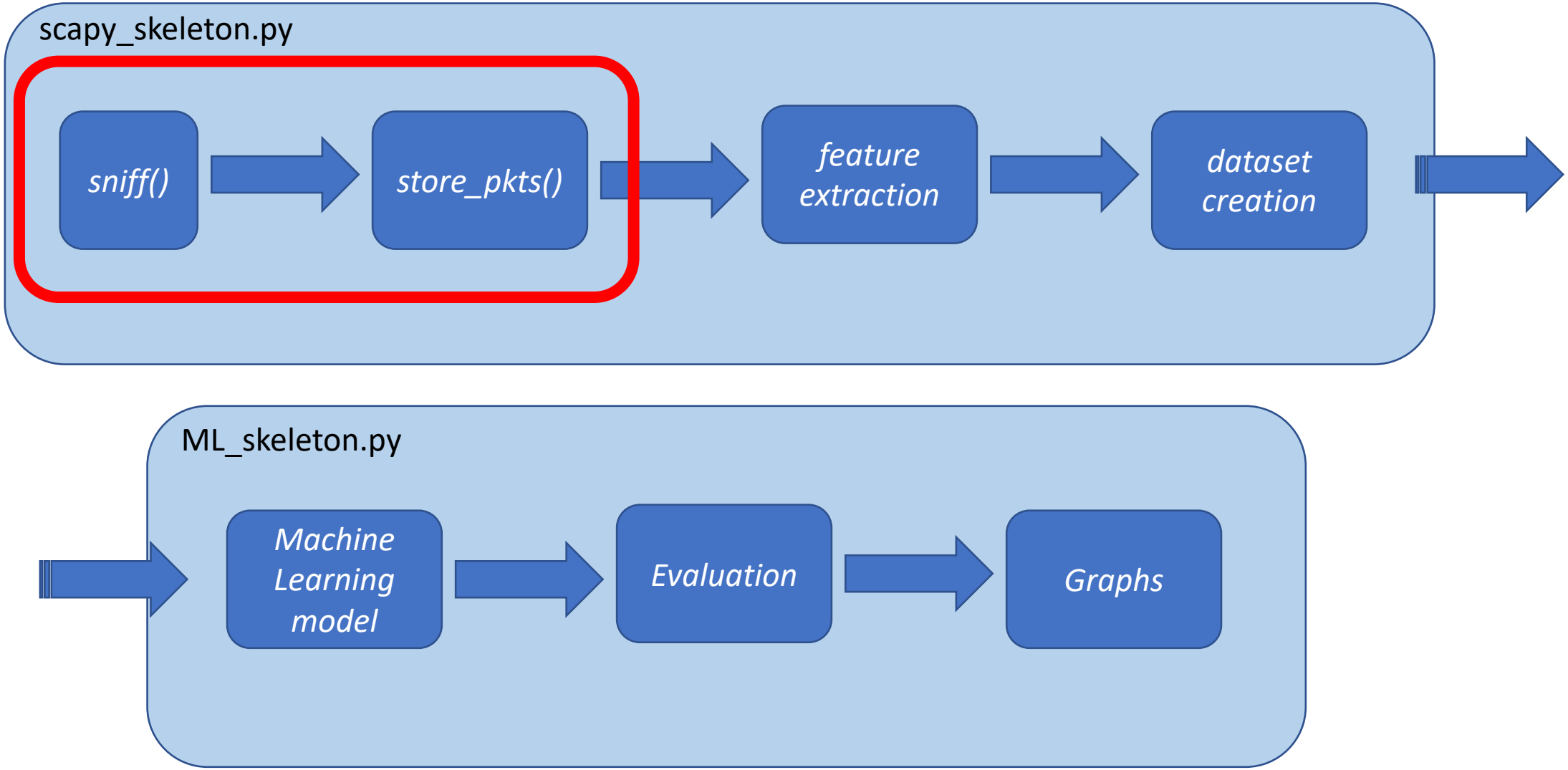
CS 371 – Class project

Network Traffic Analysis based on Machine Learning

Framework


scapy_skeleton.py
sniff()
store_pkts()
feature extraction
dataset creation
ML_skeleton.py
Machine Learning model
Evaluation
Graphs

Packet sniffing



What is sniffing?

- Interception of data, more specifically, network packets
- Can be used to analyze network traffic
- What kind of data can we extract?

CODE: scapy library

```
pkts = sniff(prn = fields_extraction, count = 10)
```

Callback

Amount of packets
to collect

CODE

```
pkts = sniff(prn = fields_extraction, count = 10)
```

Callback

Amount of packets to collect

Extract and store data!

```
def fields_extraction(x):
    print x.sprintf("{IP:%IP.src%,%IP.dst%,}"
        "{TCP:%TCP.sport%,%TCP.dport%,}"
        "{UDP:%UDP.sport%,%UDP.dport%}")
```

{PROT:%PROT.field_name1%,%PROT.field_name2%}

Let’s run some code!

sniff() function

```
###[ Ethernet ]###
  dst       = ff:ff:ff:ff:ff:ff
  src       = 78:31:c1:ef:14:2a
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 72
     id        = 42832
     flags     =
     frag      = 0
     ttl       = 64
     proto     = udp
     chksum    = 0x5cd4
     src       = 192.168.250.47
     dst       = 192.168.250.255
     \options   \
###[ UDP ]###
        sport     = 57621
        dport     = 57621
        len       = 52
        chksum    = 0x83a0
###[ Raw ]###
           load      = 'SpotUdp0\x83m(\xc1\xfc\xd9\
S0\xcc\xd5\x8bd\x8f\xdb\x9e\xdb\x0eTp7)\x13H\x8f['
```

```
###[ Ethernet ]###
  dst       = 88:e9:fe:68:e1:43
  src       = 80:2a:a8:9d:43:3d
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 166
     id        = 8191
     flags     = DF
     frag      = 0
     ttl       = 110
     proto     = tcp
     chksum    = 0xdffb
     src       = 52.173.28.179
     dst       = 192.168.250.78
     \options   \
###[ TCP ]###
        sport     = https
        dport     = 51858
        seq       = 3749973176
        ack       = 2889192337
        dataofs   = 5
        reserved  = 0
        flags     = PA
        window    = 6828
        chksum    = 0xe48e
        urgptr    = 0
        options   = []
###[ Raw ]###
           load      = '\x17\x03\x03\
xe2\x94\xd5\xcb> \xac(\x98\xd7E=\xcb\
xce\x8d\x08`\xadTO\x94\x96\xe0\xf1\xb
xeb\xa5\xe7\x17I0\x82+\x89\xe8\x0c]J\
b7\xa9\x87\xe1\x1b\xbd\xa4\x0c'
```

Transport layer: TCP & UDP

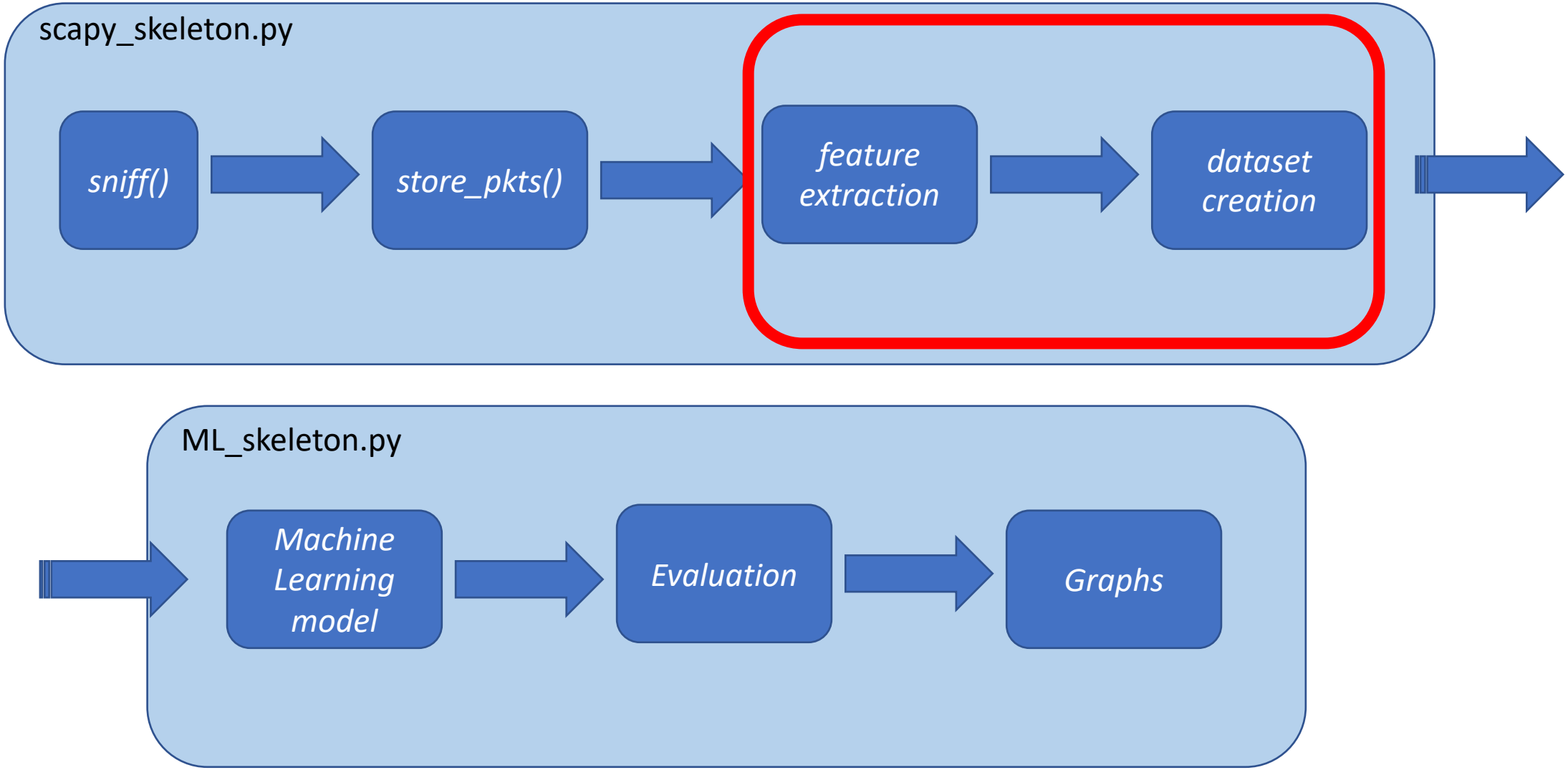
TCP

- Used for reliable connections
- Checks for: lost packets, transmission errors, packets out of order and so on….
- For example: emails, sms, internet browsing

UDP

- Used for unreliable connections with no sessions
- Does not check for error checking or flow control
- Sends packets and forgets
- For example: voice over IP, video streaming

Feature extraction



Feature extraction

- Given a dataset, a feature is an information that is particularly characteristic of that piece of data
- In a stream of data, that could be the amount of packets exchanged for the transmission of a web page, or their average size, or the time duration

Network flows

```
###[ Ethernet ]###
  dst       = 01:00:5e:53:64:6d
  src       = 0c:84:dc:3f:cf:6d
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 248
     id        = 16847
     flags     =
     frag      = 0
     ttl       = 1
     proto     = udp
     chksum    = 0x6881
     src       = 192.168.250.59
     dst       = 239.83.100.109
     \options   \
###[ UDP ]###
        sport     = 62877
        dport     = 33355
        len       = 228
        chksum    = 0x6ad6
```

- This is a single packet of a youtube video streaming
- How do we get the average size of exchanged packets?

Network flows

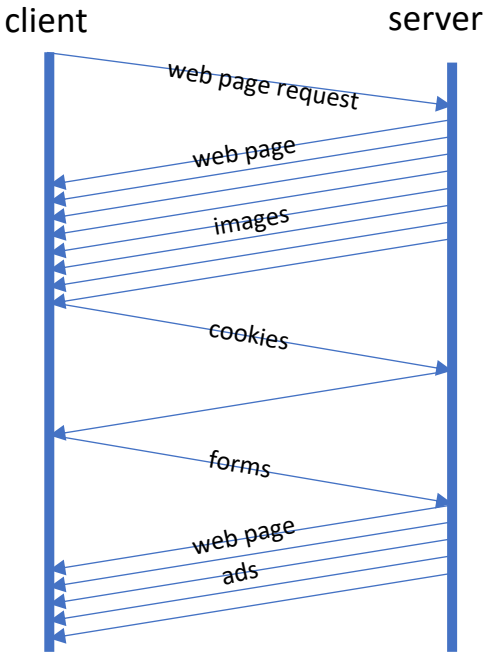
```
###[ Ethernet ]###
  dst       = 01:00:5e:53:64:6d
  src       = 0c:84:dc:3f:cf:6d
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 248
     id        = 16847
     flags     =
     frag      = 0
     ttl       = 1
     proto     = udp
     chksum    = 0x6881
     src       = 192.168.250.59
     dst       = 239.83.100.109
     \options   \
###[ UDP ]###
        sport     = 62877
        dport     = 33355
        len       = 228
        chksum    = 0x6ad6
```

- This is a single packet of a youtube video streaming
- How do we get the average size of exchanged packets?

Network flows

```
###[ Ethernet ]###
  dst       = 01:00:5e:53:64:6d
  src       = 0c:84:dc:3f:cf:6d
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 248
     id        = 16847
     flags     =
     frag      = 0
     ttl       = 1
     proto     = udp
     chksum    = 0x6881
     src       = 192.168.250.59
     dst       = 239.83.100.109
     \options   \
###[ UDP ]###
        sport     = 62877
        dport     = 33355
        len       = 228
        chksum    = 0x6ad6
```

- This is a single packet of a youtube video streaming
- How do we get the average size of exchanged packets?



Network flows

```
###[ Ethernet ]###
  dst       = 01:00:5e:53:64:6d
  src       = 0c:84:dc:3f:cf:6d
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 248
     id        = 16847
     flags     =
     frag      = 0
     ttl       = 1
     proto     = udp
     chksum    = 0x6881
     src       = 192.168.250.59
     dst       = 239.83.100.109
     \options   \
###[ UDP ]###
        sport     = 62877
        dport     = 33355
        len       = 228
        chksum    = 0x6ad6
```

- This is a single packet of a youtube video streaming
- How do we get the average size of exchanged packets?

- Network flows!
- A network flow is a tuple consisting of:
  - [src IP addr, src port, dest IP addr, dest port, tran proto]
- Transport protocol: TCP, UDP, …

Network flows

```
###[ Ethernet ]###
  dst       = 01:00:5e:53:64:6d
  src       = 0c:84:dc:3f:cf:6d
  type      = 0x800
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 248
     id        = 16847
     flags     =
     frag      = 0
     ttl       = 1
     proto     = udp
     chksum    = 0x6881
     src       = 192.168.250.59
     dst       = 239.83.100.109
     \options   \
###[ UDP ]###
        sport     = 62877
        dport     = 33355
        len       = 228
        chksum    = 0x6ad6
```

- Extract all packets that belong to the same **flow**
  - [src IP addr, src port, dest IP addr, dest port, tran proto]
- Extract the interested **value** from each packet of the flow and calculate a statistical measure (max, min, avg, std_dev…)

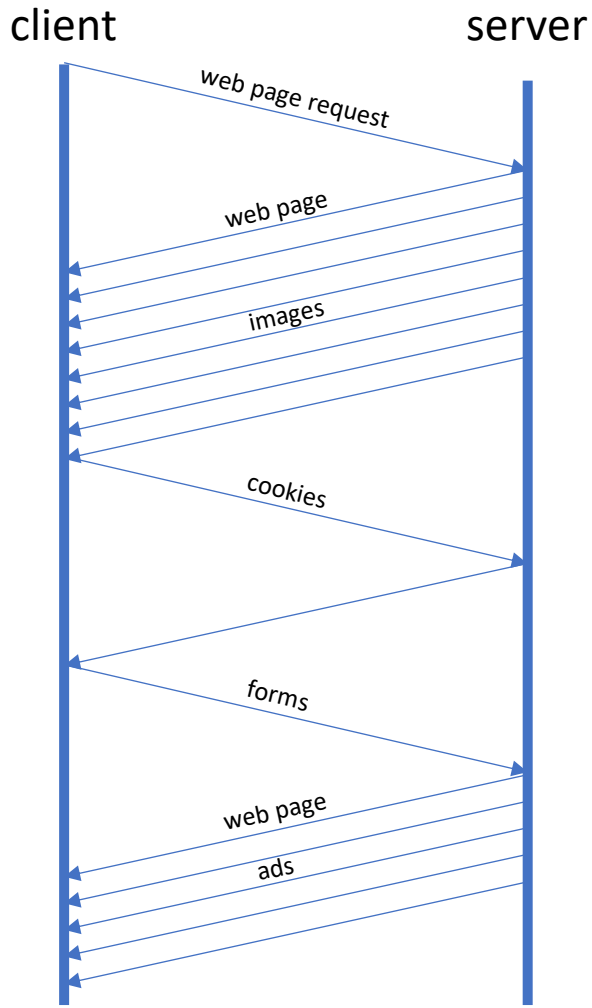
Tips for creating a dataset

- Consider using bi-directional flows
  - Put together features of mutual flows

A network flow is a tuple consisting of:

[src IP addr, src port, dest IP addr, dest port, tran proto]

[src IP addr, src port, dest IP addr, dest port, tran proto]



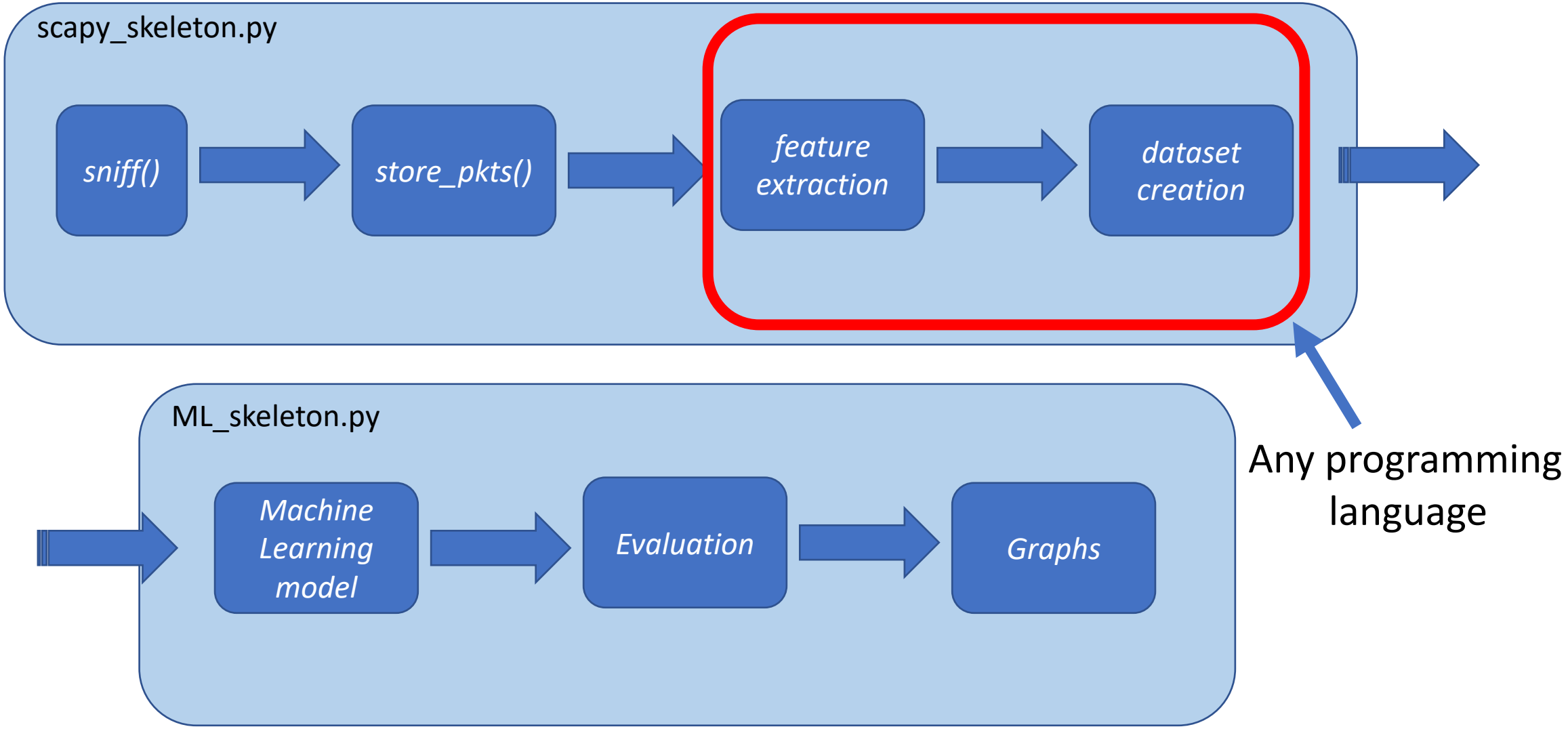
CSV FILE

2.1800001	79	702	57	67.303797	72.42155	0.02	0.0035355	2
1.3999999	111	701	57	68.351351	80.725646	0.02	0.0031623	2
1.6700001	113	708	57	69.575221	85.676698	0.03	0.0076795	2
1.9300001	96	702	57	74.885417	86.296413	0.0200002	0.0056765	2
1.49	114	72	59	59.912281	3.3354131	0.0300002	0.0104654	2
0.8900001	89	407	52	72.359551	66.426009	0.01	0.003147	0
3.75	275	414	52	67.603636	54.992295	0.28	0.0277526	0
0.9000001	389	255	52	63.784062	24.716739	0.0599999	0.0060068	0
1.9000001	118	194	52	75.822034	45.412569	0.1000001	0.0214271	0
1.9200001	136	228	52	78.639706	38.573196	0.04	0.0097954	0
1.3999999	458	1224	124	899.38646	166.33341	0.0599999	0.0095259	1
1.8199999	572	1225	100	916.07168	174.92576	0.0699999	0.0102383	1
1.6100001	563	1228	100	921.47602	178.90103	0.0599999	0.0076152	1
1.6000001	518	1224	124	971.90347	139.12581	0.0500002	0.0085671	1
1.73	524	1215	74	942.94084	176.12265	0.0999999	0.0089549	1

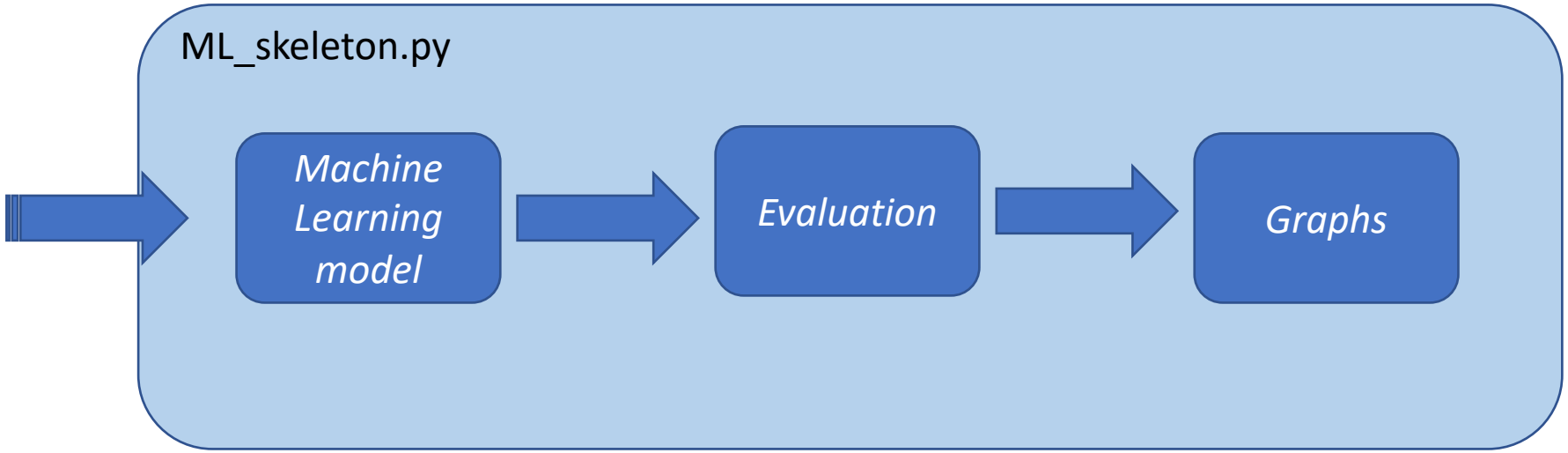
Features

Label

Framework


scapy_skeleton.py
sniff()
store_pkts()
feature extraction
dataset creation
Any programming language
ML_skeleton.py
Machine Learning model
Evaluation
Graphs

Machine Learning



Supervised Machine Learning

Training set

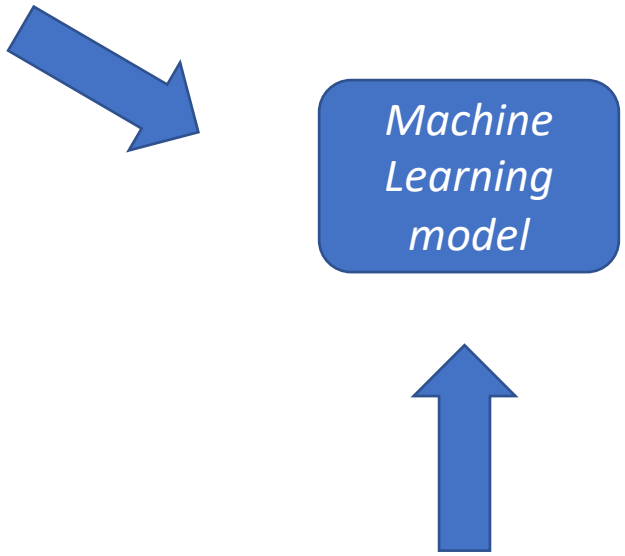
X-Values	Y-Values	Label
0.7	0.8	0
0.5	0.3	0
0.3	0.6	0
1.1	2.7	1
0.8	3.2	1
1.3	3	1
2.7	1.2	2
2.3	1.5	2
2.5	1.7	2

Machine Learning model

Supervised Machine Learning

Training set

X-Values	Y-Values	Label
0.7	0.8	0
0.5	0.3	0
0.3	0.6	0
1.1	2.7	1
0.8	3.2	1
1.3	3	1
2.7	1.2	2
2.3	1.5	2
2.5	1.7	2



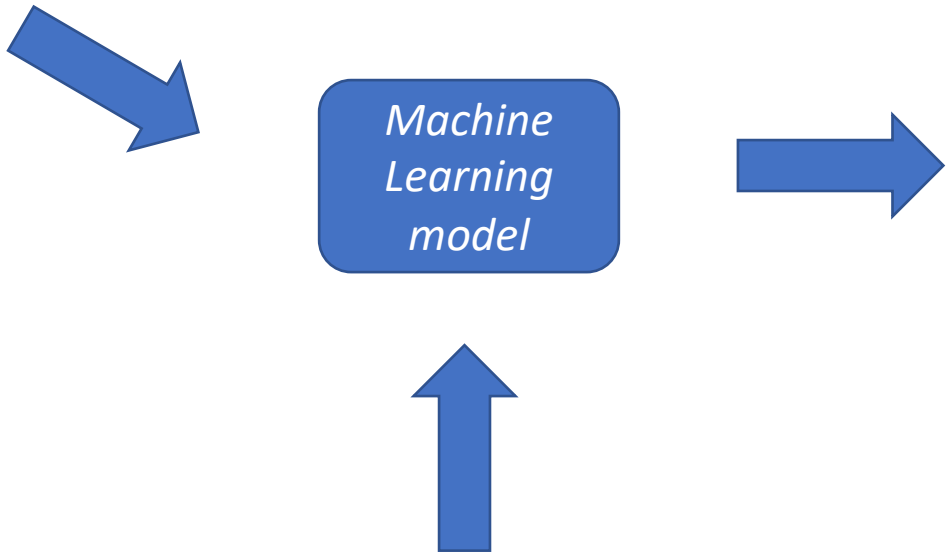
Testing set

X-Values	Y-Values	Labels
0.6	1	?
1.2	2.9	?
2.6	1.4	?

Supervised Machine Learning

Training set

X-Values	Y-Values	Label
0.7	0.8	0
0.5	0.3	0
0.3	0.6	0
1.1	2.7	1
0.8	3.2	1
1.3	3	1
2.7	1.2	2
2.3	1.5	2
2.5	1.7	2



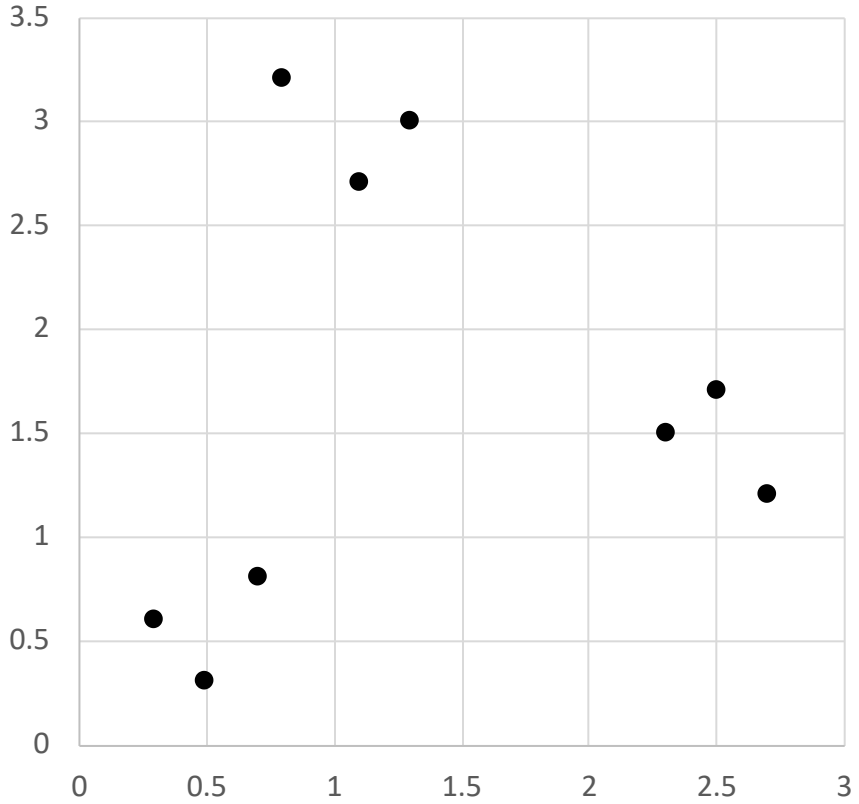
Label
0
1
2

Testing set

X-Values	Y-Values	Labels
0.6	1	?
1.2	2.9	?
2.6	1.4	?

Supervised Machine Learning

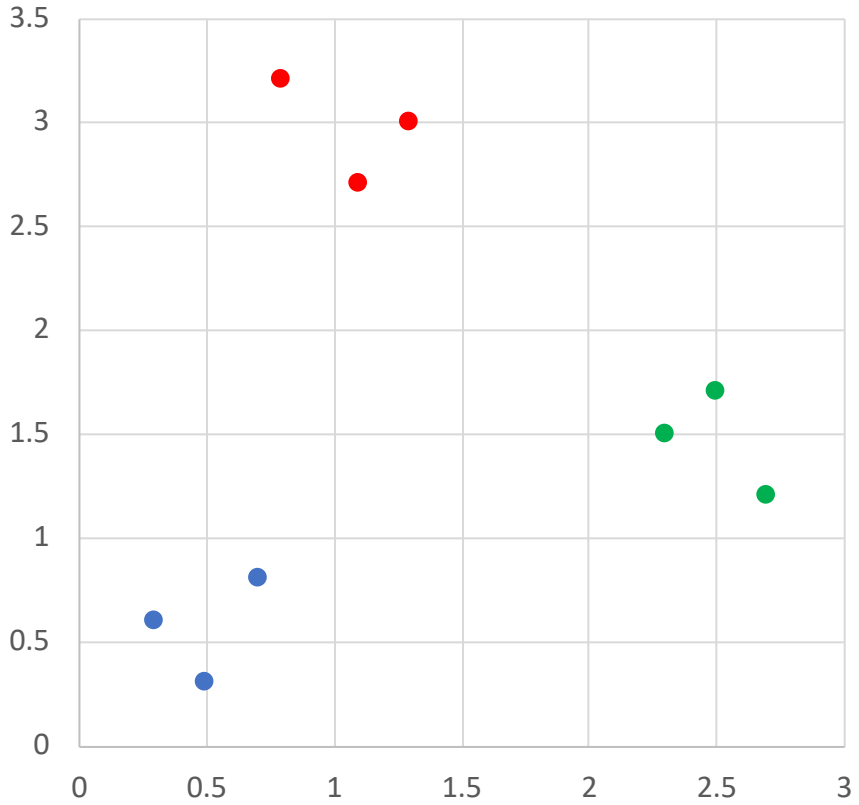
X-Values	Y-Values
0.7	0.8
0.5	0.3
0.3	0.6
1.1	2.7
0.8	3.2
1.3	3
2.7	1.2
2.3	1.5
2.5	1.7



Supervised Machine Learning

Training set

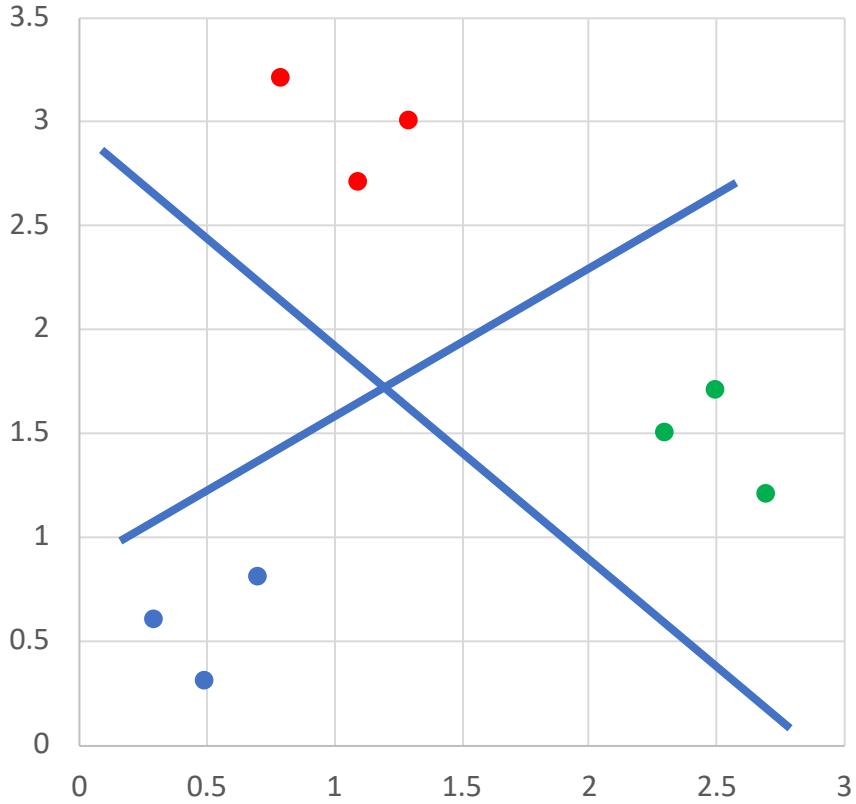
X-Values	Y-Values	Label
0.7	0.8	0
0.5	0.3	0
0.3	0.6	0
1.1	2.7	1
0.8	3.2	1
1.3	3	1
2.7	1.2	2
2.3	1.5	2
2.5	1.7	2



After training the ML model

Training set

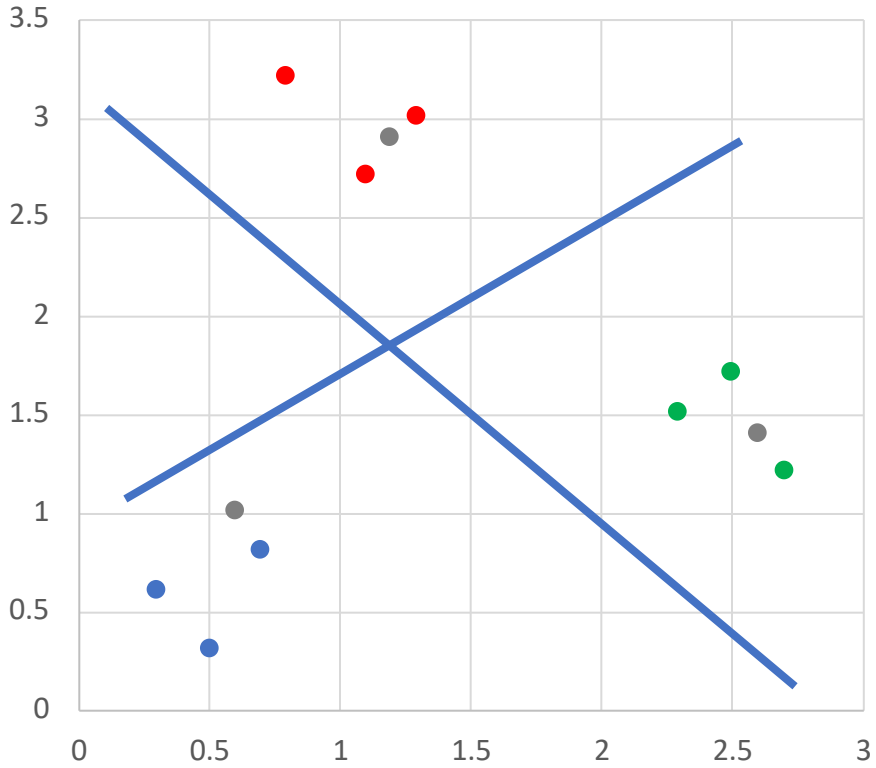
X-Values	Y-Values	Label
0.7	0.8	0
0.5	0.3	0
0.3	0.6	0
1.1	2.7	1
0.8	3.2	1
1.3	3	1
2.7	1.2	2
2.3	1.5	2
2.5	1.7	2



Supervised Machine Learning

Training set

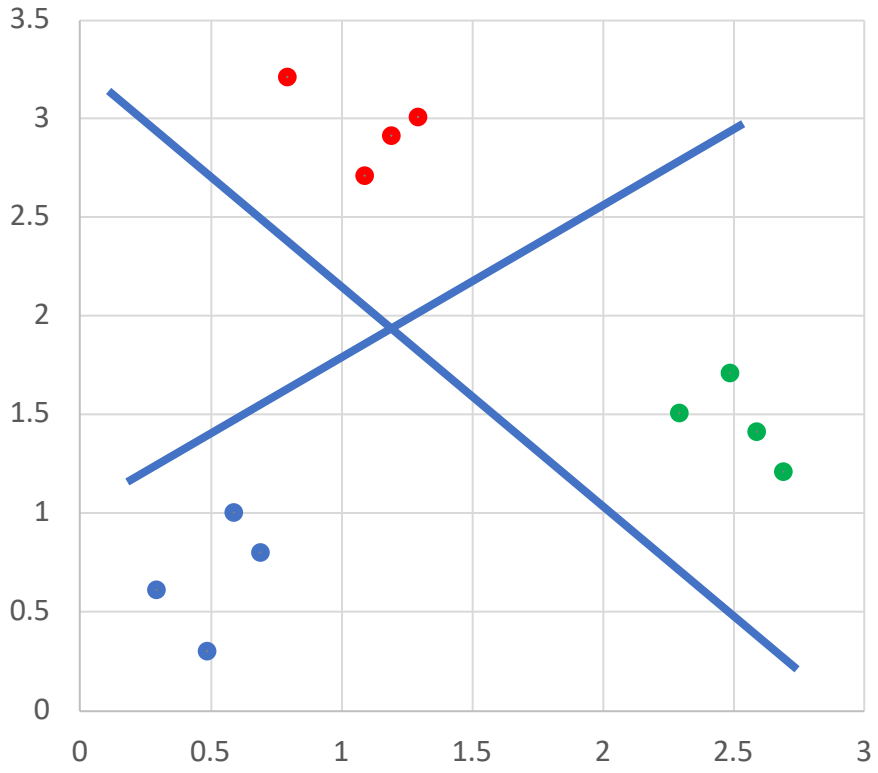
X-Values	Y-Values	Labels
0.6	1	?
1.2	2.9	?
2.6	1.4	?



Supervised Machine Learning

Training set

X-Values	Y-Values	Label
0.6	1	0
1.2	2.9	1
2.6	1.4	2



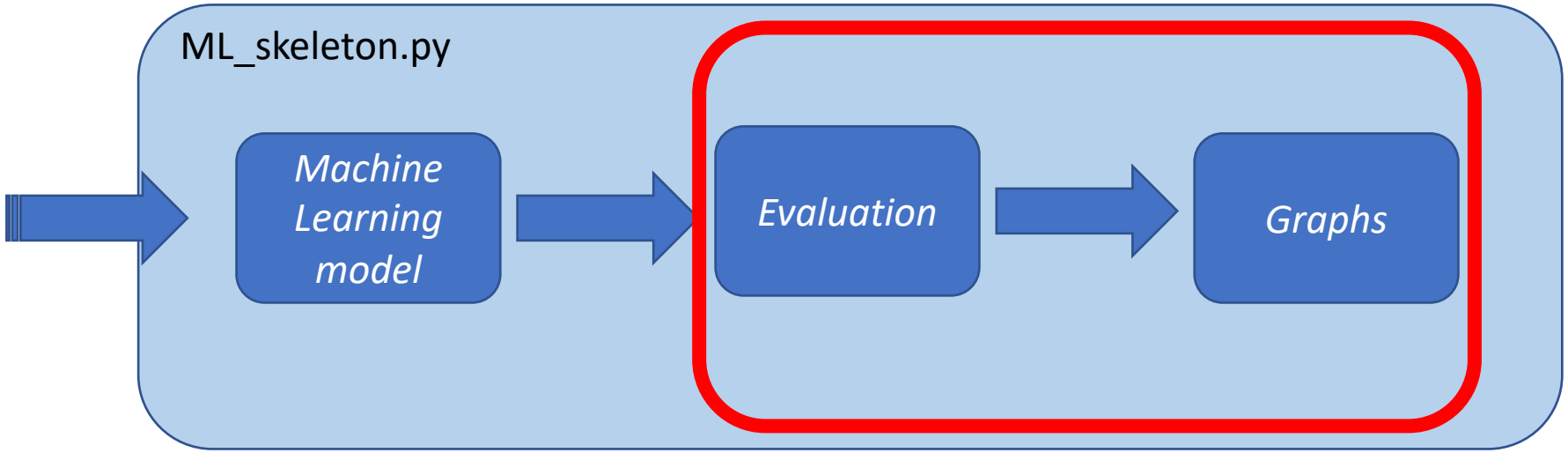
How to create a dataset

X-Values	Y-Values	Label
0.7	0.8	0
0.5	0.3	0
0.3	0.6	0
1.1	2.7	1
0.8	3.2	1
1.3	3	1
2.7	1.2	2
2.3	1.5	2
2.5	1.7	2

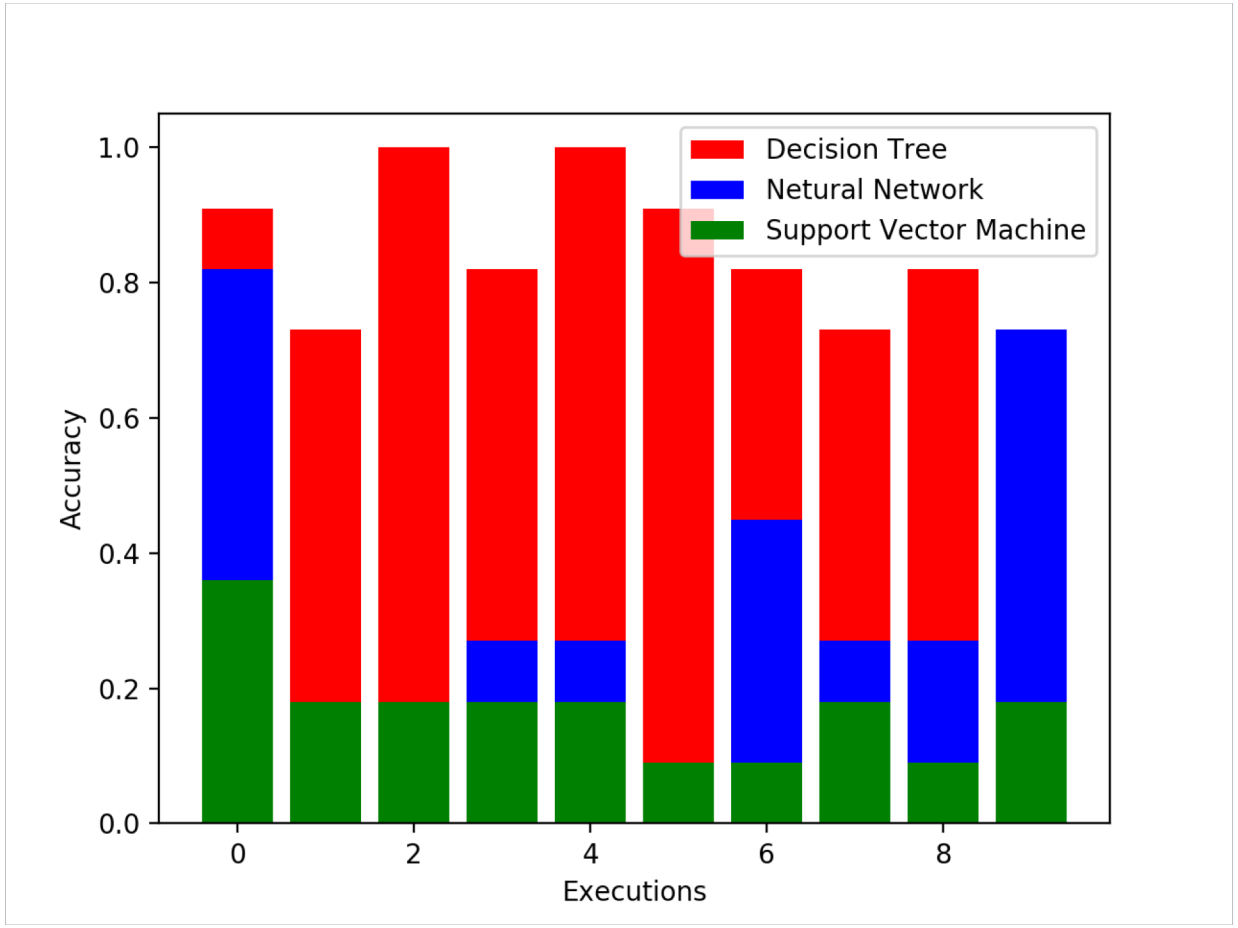
Tips for creating a dataset

- Close all applications except the one you collect data for
- Collect one activity at a time
- Clean noise from code
  - Keep only the flow with the highest amount of packets
- Use bi-directional flows

Machine Learning



Evaluation


Decision Tree
Netural Network
Support Vector Machine
Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
0
2
4
6
8
Executions

LIVE SESSION!

IF YOU HAVE QUESTIONS THAT YOU THINK MIGHT BE USEFUL TO EVERYONE, I STRONGLY ENCOURAGE YOU TO USE THE ‘DISCUSSIONS’ SECTION IN CANVAS, SO THAT EVERYONE CAN TAKE ADVANTAGE OF THE SHARED INFORMATION