Machine learning for machine data

David Andrzejewski - @davidandrzej

Data Sciences, Sumo Logic

Strata Conference – Machine Data Track

February 13, 2014

This talk: Machine Learning + Machine Data = Awesome!

- **YES**
  - overview of log data
  - solving log data problems with machine learning
  - specific examples
    - (mostly) Sumo Logic-related
    - customer use cases
  - general lessons learned

This talk: Machine Learning + Machine Data = Awesome!

- **NO** (or, not much)
  - Sumo Logic deep dive
  - Tech stack talk
    - In-memory Hadoop for real-time Cassandra SQL in hybrid clouds
  - Big data "shock and awe"
    - 800 yottabytes / second ZOMG!!11!!
  - Algorithm shootout
    - Deep learning vs random forests vs SVMs vs coin flips vs ...
  - Extreme math

The estimate E *is* asymptotically almost unbiased *in the sense that*

$$\frac{1}{n}\mathbb{E}_n(E) \underset{n\to\infty}{=} 1 + \delta_1(n) + o(1), \text{ where } |\delta_1(n)| < 5 \cdot 10^{-5} \text{ as soon as } m \geq 16.$$

Context: me

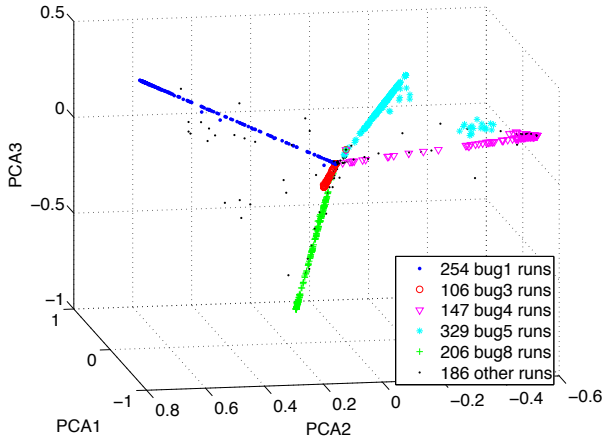
- Data sciences @ Sumo Logic
- Co-organizer @ SF ML Meetup
- Previous
  - Post-doc in knowledge discovery

sumologic

$\hat{f}(x)$

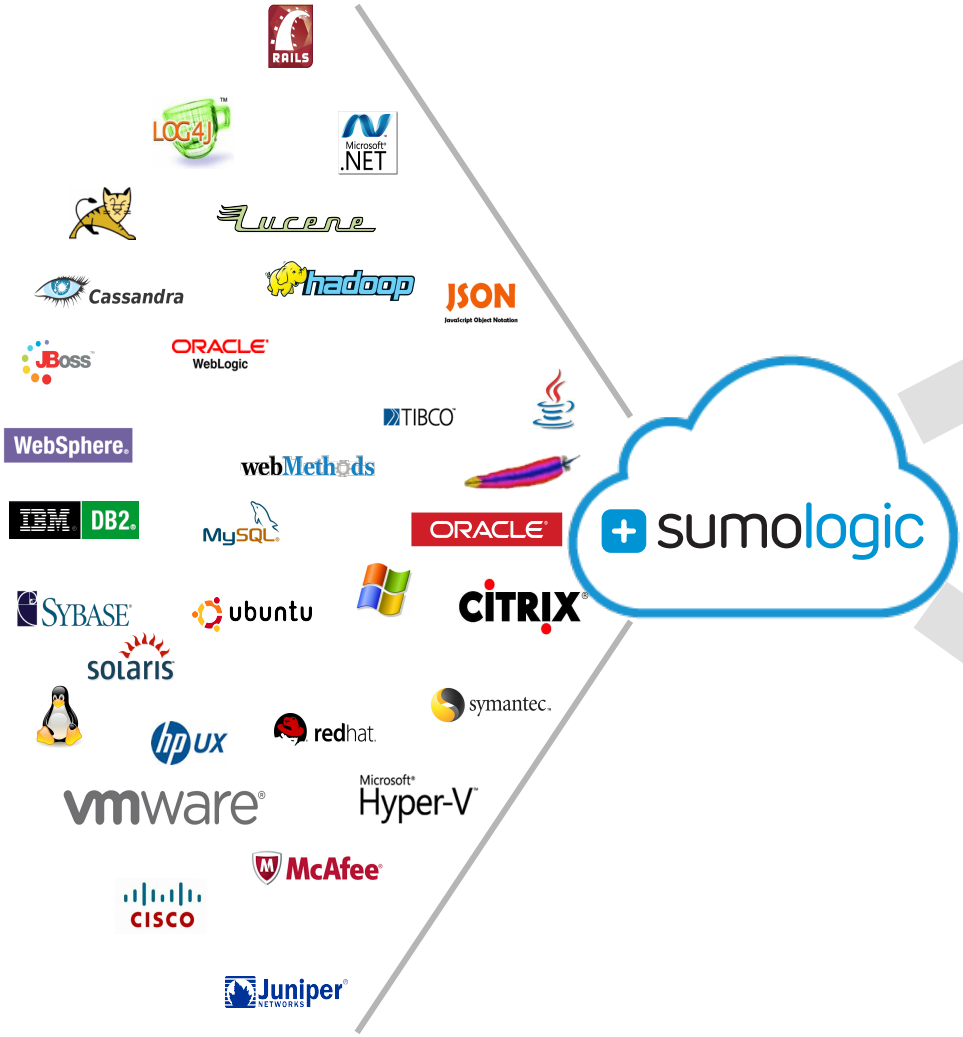
Lawrence Livermore National Laboratory

- Even more previous machine data research projects
  - University of Wisconsin--Madison
  - Microsoft Research

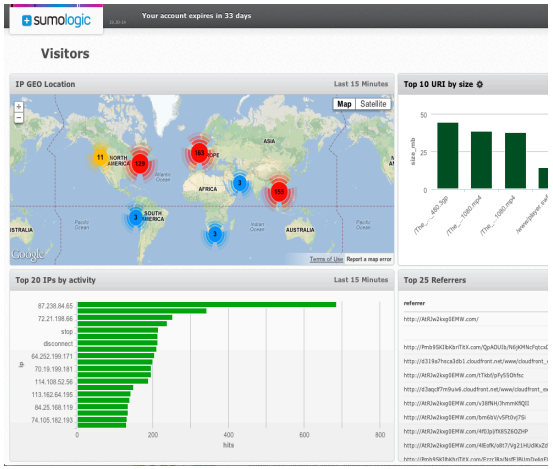


Context: Sumo Logic

"Turning Machine Data Into IT and Business Insights"



Search, monitor, visualize



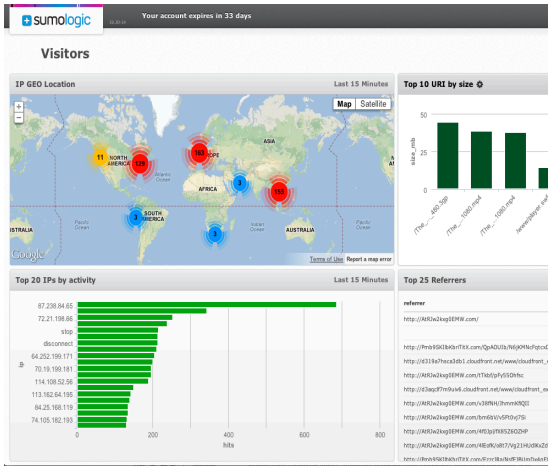
Learn, classify, predict

sumologic

Context: Sumo Logic

"Turning Machine Data Into IT and Business Insights"

Search, monitor, visualize



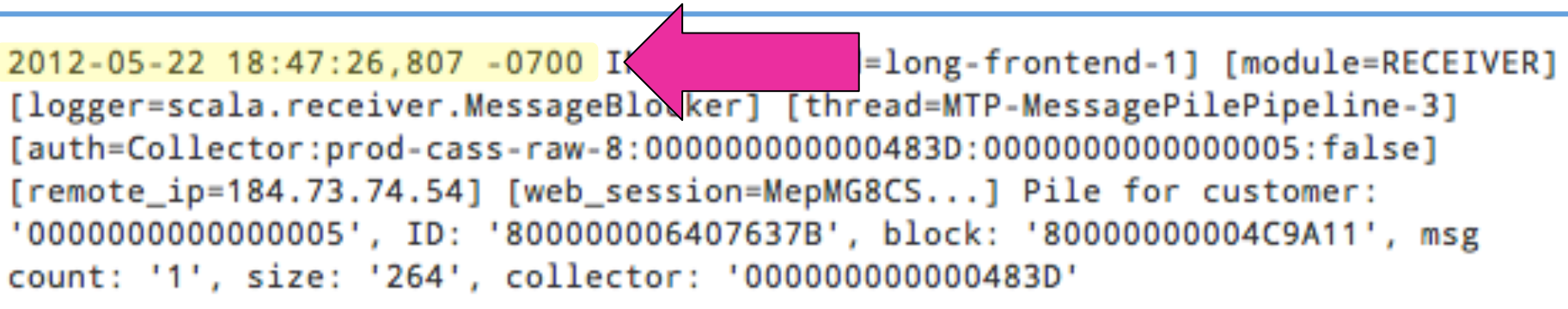
Learn, classify, predict

sumologic

Anatomy of a log message: Five W's

```
2012-05-22 18:47:26,807 -0700 INFO  [hostId=long-frontend-1] [module=RECEIVER]
[logger=scala.receiver.MessageBlocker] [thread=MTP-MessagePilePipeline-3]
[auth=Collector:prod-cass-raw-8:0000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] Pile for customer:
'0000000000000005', ID: '8000000064076378', block: '80000000004C9A11', msg
count: '1', size: '264', collector: '0000000000000483D'
```

Anatomy of a log message: Five W's

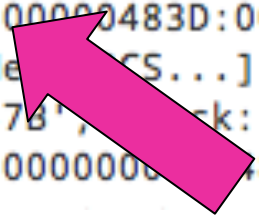


- **When?** Timestamp <u>with time zone</u>

sumologic

Anatomy of a log message: Five W's

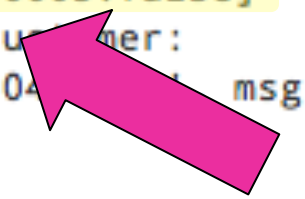
```
2012-05-22 18:47:26,807 -0700 INFO  [hostId=long-frontend-1] [module=RECEIVER]
[logger=scala.receiver.MessageBlocker] [thread=MTP-MessagePilePipeline-3]
[auth=Collector:prod-cass-raw-8:0000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=Me...CS...] Pile for customer:
'0000000000000005', ID: '8000000064076378', ...k: '80000000004C9A11', msg
count: '1', size: '264', collector: '0000000...483D'
```



- **When?** Timestamp with time zone
- **Where?** Host, module, code location

Anatomy of a log message: Five W's

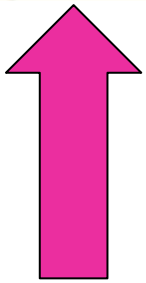
```
2012-05-22 18:47:26,807 -0700 INFO  [hostId=long-frontend-1] [module=RECEIVER]
[logger=scala.receiver.MessageBlocker] [thread=MTP-MessagePilePipeline-3]
[auth=Collector:prod-cass-raw-8:0000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] Pile for cu[illegible]mer:
'0000000000000005', ID: '8000000064076378', block: '8000000004[illegible]', msg
count: '1', size: '264', collector: '0000000000000483D'
```



- **When?** Timestamp with time zone
- **Where?** Host, module, code location
- **Who?** Authentication context

Anatomy of a log message: Five W's

```
2012-05-22 18:47:26,807 -0700 INFO  [                -frontend-1] [module=RECEIVER]
[logger=scala.receiver.MessageBlocker] [thread=MTP-MessagePilePipeline-3]
[auth=Collector:prod-cass-raw-8:000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] Pile for customer:
'0000000000000005', ID: '8000000064076378', block: '80000000004C9A11', msg
count: '1', size: '264', collector: '000000000000483D'
```



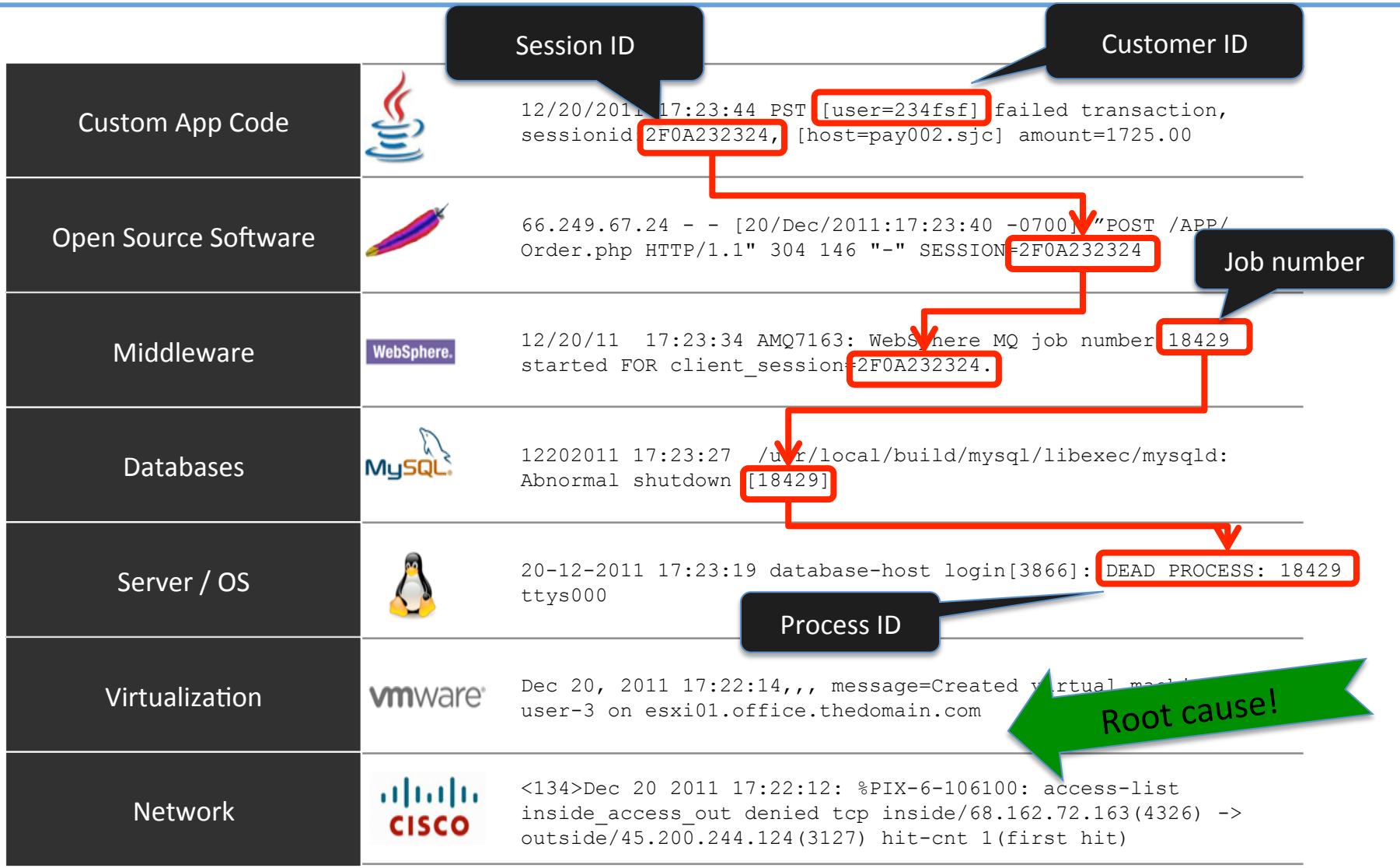
- **When?** Timestamp with time zone
- **Where?** Host, module, code location
- **Who?** Authentication context
- **What?** Log level and key-value pairs

What's missing

```
2012-05-22 18:47:26,807 -0700 INFO  [hostId=long-frontend-1] [module=RECEIVER]
[logger=scala.receiver.MessageBlocker] [thread=MTP-MessagePilePipeline-3]
[auth=Collector:prod-cass-raw-8:0000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] Pile for customer:
'0000000000000005', ID: '8000000064076378', block: '80000000004C9A11', msg
count: '1', size: '264', collector: '0000000000000483D'
```

sumologic

Traversing the stack



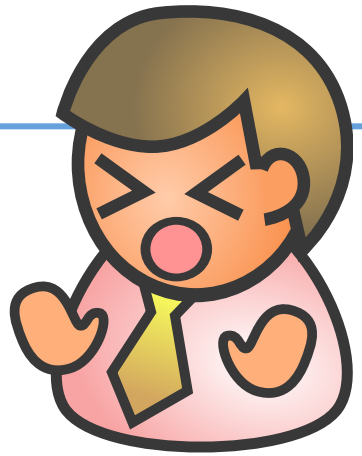
sumologic

Log use cases – “organizational perception”

Enhanced visibility into machine behaviors

- Compliance
  - Operational (SLA)
  - Regulatory (audits)
  - Security
- Availability / performance
  - Faster MTTR
- Business insights ($$$)

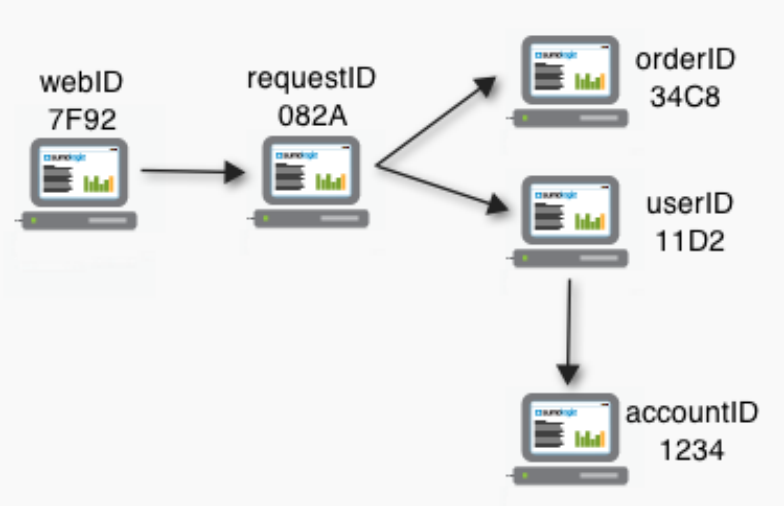
Log challenges



- (wildly) varying formats
  - printf, JSON, XML, Windows, X-delimited, ...
- Specialized knowledge

```
[2008-05-07 09:50:08.450 'App' 3560 verbose] [VpxdHeartbeat] Invalid
heartbeat from 10.17.218.46
```

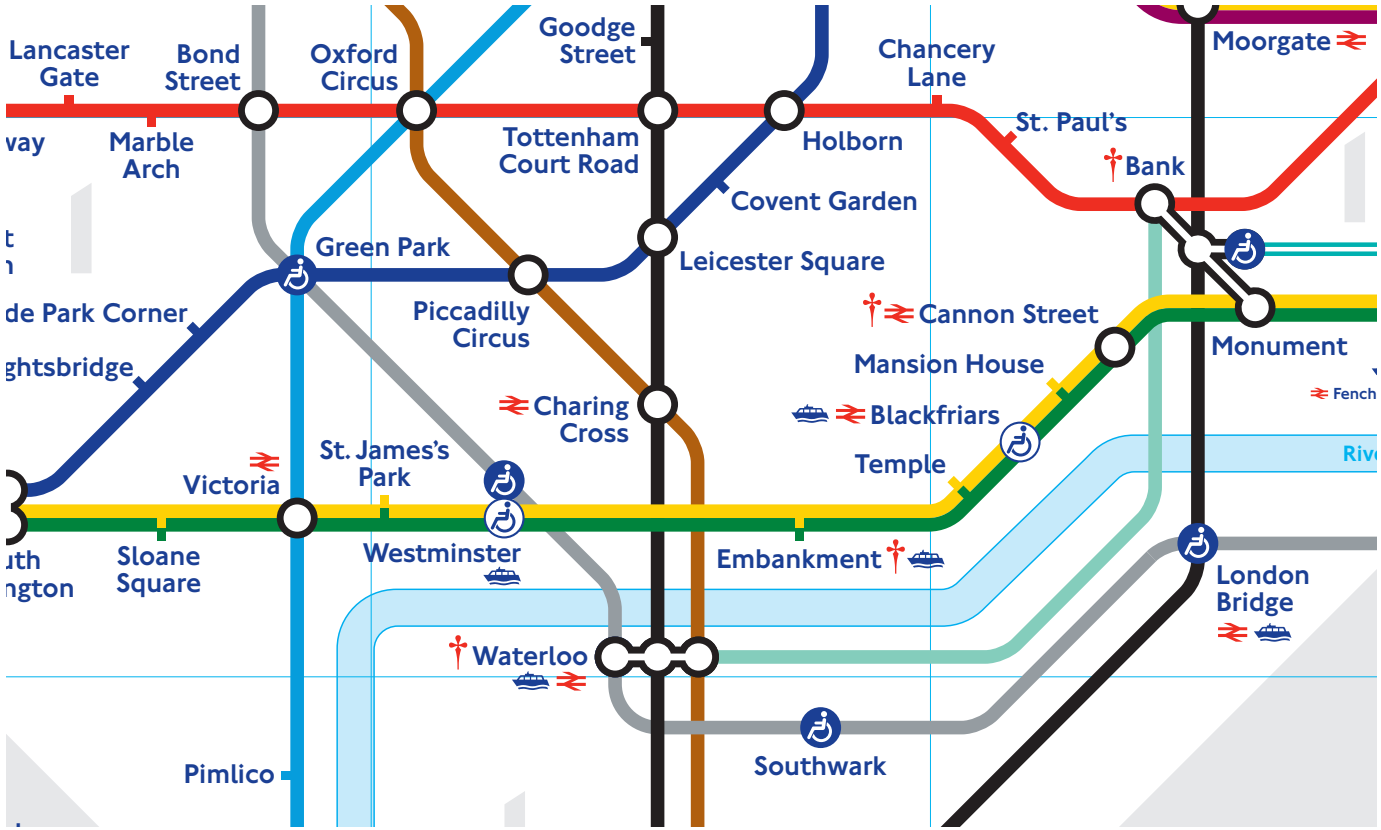
- Noise
- Cascading failures

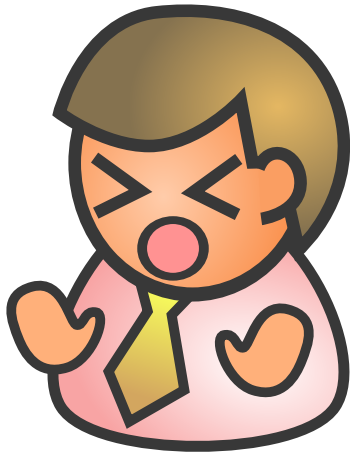


sumologic

Complexity

"A distributed system is one in which the failure of a computer you didn't even know existed can render your own computer unusable." - Leslie Lamport

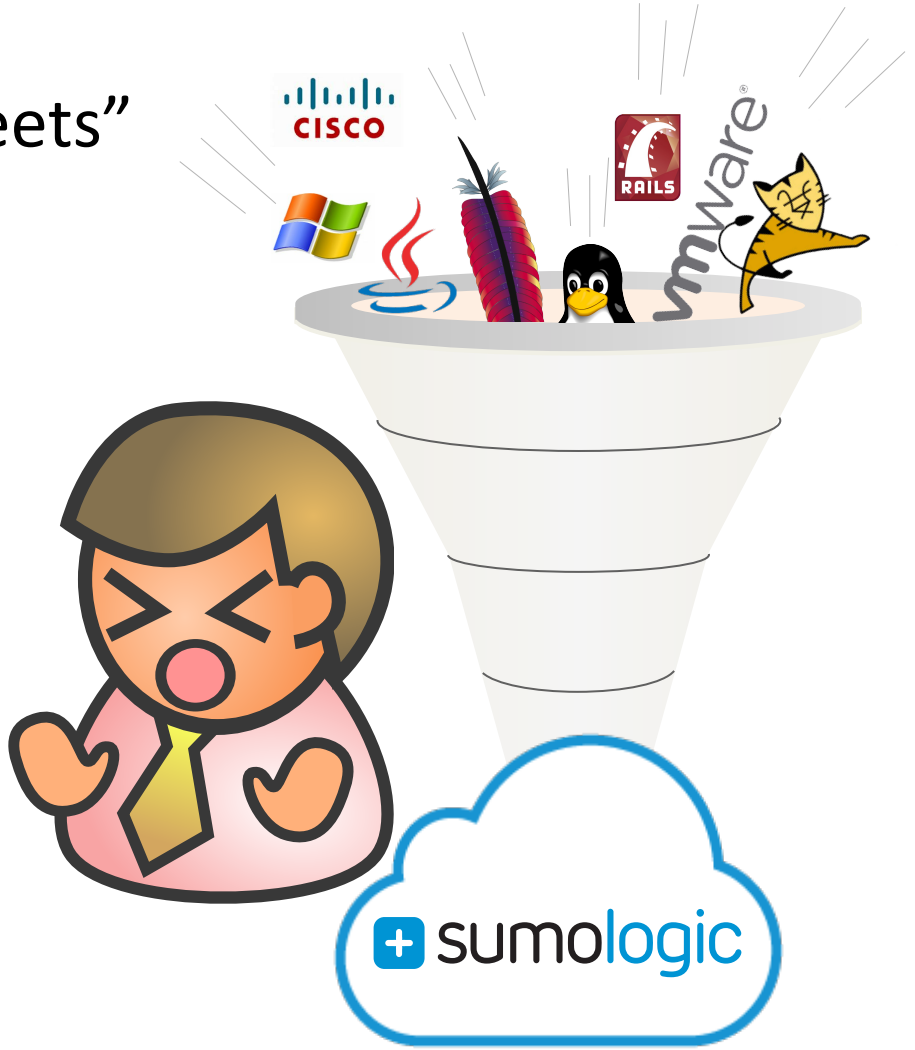




sumologic

“OMG java.lang.NullPointerException #fail”

- Logs: like “computer tweets”
- Twitter 2013*
  - Peak @ ~144k TPS
  - **Avg ~6k tweets / second**
- Log data
  - Example: 1 TB / day
  - **Avg ~25k logs / second**



* https://blog.twitter.com/2013/new-tweets-per-second-record-and-how

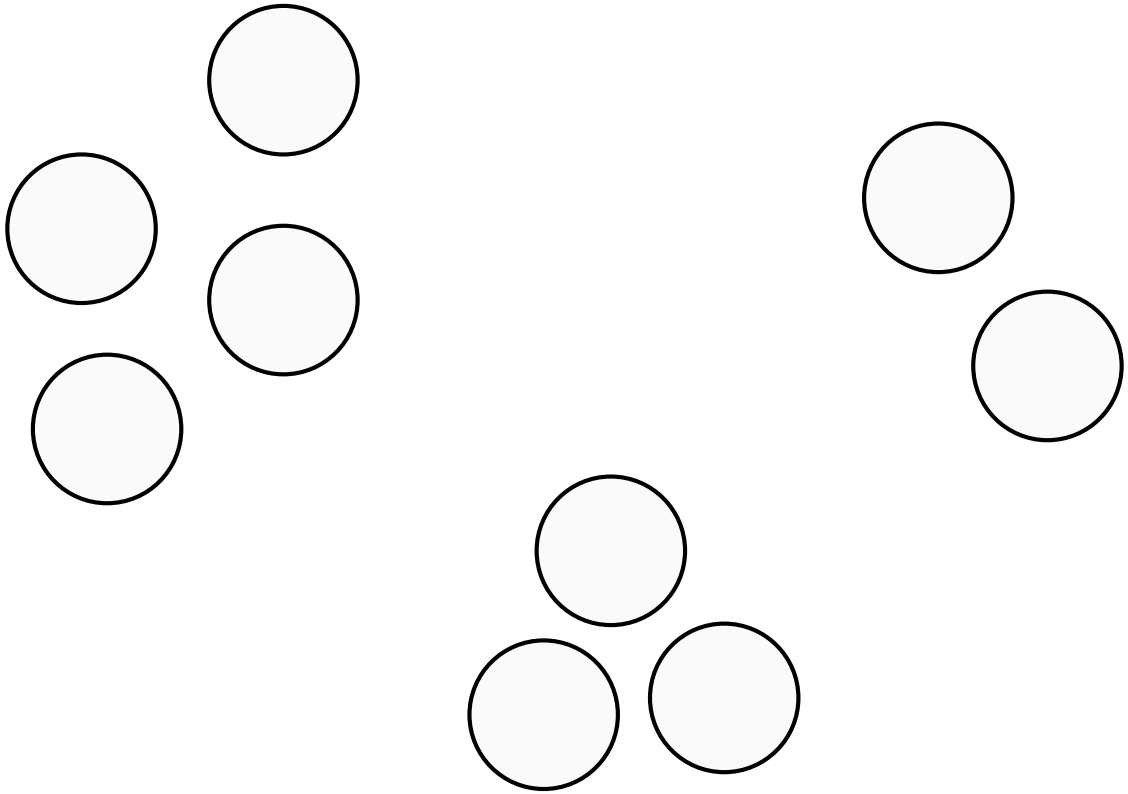
Systems that **learn from experience**

sumologic

$\hat{f}(x)$

Unsupervised clustering

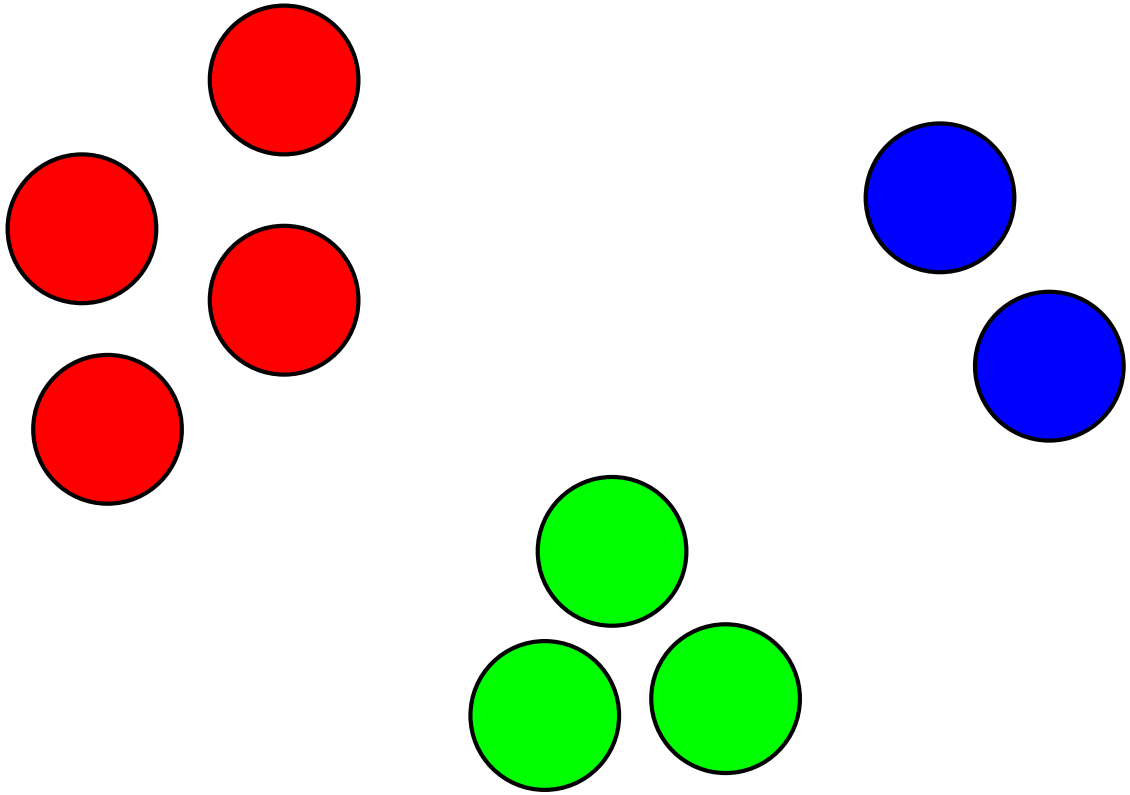
- **Given:** set of items
- **Do:** group similar items



Unsupervised clustering

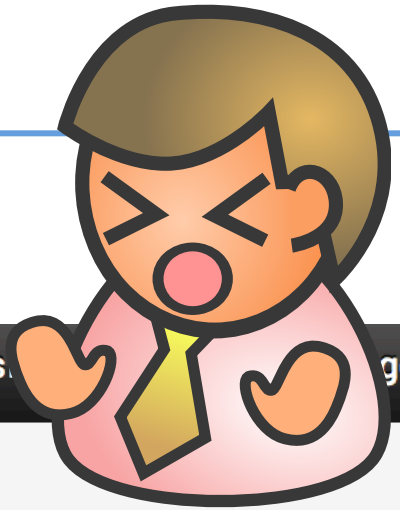
$\hat{f}(x)$

- **Given:** set of items
- **Do:** group similar items



Too many logs! “data disorientation”

~60k results: 30 minutes, one component



sumologic 20.1-2846 Search Anomalies Das… ge ▾

Unnamed Search | Unnamed Search | +

7:30 PM STATUS: Done gathering results ELAPSED TIME: 00:00:06 RESULTS: 59,063 SESS

Messages

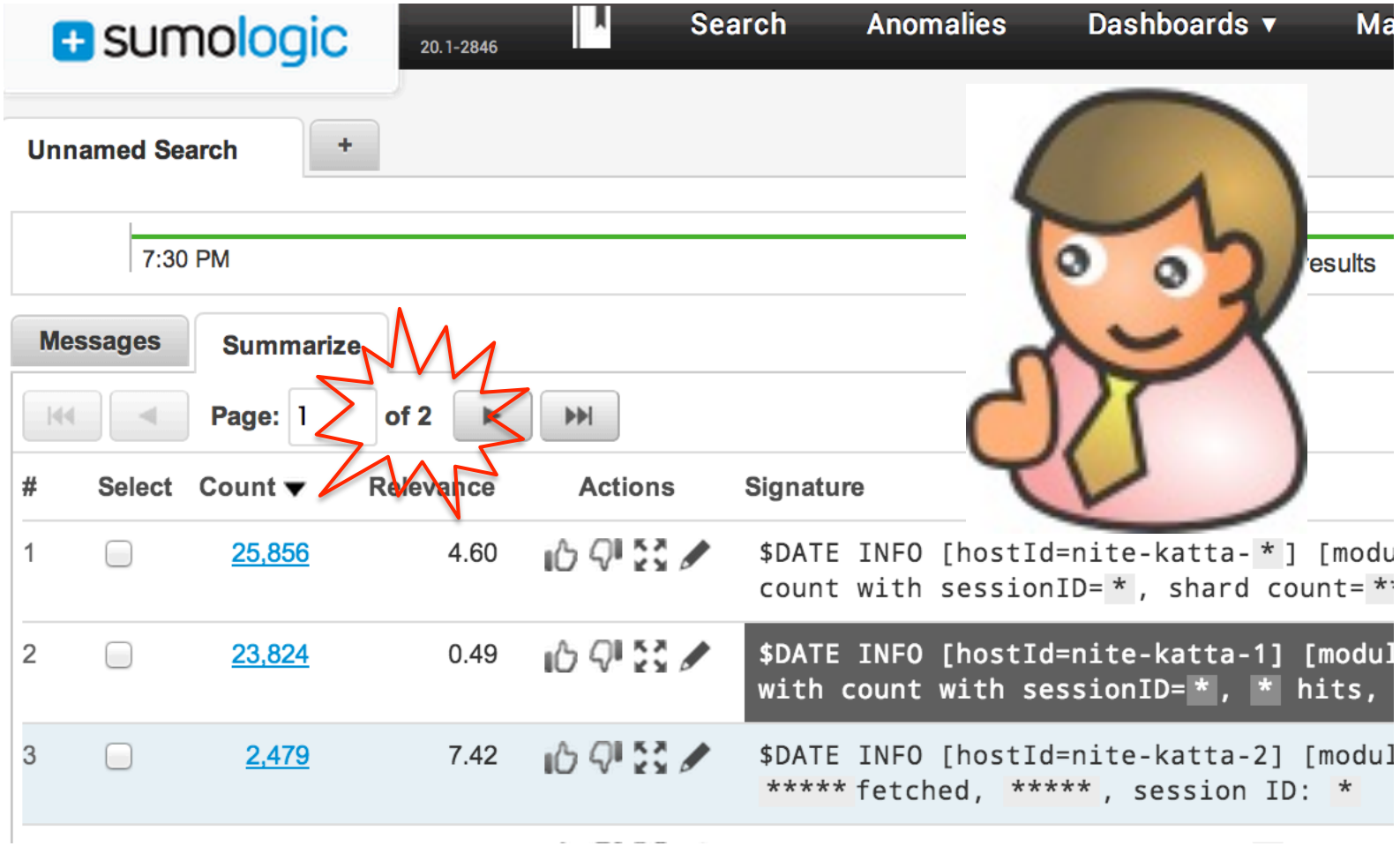
Page: 1 of 3938 LogReduce

Time	Message
02/05/2014 19:59:54.333	2014-02-05 19:59:54,333 -0800 INFO [hostId=nite-katta-1] [module=KATTA] [logger=katta_sumo.node.ShardDiskCache] [thread=160184096@qtp0-14] Shard waiting, session ID: FFFFFFFFFFFFFFF Host: nite-katta-1 ▾ Name: /usr/sumo/katta-sumo-20.1-1821/logs/katta.log ▾ Category: katta ▾

Distill logs down to **underlying structure**

```
$DATE INFO [hostId=stag-katta-*] [module=KATTA]
[localUserName=katta] [logger=katta_sumo.node.FetchQueue]
[thread=ShardDiskCache-*] Queue wait time for object:
'********#shard', ms: '*' with queue depth: '*',
immediate: '****', fetch time: '*', total time: '**'
```

LogReduce: results "compressed" **~1000x**



In the beginning, there was the printf()

```
printf("Health status check: %s is %s”,
       hostid, hoststatus)
```

Log generation

```
Health status check: zim-5 is OK
Health status check: gir-3 is OK
Health status check: gir-2 is TIMED OUT
Health status check: dib-1 is OK
```

Reverse engineering printf()

```
printf("Health status check: %s is %s",
       hostid, hoststatus)
```

Log generation

```
Health status check: zim-5 is OK
Health status check: gir-3 is OK
Health status check: gir-2 is TIMED OUT
Health status check: dib-1 is OK
```

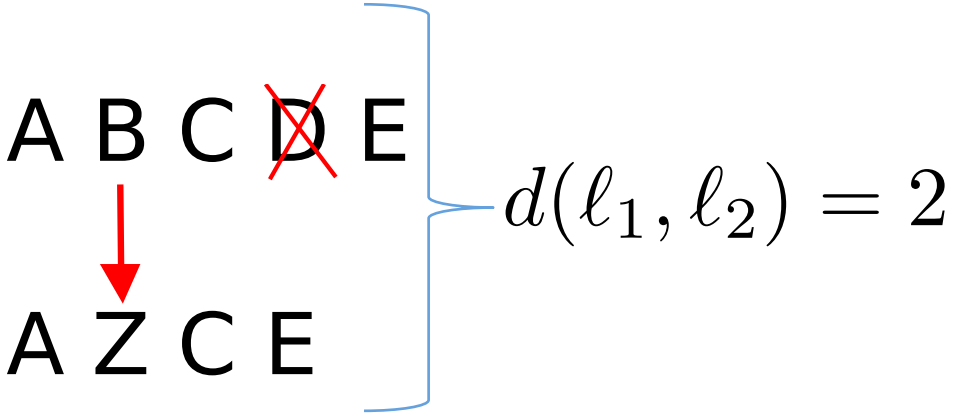

```
Health status check: *** is ***
```

$\hat{f}(x)$

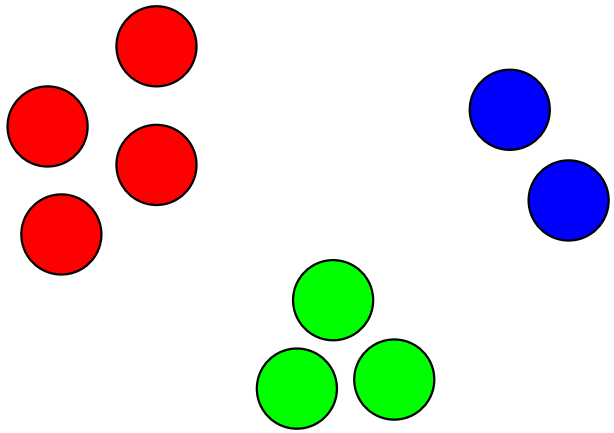
Unsupervised clustering

- **Given:** log messages
- **Do:** group by “signature”

1. Define string **distance function** (e.g., Левенште́йн)



2. Do **distance-based clustering**



Drill-down into the original raw logs

Messages | Summarize

Page: 1 of 2

#	Select	Count	Actions	Signature
1		8,497		$DATE INFO [hostIc [thread=IPC Server shards=[00000000000

2013-04-24 09:20:53,997 -0700 INFO [hostId=nite-katta-1] [module=KATTA [thread=IPC Server handler 8 on 20000] STARTED Calling getDetailsBatch shards=[0000000000000131-99F822FECEBFE19C#shard], docIds.length=2500

Host: nite-katta-1 Name: /usr/sumo/katta-sumo-20.1-658/logs/katta.log Category

2013-04-24 09:20:53,674 -0700 INFO [hostId=nite-katta-1] [module=KATTA [thread=IPC Server handler 6 on 20000] FINISHED Calling getDetailsBatch shards=[0000000000000131-99F822FECEBFE19C#shard], docIds.length=2497 af

Host: nite-katta-1 Name: /usr/sumo/katta-sumo-20.1-658/logs/katta.log Category

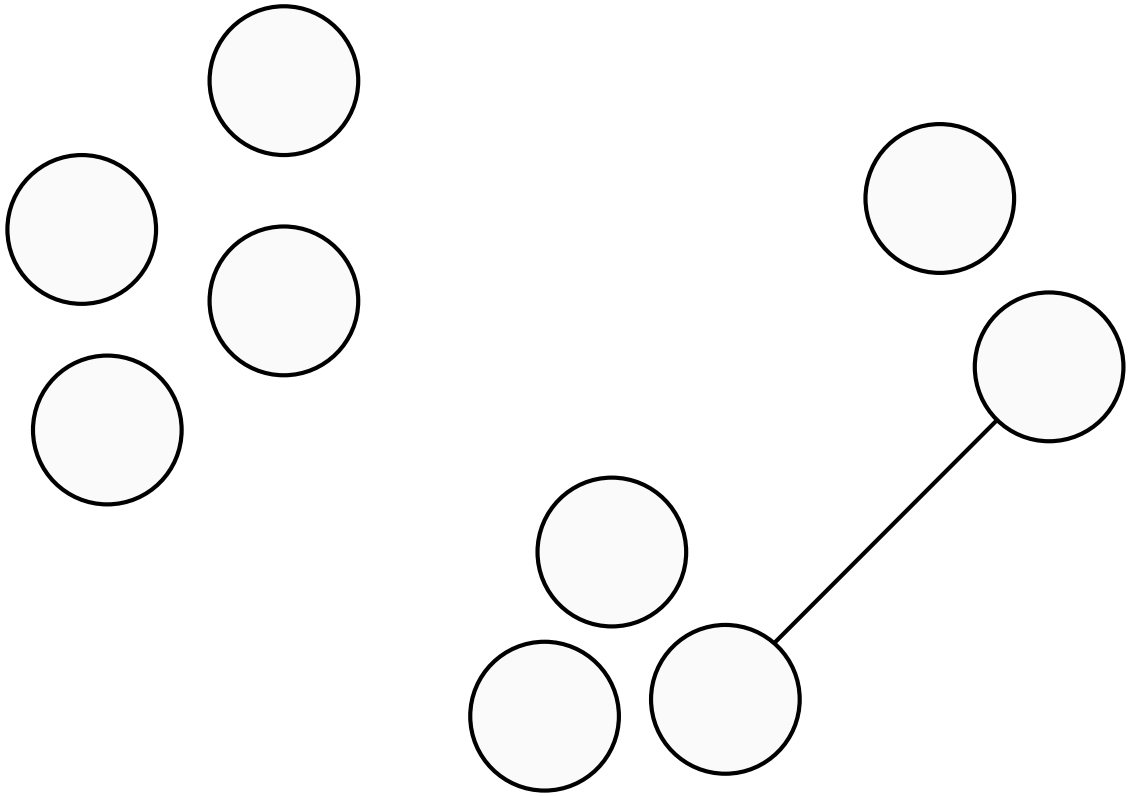
2013-04-24 09:20:53,462 -0700 INFO [hostId=nite-katta-2] [module=KATTA

sumologic

$\hat{f}(x)$

Partially supervised clustering

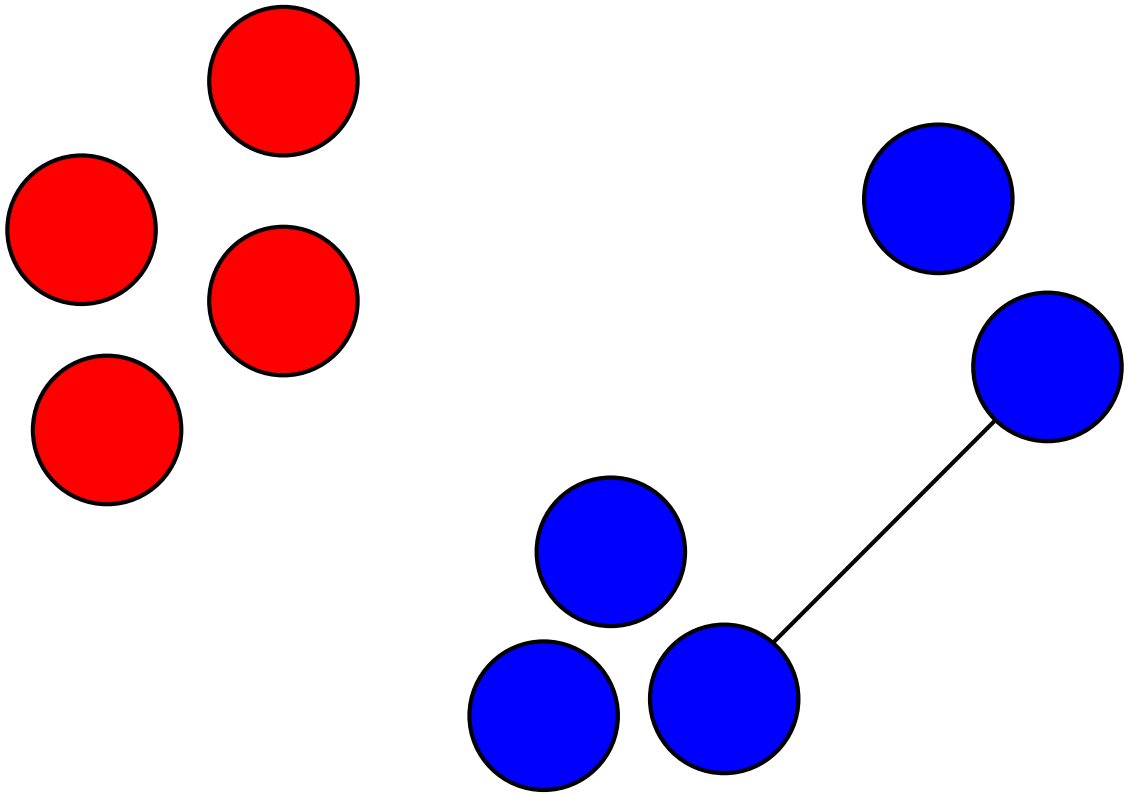
- **Given:** set of items + side info
- **Do:** group similar items



$\hat{f}(x)$

Partially supervised clustering

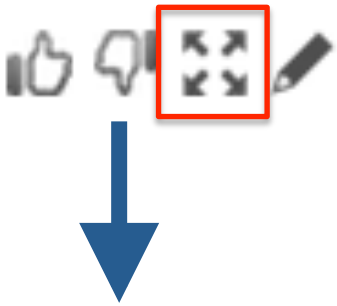
- **Given:** set of items + side info
- **Do:** group similar items



Too many wildcards!

```
$DATE INFO  [hostId=long-katta-*] [module=KATTA
[thread=IPC Server handler * on 20000] *****ED
shards=[0000000000000005-*****
```

“Hint” from human user



```
$DATE INFO  [hostId=long-katta-*] [module=KATTA
[thread=IPC Server handler * on 20000] *****ED
shards=[0000000000000005-*****
```

```
$DATE INFO  [hostId=long-katta-17] [module=KATTA]
[thread=IPC Server handler * on 20000] FINISHED C
sessionID=89B99A496F555E41, shards=[0000000000000
```

```
$DATE INFO  [hostId=long-katta-1*] [module=KATTA]
[thread=IPC Server handler * on 20000] STARTED Ca
sessionID=8B6BB3EC5AA0B6E8, shards=[0000000000000
```

Not enough wildcards!

```
$DATE INFO  [hostId=long-frontend-1]
[logger=scala.config.protocol.handler
[auth=User:scott@sumologic.com:000000
[remote_ip=173.228.89.151] [web_sessi
```

“Hint” from human user

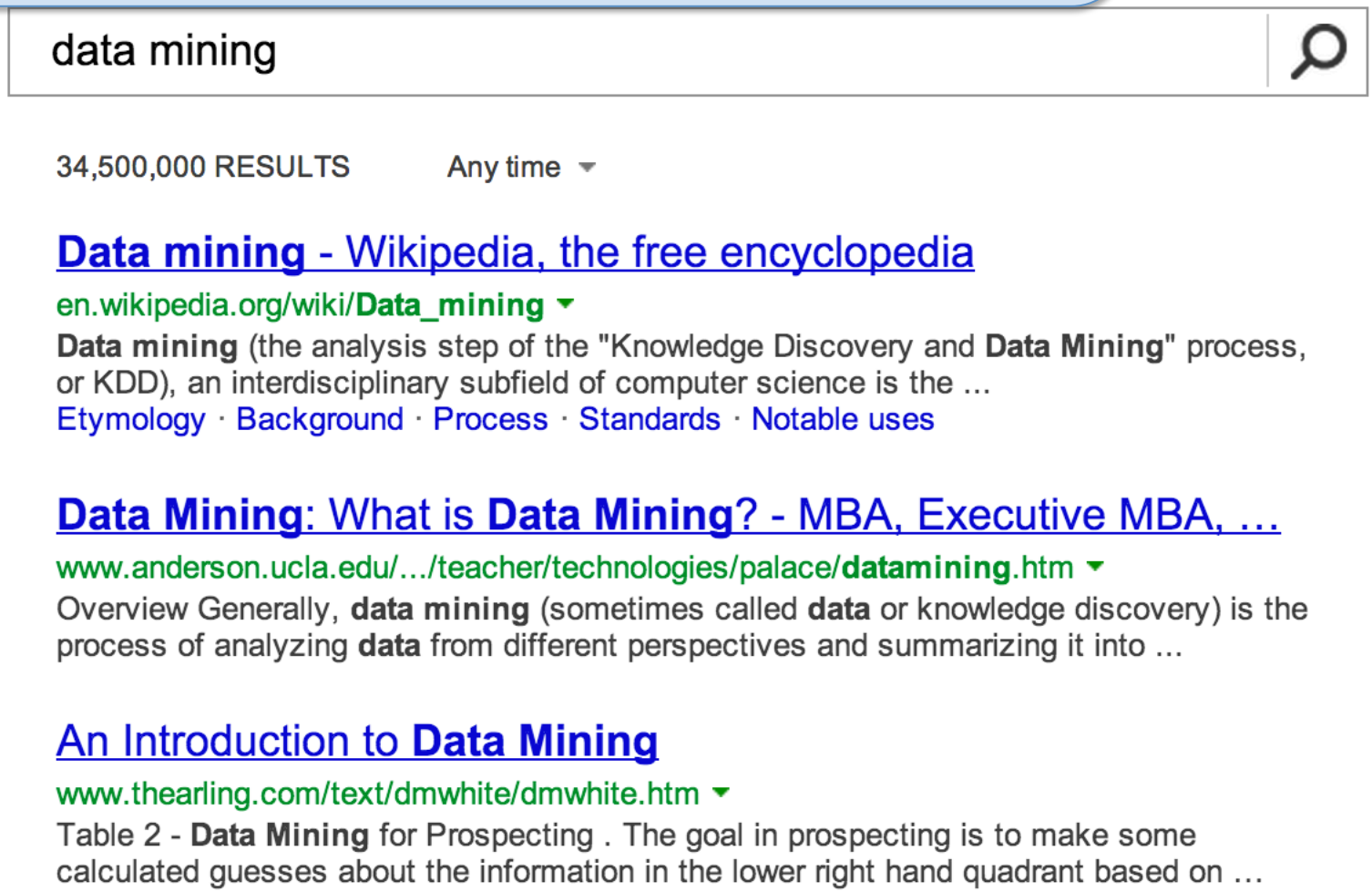
```
$DATE INFO  [hostId=long-frontend-1]
[logger=scala.config.protocol.handler
[auth=User:scott@sumologic.com:000000
[remote_ip=173.228.89.151] [web_sessi
```

```
$DATE INFO  [hostId=long-frontend-1]
[logger=scala.config.protocol.handler
[auth=User:*****:false:DefaultSumoSy
getDashboard(*****) after *** ms
```

$\hat{f}(x)$

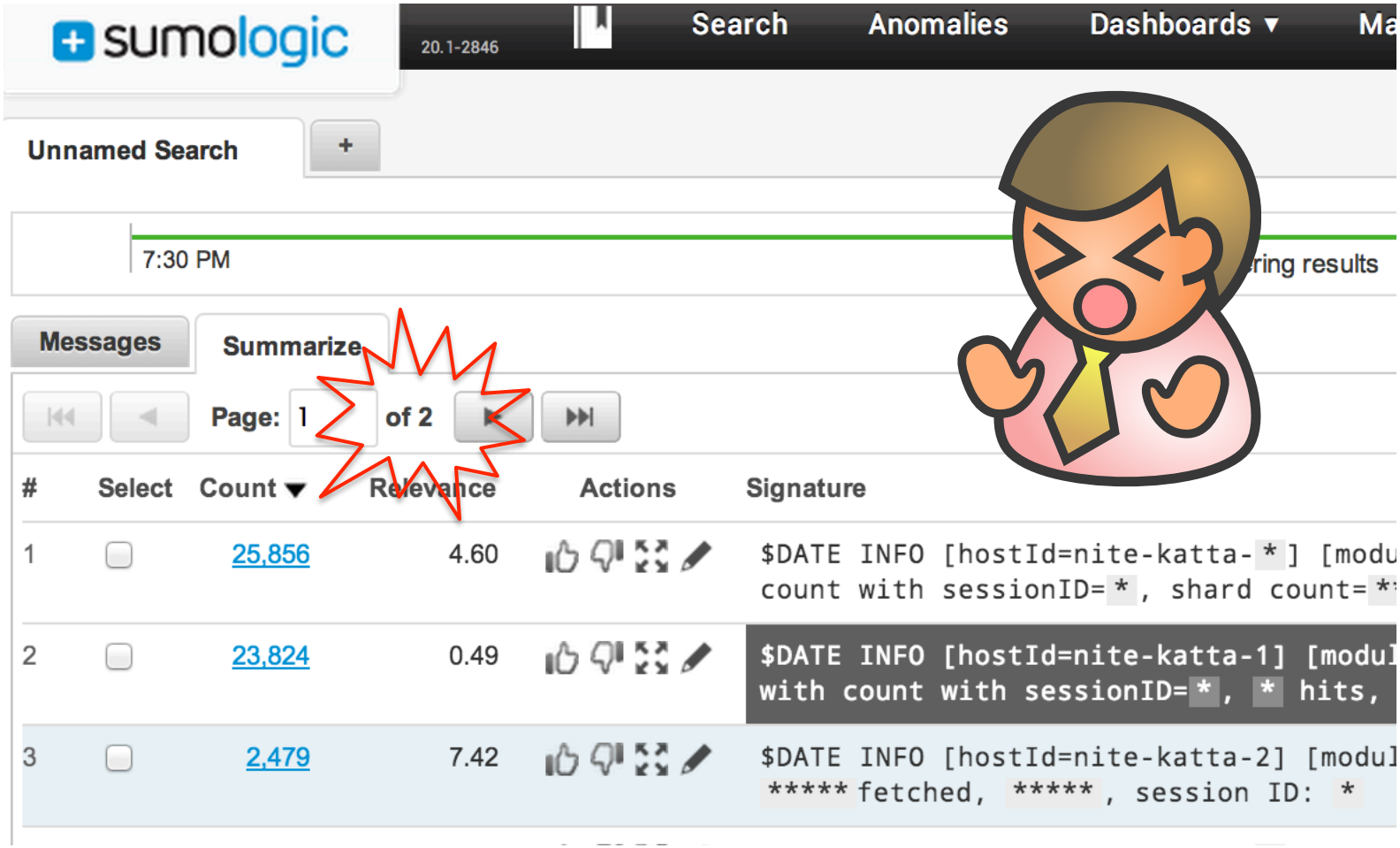
Learning to rank

- **Given:** set of items, historical data
- **Do:** rank by "relevance"



sumologic

Two pages is still too many!

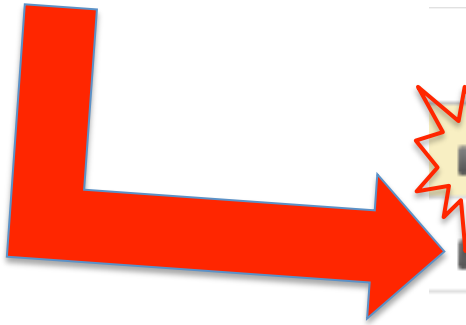


sumologic

Actions	Signature
	[hostId=tamalpais-1] INFO - JOB SUCCESS
	[hostId=tamalpais-1] INFO - ASSIGNING W
	[hostId=mirrorlake-2] INFO - login atte
	[hostId=govnelson-3] INFO - database st
	[hostId=bigsur-*] WARNING - database c
	[hostId=govnelson-2] INFO - database shu

Learning to rank

- **Given:** $\hat{f}(x)$ signatures, user activity
- **Do:** rank by "relevance"



Actions	Signature
	[hostId=govnelson-2] INFO - database shut
	[hostId=govnelson-3] INFO - database star
	[hostId=bigsur-*] WARNING - database con
	[hostId=tamalpais-1] INFO - JOB SUCCESS -
	[hostId=tamalpais-1] INFO - ASSIGNING WOR
	[hostId=mirrorlake-2] INFO - login attemp

True story: troubleshooting @ digital media co.

- Site “acting weird”
- Investigation with LogReduce
  - error logs → issue with content push/publish workflow
    - root cause
  - 50 minutes later: “object missing” errors serving content
    - user-visible outage
- Benefits
  - rapidly “skim” the logs
  - create an **alert**

More true stories

- Domains
  - financial services
  - SaaS vendors
- Use cases
  - Availability / performance
    - Mean time to investigation (MTTI) = “hours to minutes”
  - Security
    - Quickly bubble up unusual logs

General lessons

General lessons

- Combat “data disorientation”

General lessons

- Combat “data disorientation”
- Surface **latent structure**

General lessons

- Combat “data disorientation”
- Surface **latent structure**
- Link to **underlying raw data**

General lessons

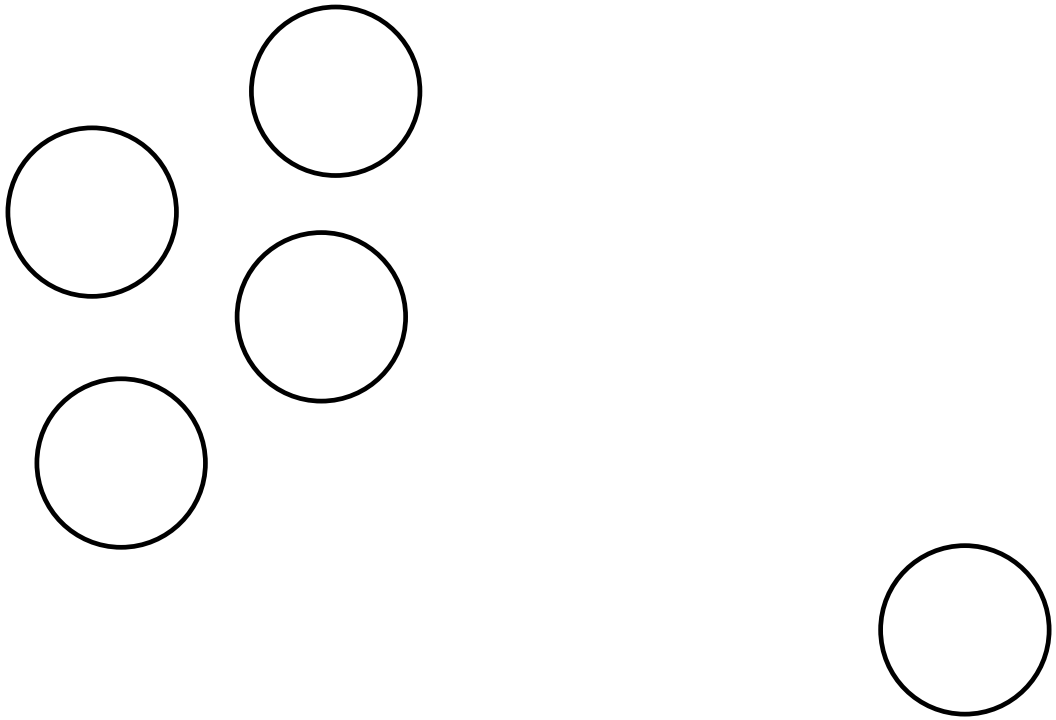
- Combat “data disorientation”
- Surface **latent structure**
- Link to **underlying raw data**
- **Empower user** to improve results

unknown unknowns

$\hat{f}(x)$

Outlier detection

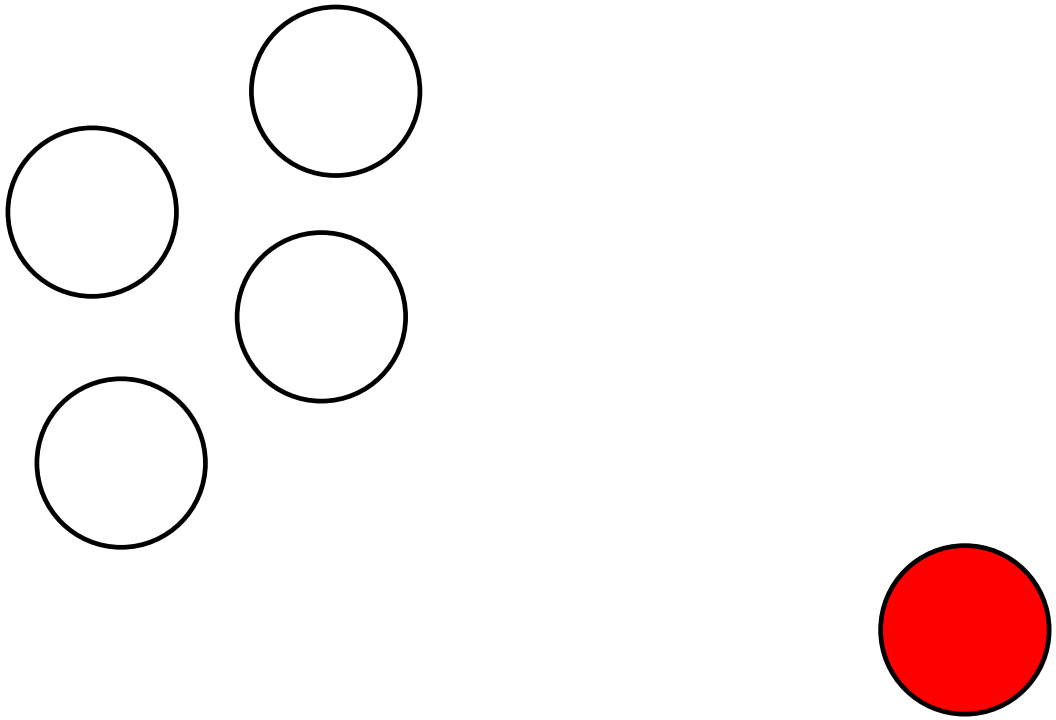
- **Given:** data points
- **Do:** identify outliers



$\hat{f}(x)$

Outlier detection

- **Given:** data points
- **Do:** identify outliers



$\hat{f}(x)$

Anomaly detection

- **Given:** log data
- **Do:** flag anomalies

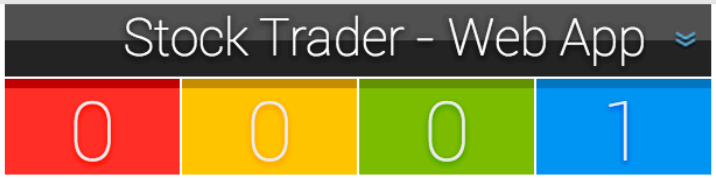
$$\begin{matrix} \texttt{Health check OK} \\ \texttt{Request processed} \\ \texttt{Txn timeout, retry} \end{matrix} \begin{bmatrix} 33 \\ 29 \\ 3 \end{bmatrix}, \begin{bmatrix} 30 \\ 26 \\ 6 \end{bmatrix}, \begin{bmatrix} 31 \\ 27 \\ 732 \end{bmatrix}$$

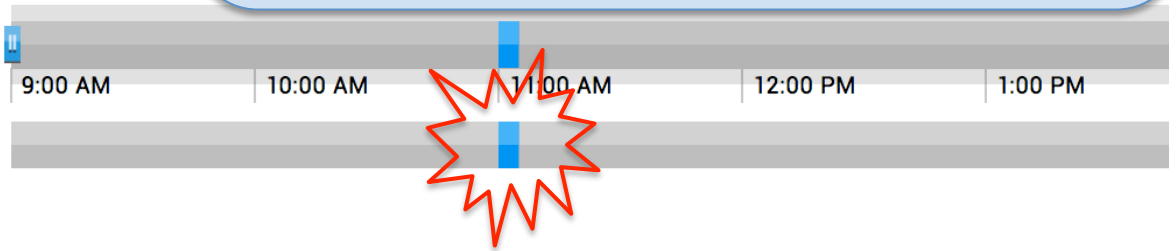
$t_1 \qquad t_2 \qquad t_3$

$\hat{f}(x)$

Anomaly detection

- **Given:** log data
- **Do:** flag anomalies

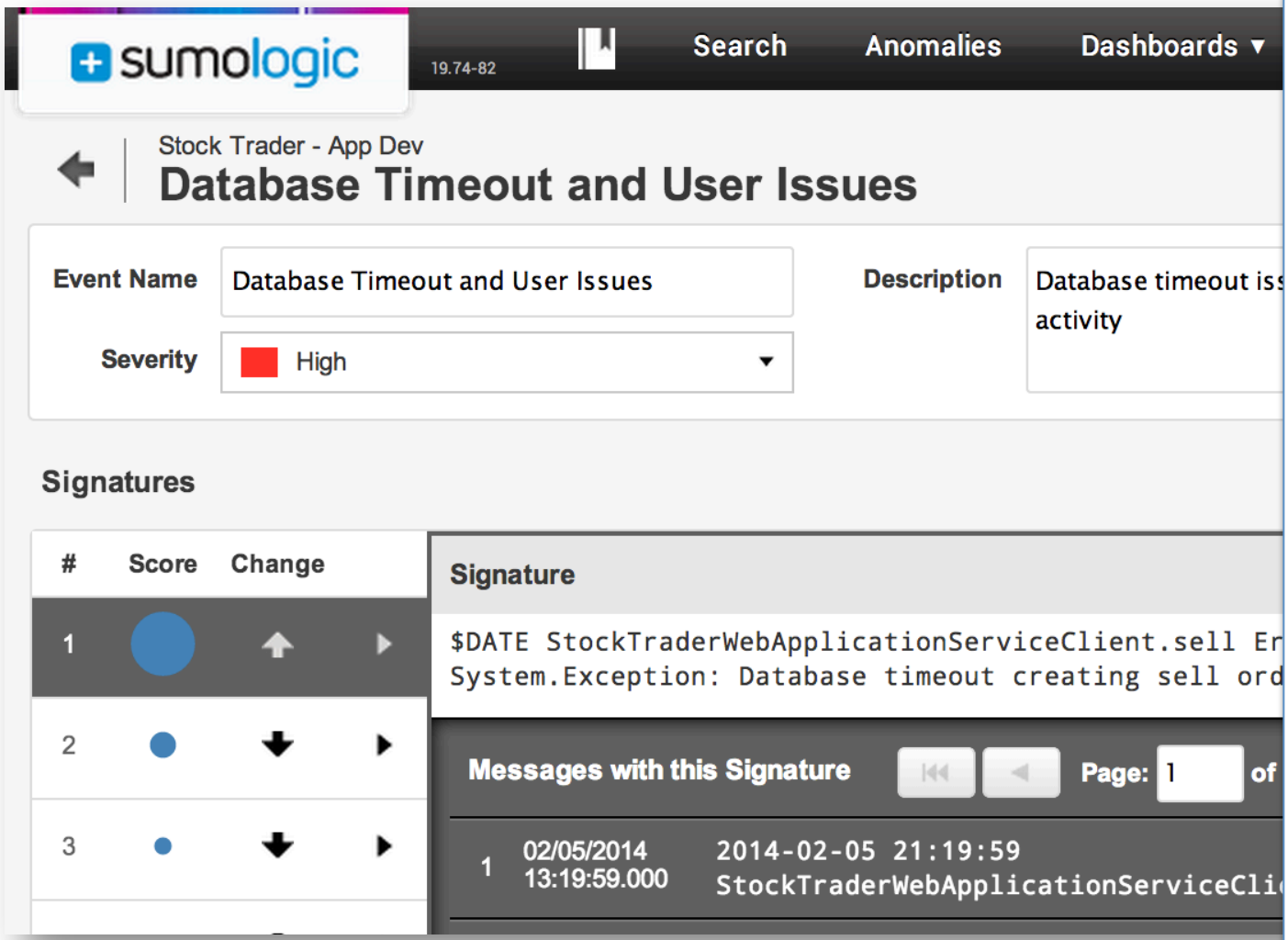


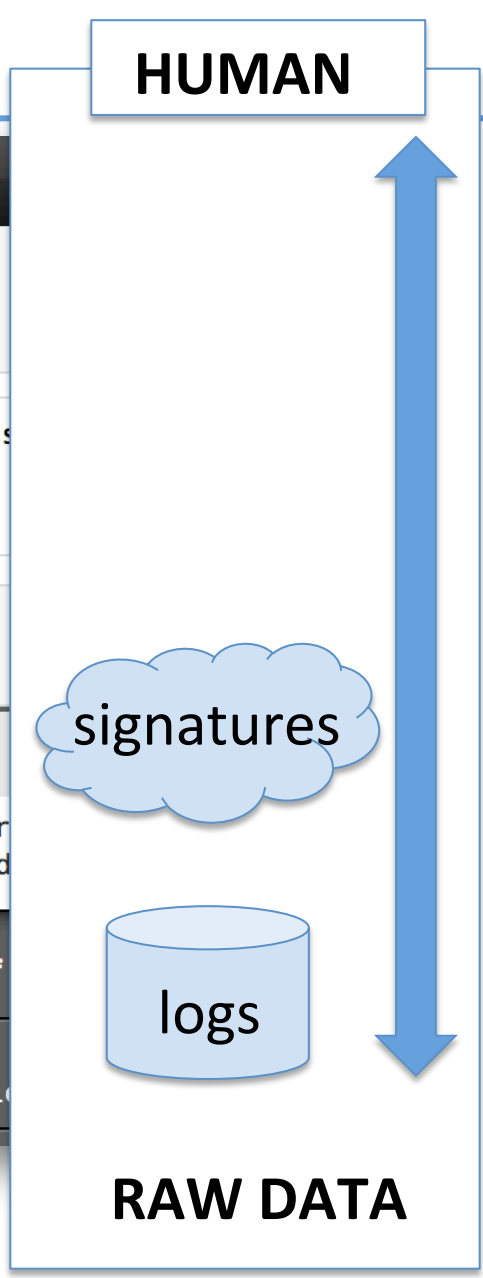


$$\begin{matrix} \texttt{Health check OK} \\ \texttt{Request processed} \\ \texttt{Txn timeout, retry} \end{matrix} \begin{bmatrix} 33 \\ 29 \\ 3 \end{bmatrix}, \begin{bmatrix} 30 \\ 26 \\ 6 \end{bmatrix}, \begin{bmatrix} 31 \\ 27 \\ 732 \end{bmatrix}$$

$t_1 \qquad t_2 \qquad t_3$

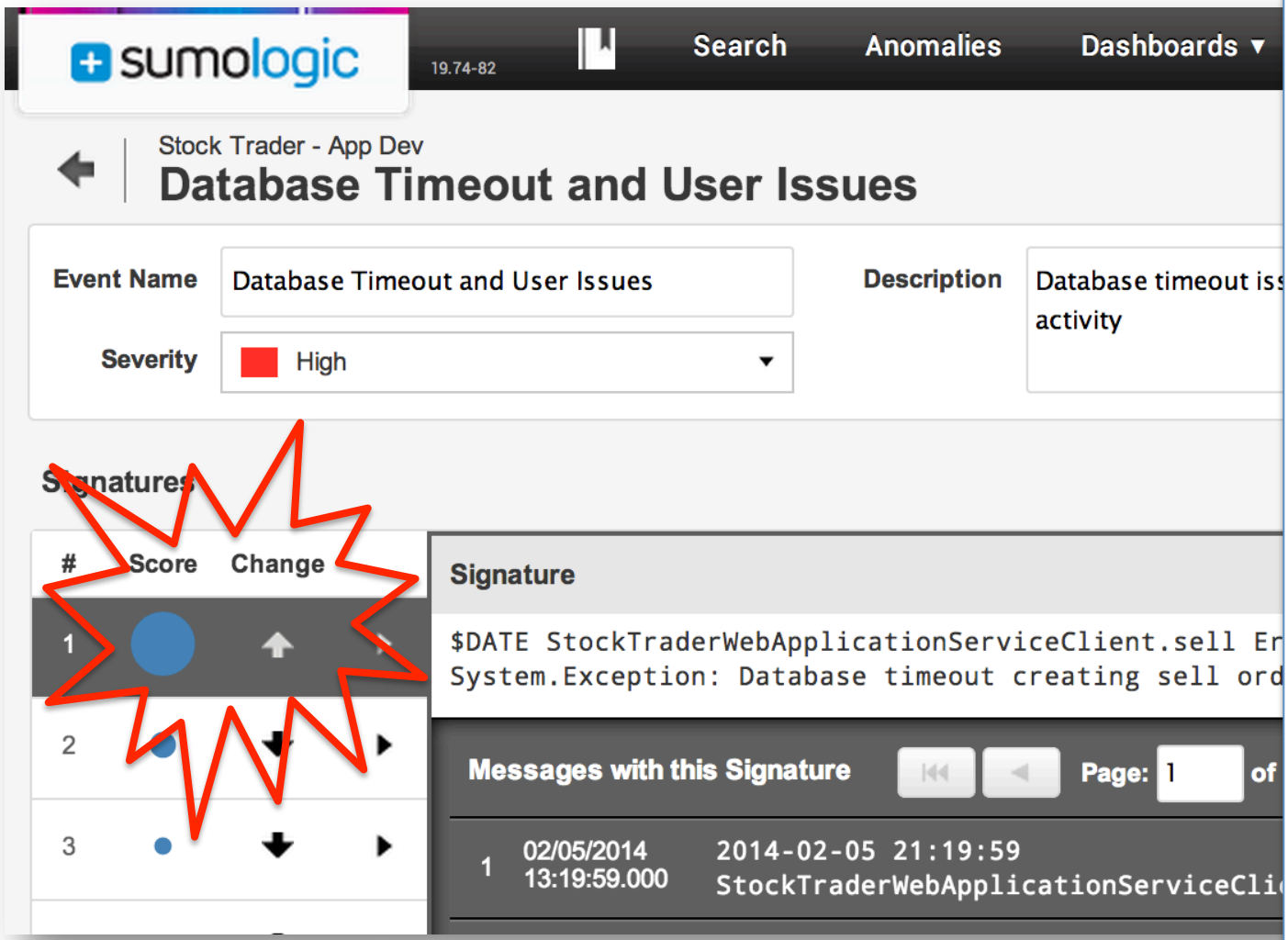
Investigate and annotate events

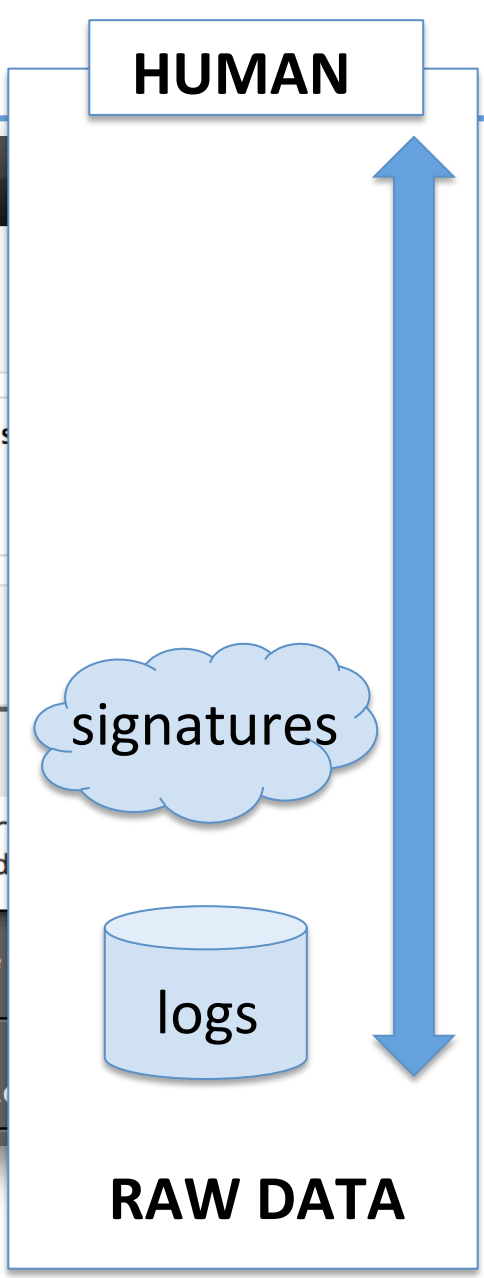




sumologic

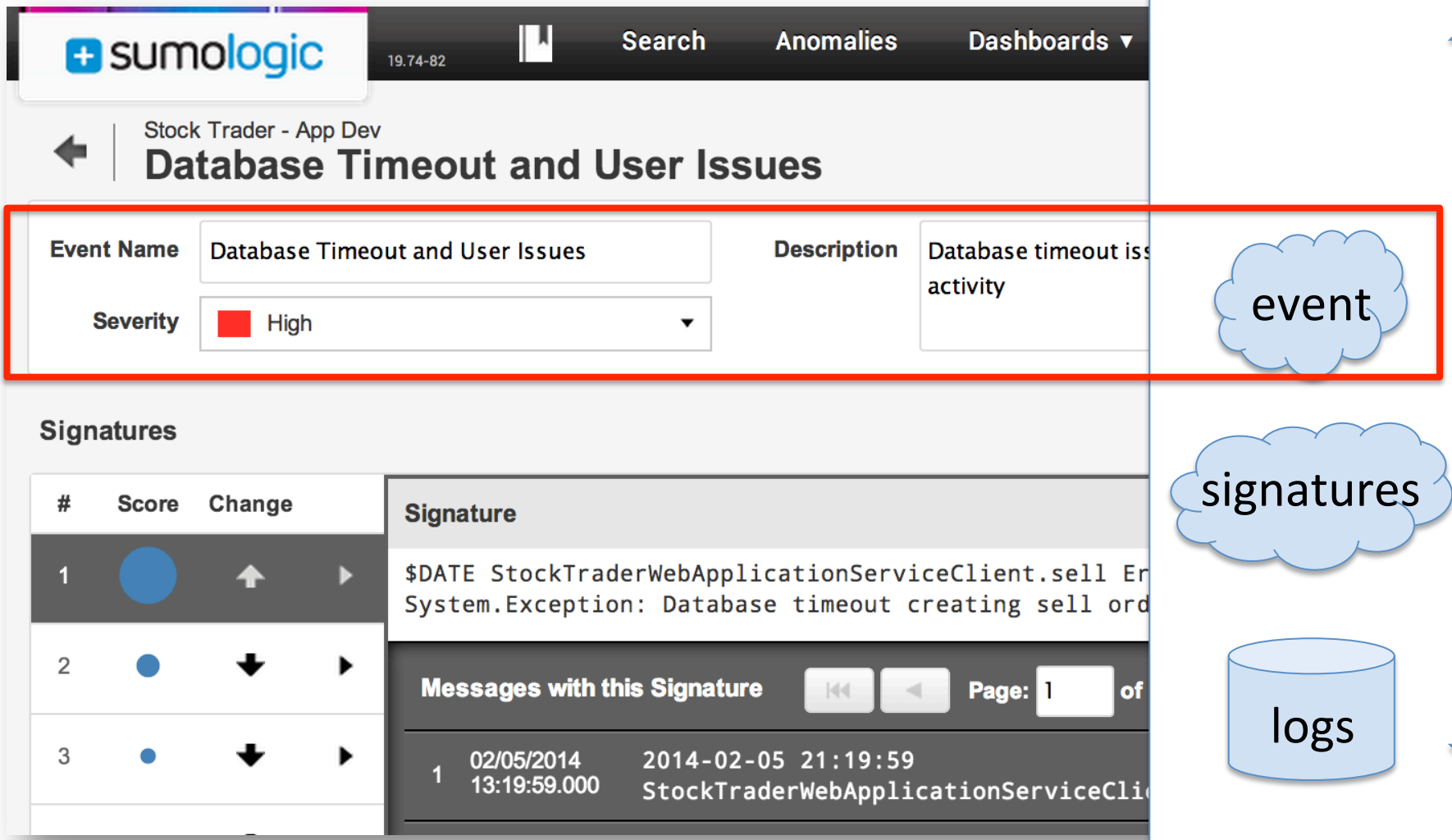
Investigate and annotate events





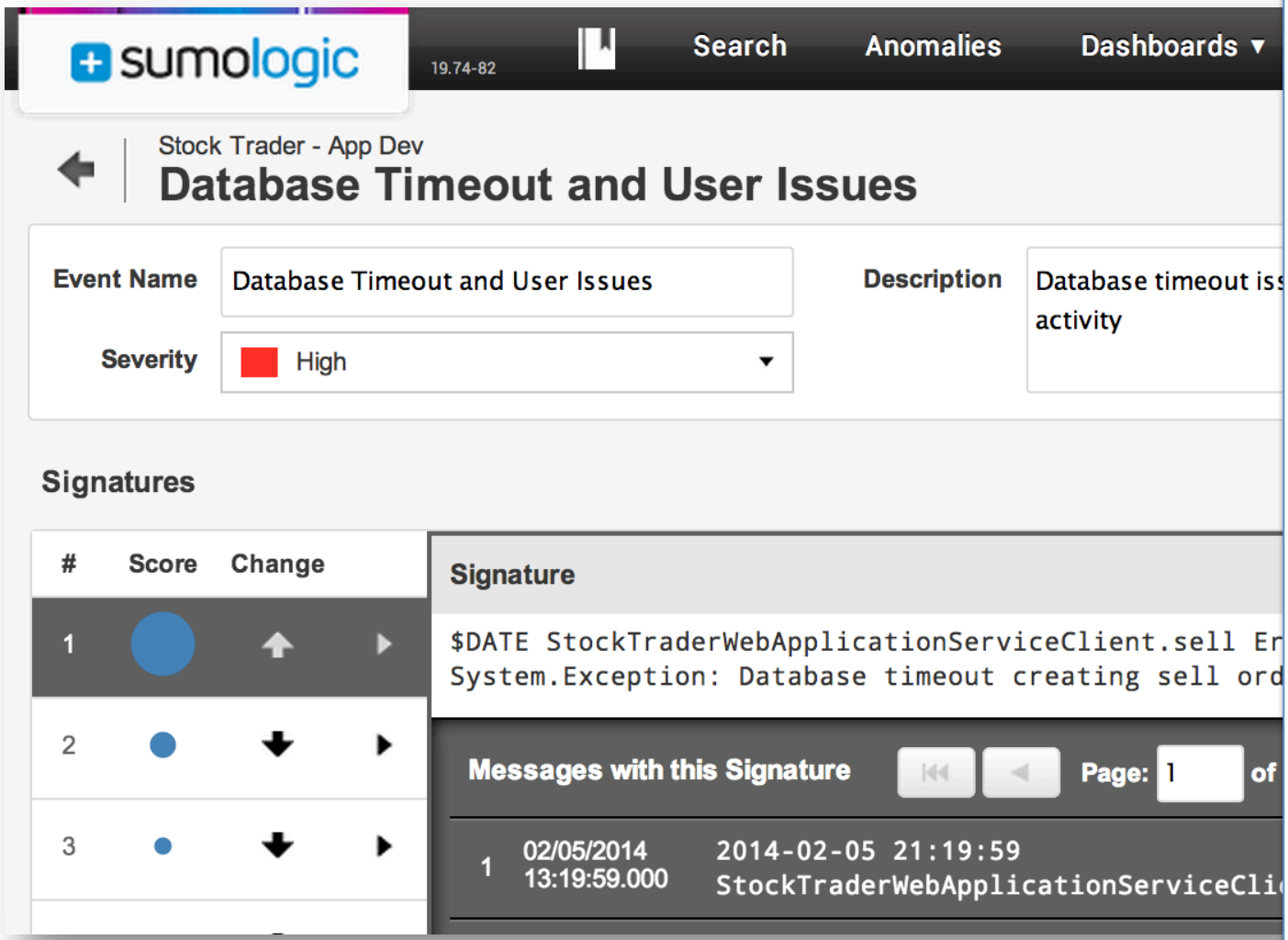
sumologic

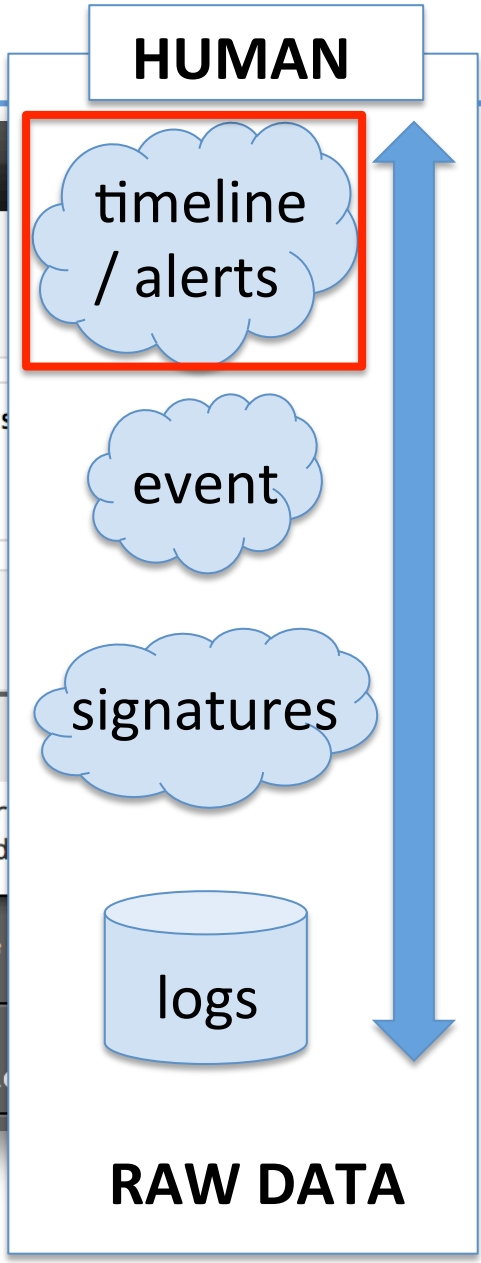
Investigate and annotate events



sumologic

Investigate and annotate events



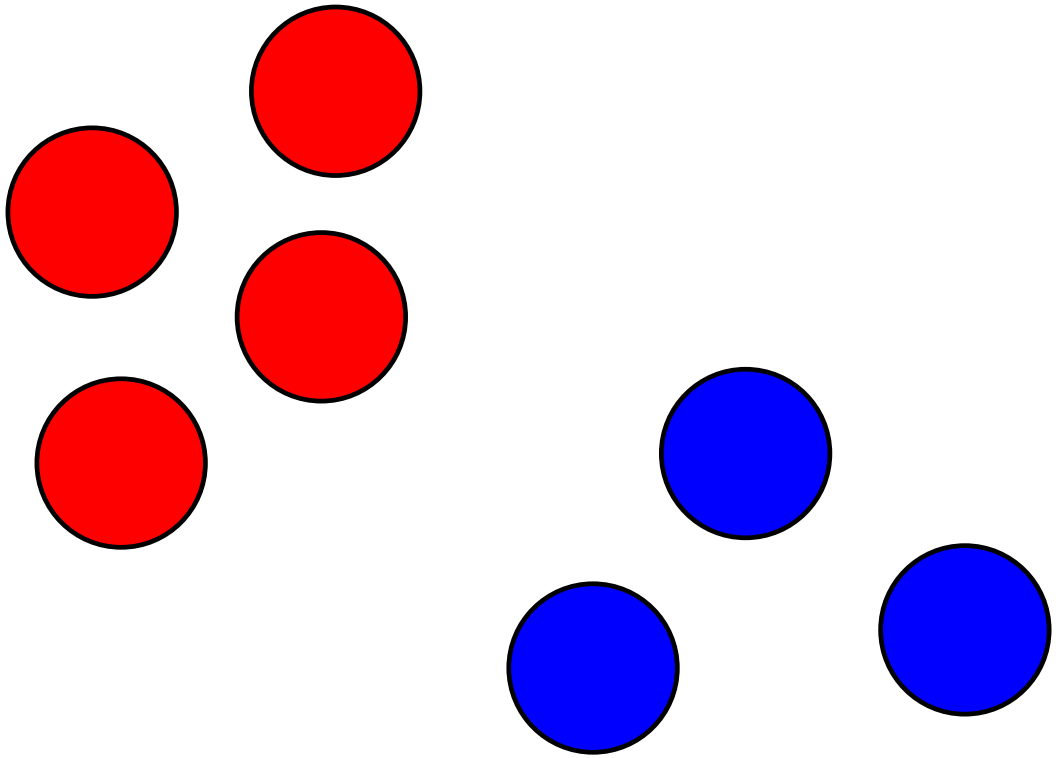


sumologic

$\hat{f}(x)$

Supervised classification

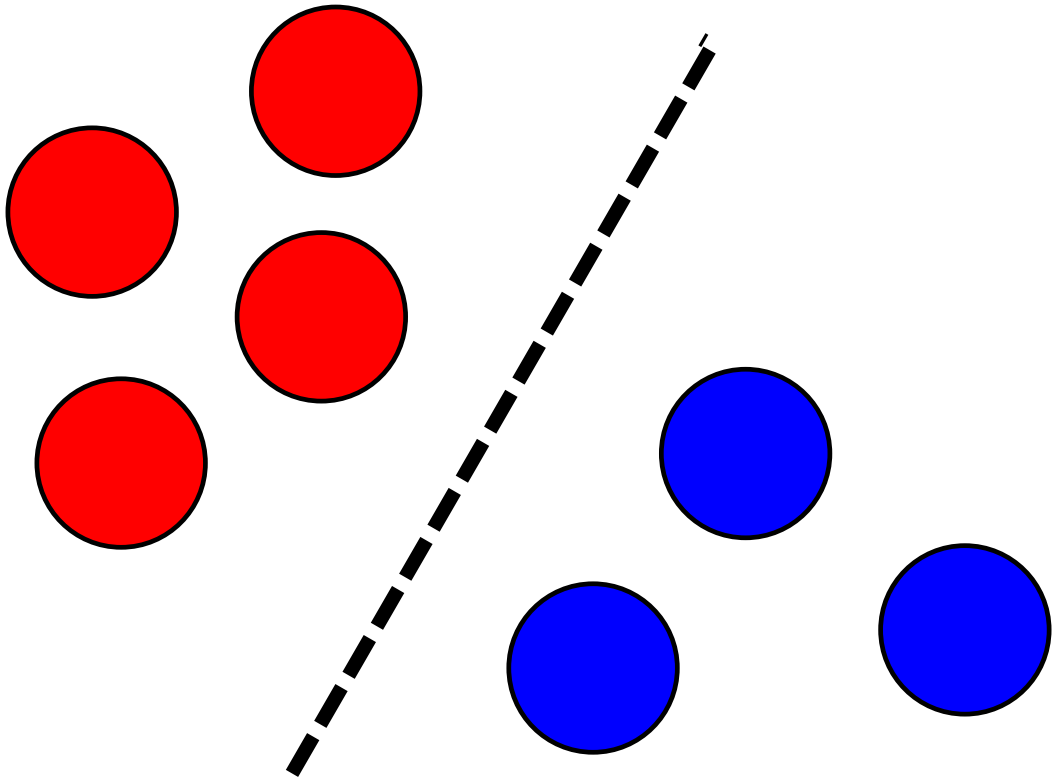
- **Given:** labeled data points
- **Do:** predict future labels



$\hat{f}(x)$

Supervised classification

- **Given:** labeled data points
- **Do:** predict future labels



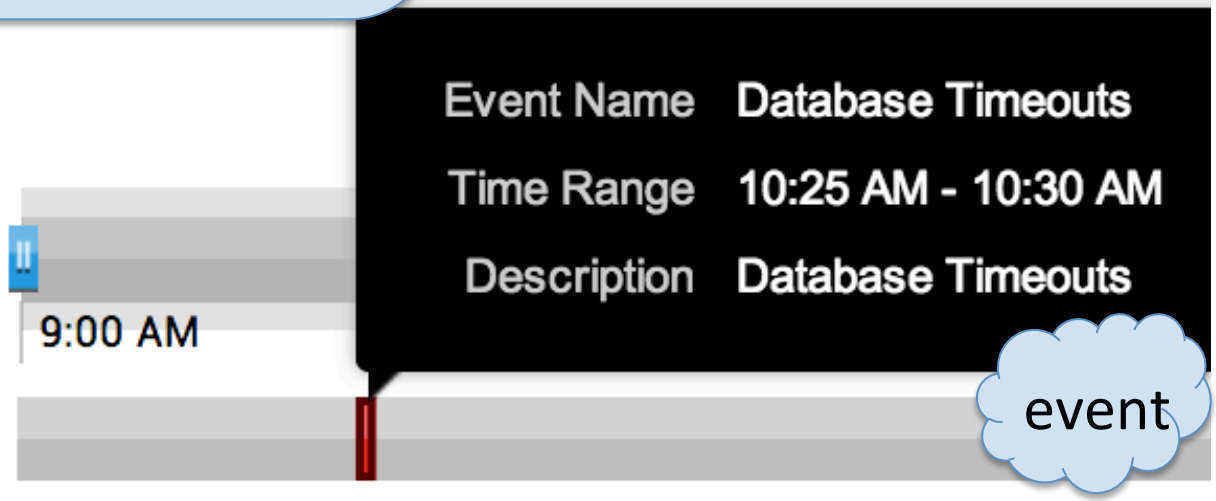
Supervised classification

$$\hat{f}(x)$$

- **Given:** log data, annotated events
- **Do:** classify new occurrences

timeline / alerts

Database Timeouts



sumologic

True stories

- SSH problems
  - configuration errors
  - script user auth failures
- Potential security events
  - surge of failed logins
- Unhappy infrastructure
  - Oracle
  - VMware

More general lessons

- **Explain** algorithm “decisions”
  - **why** was this flagged as anomaly?

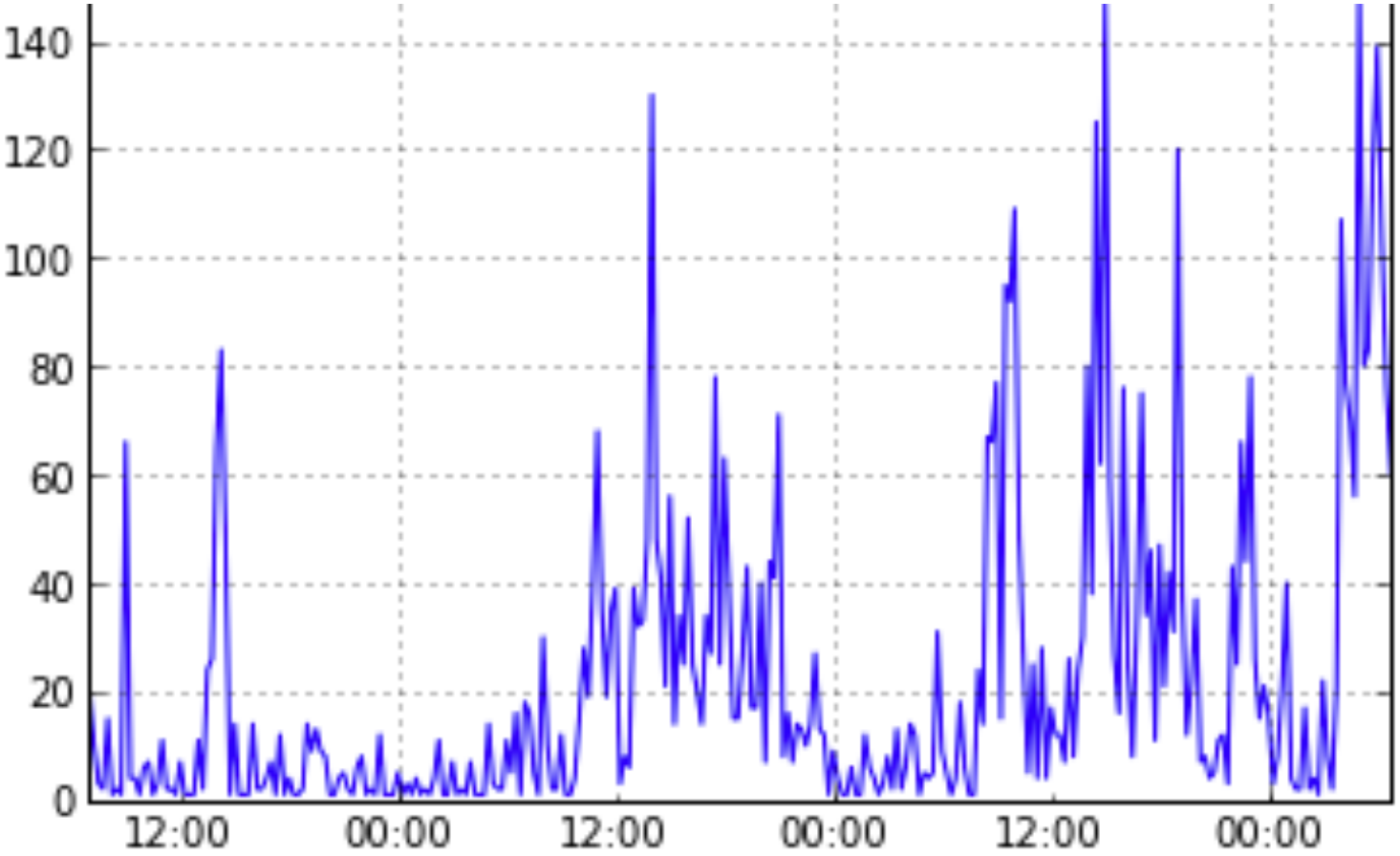
More general lessons

- **Explain** algorithm “decisions”
  - **why** was this flagged as anomaly?
- Link to **underlying raw data**
  - **(AGAIN)**

More general lessons

- **Explain** algorithm “decisions”
  - **why** was this flagged as anomaly?
- Link to **underlying raw data**
  - **(AGAIN)**
- **Empower user** to improve results
  - **(AGAIN)**

Numerical time-series data

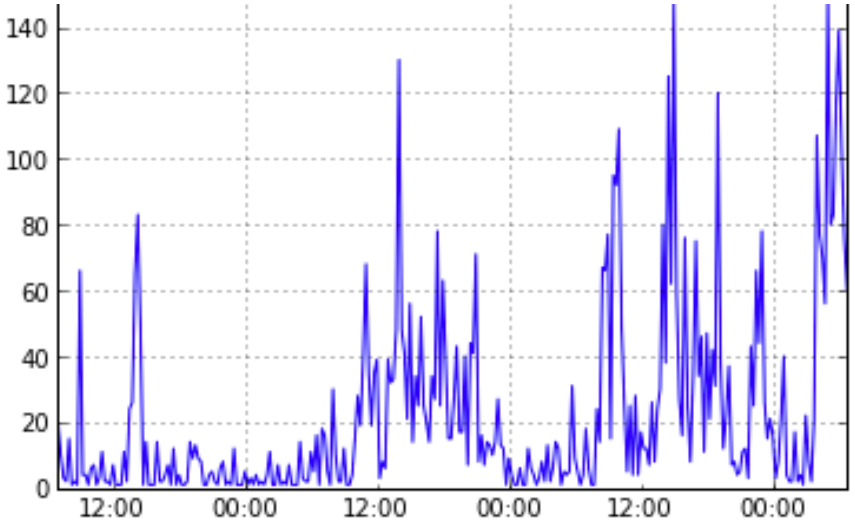


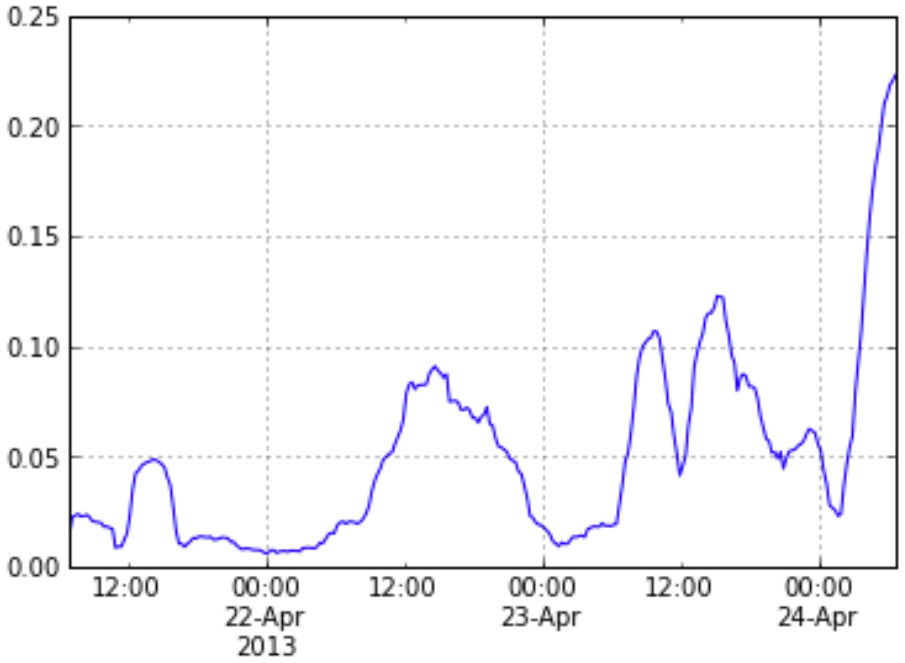
sumologic

Signal decomposition

$\hat{f}(x)$

- **Given:** time-series
- **Do:** extract model components

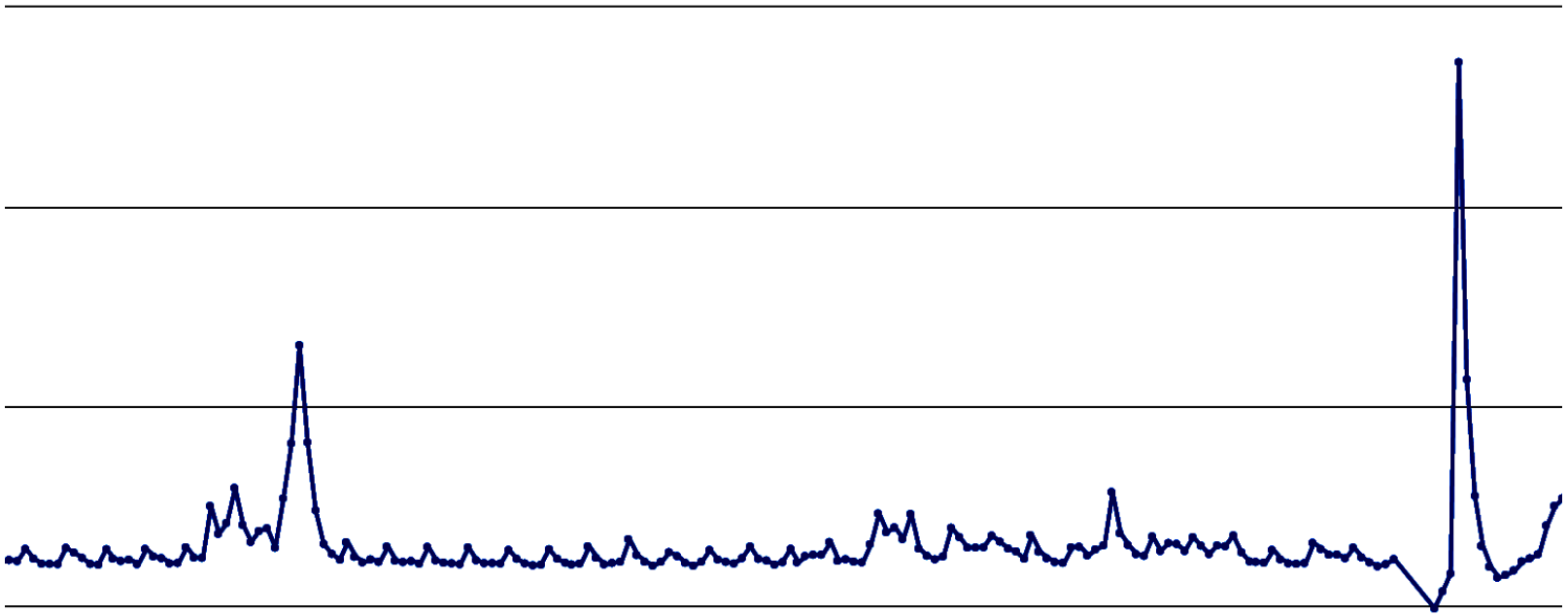




$\hat{f}(x)$

Outlier detection

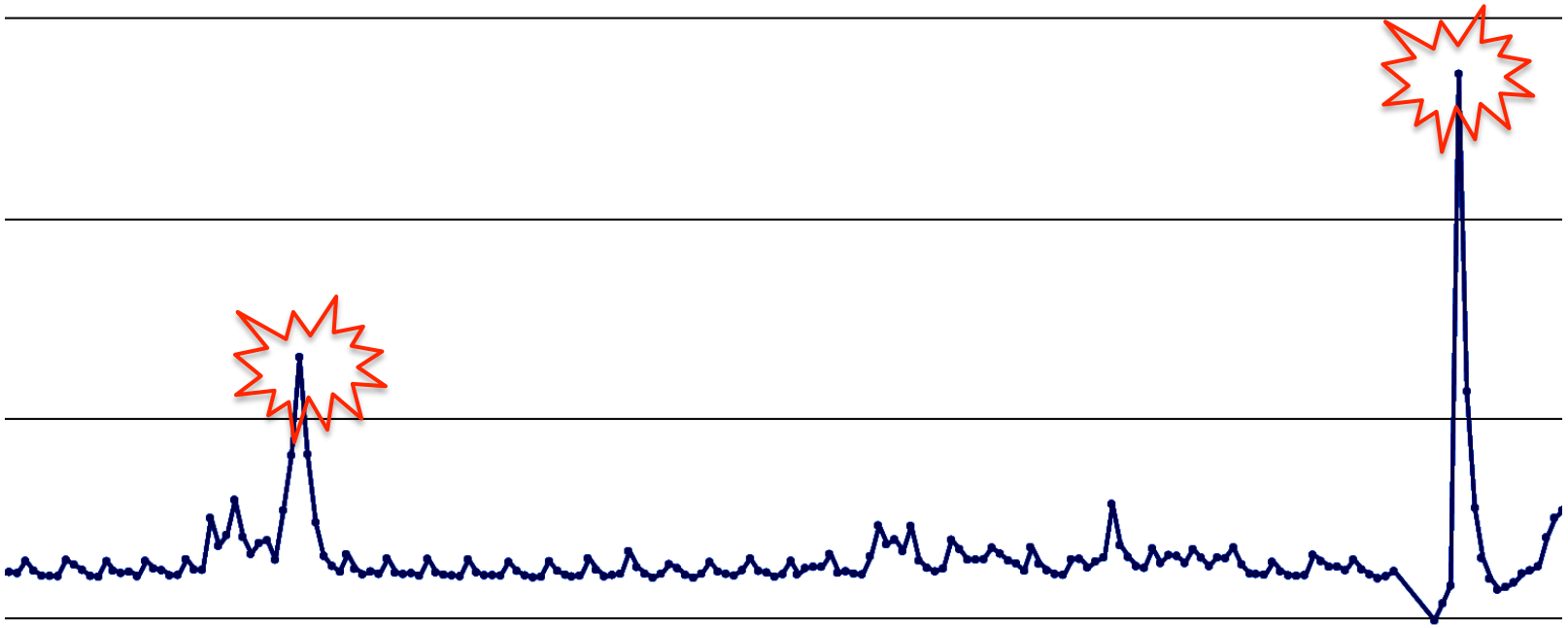
- **Given:** data points
- **Do:** identify outliers



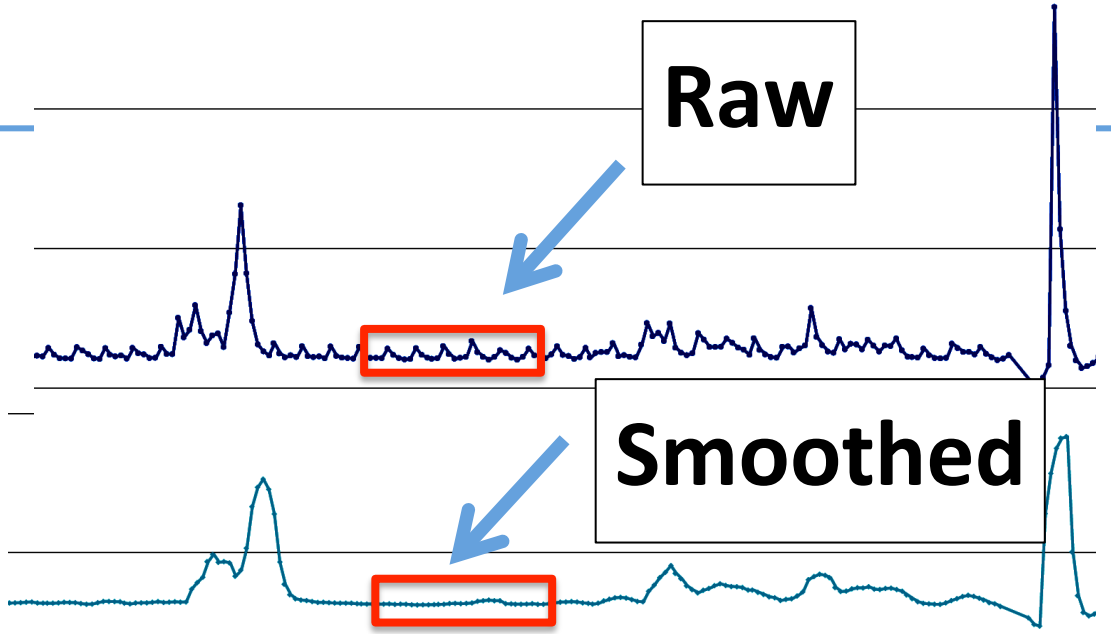
$\hat{f}(x)$

Outlier detection

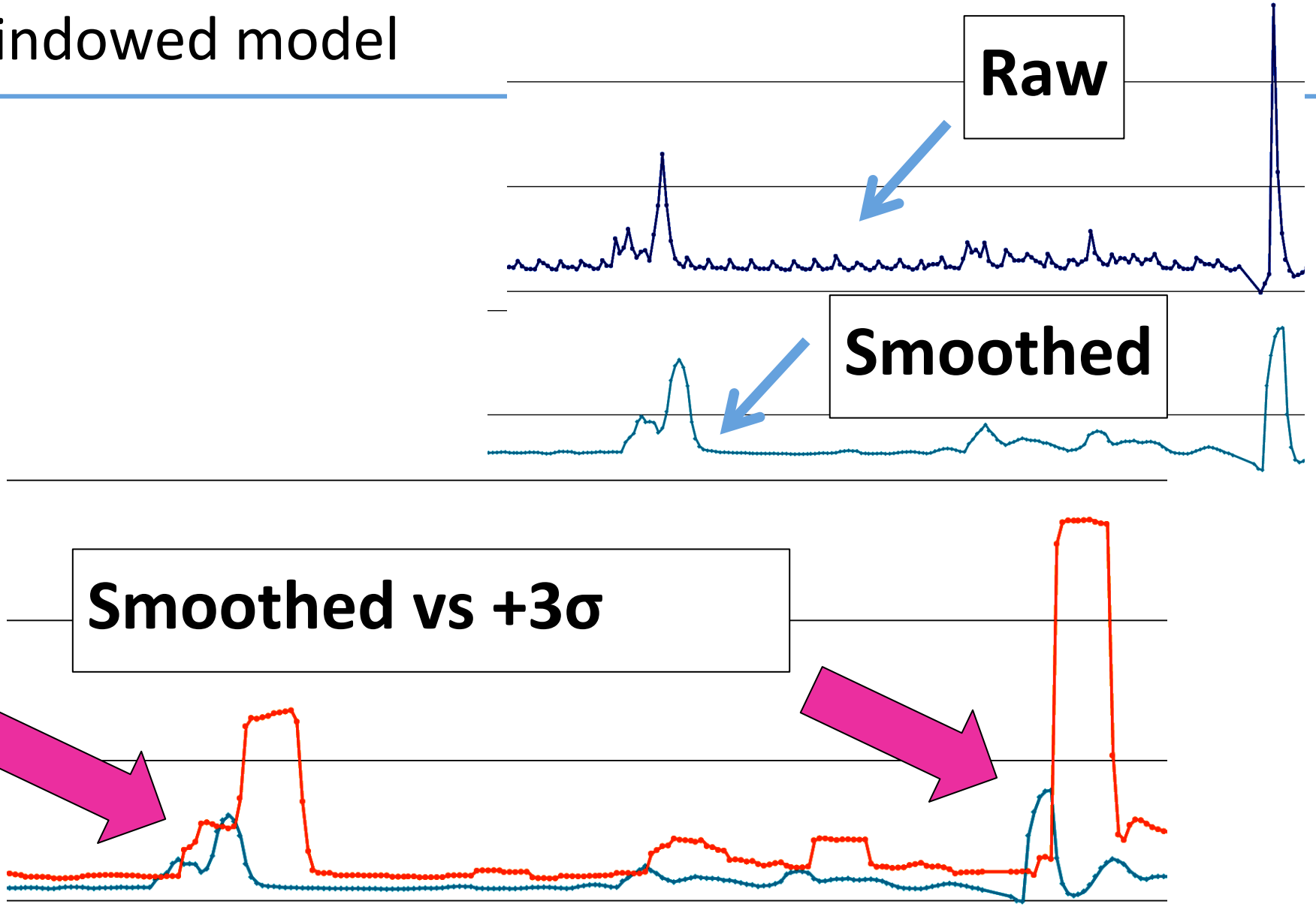
- **Given:** data points
- **Do:** identify outliers



Windowed model



Windowed model



sumologic

True stories

- Financial services: bad data points
- Security: misbehavior
- Operations: alerting

Yet more general lessons

- Time-series data analysis well-studied
- Read the literature(s)!

< OBLIGATORY PLUGS >

freesumo.com

Get Sumo Logic Free

Get a fully functioning version of our enterprise cloud-based log management and analytics service FREE

Sumo Logic Free delivers real-time troubleshooting, proactive application management and powerful IT and business insights. Our free version allows for up to three users and 500 MB per day with seven days of data retention.

SUMO LOGIC FREE

First name *

Last name *

Email *

Login credentials will be sent to this address

Company *

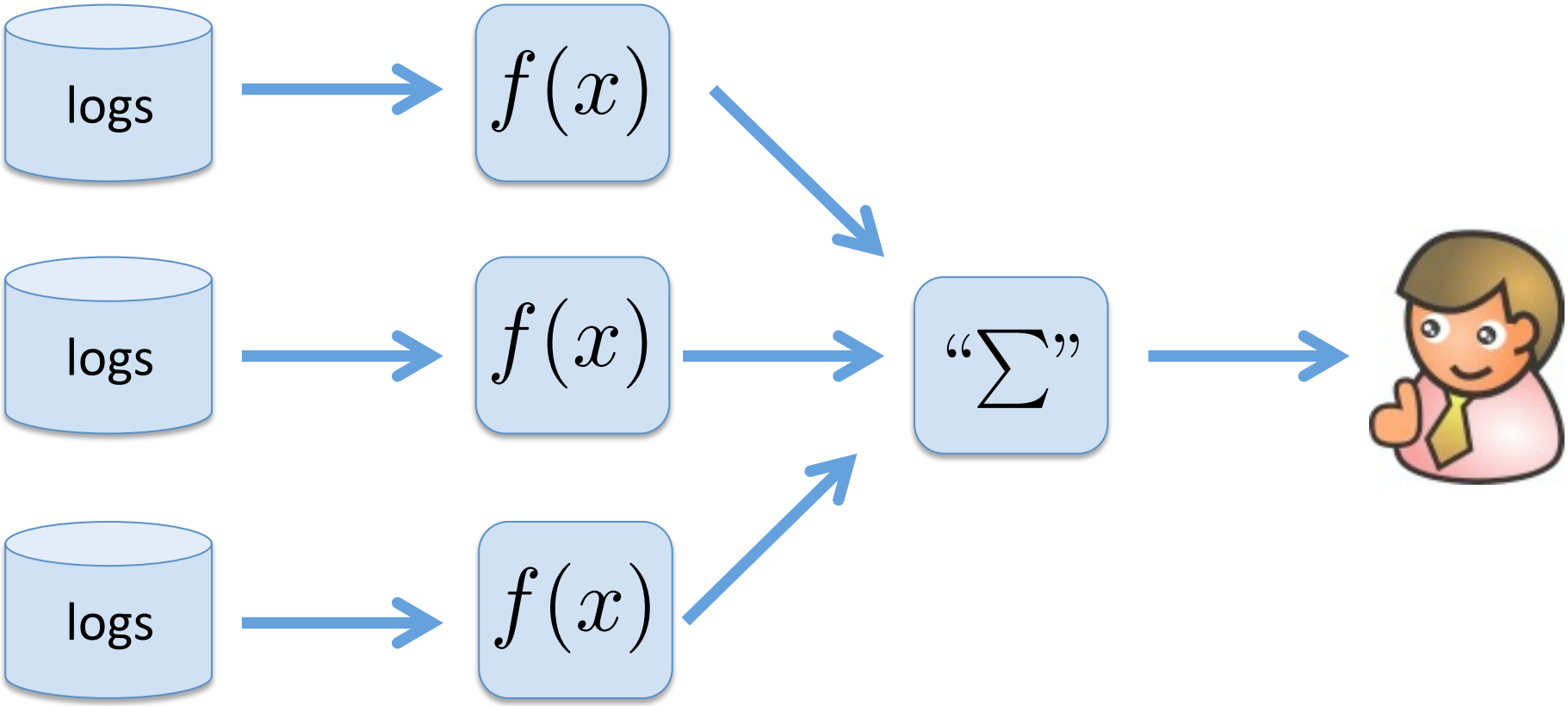
Phone *

Sign Up Now

By clicking on the "Sign Up Now" button, you agree to accept our Terms of Service

sumologic

BONUS: scale-out, streaming architecture



sumologic

BONUS: approximating with Count-Min Sketch

$$\hat{c} - c \geq 0$$

BONUS: approximating with Count-Min Sketch

$$\hat{c} - c \geq 0$$
$$\hat{c} - c \leq \epsilon N$$
$$\text{w.p. } \geq 1 - \delta$$

Approximate counting with Count-Min Sketch

$$\hat{c} - c \geq 0$$

$$\hat{c} - c \leq \epsilon N$$

$$\text{w.p.} \geq 1 - \delta$$