DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

in collaborazione con

Istituto di Informatica e Telematica

UNIVERSITÀ DI PISA

Master Universitario di I Livello in Cybersecurity

Corso "Cyber Intelligence": esercitazione su ES e Kibana

Tiziano Fagni
IIT,CNR
tiziano.fagni@iit.cnr.it

Anno accademico 2022/2023

Esercizio 1

Obiettivo

Sfruttando il template presente sul file “FilmsMapping.json” in “cint/esercitazione/elastic/mapping_films”, creare un nuovo mapping per ES che consideri anche i campi di arricchimento definiti negli Esercizi 4 e 5 su NiFi. I campi nuovi fanno riferimento a questi dati:

```
"sentiment_polarity": "-1",
"webpage":{
    "url":"https://t.co/6wixF7WHyq",
    "title":"PesNew Era su Twitter...",
    "content":"Ora sei..."
}
```

Sfruttare https://www.elastic.co/guide/en/elasticsearch/reference/current/mapping-types.html

Esercizi su query

Trovate la documentazione di Elasticsearch sulle query qui: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html

Tipi di query che vedremo:

- Full text queries
- Term level queries

Faremo le nostre query sull'indice "films" creato dal dataflow "FilmDataForQueries".

Full text queries: “match” query

```
GET /<index-name>/_search
{
  "from" : 0,
  "size" : 100,
  "query": {
        "match" : {
            "message" : {
                "query" : "this is a test",
                "operator" : "and"
            }
        }
    }
}
```

Cerca le parole specificate dalla keyword “query” nel campo padre. Posso dire se cercarle in “and” oppure “or” tramite la keyword “operator”.

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query.html

Esercizio

Sfruttando la “match” query, cercare tutti i film che verificano le query:

1. frase “rapina banca” con parole in “and” sul campo contenente la descrizione del film.
2. frase “terrorismo banca” con parole in “or” sul campo contenente la descrizione del film.
3. frase “bova morante” con parole in “or” sul campo che fa riferimento agli attori di un film.

Full text queries: “match phrase” query

```
GET /_search
{
    "query": {
        "match_phrase" : {
            "message" : {
                "query": "this is a test",
                "slop": 0
            }
        }
    }
}
```

Cerca il matching per la frase specificata nel campo indicato. Con il parametro “slop” si può indicare quanto preciso può essere fatto il matching considerando l’ordine delle parole.

Il parametro “slop” cerca di rispondere a questa domanda

“By how far apart we mean how many times do you need to move a term in order to make the query and document match?”

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query-phrase.html

https://www.elastic.co/guide/en/elasticsearch/guide/current/slop.html

Esercizio

Sfruttando la “match phrase” query e usando un campo a vostra scelta, cercare tutti i film che verificano la query:

“coppia figlio” con vari valori di “slop”

Full text queries: “match phrase prefix” query

```
GET /_search
{
    "query": {
        "match_phrase_prefix" : {
            "message": {
                "query" : "this is a t",
                "slop": 0,
                "max_expansions" : 50
            }
        }
    }
}
```

Cerca il matching nel campo indicato per la frase specificata facendo l’espansione dell’ultimo termine.

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-match-query-phrase-prefix.html

Esercizio

Sfruttando la “match phrase prefix” query e usando un campo a vostra scelta, cercare tutti i film che nella descrizione verificano la query:

“banca rap” con vari valori di “slop”.

E’ possibile utilizzare sia la scheda “Dev tools” sia la scheda “Discover”

Full text queries: Multi Match query

```
GET /_search
{
  "query": {
    "multi_match" : {
      "query":      "brown fox",
      "type":       "best_fields",
      "fields":     [ "subject^3", "message" ],
    }
  }
}
```

Stessa semantica della “match” query ma con la possibilità di utilizzare più campi di ricerca e combinare gli score.

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-multi-match-query.html

Esercizio

Sfruttando la “multi match” query, provate a cercare qualcosa sui campi “titolo_italiano” e “descrizione”

- assegnando 3 volte più importanza a quello che trovate in “titolo_italiano”.
- provando a combinare lo score finale sfruttando le diverse policy disponibili (vedi https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-multi-match-query.html).

Full text queries: “query string” query

Si usa il **“query string format”** per formulare le query

```
GET /_search
{
    "query": {
        "query_string" : {
            "default_field" : "content",
            "query" : "(this AND that) OR thus",
            "default_operator" : "OR"
        }
    }
}
```

Molti altri parametri disponibili, vedere
https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html

Query string format

- where the `status` field contains `active`

```
status:active
```

- where the `title` field contains `quick` or `brown`. If you omit the OR operator the default operator will be used

```
title:(quick OR brown)
title:(quick brown)
```

- where the `author` field contains the exact phrase `"john smith"`

```
author:"John Smith"
```

- where any of the fields `book.title`, `book.content` or `book.date` contains `quick` or `brown` (note how we need to escape the `*` with a backslash):

```
book.\*:(quick brown)
```

- where the field `title` has any non-null value:

```
_exists_:title
```

Query string format (2)

- All days in 2012:

```
date:[2012-01-01 TO 2012-12-31]
```

- Numbers 1..5

```
count:[1 TO 5]
```

- Tags between `alpha` and `omega`, excluding `alpha` and `omega`:

```
tag:{alpha TO omega}
```

- Numbers from 10 upwards

```
count:[10 TO *]
```

- Dates before 2012

```
date:{* TO 2012-01-01}
```

Query string format (3)

Ranges with one side unbounded can use the following syntax:

```
age:>10
age:>=10
age:<10
age:<=10
```

NOTE

To combine an upper and lower bound with the simplified syntax, you would need to join two clauses with an `AND` operator:

```
age:(>=10 AND <20)
age:(+>=10 +<20)
```

```
quick brown +fox -news
```

states that:

- `fox` must be present
- `news` must not be present
- `quick` and `brown` are optional — their presence increases the relevance

Query string format (4)

Multiple terms or clauses can be grouped together with parentheses, to form sub-queries:

```
(quick OR brown) AND fox
```

Groups can be used to target a particular field, or to boost the result of a sub-query:

```
status:(active OR pending) title:(full text search)^2
```

E’ possibile fare molte altra cose come ricerche fuzzy, di prossimità, con boosting, ecc.!

La documentazione completa è disponibile qui: https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-query-string-query.html#query-string-syntax

Esercizio

Dal tab “Discover” di Kibana provare a recuperare i film che matchano le condizioni:

1. Voto medio >= 7 e “gangster” dentro la descrizione.
2. (Genere è “Drammatico” e titolo include “paura”) oppure (genere è “Azione” e contenuto include “combattimento”)
3. Periodo dal 1/1/1990 al 31/12/2010 che hanno la parola esatta “New York” all’interno di qualsiasi campo testuale.

Term queries

The `term` query finds documents that contain the **exact** term specified in the inverted index. For instance:

```
POST _search
{
  "query": {
    "term" : { "user" : "Kimchy" } ❶
  }
}
```

Filters documents that have fields that match any of the provided terms (**not analyzed**). For example:

```
GET /_search
{
    "query": {
        "terms" : { "user" : ["kimchy", "elasticsearch"]}
    }
}
```

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-term-query.html

Term queries (2)

```
GET _search
{
    "query": {
        "range" : {
            "age" : {
                "gte" : 10,
                "lte" : 20,
                "boost" : 2.0
            }
        }
    }
}
```

```
GET _search
{
    "query": {
        "range" : {
            "date" : {
                "gte" : "now-1d/d",
                "lt" :  "now/d"
            }
        }
    }
}
```

https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl-range-query.html

Altri tipi di term queries utili

- exists query: verifica la presenza di un campo in un documento
- fuzzy query: cerca documenti con termini simili a quello speicificato in query sfruttando similarità basata su Levenshtein edit distance
- ids query: ritorna tutti i documenti che matchano gli ID specificati
- regexp query: ritorna tutti i documenti che matchano una regexp

Per tutto il resto...

- **Query DSL**
  - Query and filter context
  - Match All Query
  - + Full text queries
  - + Term level queries
  - + Compound queries
  - + Joining queries
  - + Geo queries
  - + Specialized queries
  - + Span queries
  - Minimum Should Match
  - Multi Term Query Rewrite

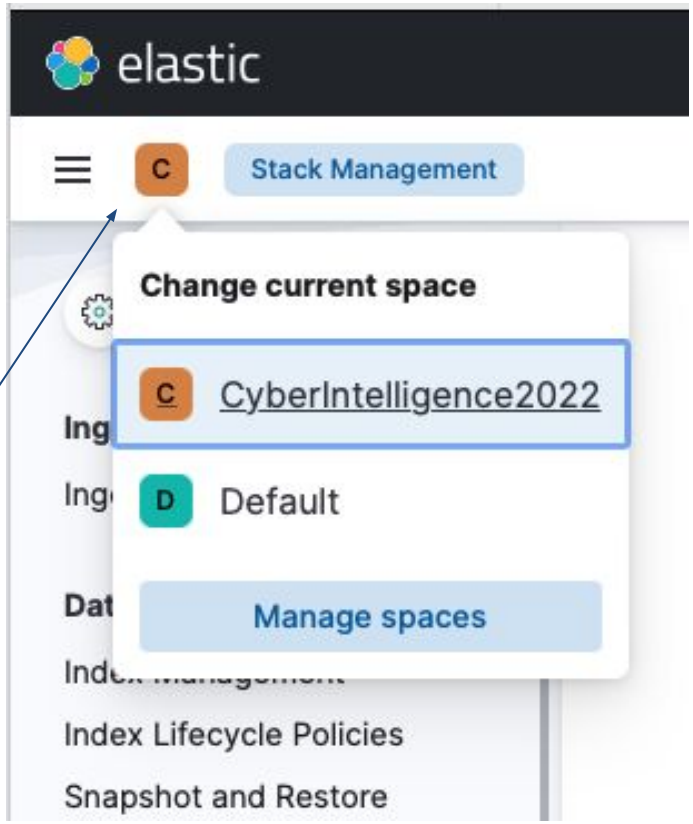
C’è la documentazione!

Guardate a partire da https://www.elastic.co/guide/en/elasticsearch/reference/current/query-dsl.html

Creazione e gestione Space su Kibana

Gli Spaces permettono di organizzare i contenuti su Kibana creando dei contenitori logici sotto cui raggruppare “Index Patterns”, viste tabellari, visualizzazioni e dashboard.

Per creare e gestire uno Space andare qui



Gli indici sono sempre condivisi tra gli Space ma gli Index Patterns no!

Query salvate

Dal tab “Discover” è possibile creare viste tabellari specifiche che visualizzano i dati con eventuali filtri applicati.

- **View1**: Creare una vista che riporta “titolo_originale”, “paese”, “durata” e “voto_medio”
- **View2**: Creare una vista che riporta “titolo_italiano”, “voti_totali”, “registi” e “attori” sui dati prefiltrati che contengono solo film con votazione media > 7.

Visualizzazione 1a e 1b

Componente “Tag Cloud”

Visualizzare i 25 registi più attivi e le 25 parole più usate nei titoli italiani dei film di genere “Thriller”

a)

b)

Visualizzazione 1c

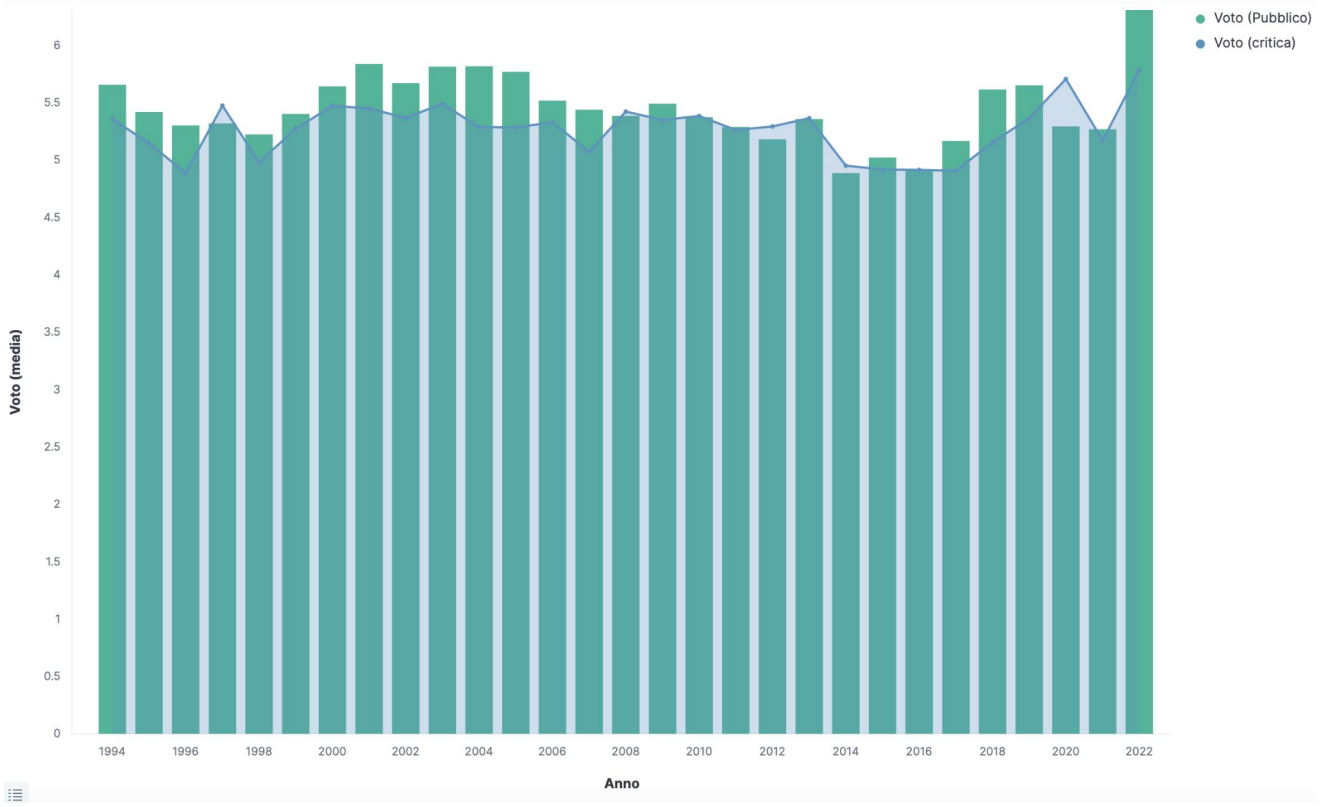
Visualizzare la tipologia di film ordinati per lunghezza media.

- Si aggregano i dati per average di “durata”
- Si creano buckets per tutti i valori del campo “genere”

Visualizzazione 2

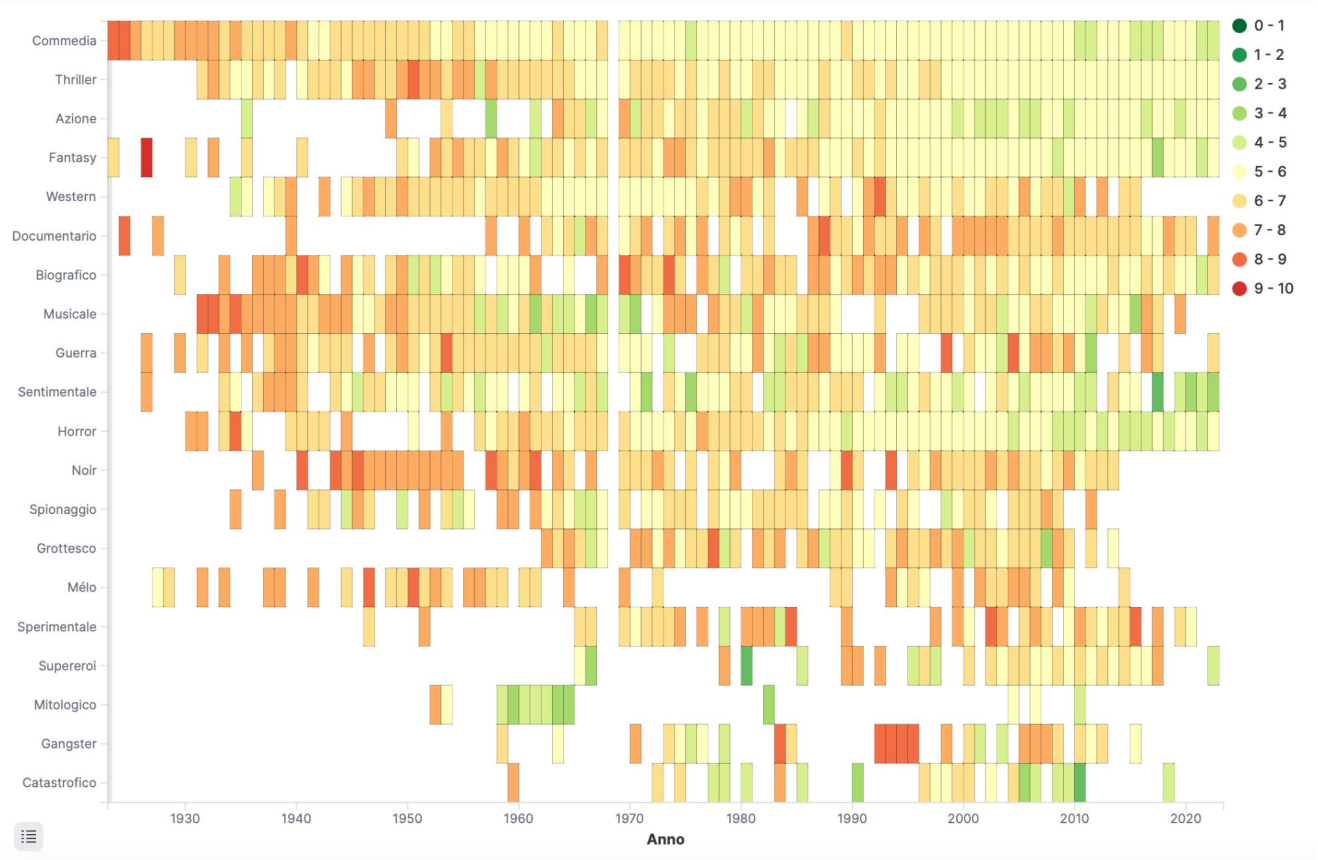
Componente “Vertical Bar”

Visualizzare la media del voto del pubblico e la media del voto della critica di tutti i film italiani considerando gli ultimi 30 anni



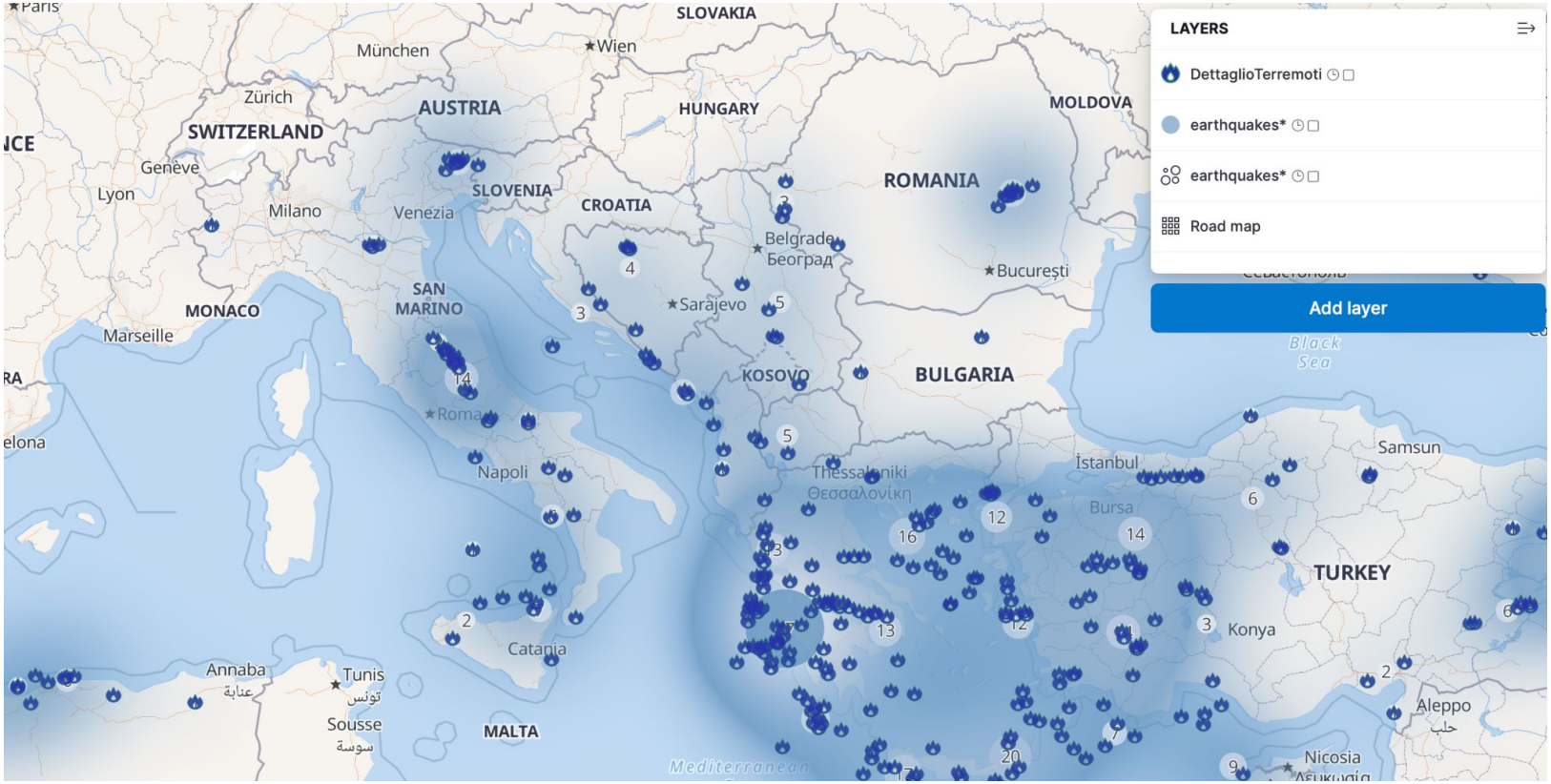
Visualizzazione 3

Componente “Heat Map”
Realizzare una heatmap che visualizzi nel corso del tempo il voto medio rispetto al genere di film.



Visualizzazione 4

Componente “Map”
Utilizzando l’indice “earthquakes” creato con il flusso “EarthquakesESIngester”, visualizzare i terremoti occorsi nel corso degli anni sfruttando i layer per “Clusters and grid”, “Heatmaps” e “Documents”



Visualizzazione 5

Componente “Goal”

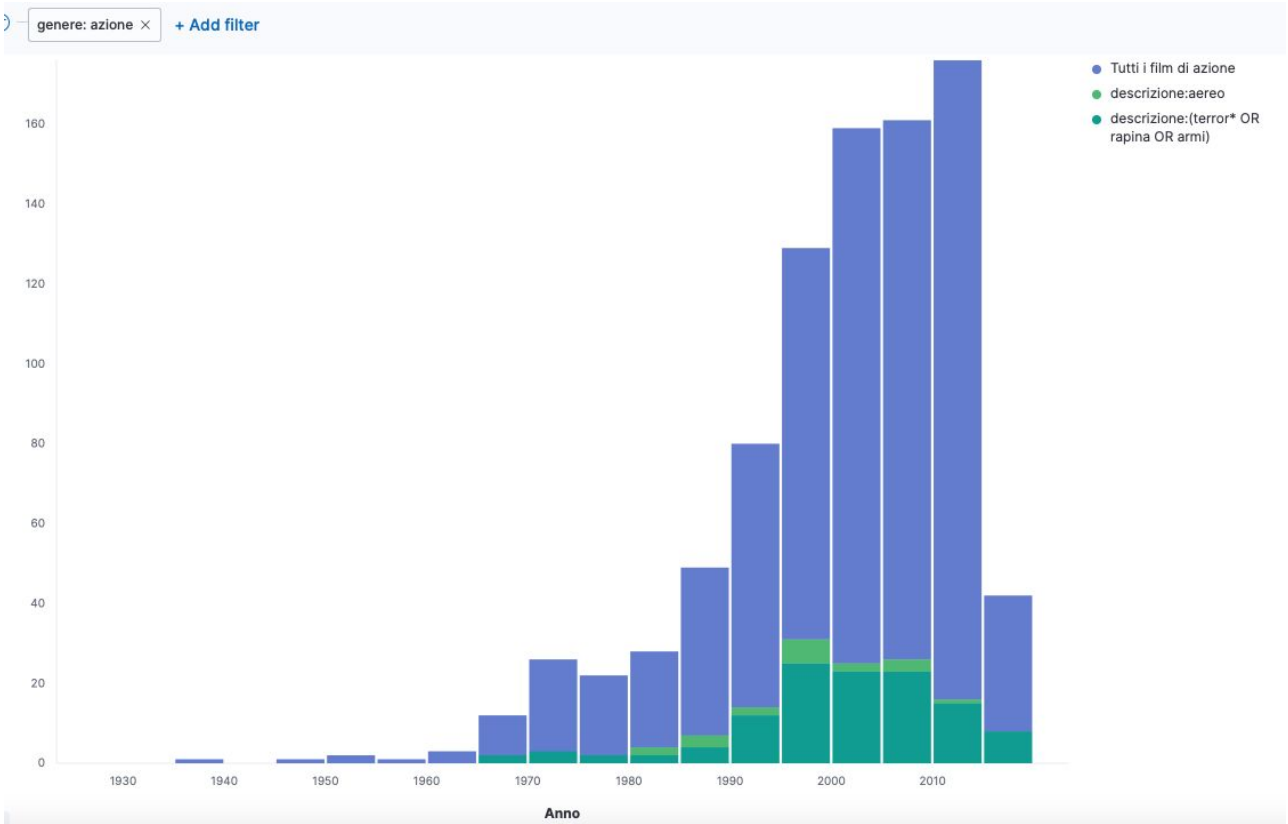
Film di tipo “drammatico” suddivisi per voto medio

Film drammatici suddivisi per voto medio

Visualizzazione 6

Componente “Vertical Bar”

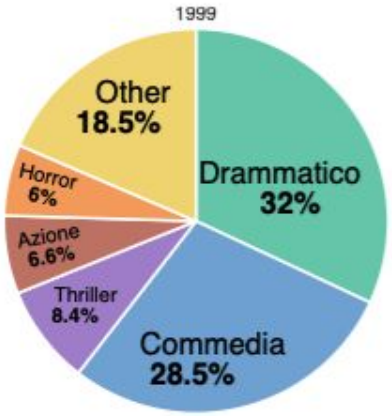
Visualizzare i film di azione nel corso degli anni (gruppi di 5 anni) suddividendoli tra quelli che hanno nella descrizione (“terror* OR rapina OR armi), quelli con in descrizione “aereo” e tutti gli altri rimanenti film di azione.

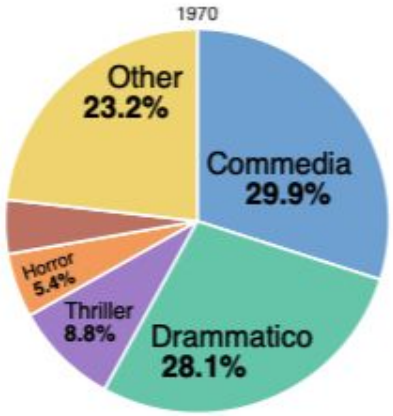


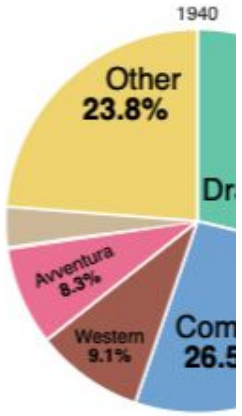
Visualizzazione 7

Componente “Pie”

Visualizzare nel corso degli anni (a gruppi di 30 anni) le 5 categorie di film più popolari. Nei Pie chart aggiungere anche il valore “Others” per tutti i film che non rientrano nelle prime 5 categorie.



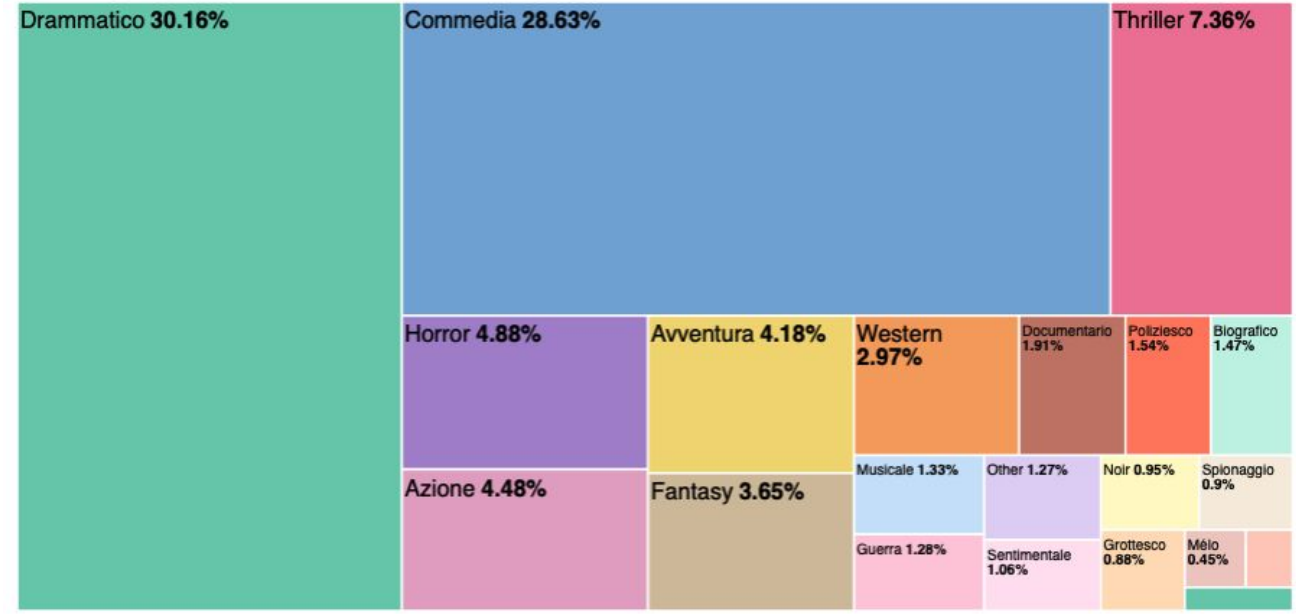




Visualizzazione 7b

Componente “Pie”

Usare la modalità “Lens” per visualizzare la distribuzione dei film all’interno dei vari generi di pellicole. Usare la visualizzazione Treemap a tale scopo.



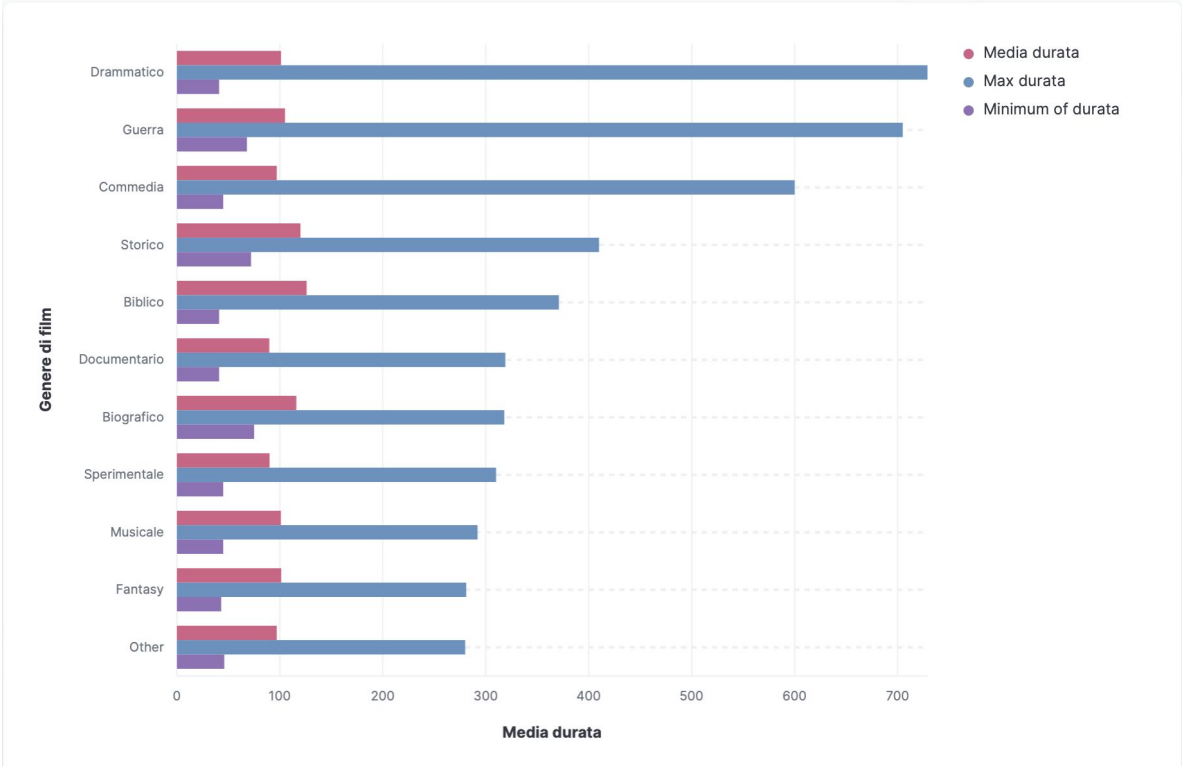
Visualizzazione 8

Usare la modalità “Lens” per creare una tabella riassuntiva che le colonne “Genere film”, “Durata media (min)”, “Max voto medio” e “Min voto medio”.

Genere film	Numero film	Durata media (min) ↓	Max voto medio	Min voto medio
Documentario	325	90	10	2.5
Drammatico	5,134	101.255	9.6	1.5
Commedia	4,873	97	9.4	1.6
Avventura	711	99	9.3	2.6
Guerra	218	105	9.3	3
Animazione	31	90	9.3	2.9
Fantasy	621	101.5	9.2	2
Thriller	1,252	99	9.1	1.8
Horror	830	92	9.1	1.5
Noir	161	97	9.1	3.4
Grottesco	149	98	9.1	2.3
Sperimentale	55	90	9	3.7
Gangster	41	106	9	4
Western	506	95	8.8	2.2

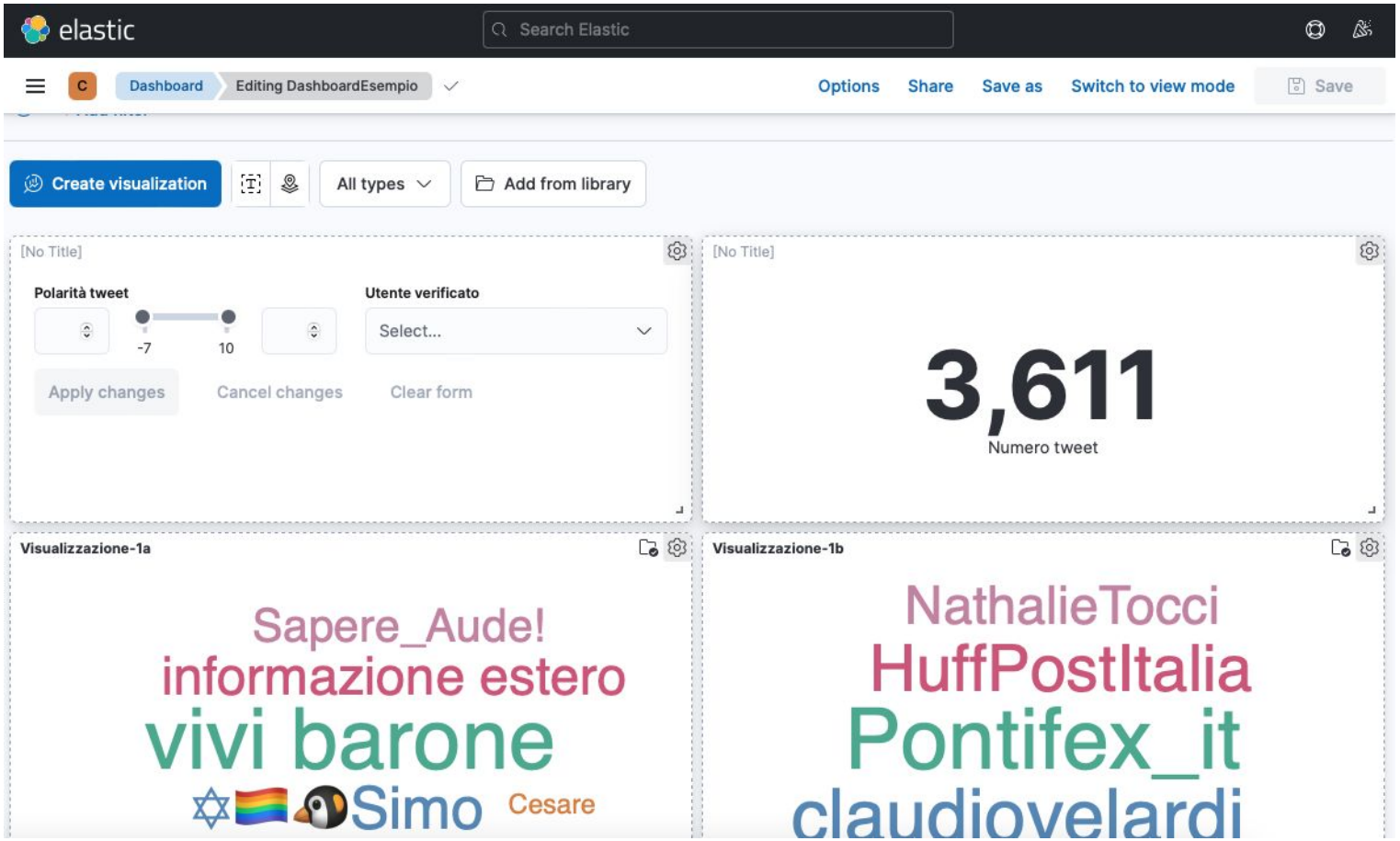
Visualizzazione 9

Sfruttare Lens per visualizzare nello stesso grafico la durata massima, media e minima dei film suddivisi per genere di film (i 10 generi con la lunghezza massima più elevata).



Adesso costruiamo una bella dashboard...

Potete usare *ricerche* e *visualizzazioni* salvate oppure crearle al volo, oltre che filtri dinamici



Ok, questo è tutto che che vi serve per fare il vostro bel progettone finale!

Se avete dubbi o domande chiedete pure! :-)