

Assignment 2 Part 2 Report

Ilaria Cattaneo

27th April 2023

1 Exercise 2

As for the previous exercise, we first need to convert the csv file into a json file. To do this, we used exactly the same procedure as for Exercise 1.

1.1 Point 1 - Dashboard creation

To create a dashboard we need to open "Dashboard" under "Analytics" section; then we clicked "Create Dashboard" and "Create a Data View", in which we named the view and connected it to our restaurants_ex index.

Edit data view

[Manage settings and view field details](#)

Name

restaurants_part2

Index pattern

restaurants_ex

Enter an index pattern that matches one or more data sources. Use an asterisk (*) to match multiple characters. Spaces and the characters , / ? " ' < > | are not allowed.

Timestamp field

Date

Select a timestamp field for use with the global time filter.

✓ Your index pattern matches 1 source.

restaurants_ex

Index

Rows per page: 10

1.1.1 Question 1

In this part we want to show some metrics about how many restaurants were reviewed, how many cities were covered and the total number of votes collected.

To do this, we can divide the question in 3 parts.

1. To make a metric about how many restaurants were reviewed, we've done the following step "Create visualization" and we dragged and dropped the field "RestaurantName.keyword". Then, we chose "Legacy Metric", from the dropdown box "Visualisation Type" with the metric on the right panel set on "Unique count of RestaurantName.keyword". In this way, we found 7'456 unique restaurants reviewed.

2. In order to count how many cities were covered, we need to make a count of distinctunique cities. To do so, we clicked on "Create visualization" and dragged and dropped the field "City/-Country/Continent.keyword".

Also in this case, we chose "Legacy Metric" from the dropdown box "Visualisation Type" and with "Unique count of City" on the right panel.

In this way, we found 143 unique cities covered.

3. To check how many votes have been collected, we've created a metric by clicking on "Create visualization" and dragging and dropping the field "Votes".

Also here, we "Legacy Metric" from the dropdown box "Visualisation Type", so the right panel is set to "Sum of Votes".

Doing this, we found that 1'481'308 votes were collected.



1.1.2 Question 2

To provide a filter over restaurants rating, in edit mode of "Dashboard" we clicked on "Controls" → "Add control"; this opens a right panel named "Create control", where we chose the field "RatingText.keyword".

RatingText.keyword	
Exists	3,719
Average	2,143
Not rated	2,094
Good	1,088
Very Good	291
Excellent	184
Poor	

1.1.3 Question 3

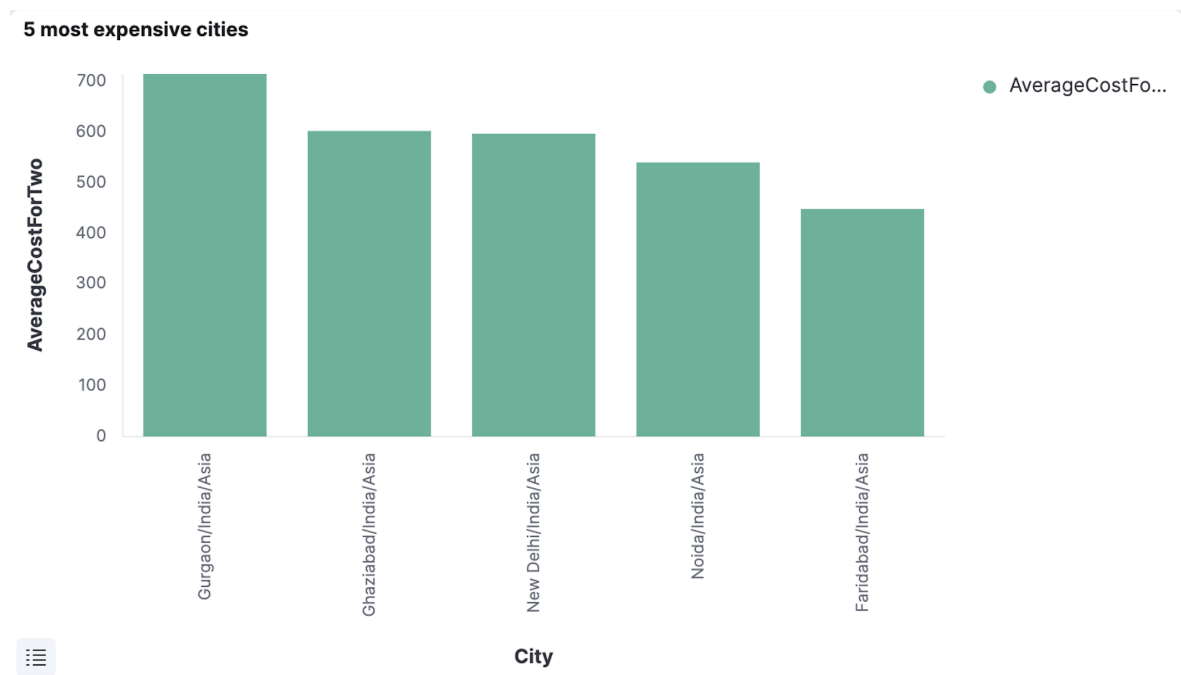
To show the top 5 cities with respect to the average cost of all restaurants in that city, we followed the following steps: "Select type" → "Aggregation based" → "Vertical bar".

In the "Metrics" section we set "Y-axis" with "Aggregation" as "Average" and "Field" as "AverageCostForTwo"; while, in the "Buckets" section we set "X-axis" with "Aggregation" as "Terms", "Field" as "City/Country/Continent.keyword", "Order by" as "Metric: AverageCostForTwo" and "Order" as "Descending" and "Size" equal to 5.

In the "Advanced" section, we wrote in the "JSON input" the chunk of code we wanted, in order to consider only the cities with a minimum number of restaurants equal to 25. The code is written below.

```
1 {
2   "min_doc_count": 25,
3   "shard_size": 1000
4 }
```

In addition to setting "min_doc_count" to 25, we needed to define the "shard_size" to 1000.



1.1.4 Question 4

In order to show the trend of restaurants reviews over time, we clicked on: "Select type" → "TSVB", because it's the best way to represent a time series.

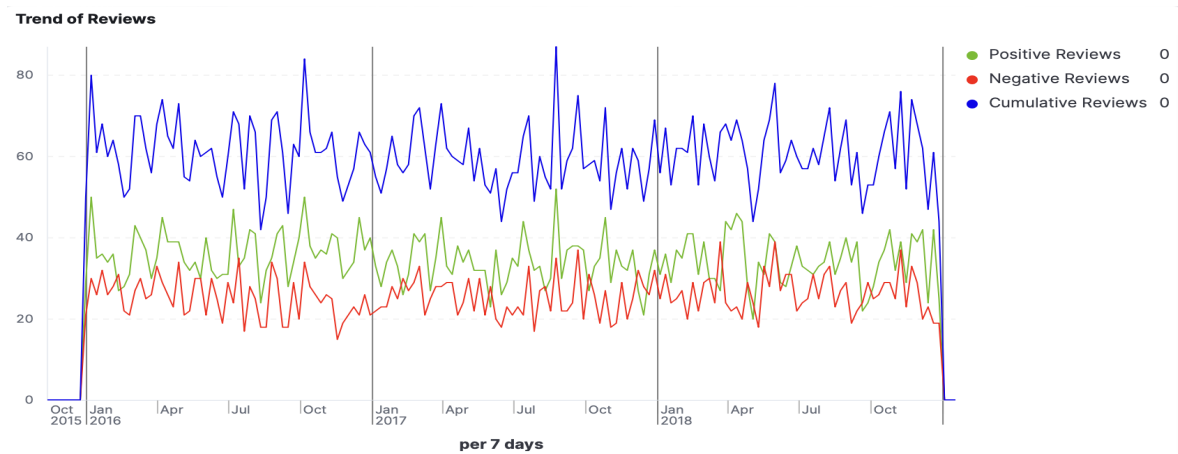
To make distinction between positive and negative reviews, we created 2 series.

The green line in the graph represents "Positive reviews", and to make it as displayed, in the right panel, in "Metrics" section we set "Aggregation" as "Count", "Group by" as "Filter" and "Query string" as "AggregateRating > 3".

The red line in the graph represents "Negative reviews", and to make it as displayed, in the right

panel, in "Metrics" section we set "Aggregation" as "Count", "Group by" as "Filter" and "Query string" as "AggregateRating <= 3".

In order to show a comparison between the previous and the total reviews, we created another time series, named "Cumulative Reviews" (represented in blue in the graph), by setting "Aggregation" as "Count" and "Group by" as "Filter".



1.1.5 Question 5

In this question, we created a map to show the location of all restaurants. To do so, we clicked "Select type" → "Maps".

Once created the map, we needed to click on "Add layer" → "Clusters".



1.1.6 Question 6

To show a table view with details about all restaurants, we created a saved search by switching to "Discover" and we selected the following fields: "RestaurantName", "City/Country/Continent", "AverageCostForTwo", "AggregateRating", "RatingText", "Votes" and "Date". We saved our saved search and then added it to our dashboard by clicking on "Add from library".

Table Search 9499 documents

Columns 1 field sorted

	Restaurant...	City/Count...	AverageCo...	Aggreg...	RatingText	Votes	Date
<input type="checkbox"/>	Spiral - Sofitel Philippine Plaza Manila	Pasay City/Philippines/Asia	6,000	4.9	Excellent	621	May 20, 2018 @ 01:10:50.0
<input type="checkbox"/>	Silantro Fil-Mex	Pasig City/Philippines/Asia	800	4.9	Excellent	1,070	Jan 30, 2018 @ 22:08:38.0
<input type="checkbox"/>	Coco Bambu	Brasilia/Brazil/South America	230	4.9	Excellent	30	Apr 27, 2016 @ 13:30:32.0
<input type="checkbox"/>	Braseiro da Gíçvea	Rio de Janeiro/Braz	100	4.9	Excellent	40	May 24, 2018 @ 18:34:59.0

Rows per page: 100

< 1 2 3 4 5 >

1.1.7 Question 7

In order to show a heat-map that gives information about the average of the number of votes for each continent and each "RatingText" type, we first needed to go to "Stack Management" → "restaurants_part2" (our data view) → "Scripted fields".

We created a scripted field called "Rating" of "Type" "String" and in the "Script" part we wrote:

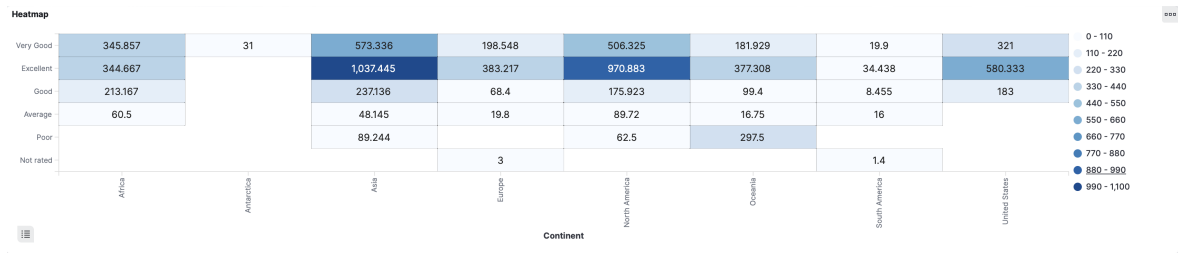
```
1 doc['RatingText.keyword'].value
```

We created a scripted field called "Continent" of "Type" "String" and in the "Script" part we wrote:

```
1 doc['City/Country/Continent.keyword'].value.splitOnToken('/')[2]
```

After doing this, we went back to "Dashboard" and we clicked on "Select type" → "Aggregation based" → "Heat map".

In the "Metrics" section we set "Aggregation" as "Average" and "Field" as "Votes", while in the "Buckets" section we set "X-axis" with "Aggregation" as "Terms", "Field" as "Continent", "Order by" as "Alphabetical", "Order" as "Ascending" and "Size" equal to 8, and then we also set "Y-axis" with "Sub aggregation" as "Terms", "Field" as "Rating", "Order by" as "Metric: Average Votes", "Order" as "Descending" and "Size" equal to 5.



1.2 Point 2 - Canvas creation

To create a canvas we clicked "Canvas" → "Create workpad".

1.2.1 Question 1

In order to create a filter we clicked on "Add element" → "Filter" → "Dropdown select".

In the right panel, in the "Display" section → "Dropdown filter" we set "Value column" as "City/Country/Continent" and "Filter column" as "City/Country/Continent". In the "Data" section, we selected "Elasticsearch documents", "Data view" as "restaurants_part2" (our data view) and "Fields" as "City/Country/Continent.keyword"

-- ANY --

1.2.2 Question 2

To show the average cost of the restaurant for 2 people based on different rating categories, we clicked on "Add element" → "Chart" → "Data table".

In the "Data" section, we selected "Elasticsearch SQL" and wrote the following query to retrieve data.

```
01 | SELECT RatingText.keyword, avg(AverageCostForTwo)
02 | FROM restaurants_ex
03 | GROUP BY RatingText.keyword
```

RatingText.keyword	avg_AverageCostForTwo_#
Average	569.5200322667383
Excellent	8400.41237113402
Good	1045.349570200573
Not rated	339.98833411105926
Poor	642.0923913043479
Very Good	3587.4016853932585

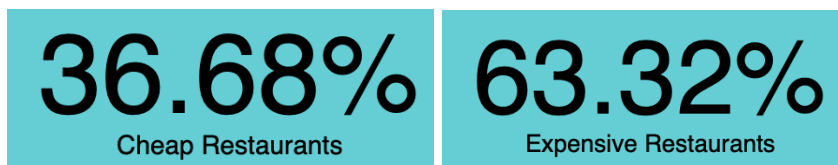
1.2.3 Question 3

To show the percentages of cheap and expensive restaurants we clicked on "Add element" → "Chart" → "Metric".

To do this we used the expression editor as follows, so that we could consider cheap (price ≤ 300) and expensive restaurants (price > 300).

(Note: the 2 chunks of code represent 2 distinct metrics in the canvas)

```
01 | kibana
02 | | selectFilter
03 | | esssql
04 | | query="SELECT count(RestaurantName) FROM restaurants_ex WHERE
    | |     AverageCostForTwo <= 300"
05 | | math "count\_RestaurantName\_/_9499"
06 | | metric "Cheap Restaurants"
07 | | metricFont={font align="center" color="#000000" family="'Open Sans',
    | |     Helvetica, Arial, sans-serif" italic=false size=60 underline=false weight
    | |     ="normal"}
08 | | labelFont={font size=14 family="'Open Sans', Helvetica, Arial, sans-serif"
    | |     color="#000000" align="center"} metricFormat="0,0.[00]%"
09 | | render
10 |
11 |
12 |
13 | kibana
14 | | selectFilter
15 | | esssql
16 | | query="SELECT count(RestaurantName) FROM restaurants_ex WHERE
    | |     AverageCostForTwo > 300"
17 | | math "count\_RestaurantName\_/_9499"
18 | | metric "Expensive Restaurants"
19 | | metricFont={font align="center" color="#000000" family="'Open Sans',
    | |     Helvetica, Arial, sans-serif" italic=false size=60 underline=false weight
    | |     ="normal"}
20 | | labelFont={font size=14 family="'Open Sans', Helvetica, Arial, sans-serif"
    | |     color="#000000" align="center"} metricFormat="0,0.[00]%"
21 | | render
```



1.2.4 Question 4

In order to show a table that displays information about the restaurants with 5 rows per page, we clicked on "Add element" → "Chart" → "Data table".

To do this we used the expression editor as follows.

```
01 | kibana
02 | | selectFilter
03 | | esssql
```

```

04 | query="SELECT RestaurantName.keyword, AverageCostForTwo, RatingText.keyword
05 | FROM restaurants_ex"
06 | | table perPage=5
07 | | render

```

RestaurantName.keyword	AverageCostForTwo	RatingText.keyword
Le Petit Souffle	1100	Excellent
Buffet 101	2000	Very Good
Vikings	2000	Very Good
Spiral - Sofitel Philippine Plaza Manila	6000	Excellent
Locavore	1100	Excellent

1.2.5 Question 5

To show some metrics about the restaurants: the number of votes, the average cost, and the average rating, we used the expression editor, as follows.

(Note: the 3 chunks of code represent 3 distinct metrics in the canvas)

```

01 | kibana
02 | | selectFilter
03 | | essql query="SELECT count(Votes) FROM restaurants_ex"
04 | | math "count_Votes_"
05 | | metric "Number of Votes"
06 | | metricFont={font size=48 family="'Open Sans', Helvetica, Arial, sans-serif"
07 | | labelFont={font size=14 family="'Open Sans', Helvetica, Arial, sans-serif"
08 | | render
09 |
10 |
11 |
12 | kibana
13 | | selectFilter
14 | | essql query="SELECT avg(AggregateRating) FROM restaurants_ex"
15 | | math "avg_AggregateRating_"
16 | | metric "Average Rating"
17 | | metricFont={font size=48 family="'Open Sans', Helvetica, Arial, sans-serif"
18 | | labelFont={font size=14 family="'Open Sans', Helvetica, Arial, sans-serif"
19 | | render
20 |
21 |
22 |
23 | kibana
24 | | selectFilter
25 | | essql query="SELECT avg(AverageCostForTwo) FROM restaurants_ex"
26 | | math "avg_AverageCostForTwo_"
27 | | metric "Average Cost"
28 | | metricFont={font size=48 family="'Open Sans', Helvetica, Arial, sans-serif"

```



```
29 |     labelFont={font size=14 family="'Open Sans', Helvetica, Arial, sans-serif"
    |         color="#000000" align="center"} metricFormat="0,0.[00]"
30 | | render
```

