



# Elastic Machine Learning Workshop

---

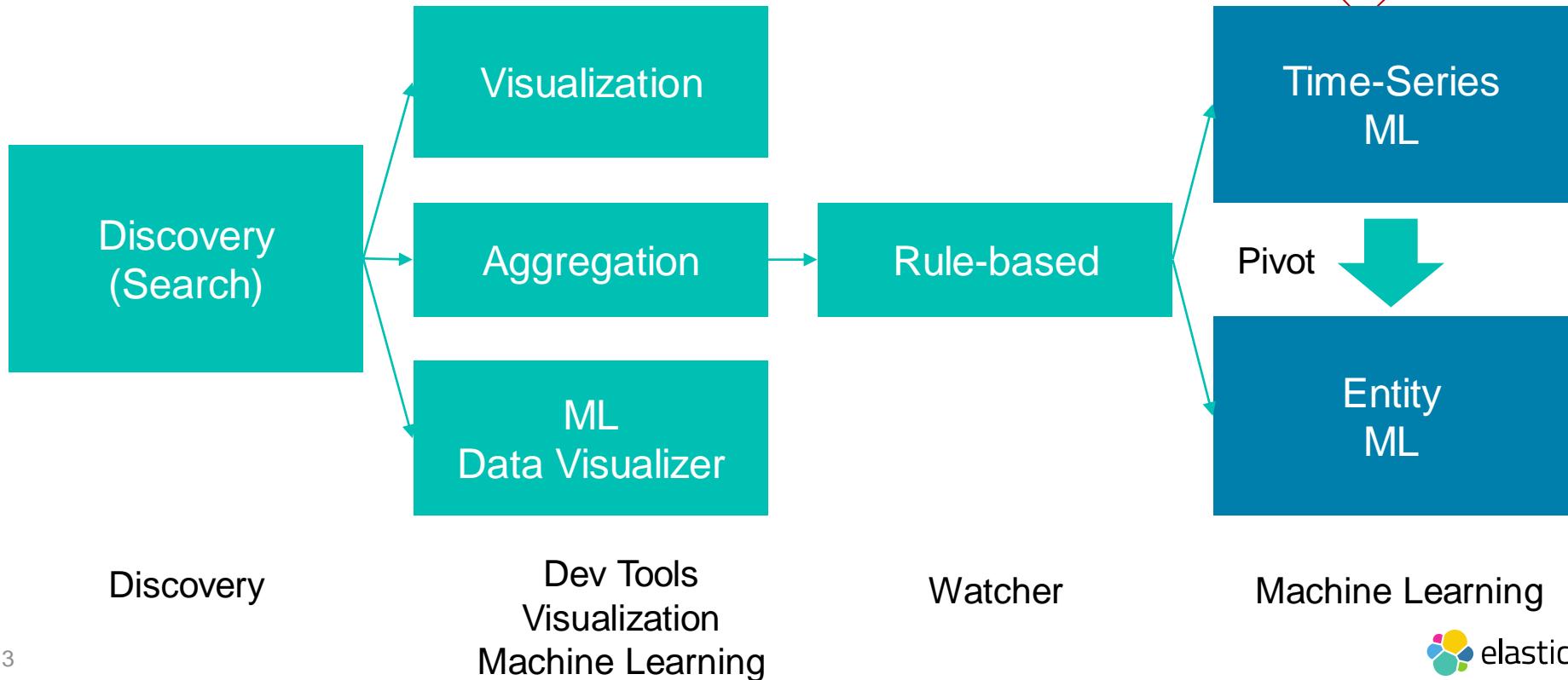
Elastic  
<http://www.elastic.co>



# Understanding your DATA is Everything (Discussion)

# Elasticsearch Analytics Features

Important



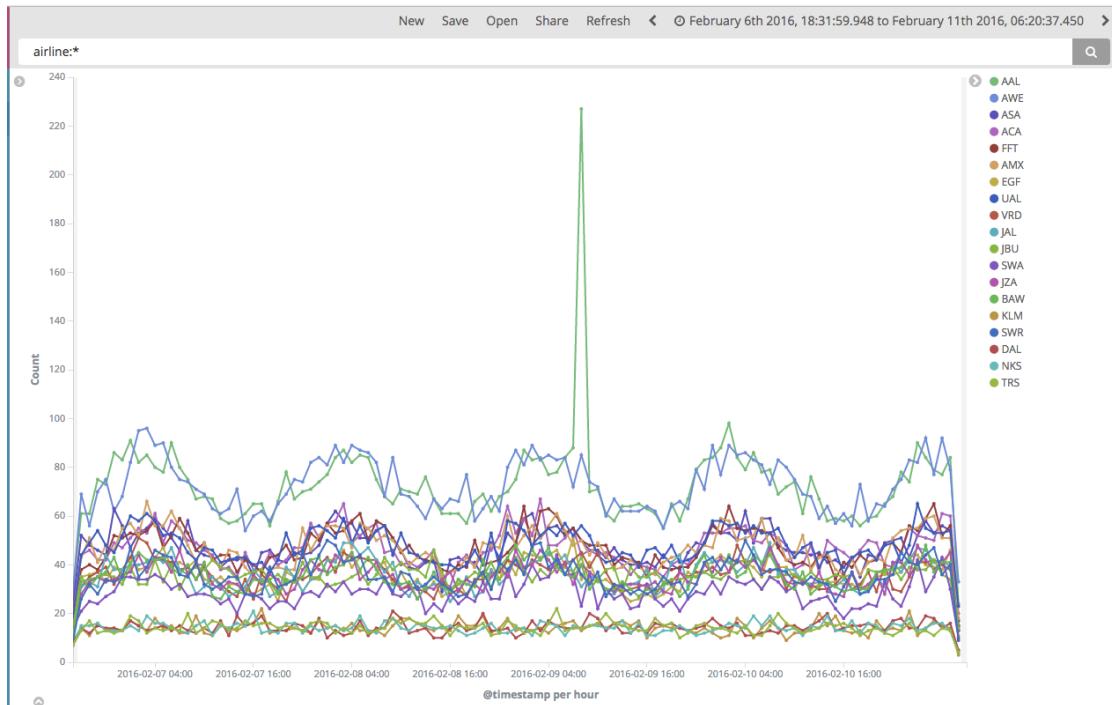
# 1. Times-Series based ML : Anomaly Detection Concepts

# What is “Abnormal”

# What is “Abnormal”?

What's abnormal here?

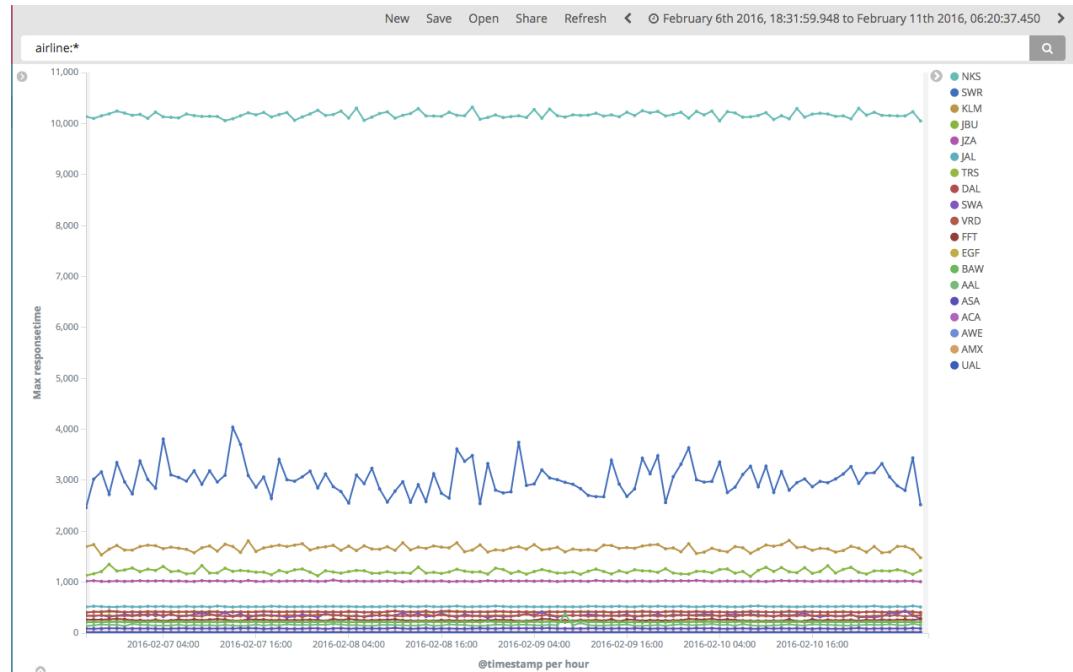
Why?



# What is “Abnormal”?

What's abnormal here?

Why?



# What is “Abnormal”?

What's abnormal here?

Why?

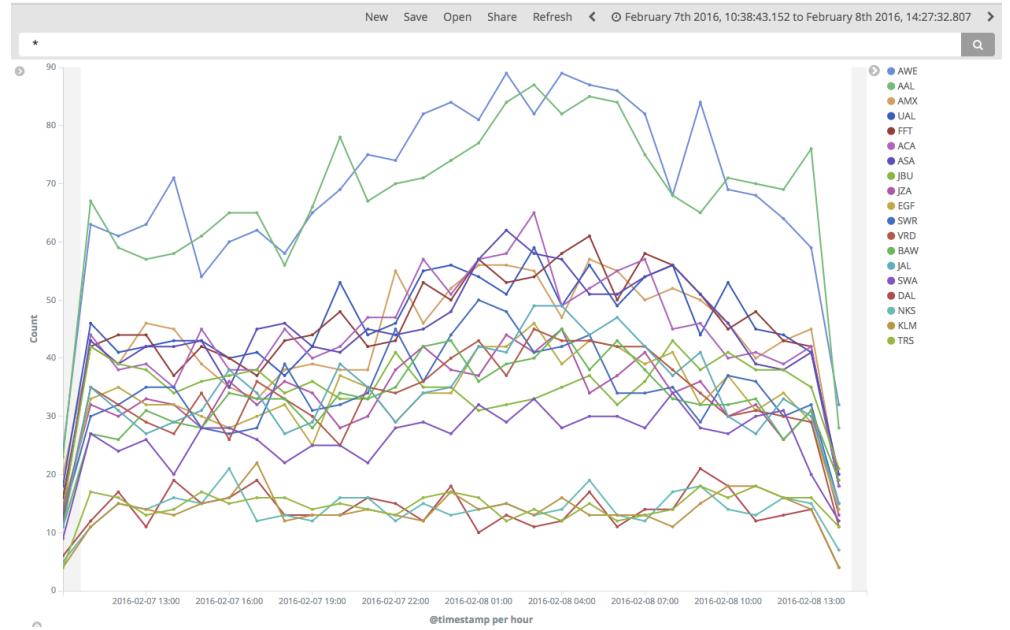


# What is “Normal”?

In general, this question can be answered in two ways:

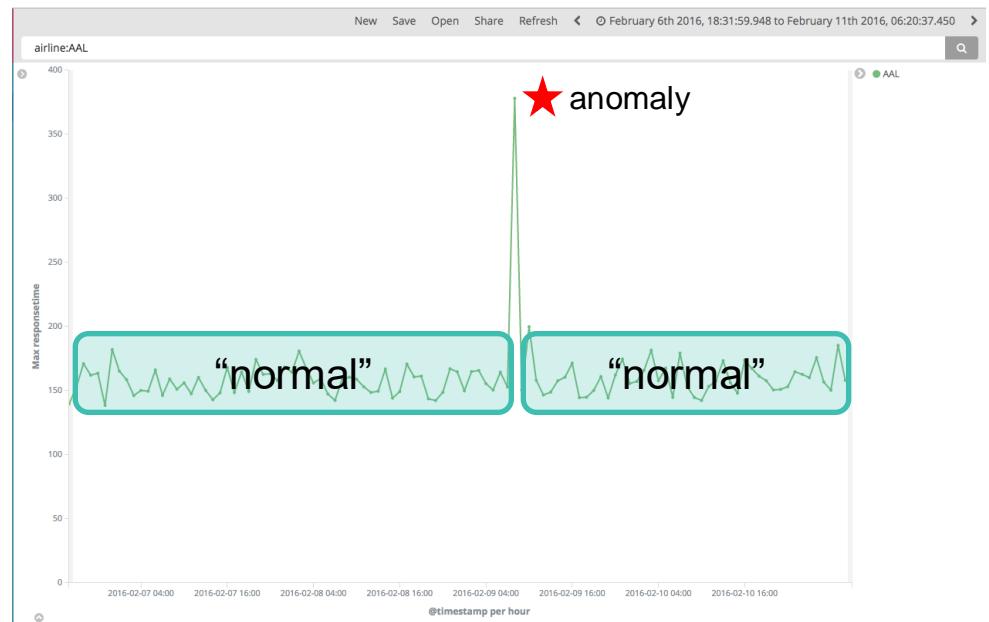
1) Something behaves in a consistent way with respect to itself, over time

2) Something behaves in a consistent way compared against similar entities



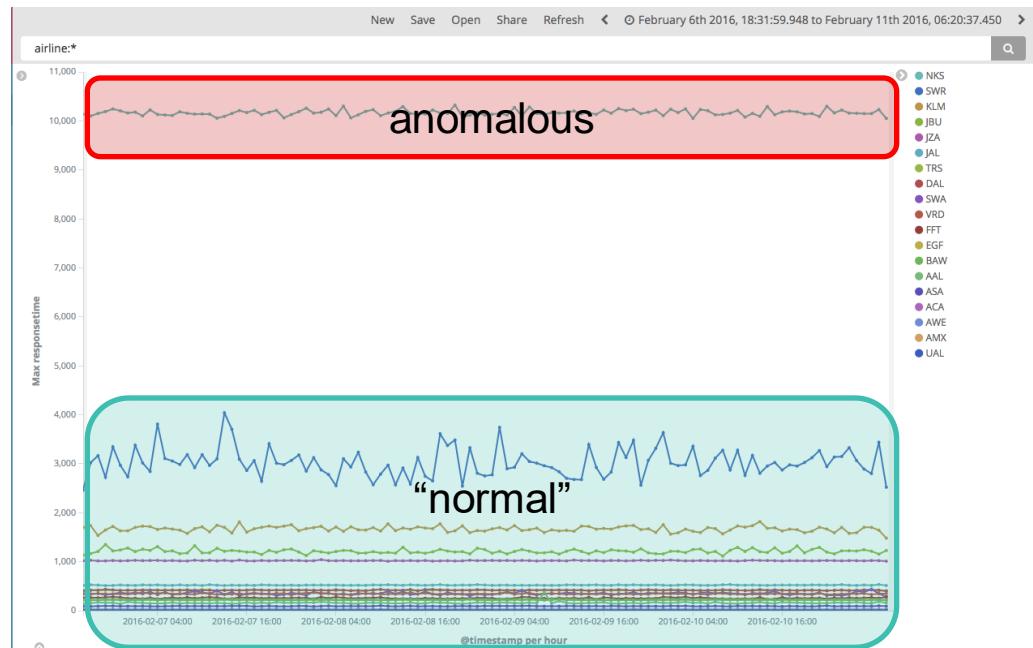
# What is “Abnormal”?

1) If something changes its behavior, compared to its own history – that change is *anomalous*.



# What is “Abnormal”?

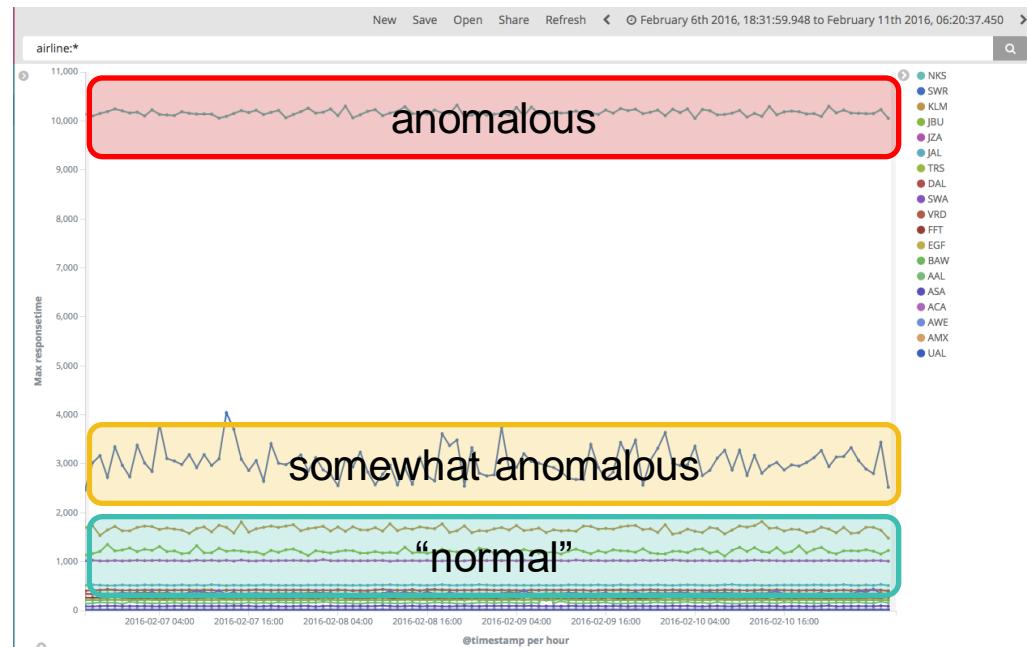
2) If something is drastically different than others within a population, then that entity is ***anomalous***.



# What is “Abnormal”?

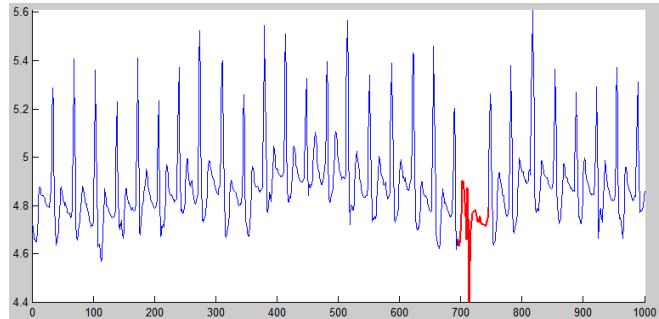
2) If something is drastically different than others within a population, then that entity is *anomalous*.

There's also the concept of being “somewhat anomalous”



# In Summary, Anomaloussness is:

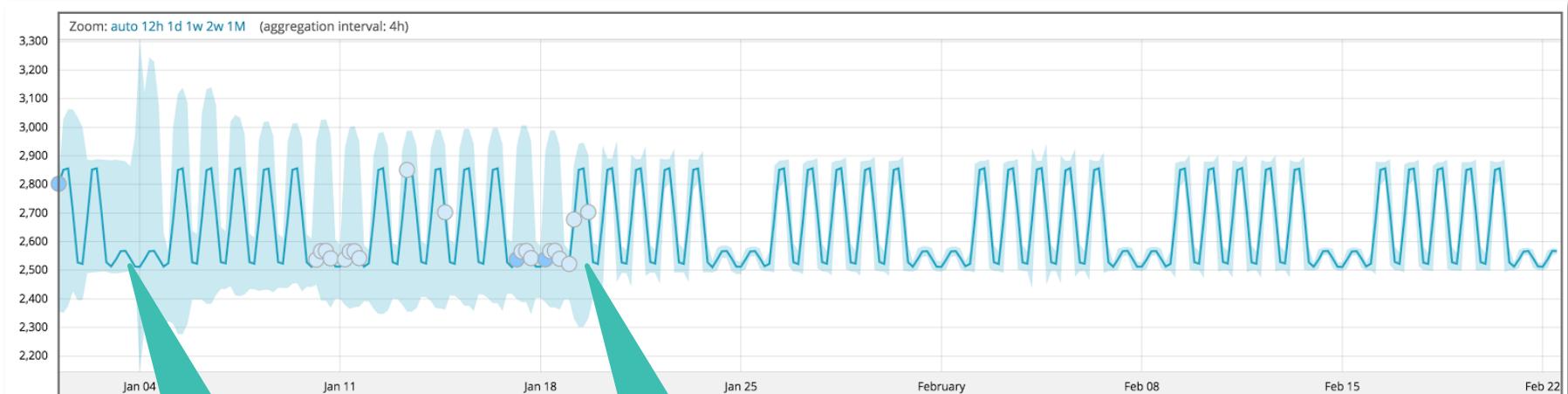
1) When an entities' ***behavior changes*** significantly and suddenly



2) When an entity is drastically ***different than others*** within a population



# Also, model needs to account for any periodicity



After 2 full days,  
daily periodicity  
has been learned.

After 2 full weeks,  
weekly periodicity  
has been learned.

# Elastic ML Framework

# 이상징후 탐지에 필수적인 16개의 알고리즘을 3 가지 방법론으로 구성해서 제안 – Elastic ML Framework

Important

X축 : Single Metric, Multi Metric Time series – 과거와 다른 행동패턴(by)

Y축 : Population Analysis – 다른 것들과 비교해서 다른 행동패턴(over)

Z축 : Rare/Unusual Analysis – 보기 드문 행동패턴(rare)

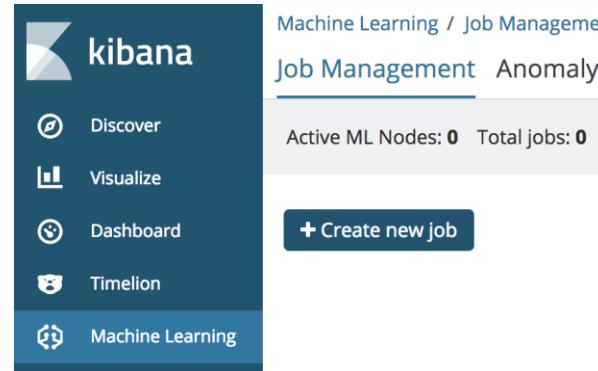
\* 몇 십년 경험을 가진 시스템 아키텍트/관리자 및 보안전문가의 노우하우(Know-How)를 시뮬레이션

# Lab 1: The Single Metric Job

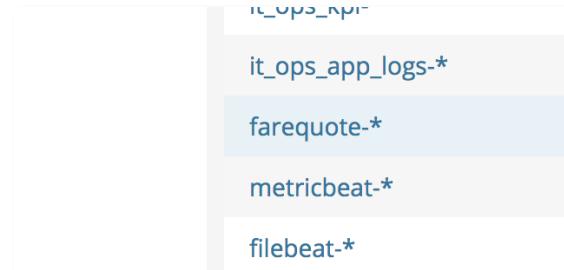
**Goal:** Detect anomalies in a single time series

# Steps to Complete

1) In Machine Learning,  
Create new job



2) Choose the “farequote-\*”  
index pattern



# Steps to Complete

3) pick the Single Metric Wizard

The screenshot shows the Kibana interface with a dark teal sidebar on the left containing icons and labels for various features: Discover, Visualize, Dashboard, Timelion, Canvas, Machine Learning (which is highlighted in blue), Infrastructure, Logs, and APM. To the right of the sidebar, the URL bar shows "Machine Learning / Job Management / Create New Job". The main content area has a light gray background with the title "Create a job from the i" partially visible. Below it, a section titled "Use a wizard" is shown with the sub-instruction "Use one of the wizards to create a machine learning j". There are two large rectangular boxes. The first box contains a green circle with a white plus sign and the text "Single metric Detect anomalies in a single time series.". The second box contains a blue circle with a white plus sign and the text "Multi metric Detect anomalies in multiple time series". At the bottom of the main content area, there is a link "Learn more about your data".

# Steps to Complete

4) Aggregation: count

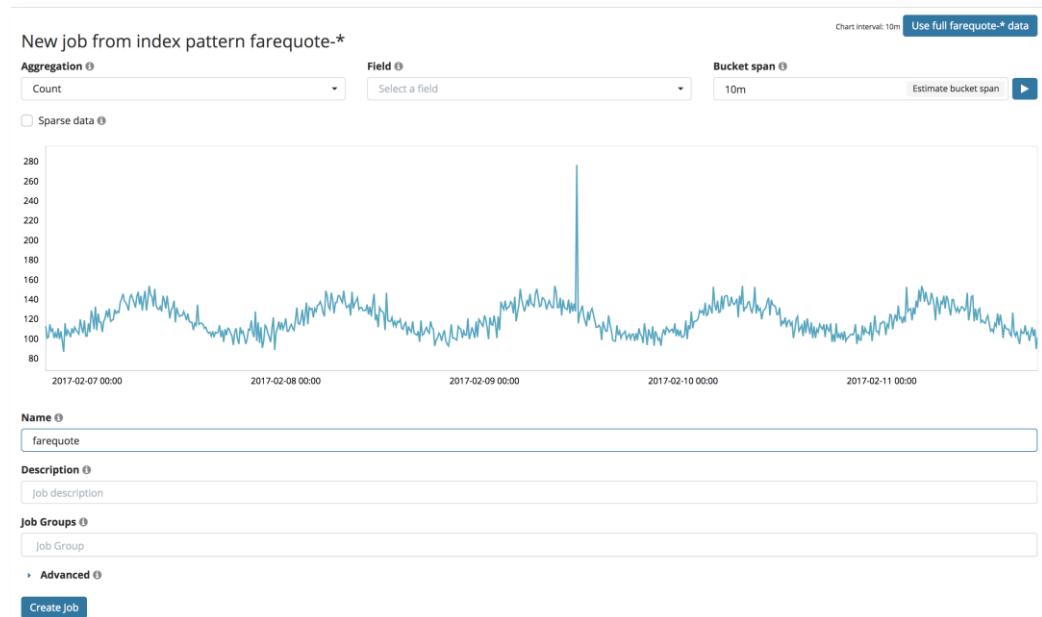
5) Field: <leave blank>

6) Bucket span: 10m

7) Click the “use full farequote data” button

8) Name: “farequote”

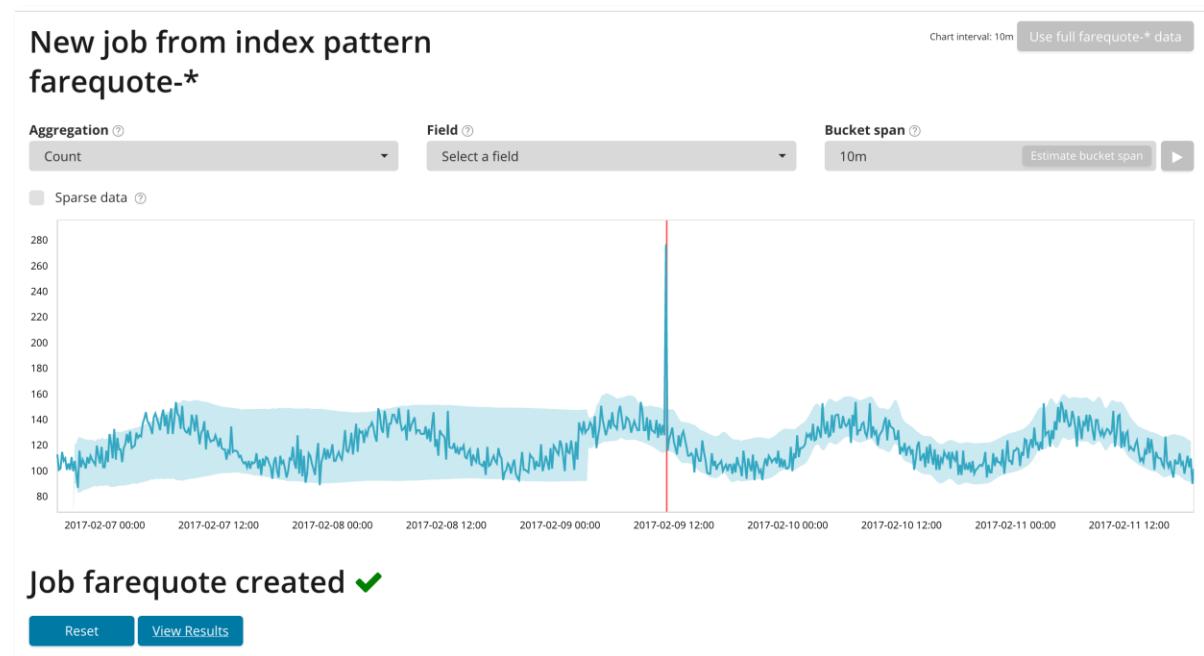
9) click “Create Job”



# Steps to Complete

10) See animated learning

11) click “View Results”



# Steps to Complete

12) Zoom in on anomaly



# Splitting the Analysis

# Feature Selection and Selection

- Which attributes of this data could be used to judge its unusualness?

## raw logs

```
2016/02/08 06:20:43 INFO [http-8680]: FareQuoteImpl - FareQuoteImpl.getFare(AAL): exiting: 92.5638  
2016/02/08 06:20:44 INFO [http-8680]: FareQuoteImpl - FareQuoteImpl.getFare(JZA): exiting: 990.4628  
2016/02/08 06:20:46 INFO [http-8680]: FareQuoteImpl - FareQuoteImpl.getFare(JBU): exiting: 877.5927  
...
```

document

```
{  
  "_index": "farequote",  
  "_type": "response",  
  "_id": "AVNQ1__XRcuRIYtw-jH",  
  "_score": 3.290889,  
  "_source": {  
    "sourcetype": "farequote",  
    "airline": "AAL",  
    "responsetime": "92.5638",  
    "time": "2016-02-08T06:20:43+0000"  
  }  
}
```



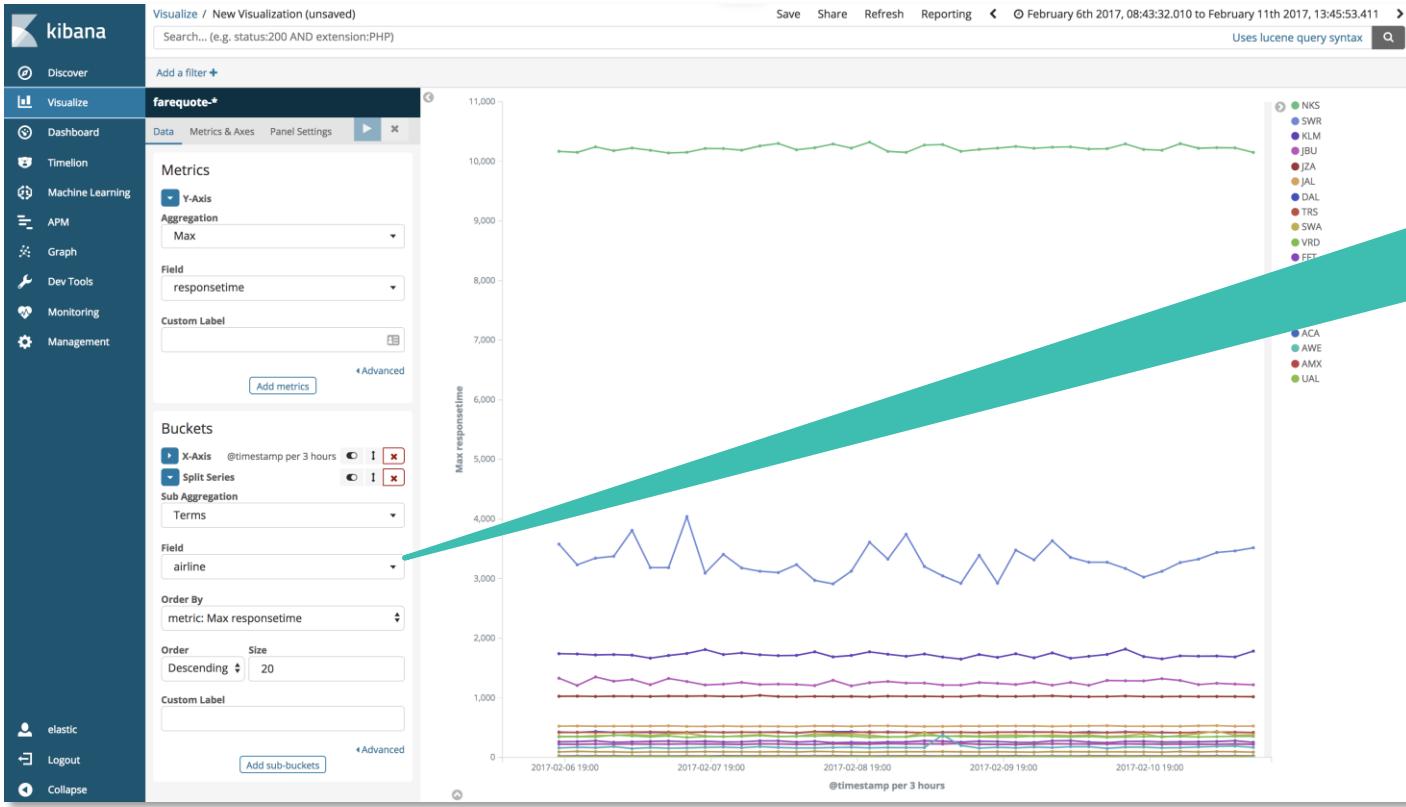
“airline”  
(categorical)

“responsetime”  
(metric)

# What Kinds of Questions Can be Answered?

QUESTION	ANSWERABLE?
Is there an unusual <b>amount</b> of requests per unit time (total)?	Yes
Is there any particular airlines with unusual <b>amounts</b> of requests per unit time?	Yes
Is the <b>total</b> response time of <b>all</b> API calls unusually long?	Yes
Is the response time of API calls <b>per airline</b> unusually long?	Yes
Are there any airlines with excessive take-off delays?	No

# In Kibana: How to Answer that Question



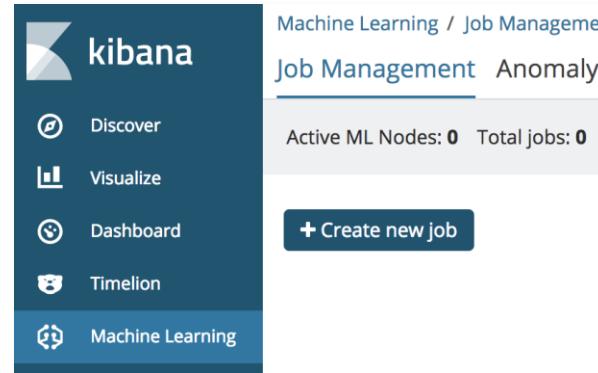
*ML has the same notion of “splitting” along a categorical field*

# Lab 2: Multi-Metric Jobs

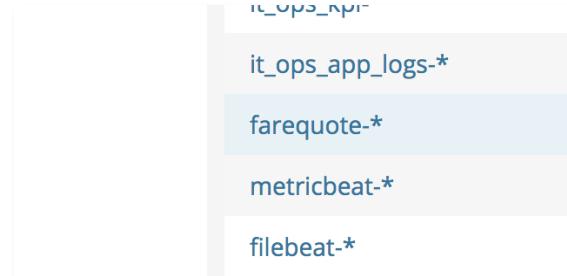
Goal: Detect anomalies in parallel time series

# Steps to Complete

1) In Machine Learning,  
Create new job



2) Choose the “farequote-\*”  
index pattern



# Steps to Complete

- 3) pick Multi metric Job Wizard



## Multi metric

Detect anomalies in multiple metrics by splitting a time series by a categorical field.

# Steps to Complete

4) choose

- event rate, count
- responsetime, max

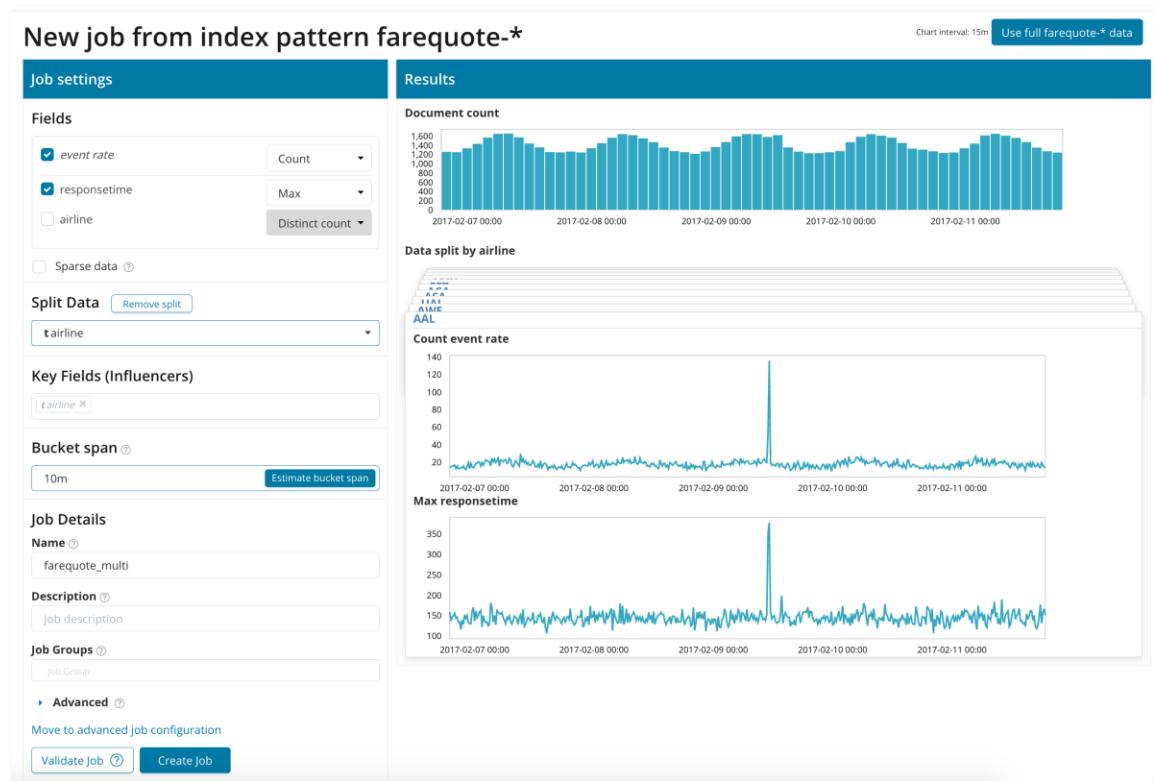
5) bucket span: 10m

6) Split Data: airline

7) click “use full farequote data”

8) Name: “farequote\_multi”

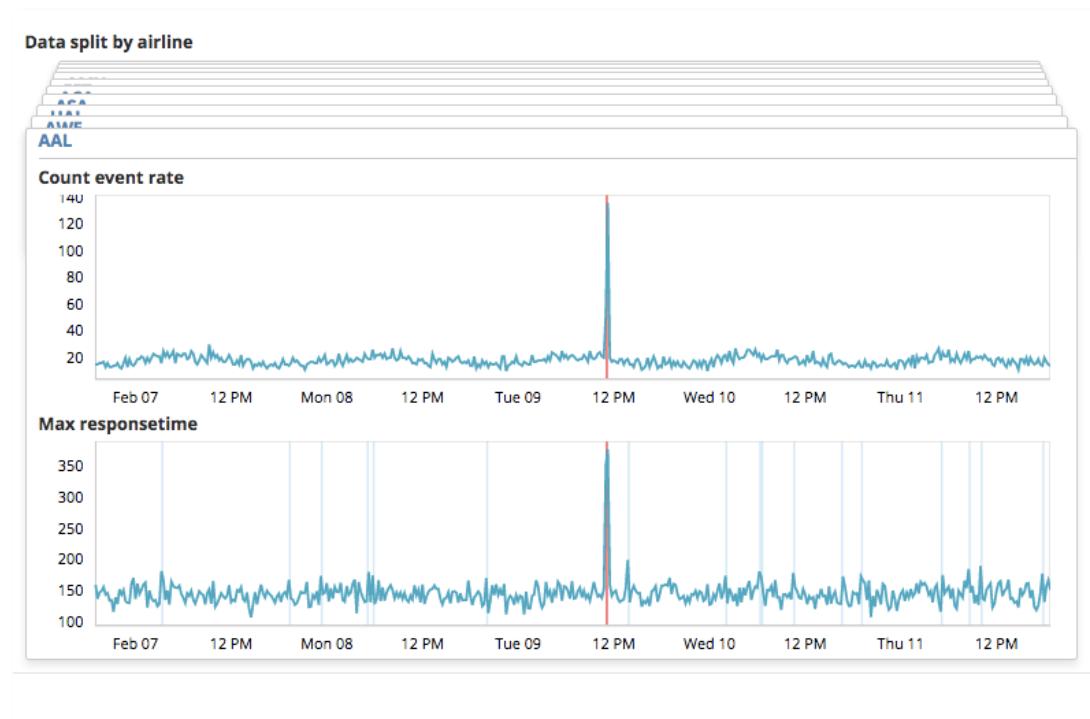
9) click “Create Job”



# Steps to Complete

10) See animated learning

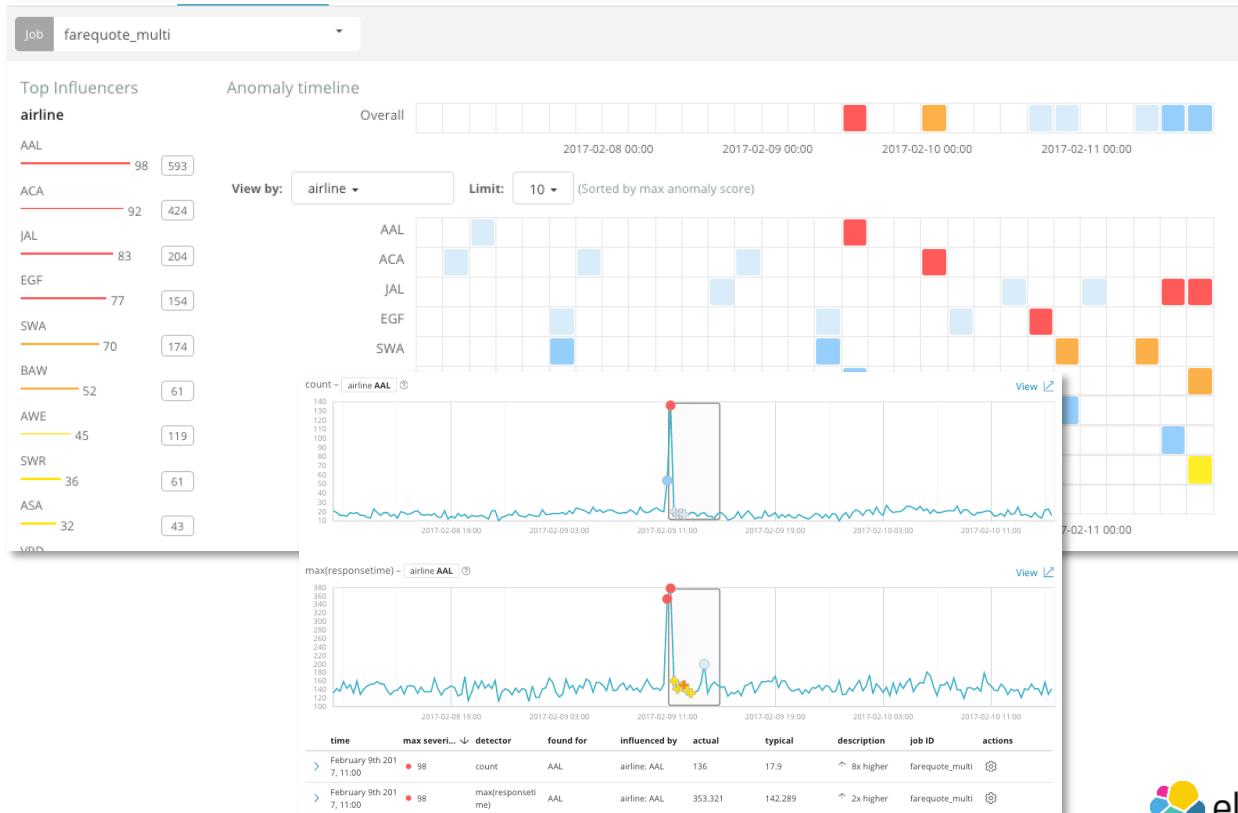
11) click “View Results”



# Steps to Complete

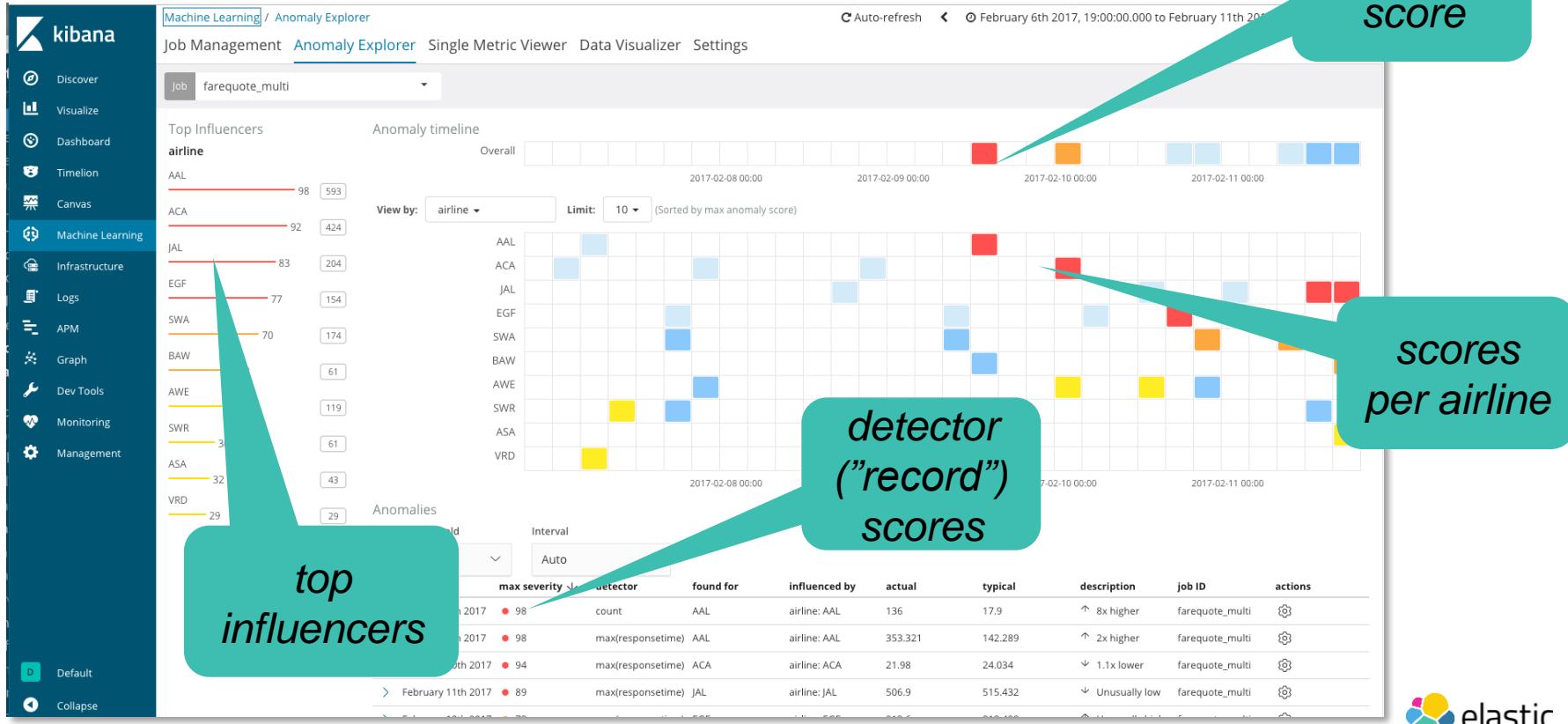
## 12) Result:

top unusual airline: AAL  
anomalies  
in both count  
and response time



# The Anomaly Explorer View

# Anomaly Explorer



# Concept: What is an Influencer?

- An Influencer is a field, selected at configuration time, that would be a logical entity "to blame" if an anomaly were to exist
- Doesn't have to be a field in the actual detector, but fields used to split the data are often good candidates
- Will get its own score based upon how influential that entity is on the anomaly

# Scoring

- Overall Job score is 90
  - How unusual is that bucket, given all airlines?
- Detector scores are 98
  - How unusual is the response time and count of airline=AAL?
- airline=AAL is the top influencer in this time range
  - 98 is the max score
  - 593 is the sum of all scores over the time range

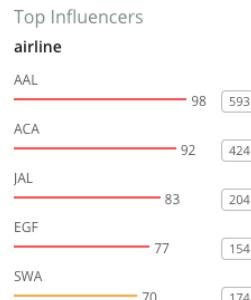
Anomaly timeline

Overall 2017-02-08 00:00 2017-02-09 00:00 2017-02-11 00:00

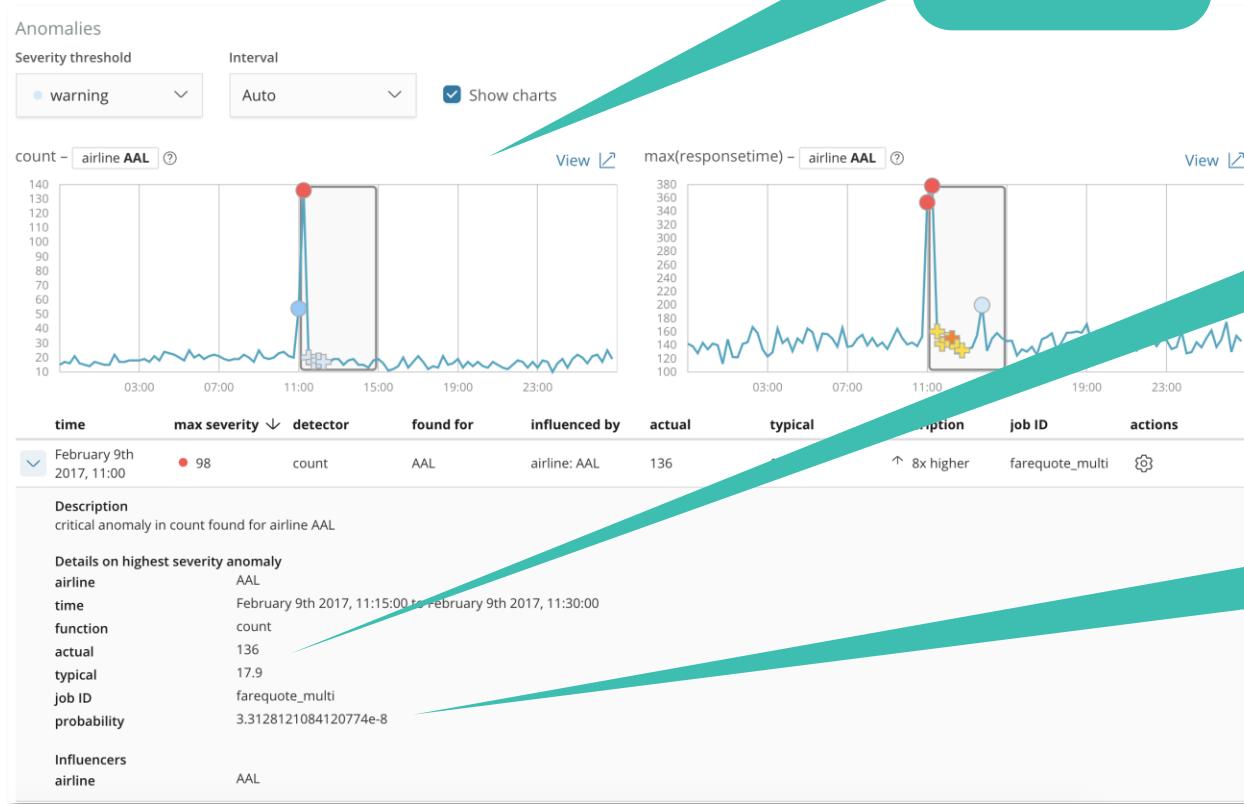
February 9th 2017, 11:00 Max anomaly score: 90

Anomalies

Severity threshold	Interval	time	max severity	detector	found for	influenced by	actual	typical	description	job ID	actions
warning	Auto	> February 9th 2017	98	count	AAL	airline: AAL	136	17.9	↑ 8x higher	farequote_multi	
		> February 9th 2017	98	max(responseTime)	AAL	airline: AAL	353.321	142.289	↑ 2x higher	farequote_multi	



# Anomaly Details



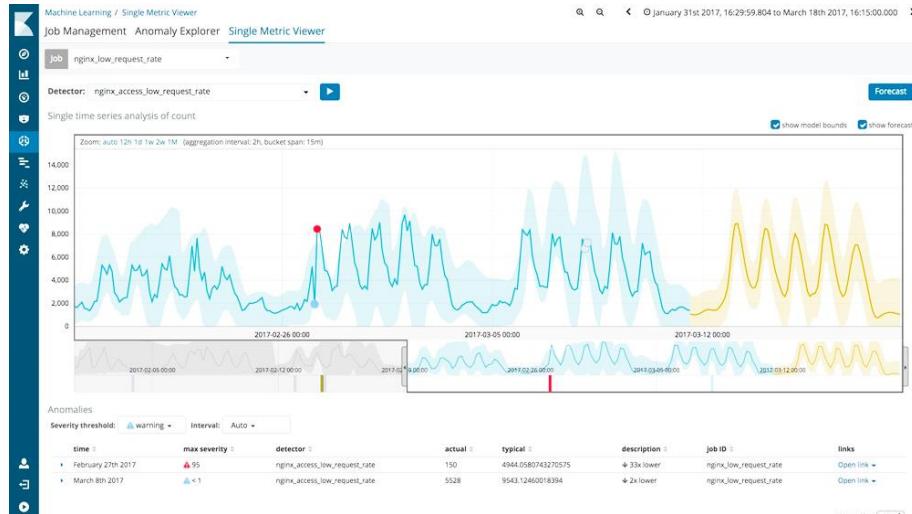
view individual anomalies

actual vs. “typical”

raw probability

# Forecasting

# Extrapolate into the future with forecasting



## Example use cases

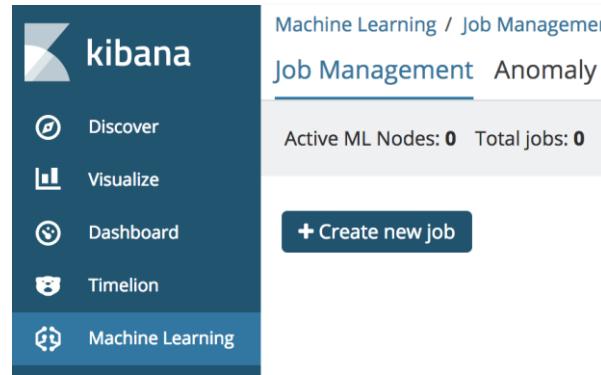
- I want to know when capacity will exceed 80%
- What is my expected visitor count next Saturday at 8am?

# Lab 3: Forecasting

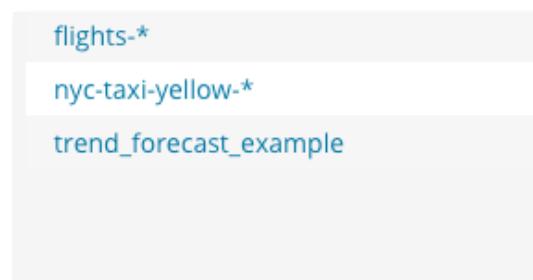
Goal: Extrapolate into the future

# Steps to Complete

1) In Machine Learning,  
Create new job



2) Choose the “trend\_forecast\_example”  
index pattern



# Steps to Complete

3) single metric job

4) sum of field “amount”

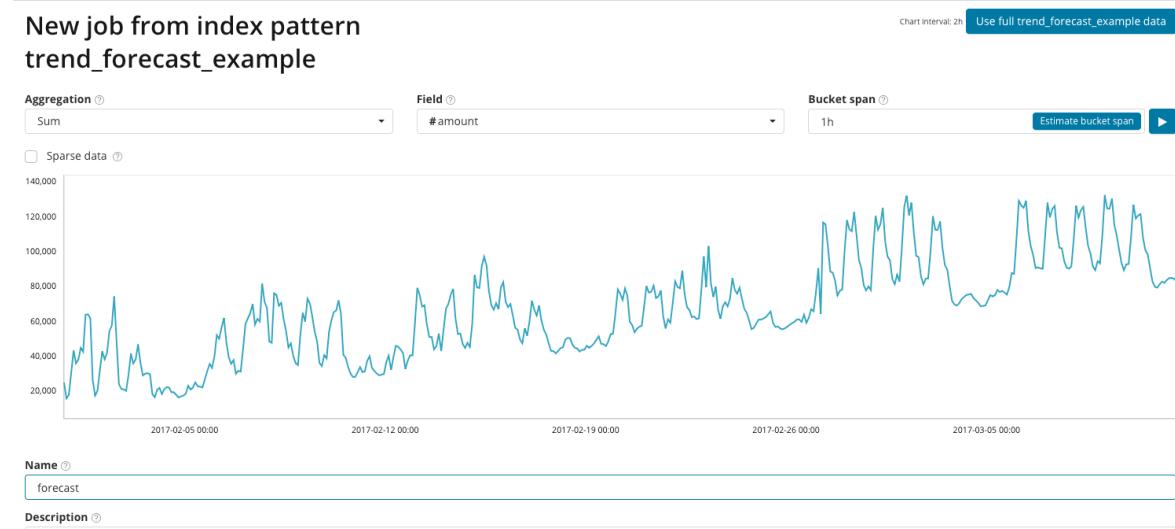
5) bucket span: 1h

6) click “use full data”

7) Name: “forecast”

8) click “Create Job”

9) click “View Results”



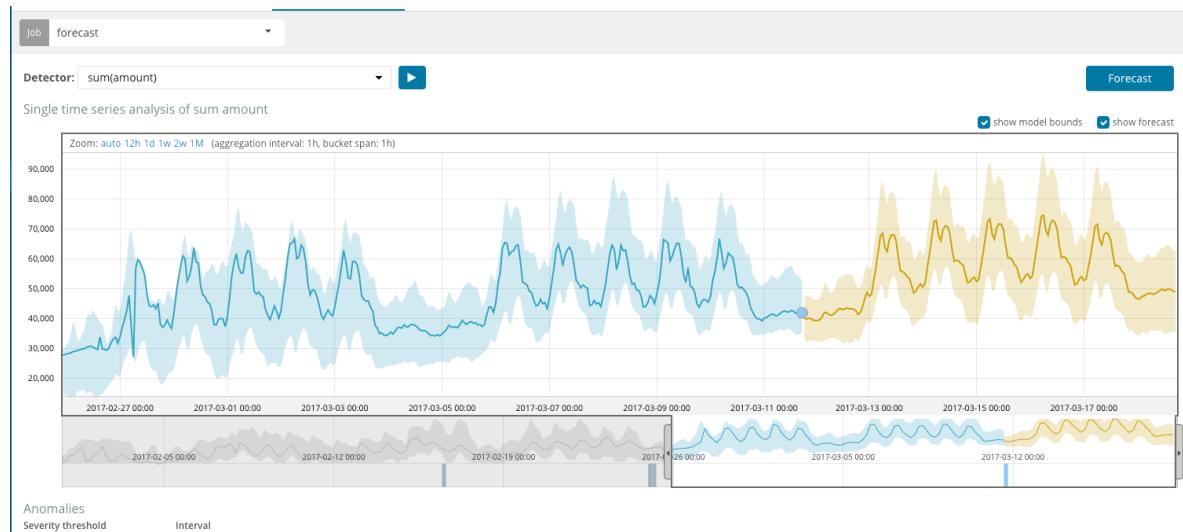
# Steps to Complete

10) click “Forecast” button

11) Enter “7d” for duration

12) click “Run” button

13) see forecast in UI  
(results also available via API)



# Population Analysis

# Population Analysis

- Use when:
  - Compare entities against peers
  - Not against own history
- Helpful when:
  - Entities have high-cardinality (i.e. external IP addresses)
  - Data for specific entities may be sparse in time (individual customers placing orders)
  - The behavior of the population as a whole is mostly homogeneous
- Not appropriate when:
  - Members of the population have vastly different behavior inherently.



## Population

Detect activity that is unusual compared to the behavior of the population.

# Population Analysis

*population field*

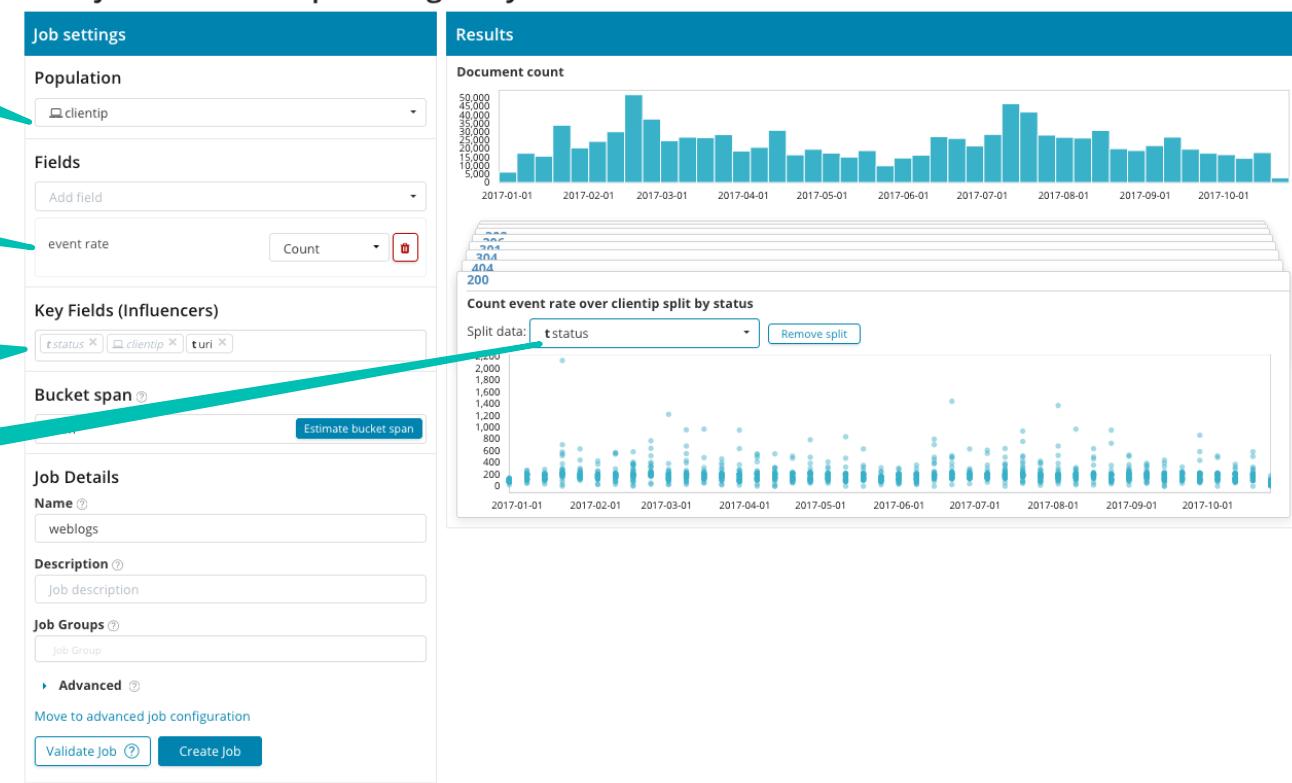
*the detector*

*influencers*

*optional split*

New job from index pattern gallery\*

Chart interval: 12h Use full gallery-\* data



# Population Analysis

clientip: 173.203.78.60  
status: 404  
uri: /wp-login.php



# Advanced Jobs

# Advanced Jobs

- Gives more flexibility
- Access to additional capabilities
- Requires more understanding of configuration parameters

Create a job from the index pattern gallery-\* 

**Use a wizard**

Use one of the wizards to create a machine learning job to find anomalies in your data.

 **Single metric**  
Detect anomalies in a single time series.

 **Multi metric**  
Detect anomalies in multiple metrics by splitting a time series by a categorical field.

 **Population**  
Detect activity that is unusual compared to the behavior of the population.

 **Advanced**  
Use the full range of options to create a job for more advanced use cases.

**Learn more about your data**

If you're not sure what type of job to create, first explore the fields and metrics in your data.

 **Data Visualizer**  
Learn more about the characteristics of your data and identify the fields for analysis with machine learning.

# Configuration of Advanced Job

*Tabs of different information*

### Create a new job

Job Details    Analysis Configuration    Datafeed    Edit JSON    Data Preview

**Name** ?  
farequote\_response

**Description** ?  
Job description

**Job Groups** ?  
Job Group

**Custom URLs** ?  
+ Add Custom URL

Use dedicated index ?

**Model memory limit** ?  
1gb

[Validate Job](#) ?   [Save](#)   [Cancel](#)

# Analysis Configuration

*Setting of  
bucket\_span*

### Create a new job

Job Details   Analysis Configuration   Datafeed   Edit JSON   Data Preview

**bucket\_span** ⓘ  
15m

**summary\_count\_field\_name** ⓘ  
Select...

**categorization\_field\_name** ⓘ  
Select...

**Detectors** ⓘ

+ Add Detector

**Influencers** ⓘ

- action
- clientip
- file
- method

*Creation of  
“detector”*

### Add new detector

**Description** ⓘ  
max(responsetime) partition\_field\_name=airline

<b>function</b> ⓘ	<b>field_name</b> ⓘ	<b>by_field_name</b> ⓘ
max	responsetime	Select...
<b>over_field_name</b> ⓘ	<b>partition_field_name</b> ⓘ	<b>exclude_frequent</b> ⓘ
Select...	airline	Select...

Help for max ⓘ

Add   Cancel

# Detector Details

- **function** – min, max, mean, etc.
- **field\_name** – field function “operates on”
- **by\_field\_name** – “dependent” split
- **partition\_field\_name** – “independent” split
- **over\_field\_name** – defines a population

Add new detector

Description ?

max(responsetime) partition\_field\_name=airline

function <small>?</small>	field_name <small>?</small>	by_field_name <small>?</small>
max	responsetime	Select...
over_field_name <small>?</small>	partition_field_name <small>?</small>	exclude_frequent <small>?</small>
Select...	airline	Select...

Help for max ↗

Add Cancel

# Datafeed Details

- Datafeed manages how ML queries Elasticsearch
- See ML docs for configuration options

### Create a new job

Job Details   Analysis Configuration   **Datafeed**   Edit JSON   Data Preview

Datafeed job ⓘ

**Query** ⓘ  
{"match\_all":{}}

**Query delay** ⓘ  
60s

**Frequency** ⓘ  
450s

**scroll\_size** ⓘ  
1000

**Index**  
gallery-\*

**Time-field name**  
@timestamp

**Buttons:** Validate Job ⓘ   Save   Cancel

# Rarity Analysis

# Rare Analysis

- Finding items that rarely occur is also often useful
  - Rarely occurring log messages
  - Rare running process names
  - Rare connection destinations
- ML has a rare function, but it should be noted that:
  - It is relative, i.e. it takes into account the frequency of other field values

A,B,A,B,A,C,B,A,B,A,C,A,B,C,A,C,B,A,C,B,A,C,X <=X is rare

A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X <= X is not rare

- Therefore it works best when there are plenty of routine messages to contrast the rare ones

# Example of Rare Analysis

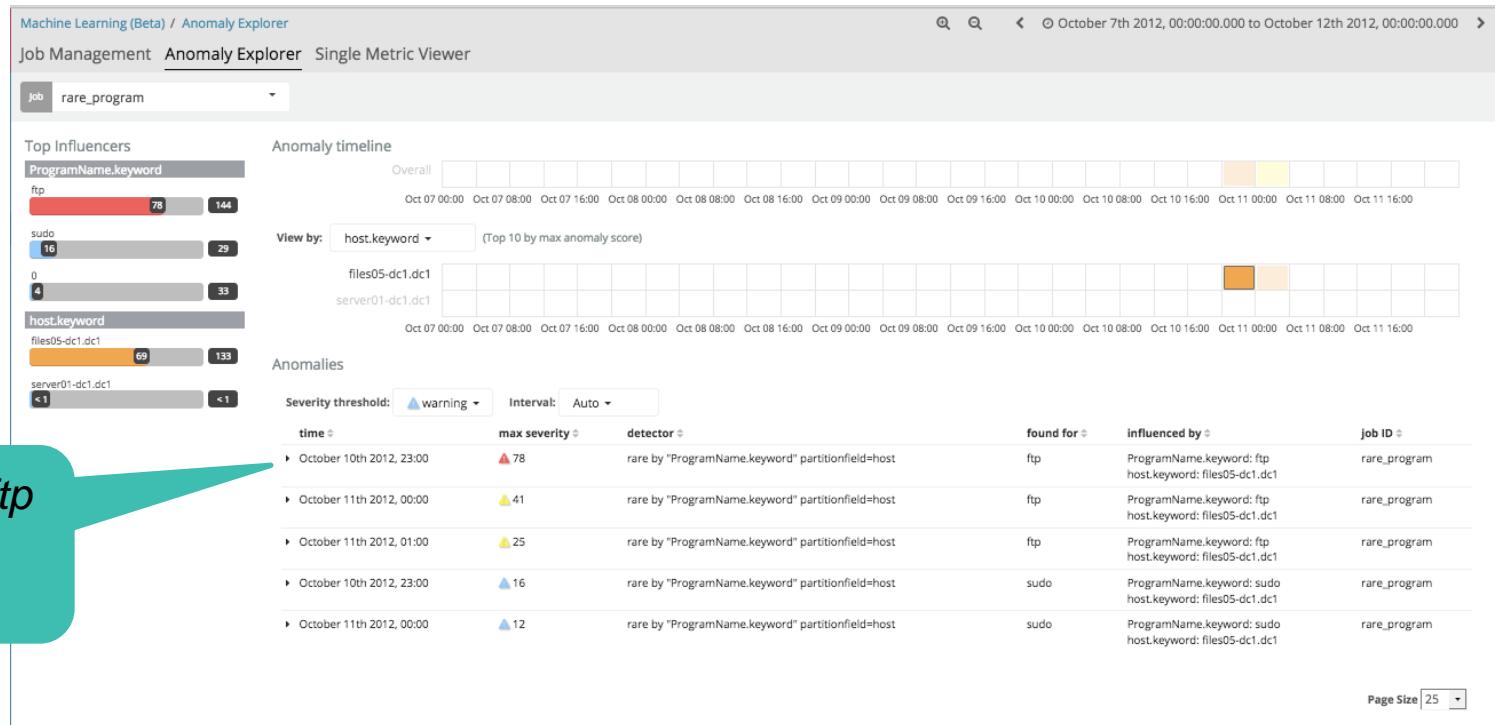
- Use Case: Security team @ services company
- Wanted to profile typical processes on each host using netstat

```
Active Internet connections (servers and established)
(index=netstat host="ids01-dc2" State=LISTEN (7/16/13 1:32:51AM))
Proto Recv-Q Send-Q Local Address          Foreign Address        State      PID/Program name
tcp      0      0  10.220.174.41:561        0.0.0.0:*
tcp      0      0  0.0.0.0:22             0.0.0.0:*
tcp      0      0  0.0.0.0:1241           0.0.0.0:*
tcp      0      0  0.0.0.0:8089           0.0.0.0:*
tcp      0      0  127.0.0.1:25            0.0.0.0:*
tcp      0      0  0.0.0.0:8000           0.0.0.0:*
tcp      0      0  0.0.0.0:8834           0.0.0.0:*
tcp      0      0  127.0.0.1:199          0.0.0.0:*
tcp      0      0  10.220.174.41:56002    10.220.174.40:3204  ESTABLISHED 4055/bashscriptd
```

- Goal was to identify rare processes that “start up and communicate” for each host, individually

# Example of Rare Analysis

detector: *rare by ProgramName partition\_field=host*



ProgramName: ftp  
host: files05-dc1.dc1  
rarely occurs

# ML Categorization

# Categorization

- What is it?
  - Automatically grouping similar log messages into the same “category”
- How?
  - Via an algorithm that looks at string similarity and clusters similar log lines
- Why?
  - Bring structure to semi-structured data so that it can be analyzed

Benefit: Find anomalies in logs without knowing what they contain

# Example

```
Oct 22 18:17:58 localhost sshd[8903]: Invalid user admin from 41.43.112.199 port 41805
Oct 22 18:17:58 localhost sshd[8903]: input_userauth_request: invalid user admin [preauth]
Oct 22 18:17:59 localhost sshd[8903]: Connection closed by 41.43.112.199 port 41805 [preauth]
Oct 22 20:58:03 localhost sshd[2074]: Received disconnect from 72.93.85.203 port 53552:11: disconnected by user
Oct 22 20:58:03 localhost sshd[2074]: Disconnected from 72.93.85.203 port 53552
Oct 22 20:58:03 localhost sshd[2072]: pam_unix(sshd:session): session closed for user ec2-user
Oct 22 21:32:54 localhost sshd[8944]: pam_unix(sshd:session): session opened for user ec2-user by (uid=0)
Oct 22 21:35:15 localhost runuser: pam_unix(runuser-l:session): session closed for user ec2-user
Oct 22 21:35:15 localhost runuser: pam_unix(runuser-l:session): session opened for user ec2-user by ec2-user(uid=0)
Oct 22 21:35:16 localhost runuser: pam_unix(runuser-l:session): session closed for user ec2-user
```

# Step 1 – remove mutable text

```
Oct 22 18:17:58 localhost sshd[8903]: Invalid user admin from 41.43.112.199 port 41805
Oct 22 18:17:58 localhost sshd[8903]: input_userauth_request: invalid user admin [preauth]
Oct 22 18:17:59 localhost sshd[8903]: Connection closed by 41.43.112.199 port 41805 [preauth]
Oct 22 20:58:03 localhost sshd[2074]: Received disconnect from 72.93.85.203 port 53552:11: disconnected by user
Oct 22 20:58:03 localhost sshd[2074]: Disconnected from 72.93.85.203 port 53552
Oct 22 20:58:03 localhost sshd[2072]: pam_unix(sshd:session): session closed for user ec2-user
Oct 22 21:32:54 localhost sshd[8944]: pam_unix(sshd:session): session opened for user ec2-user by (uid=0)
Oct 22 21:35:15 localhost runuser: pam_unix(runuser-l:session): session closed for user ec2-user
Oct 22 21:35:15 localhost runuser: pam_unix(runuser-l:session): session opened for user ec2-user by ec2-user(uid=0)
Oct 22 21:35:16 localhost runuser: pam_unix(runuser-l:session): session closed for user ec2-user
```

# Step 2 – cluster similar messages together

Oct 22 18:17:58 localhost sshd[8903]: Invalid user admin from 41.43.112.199 port 41805

Oct 22 18:17:58 localhost sshd[8903]: input\_userauth\_request: invalid user admin [preauth]

Oct 22 18:17:59 localhost sshd[8903]: Connection closed by 41.43.112.199 port 41805 [preauth]

Oct 22 20:58:03 localhost sshd[2074]: Received disconnect from 72.93.85.203 port 53552:11: disconnected by user

Oct 22 20:58:03 localhost sshd[2074]: Disconnected from 72.93.85.203 port 53552

Oct 22 20:58:03 localhost sshd[2072]: pam\_unix(sshd session): session closed for user ec2-user

Oct 22 21:32:54 localhost sshd[8944]: pam\_unix(sshd session): session opened for user ec2-user by (uid=0)

Oct 22 21:35:15 localhost runuser: pam\_unix(runuser-l session): session closed for user ec2-user

Oct 22 21:35:15 localhost runuser: pam\_unix(runuser-l session): session opened for user ec2-user by ec2-user(uid=0)

Oct 22 21:35:16 localhost runuser: pam\_unix(runuser-l session): session closed for user ec2-user

## Step 2 – cluster similar messages together

Oct 22 18:17:58 localhost sshd[8903]: Invalid user admin from 41.43.112.199 port 41805

mlcategory:1

Oct 22 18:17:58 localhost sshd[8903]: input\_userauth\_request: invalid user admin [preauth]

mlcategory:2

Oct 22 18:17:59 localhost sshd[8903]: Connection closed by 41.43.112.199 port 41805 [preauth]

mlcategory:3

Oct 22 20:58:03 localhost sshd[2074]: Received disconnect from 72.93.85.203 port 53552:11: disconnected by user

mlcategory:4

Oct 22 20:58:03 localhost sshd[2074]: Disconnected from 72.93.85.203 port 53552

mlcategory:5

mlcategory:6

Oct 22 20:58:03 localhost sshd[2072]: pam\_unix(sshd:session): session closed for user ec2-user

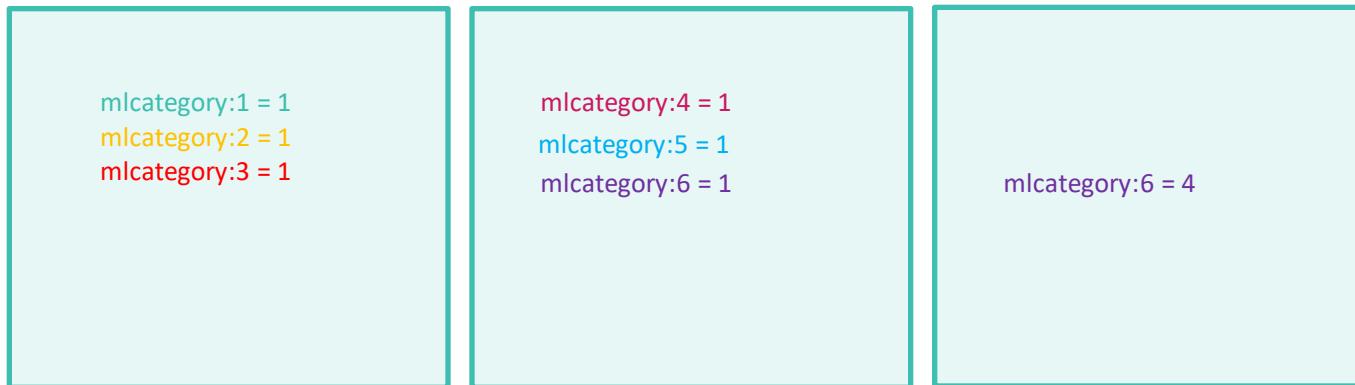
Oct 22 21:32:54 localhost sshd[8944]: pam\_unix(sshd:session): session opened for user ec2-user by (uid=0)

Oct 22 21:35:15 localhost runuser: pam\_unix(runuser-l:session): session closed for user ec2-user

Oct 22 21:35:15 localhost runuser: pam\_unix(runuser-l:session): session opened for user ec2-user by ec2-user(uid=0)

Oct 22 21:35:16 localhost runuser: pam\_unix(runuser-l:session): session closed for user ec2-user

## Step 3 – count per time bucket



time

# Categorization Example

- Configure an ML job to use:
  - “message” as the `categorization_field_name`
  - there will be a new, “magic” field called “`mlcategory`” that is dynamically created by ML to group similar messages together

The screenshot shows the 'Categorization' section of the Elasticsearch Machine Learning Settings. It includes fields for 'bucket\_span' (set to 10m), 'summary\_count\_field\_name' (empty), 'categorization\_field\_name' (set to 'message'), and a 'Categorization Filters' section with a '+ Add Categorization Filter' button. Below this is a 'Detectors' section containing a single entry: 'Unusual message counts count by mlcategory' with edit and delete icons, and a '+ Add Detector' button.

bucket\_span ⓘ  
10m

summary\_count\_field\_name ⓘ

categorization\_field\_name ⓘ  
message

Categorization Filters ⓘ  
+ Add Categorization Filter

Detectors ⓘ

Unusual message counts  
count by `mlcategory`

+ Add Detector

# Categorization Example – count by mlcategory

Anomaly timeline



View by: job ID (Top 10 by max anomaly score)



Anomalies

Severity threshold:	warning	Interval:	Auto	time	max severity	detector	found for	actual	typical	description	job ID	links	category examples
►	February 8th 2016, 10:00	▲ 66	count by mlcategory	mlcategory 11	49	0.0820658	▲ More than 100x higher	logs		Open link ↴ Fall To Connect Database ReActivate Application / Client Connection !			
►	February 8th 2016, 10:00	▲ 66	count by mlcategory	mlcategory 10	49	0.0820658	▲ More than 100x higher	logs		Open link ↴ DBMS ERROR : db=10.16.1.63!svc_prod#uid=dbadmin1;pwd=##### ! DBMS ERROR : db=svc_prod Err=-17 [Microsoft][ODBC SQL Server Driv			
►	February 8th 2016, 10:00	▲ 43	count by mlcategory	mlcategory 9	1	0.00336345	▲ More than 100x higher	logs		Open link ↴ DB Not Updated [Master] Table;dbhost=dbserver.acme.com;physical			
►	February 8th 2016, 05:00	▲ 16	count by mlcategory	mlcategory 6	1	0.00502013	▲ More than 100x higher	logs		Open link ↴ Transaction Match In DB / Duplicate Transaction;dbhost=dbserver.ac			
►	February 8th 2016, 09:00	▲ 8	count by mlcategory	mlcategory 6	1	0.00657718	▲ More than 100x higher	logs		Open link ↴ Transaction Match In DB / Duplicate Transaction;dbhost=dbserver.ac			
►	February 8th 2016, 10:00	▲ 4	count by mlcategory	mlcategory 2	49	0.081315	▲ More than 100x higher	logs		Open link ↴ REC Not INSERTED [DB TRAN] Table;dbhost=dbserver.acme.com;physi			
►	February 8th 2016, 10:00	▲ 2	count by mlcategory	mlcategory 5	7	0.0600863	▲ More than 100x higher	logs		Open link ↴ Opening Database = DRIVER={SQL Server};SERVER=10.16.1.63!netwo			
►	February 8th 2016, 10:00	▲ 2	count by mlcategory	mlcategory 3	1	0.0128763	▲ 78x higher	logs		Opening Database = DRIVER={SQL Server};SERVER=127.0.0.1;network			
►	February 8th 2016, 06:00	▲ 2	count by mlcategory	mlcategory 7	1	0.013673	▲ 73x higher	logs		Opening Database = DRIVER={SQL Server};SERVER=sssvcdbj1.acme.com!svc_prod#uid=dbadmin1;pwd=#####;db			
►	February 8th 2016, 10:00	▲ 1	count by mlcategory	mlcategory 4	2	0.0202842	▲ 99x higher	logs		Using: 10.16.1.63!svc_prod#uid=dbadmin1;pwd=#####;dbhost=dbserver.acme.com			
►										Using: sssvcdbj1.acme.com!svc_prod#uid=dbadmin1;pwd=#####;db			
►										Actual Transaction Not Found In DB To VOID;dbhost=dbserver.acme.com			
►										012 Head Office Link Active 127.0.0.1;dbhost=dbserver.acme.com;ph			

category name

example matching log messages



# Lab 4, Track 1: Root Cause Analysis

# Lab 4: Track 1 - Problem

- Transaction processing app records # transactions per minute in an index called “it\_ops\_kpi-\*”. Find the anomaly in it.
- Also, SQL Database and Network performance metrics were gathered for the application environment (it\_ops\_network-\* and it\_ops\_sql-\*). Use multi-metric jobs to see what might be going on in there.
- Lastly, there’s an index (it\_ops\_app\_logs-\*) that has the application’s log lines in them. See if there’s anything anomalous in there as well.
- View all jobs correlated together in the Anomaly Explorer

# Lab 4: Track 1 - Solution

- Create and Advanced job for:
  - “it\_ops\_app\_logs-\*”
  - Create a “count by mlcategory” job for the log events
    - use “message” as the categorization\_field\_name
- Create Multi-Metric jobs for the following indices:
  - it\_ops\_sql-\*
  - it\_ops\_network-\*
- Create a “low\_sum” Single-Metric job for the following :
  - it\_ops\_kpi-\*
- Group all jobs into the same Job Group name
- View all jobs overlaid in the Explorer View

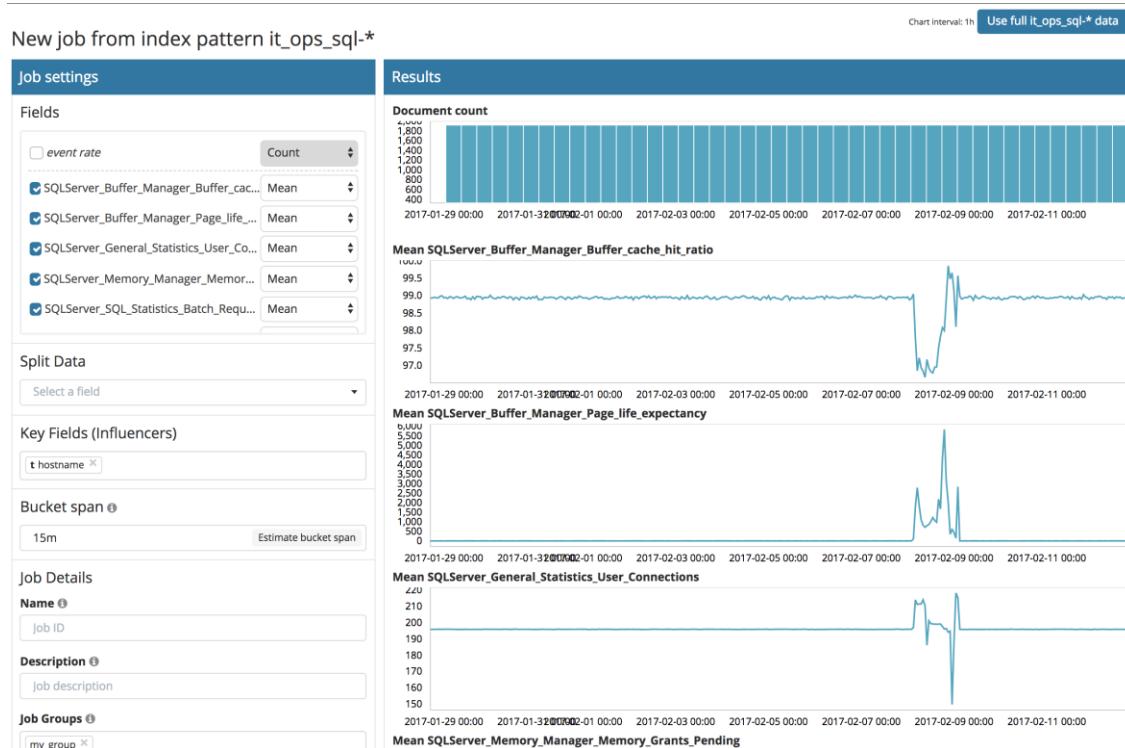
# Lab 4: Track 1 - Solution

- For index:it\_ops\_app\_logs
  - create an advanced job
  - put job in a group
- make sure you choose “message” for categorization\_field\_name
- detector is: count with by\_field\_name of “mlcategory”

The screenshot shows the Elasticsearch Machine Learning Job Management interface. A modal window titled "Add new detector" is open, prompting for configuration details. The "Description" field contains "count by mlcategory". The "function" field is set to "count". The "field\_name" dropdown is set to "Select...". The "by\_field\_name" dropdown is set to "mlcategory". The "over\_field\_name" dropdown is set to "Select...". The "partition\_field\_name" dropdown is set to "Select...". The "exclude\_frequent" dropdown is set to "Select...". At the bottom of the modal are "Add" and "Cancel" buttons. In the background, the main interface shows a "Create a new job" form with fields for "Job Details", "bucket\_span" (set to 15m), "summary\_count" (set to Select...), and "categorization\_field\_name" (set to message). Below this, there's a "Categorization Fields" section with a "+ Add Categorization Field" button. At the bottom, there's a "Detectors" section with a "+ Add Detector" button.

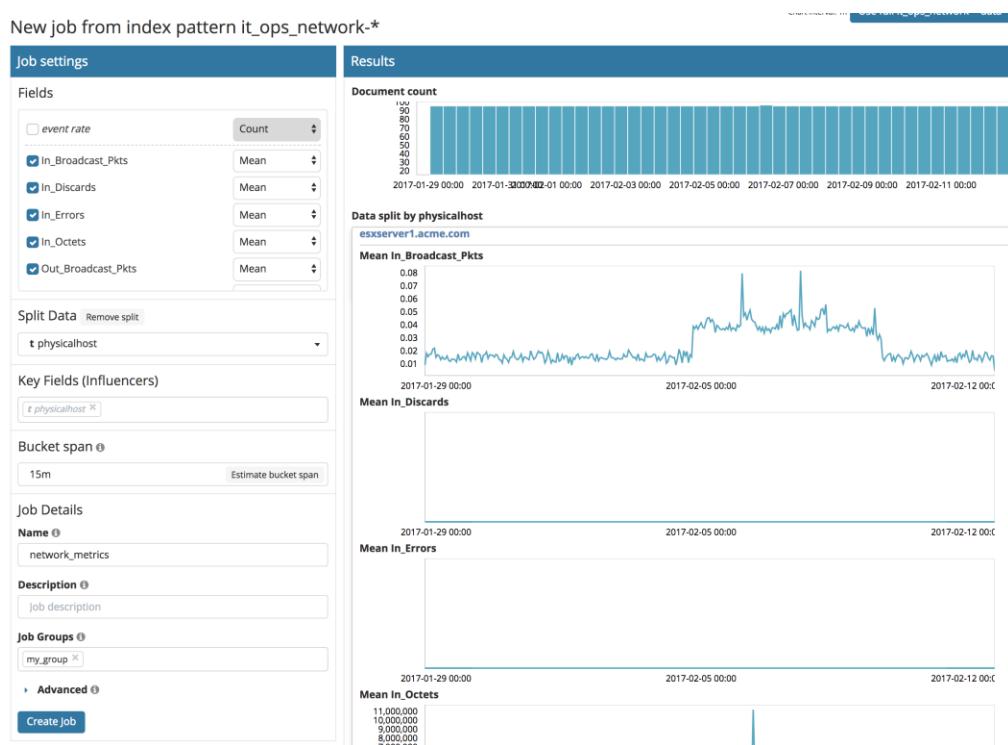
# Lab 4: Track 1 - Solution

- For index:it\_ops\_sql-\*
  - create multi-metric job
  - “mean” for all SQL metrics
- Also put job in same Job Group as the app logs job



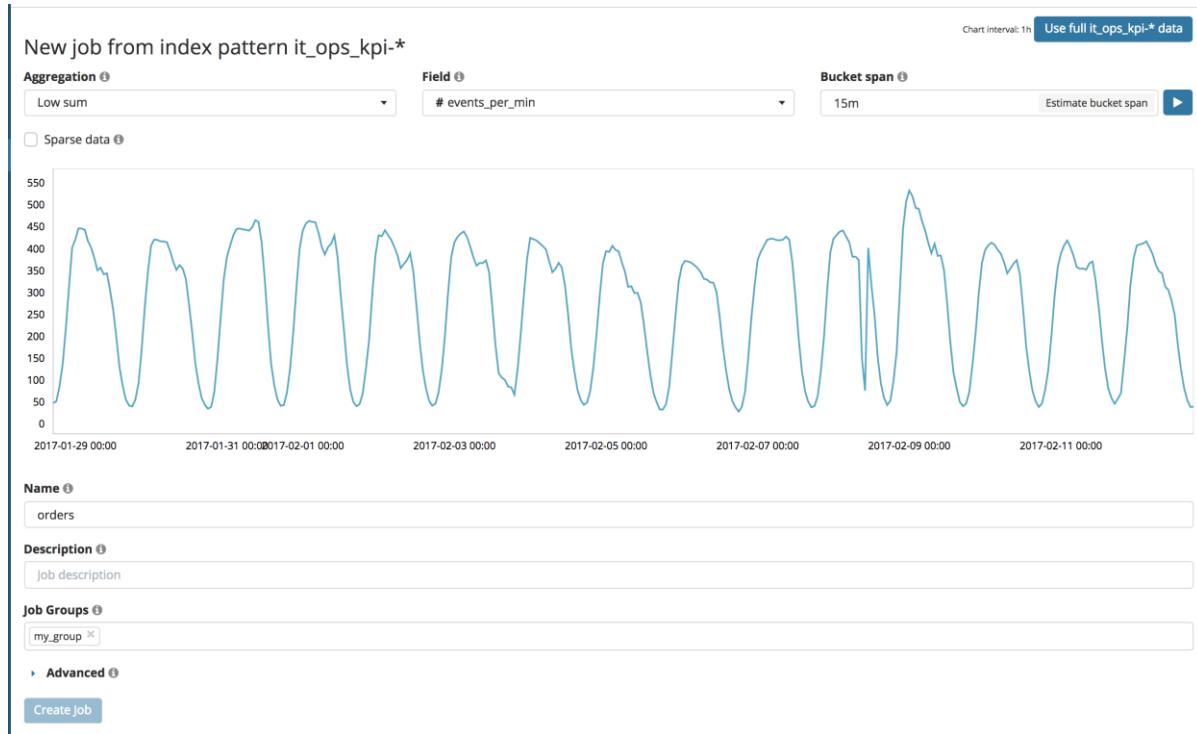
# Lab 4: Track 1 - Solution

- For index:it\_ops\_network-\*
  - create multi-metric job
  - “mean” for all metrics
- Also put job in same Job Group as the app logs job



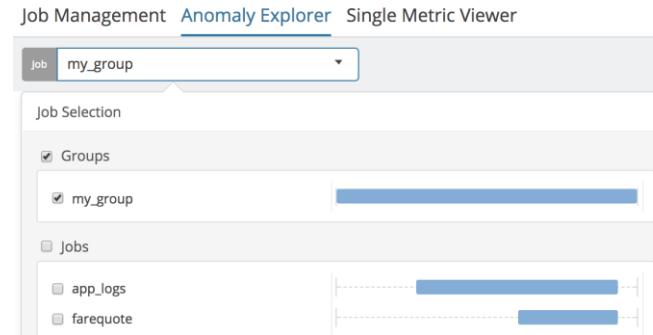
# Lab 4: Track 1 - Solution

- For index:it\_ops\_kpi-\*
  - create single-metric job
  - “low\_sum” for field “events\_per\_min”
- Also put job in same Job Group as the app logs job

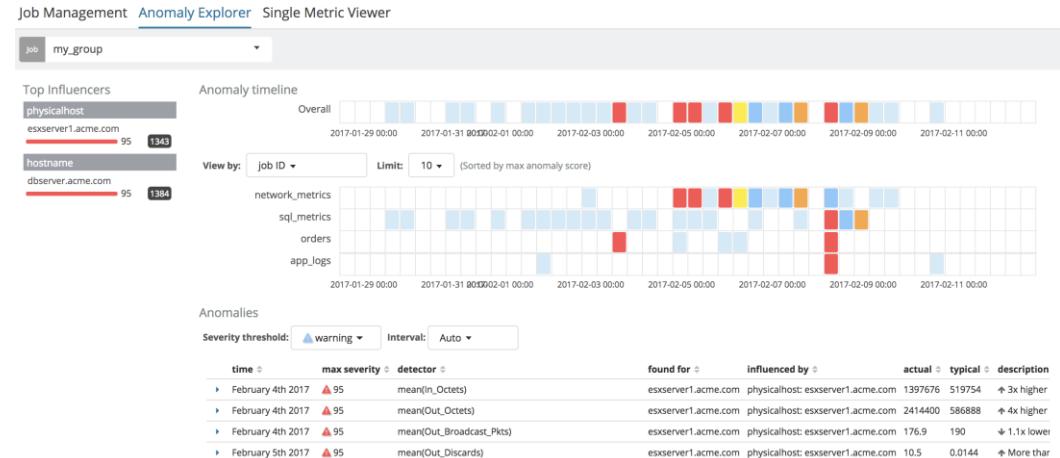


# Lab 4: Track 1 - Solution

- View all jobs in Group



- See correlated anomalies



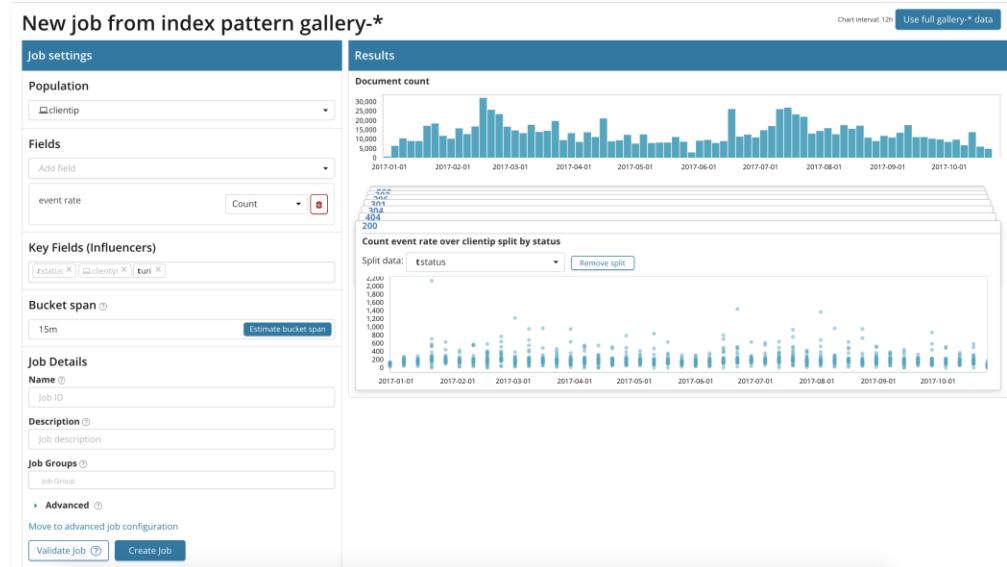
# Lab 4, Track 2: Analyze Web Logs

## Lab 4: Track 2 - Problem

- There's been suspicion that automated bots are slamming the web site from time to time.
- Use a population analysis job to find any rogue IPs and see what URLs were being requested
- Data is in gallery-\* index

# Lab 4: Track 2 - Solution

- Create and population job for:
  - “gallery-\*”
  - count by status over clientIP
  - uri as additional Influencer



# Lab 4: Track 2 - Solution

- See rogue IP 173.203.78.60
- Thousands of requests for URI: wp-login.php
- This is a low-intelligence brute force attack
- Fortunately, that web page doesn't exists on webserver, hence 404 status code



# ML Recipes

# Complex Detector Example

- **high\_count by ErrorCode over Host partition\_field=App**
  - Partition: App
    - Split the data into separate data sets for each App. Create a model (and population) for each.
  - Over: Host
    - For each App, create a population model out of all the Hosts and detect when a Host stands out.
  - By: ErrorCode
    - When modeling and evaluating Hosts, baseline each ErrorCode separately.
  - Function: High\_Count
    - Measure the number of times an ErrorCode is observed in the time bucket.

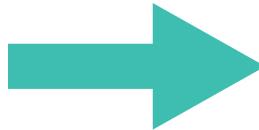
<https://www.elastic.co/kr/what-is/elasticsearch-machine-learning/recipes>

## 2. Entity based ML : Data Frame Analytics Concepts

# Elastic Data frames transforms

## 1<sup>st</sup> Transform: Pivot

```
Original Data:  
{  
    index: reviews,  
    user: rich,  
    vendor: acme,  
    review: 4,  
    timestamp: 2019-04-26T10:12:01  
,  
{  
    index: reviews,  
    user: rich,  
    vendor: transpop,  
    review: 3,  
    timestamp: 2019-04-28T10:17:42  
,  
...  
}
```



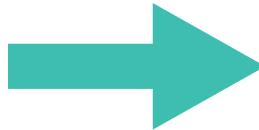
```
Output stream:  
{  
    index: reviews,  
    user: rich,  
    avg_review: 3.5,  
    num_reviews: 2  
}
```

*Entity-centric indexing/summarization from a data stream*

# Elastic Data frames transforms

## 1<sup>st</sup> Transform: Pivot

```
Original Data:  
{  
    index: reviews,  
    user: rich,  
    vendor: acme,  
    review: 4,  
    timestamp: 2019-04-26T10:12:01  
,  
{  
    index: reviews,  
    user: rich,  
    vendor: transpop,  
    review: 3,  
    timestamp: 2019-04-28T10:17:42  
,  
{  
    index: reviews,  
    user: rich,  
    vendor: vcentrix,  
    review: 5,  
    timestamp: 2019-04-29T09:27:12  
,  
...
```



```
Output stream:  
{  
    index: reviews,  
    user: rich,  
    avg_review: 4,  
    num_reviews: 3  
}
```

*Entity-centric indexing/summarization from a data stream*

# “DataFrames”

What do people do with it?

- Analyze multiple variables together
- Model real-world situations
- Find outliers
- Make predictions

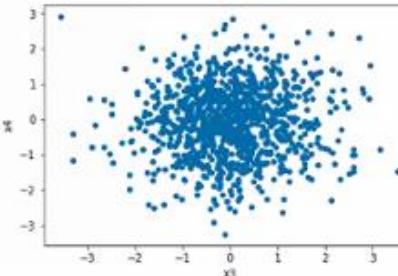
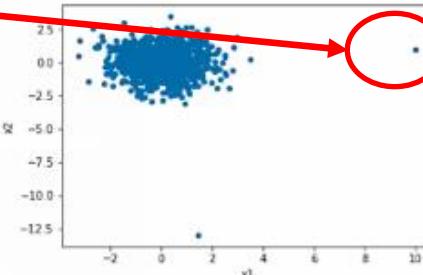
```
In [5]: import pandas as pd
from matplotlib import pyplot as plt

In [15]: dfx = pd.read_csv('/Users/thomasveasey/data/x')
dfx.head()

Out[15]:
   id      x1      x2      x3      x4      x5      x6
0  1  10.00000  0.988253 -1.279285  0.373475 -0.829858 -1.530520
1  2   1.452591 -13.000000  0.266071 -0.067993  0.109710  0.780813
2  3  -1.37421 -1.380247 -0.982101 -1.150864 -0.425088  0.999819
3  4  -0.310192 -0.638912 -0.939306  1.544084 -0.933099  0.167036
4  5  -0.305223 -1.793177  2.122106  0.522571  0.455539 -0.312145
```

```
In [11]: dfx = pd.read_csv('/Users/thomasveasey/data/x')
dfx.plot.scatter(x = 'x1', y = 'x2')
dfx.plot.scatter(x = 'x3', y = 'x4')

Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x121f42eb8>
```



## Use Cases

- Sales Analysis (House prices, etc.)
- Unusual Transactions (Fraud, etc.)
- User Behavior (Customer churn, etc.)

# “DataFrames”

## Common Challenges?

- Analysis often limited by RAM of single machine
- Data Scientists cannot/don't want to design and manage a scalable architecture for analysis
- Data may already be collected and centralized in a "Big Data" tool, but cannot "use it there"
- Data may exist in a time-series format and would need to be transformed

# Lab 5, Transforms

# Elastic Data frames transforms

## 1<sup>st</sup> Transform: Pivot

The screenshot shows the 'Define pivot' step in the 'New data frame' wizard. On the left, there's a sidebar with icons for index patterns, queries, group by, aggregations, and advanced editor. The main area has three tabs: 'Source index filebeat-\*' (showing 8 of 24 fields), 'Data frame pivot preview' (with a message: 'Pivot preview not available. Please choose at least one group-by field and aggregation.'), and a 'Next' button.

**Source index filebeat-\***

@timestamp	event.module	http.response.status	os.name	source.address	url.original
January 31st 2017, 18:11:46	nginx	200	Windows 7	10.252.178.245	/wp/products
January 31st 2017, 18:50:05	nginx	404	Windows 7	10.148.81.147	/wp-login.php
January 31st 2017, 18:22:28	nginx	304	Windows 10	10.93.12.146	/favicon.ico
January 31st 2017, 18:49:10	nginx	200	Windows 10	10.76.111.42	/cloud
January 31st 2017, 20:11:46	nginx	200	Windows 7	10.16.227.39	/favicon-32x32.png

**Data frame pivot preview**

Pivot preview not available  
Please choose at least one group-by field and aggregation.

**Transform details**

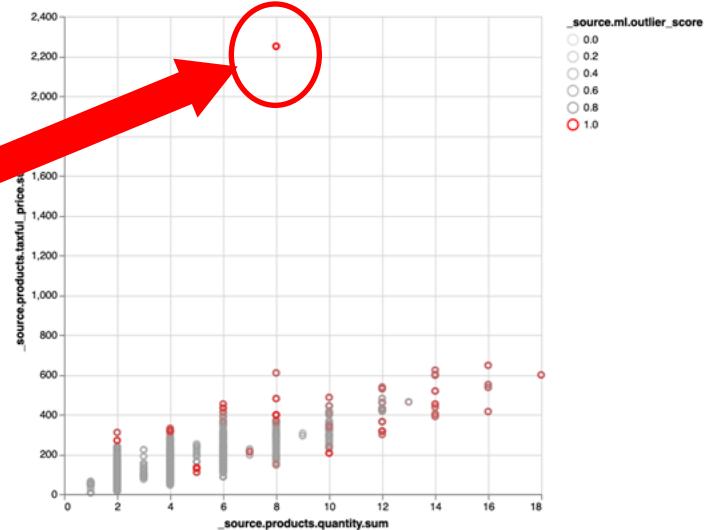
**Create**

*Entity-centric indexing/summarization from a data stream*

# Elastic “Data frames analytics”

2<sup>st</sup> Analysis – Multi-dimensional outlier detection, Regression, Classification...

```
"customer_full_name" : {  
    "keyword" : "Wagdi Shaw"  
},  
"ml__id_copy" : "Vyu9e08pKNasT-9TLV9p3k0AAAAAAA",  
"products" : {  
    "taxful_price" : {  
        "sum" : 2250.0  
    },  
    "quantity" : {  
        "sum" : 8.0  
    }  
},  
"ml" : {  
    "outlier_score" : 0.9848338961601257  
    "feature_influence.products.quantity.sum" : 0.007586637046188116,  
    "feature_influence.products.taxful_price.sum" : 0.992413341999054  
}
```



# Topics in Production

# Custom URLs: Linking to other places, in context

# Custom URLs in Job config

*Name the  
label*

*where to  
link*

*fields for  
filtering*

The screenshot shows the 'Edit' interface of a job configuration tool. The 'Custom URLs' tab is active. A modal window titled 'Create new custom URL' is open, containing the following fields:

- Label:** Raw data
- Link to:** Discover (selected)
- Index pattern:** farequote-\*
- Query entities:** airline
- Time range:** auto

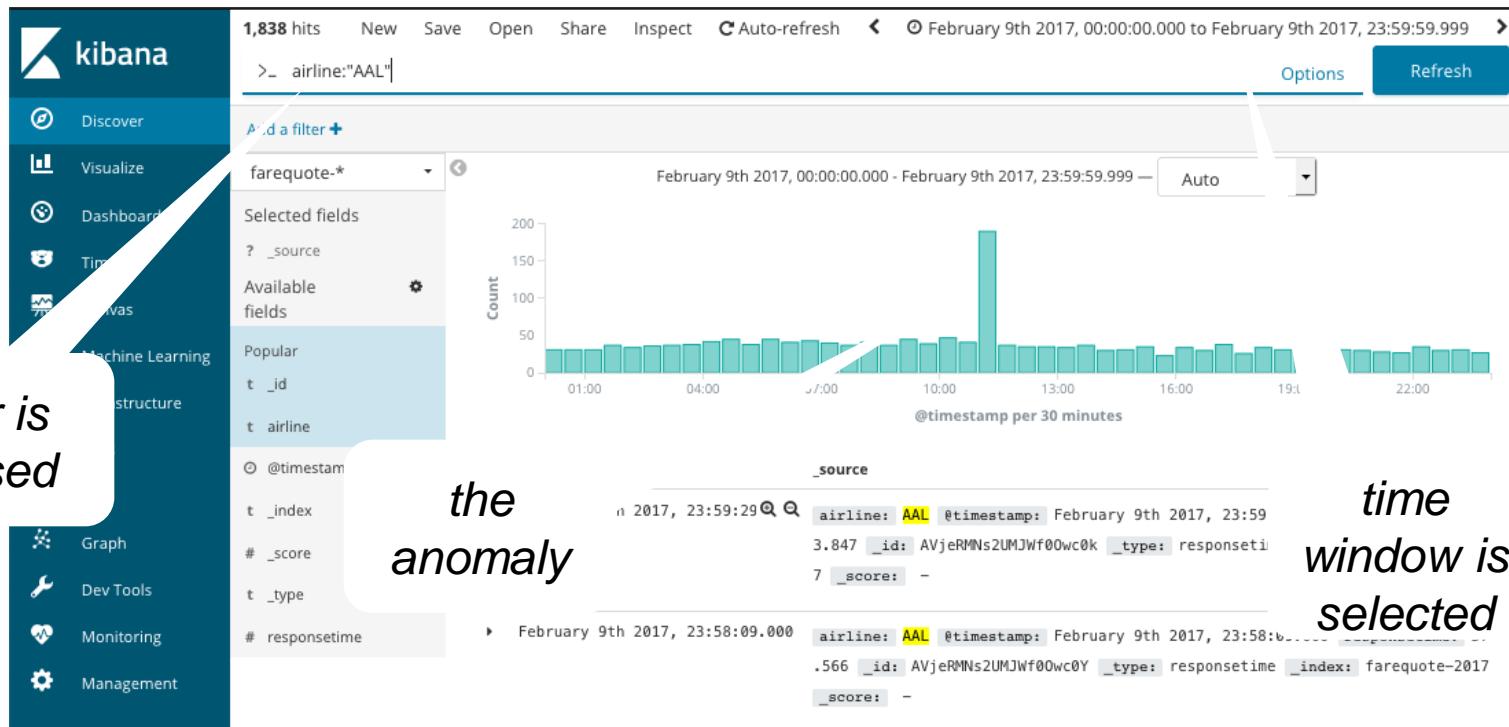
At the bottom of the modal are 'Add' and 'Test' buttons.

# In Anomaly Explorer

Anomalies									
Severity threshold		Interval							
time	max severi... ↓	detector	found for	influenced by	actual	typical	description	job ID	actions
> February 9th 2017	● 98	count	AAL	airline: AAL	136	17.9	↑ 8x higher	farequote_multi	⚙️
> February 9th 2017	● 98	max(responsetime)	AAL	airline: AAL	353.321	142.289	↑ 2x higher	farequote	↗ Raw data
> February 10th 2017	● 94	max(responsetime)	ACA	airline: ACA	21.98	24.034	↓ 1.1x lower	rarequote	↗ View series
> February 11th 2017	● 89	max(responsetime)	JAL	airline: JAL	506.9	515.432	↓ 1.1x usually low	farequote	⚙️ Configure rules

*Link now  
available*

# Clicking on Custom URL, passing context



# Custom Rules

# Adding Custom Rules to ML (v6.4+)



Define rules that:

- Ignore the creation of anomalies
- Disqualify data from being modeled

Meet certain conditions:

- on “actual”, “typical” or difference

Optional scoping

- limit when rule is applied

The screenshot shows the 'Create Rule' dialog box. At the top, it displays the job ID 'farequote\_multi', detector 'max(responsetime)', and selected anomaly 'actual 353.3, typical 142.3'. Below this is a descriptive text about rules. The 'Action' section contains two checkboxes: 'Skip result (recommended)' (checked) and 'Skip model update'. The 'Conditions' section has a checked checkbox for 'Add numeric conditions for when the rule applies' and a condition 'WHEN actual IS LESS THAN 300'. The 'Scope' section has an unchecked checkbox for 'Add a filter list to limit where the rule applies'. A note at the bottom states that changes take effect after cloning and rerunning the job.

Create Rule

Job ID: farequote\_multi  
Detector: max(responsetime)  
Selected anomaly: actual 353.3, typical 142.3

Rules instruct anomaly detectors to change their behavior based on domain-specific knowledge that you provide. When you create a rule, you can specify conditions, scope, and actions. When the conditions of a rule are satisfied, its actions are triggered. [Learn more](#)

Action

Choose the action(s) to take when the rule matches an anomaly.

Skip result (recommended) ⓘ  
 Skip model update ⓘ

Conditions

Add numeric conditions for when the rule applies. Multiple conditions are combined using AND.

WHEN actual IS LESS THAN 300 ⓘ

Add new condition

Scope

Add a filter list to limit where the rule applies.

Rerun job

Changes to rules take effect for new results only.

To apply these changes to existing results you must clone and rerun the job. Note rerunning the job may take some time and should only be done once you have completed all your changes to the rules for this job.

X Close Save

# Calendars

# Ignoring Timeframes with Calendars

Define timeframes for ML to

- not create anomalies
- not allow model to see new data

Good for known, upcoming times

Calendars can apply to individual jobs or job groups

The screenshot shows a table with one row. The first column is 'ID' with value 'nyc\_holidays'. The second column is 'Jobs' with value 'taxi'. The third column is 'Events' with value '12 events'. There are 'Edit' and 'Delete' buttons at the bottom right.

ID	Jobs	Events
nyc_holidays	taxi	12 events

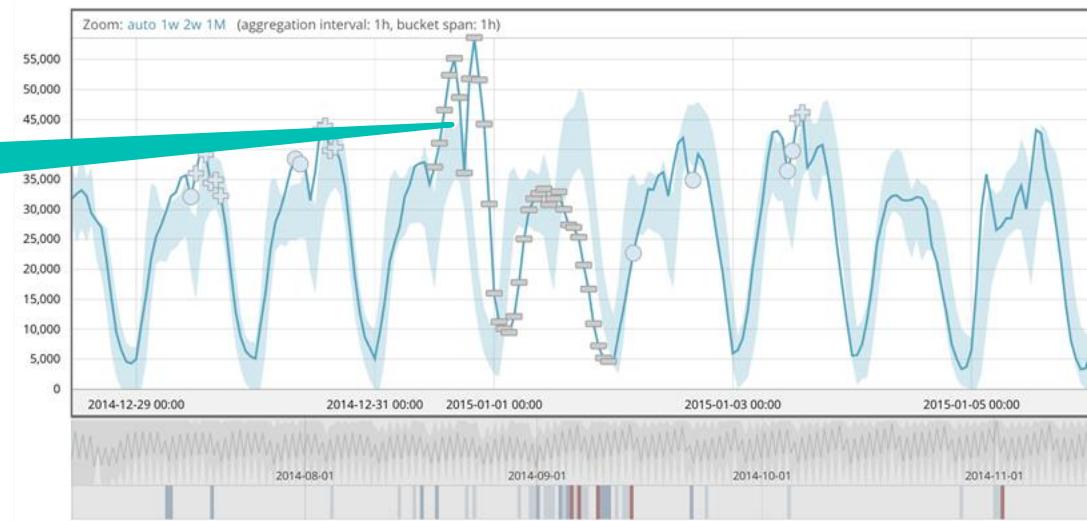
The screenshot shows an 'Edit calendar' form for 'nyc\_holidays'. It has fields for 'Calendar ID' (set to 'nyc\_holidays') and 'Description'. Under 'Jobs', there is a dropdown with 'jobs'. Under 'Groups', there is a dropdown with 'taxi'. The 'Events' section contains a table with 12 rows, each representing a holiday with its start and end dates. Buttons for 'New event' and 'Import events' are at the top of the events table.

Description	Start	End	Action
Christmas Day	2014-12-25 00:00:00	2014-12-26 00:00:00	Delete
Christmas Eve	2014-12-24 12:00:00	2014-12-25 00:00:00	Delete
Columbus Day	2014-10-13 00:00:00	2014-10-14 00:00:00	Delete
Day after Christmas	2014-12-26 00:00:00	2014-12-27 00:00:00	Delete
Day after Thanksgiving	2014-11-28 00:00:00	2014-11-29 00:00:00	Delete
Independence Day	2014-07-04 00:00:00	2014-07-05 00:00:00	Delete
Labor Day	2014-09-01 00:00:00	2014-09-02 00:00:00	Delete
Martin Luther King Jr Day	2015-01-19 00:00:00	2015-01-20 00:00:00	Delete
New Years Day	2015-01-01 00:00:00	2015-01-02 00:00:00	Delete
New Years Eve	2014-12-31 12:00:00	2015-01-01 00:00:00	Delete
Thanksgiving	2014-11-27 00:00:00	2014-11-28 00:00:00	Delete
Veterans Day	2014-11-11 00:00:00	2014-11-12 00:00:00	Delete

Times are displayed in the browser timezone

# Ignoring Timeframes with Calendars

*Ignored times marked  
with  
“—” symbol*





# Let's apply ML to your DATA!!!

# Thank you

<https://www.elastic.co/kr/what-is/elasticsearch-machine-learning>