# practica-final-LuisAngulo

February 20, 2024

## 1 Práctica final

En esta prática vamos responder una serie de preguntas utilizando busqueda de elastic search sobre un set de datos.

Para entregar la practica, una vez resueltos los ejercicións se debera exportar el notebook a PDF y subirse a la plataforma.

### 1.1 Introducción

Este primer bloque de código sirve para configurar el Notebook

[1]:
```
from IPython.display import JSON
```

Ahora vamos a descargar el cliente de ElasticSearch en Python.

[2]:
```
pip install elasticsearch==7.10.1
```

```
Collecting elasticsearch==7.10.1
  Downloading elasticsearch-7.10.1-py2.py3-none-any.whl (322 kB)
                             322.1/322.1
kB 3.3 MB/s eta 0:00:0000:01
Requirement already satisfied: urllib3<2,>=1.21.1 in
/opt/conda/lib/python3.10/site-packages (from elasticsearch==7.10.1) (1.26.11)
Requirement already satisfied: certifi in /opt/conda/lib/python3.10/site-
packages (from elasticsearch==7.10.1) (2022.9.24)
Installing collected packages: elasticsearch
Successfully installed elasticsearch-7.10.1
Note: you may need to restart the kernel to use updated packages.
```

Por último, creamos la conexion con el servidor de Elastic Search desplegado

[3]:
```
from elasticsearch import Elasticsearch
es = Elasticsearch(
    ['elasticsearch']
)
JSON(es.info())
```

[3]:
```
<IPython.core.display.JSON object>
```

### 1.2 Importando los datos

En primer lugar vamos a descargar los datos usando el comando:

[4]:
```
!wget "https://gist.githubusercontent.com/aagea/
 ↪82d2eec2ecdcc49798a5707263ce5bb3/raw/
 ↪cc42b39d7a84ff170c8dc532d90ab458017b6eed/Employees50K.json"
```

```
--2024-02-18 17:01:34--  https://gist.githubusercontent.com/aagea/82d2eec2ecdcc4
9798a5707263ce5bb3/raw/cc42b39d7a84ff170c8dc532d90ab458017b6eed/Employees50K.jso
n
Resolving gist.githubusercontent.com (gist.githubusercontent.com)…
185.199.108.133, 185.199.110.133, 185.199.109.133, …
Connecting to gist.githubusercontent.com
(gist.githubusercontent.com)|185.199.108.133|:443… connected.
HTTP request sent, awaiting response… 200 OK
Length: 17054496 (16M) [text/plain]
Saving to: ‘Employees50K.json’

Employees50K.json   100%[===================>]  16.26M  96.8MB/s    in 0.2s

2024-02-18 17:01:35 (96.8 MB/s) - ‘Employees50K.json’ saved [17054496/17054496]
```

### 1.3 Preguntas

**(1 punto)** Carga los datos en elastic.

[5]:
```
es.indices.delete(index="companydatabase", ignore=[400, 404])
!curl -H "Content-Type: application/json" -XPOST "http://elasticsearch:9200/
 ↪companydatabase/_bulk?pretty" --data-binary "@Employees50K.json" >> /dev/null
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 34.5M  100 18.2M  100 16.2M   604k   538k  0:00:30  0:00:30 --:--:--
5229k:16  0:00:16 --:--:--     0
```

**(2 punto)** Recupera las mujeres mayores de 40 años.

[35]:
```
request_body = {
  "query": {
    "bool": {
      "must": [
        { "match": { "Gender": "Female" } },
        { "range": { "Age": { "gt": 40 } } }
      ]
    }
  }
}
```

```
JSON(es.search(index="companydatabase", body=request_body))
```

[35]: <IPython.core.display.JSON object>

(2 punto) Recuper los hombres que su nombre empieza por “Will”.

[36]:
```
JSON(es.indices.get_mapping(index="companydatabase"))
```

[36]: <IPython.core.display.JSON object>

[37]:
```
request_body={
  "query": {
      "bool":{
          "must":[
              {"term":{"Gender.keyword":"Male"}},
              {"prefix":{"FirstName.keyword":{
                "value":"will",
                "case_insensitive":True}}}
          ]
    }
  }
}
JSON(es.search(index="companydatabase", body=request_body))
```

[37]: <IPython.core.display.JSON object>

**(2 punto)** Calcula brecha salarial entre hombres y mujeres.

[38]:
```
es.indices.get_mapping(index="companydatabase")
```

[38]:
```
{'companydatabase': {'mappings': {'properties': {'Address': {'type': 'text',
     'fields': {'keyword': {'type': 'keyword', 'ignore_above': 256}}},
    'Age': {'type': 'long'},
    'DateOfJoining': {'type': 'date'},
    'Designation': {'type': 'text',
     'fields': {'keyword': {'type': 'keyword', 'ignore_above': 256}}},
    'FirstName': {'type': 'text',
     'fields': {'keyword': {'type': 'keyword', 'ignore_above': 256}}},
    'Gender': {'type': 'text',
     'fields': {'keyword': {'type': 'keyword', 'ignore_above': 256}}},
    'Interests': {'type': 'text',
     'fields': {'keyword': {'type': 'keyword', 'ignore_above': 256}}},
    'LastName': {'type': 'text',
     'fields': {'keyword': {'type': 'keyword', 'ignore_above': 256}}},
    'MaritalStatus': {'type': 'text',
     'fields': {'keyword': {'type': 'keyword', 'ignore_above': 256}}},
    'Salary': {'type': 'float'}}}}}
```

[39]:
```
mapping_type= {
  "mappings": {
    "properties": {
      "Address": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 256}}
      },
      "Age": {
        "type": "long"
      },
      "Designation": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 256}}
      },
      "FirstName": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 256}}
      },
        "Gender": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 256}}
      },
      "Interests": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 256 }}
      },
        "LastName": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 256}}
```

```
        },
        "MaritalStatus": {
        "type": "text",
        "fields": {
          "keyword": {
            "type": "keyword",
            "ignore_above": 256}}
      },
        "Salary": {
        "type": "float"
      }

    }
  }
}

es.indices.delete(index="companydatabase",ignore=[400,404])
es.indices.create(index="companydatabase",body=mapping_type)
```

[39]:
```
{'acknowledged': True, 'shards_acknowledged': True, 'index': 'companydatabase'}
```

[40]:
```
!curl -H "Content-Type: application/json" -XPOST "http://elasticsearch:9200/
 ↪companydatabase/_bulk?pretty" --data-binary "@Employees50K.json" >> /dev/null
```

```
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 34.5M  100 18.2M  100 16.2M  1253k  1117k  0:00:14  0:00:14 --:--:--
3984k:-- 5194k
```

[41]:
```
request_body = {
    "aggs": {
        "salaries_by_gender": {"terms": {"field": "Gender.keyword"},
            "aggs": { "average_salary": {"avg": {"field": "Salary"}}}
        }
    }
}
result=es.search(index="companydatabase", body=request_body)
male_avg_salary =␣
 ↪result['aggregations']['salaries_by_gender']['buckets'][0]['average_salary']['value']
female_avg_salary =␣
 ↪result['aggregations']['salaries_by_gender']['buckets'][1]['average_salary']['value']
brecha_salarial=male_avg_salary-female_avg_salary
print("La brecha salarial es: ",brecha_salarial)
```

```
La brecha salarial es:  126.27789588547603
```

**(3 punto)** Calcula cuales son los intereses más comunes de los empleados.

[42]:
```
request_body = {
    "aggs": {
        "common_interests": {"terms": {"field": "Interests.keyword"}}
    }
}
JSON(es.search(index="companydatabase", body=request_body))
```

[42]: <IPython.core.display.JSON object>