الكلية متعددة التخصصات تارودانت
ⵜⴰⵢⵎⵍⴰⵢⵜ ⵜⴰⵣⵉⵣⴰⵡⵉⵜ ⵏ ⵜⵓⵙⵏⴰⵡⵜ
FACULTE POLYDISCIPLINAIRE TAROUDANT

Année universitaire 2022/2023

Challenges
Big data - 1
BIA 2022-2023

Encadré par :

- Pr. El HAJJI Mohamed

Réalisé par :

- BENAABIDATE Ahmed Walid.

Challenges

Big data - 1

BIA 2022-2023

Challenge 1

Proposer une solution pour mettre en œuvre un système d'analyse de sentiments basé sur une solution Big Data basée sur les commentaires des visiteurs de site web HESPRESS, qui supporte le data Ingestion, le streaming, le batch processing, et le dashboarding temps réel.

Solution

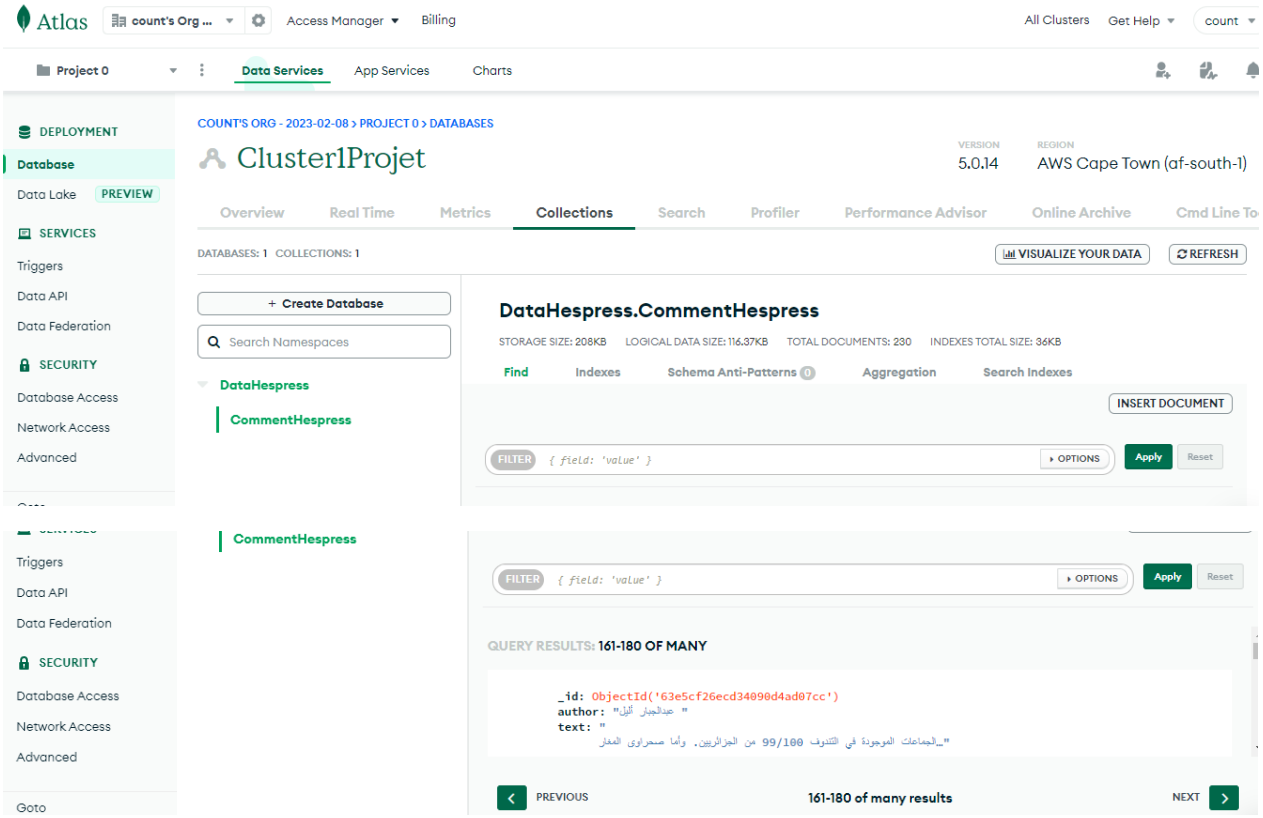
Pour mettre en œuvre un système d'analyse de sentiments pour les commentaires des visiteurs du site web HESPRESS, on a choisi une architecture qui permet de traiter et de visualiser les commentaires des visiteurs en temps réel pour déterminer leur sentiment.

Voilà l'architecture :

- Data Ingestion: Configurer Apache Kafka pour ingérer en temps réel les commentaires des visiteurs dans un cluster distribué.
- Streaming: Utiliser Apache Spark pour traiter les commentaires en temps réel, en utilisant les flux de données provenant d'Apache Kafka, et en les analysant pour déterminer le sentiment.
- Batch Processing: Configurer Apache Cassandra pour stocker les données des commentaires en grande quantité, ce qui permettra de traiter les données en lots pour obtenir des insights plus approfondis sur les tendances de sentiment au fil du temps.
- Dashboarding temps réel: Utiliser un outil de tableau de bord en temps réel pour visualiser les résultats de l'analyse de sentiments.

D'abord on a commencé par le Scraping des commentaires du site web **HEPRESS**

On a créé une base de données sur « **mongoDB »** pour stocker les commentaires



Récupération et stockage des commentaires de HESPRESS via l'utilisation de la technique de web Scraping dans MongoDB.

```python
import requests
from bs4 import BeautifulSoup
from pymongo import MongoClient

# Définir l'URL à scraper
url="https://www.hespress.com/%d8%a5%d9%86%d9%87%d8%a7%d8%a1-%d9%85%d9%87%d8%a7%d9%85-%d8%b3%d9%81%d9%8a%d8%b1-%d8%a7%d9%84%d9%85%d8%ba%d8%b1%d8%a8-%d8%a8%d9%81%d8%b1%d9%
# Faire une requête GET au site Web
response = requests.get(url)

# Analyser HTML en utilisant BeautifulSoup
soup = BeautifulSoup(response.text, "html.parser")

# Extraire les commentaires du HTML
comments = soup.find_all("div", class_="comments")

# print(comments)

# Connexion à l'instance MongoDB
client = MongoClient("mongodb+srv://walid:123@cluster1projet.84eisxu.mongodb.net/?retryWrites=true&w=majority")

# Obtenir la collection de commentaires
comments_collection = client.DataHespress.CommentHespress

comments_to_insert = []

# Parcourez les commentaires et extrayez le texte des deux éléments div
for comment in comments:
    for ul in comment.find_all('ul'):
        for li in ul.find_all('li'):
            div_container = li.find('div', {'class': 'comment-body'})
            div1 = div_container.find('div', {'class': 'comment-head'})
            div2 = div_container.find('div', {'class': 'comment-text'})
            comment_obj = {"author": div1.text.split("\n")[2], "text": div2.text}
            comments_to_insert.append(comment_obj)

# insérer tous les commentaires
comments_collection.insert_many(comments_to_insert)
```

Les commentaires extraits de la page Web HESPRESS sont ingérés dans un topic Kafka appelé « **topicExam** »

```python
from confluent_kafka import Producer, Consumer
import json

# Kafka producer
producer = Producer({'bootstrap.servers': 'localhost:9092'})

# Parcourez les commentaires et extrayez le texte des deux éléments div
for comment in comments:
    for ul in comment.find_all('ul'):
        for li in ul.find_all('li'):
            div_container = li.find('div', {'class': 'comment-body'})
            div1 = div_container.find('div', {'class': 'comment-head'})
            div2 = div_container.find('div', {'class': 'comment-text'})
            comment_obj = {"author": div1.text.split("\n")[2], "text": div2.text}
            comment_json = json.dumps(comment_obj)
            producer.produce('topicExam', value=bytes(comment_json, 'utf-8'))
            producer.flush()
            print(f'Published message: {comment_json}')


# Kafka consumer
conf = {'bootstrap.servers': 'localhost:9092',
        'group.id': 'mygroup',
        'auto.offset.reset': 'earliest'}

consumer = Consumer(conf)

consumer.subscribe(['topicExam'])
```

```python
while True:
    msg = consumer.poll(1.0)
    if msg is None:
        continue
    if msg.error():
        if msg.error().code() == KafkaError._PARTITION_EOF:
            print('Reached end of topic {} [{}] at offset {}'.format(
                msg.topic(), msg.partition(), msg.offset()))
        else:
            print('Error occured: {}'.format(msg.error()))
    else:
        comment = json.loads(msg.value())
        print(comment)
consumer.close()
```


```
Published message: {"author": "\u0632\u0643\u0631\u064a\u0627\u0621 ", "text": "\n\u0641\u0631\u0646\u0633\u0627 \u0643\u0627\u0646\u062a \u062f\u0627\u0626\u0645\u0627 \u0636\u062f \u0645\u0635\u0627\u0644\u062d \u0627\u0644\u0645\u063a\u0631\u0628 \u0630\u0644\u0643 \u0645\u0639\u0631\u0648\u0641 \u0628\u0642\u0631\u0627\u0631\u0627\u062a\u0647\u0627 \u0648\u062a\u0635\u0631\u064a\u062d\u0627\u062a \u0627\u0644\u0628\u0631\u0644\u0645\u0627\u0646\u064a\u064a\u0646 \u0627\u0644\u0641\u0631\u0646\u0633\u064a\u0646 \u0641\u064a \u0627\u0644\u0628\u0631\u0644\u0645\u0627\u0646 \u0627\u0644\u0623\u0648\u0631\u0648\u0628\u064a \u0639\u062f\u0645 \u0641\u062a\u062d \u0642\u0646\u0635\u0644\u064a\u0629 \u0641\u0631\u0646\u0633\u064a\u0629 \u0641\u064a \u0627\u0644\u0635\u062d\u0631\u0627\u0621 \u0627\u0644\u0645\u063a\u0631\u0628\u064a\u0629 \u0643\u0627\u0646\u062a \u0625\u0634\u0627\u0631\u0629 \u0641\u0631\u0646\u0633\u064a\u0629 \u0648\u0627\u0636\u062d\u0629 \u0623\u0646\u0647\u0627 \u0636\u062f \u0627\u0644\u0645\u063a\u0631\u0628 \u0648\u0648\u062d\u062f\u0629 \u062a\u0631\u0627\u0628\u0647\n"}
Published message: {"author": "\u0627\u0644\u0649 \u0627\u0644\u0627\u0645\u0627\u0645 \u064a\u0627 \u062d\u0643\u0627\u0645 \u0627\u0644\u062c\u0632\u0627\u0626\u0631 ", "text": "\n\u0627\u0644\u0645\u063a\u0631\u0628 \u0623\u0646\u0647\u0649 \u0645\u0647\u0627\u0645 \u0627\u0644\u0633\u0641\u064a\u0631 \u0627\u0644\u0645\u063a\u0631\u0628\u064a \u0641\u064a \u0641\u0631\u0646\u0633\u0627 \u0648\u0627\u0644\u062c\u0632\u0627\u0626\u0631 \u0627\u0633\u062a\u062f\u0639\u062a \u0633\u0641\u064a\u0631\u0647\u0627 \u0628\u0628\u0627\u0631\u064a\u0633 \u0644\u0644\u062a\u0634\u0627\u0648\u0631 \u060c \u0627\u0646\u0647\u0648\u0627 \u0645\u0647\u0627\u0645\u0647 \u0627\u0646\u062a\u0645 \u0627\u064a\u0636\u0627 \u0648\u0627\u0641\u062a\u062d\u0648\u0627 \u0627\u0644\u062d\u062f\u0648\u062f \u0648\u0627\u0644\u0627\u062c\u0648\u0627\u0621 \u0645\u0639 \u0627\u0644\u0645\u063a\u0631\u0628 \u062d\u064a\u0646\u0647\u0627 \u062a\u0643\u0648\u0646\u0648\u0627 \u0642\u062f \u0627\u0646\u062a\u0642\u0645\u062a\u0645 \u0644\u062a\u0627\u0631\u064a\u062e\u0643\u0645 \u0627\u064a\u0647\u0627 \u0627\u0644\u062c\u0632\u0627\u0626\u0631\u064a\u0648\u0646 \u0645\u0646 \u0641\u0631\u0646\u0633\u0627\n"}
{'author': 'زكرياء ', 'text': '\nفرنسا كانت دائما ضد مصالح المغرب ذلك معروف بقراراتها وتصريحات البرلمانيين الفرنسين في البرلمان الأوروبي عدم فتح قنصلية فرنسية في الصحراء المغربية كانت إشارة فرنسية واضحة أنها ضد المغرب ووحدة ترابه\n'}
{'author': 'صفريوي ', 'text': '\nاعتقد ان علاقة القصر الملكي بالاليزي وثيقة وقوية عكس مايزال بته ولعل ارقام التبادل الاقتصادي الاخير خير دليل على ارتباط رجال الاعمال المغاربة بفرنسا وعملهم الذوب على عدم فقدان السوق الفرنسية . المؤشر الثالث هو التاثيرات الفرنسية التي عادت الى التصاعد\nان يعود السفير او لا يعود فهو لن يفيد شي لان عملهم ليس بالطموح التي يتوخاه المواطن او المهاجر المقيم هناك، فالارقام المالية والعملة الصعبة هي التي تسيل لعاب والمتحكم في كل شي.\n'}
{'author': 'kala ', 'text': '\nفلنكن صريحين عمر فرنسا كانت صديقة، راعي البلاد نصره الله كان لهم بالمثل مصلحة مقابل مصلحة، لسنا الجزائر حتي نتنازل عن قراراتنا السياسة، نحن شعب الحسن رحمه الله وشعب محمدنا السادس ،والويل لمن اراد بنا شرا، وليحفظ الله محمدنا السادس من كيد الكائدين لمن قريب ولمن بعيد والويل للخونة ، وعاش المغرب تحت راية العلويين الاشراف\n'}
{'author': 'محمد الفاسي ', 'text': '\nفرنسا لازالت تعيش أوهام الاستعمار ولم تتخلص بعد من ماضيها الملطخ بدماء الأبرياء خصوصا في القارة الأفريقية وتريد احياء السيطرة والهيمنة من جديد وتنظر إلى مستعمراتها السابقة باستعلاء كبير ومقزز. يجب على جميع الأحرار مقاطعة هذه الدولة المستبدة.\n'}
{'author': 'ramzi ', 'text': '\nالمغرب دولة قوية الحمد لله وتتمتع باطر شابة يعول عليها في جميع المجالات السياسية الاقتصادية والخارجية اللهم بارك وحفظكم وحفظ جلالة الملك المفدى حفظه الله محمد السادس نصره\nالله\nتحية للمغاربة الأحرار ولكل المتتبعين تحية وتقدير\n'}
{'author': 'الرباط ici عبداللطيف ', 'text': '\nتركيع فرنسا المتعجرفة هو بيد مستعمراتها السابقة في قارة أفريقيا إن قامت بالمطالبة باسترجاع خيراتها المنهوبة إبان احتلالها من طرف الدولة الفرنسية.عندئذ سيتم ترييش الديك الفرنسي وسيتم الكشف عن عوراته. \n'}
```

On utilise Spark pour traiter les données par lots en groupant les commentaires selon l'auteur et en transformant les valeurs en une liste.

```
from pyspark.sql import SparkSession

# Créer une session Spark
spark = SparkSession.builder.appName("BatchProcessing").getOrCreate()

# Charger les données du fichier JSON local dans un DataFrame
df = spark.read.option("multiline","true").json(r"C:\Users\hp2\Desktop\CommentHespress.json")

# Effectuer un traitement par lots sur les données
result = df.rdd\
    .map(lambda x: (x.author, x.text))\
    .groupByKey()\
    .mapValues(list)\
    .collect()

# Show result
for author, text in result:
    print(f"Author: {author}\nText: {text}\n")

# Afficher le résultat
spark.stop()
```

```
Author: بوخلفة
Text: ['\n اغلبية اتفاقيات المغرب مع الاوربيين. هي اتفاقيات المغرب خاسر واوربا رابحة. لا توجد اتفاقية رابح رابح. المغرب اجر سواحله للبواخر الأوربية ب 30مليون يورو على مدى خمس سنوات. في الوقت الذي تربح اوربا 4. ملايير يزرو من سمك المغرب سنويا. انه اكبر نهب لمقدرات المغاربة. 30 مليون يوروا لا تبني حتى ملعب او مستشفى متوسط او طريق بطول 10كلم \n']

Author: Farid
Text: ['\nهذاك من يعطي صورة جميلة عن بلده وهناك من يشوه\n.المافيا المغربية في هولندا . التغلغل الفساد القتل الجريمة المنظمة. المخدرات. ولهذا وهولاندا تريد إرجاع هؤلاء المجرمين إلى وطنهم الام. لاحول ولا قوة الا بالله\n. لأن\n.من الصعب لان هناك شناقة وسطاء في كل شيء. واحد يبيع لاخر وهكذا. الاهم هو الضغط على شركات المحروقات لإرجاع ثمن المحروقات الى ثمنه الاصلي\n', '\n.هيا ويخبشها بأعمال إجرامية المحروقات عصب الحياة .\n سف لو كنا شعبا واعيا لقاطعنا محطة بعينها لمدة طويلة ولجعلنا يوم الاحد يوم بدون محطة أو بدون سيارة لضرب مداخيل المحطات من جهة و للحد من التلوث من جهة أخرى .\n']

Author: عبدالله768
Text: ['\nلماذا تحتقرون المغرب\nمن قبل لم تكن اتفاقية لتبادل المجرمين اما اذا تم توقيعها فسترى هل سيتم تسليمه ام لا ما عدا اذا مهدت له هولندا الطريق للهروب لبلد آخر\nردا على الذي يدافع عن شعو\n']
```

On utilise Spark pour effectuer un traitement par lots sur les données en appliquant une analyse des sentiments à l'aide de la bibliothèque « **TextBlob** »

```
from pyspark.sql import SparkSession
from textblob import TextBlob

# Créer une session Spark
spark = SparkSession.builder.appName("BatchProcessing").getOrCreate()

# Charger les données du fichier JSON local dans un DataFrame
df = spark.read.option("multiline","true").json(r"C:\Users\hp2\Desktop\CommentHespress.json")

# Effectuer un traitement par lots sur les données
result = df.rdd\
    .map(lambda x: (x.author, x.text))\
    .groupByKey()\
    .mapValues(list)\
    .collect()

# Effectuez une analyse des sentiments à l'aide de TextBlob
for author, text in result:
    sentiment_scores = [TextBlob(t).sentiment.polarity for t in text]
    average_sentiment = sum(sentiment_scores) / len(sentiment_scores)

    print(f"Author: {author}\nAverage sentiment: {average_sentiment}\n")

# Arrêter la session Spark
spark.stop()
```

```
Author: الفتنة
Average sentiment: 0.0


Author: العريشي
Average sentiment: -0.75

Author: زكرياء
Average sentiment: 0.0

Author: صفريوي
Average sentiment: 0.0

Author: kala
Average sentiment: 0.0
```