



O'REILLY®

Strata  
MAKING DATA WORK

# 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*

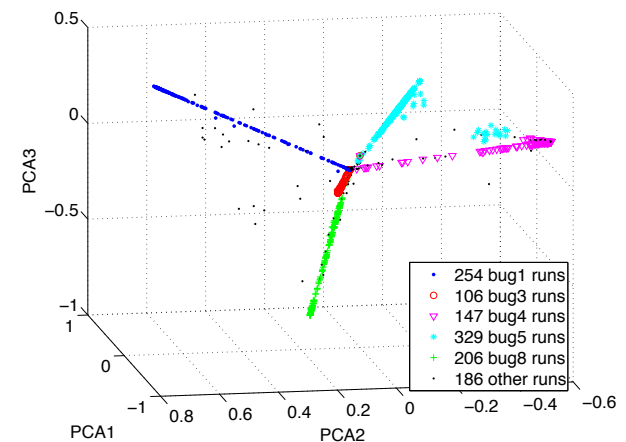
$$\frac{1}{n} \mathbb{E}_n(E) \underset{n \rightarrow \infty}{=} 1 + \delta_1(n) + o(1), \text{ where } |\delta_1(n)| < 5 \cdot 10^{-5} \text{ as soon as } m \geq 16.$$

# Context: me

- Data sciences @ Sumo Logic
- Co-organizer @ SF ML Meetup
- Previous
  - Post-doc in knowledge discovery
  
- Even more previous machine data research projects
  - University of Wisconsin--Madison
  - Microsoft Research



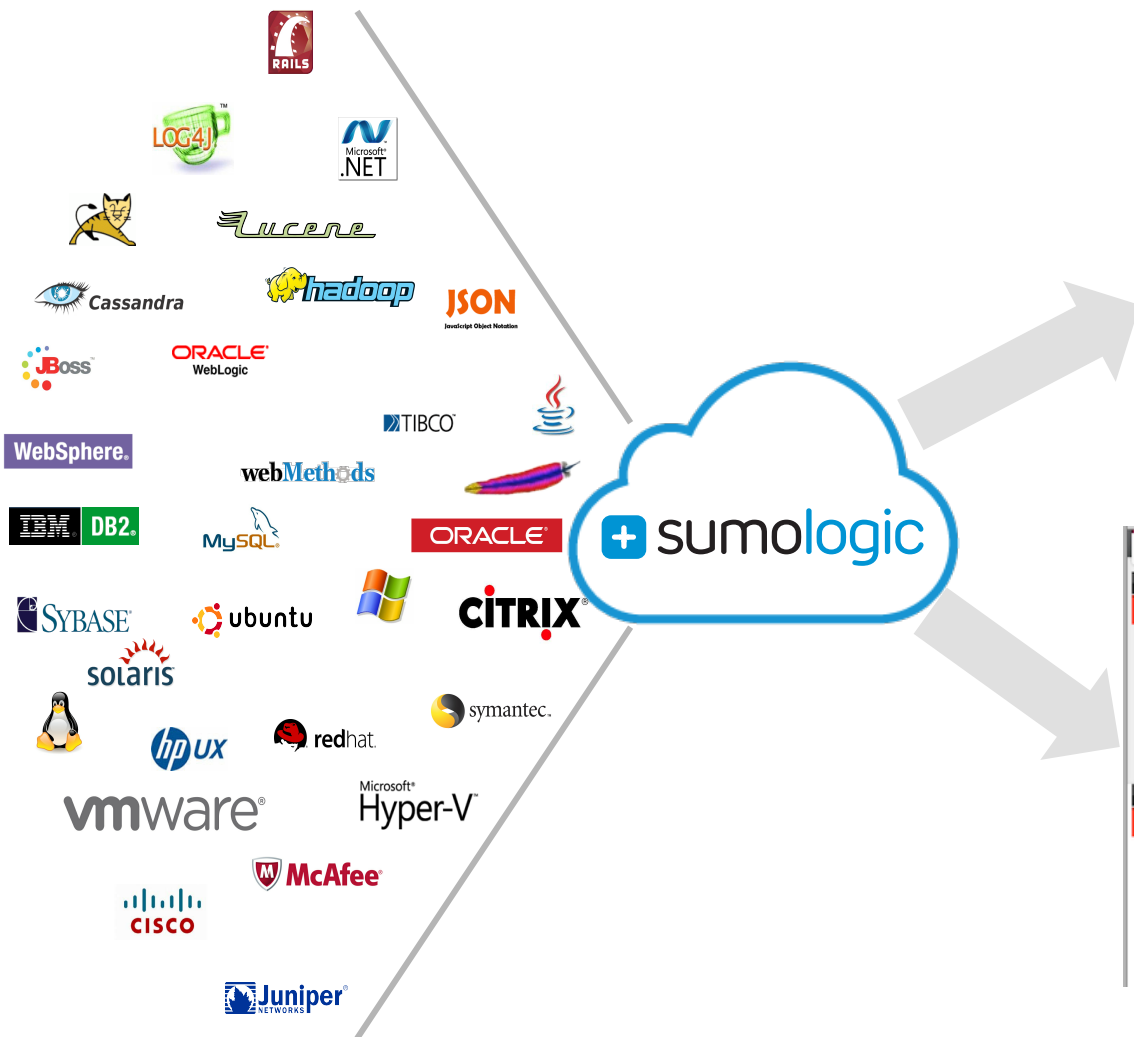
$$\hat{f}(x)$$



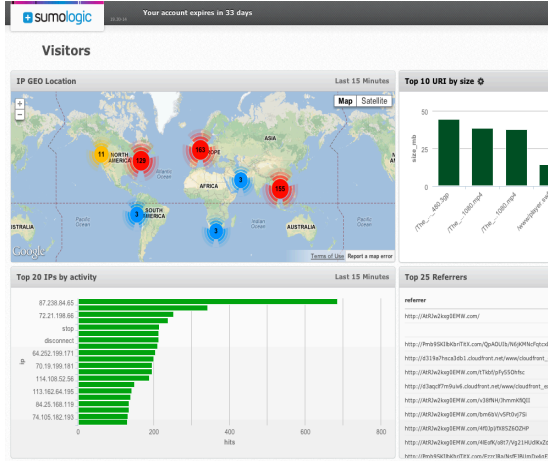


# Context: Sumo Logic

“Turning Machine Data Into IT and Business Insights”



*Search, monitor, visualize*

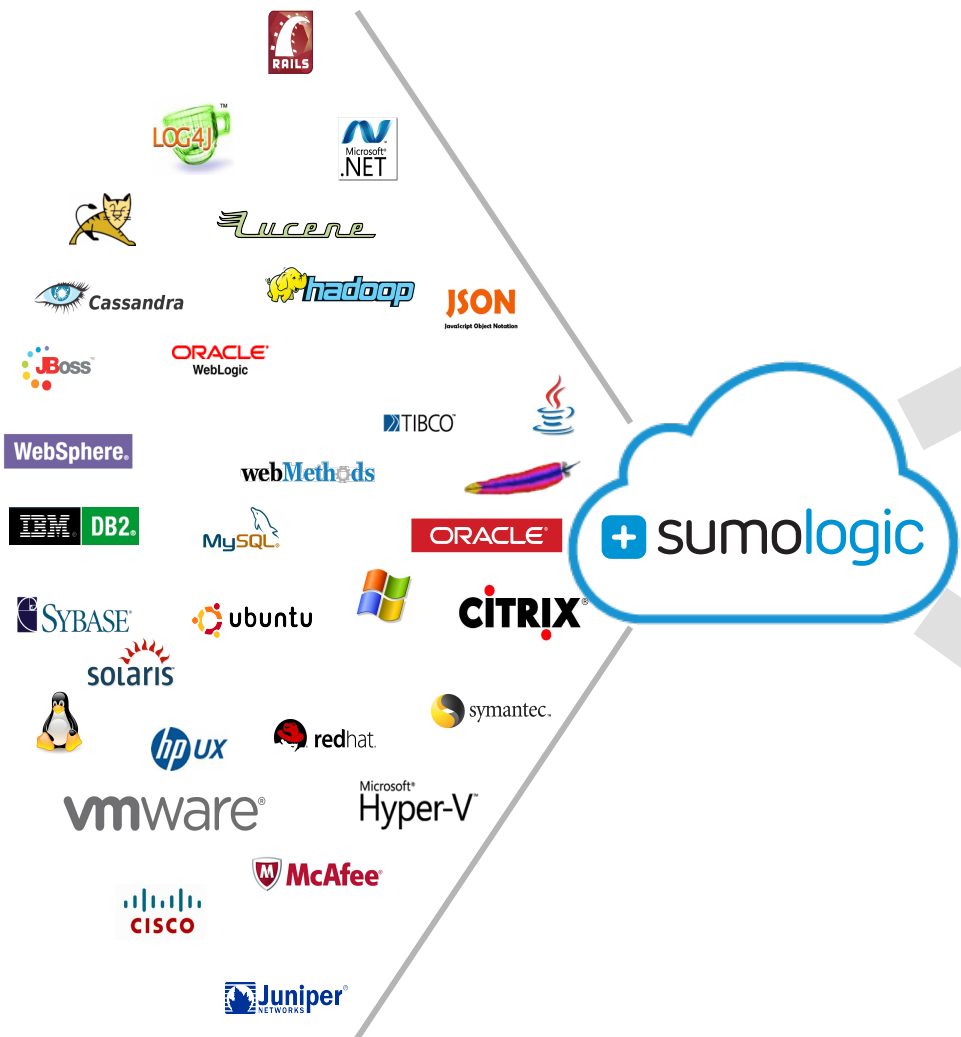


*Learn, classify, predict*

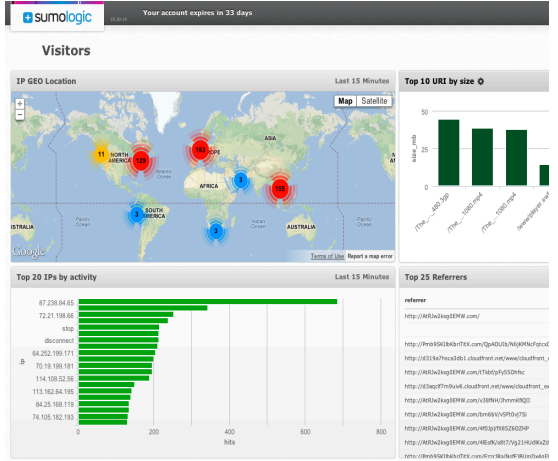


# Context: Sumo Logic

“Turning Machine Data Into IT and Business Insights”



*Search, monitor, visualize*



*Learn, classify, predict*

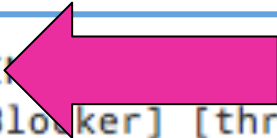


# Anatomy of a log message: Five W's

```
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-8:000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] Pile for customer:
'0000000000000005', ID: '800000006407637B', block: '80000000004C9A11', msg
count: '1', size: '264', collector: '000000000000483D'
```

# Anatomy of a log message: Five W's

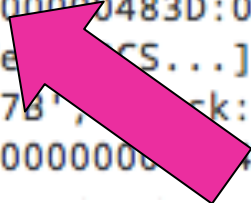
```
2012-05-22 18:47:26,807 -0700 I [ip=184.73.74.54] [host=long-frontend-1] [module=RECEIVER]
[logger=scala.receiver.MessageBlocker] [thread=MTP-MessagePipeline-3]
[auth=Collector:prod-cass-raw-8:000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] Pile for customer:
'0000000000000005', ID: '800000006407637B', block: '80000000004C9A11', msg
count: '1', size: '264', collector: '000000000000483D'
```



- **When?** Timestamp with time zone

# Anatomy of a log message: Five W's

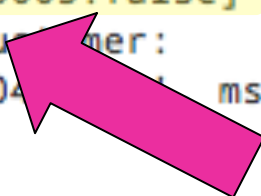
```
2012-05-22 18:47:26,807 -0700 INFO [hostId=long-frontend-1] [module=RECEIVER]
[logger=scala.receiver.MessageBlocker] [thread=MTP-MessagePilePipeline-3]
[auth=Collector:prod-cass-raw-8:0000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=Me...CS...] File for customer:
'0000000000000005',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



- **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-MessagePipeline-3]
[auth=Collector:prod-cass-raw-8:000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] Pile for customer:
'0000000000000005', ID: '800000006407637B', block: '8000000000000000' msg
count: '1', size: '264', collector: '000000000000483D'
```



- **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-MessagePipeline-3]
[auth=Collector:prod-cass-raw-8:000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] File for customer:
'0000000000000005', ID: '800000006407637B', 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





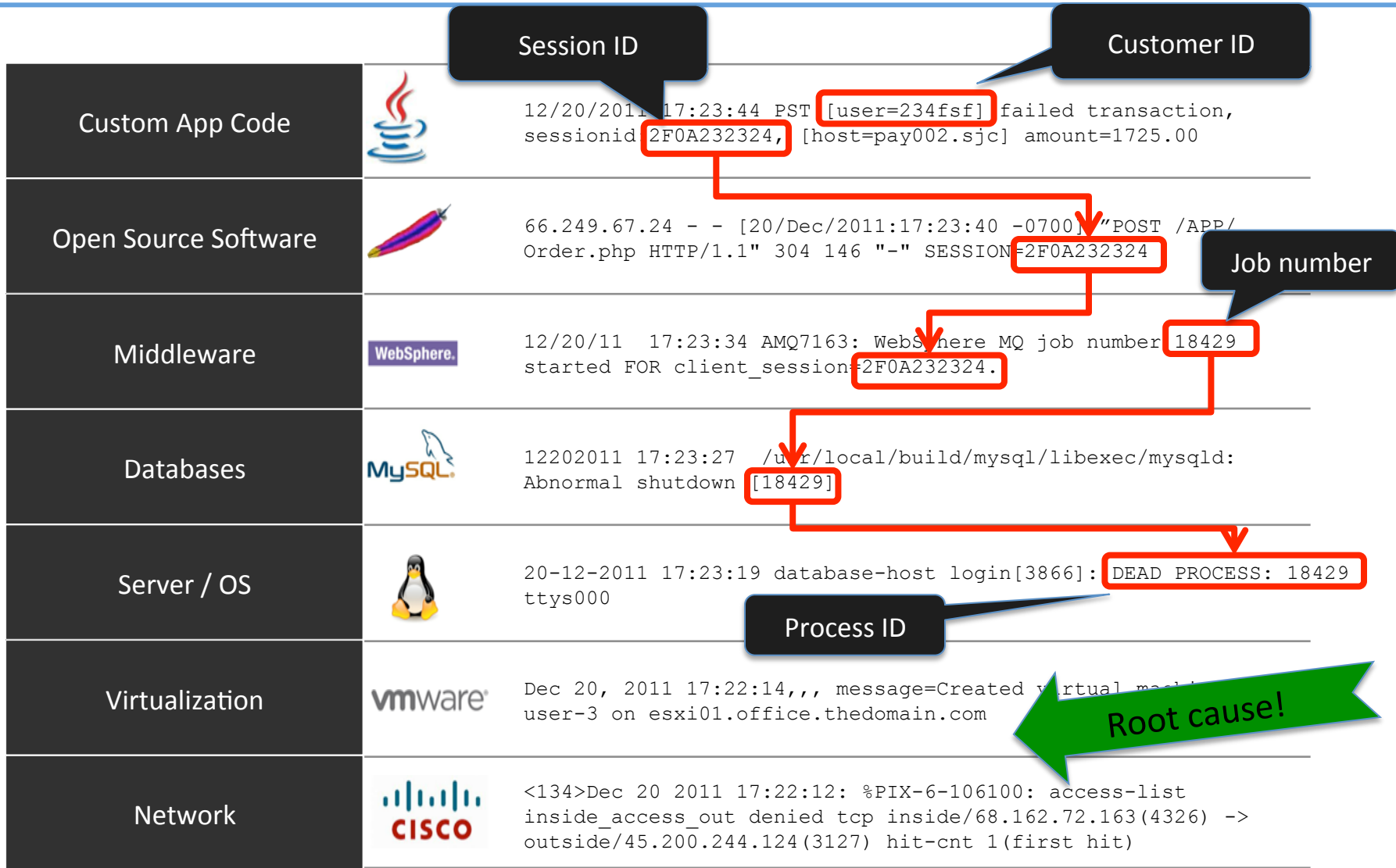
# What's missing

```
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-8:000000000000483D:0000000000000005:false]
[remote_ip=184.73.74.54] [web_session=MepMG8CS...] Pile for customer:
'0000000000000005', ID: '800000006407637B', block: '80000000004C9A11', msg
count: '1', size: '264', collector: '000000000000483D'
```





# Traversing the stack



Session ID

Customer ID

Job number

Process ID

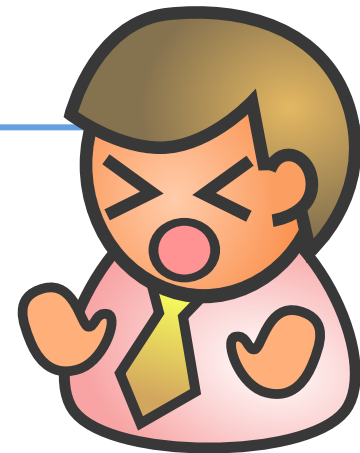
Root cause!

# Log use cases – “organizational perception”

## Enhanced visibility into machine behaviors

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

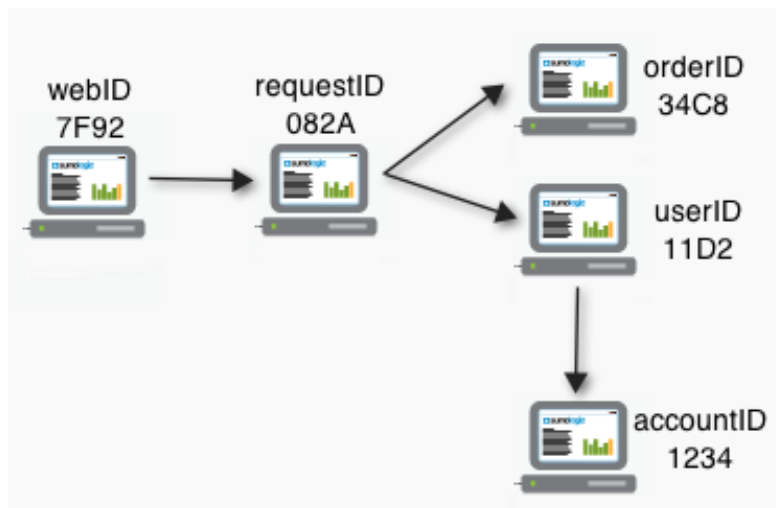
# Log challenges



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

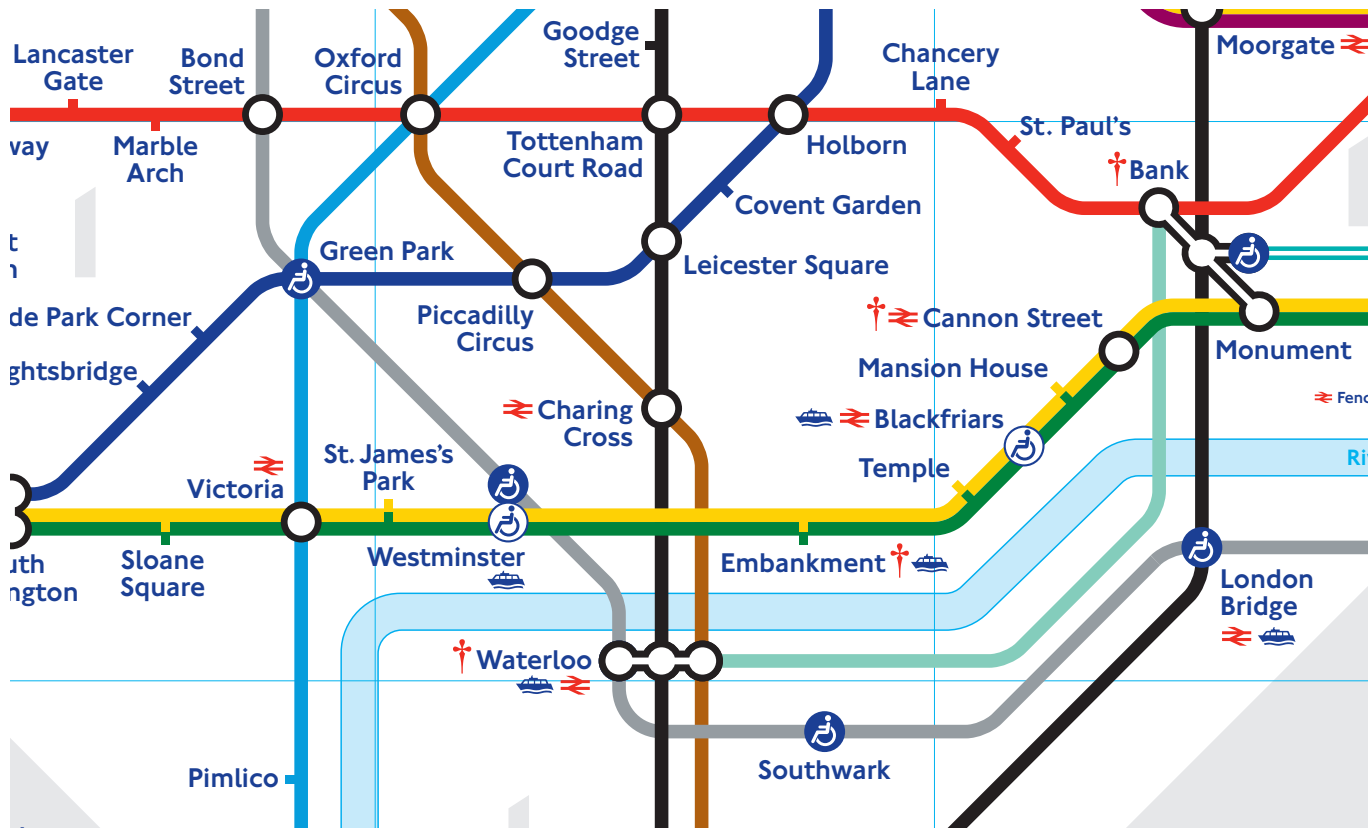
```
[2008-05-07 09:50:08.450 'App' 3560 verbose] [VpxdHeartbeat] Invalid heartbeat from 10.17.218.46
```

- Noise
- Cascading failures



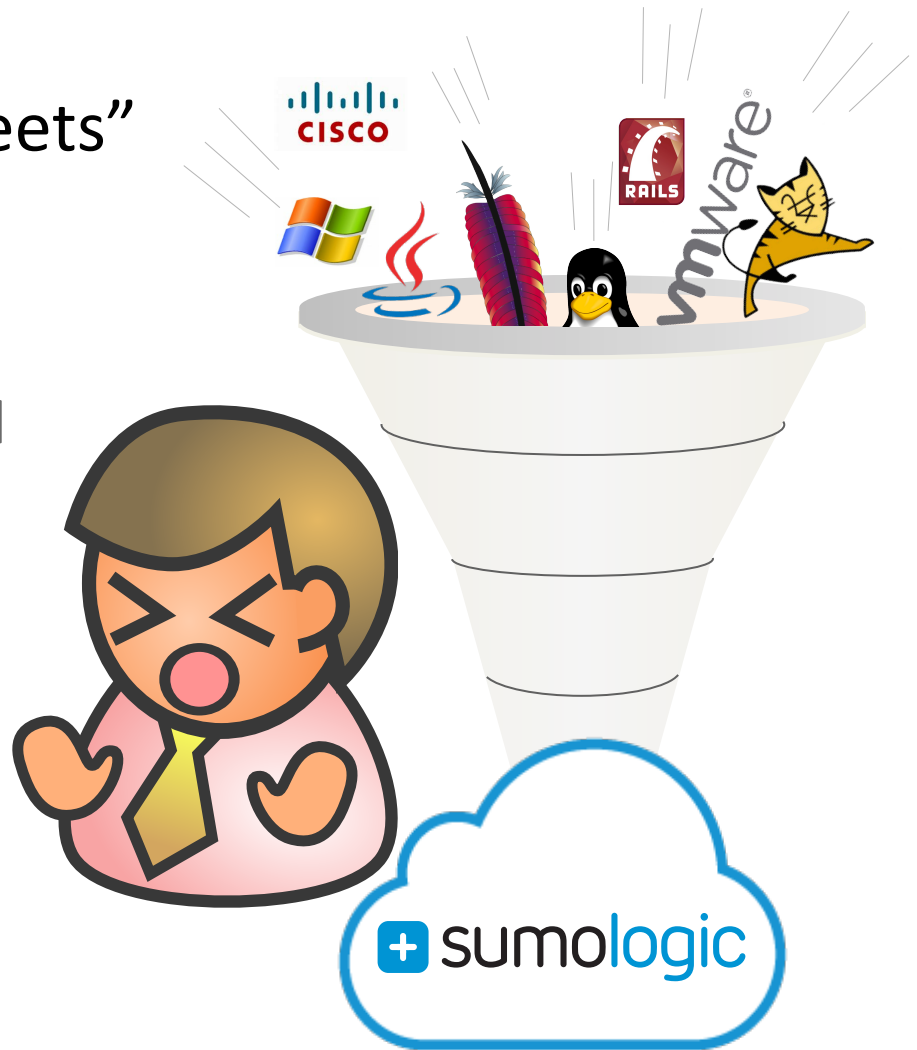
# Complexity

“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



# “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>

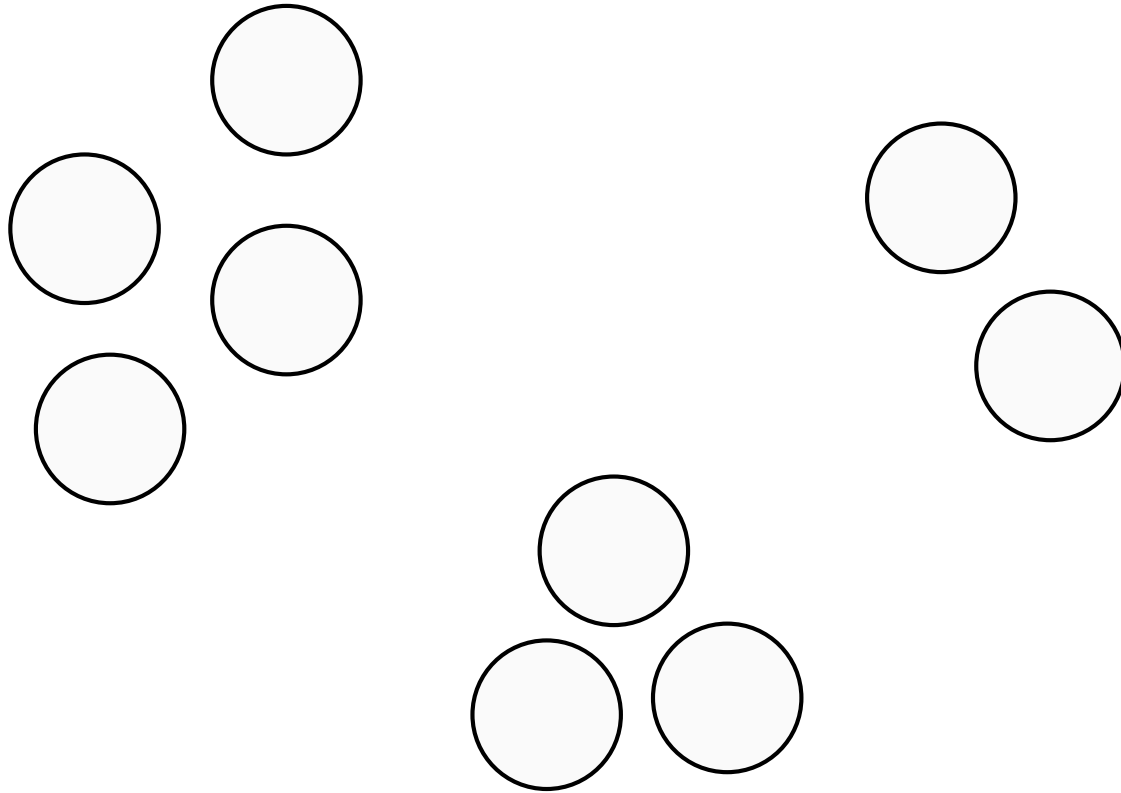
# Systems that learn from experience



## Unsupervised clustering

$\hat{f}(x)$

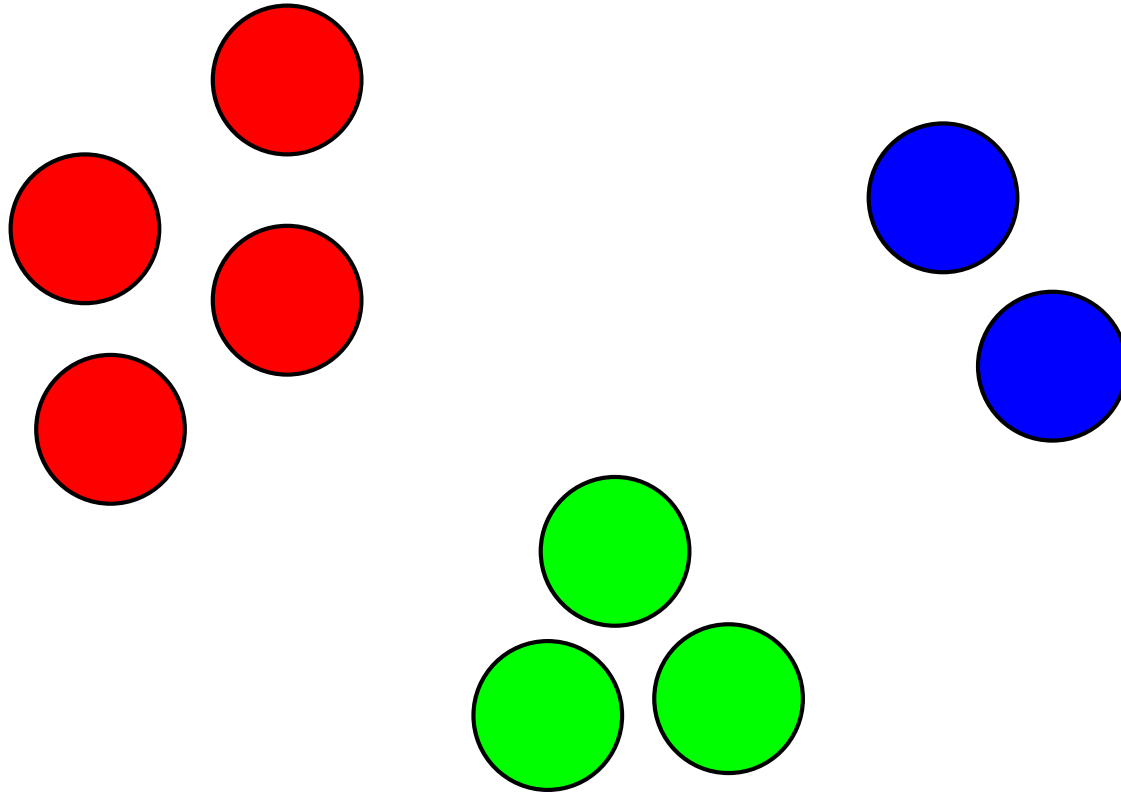
- **Given:** set of items
- **Do:** group similar items



$\hat{f}(x)$ 

## Unsupervised clustering

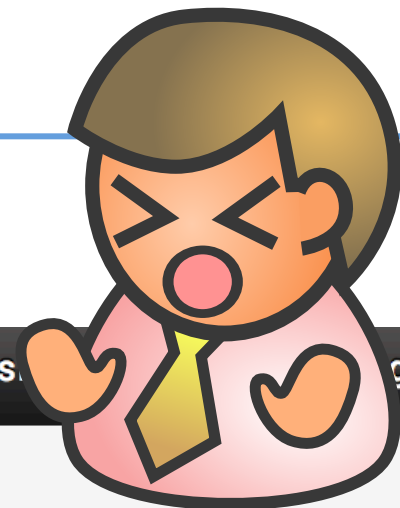
- **Given:** set of items
- **Do:** group similar items





# Too many logs! "data disorientation"

~60k results: 30 minutes, one component



sumologic 20.1-2846 Search Anomalies Das... ge ▾

Innamed Search Unnamed Search +

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

Messages

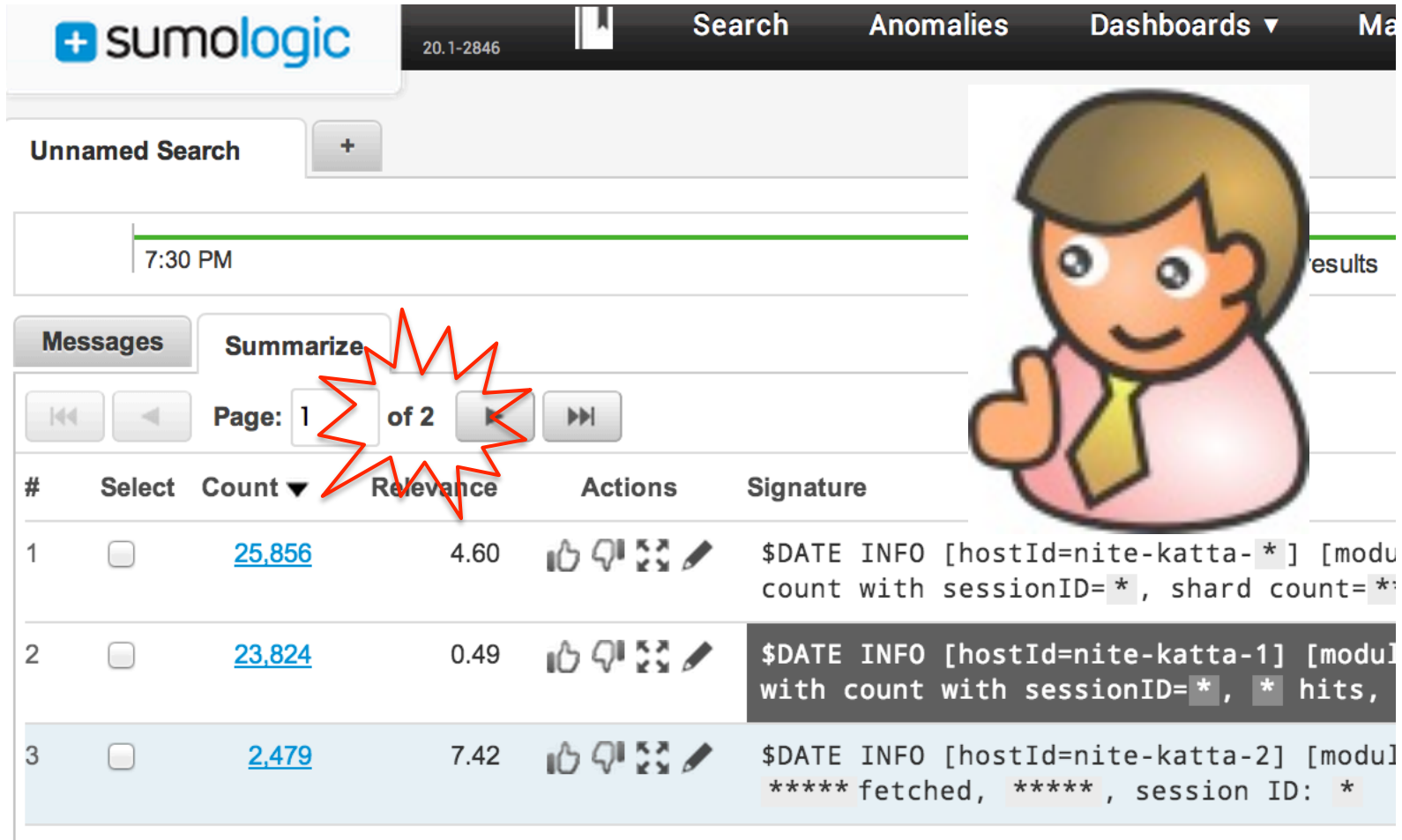
Page: 1 of 3938 LogReduce

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: FFFFFFFFFFFFFFFF 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



The screenshot shows the Sumologic interface with a search results table. A red starburst highlights the 'Count' and 'Relevance' columns. The table has the following data:

#	Select	Count	Relevance	Actions	Signature
1	<input type="checkbox"/>	<a href="#">25,856</a>	4.60		\$DATE INFO [hostId=nite-katta-*] [modu count with sessionId=*, shard count=*
2	<input type="checkbox"/>	<a href="#">23,824</a>	0.49		\$DATE INFO [hostId=nite-katta-1] [modu with count with sessionId=*, * hits,
3	<input type="checkbox"/>	<a href="#">2,479</a>	7.42		\$DATE INFO [hostId=nite-katta-2] [modu ***** fetched, ***** , session ID: *

Navigation controls include 'Messages', 'Summarize', and 'Page: 1 of 2'. A cartoon character is visible on the right side of the interface.

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
```



"magic"

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

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

## Unsupervised clustering

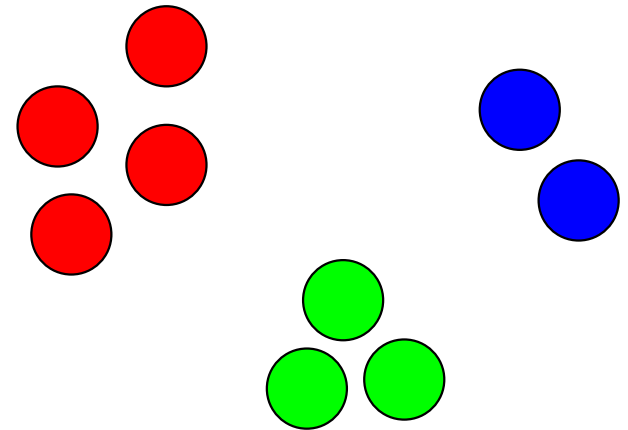
- **Given:** log messages
- **Do:** group by “signature”

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

A B C ~~D~~ E  
↓  
A Z C E

$$d(\ell_1, \ell_2) = 2$$

2. Do **distance-based clustering**



# Drill-down into the original raw logs



Messages Summarize

Page: 1 of 2

#	Select	Count	Actions	Signature
1	<input type="checkbox"/>	<u>8,497</u>		\$DATE INFO [hostId [thread=IPC Server shards=[000000000000

```
2013-04-24 09:20:53,997 -0700 INFO [hostId=nite-katta-1] [module=KATT/
[thread=IPC Server handler 8 on 20000] STARTED Calling getDetailsBatch
shards=[0000000000000000131-99F822FECEBFE19C#shard], docIds.length=2500
Host: nite-katta-1 Name: /usr/sumo/katta-sumo-20.1-658/logs/katta.log Category
```

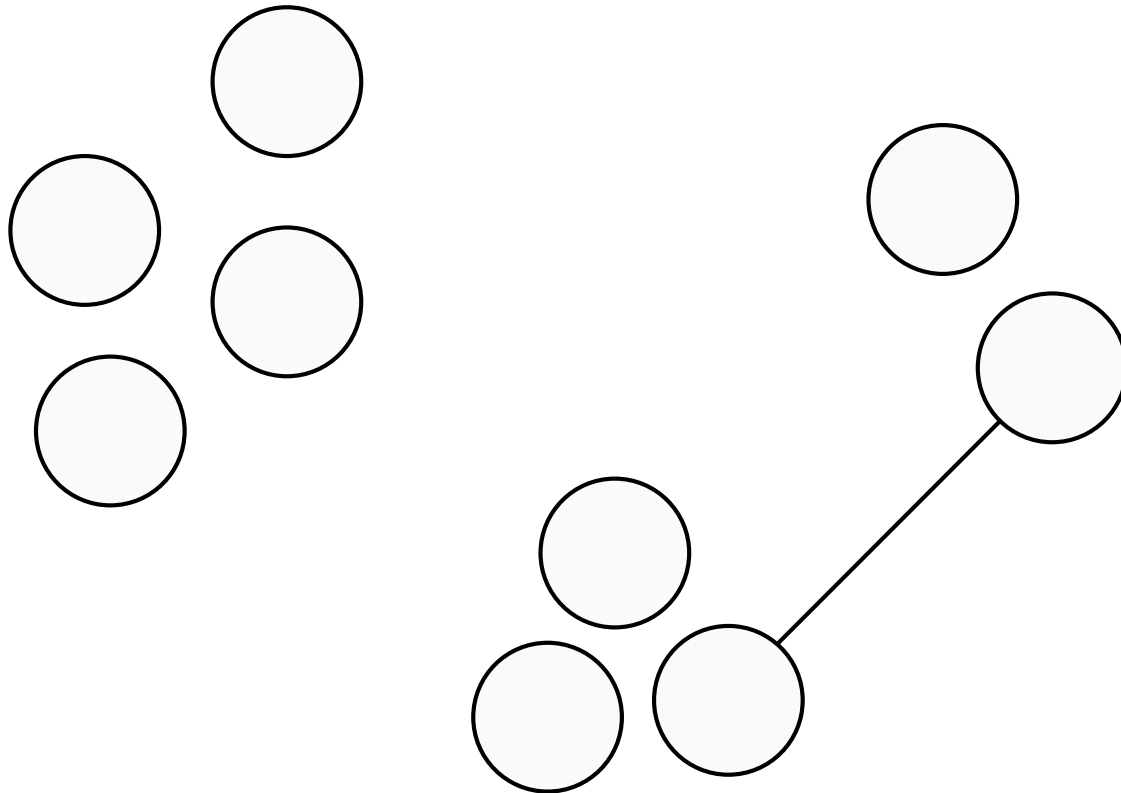
```
2013-04-24 09:20:53,674 -0700 INFO [hostId=nite-katta-1] [module=KATT/
[thread=IPC Server handler 6 on 20000] FINISHED Calling getDetailsBatch
shards=[0000000000000000131-99F822FECEBFE19C#shard], docIds.length=2497 at
Host: nite-katta-1 Name: /usr/sumo/katta-sumo-20.1-658/logs/katta.log Category
```

```
2013-04-24 09:20:53,462 -0700 INFO [hostId=nite-katta-1] [module=KATT/
```

$\hat{f}(x)$ 

## Partially supervised clustering

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

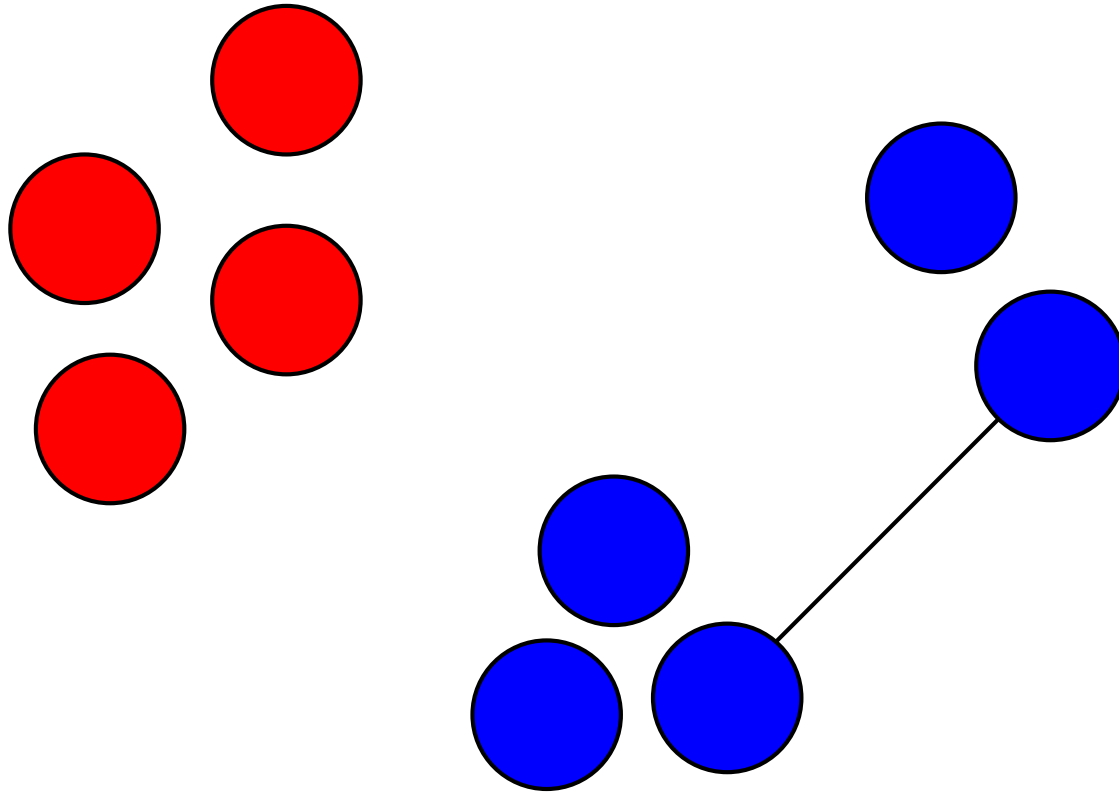




$\hat{f}(x)$ 

## Partially supervised clustering

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



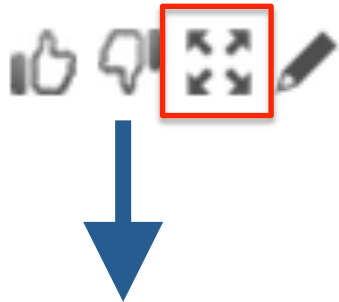
# Too many wildcards!



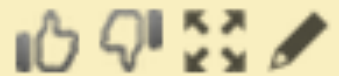
```
$DATE INFO [hostId=long-katta-*] [module=KATTA  
[thread=IPC Server handler * on 20000] ***** ED  
shards=[000000000000000005-*****
```



# “Hint” from human user



```
$DATE INFO [hostId=long-katta-*] [module=KATTA  
[thread=IPC Server handler * on 20000] ***** ED  
shards=[000000000000000005-*****
```



```
$DATE INFO [hostId=long-katta-17] [module=KATTA]  
[thread=IPC Server handler * on 20000] FINISHED C  
sessionID=89B99A496F555E41, shards=[000000000000000000
```



```
$DATE INFO [hostId=long-katta-1*] [module=KATTA]  
[thread=IPC Server handler * on 20000] STARTED Ca  
sessionID=8B6BB3EC5AA0B6E8, shards=[000000000000000000
```


# 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”

data mining



34,500,000 RESULTS

Any time ▾

## [Data mining - Wikipedia, the free encyclopedia](#)

[en.wikipedia.org/wiki/Data\\_mining](https://en.wikipedia.org/wiki/Data_mining) ▾

**Data mining** (the analysis step of the "Knowledge Discovery and **Data Mining**" process, or KDD), an interdisciplinary subfield of computer science is the ...

[Etymology](#) · [Background](#) · [Process](#) · [Standards](#) · [Notable uses](#)

## [Data Mining: What is Data Mining? - MBA, Executive MBA, ...](#)

[www.anderson.ucla.edu/.../teacher/technologies/palace/datamining.htm](http://www.anderson.ucla.edu/.../teacher/technologies/palace/datamining.htm) ▾

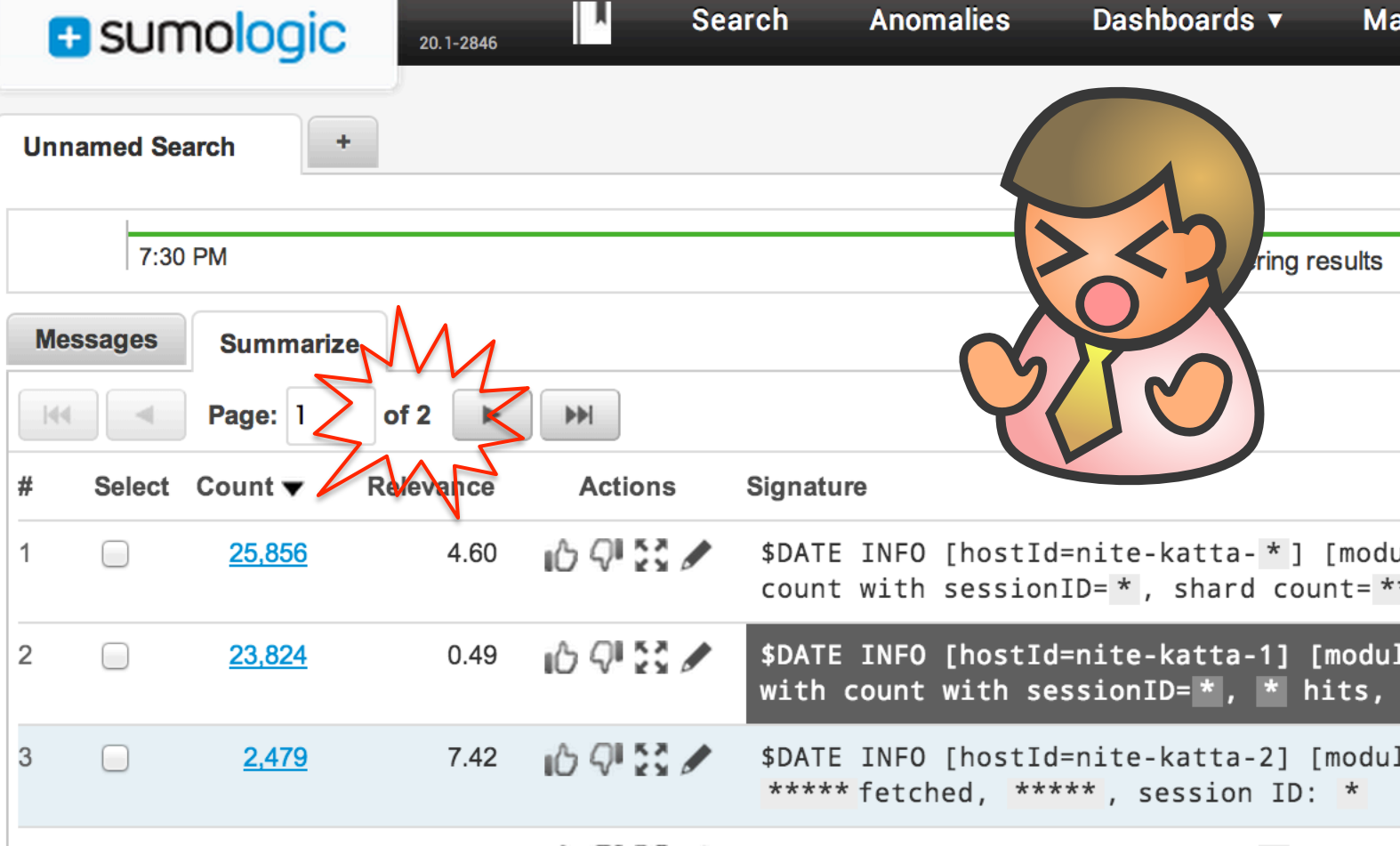
Overview Generally, **data mining** (sometimes called **data** or knowledge discovery) is the process of analyzing **data** from different perspectives and summarizing it into ...

## [An Introduction to Data Mining](#)

[www.thearing.com/text/dmwhite/dmwhite.htm](http://www.thearing.com/text/dmwhite/dmwhite.htm) ▾

Table 2 - **Data Mining** for Prospecting . The goal in prospecting is to make some calculated guesses about the information in the lower right hand quadrant based on ...

# Two pages is still too many!



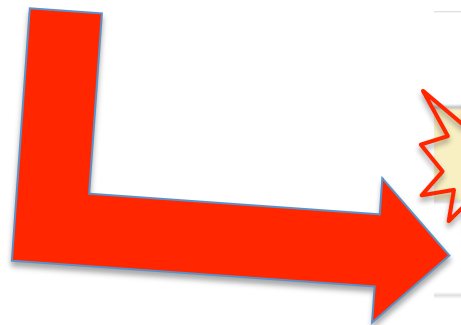
The screenshot shows the Sumologic search interface. At the top, there's a navigation bar with 'sumologic', '20.1-2846', and menu items 'Search', 'Anomalies', 'Dashboards', and 'Ma'. Below this is a search bar labeled 'Unnamed Search' with a '+' button. A timestamp '7:30 PM' is visible. The main content area has tabs for 'Messages' and 'Summarize'. Below the tabs is a pagination control showing 'Page: 1 of 2' with navigation arrows. A red starburst graphic highlights the 'Page: 1 of 2' text. To the right, a cartoon character with a frustrated expression is shown. Below the pagination is a table with columns: '#', 'Select', 'Count', 'Relevance', 'Actions', and 'Signature'. The table contains three rows of search results.

#	Select	Count	Relevance	Actions	Signature
1	<input type="checkbox"/>	<a href="#">25,856</a>	4.60		<code>\$DATE INFO [hostId=nite-katta-*] [modu count with sessionId=*, shard count=*</code>
2	<input type="checkbox"/>	<a href="#">23,824</a>	0.49		<code>\$DATE INFO [hostId=nite-katta-1] [modu with count with sessionId=*, * hits,</code>
3	<input type="checkbox"/>	<a href="#">2,479</a>	7.42		<code>\$DATE INFO [hostId=nite-katta-2] [modu ***** fetched, ***** , session ID: *</code>

Actions	Signature
👍 👎 🔄 ✎	[hostId=tamalpais-1] INFO - JOB SUCCESS
👍 👎 🔄 ✎	[hostId=tamalpais-1] INFO - ASSIGNING WOR
👍 👎 🔄 ✎	[hostId=mirrorlake-2] INFO - login atter
👍 👎 🔄 ✎	[hostId=govnelson-3] INFO - database sta
👍 👎 🔄 ✎	[hostId=bigsur-*] WARNING - database co
👍 👎 🔄 ✎	[hostId=govnelson-2] INFO - database shut

## Learning to rank

- Given:
  - signatures,
  - user activity
- Do: rank by "relevance"

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


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

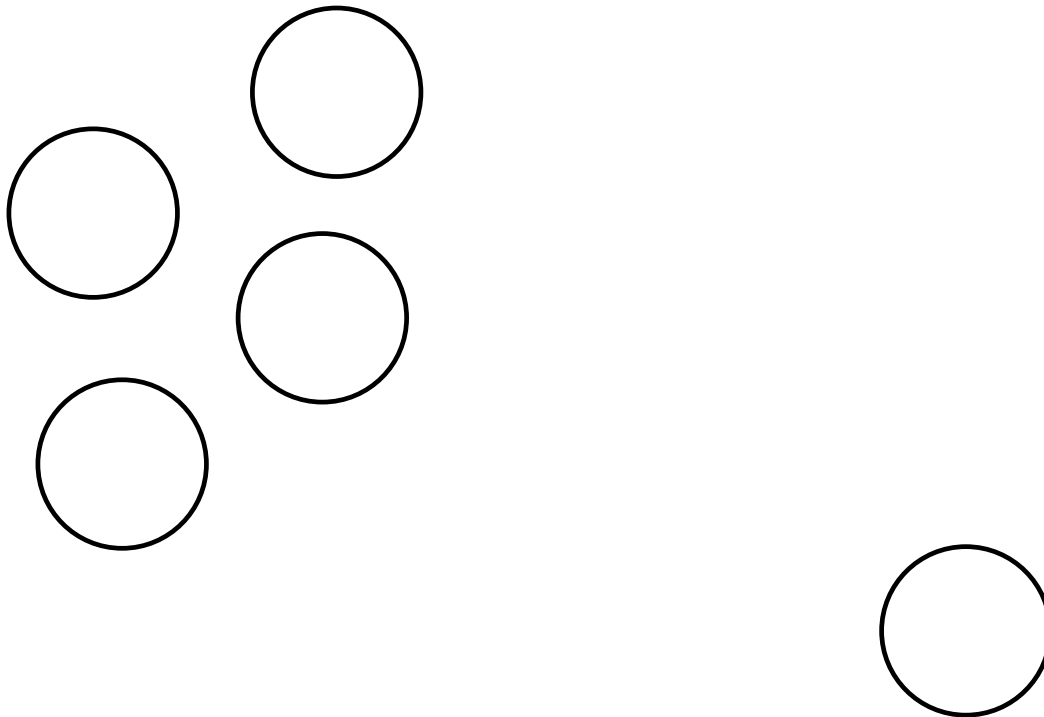




$\hat{f}(x)$ 

## Outlier detection

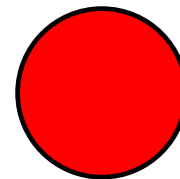
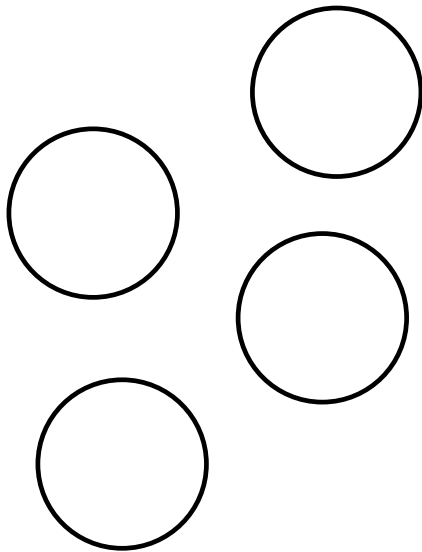
- **Given:** data points
- **Do:** identify outliers



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

## Outlier detection

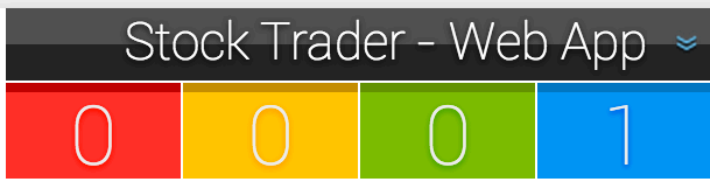
- **Given:** data points
- **Do:** identify outliers



$\hat{f}(x)$  **Anomaly detection**

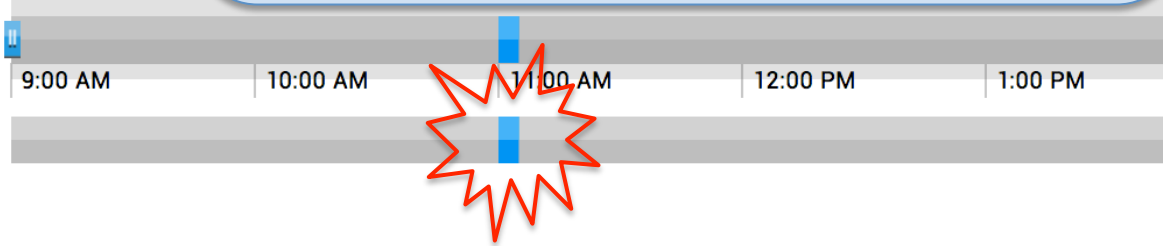
- **Given:** log data
- **Do:** flag anomalies

Health check OK	$\left[ \begin{array}{c} 33 \\ 29 \\ 3 \end{array} \right]$	,	$\left[ \begin{array}{c} 30 \\ 26 \\ 6 \end{array} \right]$	,	$\left[ \begin{array}{c} 31 \\ 27 \\ 732 \end{array} \right]$
Request processed					
Txn timeout, retry					
	$t_1$		$t_2$		$t_3$



$\hat{f}(x)$  **Anomaly detection**

- Given: log data
- Do: flag anomalies



Health check OK	$\begin{bmatrix} 33 \\ 29 \\ 3 \end{bmatrix}$	,	$\begin{bmatrix} 30 \\ 26 \\ 6 \end{bmatrix}$	,	$\begin{bmatrix} 31 \\ 27 \\ 732 \end{bmatrix}$
Request processed					
Txn timeout, retry					
	$t_1$		$t_2$		$t_3$

# Investigate and annotate events

The screenshot shows the Sumologic interface for investigating an event. At the top, the Sumologic logo is on the left, and navigation links for 'Search', 'Anomalies', and 'Dashboards' are on the right. Below the navigation, the page title is 'Stock Trader - App Dev Database Timeout and User Issues'. The event details section shows the 'Event Name' as 'Database Timeout and User Issues' and the 'Severity' as 'High'. The 'Description' field contains the text 'Database timeout iss activity'. Below this, the 'Signatures' section is visible, featuring a table with columns for '#', 'Score', and 'Change'. The first signature is highlighted, showing a score of 1 and a change of up. The signature text is '\$DATE StockTraderWebApplicationServiceClient.sell Er System.Exception: Database timeout creating sell ord'. Below the signature table, there is a section for 'Messages with this Signature' with pagination controls and a list of messages, including one from '02/05/2014 13:19:59.000'.



# Investigate and annotate events

The screenshot shows the Sumologic interface for an event titled "Database Timeout and User Issues". The event name is "Database Timeout and User Issues" and the description is "Database timeout iss activity". The severity is set to "High". Below this, there is a "Signatures" section with a table of results. A red starburst highlights the first signature, which has a score of 1 and a change of up. The signature text is "\$DATE StockTraderWebApplicationServiceClient.sell Er System.Exception: Database timeout creating sell ord". Below the signature, there is a "Messages with this Signature" section showing a list of messages with their timestamps and source information.

#	Score	Change
1	1	↑
2	1	↓
3	1	↓

**Signature**

```
$DATE StockTraderWebApplicationServiceClient.sell Er System.Exception: Database timeout creating sell ord
```

**Messages with this Signature**

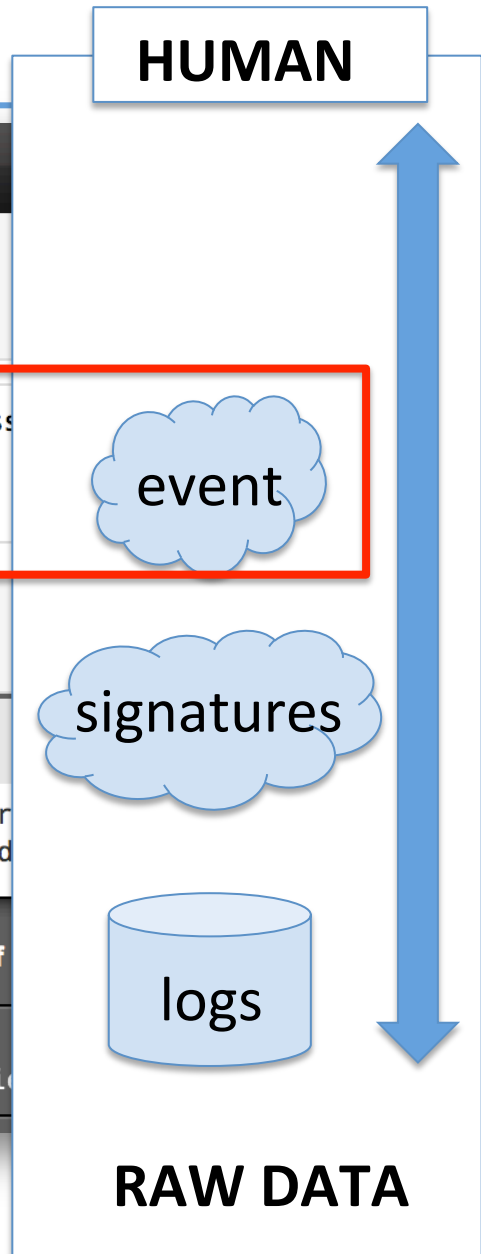
1	02/05/2014 13:19:59.000	2014-02-05 21:19:59	StockTraderWebApplicationServiceCli



# Investigate and annotate events

The screenshot shows the Sumologic interface for investigating an event. The event name is "Database Timeout and User Issues" with a severity of "High". The description is "Database timeout iss activity". Below this, the "Signatures" section shows a table of signatures with columns for #, Score, Change, and Signature. The first signature is "\$DATE StockTraderWebApplicationServiceClient.sell Er System.Exception: Database timeout creating sell ord".

#	Score	Change	Signature
1	●	↑	\$DATE StockTraderWebApplicationServiceClient.sell Er System.Exception: Database timeout creating sell ord
2	●	↓	
3	●	↓	



# Investigate and annotate events

The screenshot shows the Sumologic interface for investigating an event. At the top, the Sumologic logo is on the left, and navigation links for 'Search', 'Anomalies', and 'Dashboards' are on the right. The page title is 'Stock Trader - App Dev Database Timeout and User Issues'. Below this, there are fields for 'Event Name' (Database Timeout and User Issues) and 'Description' (Database timeout iss activity). The 'Severity' is set to 'High'. A 'Signatures' section contains a table with columns for '#', 'Score', 'Change', and 'Signature'. The first signature is '\$DATE StockTraderWebApplicationServiceClient.sell Er System.Exception: Database timeout creating sell ord'. Below the table, there are controls for 'Messages with this Signature' and a pagination indicator 'Page: 1 of'. The table shows one message with a timestamp of '02/05/2014 13:19:59.000' and the same signature text.

#	Score	Change	Signature
1	●	↑	\$DATE StockTraderWebApplicationServiceClient.sell Er System.Exception: Database timeout creating sell ord
2	●	↓	
3	●	↓	

Messages with this Signature Page: 1 of

1	02/05/2014 13:19:59.000	2014-02-05 21:19:59	StockTraderWebApplicationServiceCli
---	-------------------------	---------------------	-------------------------------------

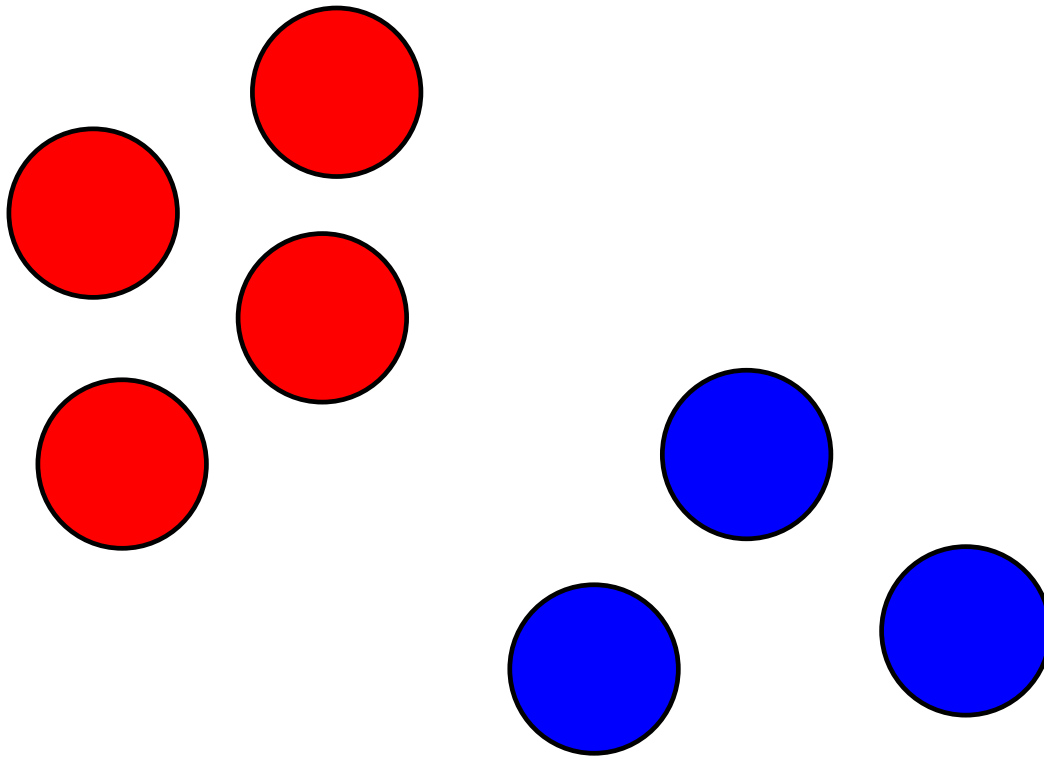




$$\hat{f}(x)$$

## Supervised classification

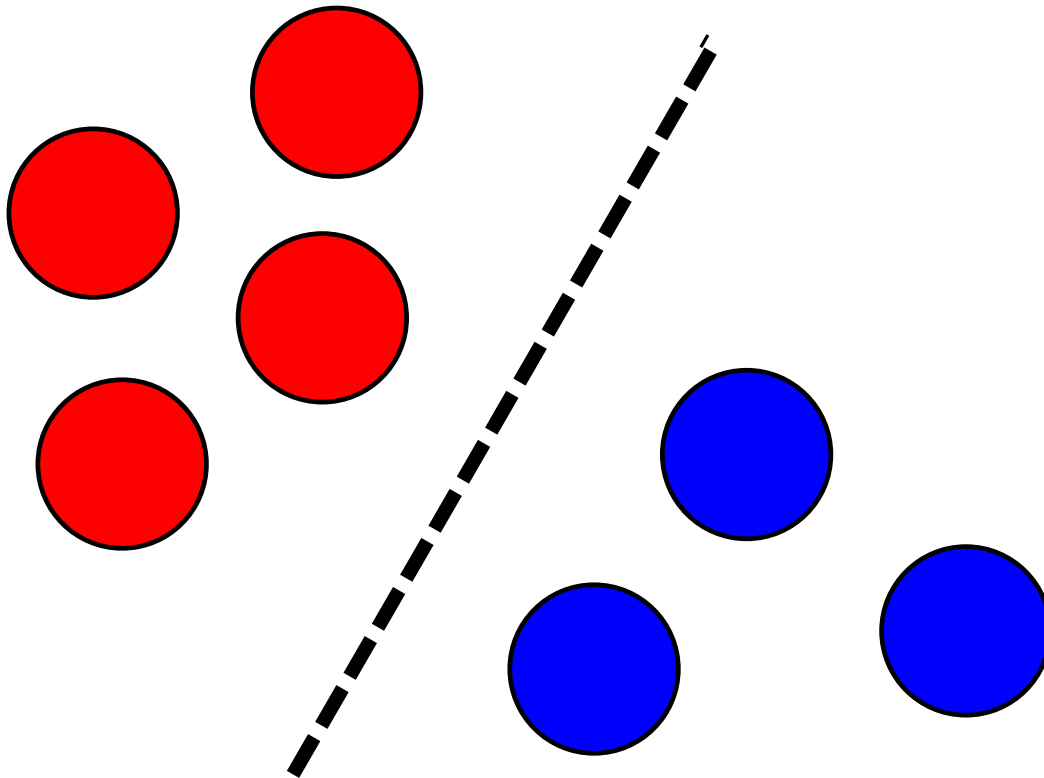
- **Given:** labeled data points
- **Do:** predict future labels



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

## Supervised classification

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



## Supervised classification

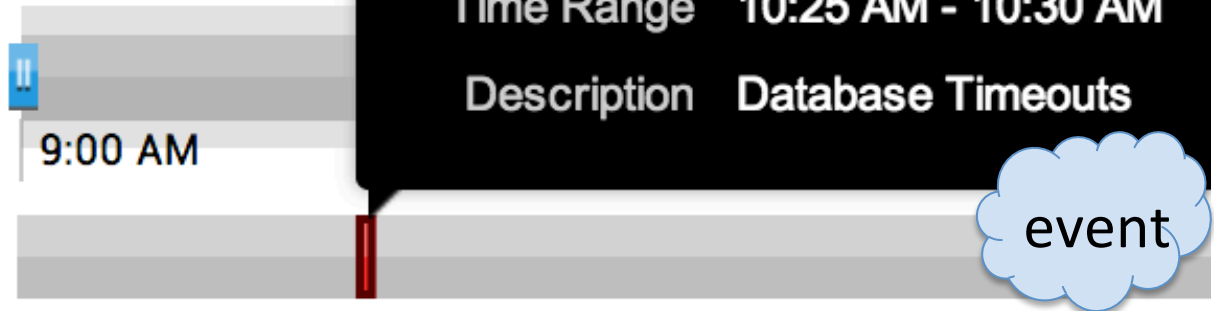
 $\hat{f}(x)$ 

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



timeline  
/ alerts

Database Timeouts



event

# 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

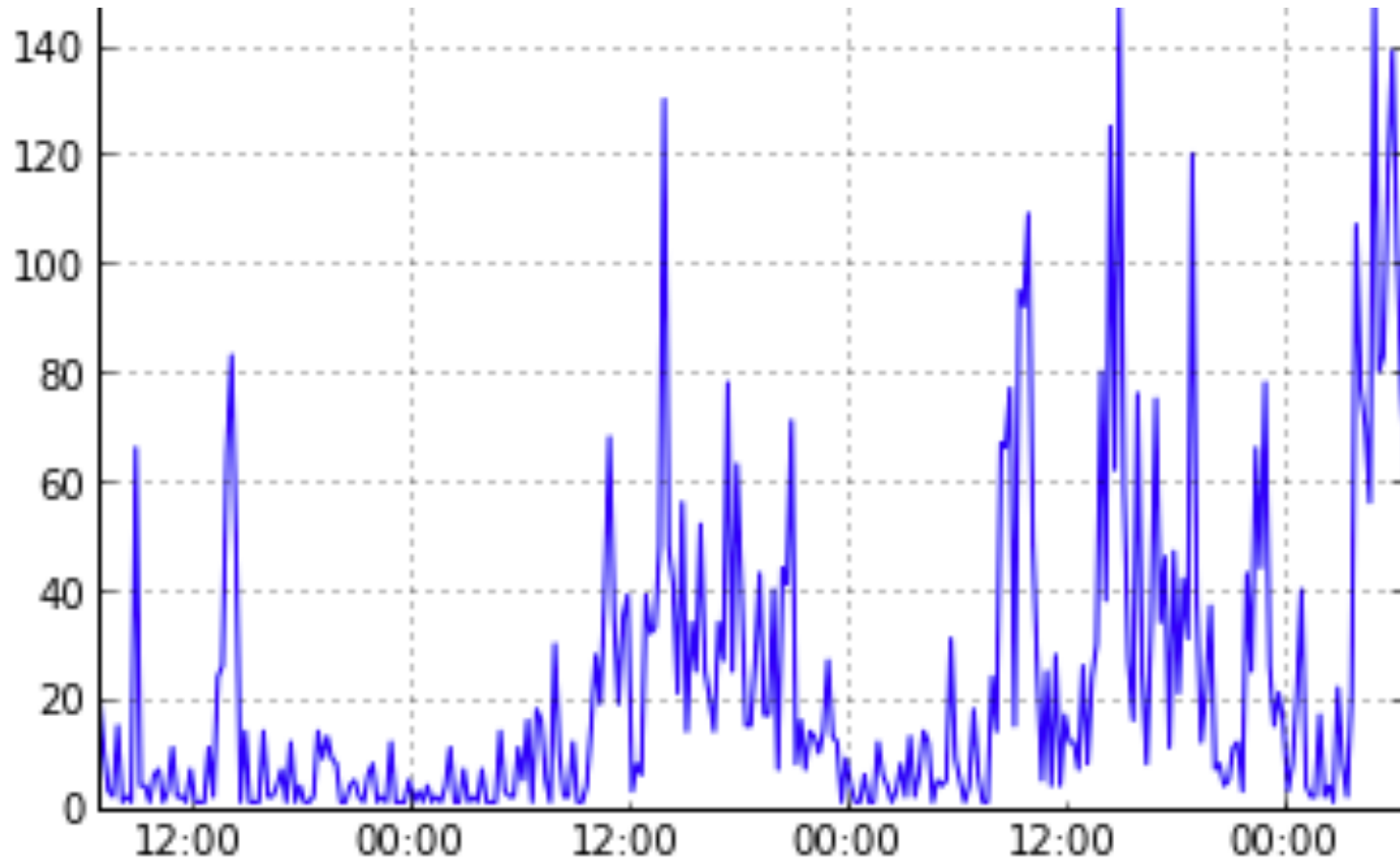
---



- **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

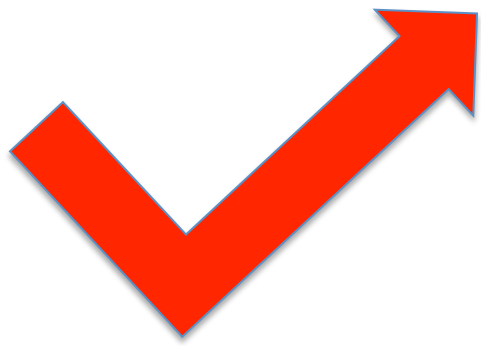
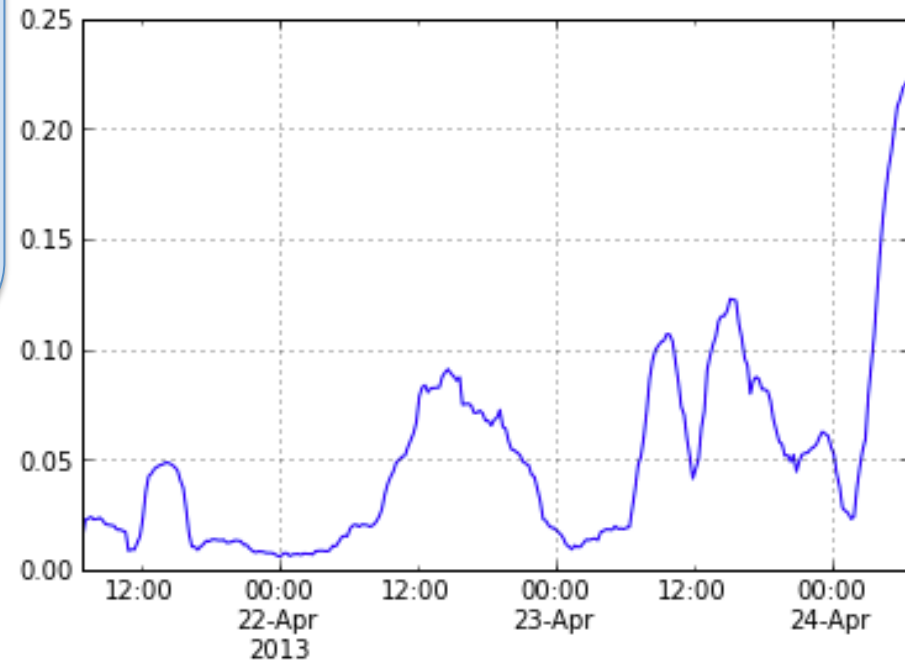
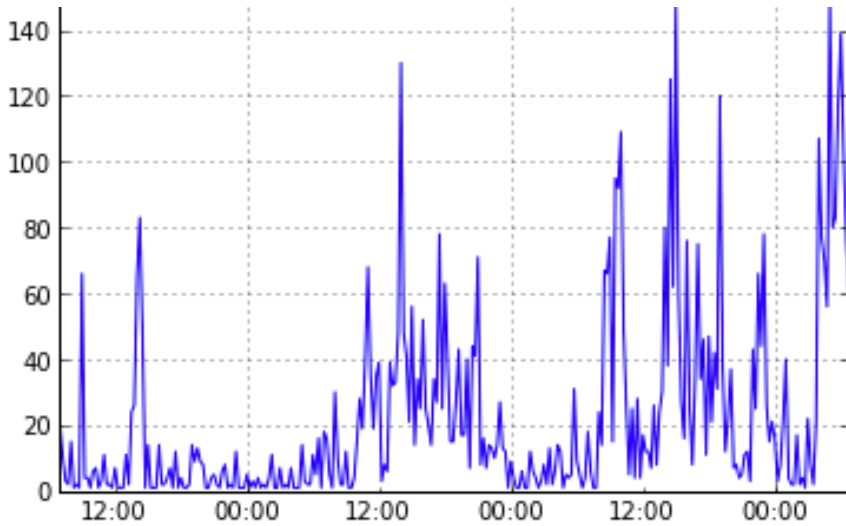




# Signal decomposition

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

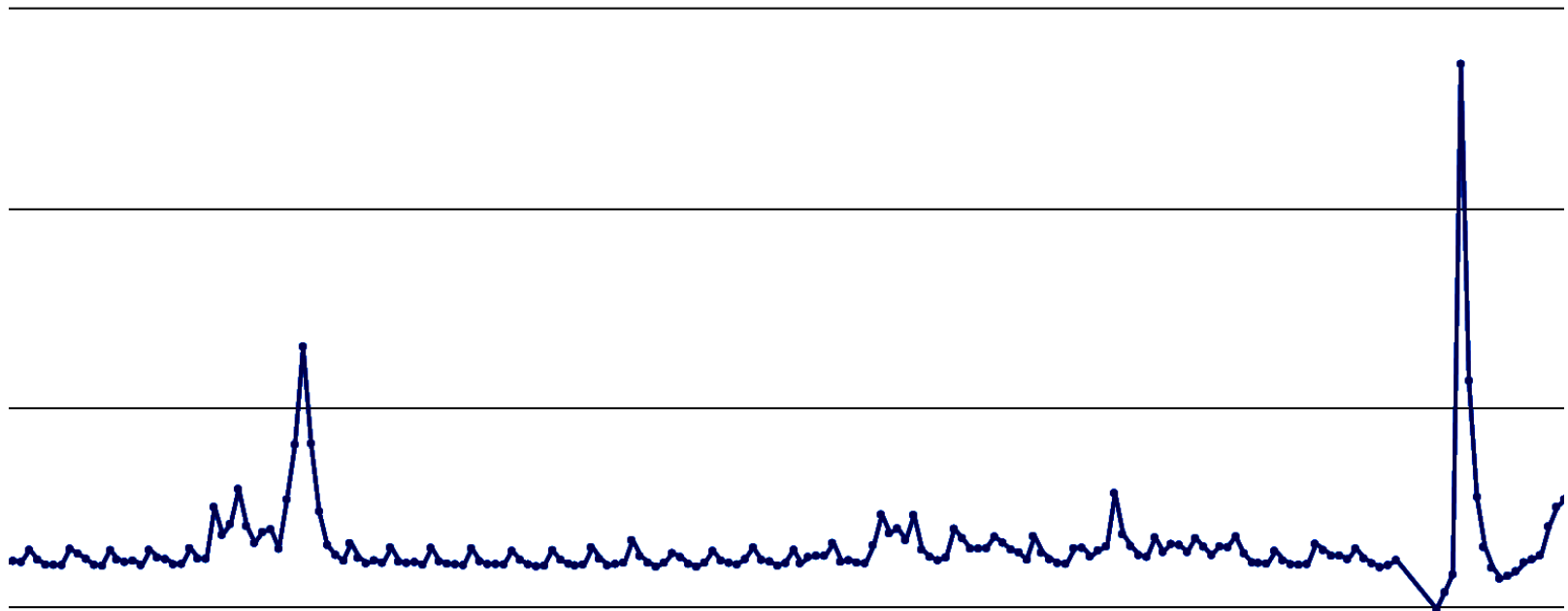
- **Given:** time-series
- **Do:** extract model components



$\hat{f}(x)$ 

## Outlier detection

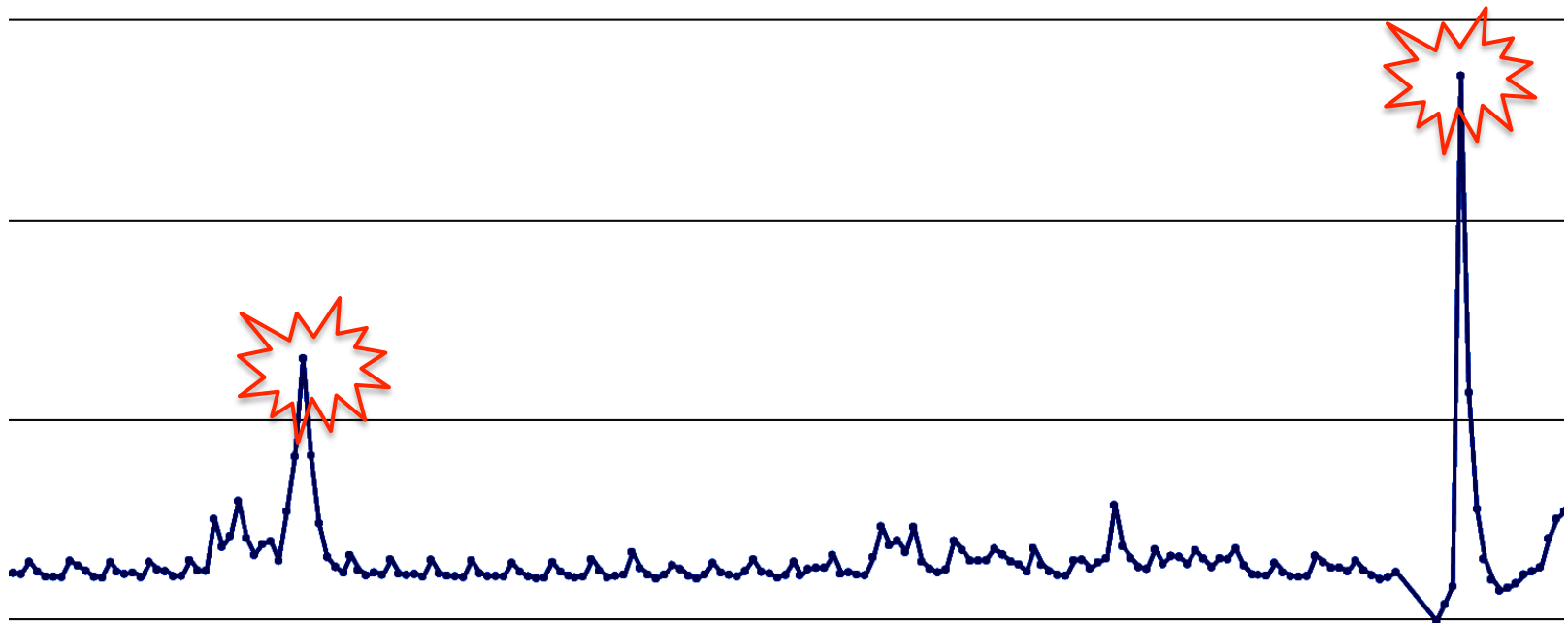
- **Given:** data points
- **Do:** identify outliers



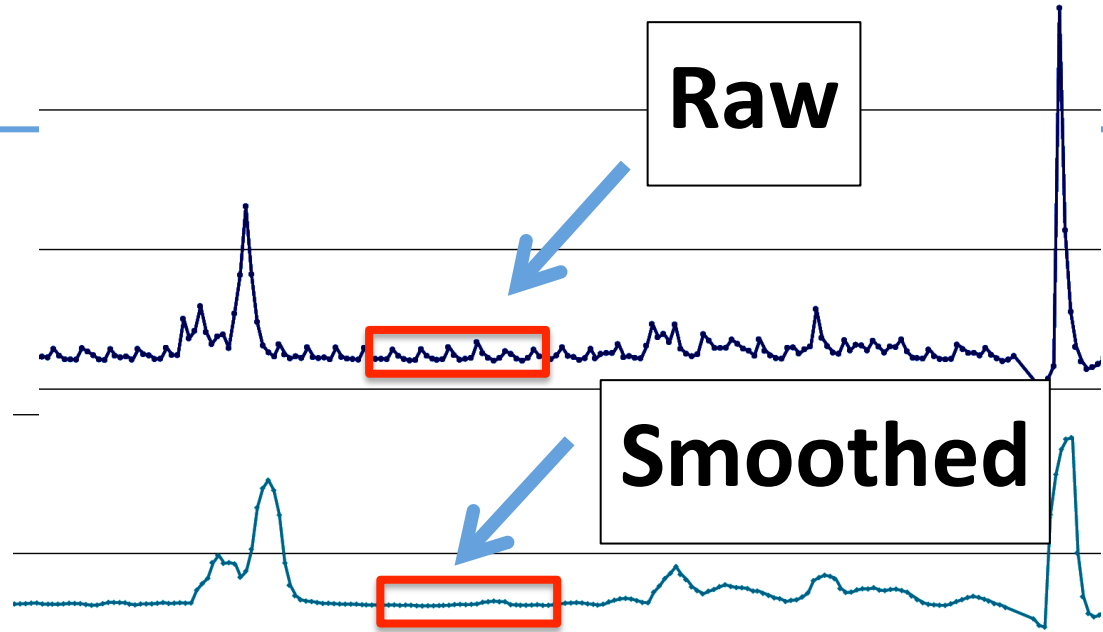
$$\hat{f}(x)$$

## Outlier detection

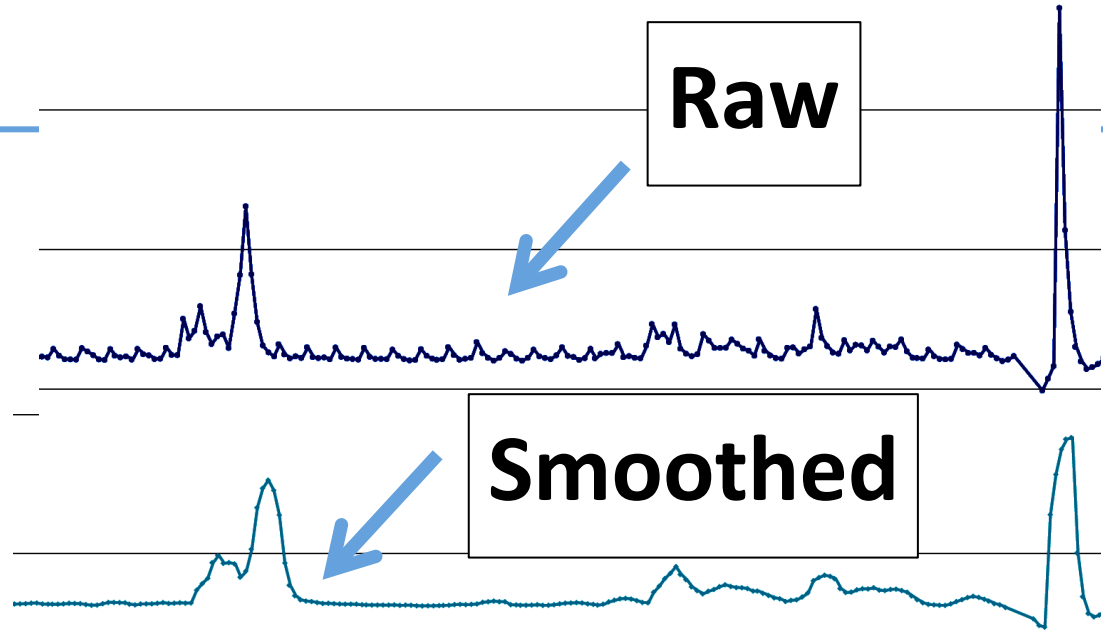
- **Given:** data points
- **Do:** identify outliers



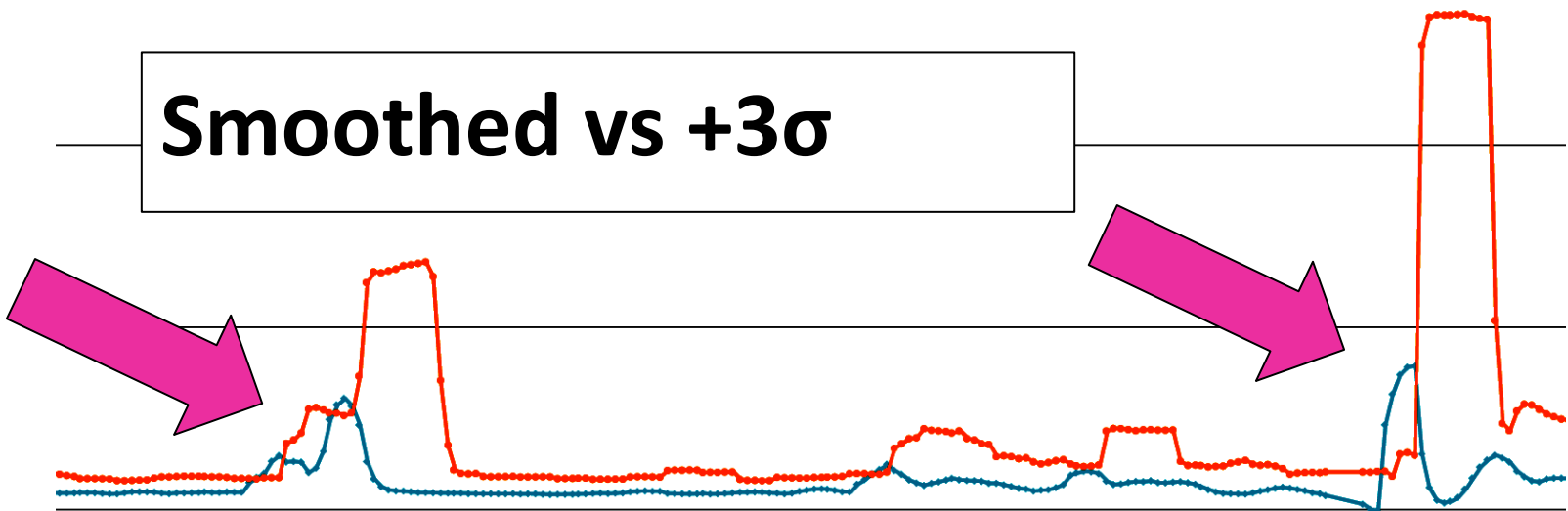
# Windowed model



# Windowed model



## Smoothed vs $+3\sigma$



# 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 \*

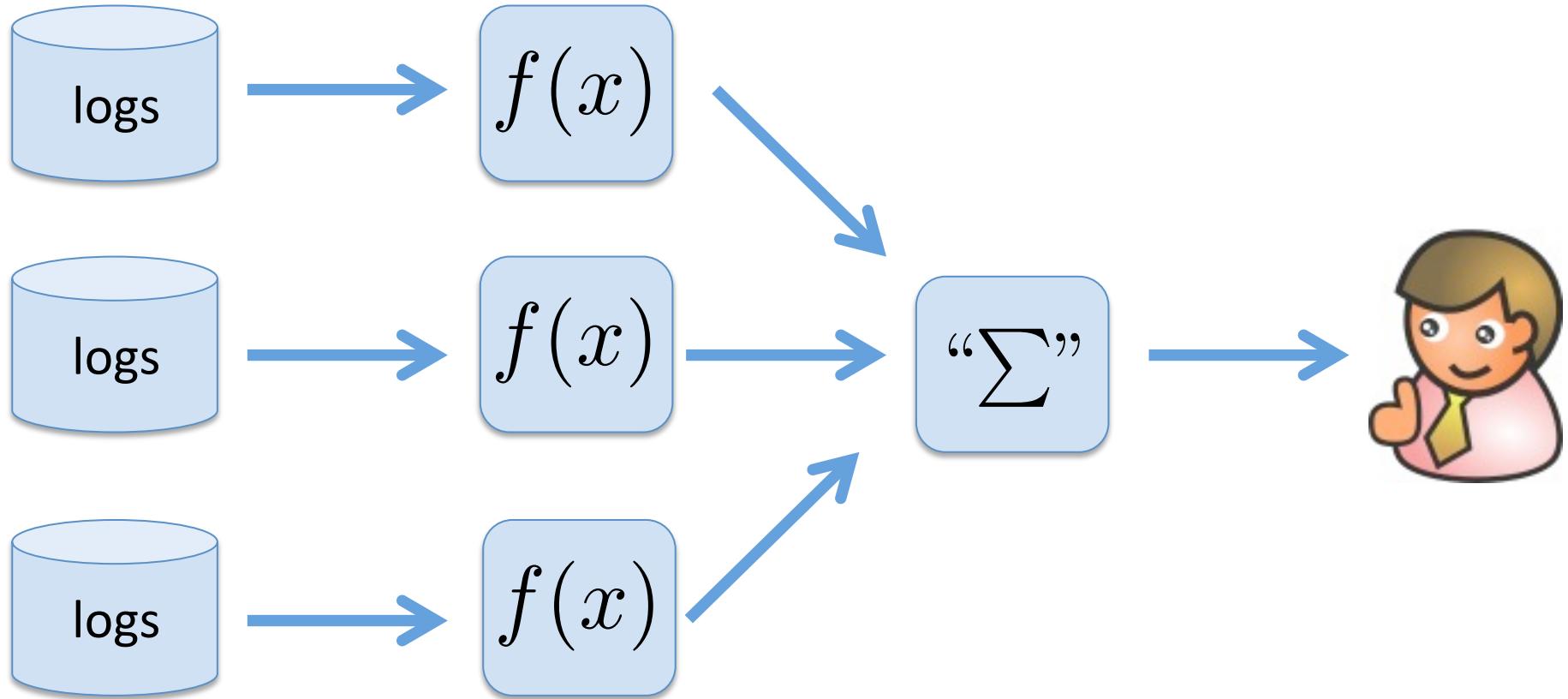
Phone \*

[Sign Up Now](#)

By clicking on the "Sign Up Now" button, you agree to accept our [Terms of Service](#)



# BONUS: scale-out, streaming architecture



# 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

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

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

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