

# Machine Learning con Elasticsearch

<b>Machine Learning con Elasticsearch</b>	<b>1</b>
Setup del entorno	2
Demo 1. Machine Learning no supervisado para detección de anomalías	4
Demo 2. Outlier Detection y Machine Learning supervisado	14
Detección de valores atípicos / Outlier Detection	15
Machine Learning Supervisado - Regresión	21
Machine Learning Supervisado - Clasificación	24
Inferencia	26
Demo 3. Carga de un fichero JSON/CSV	32
Gist	35

Para ejecutar las demos se debe clonar el proyecto <https://github.com/immavalls/viu-elk-ml-talk>

```
git clone git@github.com:immavalls/viu-elk-ml-talk.git
```

E instalar los prerequisites documentados. Estas demos requieren **docker** y **docker-compose**.

## Setup del entorno

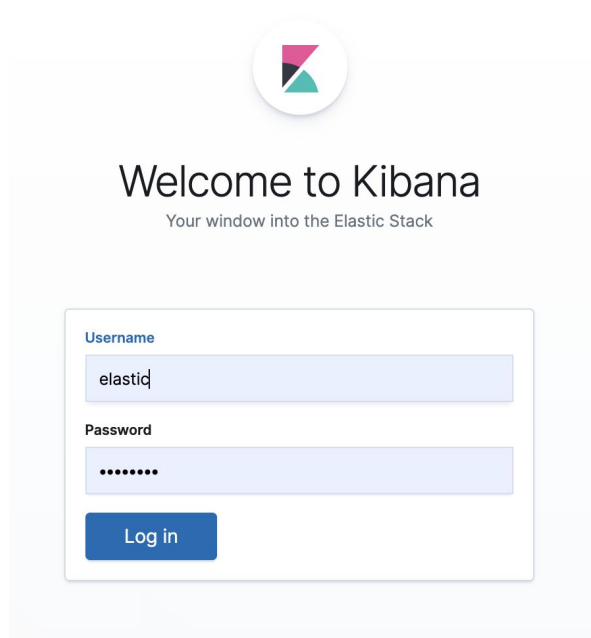
Para ello, situados en el raíz del proyecto **viu-elk-m1-talk**, ejecutar:

```
docker-compose up -d
```

Comprobar que ha arrancado correctamente:

<http://localhost:5601/>

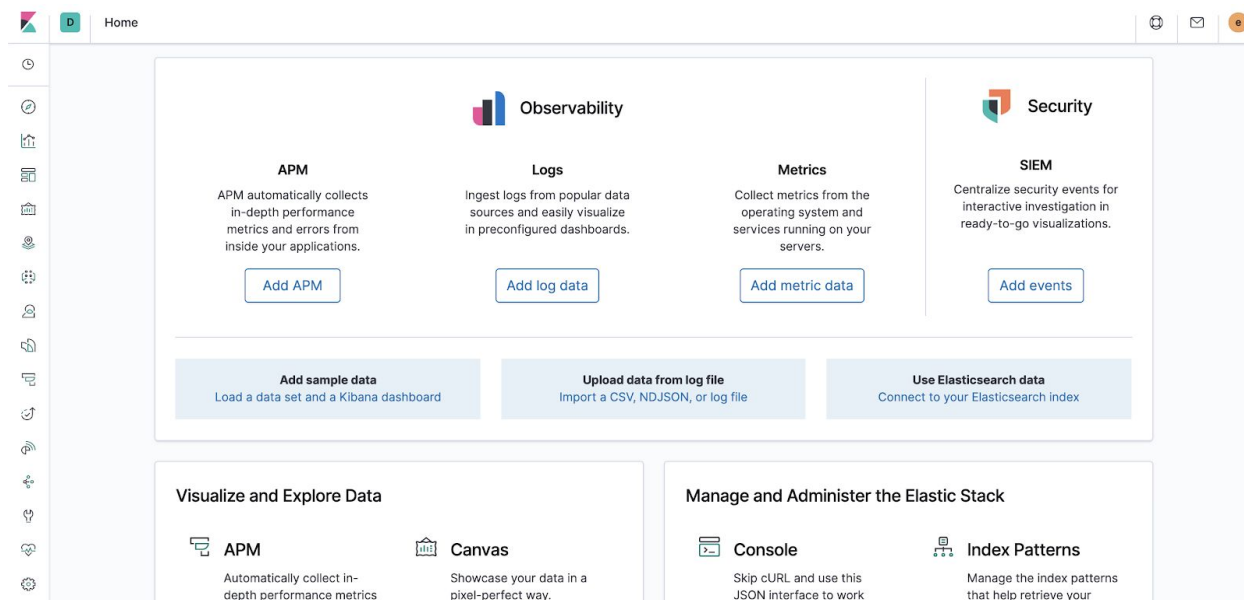
Si Elasticsearch y Kibana han arrancado correctamente, veremos el pantalla de Login de Kibana.



Entrar con:

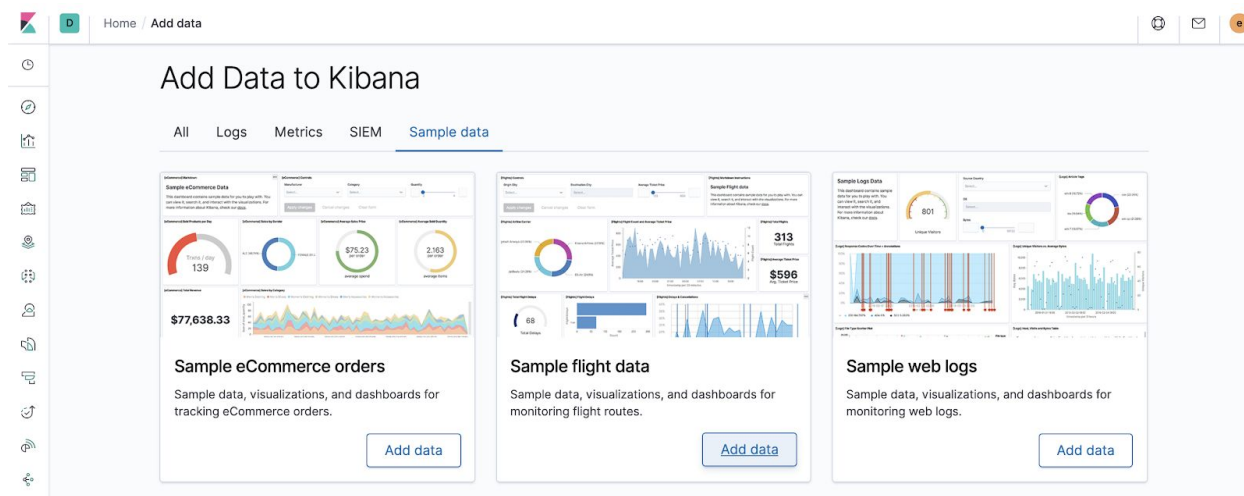
**Usuario:** elastic

**Contraseña:** changeme



Vamos a cargar datos de prueba que proporciona Kibana, para facilitar la exploración del Stack.

En la parte central de la pantalla Home, Pulsar en “Add sample data!”, “Load a data set and a Kibana dashboard”.



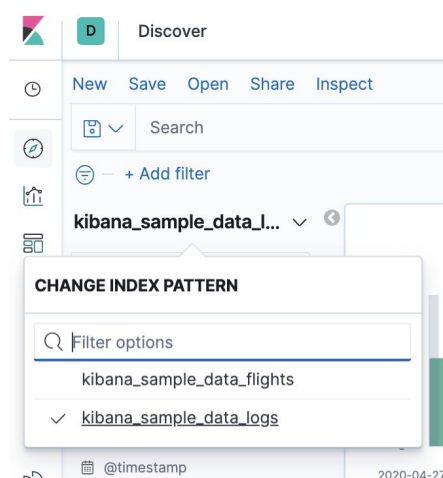
Pulsar “Add data” en **Sample flight data** y en **Sample web logs**.

## Demo 1. Machine Learning no supervisado para detección de anomalías

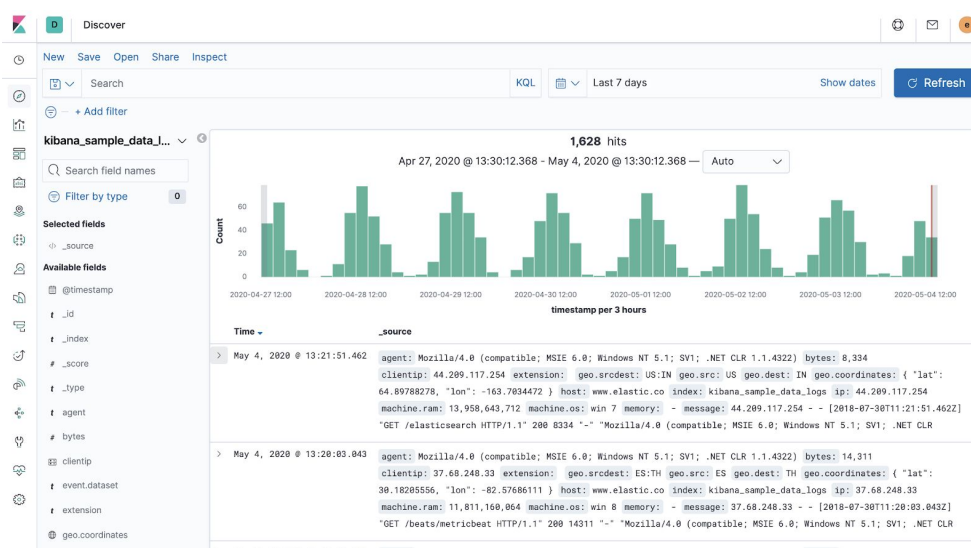
Esta demo va a usar los datos de ejemplo que vienen incorporados en Kibana, datos que hemos cargado en el punto anterior de setup.

Usaremos los datos de logs de acceso a nuestra web, los que hemos cargado como **“Sample web logs”**.

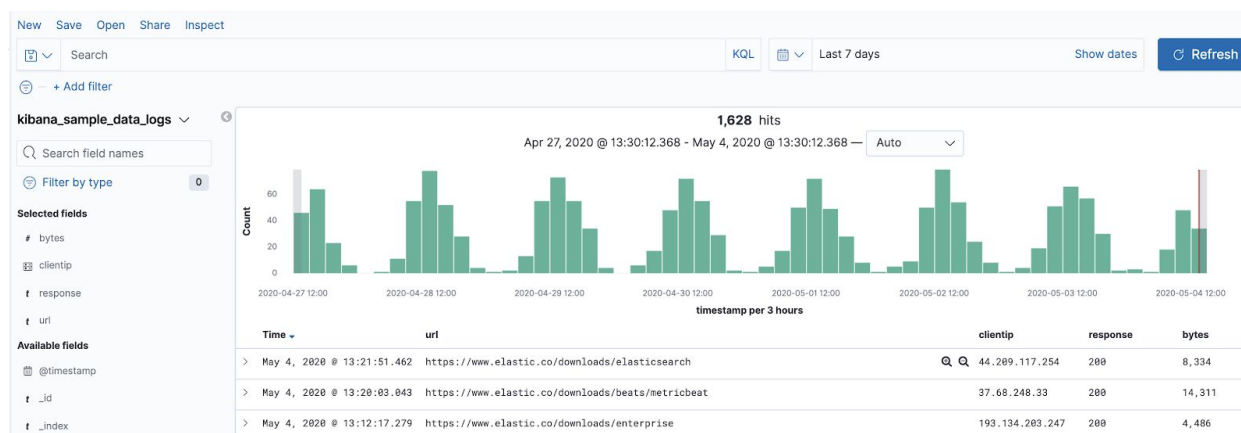
Si pulsamos en el menú de la izquierda Discover y seleccionamos “kibana\_sample\_data\_logs”



Podemos estudiar el formato de estos datos en el tiempo.



Para cada petición tenemos la URL, la IP del cliente que la pidió, el código de respuesta http (200, 404, 503, etc.), el número de bytes de esa petición, etc.

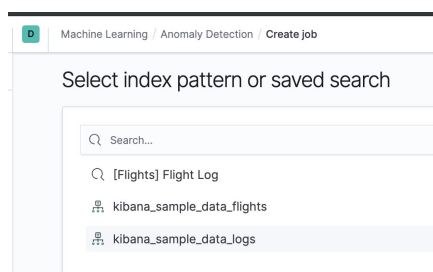


Pulsaremos en el menú de la izquierda en Machine Learning . Iremos a la pantalla de gestión de jobs, haciendo click en la segunda pestaña “Anomaly Detection”.

Overview Anomaly Detection Data Frame Analytics Data Visualizer

Pulsaremos .

Seleccionamos el índice “kibana\_sample\_data\_logs”.



Dado que el índice “kibana\_sample\_data\_logs” sigue Elastic Common Schema (ECS)<sup>1</sup> al nombrar sus campos, Machine Learning reconoce los datos y nos propone algunos jobs pre-construidos<sup>2</sup> que podemos utilizar directamente.

<sup>1</sup> <https://www.elastic.co/guide/en/ecs/current/index.html>

<sup>2</sup> <https://www.elastic.co/guide/en/machine-learning/current/ootb-ml-jobs.html>

## Create a job from the index pattern kibana\_sample\_data\_logs

### Use a supplied configuration

The fields in your data have been recognized as matching known configurations. Select to create a set of machine learning jobs and associated dashboards.



#### Kibana sample data web logs

Find anomalies in Kibana  
sample web logs data

Hacemos click en



#### Kibana sample data web logs

Nos pide que le demos un nombre a los jobs que vamos a crear, ya que va a crear 3 jobs. Pongamos “viu\_”, por ejemplo.

## New job from index pattern kibana\_sample\_data\_logs

### Job settings

#### Job ID prefix

A prefix which will be added to the  
beginning of each job ID.

Job ID prefix

viu\_

☒ Start datafeed after save

☒ Use full kibana\_sample\_data\_logs data

> Advanced

Create Jobs

### Jobs

viu\_url\_scanning

Find client IPs accessing an unusually high distinct count of URLs

kibana\_sample\_data kibana\_sample\_web\_logs

viu\_response\_code\_rates

Find unusual event rates by HTTP response code (high and low)

kibana\_sample\_data kibana\_sample\_web\_logs

viu\_low\_request\_rate

Find unusually low request rates

kibana\_sample\_data kibana\_sample\_web\_logs

Vamos a crear estos jobs, pulsando Create Jobs. Esto crea cada uno de los tres jobs, que lee los datos en los índices “kibana\_sample\_data\_logs”.

Volvemos vía menú superior a “Anomaly Detection”

## New job from index pattern kibana

Y observamos que hay 3 jobs.

Overview

Anomaly Detection

Data Frame Analytics

Data Visualizer

🕒

▼

30 seconds

Job Management

Anomaly Explorer

Single Metric Viewer

Settings

Anomaly detection jobs

Active ML Nodes: 0

Total jobs: 3

Open jobs: 0

Closed jobs: 3

Active datafeeds: 0

Refresh

Create new job

🔍 Search...

Opened	Closed	Failed	Started	Stopped	Group

<input type="checkbox"/>	ID ↑	Description	Processed records	Memory status	Job state	Datafeed state	Latest timestamp	Actions
<input type="checkbox"/>	> viu_low_request_rate	Find unusually low request rates kibana_sample_data kibana_sample_web_logs	1,216	ok	closed	stopped	2020-06-25 22:49:29	
<input type="checkbox"/>	> viu_response_code_rates	Find unusual event rates by HTTP response code (high and low) kibana_sample_data kibana_sample_web_logs	14,073	ok	closed	stopped	2020-06-25 22:49:29	
<input type="checkbox"/>	> viu_url_scanning	Find client IPs accessing an unusually high distinct count of URLs kibana_sample_data kibana_sample_web_logs	14,073	ok	closed	stopped	2020-06-25 22:49:29	

Rows per page: 10

<

1


>

En el primer job, estamos buscando tasas de peticiones bajas. Es decir, **caídas de visitas en la web**.

En el segundo estamos mirando la **tasa de los códigos de respuesta**. Este es un ejemplo de job que modela múltiples series temporales. Modela la tasa de cada código de respuesta http, 200, 404, 503, etc. y busca cambios en la tasa de ocurrencia de esas respuestas a lo largo del tiempo.

Y el último está detectando **escaneo de urls**. El job está mirando si hay IPs de clientes accediendo a un número inusualmente alto de URLs distintas.

Vamos a empezar viendo que nos ha detectado el primer job, de tasas bajas de peticiones.

Donde estamos buscando caídas de visitantes en nuestra web. Pulsamos  para visualizar el resultado de este job.

[Job Management](#)

[Anomaly Explorer](#)

[Single Metric Viewer](#)

[Settings](#)

# Anomaly detection jobs

Active ML Nodes: 0

Total jobs: 3

Open jobs: 0

Closed jobs: 3

Active datafeeds: 0

Refresh

Create new job

Q Search...

Opened

Closed

Failed

Started

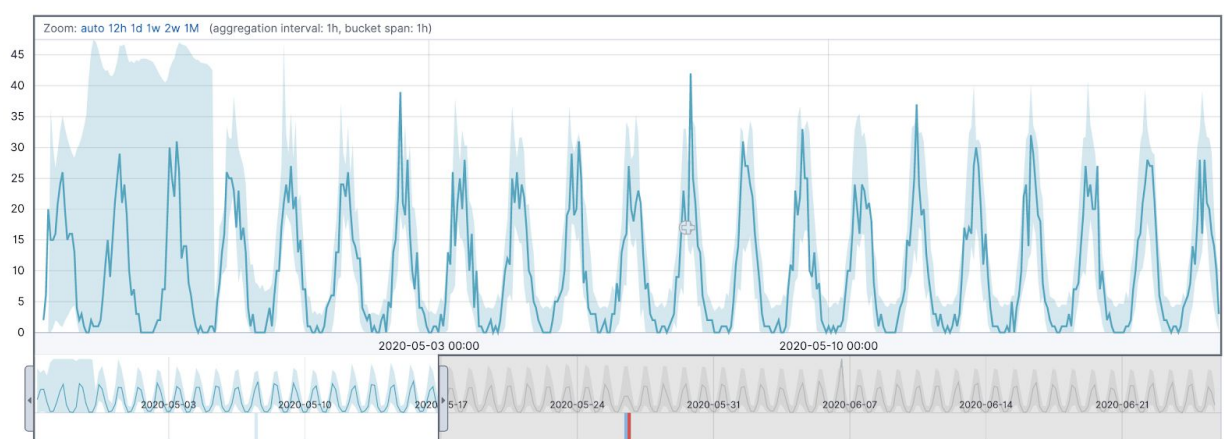
Stopped

Group

<input type="checkbox"/>	ID ↑	Description	Processed records	Memory status	Job state	Datafeed state	Latest timestamp	Actions
<input type="checkbox"/>	> viu_low_request_rate	<div>Find unusually low request rates</div> <div>kibana_sample_data</div> <div>kibana_sample_web_logs</div>	1,216	ok	closed	stopped	2020-06-25 22:49:29	<div><div>Visualize</div><div>Details</div><div>More</div></div>
<input type="checkbox"/>	> viu_response_code_rates	<div>Find unusual event rates by HTTP response code (high and low)</div> <div>kibana_sample_data</div> <div>kibana_sample_web_logs</div>	14,073	ok	closed	stopped	2020-06-25 22:49:29	<div><div>Visualize</div><div>Details</div><div>More</div></div>

Open viu\_low\_request\_rate in Single Metric Viewer

Usando la barra inferior, nos podemos desplazar al inicio de la serie temporal.



La línea azul son los datos reales. Y el sombreado azul son los límites de nuestro modelo.

Observamos que tenemos un patrón diario de visitas, con hora pico y horas valle.

Al inicio del job, los límites son muy amplios. No hemos visto todavía suficientes datos para conocer cómo se comporta, todavía.

En este conjunto de datos, a partir del cuarto día ya identificamos que el patrón es diario.

Si nos desplazamos un poco adelante en el tiempo, veremos nuestra primera **anomalía crítica**.





Vemos que hubo una caída a cero en las visitas de la web. Supongamos que investigamos ese momento, y vimos que tuvimos alguna caída de la web, en algún sistema, que impedía a los usuarios acceder. Podemos por ejemplo introducir una anotación a este periodo de tiempo, ya que hemos hecho una investigación de los datos y queremos dejar constancia para quien quiera saber qué pasó.

Seleccionamos sobre la gráfica y creamos la anotación.

Overview [Anomaly Detection](#) Data Frame Analytics Data Visualizer

Apr 26, 2020 @ 05:57:

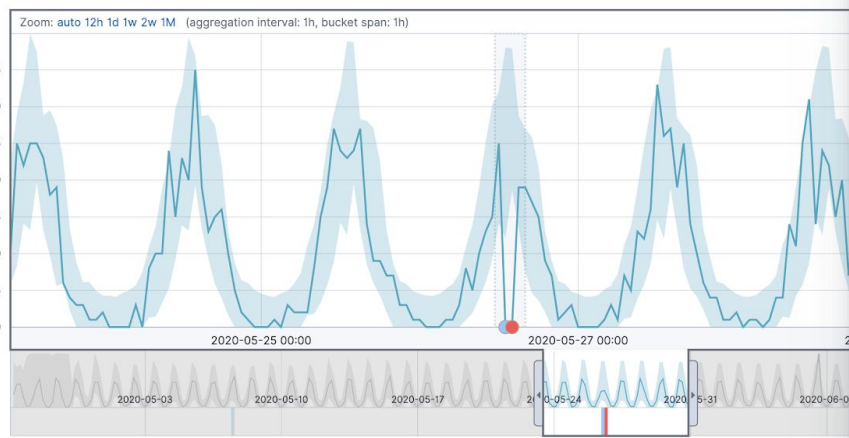
Job Management Anomaly Explorer [Single Metric Viewer](#) Settings

viu\_low\_request\_rate [Edit job selection](#)

Detector

Low request rates

Single time series analysis of count



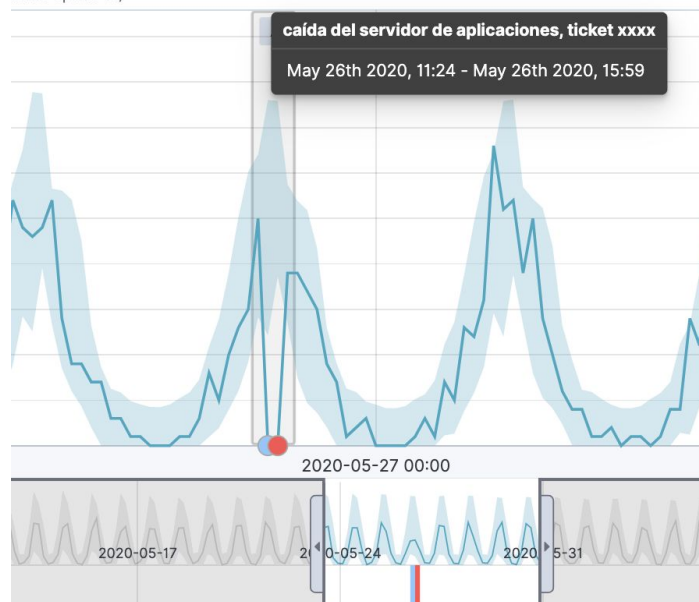
Add annotation

**Job ID** viu\_low\_request\_rate  
**Start** May 26th 2020, 11:24:20  
**End** May 26th 2020, 15:59:03

Annotation text

caída del servidor de aplicaciones, ticket xxxx

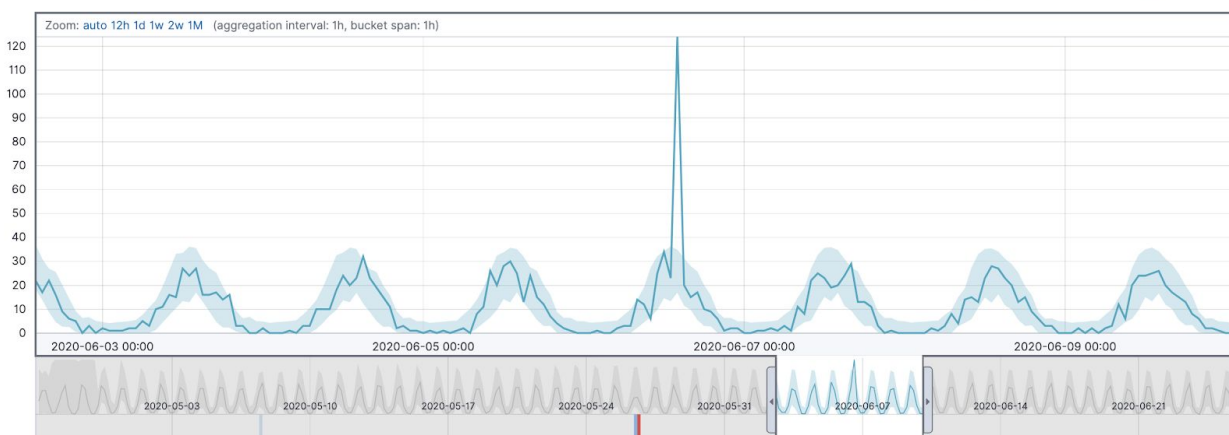
ticket span: 1h)



Más adelante en el tiempo, veremos que hay un pico de visitas muy alto.

Single time series analysis of count

☒ show model bounds ☒ annotations



Anomalies

Severity threshold

☐ warning

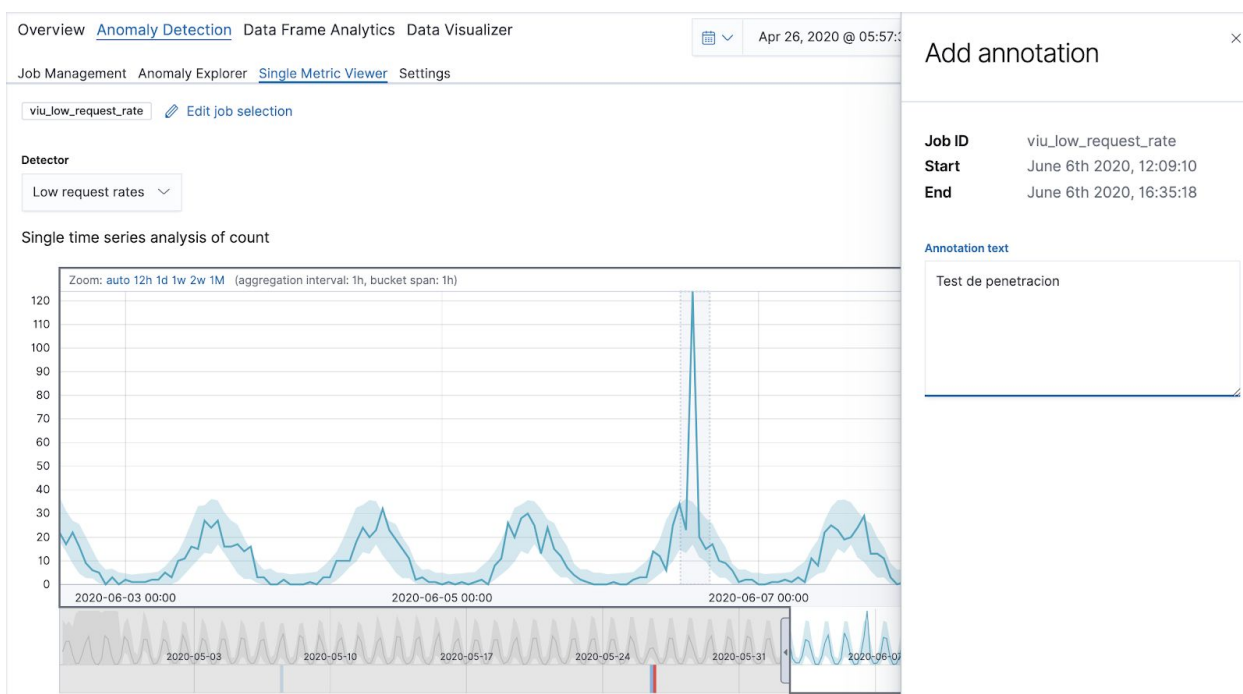
Interval

☐ Auto

No matching anomalies found

En este job no aparece como anomalía, porque estamos buscando sólo caídas de visitas a nuestra web, no picos. Hemos configurado el job para que nos avise de caídas de visitas sólo, esa fue nuestra elección, basado en cuando queríamos recibir alertas.

Pongamos que sabemos que, en este caso, se trataba de un test de penetración, y lo podríamos anotar también. De forma que si algún compañero o compañera lo ve en el futuro sepa que se trata de una prueba controlada.

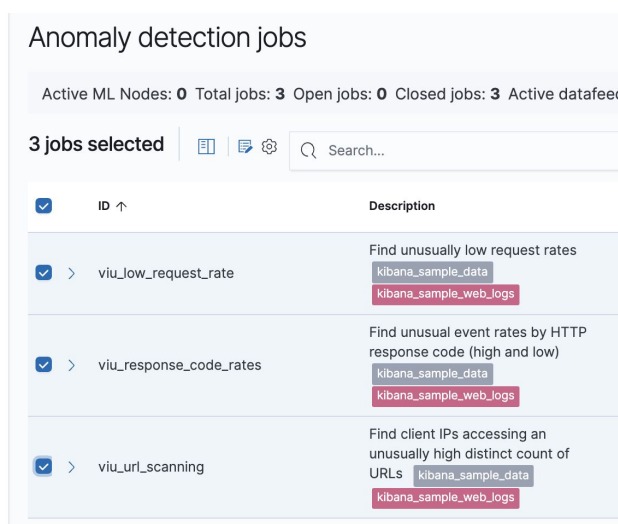


Volvemos a la página de gestión de jobs, seleccionando **“Job Management”** en el segundo nivel de menús superior.

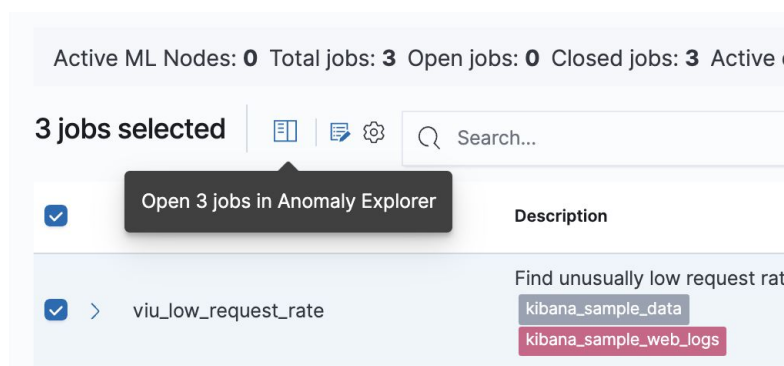


Una de las ventajas es que no tenemos porqué mirar estos jobs por separado, sino que podemos seleccionar los tres y ver una vista con todos.

Para ello marcamos los tres jobs:



Y pulsamos



Para abrir los tres jobs a la vez en el explorador de anomalías.

Ahora vemos en la parte superior la puntuación total de las anomalías en el tiempo, en

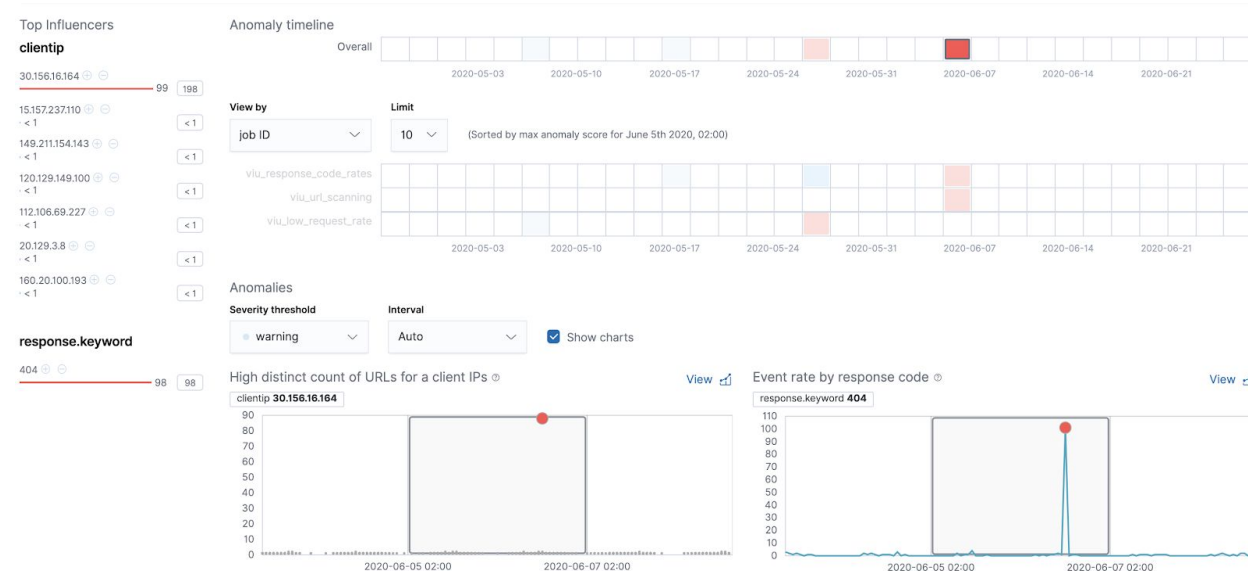
“Overall”. Y observamos que han habido dos momentos críticos.



Seleccionamos la primera anomalía y podemos ver el detalle. Detectamos esta anomalía con el job de caída de visitas. Y también se detectó como un número inusual, cero, de códigos de respuesta 200 ok http.



Seleccionando la segunda anomalía, podemos ver que los dos jobs que han detectado anomalía son el de escaneo de nuestra web, y el de códigos inusuales de respuesta para código http 404, página no encontrada. Que coincide con un test de penetración.



Hay un número elevado de respuestas 404 y podemos ver que son de la misma IP, 30.156.16.164.

## Demo 2. Outlier Detection y Machine Learning supervisado

## Detección de valores atípicos / Outlier Detection

Para esta demo, partiremos de los datos “kibana\_samples\_data\_logs”, y los vamos a pivotar, transformar, para conseguir un modelo entity-centric, en vez de lo al serie temporal que tiene el índice.

Lo que queremos es agrupar por IPs de cliente, calculando cuántos errores (código de respuesta 404) vemos por cada IP, cuantos códigos 200, la suma total de bytes y el número de URLs distintas que ha visitado.

Con la intención de localizar, de forma no supervisada, si hay alguna IP que se comporta de forma atípica para estos casos.

El primer paso será crear estos datos transformados. Para ello podemos usar Kibana UI<sup>3</sup>, o directamente via API REST. En esta demo haremos uso del API REST.

Seleccionamos en el menú de la izquierda de Kibana  Dev Tools y ejecutamos lo siguiente:

```
PUT _transform/clientip-activity
{
  "id": "clientip-activity",
  "source": {
    "index": [
      "kibana_sample_data_logs"
    ],
    "query": {
      "match_all": {}
    }
  },
  "dest": {
    "index": "clientip-activity"
  },
  "pivot": {
    "group_by": {
      "clientip": {
        "terms": {
          "field": "clientip"
        }
      }
    },
    "aggregations": {
      "sum_bytes": {
        "sum": {
          "field": "bytes"
        }
      },
      "url_cardinality": {
        "cardinality": {
          "field": "url.keyword"
        }
      }
    }
  }
}
```

<sup>3</sup> <https://www.elastic.co/guide/en/elasticsearch/reference/7.6/ecommerce-transforms.html>

```


    },
    "count_success": {
      "scripted_metric": {
        "init_script": "state.success_status = 0",
        "map_script": "state.success_status += doc['response.keyword'].value == '200' ? 1 : 0",
        "combine_script": "state.success_status",
        "reduce_script": "long success = 0; for (a in states) { success += a } return success"
      }
    },
    "count_errors": {
      "scripted_metric": {
        "init_script": "state.success_status = 0",
        "map_script": "state.success_status += doc['response.keyword'].value == '404' ? 1 : 0",
        "combine_script": "state.success_status",
        "reduce_script": "long success = 0; for (a in states) { success += a } return success"
      }
    }
  },
  "description": "Client IP activity"
}

POST _transform/clientip-activity/_start

```

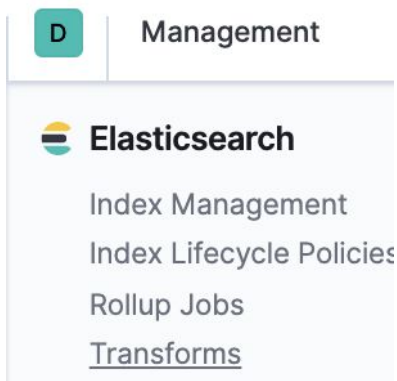
Esto crea una **transformación** llamada **“clientip-activity”**, que crea datos pivotados sobre el índice **“clientip-activity”**. Y la inicia.

Podremos comprobar los resultados en Kibana. Si vamos a

 Management

, y

seleccionamos **“Transforms”**:



Veremos la transformación definida.



# Transforms

BETA

[Transform docs](#)

Use transforms to pivot existing Elasticsearch indices into summarized or entity-centric indices.

Total transforms: 1 Batch: 1 Continuous: 0 Started: 0

Status ▾
Mode ▾

Reload

Create a transform

<input type="checkbox"/>	ID ↑	Description	Source index	Destination index	Status	Mode	Progress	Actions
<input type="checkbox"/>	clientip-activity	Client IP activity	kibana_sample_d...	clientip-activity	stopped	batch	100%	<div>Start</div> <div>Delete</div>

Transform details JSON Messages Preview

**State**

**ID** clientip-activity

**state** stopped

**Stats**

**pages\_processed** 4

**documents\_processed** 14074

**documents\_indexed** 1001

**trigger\_count** 1

**index\_time\_in\_ms** 104

**index\_total** 3

**index\_failures** 0

**search\_time\_in\_ms** 400

**search\_total** 4

**search\_failures** 0

**exponential\_avg\_checkpoint\_duration\_ms** 0

**exponential\_avg\_documents\_indexed** 0

**exponential\_avg\_documents\_processed** 0

**Checkpointing**

**last\_checkpoint** 1

**last\_timestamp** May 4th 2020, 14:09:34

**last\_timestamp\_millis** 1588594174926

Rows per page: 10 ▾

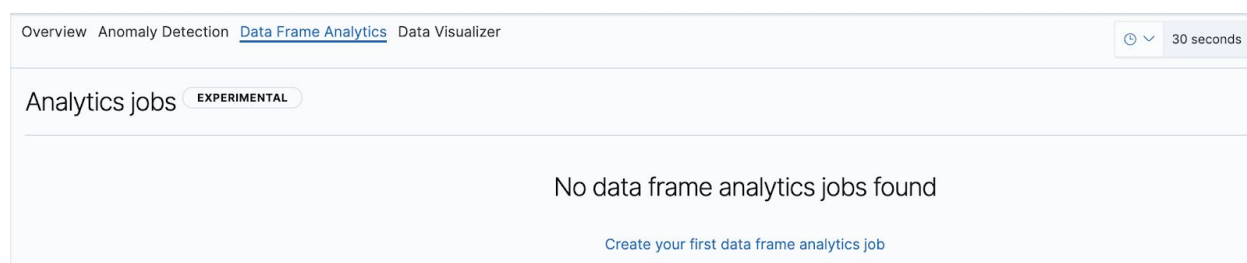
< 1 >

Y podemos previsualizar los datos pulsando en la pestaña

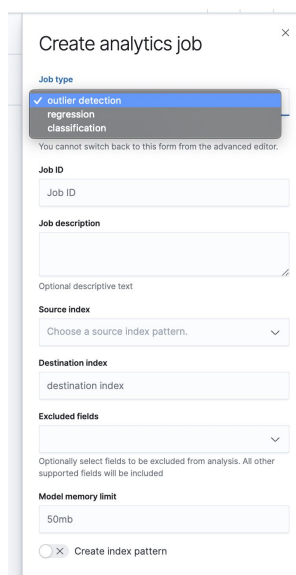
Preview

clientip	count_errors	count_success	sum_bytes	url_cardinality
0.72.176.46	0	14	74808	11
0.207.229.147	1	9	43395	11
0.209.144.101	0	14	87795	10
1.5.239.89	0	11	66549	9
1.8.196.147	0	18	102723	13
1.40.4.126	1	12	85778	10
1.69.35.74	3	13	144607	15
1.145.31.121	0	13	84801	9
1.149.149.212	0	10	62960	11
1.206.237.153	1	13	69154	12

A partir de aquí, ya podemos ir a Machine Learning y seleccionar “**Data Frame Analytics**” en el menú superior.



Pulsamos [Create your first data frame analytics job](#) y seleccionamos “Outlier Detection”.



Nombramos el job, seleccionamos como índice origen “**clientip-activity**”, el índice que hemos creado con la transformación. E indicamos el índice para los resultados.

×

Create analytics job

Job type

outlier detection

▼

Outlier detection jobs require a source index that is mapped as a table-like data structure and will only analyze numeric and boolean fields. Please use the advanced editor to add custom options to the configuration.

☐ Enable advanced editor

You cannot switch back to this form from the advanced editor.

Job ID

clientip-outliers

Job description

Optional descriptive text

Source index

clientip-activity

▼

Destination index

results-clientip-activity

Excluded fields

▼

Optionally select fields to be excluded from analysis. All other supported fields will be included

Model memory limit

427kb

☒ Create index pattern

Pulsamos “Create” y “Start”.

Y volvemos a “Data Frame Analytics” para visualizar el resultado.

D

Machine Learning / Data Frame Analytics

Overview

Anomaly Detection

Data Frame Analytics

Data Visualizer

⌚

30 seconds

Analytics jobs

EXPERIMENTAL

Total analytics jobs: 1 Running: 0 Stopped: 1

Refresh

Create analytics job

Search...

Status ▼

ID ↑	Description	Source index	Destination index	Type	Status	Progress	Actions
clientip-outliers		clientip-activity	results-clientip-activity	outlier_detection	stopped	100%	View

Rows per page: 10

<

1

>

Seleccionamos  View

## Analytics exploration EXPERIMENTAL

Outlier detection job ID clientip-outliers stopped

Showing first 1000 documents

Feature influence score 0 0.2 0.4 0.6 0.8 1

Q E.g. avg>0.5

clientip	ml_outlier_score ↓	count_errors	count_success	sum_bytes	url_cardinality
30.156.16.164	0.9986892938613892	100	0	183100	88
81.84.213.90	0.9979990085106931	0	3	7998	4
50.184.59.162	0.9973342418670654	0	25	115577	16
16.241.165.21	0.9970191121101379	1	24	153285	18
239.255.110.98	0.9965410232543945	3	20	104168	20
164.85.94.243	0.9947751760482788	2	25	160518	20
111.237.144.54	0.9934860467910767	0	24	176720	18
221.241.228.46	0.9856972694396973	5	8	37824	11
57.65.101.133	0.9822317361831665	0	21	127496	11
246.106.125.113	0.9812902808189392	1	22	129121	20
236.212.255.77	0.9438316226005554	1	22	138839	16
247.240.202.244	0.9280132908821106	4	16	131384	18
172.0.84.195	0.895264467353821	3	4	37124	7
1.69.35.74	0.8925296664237976	3	13	144607	15
186.153.168.71	0.8572630882263184	0	22	127209	12
184.31.17.237	0.8493184447288513	0	5	29086	3
59.153.90.6	0.8386192321777344	1	9	95585	9
107.113.86.210	0.8346256017684937	1	9	34524	10
28.149.123.243	0.8295428156852722	3	10	88741	9
177.120.218.48	0.8255603313446045	0	19	169258	13
214.214.40.183	0.7924566864967346	0	9	25221	8
28.11.251.5	0.7635741233825684	1	7	67843	6
149.211.154.143	0.7441913485527039	1	10	33835	11
58.139.164.111	0.7239632606506348	2	11	39069	9
88.209.41.99	0.7057907581329346	4	10	75843	9

Rows per page: 25

< 1 2 3 4 5 ... 40 >

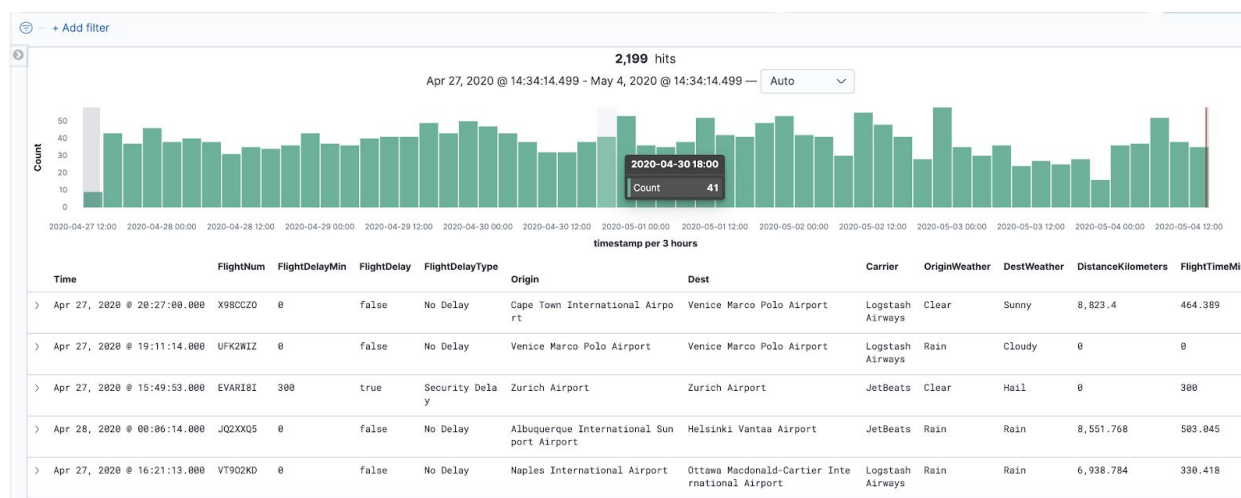
Y vemos que los resultados salen ordenados por “ml\_outlier\_score”, con un valor entre 0 y 1. Y estamos ordenando con los valores más inusuales en las filas superiores.

El sombreado de colores intenta mostrar porque motivo los valores, las IPs en este caso, piensa nuestro modelo que los valores son atípicos.

Por ejemplo, en la primera IP, basados en las 4 características que estamos mirando, número de errores 404, número de peticiones exitosas (200 ok), el total de bytes que se ha descargado o las distintas urls que ha visitado, vemos que el factor que ha incidido más en que sea una anomalía es el número de errores.

## Machine Learning Supervisado - Regresión

En este segundo ejemplo de analítica con Data Frames, vamos a usar los datos sobre vuelos que tenemos como ejemplo en Kibana, en el índice “**kibana\_samples\_data\_flights**”. En este índice tenemos documentos con diversos campos: aeropuerto origen, destino, tiempo en origen y destino, la distancia del viaje. Y sabemos para estos vuelos si se retrasaron o no, y cuánto.



Usaremos estos datos para entrenar un modelo que nos permita predecir futuros retrasos.

Crearemos un job de tipo regresión, ya que queremos predecir el retraso, un valor numérico.

En la configuración del job, la variable dependiente será “**FlightDelayMin**”, el campo en nuestros datos que contiene el retraso que se produjo.

Podemos seleccionar qué porcentaje de los datos queremos usar para entrenar y cual para validar el modelo. Lo dejaremos en 80% en este caso, en que tenemos pocos datos.

Y finalmente excluirémos de los datos algunos campos. En este caso, “**Cancelled**” porque nuestros datos no son limpios, y tenemos vuelos cancelados que marcan retraso. Y las dos variables que describen la variable dependiente, “**FlightDelay**” y “**FlightDelayType**”.

## Create analytics job ×

### Job type

regression

Regression jobs will only analyze numeric fields. Please use the advanced editor to apply custom options such as the prediction field name.

☐ Enable advanced editor

You cannot switch back to this form from the advanced editor.

### Job ID

flight\_delays

### Job description

Optional descriptive text

### Source index

kibana\_sample\_data\_flights

### Destination index

results-flight-delays

### Dependent variable

FlightDelayMin

### Training percent

0  80  100

### Excluded fields

Cancelled × FlightDelay × × ▼

FlightDelayType ×

Optionally select fields to be excluded from analysis. All other supported fields will be included

### Model memory limit

106569kb

☐ × Create index pattern

Creamos e iniciamos el job.

Analytics jobs EXPERIMENTAL

Total analytics jobs: 2 Running: 0 Stopped: 2 Refresh Create analytics job

Search...

Status

ID ↑	Description	Source index	Destination index	Type	Status	Progress	Actions
clientip-outliers		clientip-activity	results-clientip-activity	outlier_detection	stopped	100%	<span>View</span> <span>...</span>
flight_delays		kibana_sample_data_flights	results-flight-delays	regression	stopped	100%	<span>View</span> <span>...</span>

Rows per page: 10

< 1 >

Pulsamos en View para ver los resultados.

Destination index for regression job ID flight\_delays stopped 8 of 31 fields selected

Showing first 1000 documents for which predictions exist

Testing Training

ml.is_training	ml.FlightDelayMin_predict...	FlightDelayMin	AvgTicketPrice	Cancelled	Carrier	dayOfWeek	Dest
Yes	331.1597900390625	345	755.209875956966	No	ES-Air	5	Comodoro Arturo Merino Benitez International Airport
Yes	330.7502746582031	330	1143.47865794209	No	Logstash Airways	6	London Gatwick Airport
Yes	327.0926513671875	300	985.4856403355963	No	ES-Air	6	Treviso-Sant'Angelo Airport
Yes	325.0828552248094	300	633.7264358700636	No	ES-Air	6	Xian Xianyang International Airport
Yes	324.8549499511719	330	905.8597784108845	No	Logstash Airways	1	Warsaw Chopin Airport
Yes	324.7006530761719	360	581.6918245765773	No	ES-Air	6	Oslo Gardermoen Airport
Yes	324.66265869140625	360	817.3689521751049	No	JetBeats	6	Syracuse Hancock International Airport
Yes	323.8282470703125	360	242.37729470102494	No	Kibana Airlines	4	Tokyo Haneda International Airport
Yes	323.7989501953125	360	209.33309339402115	No	Kibana Airlines	2	Warsaw Chopin Airport
Yes	323.59625244140625	360	149.32434581984649	No	JetBeats	2	Xian Xianyang International Airport
Yes	323.59625244140625	360	254.94826538904468	No	Logstash Airways	5	Xian Xianyang International Airport
Yes	322.6275280761719	360	375.5430524836813	No	ES-Air	6	Vienna International Airport

**ml.FlightDelay\_predict** es nuestra predicción. Y podemos comparar con el valor real en la siguiente columna, **"FlightDelayMin"**.

Podemos filtrar sólo datos de training, o de test mediante

Testing Training

Analytics exploration EXPERIMENTAL

Evaluation of regression job ID flight\_delays stopped Regression evaluation docs

Generalization error	Training error
2,560 docs evaluated	0 docs evaluated
<b>3670</b>	<b>--</b>
Mean squared error ⓘ	Mean squared error ⓘ
<b>0.603</b>	<b>--</b>
R squared ⓘ	R squared ⓘ

Destination index for regression job ID flight\_delays stopped 8 of 31 fields selected

Showing first 1000 documents for which predictions exist

Testing Training

ml.is_training	ml.FlightDelayMin_predict...	FlightDelayMin	AvgTicketPrice	Cancelled	Carrier	dayOfWeek	Dest
No	317.09881591796875	345	1117.2940242419768	No	Logstash Airways	6	Venice Marco Polo Airport
No	314.57574462890625	345	1008.5355434119128	No	Logstash Airways	6	Verona Villafranca Airport
No	312.1139831542969	285	1146.2926285027595	No	ES-Air	5	Malpensa International Airport
No	310.21380615234375	360	963.1108440752872	No	ES-Air	2	Kempegowda International Airport

Para comprobar la bondad del modelo<sup>4</sup> tenemos los datos en la cabecera.

Evaluation of regression job ID flight\_delays stopped Regression evaluation docs

Generalization error	Training error
2,560 docs evaluated	10,499 docs evaluated
<b>3670</b>	<b>2920</b>
Mean squared error ⓘ	Mean squared error ⓘ
<b>0.603</b>	<b>0.689</b>
R squared ⓘ	R squared ⓘ

4

<https://www.elastic.co/guide/en/machine-learning/7.6/ml-dfanalytics-evaluate.html#ml-dfanalytics-regression-evaluation>

## Machine Learning Supervisado - Clasificación

También podemos usar un modelo de clasificación para identificar retrasos. De forma similar, creamos un job de tipo clasificación.

×

Create analytics job

Job type

classification

Classification jobs require a source index that is mapped as a table-like data structure and supports fields that are numeric, boolean, text, keyword or ip. Please use the advanced editor to apply custom options such as the prediction field name.

☐ Enable advanced editor

You cannot switch back to this form from the advanced editor.

Job ID

flight\_delays\_classification

Job description

Optional descriptive text

Source index

kibana\_sample\_data\_flights

Destination index

results-flight\_delays-classification

Dependent variable

FlightDelay

Training percent

0 80 100

Excluded fields

Cancelled ×

FlightDelayMin ×

FlightDelayType ×

×

↓

Optionally select fields to be excluded from analysis. All other supported fields will be included

Model memory limit

106569kb

☐ ×

Create index pattern

Guardar y arrancar el job. Y podemos visualizar. Este job tardará unos minutos.



## Analytics exploration EXPERIMENTAL

Evaluation of classification job ID `flight_delays_classification` stopped

[Classification evaluation docs](#)

Normalized confusion matrix <sup>Ⓢ</sup>

13,059 docs evaluated

		Predicted label	
		false	true
Actual label	False	87%	13%
	True	15%	85%

Destination index for classification job ID `flight_delays_classification` stopped

8 of 31 fields selected <sup>Ⓢ</sup>

Showing first 1000 documents for which predictions exist

Q E.g. avg>0.5

Testing Training

ml.is_training	ml.FlightDelay_prediction ↓	FlightDelay	AvgTicketPrice	Cancelled	Carrier	dayOfWeek	Dest
Yes	Yes	Yes	712.9919051086614	No	Kibana Airlines	0	Turin Airport
Yes	Yes	Yes	574.9962482490664	Yes	Logstash Airways	1	Winnipeg / James Armstrong Richardson International Airport
Yes	Yes	Yes	964.6662147056691	No	Logstash Airways	5	Sheremetyevo International Airport
Yes	Yes	Yes	636.5628472776802	No	ES-Air	6	Genoa Cristoforo Colombo Airport
Yes	Yes	Yes	722.547338592276	No	ES-Air	1	El Dorado International Airport
Yes	Yes	No	1181.9821281122495	No	Kibana Airlines	6	Edmonton International Airport
Yes	Yes	Yes	535.4327983537714	No	ES-Air	2	Rajiv Gandhi International Airport

Para evaluar la bondad de job, tenemos en la cabecera de los resultados la matriz de confusión <sup>5</sup>, que nos indica falsos y verdaderos positivos y negativos.

De nuevo, podemos separar los datos de entrenamiento de los de training:

## Analytics exploration EXPERIMENTAL

Evaluation of classification job ID `flight_delays_classification` stopped

[Classification evaluation docs](#)

Normalized confusion matrix <sup>Ⓢ</sup>

2,612 docs evaluated

		Predicted label	
		false	true
Actual label	False	82%	18%
	True	25%	75%

Destination index for classification job ID `flight_delays_classification` stopped

8 of 31 fields selected <sup>Ⓢ</sup>

Showing first 1000 documents for which predictions exist

Q ml.is\_training=false

Testing Training

ml.is_training	ml.FlightDelay_prediction ↓	FlightDelay	AvgTicketPrice	Cancelled	Carrier	dayOfWeek	Dest
No	Yes	Yes	592.8260663956739	No	ES-Air	2	Treviso-Sant'Angelo Airport
No	Yes	Yes	361.76765888695354	Yes	Logstash Airways	6	Dubai International Airport
No	Yes	Yes	648.9550719685651	No	Logstash Airways	5	Zurich Airport
No	Yes	Yes	808.307379630997	No	JetBeats	4	Turin Airport

## Inferencia

Finalmente, cuando tenemos un modelo entrenado, lo que queremos es usarlo en datos nuevos, para aplicar esa predicción. Aquí entra en juego el procesador de inferencia<sup>6</sup>, que se puede especificar dentro de una pipeline de ingesta para enriquecer documentos.

Aquí veremos una simulación de esta pipeline de ingesta, que permite enriquecer documentos. Para ello, obtenemos primero los modelos existentes en nuestros Elasticsearch:

```
GET _ml/inference/?filter_path=trained_model_configs.model_id
```

En mi ejemplo, me devuelve estos 3:

```
{
  "trained_model_configs" : [
    {
      "model_id" : "flight_delays-1588596108197"
    },
    {
      "model_id" : "flight_delays_classification-1588596958454"
    },
    {
      "model_id" : "lang_ident_model_1"
    }
  ]
}
```

Para predecir el retraso de un nuevo vuelo, pasamos los datos de un vuelo, **no etiquetado** como retrasado o no, por una pipeline de ingesta un vuelo ejemplo, usando el modelo de regresión **"flight\_delays-1588596108197"**:

```
POST _ingest/pipeline/_simulate?filter_path=docs.doc._source
{
  "pipeline": {
    "processors": [
      {
        "inference": {
          "model_id": "flight_delays-1588596108197",
          "inference_config": {
            "regression": {
              "results_field": "PredictedFlightDelay"
            }
          },
          "field_mappings": {}
        }
      }
    ]
  },
  "docs": [
```

<sup>6</sup> <https://www.elastic.co/guide/en/machine-learning/7.6/ml-inference.html>

```
{
  "_source": {
    "FlightNum": "OODIP58",
    "DestCountry": "CN",
    "OriginWeather": "Thunder & Lightning",
    "OriginCityName": "Abu Dhabi",
    "AvgTicketPrice": 252.9119662217096,
    "DistanceMiles": 3032.4467769272865,
    "DestWeather": "Sunny",
    "Dest": "Chengdu Shuangliu International Airport",
    "OriginCountry": "AE",
    "dayOfWeek": 0,
    "DistanceKilometers": 4880.250025767267,
    "timestamp": "2020-02-10T12:05:14",
    "DestLocation": {
      "lat": "30.57850075",
      "lon": "103.9469986"
    },
    "DestAirportID": "CTU",
    "Carrier": "Kibana Airlines",
    "FlightTimeMin": 490.3500017178178,
    "Origin": "Abu Dhabi International Airport",
    "OriginLocation": {
      "lat": "24.43300056",
      "lon": "54.65110016"
    },
    "DestRegion": "SE-BD",
    "OriginAirportID": "AUH",
    "OriginRegion": "SE-BD",
    "DestCityName": "Chengdu"
  }
}
```

Que nos añadirá a ese documento la predicción del tiempo de retraso de este vuelo:

```
"ml" : {
  "inference" : {
    "PredictedFlightDelay" : 157.03950906533115,
    "model_id" : "flight_delays-1588596108197"
  }
},
```

De forma similar se puede realizar usando el modelo de clasificación:

```
POST _ingest/pipeline/_simulate?
{
  "pipeline": {
    "processors": [
      {
        "inference": {
          "model_id": "flight_delays_classification-1588596958454",
          "inference_config": {
            "classification": {
              "num_top_classes": 1
            }
          },
        },
      },
    ],
    "field_mappings": {}
  }
}
```

```
    }
  }
},
"docs": [
  {
    "_source": {
      "FlightNum": "OODIP58",
      "DestCountry": "CN",
      "OriginWeather": "Thunder & Lightning",
      "OriginCityName": "Abu Dhabi",
      "AvgTicketPrice": 252.9119662217096,
      "DistanceMiles": 3032.4467769272865,
      "DestWeather": "Sunny",
      "Dest": "Chengdu Shuangliu International Airport",
      "OriginCountry": "AE",
      "dayOfWeek": 0,
      "DistanceKilometers": 4880.250025767267,
      "timestamp": "2020-02-10T12:05:14",
      "DestLocation": {
        "lat": "30.57850075",
        "lon": "103.9469986"
      },
      "DestAirportID": "CTU",
      "Carrier": "Kibana Airlines",
      "FlightTimeMin": 490.3500017178178,
      "Origin": "Abu Dhabi International Airport",
      "OriginLocation": {
        "lat": "24.43300056",
        "lon": "54.65110016"
      },
      "DestRegion": "SE-BD",
      "OriginAirportID": "AUH",
      "OriginRegion": "SE-BD",
      "DestCityName": "Chengdu"
    }
  }
]
```

Que en este caso predice retraso:

```
"ml" : {
  "inference" : {
    "top_classes" : [
      {
        "class_name" : "1",
        "class_probability" : 0.7475580543324547,
        "class_score" : 1.3345643529928286
      }
    ],
    "predicted_value" : "1",
    "model_id" : "flight_delays_classification-1588596958454"
  }
},
```

Adicionalmente, Elasticsearch lleva precargado un modelo para identificar el idioma de un documento. Probamos con 3 documentos:

```
POST _ingest/pipeline/_simulate?filter_path=docs.doc._source.text,docs.doc._source.ml.inference.top_classes
{
  "pipeline":{
    "processors":[
      {
        "inference":{
          "model_id":"lang_ident_model_1",
          "inference_config":{
            "classification":{
              "num_top_classes":1
            }
          },
          "field_mappings":{
            }
          }
        ]
      },
      "docs":[
        {
          "_source":{
            "text":"Con frecuencia usamos arquitecturas hot-warm para sacar el máximo provecho de nuestro hardware. Son particularmente útiles cuando tenemos datos basados en el tiempo, como registros, métricas y datos de APM. La mayoría de estas configuraciones se basan en el hecho de que estos datos son de sólo lectura (después de la ingesta) y que los índices pueden estar basados en el tiempo (o el tamaño). Por lo tanto, se pueden eliminar con facilidad según nuestro período de retención deseado. Con este tipo de arquitectura, categorizamos los nodos de Elasticsearch en dos tipos: "hot" y "warm"."
```

hardware. Son particularmente útiles cuando tenemos datos basados en el tiempo, como registros, métricas y datos de APM. La mayoría de estas configuraciones se basan en el hecho de que estos datos son de sólo lectura (después de la ingesta) y que los índices pueden estar basados en el tiempo (o el tamaño). Por lo tanto, se pueden eliminar con facilidad según nuestro período de retención deseado. Con este tipo de arquitectura, categorizamos los nodos de Elasticsearch en dos tipos: "hot" y "warm".

```
          }
        },
        {
          "_source":{
            "text":"Nous avons souvent recours aux architectures hot-warm lorsque nous voulons tirer pleinement parti de notre matériel. Celles-ci sont particulièrement utiles lorsque nous disposons de données temporelles, comme les logs, les indicateurs et les données APM. Pour la plupart des configurations, les données sont en lecture seule (après l'ingestion) et les index sont basés sur une durée (ou une taille). Il est donc facile de les supprimer selon la durée pendant laquelle nous souhaitons les conserver. Dans ce type d'architecture, nous classons les nœuds Elasticsearch en deux catégories : «hot» et «warm»."
```

Nous avons souvent recours aux architectures hot-warm lorsque nous voulons tirer pleinement parti de notre matériel. Celles-ci sont particulièrement utiles lorsque nous disposons de données temporelles, comme les logs, les indicateurs et les données APM. Pour la plupart des configurations, les données sont en lecture seule (après l'ingestion) et les index sont basés sur une durée (ou une taille). Il est donc facile de les supprimer selon la durée pendant laquelle nous souhaitons les conserver. Dans ce type d'architecture, nous classons les nœuds Elasticsearch en deux catégories : «hot» et «warm».

```
          }
        },
        {
          "_source":{
            "text":"Hot-Warm architectures are often used when we want to get the most out of our hardware. It is particularly useful when we have time-based data, like logs, metrics, and APM data. Most of these setups rely on the fact that this data is read-only (after ingest) and that indices can be time(or size)-based. So they can be easily deleted based on our desired retention period. In this architecture, we categorize Elasticsearch nodes into two types: 'hot' and 'warm'."
```

Hot-Warm architectures are often used when we want to get the most out of our hardware. It is particularly useful when we have time-based data, like logs, metrics, and APM data. Most of these setups rely on the fact that this data is read-only (after ingest) and that indices can be time(or size)-based. So they can be easily deleted based on our desired retention period. In this architecture, we categorize Elasticsearch nodes into two types: 'hot' and 'warm'.


```
          }
        ]
      }
    }
  }
```

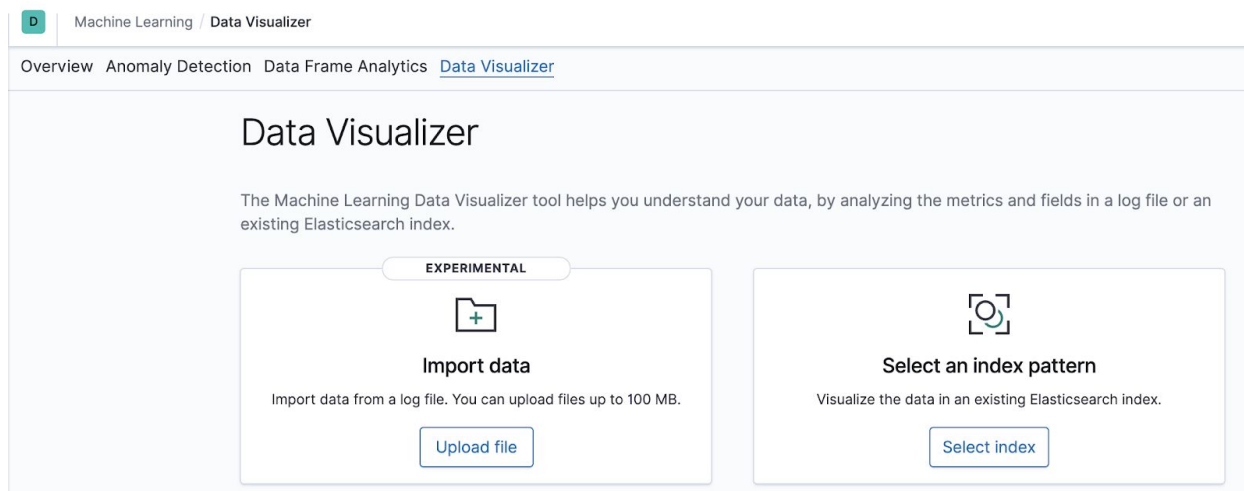
Con resultado:

```
{
  "docs" : [
    {
      "doc" : {
        "_source" : {
          "text" : "Con frecuencia usamos arquitecturas hot-warm para sacar el máximo provecho de nuestro hardware. Son particularmente útiles cuando tenemos datos basados en el tiempo, como registros, métricas y datos de APM. La mayoría de estas configuraciones se basan en el hecho de que estos datos son de sólo lectura (después de la ingesta) y que los índices pueden estar basados en el tiempo (o el tamaño). Por lo tanto, se pueden eliminar con facilidad según nuestro periodo de retención deseado. Con este tipo de arquitectura, categorizamos los nodos de Elasticsearch en dos tipos: "hot" y "warm".",
          "ml" : {
            "inference" : {
              "top_classes" : [
                {
                  "class_name" : "es",
                  "class_probability" : 0.9999983342656864,
                  "class_score" : 0.9999983342656864
                }
              ]
            }
          }
        }
      },
      {
        "doc" : {
          "_source" : {
            "text" : "Nous avons souvent recours aux architectures hot-warm lorsque nous voulons tirer pleinement parti de notre matériel. Celles-ci sont particulièrement utiles lorsque nous disposons de données temporelles, comme les logs, les indicateurs et les données APM. Pour la plupart des configurations, les données sont en lecture seule (après l'ingestion) et les index sont basés sur une durée (ou une taille). Il est donc facile de les supprimer selon la durée pendant laquelle nous souhaitons les conserver. Dans ce type d'architecture, nous classons les nœuds Elasticsearch en deux catégories : «hot» et «warm».",
            "ml" : {
              "inference" : {
                "top_classes" : [
                  {
                    "class_name" : "fr",
                    "class_probability" : 0.9999994706546358,
                    "class_score" : 0.9999994706546358
                  }
                ]
              }
            }
          }
        },
        {
          "doc" : {
            "_source" : {
              "text" : "Hot-Warm architectures are often used when we want to get the most out of our hardware. It is particularly useful when we have time-based data, like logs, metrics, and APM data. Most of these setups rely on the fact that this data is read-only (after ingest) and that indices can be time(or size)-based. So they can be easily deleted based on our desired retention period. In this architecture, we categorize Elasticsearch nodes into two types: 'hot' and 'warm'.",
              "ml" : {
                "inference" : {
                  "top_classes" : [
                    {
                      "class_name" : "en",
                      "class_probability" : 0.9999963603185801,
```

```
    "class_score" : 0.9999963603185801
  }
}
}
```

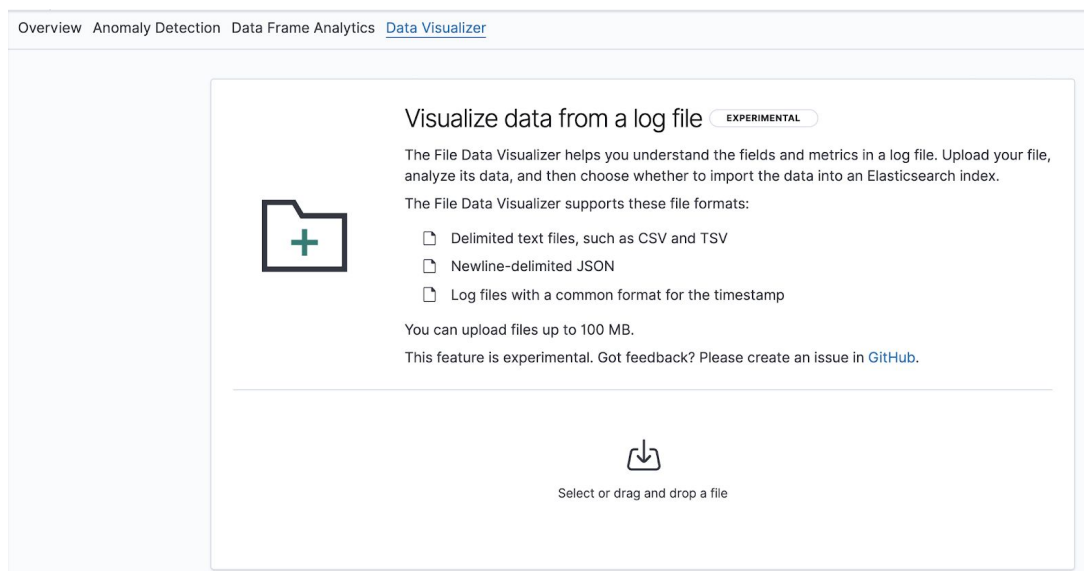
## Demo 3. Carga de un fichero JSON/CSV

Machine Learning nos ofrece una forma de cargar datos a partir de ficheros. Pulsando en el menú de la izquierda  **Machine Learning**, seleccionaremos en el menú superior “**Data Visualizer**”



Seleccionamos en este caso

[Upload file](#)





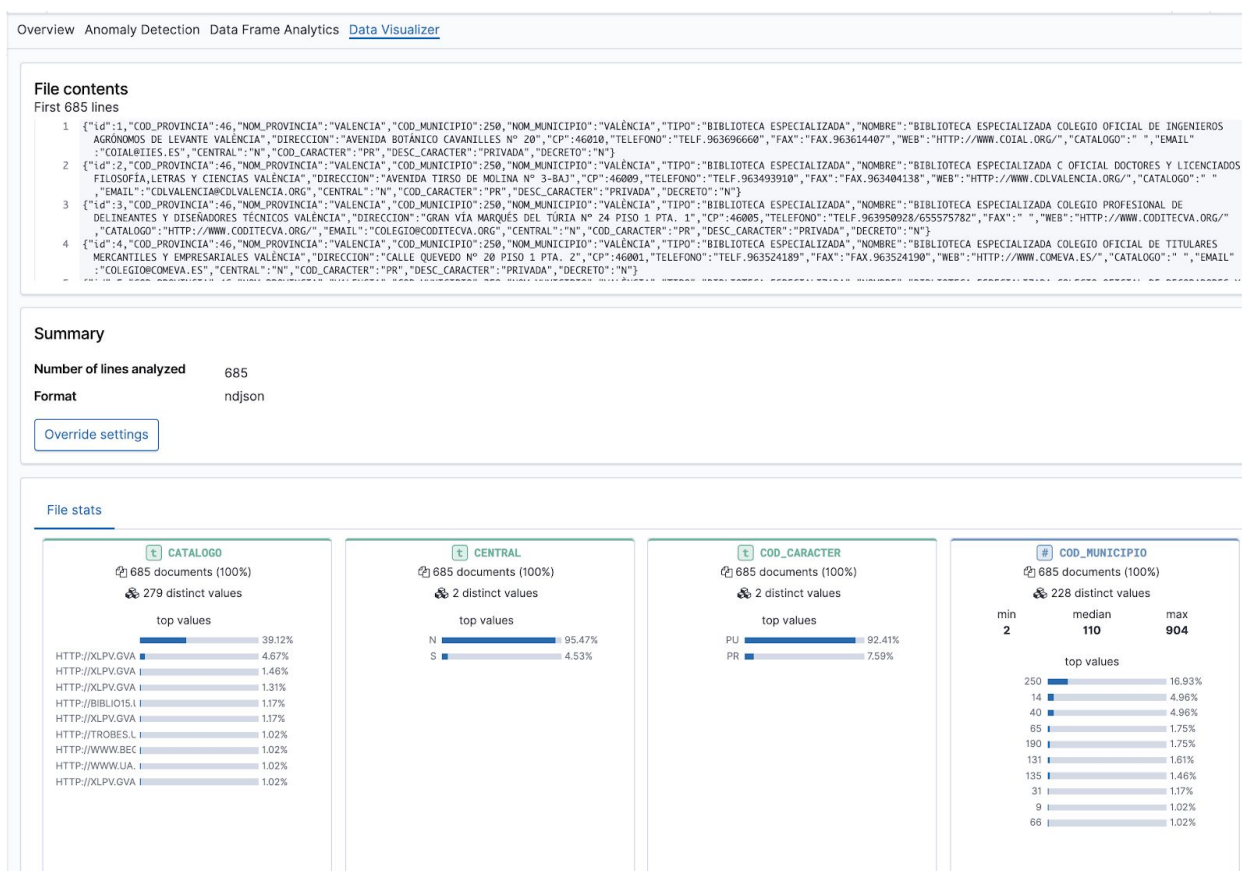


Y en Select or drag and drop a file seleccionamos el fichero que tenemos en el proyecto

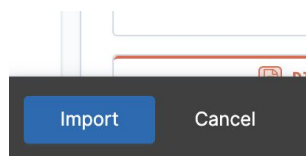
viu-elk-ml-talk/doc/dataset/bibliotecas-comunidad-valenciana-2020.ndjson. Se trata del listado de bibliotecas de la Comunitat Valenciana en 2020 obtenido de <https://dadesobertes.gva.es/es/dataset/cul-dir-bibliotecas-2020>, que hemos convertido a formato **ndjson**.

El visualizador reconoce los campos, y nos permite realizar cambios si deseamos, mediante

Override settings



Para esta prueba simplemente pulsamos en la parte inferior izquierda del navegador **“Import”**. Sin modificar nada.



Nos pedirá el nombre del índice a crear, por ejemplo, “*bibliotecas-valencia-mi*”. E importamos.

Comprobamos que ha importado los datos:

Machine Learning / Data Visualizer / File

[Overview](#)
[Anomaly Detection](#)
[Data Frame Analytics](#)
[Data Visualizer](#)

Import data

EXPERIMENTAL

Simple

Advanced

Index name

bibliotecas-valencia-ml

Create index pattern

Reset

File processed

Index created

Data uploaded

Index pattern created

Import complete

Index	bibliotecas-valencia-ml
Index pattern	bibliotecas-valencia-ml
Documents ingested	685

View index in Discover

Open in Data Visualizer

Index Management

Index Pattern Management

Y a partir de aquí podemos ir a Discover, u otras áreas de Kibana, para comprobar que se han creado estos 685 documentos.

**Discover**

New Save Open Share Inspect

Search KQL Refresh

+ Add filter

**bibliotecas-valencia-ml** 685 hits

Search field names

Filter by type 0

**Selected fields**

- \_source

**Available fields**

- CATALOGO
- CENTRAL
- COD\_CARACTER
- COD\_MUNICIPIO
- COD\_PROVINCIA
- CP

```
{
  "id": 1,
  "COD_PROVINCIA": 46,
  "NOM_PROVINCIA": "VALENCIA",
  "COD_MUNICIPIO": 250,
  "NOM_MUNICIPIO": "VALENCIA",
  "TIPO": "BIBLIOTECA ESPECIALIZADA",
  "NOMBRE": "BIBLIOTECA ESPECIALIZADA COLEGIO OFICIAL DE INGENIEROS AGRONOMOS DE LEVANTE VALENCIA",
  "DIRECCION": "AVENIDA BOTANICO CAVANILLES N° 20",
  "CP": 46,010,
  "TELEFONO": "TEL: 963696660",
  "FAX": "FAX: 963614407",
  "WEB": "HTTP://WWW.COITAL.ORG/",
  "CATALOGO": "EMAIL: COITAL@IES.ES",
  "CENTRAL": "N",
  "COD_CARACTER": "PR",
  "DESC_CARACTER": "PRIVADA",
  "DECRETO": "N",
  "_id": "RJS33EBswK7Ra1qEj",
  "_type": "_doc",
  "_index": "bibliotecas-valencia-ml",
  "_score": 0
}
```

```
{
  "id": 2,
  "COD_PROVINCIA": 46,
  "NOM_PROVINCIA": "VALENCIA",
  "COD_MUNICIPIO": 250,
  "NOM_MUNICIPIO": "VALENCIA",
  "TIPO": "BIBLIOTECA ESPECIALIZADA",
  "NOMBRE": "BIBLIOTECA ESPECIALIZADA C OFICIAL DOCTORES Y LICENCIADOS FILOSOFIA, LETRAS Y CIENCIAS VALENCIA",
  "DIRECCION": "AVENIDA TIRSO DE MOLINA N° 3-BAJ",
  "CP": 46,009,
  "TELEFONO": "TEL: 963493918",
  "FAX": "FAX: 963404138",
  "WEB": "HTTP://WWW.CDLVALENCIA.ORG/",
  "CATALOGO": "EMAIL: CDLVALENCIA@CDLVALENCIA.ORG",
  "CENTRAL": "N",
  "COD_CARACTER": "PR",
  "DESC_CARACTER": "PRIVADA",
  "DECRETO": "N",
  "_id": "RZSS33EBswK7Ra1qEj",
  "_type": "_doc",
  "_index": "bibliotecas-valencia-ml",
  "_score": 0
}
```

```
{
  "id": 3,
  "COD_PROVINCIA": 46,
  "NOM_PROVINCIA": "VALENCIA",
  "COD_MUNICIPIO": 250,
  "NOM_MUNICIPIO": "VALENCIA",
  "TIPO": "BIBLIOTECA ESPECIALIZADA",
  "NOMBRE": "BIBLIOTECA ESPECIALIZADA COLEGIO PROFESIONAL DE DELINCUENTES Y DISEÑADORES TÉCNICOS VALENCIA",
  "DIRECCION": "GRAN VÍA MARQUÉS DEL TURIA N° 24 PISO 1",
  "PTA": 1,
  "CP": 46,005,
  "TELEFONO": "TEL: 963959828/65557582",
  "FAX": "FAX: ",
  "WEB": "HTTP://WWW.CODITECVA.ORG/",
  "CATALOGO": "HTTP://WWW.CODITECVA.ORG/",
  "EMAIL": "COLEGIO@CODITECVA.ORG",
  "CENTRAL": "N",
  "COD_CARACTER": "PR",
  "DESC_CARACTER": "PRIVADA",
  "DECRETO": "N",
  "_id": "RpS333EBswK7Ra1qEj",
  "_type": "_doc",
  "_index": "bibliotecas-valencia-ml",
  "_score": 0
}
```

## Gist

<https://gist.github.com/immavalls/aecf1cc361e3fcb957aed711c340e712>