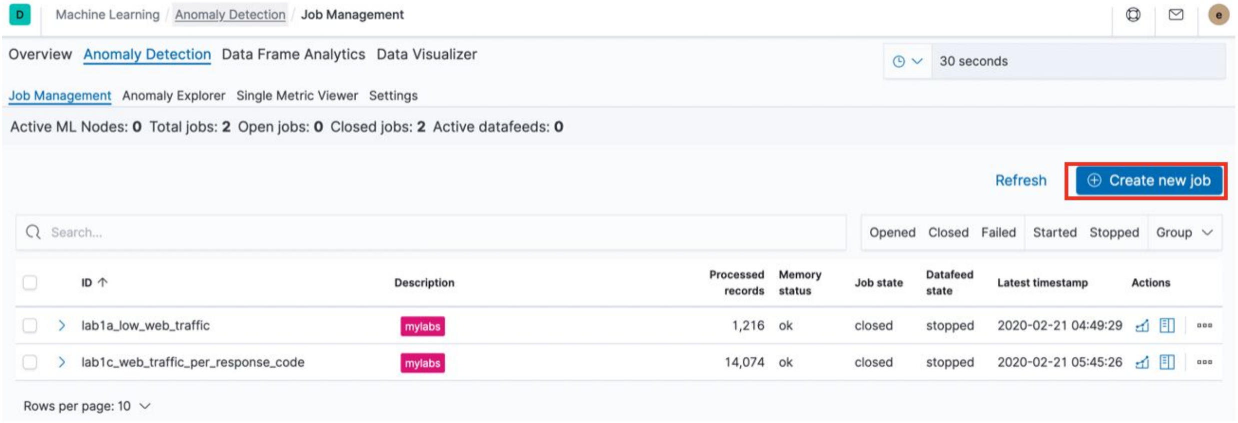
Lab 3 - Population Analysis, Data Frame & Outlier Detection

In this lab, we will be performing the following on sample eCommerce data:

a. Set up a population analysis job to find unusual customers
b. Use Dataframe Transform to create a customer entity-centric index
c. Use Machine Learning's Outlier Detection to find unusual customers

A - Set Up a Population Analysis Job

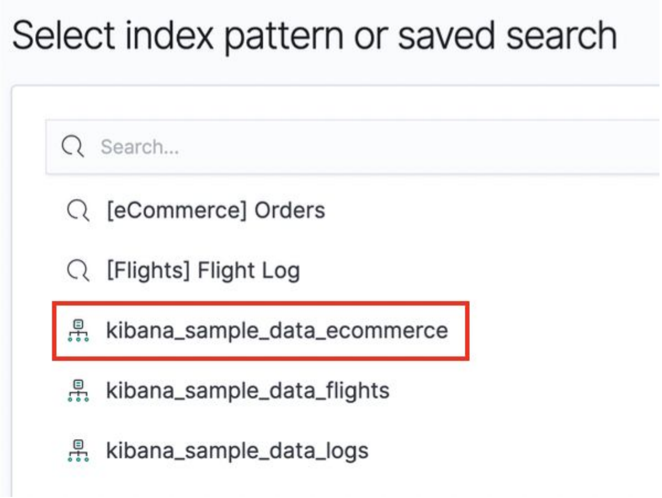
1. Click on Kibana > Machine Learning > "Anomaly Detection"
2. Click on the link to "Create new job"



Active ML Nodes: **0** Total jobs: **2** Open jobs: **0** Closed jobs: **2** Active datafeeds: **0**

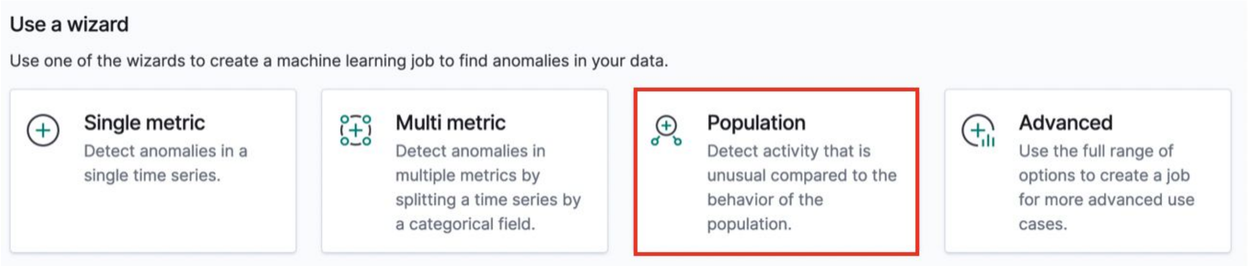
ID ↑	Description	Processed records	Memory status	Job state	Datafeed state	Latest timestamp	Actions
lab1a_low_web_traffic	mylabs	1,216	ok	closed	stopped	2020-02-21 04:49:29	
lab1c_web_traffic_per_response_code	mylabs	14,074	ok	closed	stopped	2020-02-21 05:45:26	

3. Select the "kibana_sample_data_ecommerce" index.


Select index pattern or saved search

[eCommerce] Orders
[Flights] Flight Log
kibana_sample_data_ecommerce
kibana_sample_data_flights
kibana_sample_data_logs

4. Click on the link to create a Population Analysis job

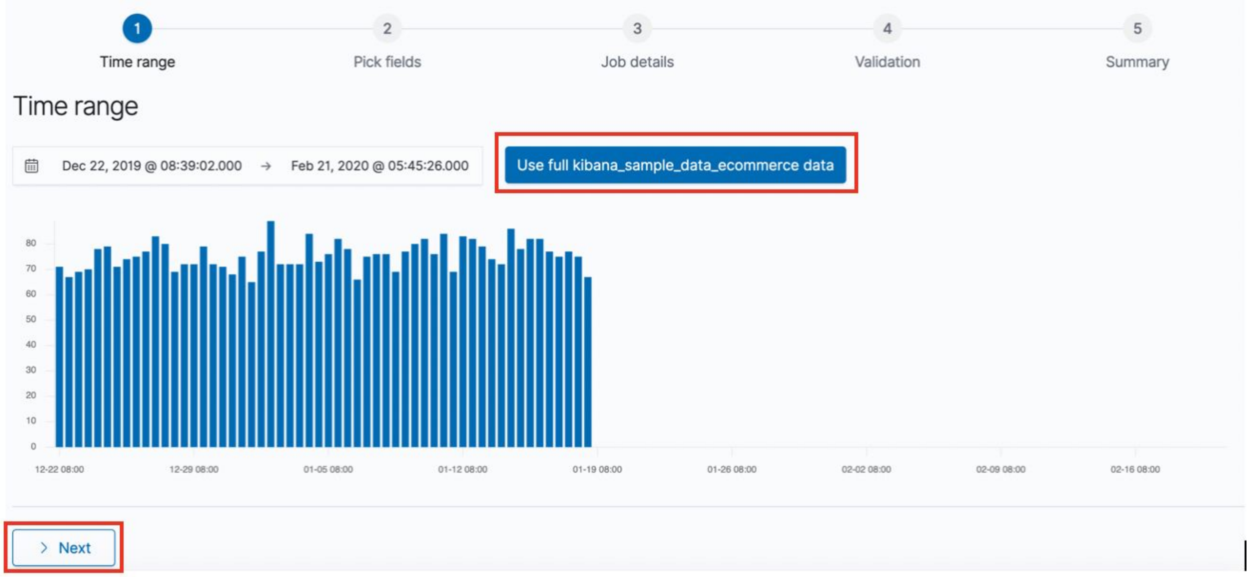


Use a wizard

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.

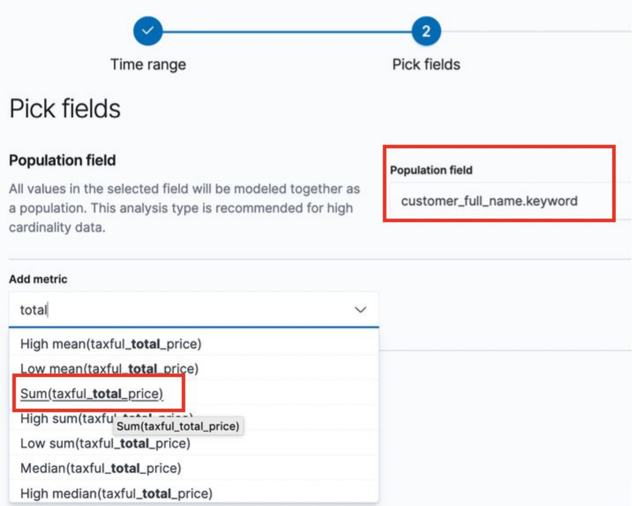
5. Click on the “Use full kibana_sample_data_ecommerce data” button, then click on “Next”



Time range

Dec 22, 2019 @ 08:39:02.000 → Feb 21, 2020 @ 05:45:26.000

6. Use "customer_full_name.keyword" as the Population field
Add Metric: use "sum" on the field "total_taxful_price"



Pick fields

Population field

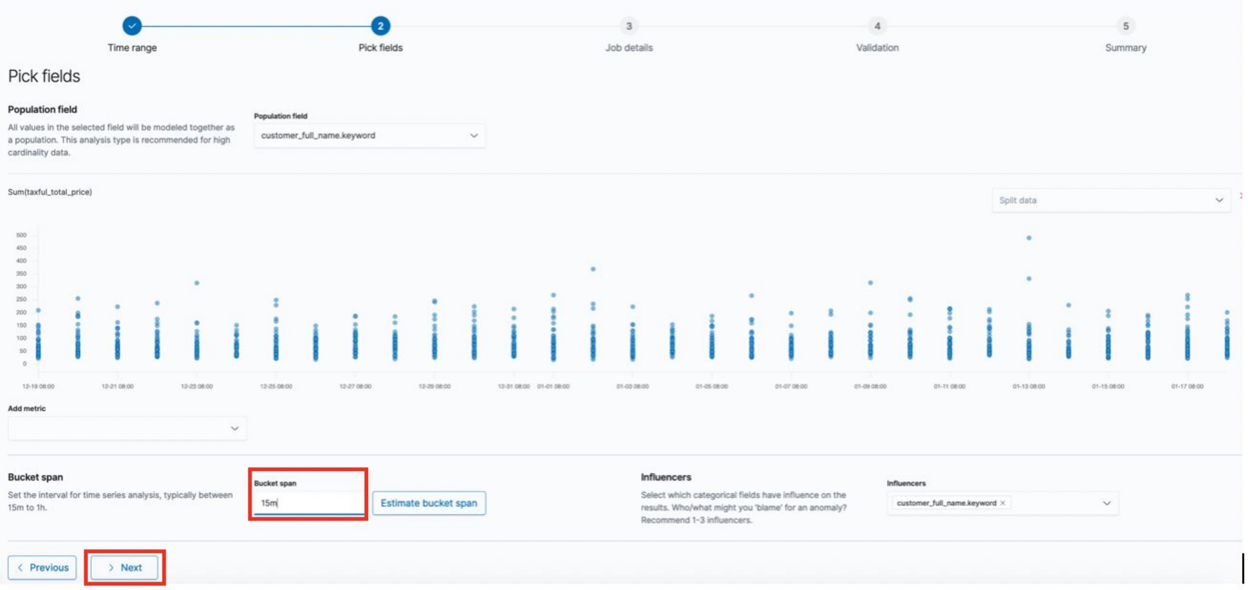
All values in the selected field will be modeled together as a population. This analysis type is recommended for high cardinality data.

Population field: customer_full_name.keyword

Add metric: total

High mean(taxful_**total**_price)
Low mean(taxful_**total**_price)
Sum(taxful_**total**_price)
High sum(taxful_**total**_price)
Sum(taxful_total_price)
Low sum(taxful_**total**_price)
Median(taxful_**total**_price)
High median(taxful_**total**_price)

7. Use 15m bucket span and click on “Next”



Pick fields

Population field

All values in the selected field will be modeled together as a population. This analysis type is recommended for high cardinality data.

Population field: customer_full_name.keyword

Add metric

Bucket span

Set the interval for time series analysis, typically between 15m to 1h.

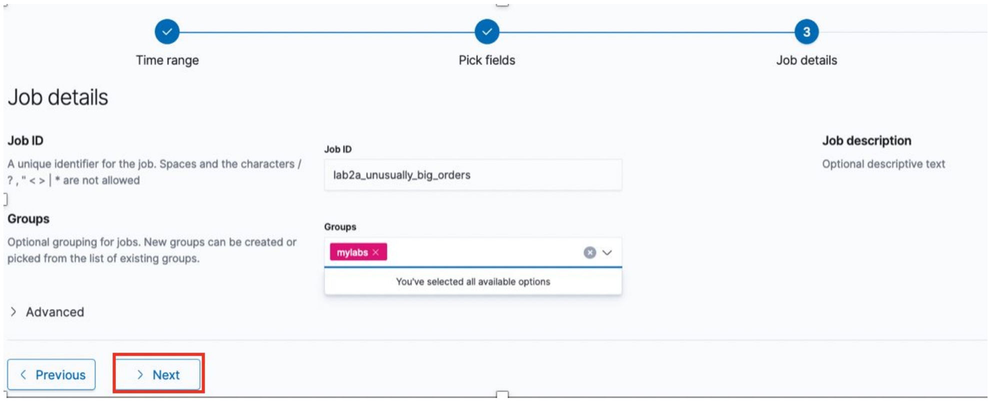
Bucket span: 15m

Influencers

Select which categorical fields have influence on the results. Who/what might you 'blame' for an anomaly? Recommend 1-3 influencers.

Influencers: customer_full_name.keyword ×

8. Name the job "lab3a_unusually_big_orders" and place it under “mylabs” group



Job details

Job ID

A unique identifier for the job. Spaces and the characters / ? , " < > | * are not allowed

Job ID: lab2a_unusually_big_orders

Groups

Optional grouping for jobs. New groups can be created or picked from the list of existing groups.

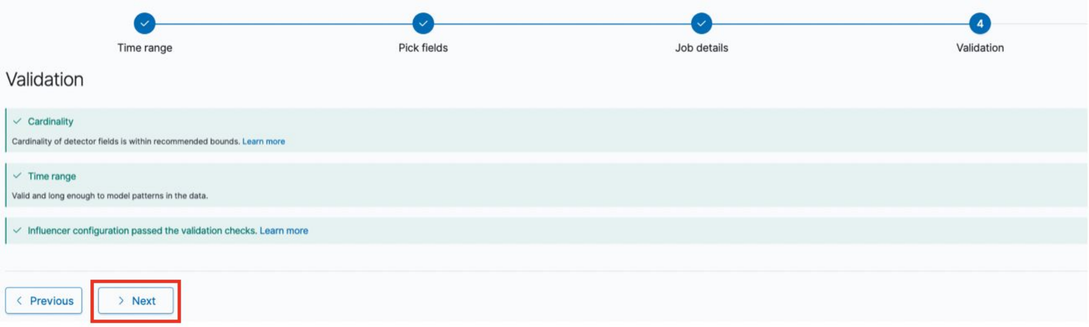
Groups: mylabs

You've selected all available options

Job description

Optional descriptive text

9. Click "Next" after passing the job validation



Validation

✓ Cardinality

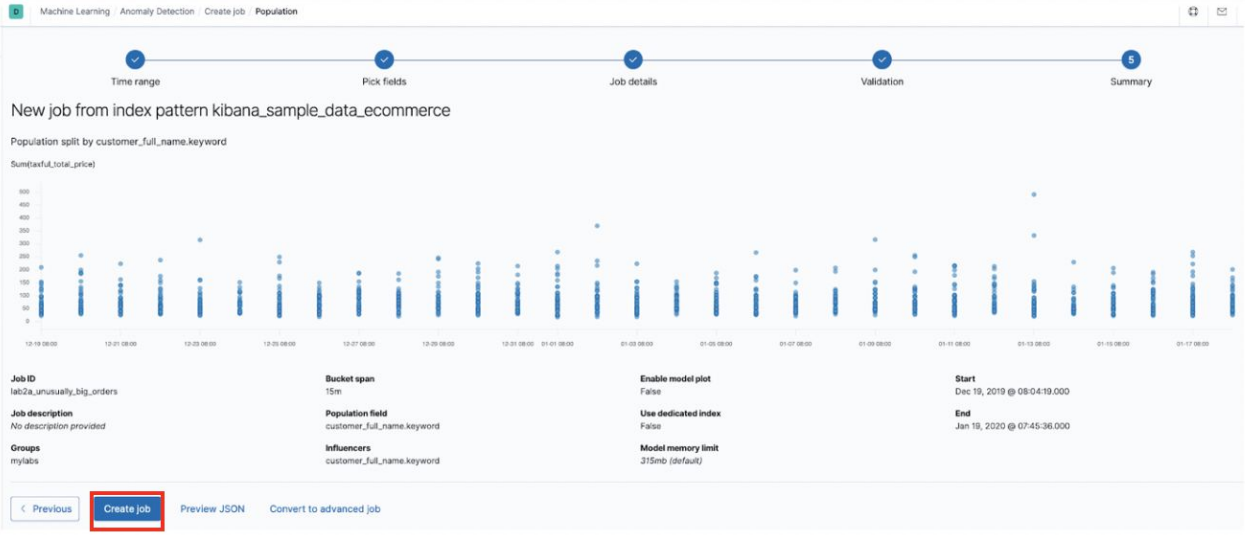
Cardinality of detector fields is within recommended bounds. Learn more

✓ Time range

Valid and long enough to model patterns in the data.

✓ Influencer configuration passed the validation checks. Learn more

10. Review the Job Settings and click on the "Create job" link to start the job


D
Machine Learning / Anomaly Detection / Create job / Population
Time range
Pick fields
Job details
Validation
5
Summary

New job from index pattern kibana_sample_data_ecommerce

Population split by customer_full_name.keyword

Job ID
lab2a_unusually_big_orders

Job description
No description provided

Groups
mylabs

Bucket span
15m

Population field
customer_full_name.keyword

Influencers
customer_full_name.keyword

Enable model plot
False

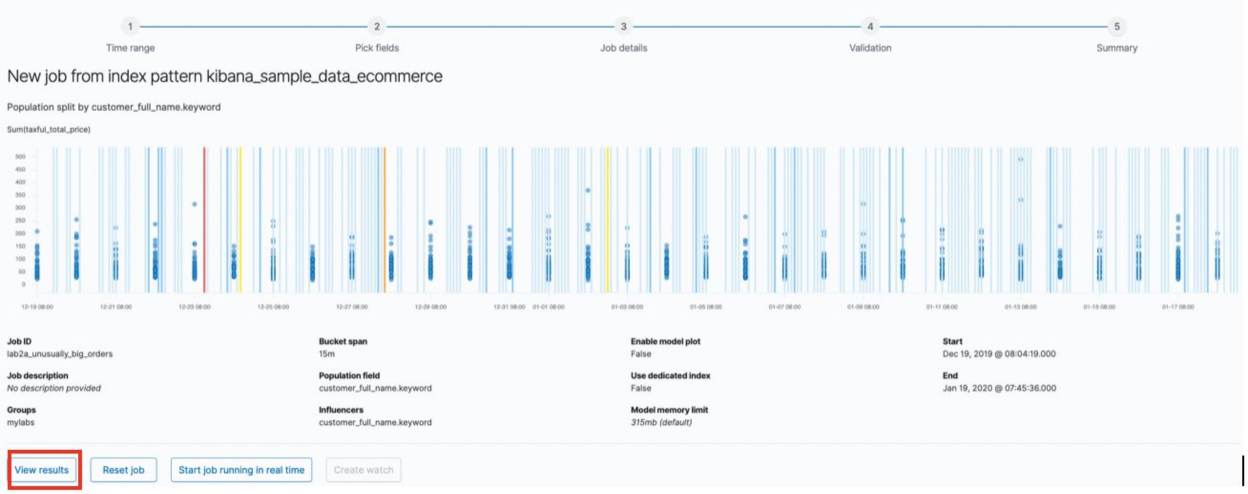
Use dedicated index
False

Model memory limit
315mb (default)

Start
Dec 19, 2019 @ 08:04:19.000

End
Jan 19, 2020 @ 07:45:36.000

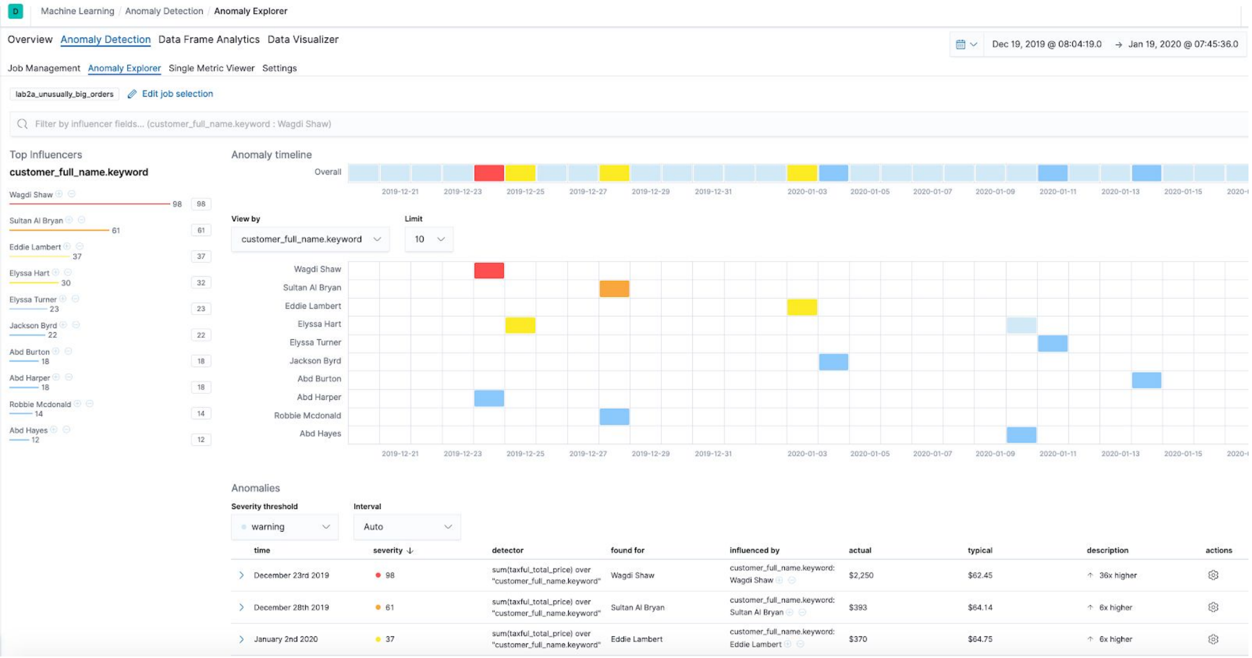
11. Wait around half a minute while the job completes. Click on the “View Results” link to review the results



New job from index pattern kibana_sample_data_ecommerce

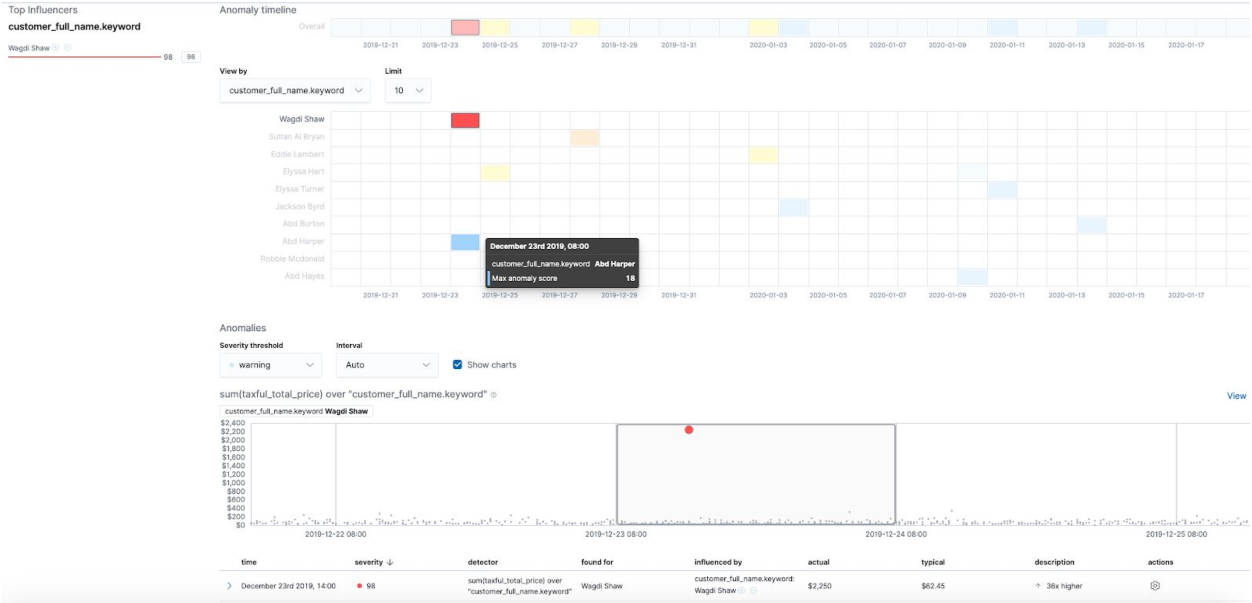
Job ID lab2a_unusually_big_orders	**Bucket span** 15m	**Enable model plot** False	**Start** Dec 19, 2019 @ 08:04:19.000
Job description *No description provided*	**Population field** customer_full_name.keyword	**Use dedicated index** False	**End** Jan 19, 2020 @ 07:45:36.000
Groups mylabs	**Influencers** customer_full_name.keyword	**Model memory limit** *315mb (default)*	

12. Customer "Wagdi Shaw" is the highest anomalous customer



time	severity ↓	detector	found for	influenced by	actual	typical	description	actions
December 23rd 2019	● 98	sum(taxful_total_price) over "customer_full_name.keyword"	Wagdi Shaw	customer_full_name.keyword: Wagdi Shaw	$2,250	$62.45	↑ 36x higher	⚙
December 28th 2019	● 61	sum(taxful_total_price) over "customer_full_name.keyword"	Sultan Al Bryan	customer_full_name.keyword: Sultan Al Bryan	$393	$64.14	↑ 6x higher	⚙
January 2nd 2020	● 37	sum(taxful_total_price) over "customer_full_name.keyword"	Eddie Lambert	customer_full_name.keyword: Eddie Lambert	$370	$64.75	↑ 6x higher	⚙

13. Explore Anomalous User -
Click on red tile for user "Wagdi Shaw"



View by: customer_full_name.keyword

Limit: 10

December 23rd 2019, 08:00
customer_full_name.keyword **Abd Harper**
Max anomaly score 18

Severity threshold: warning

Interval: Auto

☑ Show charts

sum(taxful_total_price) over "customer_full_name.keyword"

customer_full_name.keyword **Wagdi Shaw**

time	severity ↓	detector	found for	influenced by	actual	typical	description	actions
December 23rd 2019, 14:00	● 98	sum(taxful_total_price) over "customer_full_name.keyword"	Wagdi Shaw	customer_full_name.keyword: Wagdi Shaw	$2,250	$62.45	↑ 36x higher	⚙

Note:

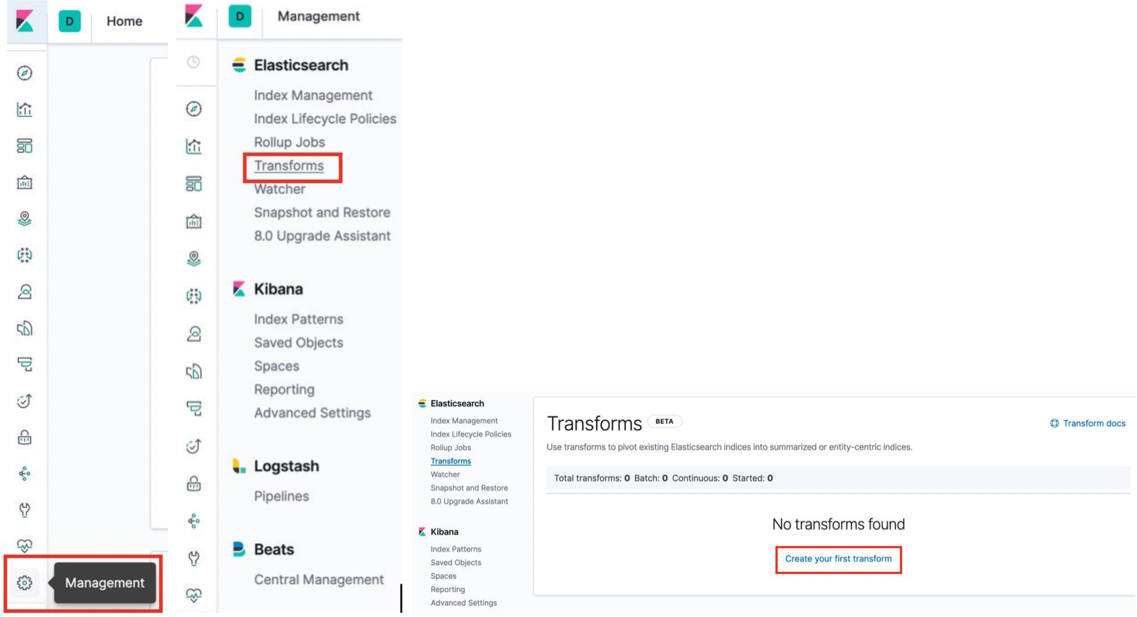
- Wagdi Shaw had a purchase of $2250, much more than the typical user purchase of $62.45
- (Optional) You could repeat the process of creating custom URL to raw data (as performed in Lab 1d) to see what Wagdi Shaw ordered

B - Data Frame Transformation

Use Data Frame analysis on Sample e-commerce logs to find users that spend more money than others

Note: This is similar to the Population Analysis, except that this is not time-series oriented. In the Population demo, the oddest user ("Wagdi Shaw") was deemed anomalous because the order placed at a particular time was much bigger than the typical user. But, what if Wagdi Shaw had spread out their spending over many orders over a long period of time. We may still desire to know who our biggest spenders are, regardless of what they do on a moment by moment basis. Therefore, we need to summarize their behavior over all time (or a longer span of time)

1. Transform the Data -
   We need to Transform the data from the time-series domain to an entity-centric index.
   Click on Kibana > Management > Transforms > Create your first transform

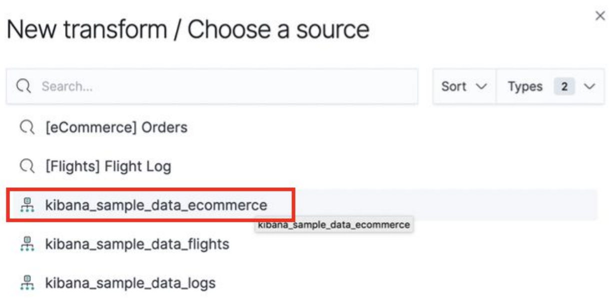


Use transforms to pivot existing Elasticsearch indices into summarized or entity-centric indices.

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

No transforms found

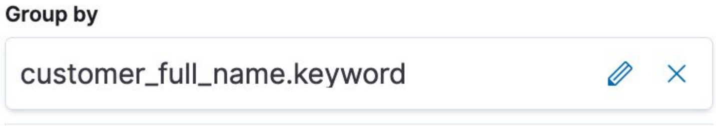
2. Choose “kibana_sample_data_ecommerce” as the source



New transform / Choose a source

- [eCommerce] Orders
- [Flights] Flight Log
- kibana_sample_data_ecommerce
- kibana_sample_data_flights
- kibana_sample_data_logs

3. Group By - Pivot on ("Group by") the "customer_full_name.keyword"

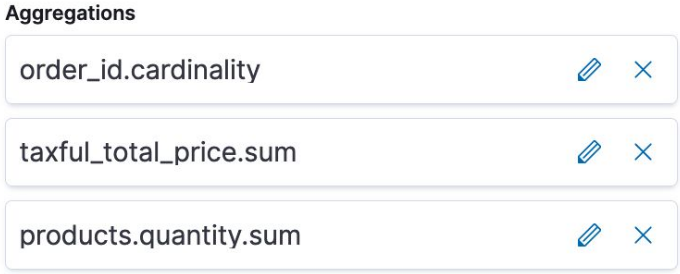


Group by

customer_full_name.keyword

4. Create the following column aggregations
   a. order_id.cardinality
   b. taxful_total_price.sum
   c. products.quantity.sum

   Note: This just means that we are aggregating for each customer name:
   a. The number of unique orders purchased to date
   b. Total amount they've ever spent on the e-commerce store, and
   c. The total number of products they've ever bought (by summing those fields over all time)

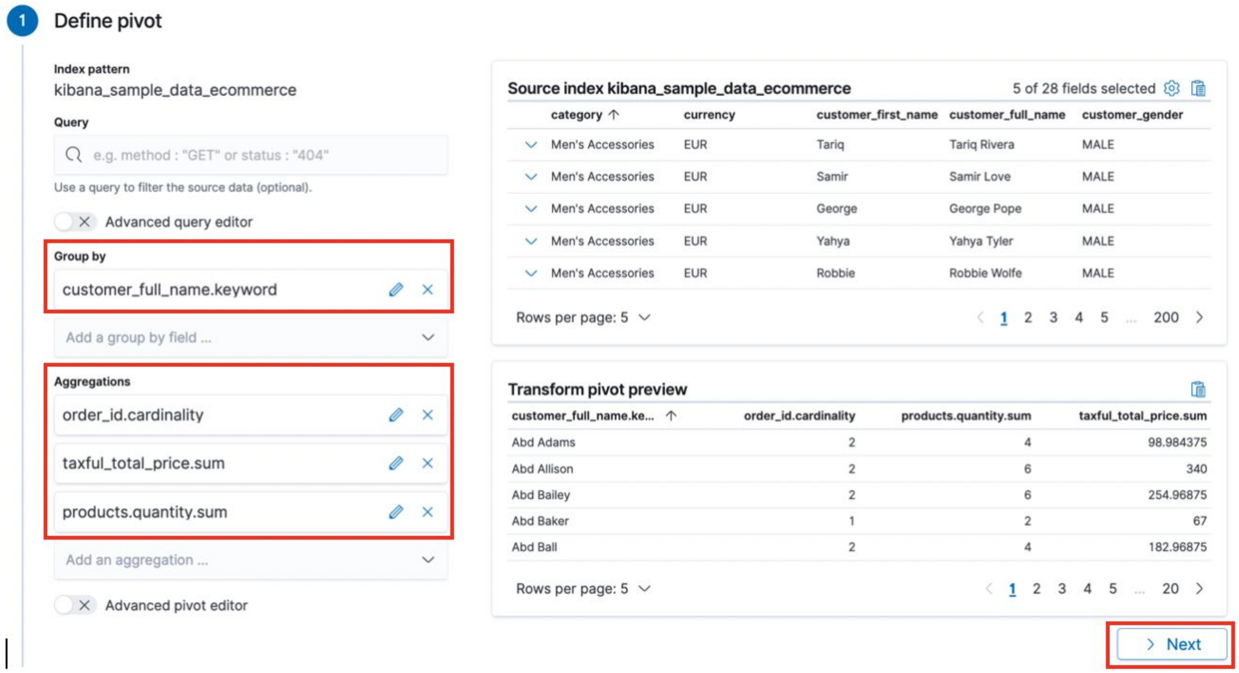


Aggregations

order_id.cardinality

taxful_total_price.sum

products.quantity.sum



1 Define pivot

Index pattern
kibana_sample_data_ecommerce

Query
e.g. method : "GET" or status : "404"
Use a query to filter the source data (optional).

Advanced query editor

Group by
customer_full_name.keyword

Add a group by field ...

Aggregations
order_id.cardinality
taxful_total_price.sum
products.quantity.sum

Add an aggregation ...

Advanced pivot editor

Source index kibana_sample_data_ecommerce — 5 of 28 fields selected

category ↑	currency	customer_first_name	customer_full_name	customer_gender
Men's Accessories	EUR	Tariq	Tariq Rivera	MALE
Men's Accessories	EUR	Samir	Samir Love	MALE
Men's Accessories	EUR	George	George Pope	MALE
Men's Accessories	EUR	Yahya	Yahya Tyler	MALE
Men's Accessories	EUR	Robbie	Robbie Wolfe	MALE

Rows per page: 5 — 1 2 3 4 5 ... 200

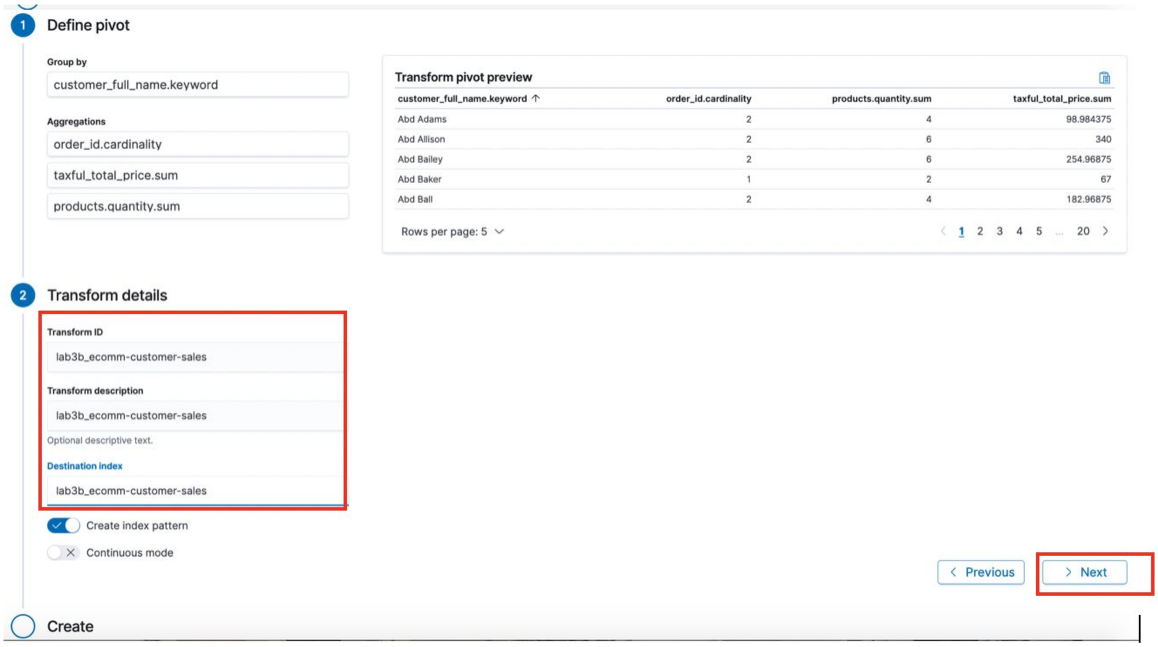
Transform pivot preview

customer_full_name.ke... ↑	order_id.cardinality	products.quantity.sum	taxful_total_price.sum
Abd Adams	2	4	98.984375
Abd Allison	2	6	340
Abd Bailey	2	6	254.96875
Abd Baker	1	2	67
Abd Ball	2	4	182.96875

Rows per page: 5 — 1 2 3 4 5 ... 20

Click on the “Next” button to continue

5. Name both the Transform ID and Destination Index as : “lab3b_ecomm_customer_sales” and click on “Next”



1 Define pivot

Group by
customer_full_name.keyword

Aggregations
order_id.cardinality
taxful_total_price.sum
products.quantity.sum

Transform pivot preview

customer_full_name.keyword ↑	order_id.cardinality	products.quantity.sum	taxful_total_price.sum
Abd Adams	2	4	98.984375
Abd Allison	2	6	340
Abd Bailey	2	6	254.96875
Abd Baker	1	2	67
Abd Ball	2	4	182.96875

Rows per page: 5 — 1 2 3 4 5 ... 20

2 Transform details

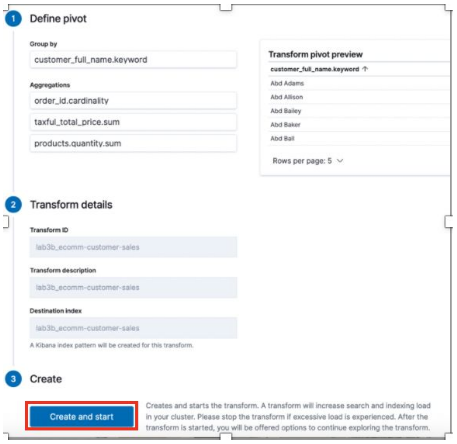
Transform ID
lab3b_ecomm-customer-sales

Transform description
lab3b_ecomm-customer-sales
Optional descriptive text.

Destination index
lab3b_ecomm-customer-sales

Create index pattern
Continuous mode

6. Go ahead and click on the “Create and Start” button



1 Define pivot

Group by: customer_full_name.keyword

Aggregations: order_id.cardinality, taxful_total_price.sum, products.quantity.sum

Transform pivot preview
customer_full_name.keyword ↑
Abd Adams
Abd Allison
Abd Bailey
Abd Baker
Abd Ball

Rows per page: 5

2 Transform details

Transform ID: lab3b_ecomm-customer-sales

Transform description: lab3b_ecomm-customer-sales

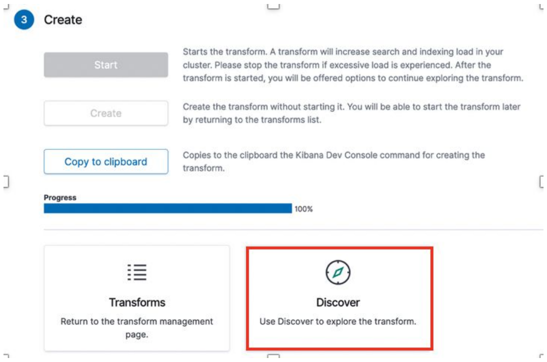
Destination index: lab3b_ecomm-customer-sales

A Kibana index pattern will be created for this transform.

3 Create

Create and start — Creates and starts the transform. A transform will increase search and indexing load in your cluster. Please stop the transform if excessive load is experienced. After the transform is started, you will be offered options to continue exploring the transform.

7. After the job has completed, click Discover to see the new entity centric (transformed and summarized) index!



3 Create

Start — Starts the transform. A transform will increase search and indexing load in your cluster. Please stop the transform if excessive load is experienced. After the transform is started, you will be offered options to continue exploring the transform.

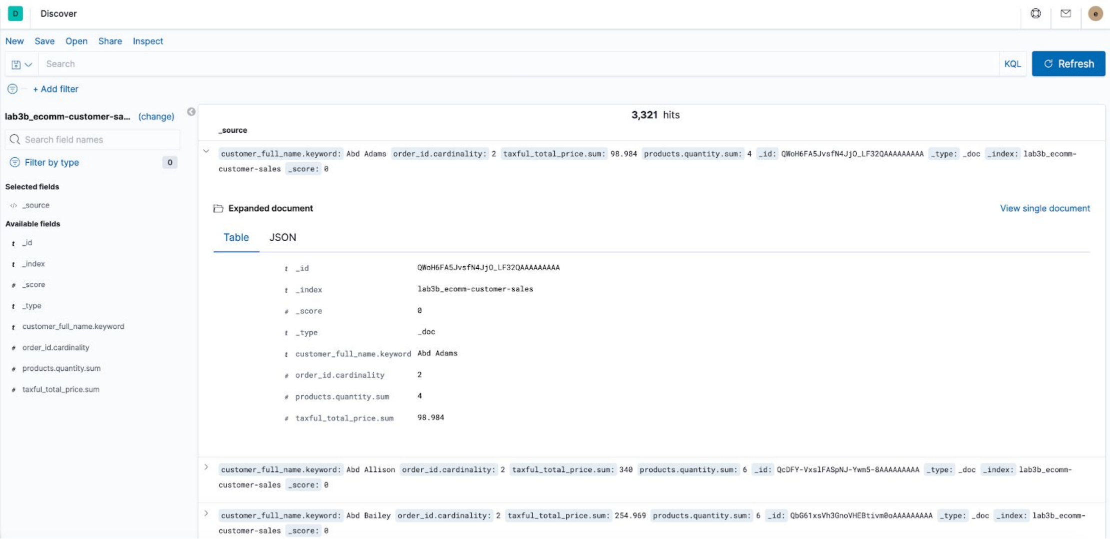
Create — Create the transform without starting it. You will be able to start the transform later by returning to the transforms list.

Copy to clipboard — Copies to the clipboard the Kibana Dev Console command for creating the transform.

Progress 100%

Transforms — Return to the transform management page.

Discover — Use Discover to explore the transform.



lab3b_ecomm-customer-sa... (change)

Search field names

Filter by type 0

Selected fields
- _source

Available fields
- _id
- _index
- _score
- _type
- customer_full_name.keyword
- order_id.cardinality
- products.quantity.sum
- taxful_total_price.sum

3,321 hits

_source

```
customer_full_name.keyword: Abd Adams order_id.cardinality: 2 taxful_total_price.sum: 98.984 products.quantity.sum: 4 _id: QWoH6FA5JvsfN4JjO_LF32QAAAAAAAAA _type: _doc _index: lab3b_ecomm-customer-sales _score: 0
```

Expanded document — View single document

Table | JSON

```
t _id                          QWoH6FA5JvsfN4JjO_LF32QAAAAAAAAA
t _index                       lab3b_ecomm-customer-sales
# _score                       0
t _type                        _doc
t customer_full_name.keyword   Abd Adams
# order_id.cardinality         2
# products.quantity.sum        4
# taxful_total_price.sum       98.984
```

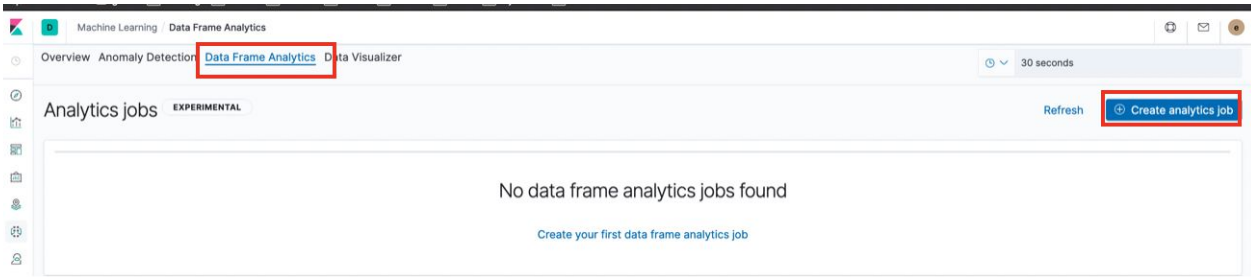
```
customer_full_name.keyword: Abd Allison order_id.cardinality: 2 taxful_total_price.sum: 340 products.quantity.sum: 6 _id: QcDFY-VxslFASpNJ-Ywm5-8AAAAAAAAA _type: _doc _index: lab3b_ecomm-customer-sales _score: 0

customer_full_name.keyword: Abd Bailey order_id.cardinality: 2 taxful_total_price.sum: 254.969 products.quantity.sum: 6 _id: QbG61xsVh3GnoVHEBtivm0oAAAAAAAAA _type: _doc _index: lab3b_ecomm-customer-sales _score: 0
```

This was the first step, to get the data in this form - we can now analyze it further! Let's find the unusual shoppers!

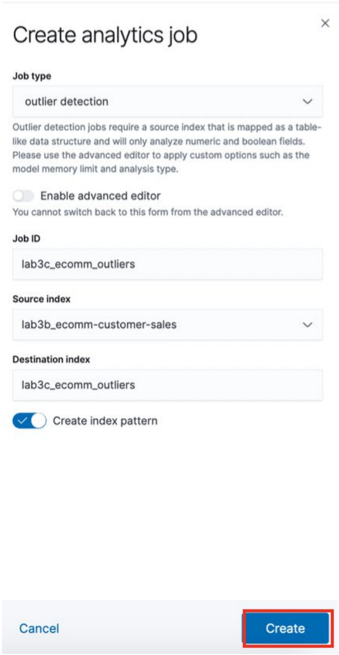
C - Outlier Detection

1. Click on Kibana > Machine Learning > Data Frame Analytics link. Then click on “Create analytics job”



No data frame analytics jobs found

2. Choose Job type as “outlier detection”.
   Enter the index from lab3b as the source index and name this job as well as destination index as **lab3c_ecomm_outliers.** Go ahead and click on the “Create” and “Start” buttons.



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 apply custom options such as the model memory limit and analysis type.

○ Enable advanced editor

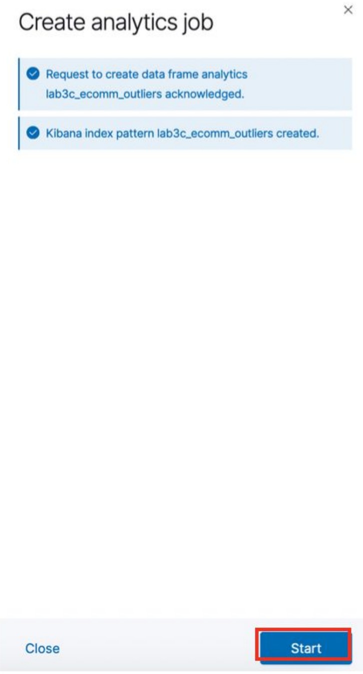
You cannot switch back to this form from the advanced editor.

Job ID: lab3c_ecomm_outliers

Source index: lab3b_ecomm-customer-sales

Destination index: lab3c_ecomm_outliers

☑ Create index pattern

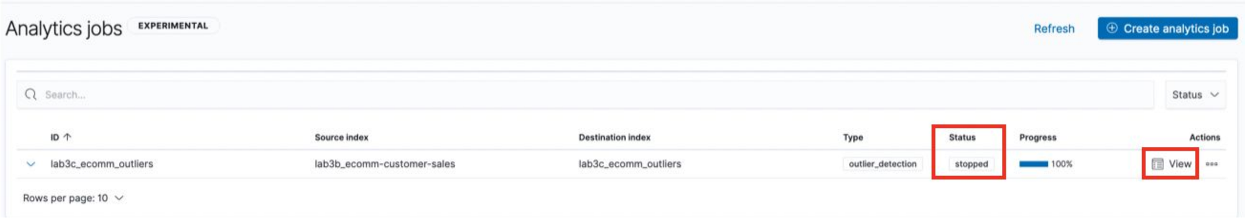


Create analytics job

✓ Request to create data frame analytics lab3c_ecomm_outliers acknowledged.

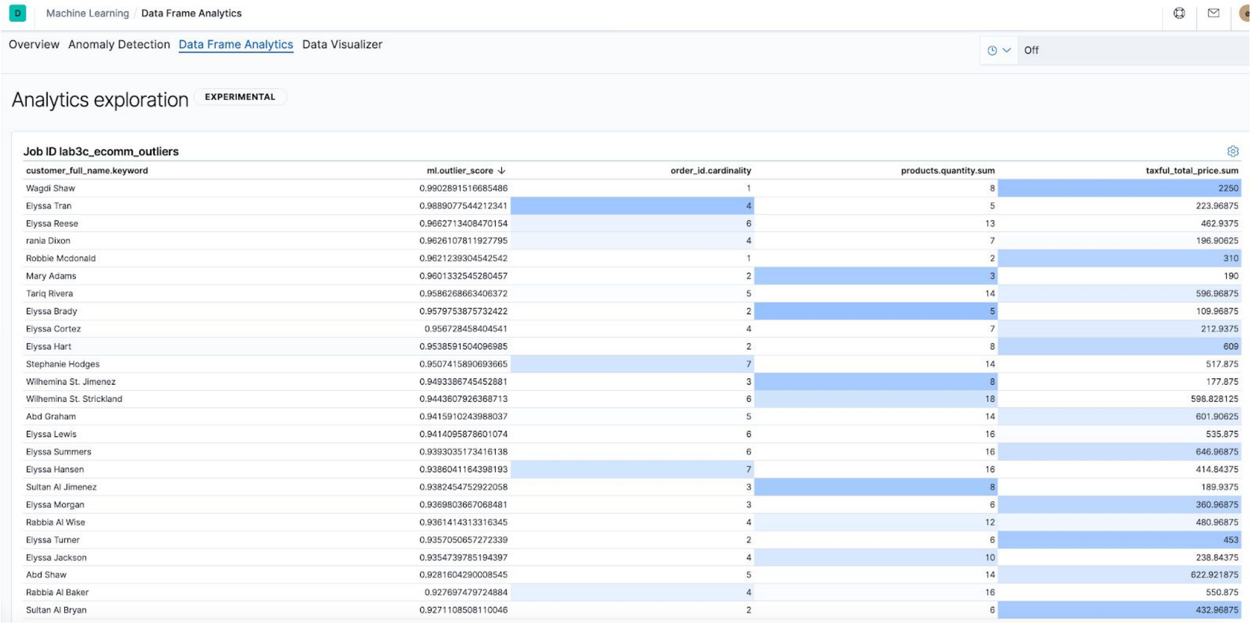
✓ Kibana index pattern lab3c_ecomm_outliers created.

3. When the status of the job is “stopped” (it takes a few seconds), click on the “View” button to view the results



ID ↑	Source index	Destination index	Type	Status	Progress	Actions
lab3c_ecomm_outliers	lab3b_ecomm-customer-sales	lab3c_ecomm_outliers	outlier_detection	stopped	100%	View

4. We are shown a data table with customers ranked by their outlier score. The data frame analytics job creates an index that contains the original data and outlier scores for each document. The outlier score indicates how different each entity is from other entities.
Note that:
    - The ml.outlier score is a value between 0 and 1. The larger the value, the more likely they are to be an outlier.
    - In addition to an overall outlier score, each document is annotated with feature influence values for each field. These values add up to 1 and indicate which fields are the most important in deciding whether an entity is an outlier or inlier. For example, the dark shading on the products.taxful_price.sum field for Wagdi Shaw indicates that the sum of the product prices was the most influential feature in determining that Wagdi is an outlier.



Job ID lab3c_ecomm_outliers

customer_full_name.keyword	ml.outlier_score ↓	order_id.cardinality	products.quantity.sum	taxful_total_price.sum
Wagdi Shaw	0.9902891516685486	1	8	2250
Elyssa Tran	0.9889077544212341	4	5	223.96875
Elyssa Reese	0.9662713408470154	6	13	462.9375
rania Dixon	0.9626107811927795	4	7	196.90625
Robbie Mcdonald	0.9621239304542542	1	2	310
Mary Adams	0.9601332545280457	2	3	190
Tariq Rivera	0.9586268663406372	5	14	596.96875
Elyssa Brady	0.9579753875732422	2	5	109.96875
Elyssa Cortez	0.956728458404541	4	7	212.9375
Elyssa Hart	0.9538591504096985	2	8	609
Stephanie Hodges	0.9507415890693665	7	14	517.875
Wilhemina St. Jimenez	0.9493386745452881	3	8	177.875
Wilhemina St. Strickland	0.9443607926368713	6	18	598.828125
Abd Graham	0.9415910243988037	5	14	601.90625
Elyssa Lewis	0.9414095878601074	6	16	535.875
Elyssa Summers	0.9393035173416138	6	16	646.96875
Elyssa Hansen	0.9386041164398193	7	16	414.84375
Sultan Al Jimenez	0.9382454752922058	3	8	189.9375
Elyssa Morgan	0.9369803667068481	3	6	360.96875
Rabbia Al Wise	0.9361414313316345	4	12	480.96875
Elyssa Turner	0.9357050657272339	2	6	453
Elyssa Jackson	0.9354739785194397	4	10	238.84375
Abd Shaw	0.9281604290008545	5	14	622.921875
Rabbia Al Baker	0.927697479724884	4	16	550.875
Sultan Al Bryan	0.9271108508110046	2	6	432.96875