

Rapport :

Twitter Sentiment Analysis Using
Elasticsearch, Logstash and Kibana

Table de matières

Introduction	3
Partie 1 : Définitions et concepts	3
1.Elasticsearch	3
2.Logstash	3
3.Kibana	3
4.Twitter Developer Account	4
5.Twitter API Keys	4
 Partie 2 : Réalisation	 4
1.Environnement adopté	4
2.Réalisation de twitter sentiment analysis	5
a) Extraction du flux de données à partir de l'API	4
b) Analyse de sentiments	4
c) Indexation des données sur Elasticsearch	4
d) Création du Dashboard et visualisation avec Kibana	4
 Partie 3 : Exécution	 6
1. Exemple de « Congress ».....	4
2. Exemple de « SpiderManNoWayHome »	4

Introduction :

- Ce projet consiste à évaluer les sentiments des gens par rapport à un post sur twitter, le sentiment peut être soit positif, négatif ou neutre, pour cette raison on fera des visualisations sous forme de graphe afin d'avoir une vision globale par rapport au sujet.

Définitions et concepts :

1) Elasticsearch

- Un outil open source capable de recherche et indexation. C'est un serveur de recherche open source basé sur Apache Lucene. Il peut facilement accéder à JSON documents via l'API REST HTTP.

2) Logstash

- Un outil open source utilisé pour gérer les journaux et événements. Il peut envoyer des journaux à partir de n'importe quelle source comme TCP/UDP, fichiers, syslog, journaux d'événements, STDIN, socket web, Zerromq , analyser n'importe quelle entrée comme JSON, csv, XML, multi-ligne, Regex, Grok. Il obtient le bon horodatage, les indexe et leur permet de les rechercher et de les stocker.

3) Kibana

- Un outil d'analyse open source utilisé pour analyser, visualiser et rechercher des données. Il fournit d'énormes types de visualisations pour visualiser les données sous forme de graphiques, cartes, tableaux, histogrammes. Il facilite également l'interface pour le Web contenu basé qui peut gérer des données énormes avec facilité. En plus de créer des tableaux de bord, il peut également interroger en temps réel données, car les tableaux de bord fournissent une interface pour

JSON documents. Ils peuvent travailler avec les données Elasticsearch sans toute exigence de codage.

4) Twitter developer account

- Un compte développeur qui est utilisé pour gérer l'accès à l'API Twitter.
- La plate-forme de développement de Twitter vous permet d'exploiter la puissance de la plate-forme ouverte, globale, en temps réel et historique de Twitter au sein de vos propres applications. La plate-forme fournit des outils, des ressources, des données et des produits API pour vous permettre d'intégrer et d'étendre l'impact de Twitter grâce à la recherche, aux solutions et plus encore.

5) Twitter API keys

- L'API Twitter est un ensemble de points de terminaison programmatiques qui peuvent être utilisés pour comprendre ou construire la conversation sur Twitter.
- Cette API permet de rechercher et de récupérer, d'interagir avec ou de créer une variété de ressources différentes, notamment les suivantes :
 - Tweets
 - Utilisateurs
 - Les espaces
 - Messages directs
 - Listes
 - Les tendances
 - Médias
 - Des endroits.

Réalisation :

1) Environnement adopté

- Microsoft Visual Studio est un environnement de développement intégré (IDE) conçu comme son nom l'indique, par Microsoft. Il est utilisé pour le développement de programme informatique pour les systèmes d'exploitation Microsoft Windows, mais aussi pour le développement de site web, d'applications web et services web. Il peut produire aussi bien du code natif que du code managé. Visual Studio inclus un éditeur de code supportant IntelliSense.
- Elasticsearch version 7.15.2 pour windows-x64
- Kibana version 7.15.2 pour windows-x64

2) Réalisation de twitter sentiment analysis

a. Extraction du flux de données à partir de l'API

- Création d'un compte développeur twitter et avoir les codes API pour faire une extraction des données par python.

```
consumer_key = "  
consumer_secret = '  
access_token = "  
access_token_secret = "  
google_api_key = ""
```

- Au début de l'extraction, on choisit un mot clé ou un sujet qui nous intéresse à extraire ses données dans twitter.

```
def main():  
    topics = []  
    if len(sys.argv) == 1:  
        topics = ['Congress']  
    else:  
        for topic in sys.argv[1:]:  
            topics.append(topic)  
  
    index = "tweet-sentiment"  
    doc_type = "new-tweet"
```

- La classe GoogleAPIHandler possède une fonction get_geo_info qui a pour rôle de détecter la localisation des tweets.

```
class GoogleAPIHandler:
    def __init__(self, key):
        self.key = key

    def get_geo_info(self, latitude, longitude):

        country = None
        resp = None

        latlng = latitude + ',' + longitude

        link = ("https://maps.googleapis.com/maps/api/geocode/json?latlng=" +
                latlng + "&key=" + self.key)
        resp = requests.get(link)

        if resp:
            for result in resp.json()['results']:
                for item in result["address_components"]:
                    if 'country' in item['types']:
                        country = item["long_name"]
                        break
                    if country is not None:
                        break

        return country
```

b. Analyse des sentiments

- Parmi les fonctions que TweetStreamListener possèdent, il y'a celles de la détection et extraction et analyses des sentiments.
- On extrait et analyse les sentiments par get_sentiment

```
def _get_sentiment(self, decoded):
    tweet = TextBlob(decoded)

    subjectivity = tweet.sentiment.subjectivity
    polarity = tweet.sentiment.polarity

    if polarity < 0:
        sentiment = "negative"
    elif polarity == 0:
        sentiment = "neutral"
    else:
        sentiment = "positive"

    return polarity, subjectivity, sentiment
```

c. Indexation des données sur Elasticsearch

- Après l'extraction des sentiments, on fait un mapping puis on indexe ses sentiments dans Elasticsearch par la fonction on_data.

```
def on_data(self, data):

    print("> Retrievd a tweet")

    dict_data = (data)

    polarity, subjectivity, sentiment = self._get_sentiment(dict_data)
    print("[sentiment]", sentiment)
    print("[polarity]", polarity)

    doc = {
        "polarity": polarity,
        "subjectivity": subjectivity,
        "sentiment": sentiment,
    }

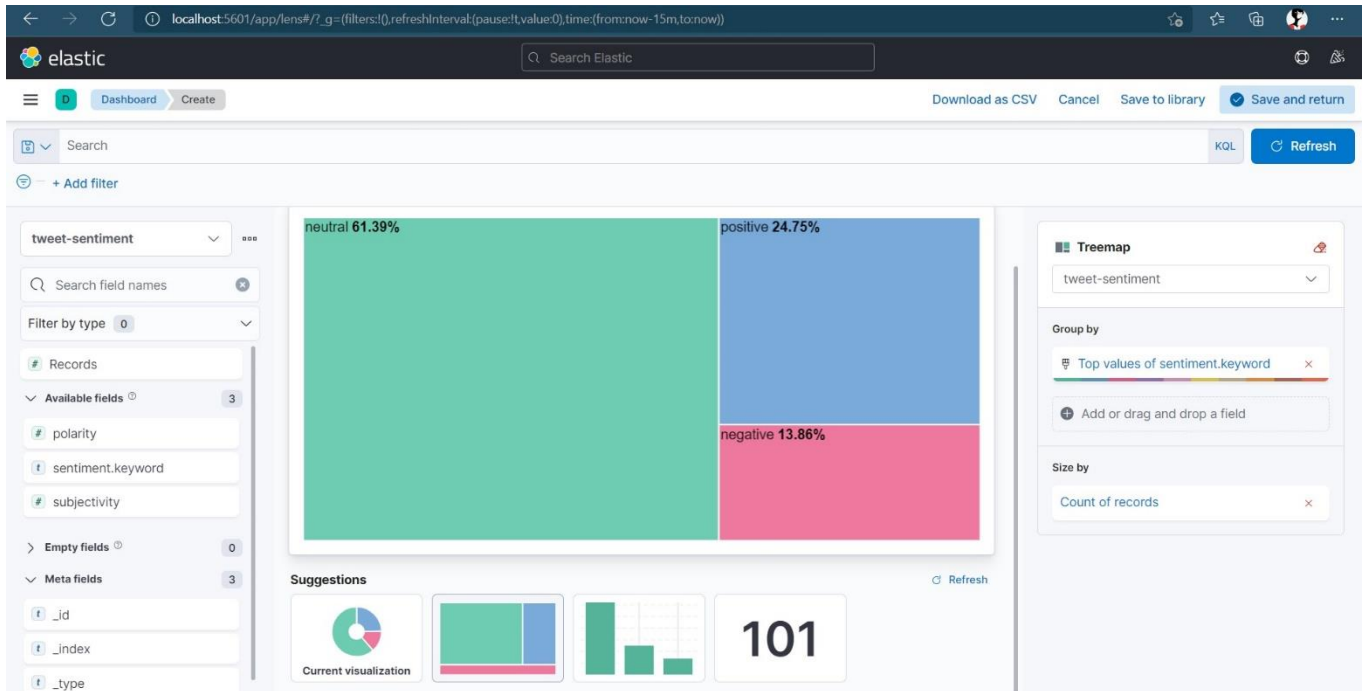
    es = Elasticsearch("http://elastic:changeme@localhost:9200")
    es.index(index=self.index,
            doc_type=self.doc_type,
            body=doc)

    return True
```

d. Création du Dashboard et visualisation avec Kibana

The screenshot shows the Kibana interface for the 'tweet-sentiment' index pattern. The left sidebar contains navigation links for Management, Ingest, Data, Alerts and Insights, and Kibana. The main content area displays the 'Fields (9)' tab, showing a table of field mappings from Elasticsearch.

Name ↑	Type	Format	Searchable	Aggregatable	Excluded
_id	_id		•	•	
_index	_index		•	•	
_score					
_source	_source				
_type	_type		•	•	
polarity	float		•	•	
sentiment	text		•		
sentiment.keyword	keyword		•	•	



Exécution :

1. Exemple de « Congress »

- On écrit le mot clé : congress dans la zone des sujets pour extraire les différents tweets et sentiments.
- On choisit le nom d'index : **tweet_congress_sentiment**.

```
topics = []
if len(sys.argv) == 1:
    topics = ['Congress']
else:
    for topic in sys.argv[1:]:
        topics.append(topic)

index = "tweet_congress_sentiment"
doc_type = "congress-tweet"
```


- On exécute et on remarque notre output qui contient plusieurs sentiments extraits.

```

PROBLEMS  OUTPUT  TERMINAL  DEBUG CONSOLE

=> Retrievd a tweet
[sentiment] positive
[polarity] 0.15
=> Retrievd a tweet
[sentiment] neutral
[polarity] 0.0
=> Retrievd a tweet
[sentiment] neutral
[polarity] 0.0
=> Retrievd a tweet
[sentiment] neutral
[polarity] 0.0
=> Retrievd a tweet
[sentiment] neutral
[polarity] 0.0
=> Retrievd a tweet
[sentiment] neutral
[polarity] 0.0
=> Retrievd a tweet
[sentiment] neutral
[polarity] 0.0
=> Retrievd a tweet
[sentiment] positive

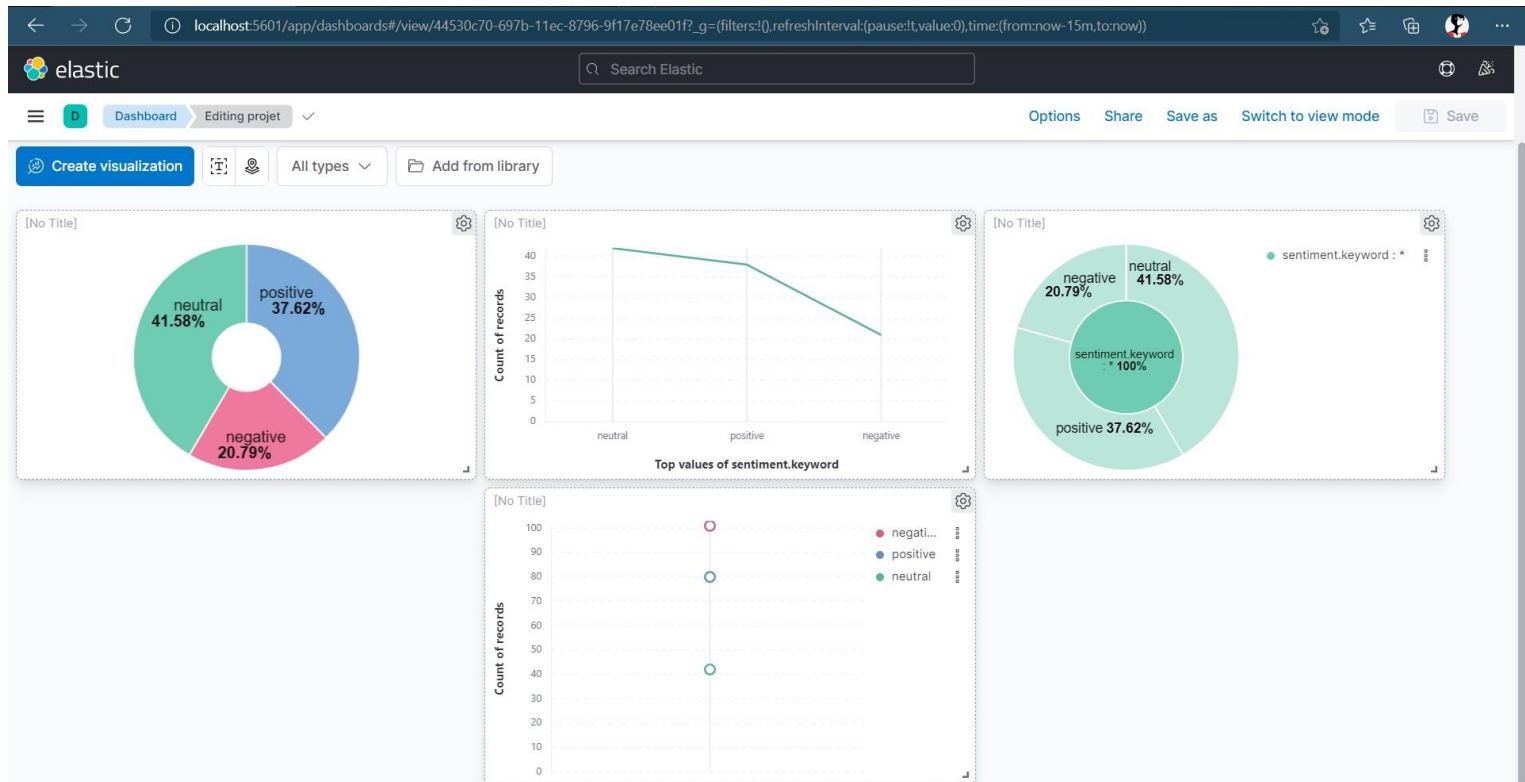
```

- On crée un index patterns :

The screenshot shows the Kibana interface for managing index patterns. The left sidebar contains navigation links for Management, Ingest, Data, Alerts and Insights, and Kibana. The main content area displays the 'tweet_congress_sentiment' index pattern. Below the title, there is a search bar and a table of fields. The table has columns for Name, Type, Format, Searchable, Aggregatable, and Excluded. The fields listed are: _id, _index, _score, _source, _type, polarity, sentiment, sentiment.keyword, and subjectivity.

Name	Type	Format	Searchable	Aggregatable	Excluded
_id	_id		•	•	
_index	_index		•	•	
_score					
_source	_source				
_type	_type		•	•	
polarity	float		•	•	
sentiment	text		•		
sentiment.keyword	keyword		•	•	
subjectivity	float		•	•	

- Passant au Dashboard pour faire une visualisation :



2. Exemple de « SpiderManNoWayHome »

- On écrit le mot clé : SpiderManNoWayHome dans la zone des sujets pour extraire les différents tweets et sentiments.
- On choisit le nom d'index : **Spidey_sentiment**.

```
topics = []
if len(sys.argv) == 1:
    # Default topics
    topics = ['SpiderManNoWayHome']
else:
    for topic in sys.argv[1:]:
        topics.append(topic)

index = "spidey-sentiment"
doc_type = "new-tweet"
```

- On exécute et on remarque notre output qui contient plusieurs sentiments extraits pour le nouveau film.

```
PS C:\Users\LENOVO> & C:\Users\LENOVO\AppData\Local\Programs\Python\Python39\python.exe "c:/Users/LENOVO/Desktop/Studies/twitter sentiment sim/main.py"
==> Topics ['SpiderManNoWayHome']
==> Index: spidey-sentiment, doc type: new-tweet
==> Start retrieving tweets...
=> Retrieved a tweet
[sentiment] positive
[polarity] 0.05833333333333334
c:\Users\LENOVO\Desktop\Studies\twitter sentiment sim\main.py:91: DeprecationWarning: The 'body' parameter is deprecated for the 'index' API and will be re
s://github.com/elastic/elasticsearch-py/issues/1698 for more information
  es.index(index=self.index,
C:\Users\LENOVO\AppData\Local\Programs\Python\Python39\lib\site-packages\elasticsearch\connection\base.py:200: ElasticsearchWarning: Elasticsearch built-in
could be accessible to anyone. See https://www.elastic.co/guide/en/elasticsearch/reference/7.15/security-minimal-setup.html to enable security.
  warnings.warn(message, category=ElasticsearchWarning)
C:\Users\LENOVO\AppData\Local\Programs\Python\Python39\lib\site-packages\elasticsearch\connection\base.py:200: ElasticsearchWarning: [types removal] Specif
points instead ({index}/{doc}/{id}, {index}/{_doc}, or {index}/{_create}/{id}).
  warnings.warn(message, category=ElasticsearchWarning)
=> Retrieved a tweet
[sentiment] positive
[polarity] 0.25
=> Retrieved a tweet
[sentiment] neutral
[polarity] 0.0
=> Retrieved a tweet
[sentiment] neutral
[polarity] 0.0
```

- On crée un index patterns :

The screenshot shows the Kibana interface for the 'spidey-sentiment' index pattern. The left sidebar contains navigation links for Data, Alerts and Insights, Kibana, and Stack. The main content area displays the index pattern name and a table of fields.

spidey-sentiment

View and edit fields in **spidey-sentiment**. Field attributes, such as type and searchability, are based on [field mappings](#) in Elasticsearch.

Fields (9) Scripted fields (0) Field filters (0)

Search All field types Add field

Name ↑	Type	Format	Searchable	Aggregatable	Excluded
_id	_id		•	•	
_index	_index		•	•	
_score					
_source	_source				
_type	_type		•	•	
polarity	float		•	•	
sentiment	text		•		
sentiment.keyword	keyword		•	•	
subjectivity	float		•	•	

- Passant au Dashboard pour faire une visualisation :

