Université Chouaib Doukkali
Ecole Nationale des Sciences Appliquées d'El Jadida

MODULE :

ARCHITECTURE ET INFRASTRUCTURE BIG DATA

Filière : **2ITE3**

Projet 10 :
Mise en place d'un data lake pour la visualisation des données financières en utilisant Apache Kafka, Apache Spark, Apache Beam, Apache Druid et Streamlit

Réalisé Par :
AMINE ELMANSOURI
CHARAFEDDINE ELBAHJA
YASSINE HASSINI

Professeur :
Pr. FAHD KALLOUBI

ANNEE UNIVERSITAIRE : 2023-2024

Table des matières

Liste des Figures

1. Objectif

Ce projet vise à établir un data lake pour visualiser des données financières en utilisant Apache Spark, Apache Beam, Apache Druid, et Streamlit. Deux sources de données, dont Yahoo Finance et l'API New York Times, sont intégrées via Apache Beam et enregistrées au format Parquet. En utilisant Spark ML, plusieurs modèles de sentiment analysis sont entraînés sur un ensemble de données de Kaggle, et le meilleur modèle est retenu. Les données sont ensuite analysées avec Spark SQL pour prédire les scores de sentiment NYT, et le résultat est enregistré dans Apache Druid. Enfin, Streamlit offre une interface interactive pour explorer les résultats, incluant le tableau de données et les statistiques de sentiment par jour etc...

2. Installation du docker et Docker compose

L'installation de Docker et Docker Compose se fait en plusieurs étapes. Voici une brève description des étapes pour l'installation de Docker et Docker Compose :

1. Installation de Docker :

- Rendez-vous sur le site officiel de Docker pour télécharger la version appropriée de Docker pour votre système d'exploitation : https://docs.docker.com/get-docker/
- Suivez les instructions d'installation spécifiques à votre système d'exploitation.

2. Vérification de l'installation de Docker :

- Après l'installation, exécutez la commande suivante dans un terminal ou une invite de commandes pour vérifier que Docker est correctement installé :
  - ⇨ **docker --version**

3. Installation de Docker Compose :

- Docker Compose est généralement inclus avec Docker sur la plupart des plateformes. Si ce n'est pas le cas, suivez les instructions spécifiques à votre système d'exploitation pour installer Docker Compose : **https://docs.docker.com/compose/install/**

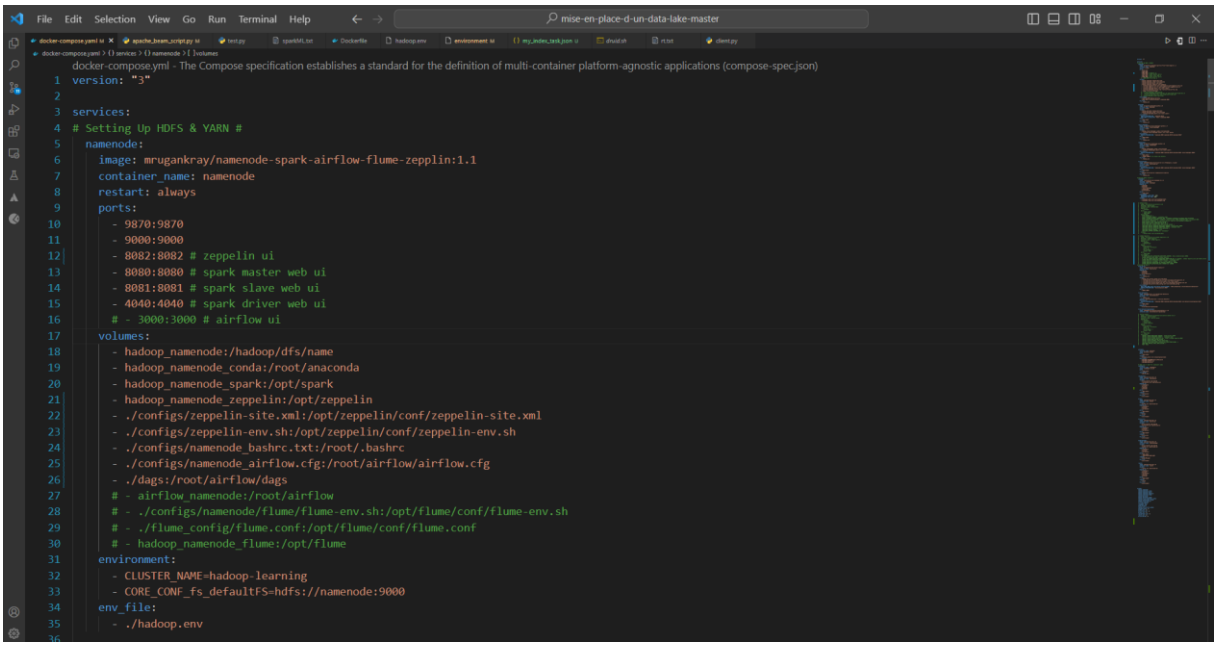
4. Vérification de l'installation de Docker Compose :

- Après l'installation, exécutez la commande suivante pour vérifier que Docker Compose est correctement installé :
  - ⇨ **docker-compose –version**

3. Configuration de l'Environnement Dockerisé

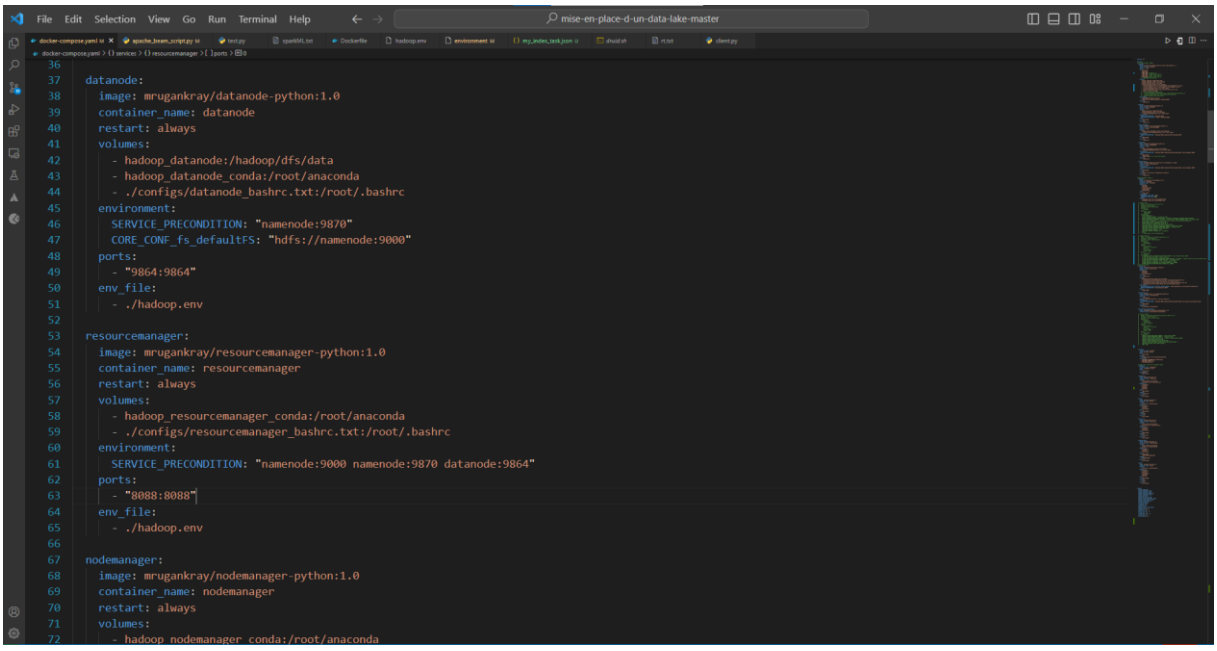
1. Création des Conteneurs

Élaborez un fichier docker-compose.yml décrivant les services nécessaires pour cette application se référer au fichier docker-compose.yml dans le dossier du projet.



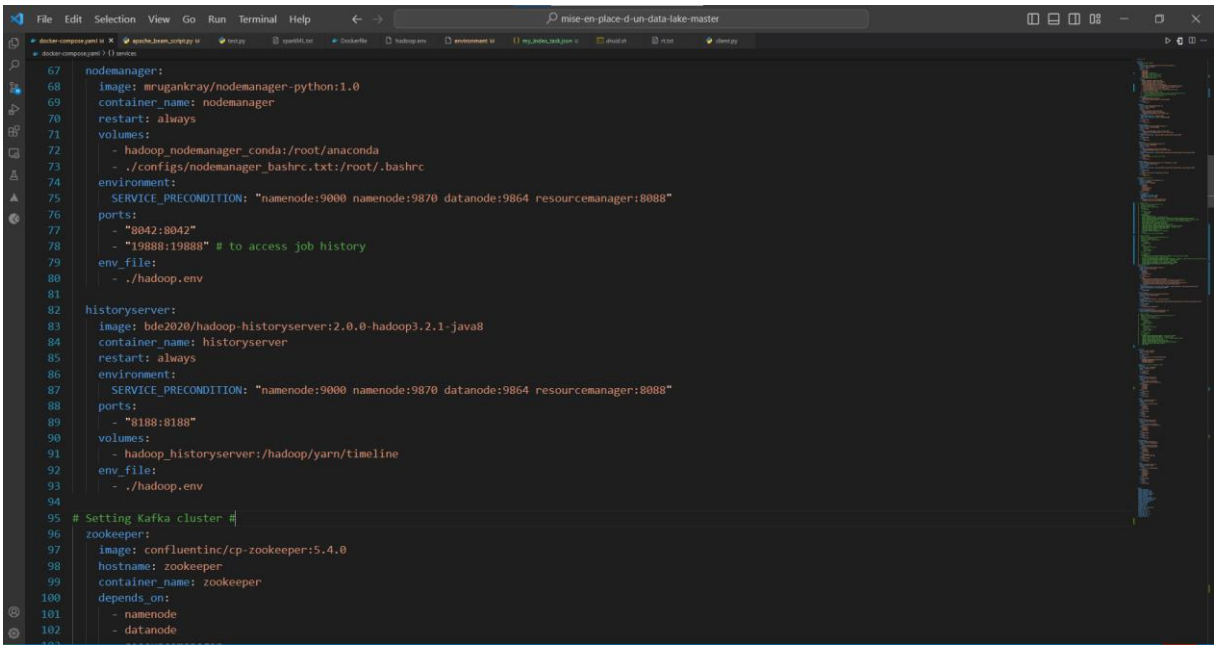
```yaml
version: "3"

services:
# Setting Up HDFS & YARN #
  namenode:
    image: mrugankray/namenode-spark-airflow-flume-zepplin:1.1
    container_name: namenode
    restart: always
    ports:
      - 9870:9870
      - 9000:9000
      - 8082:8082 # zeppelin ui
      - 8080:8080 # spark master web ui
      - 8081:8081 # spark slave web ui
      - 4040:4040 # spark driver web ui
      # - 3000:3000 # airflow ui
    volumes:
      - hadoop_namenode:/hadoop/dfs/name
      - hadoop_namenode_conda:/root/anaconda
      - hadoop_namenode_spark:/opt/spark
      - hadoop_namenode_zeppelin:/opt/zeppelin
      - ./configs/zeppelin-site.xml:/opt/zeppelin/conf/zeppelin-site.xml
      - ./configs/zeppelin-env.sh:/opt/zeppelin/conf/zeppelin-env.sh
      - ./configs/namenode_bashrc.txt:/root/.bashrc
      - ./configs/namenode_airflow.cfg:/root/airflow/airflow.cfg
      - ./dags:/root/airflow/dags
      # - airflow_namenode:/root/airflow
      # - ./configs/namenode/flume/flume-env.sh:/opt/flume/conf/flume-env.sh
      # - ./flume_config/flume.conf:/opt/flume/conf/flume.conf
      # - hadoop_namenode_flume:/opt/flume
    environment:
      - CLUSTER_NAME=hadoop-learning
      - CORE_CONF_fs_defaultFS=hdfs://namenode:9000
    env_file:
      - ./hadoop.env
```

Figure 1:Le contenu de fichier docker-compose.yml-partie1



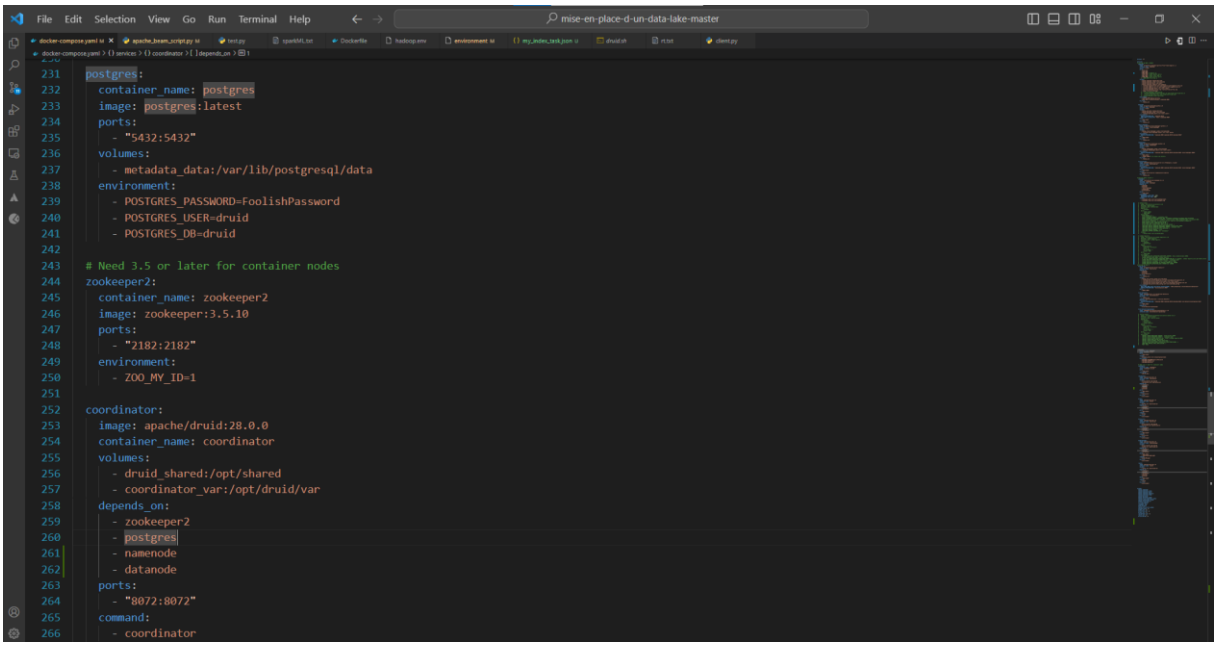
```yaml
  datanode:
    image: mrugankray/datanode-python:1.0
    container_name: datanode
    restart: always
    volumes:
      - hadoop_datanode:/hadoop/dfs/data
      - hadoop_datanode_conda:/root/anaconda
      - ./configs/datanode_bashrc.txt:/root/.bashrc
    environment:
      SERVICE_PRECONDITION: "namenode:9870"
      CORE_CONF_fs_defaultFS: "hdfs://namenode:9000"
    ports:
      - "9864:9864"
    env_file:
      - ./hadoop.env

  resourcemanager:
    image: mrugankray/resourcemanager-python:1.0
    container_name: resourcemanager
    restart: always
    volumes:
      - hadoop_resourcemanager_conda:/root/anaconda
      - ./configs/resourcemanager_bashrc.txt:/root/.bashrc
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864"
    ports:
      - "8088:8088"
    env_file:
      - ./hadoop.env

  nodemanager:
    image: mrugankray/nodemanager-python:1.0
    container_name: nodemanager
    restart: always
    volumes:
      - hadoop_nodemanager_conda:/root/anaconda
```

Figure 2:Le contenu de fichier docker-compose.yml-partie2



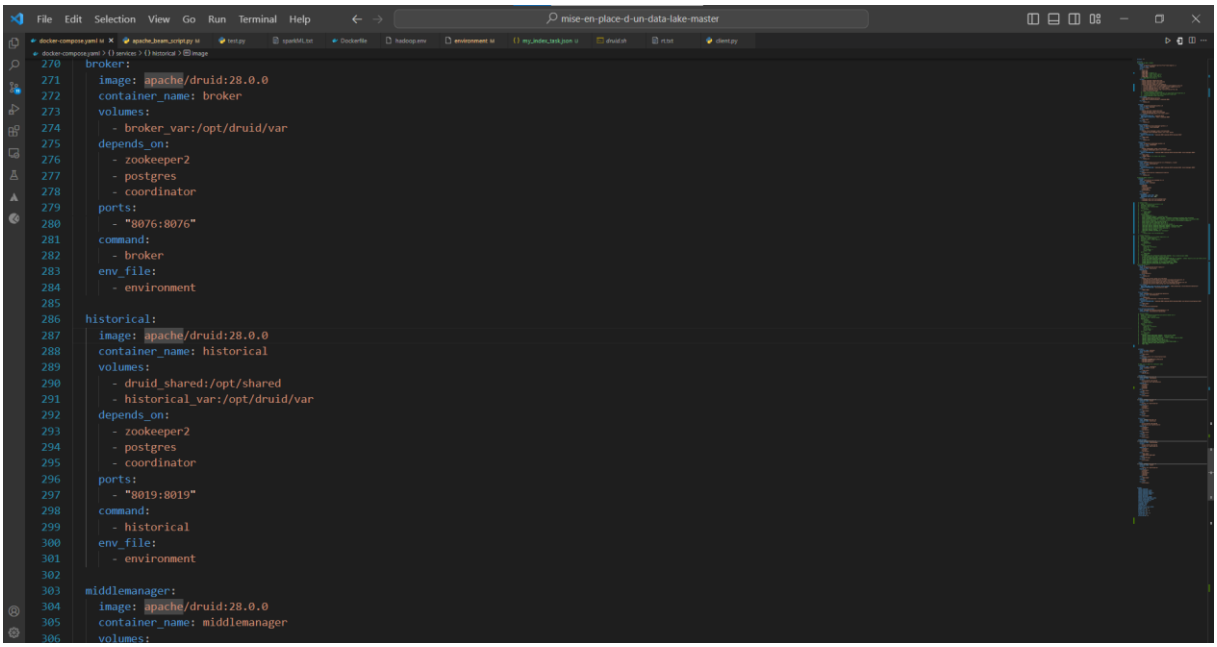
```
  nodemanager:
    image: mrugankray/nodemanager-python:1.0
    container_name: nodemanager
    restart: always
    volumes:
      - hadoop_nodemanager_conda:/root/anaconda
      - ./configs/nodemanager_bashrc.txt:/root/.bashrc
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864 resourcemanager:8088"
    ports:
      - "8042:8042"
      - "19888:19888" # to access job history
    env_file:
      - ./hadoop.env

  historyserver:
    image: bde2020/hadoop-historyserver:2.0.0-hadoop3.2.1-java8
    container_name: historyserver
    restart: always
    environment:
      SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864 resourcemanager:8088"
    ports:
      - "8188:8188"
    volumes:
      - hadoop_historyserver:/hadoop/yarn/timeline
    env_file:
      - ./hadoop.env

# Setting Kafka cluster #
  zookeeper:
    image: confluentinc/cp-zookeeper:5.4.0
    hostname: zookeeper
    container_name: zookeeper
    depends_on:
      - namenode
      - datanode
```

Figure 3:Le contenu de fichier docker-compose.yml-partie3



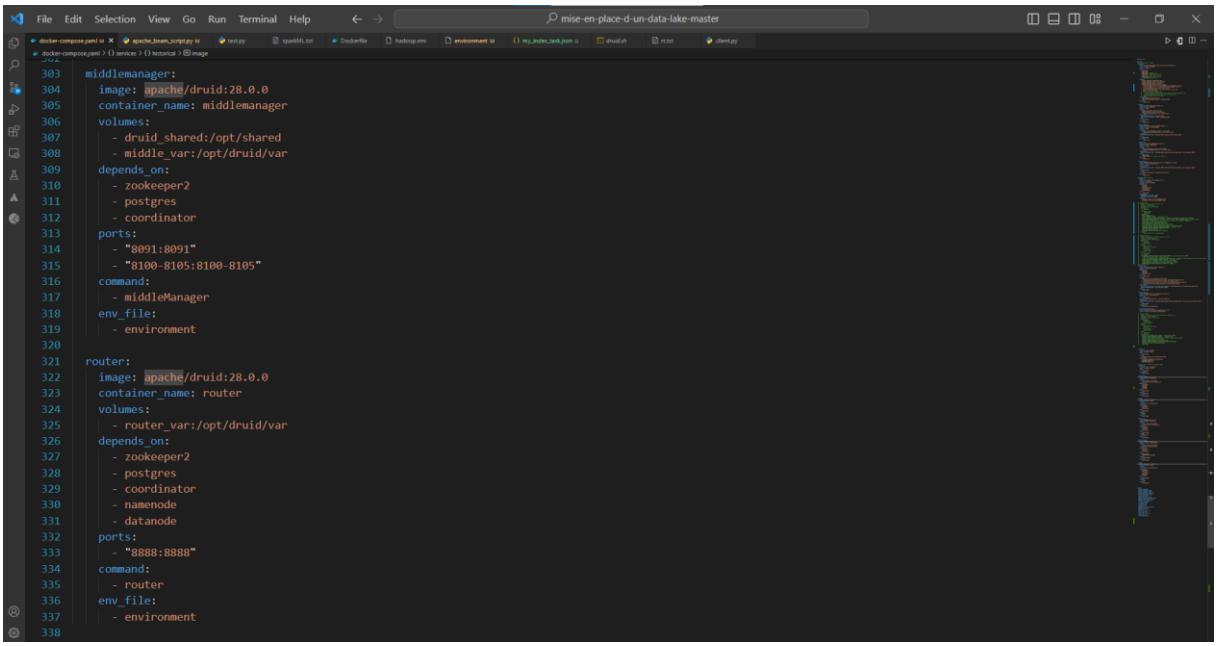
```
  postgres:
    container_name: postgres
    image: postgres:latest
    ports:
      - "5432:5432"
    volumes:
      - metadata_data:/var/lib/postgresql/data
    environment:
      - POSTGRES_PASSWORD=FoolishPassword
      - POSTGRES_USER=druid
      - POSTGRES_DB=druid

  # Need 3.5 or later for container nodes
  zookeeper2:
    container_name: zookeeper2
    image: zookeeper:3.5.10
    ports:
      - "2182:2182"
    environment:
      - ZOO_MY_ID=1

  coordinator:
    image: apache/druid:28.0.0
    container_name: coordinator
    volumes:
      - druid_shared:/opt/shared
      - coordinator_var:/opt/druid/var
    depends_on:
      - zookeeper2
      - postgres
      - namenode
      - datanode
    ports:
      - "8072:8072"
    command:
      - coordinator
```

Figure 4:Le contenu de fichier docker-compose.yml-partie4



```
  broker:
    image: apache/druid:28.0.0
    container_name: broker
    volumes:
      - broker_var:/opt/druid/var
    depends_on:
      - zookeeper2
      - postgres
      - coordinator
    ports:
      - "8076:8076"
    command:
      - broker
    env_file:
      - environment

  historical:
    image: apache/druid:28.0.0
    container_name: historical
    volumes:
      - druid_shared:/opt/shared
      - historical_var:/opt/druid/var
    depends_on:
      - zookeeper2
      - postgres
      - coordinator
    ports:
      - "8019:8019"
    command:
      - historical
    env_file:
      - environment

  middlemanager:
    image: apache/druid:28.0.0
    container_name: middlemanager
    volumes:
```

Figure 5:Le contenu de fichier docker-compose.yml-partie5



```
  middlemanager:
    image: apache/druid:28.0.0
    container_name: middlemanager
    volumes:
      - druid_shared:/opt/shared
      - middle_var:/opt/druid/var
    depends_on:
      - zookeeper2
      - postgres
      - coordinator
    ports:
      - "8091:8091"
      - "8100-8105:8100-8105"
    command:
      - middleManager
    env_file:
      - environment

  router:
    image: apache/druid:28.0.0
    container_name: router
    volumes:
      - router_var:/opt/druid/var
    depends_on:
      - zookeeper2
      - postgres
      - coordinator
      - namenode
      - datanode
    ports:
      - "8888:8888"
    command:
      - router
    env_file:
      - environment
```

Figure 6:Le contenu de fichier docker-compose.yml-partie6

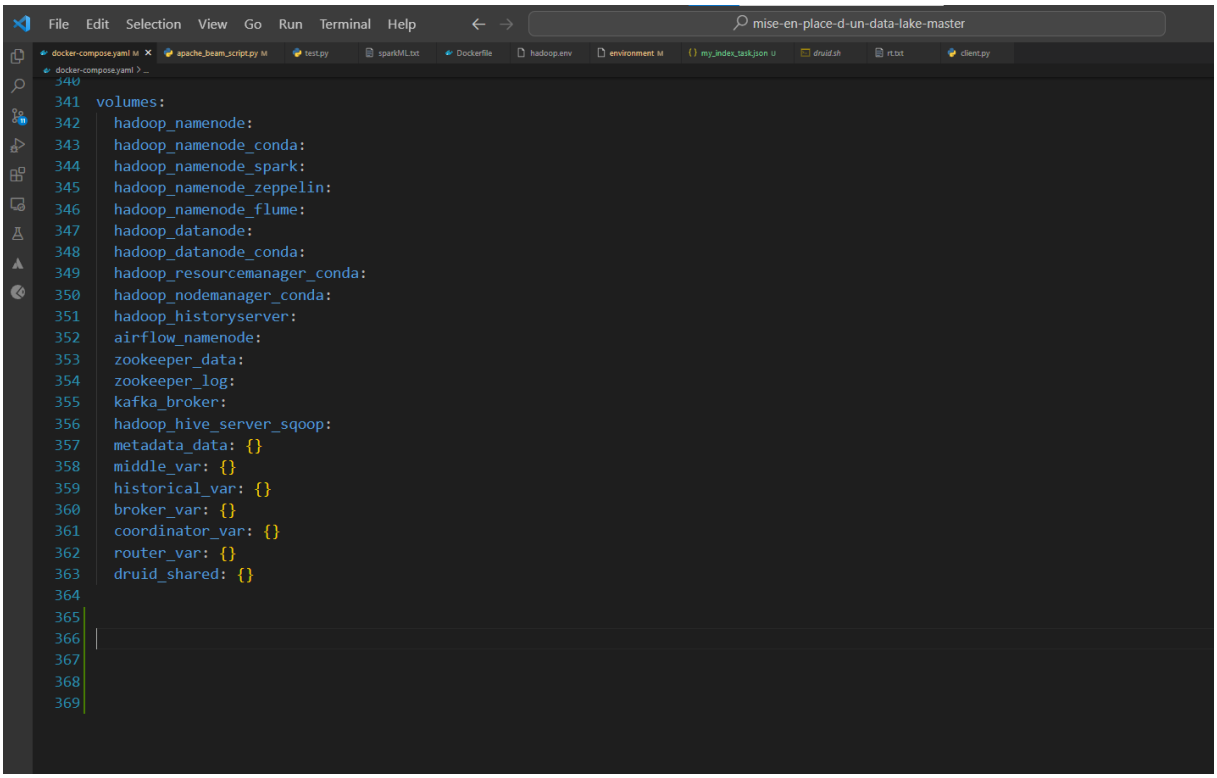


Figure 7:Les volumes des conteneurs

2. Exécution des conteneurs

Pour lancer les conteneurs spécifiés dans votre fichier docker-compose.yml et les exécuter en mode arrière-plan, vous devez utiliser la commande suivante : **docker-compose up -d.**

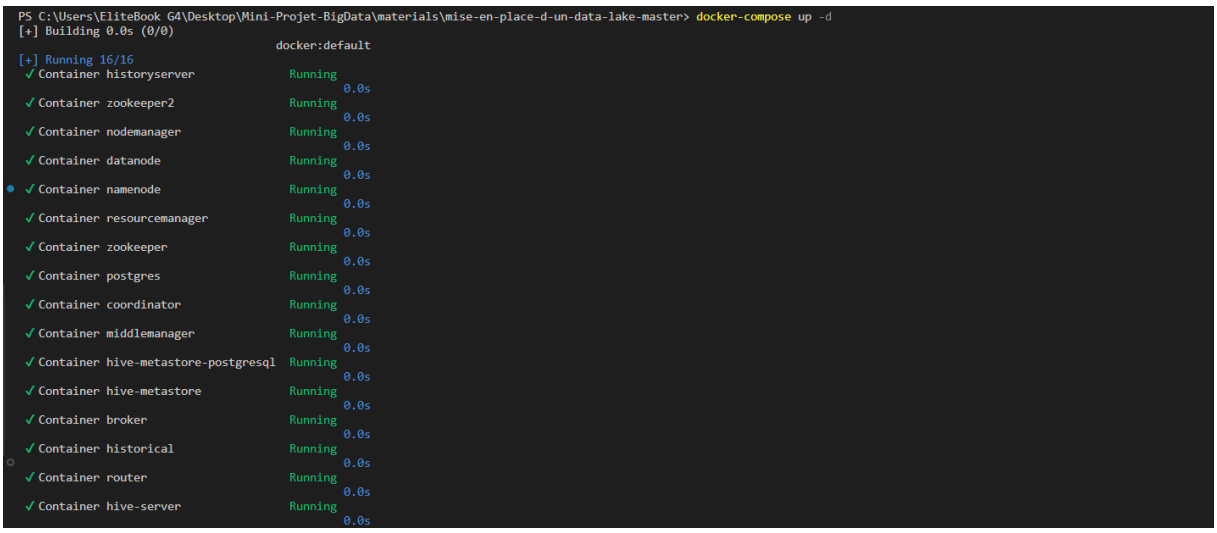
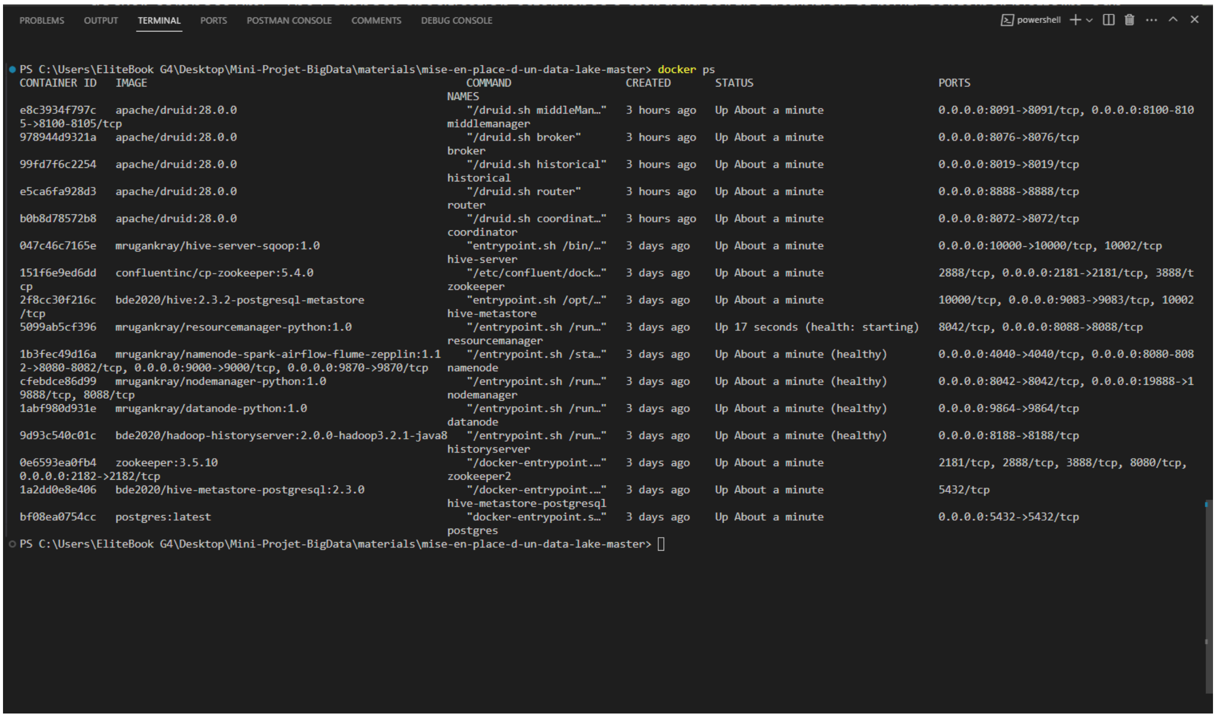


Figure 8:Exécution des conteneurs

Une fois les conteneurs démarrés, vous avez la possibilité d'afficher les identifiants des conteneurs actifs et leurs ports respectifs en exécutant la commande : **docker ps**.



Ou bien vous pouvez consulter l'interface de Docker Desktop.

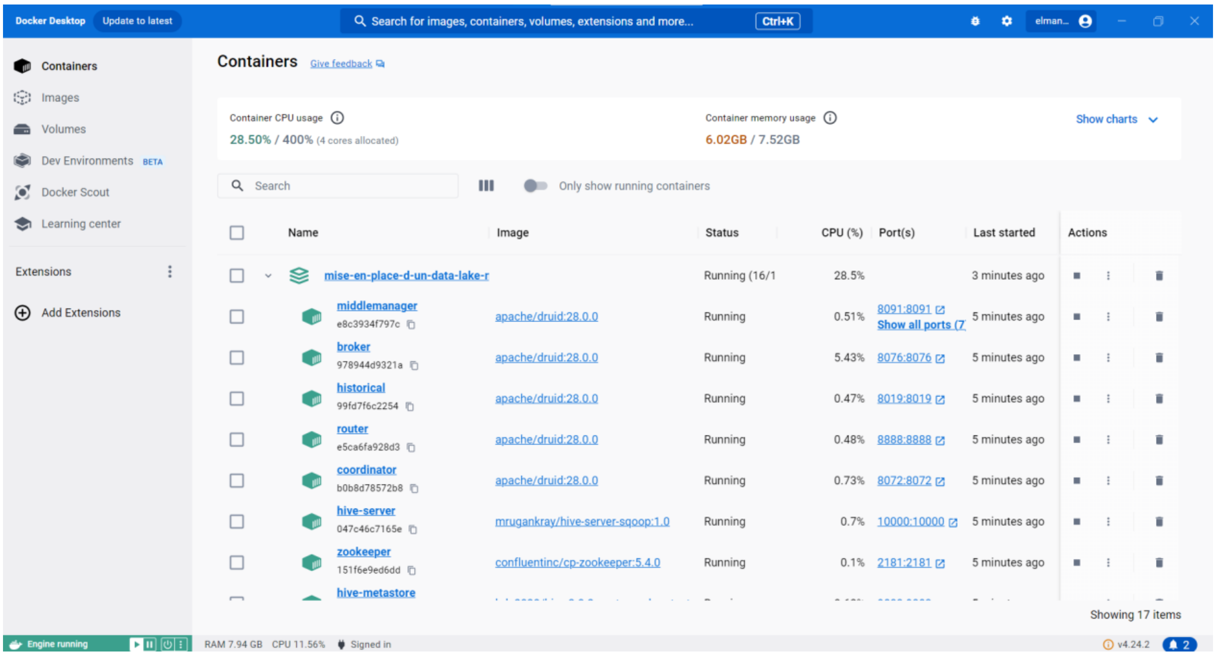


Figure 9:interface de docker Desktop

4. Configuration des api New York Times et Yahoo Finance

1. Api New York Times

Vous devez consulter le lien suivant : https://developer.nytimes.com/apis

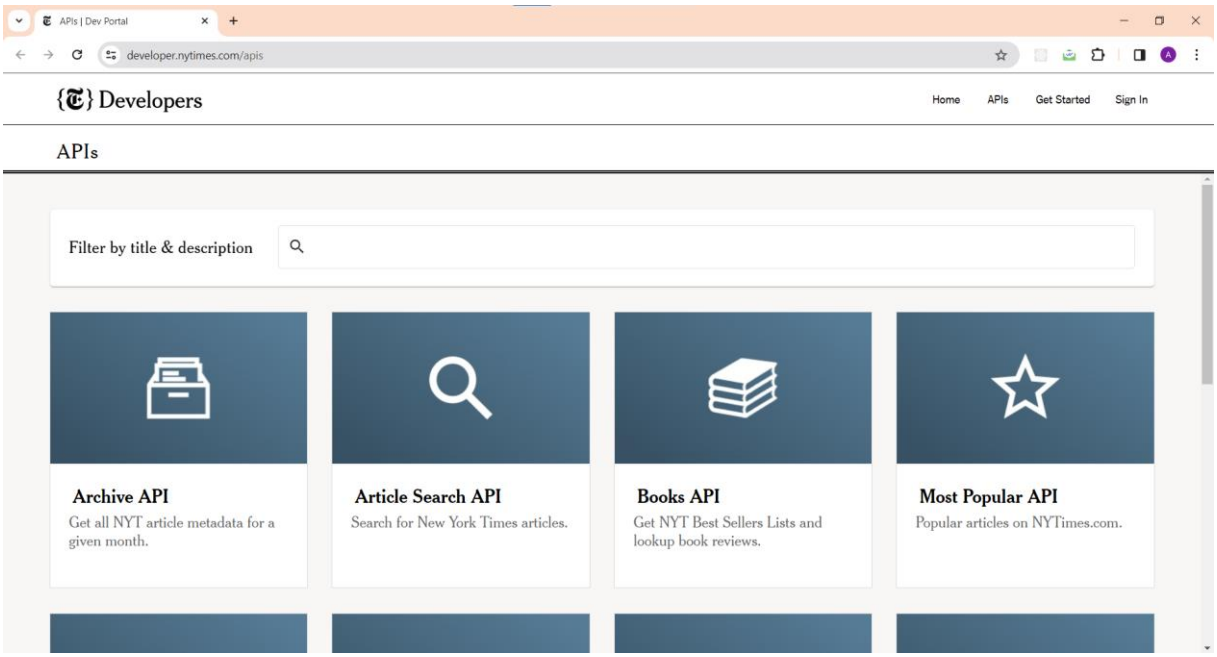
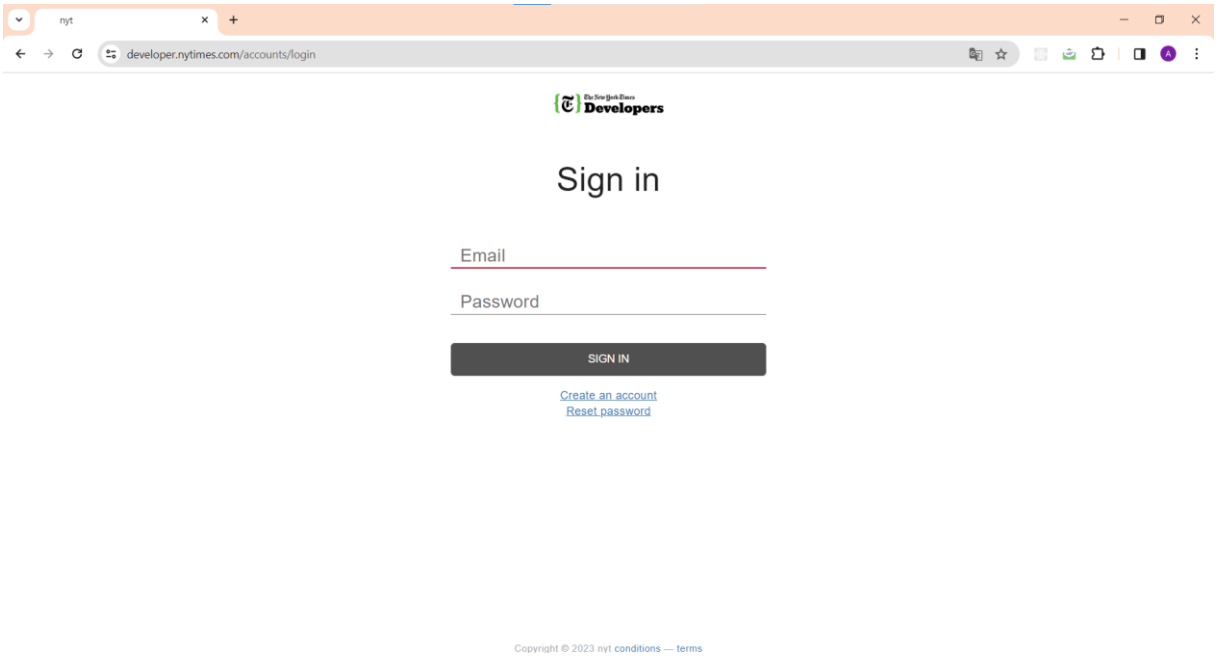
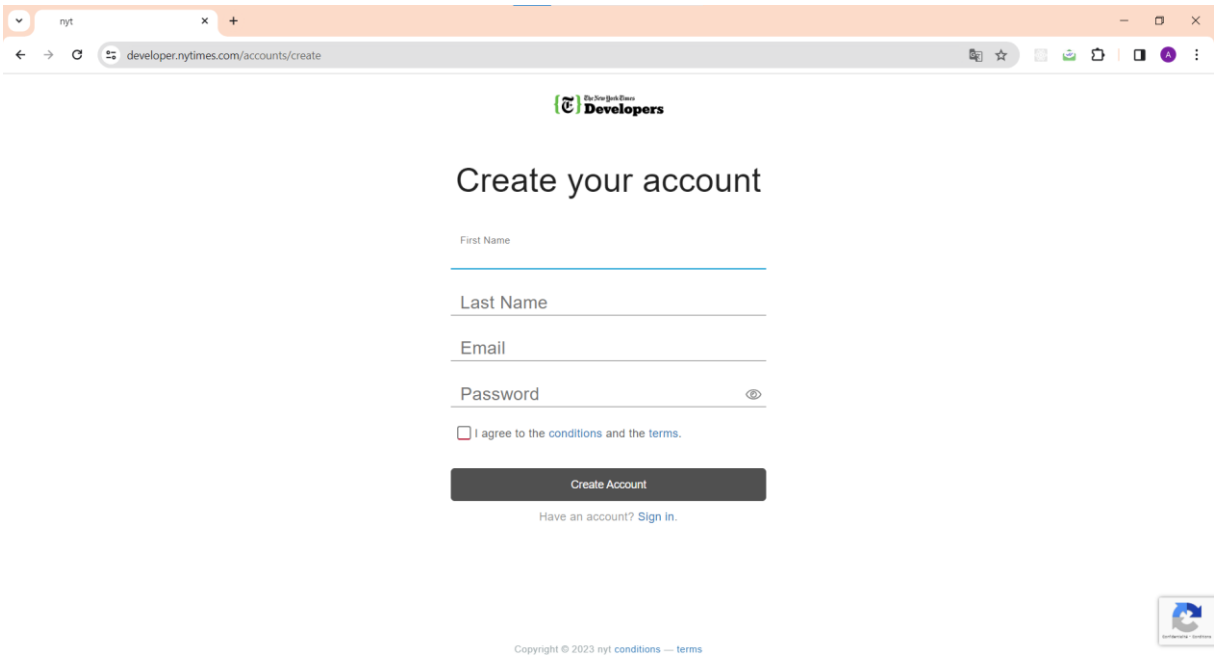


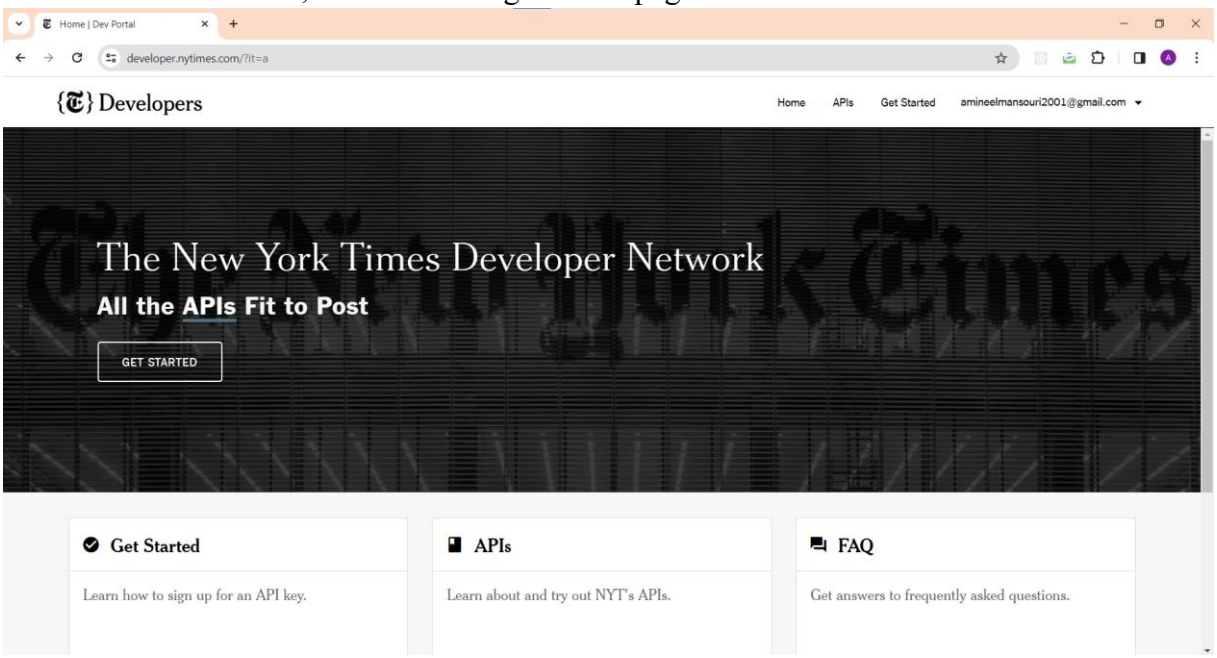
Figure 10:Interface des apis New York Times

Puis vous cliquez sur Sign In et après vous serez amené à créer un nouveau compte





Une fois vous terminez, vous serez dirigé vers la page suivante :



Vous choisissez Apis, ensuite cliquez sur Archive API

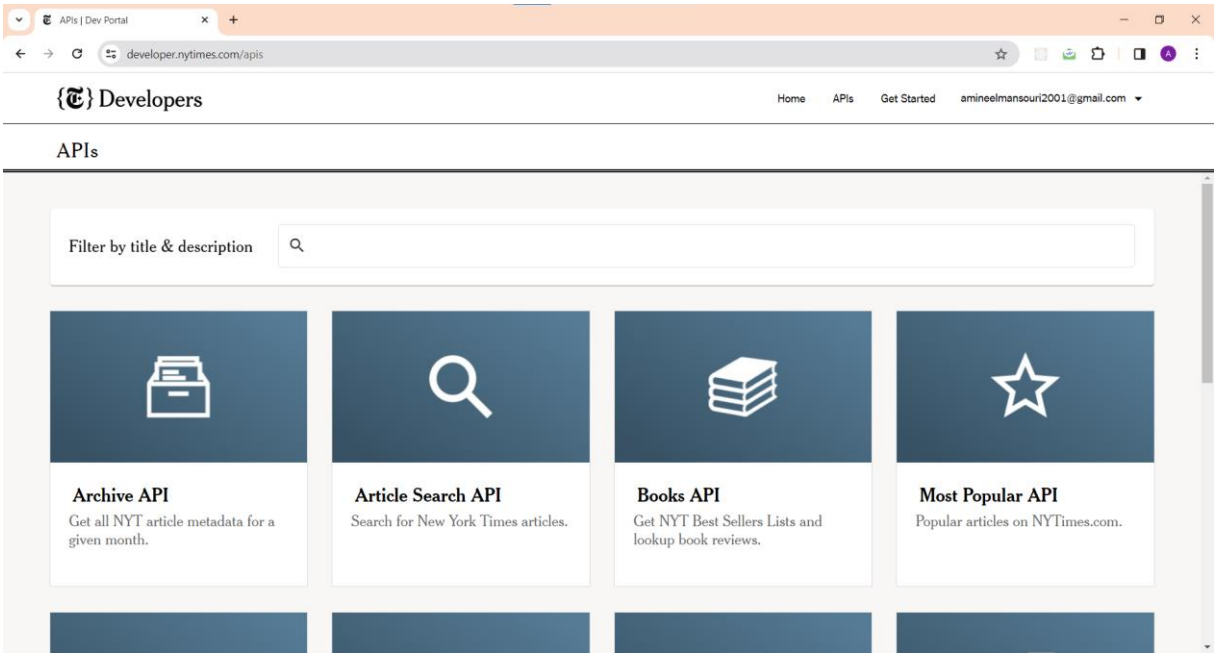


Figure 11:Apis New York Times

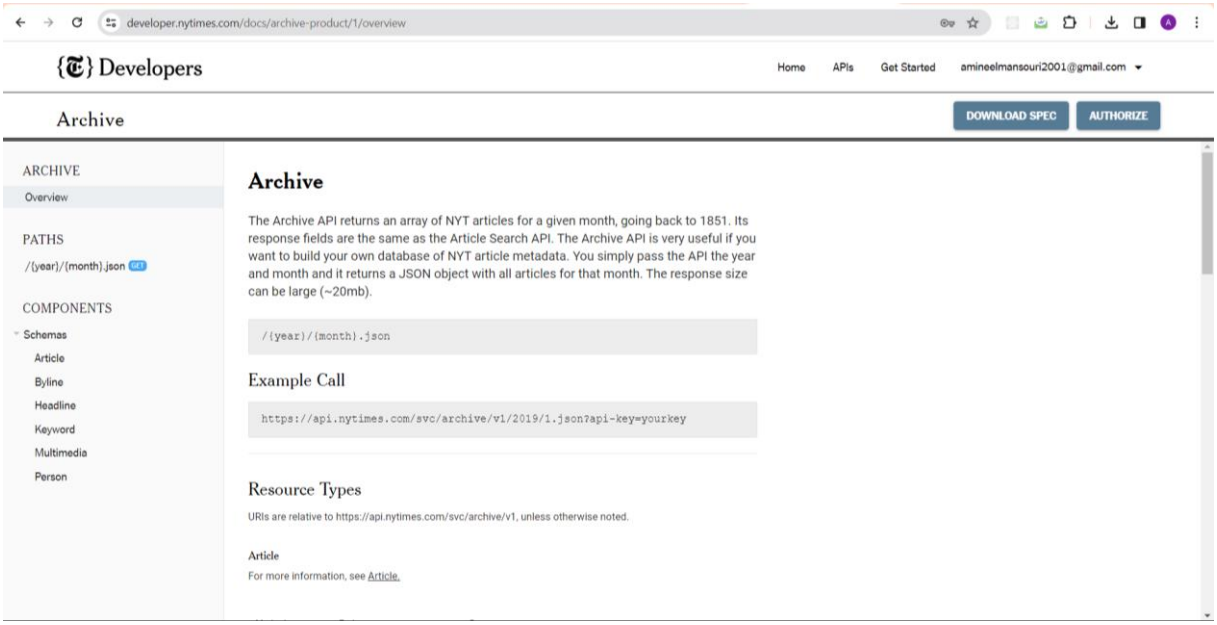
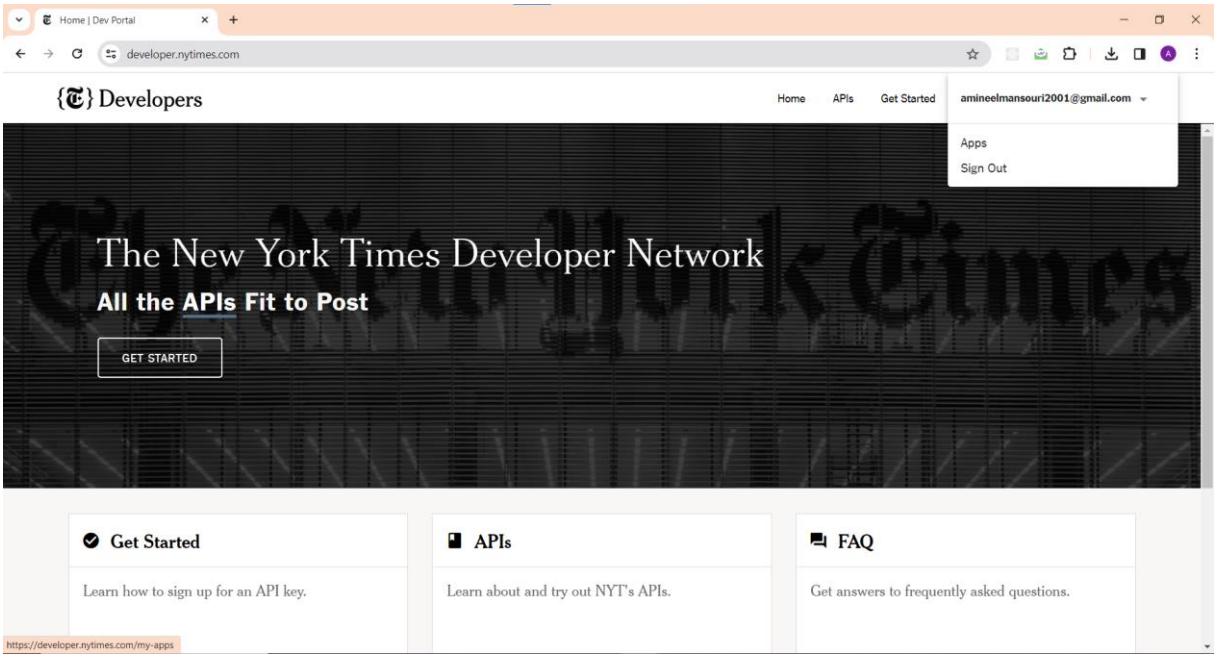
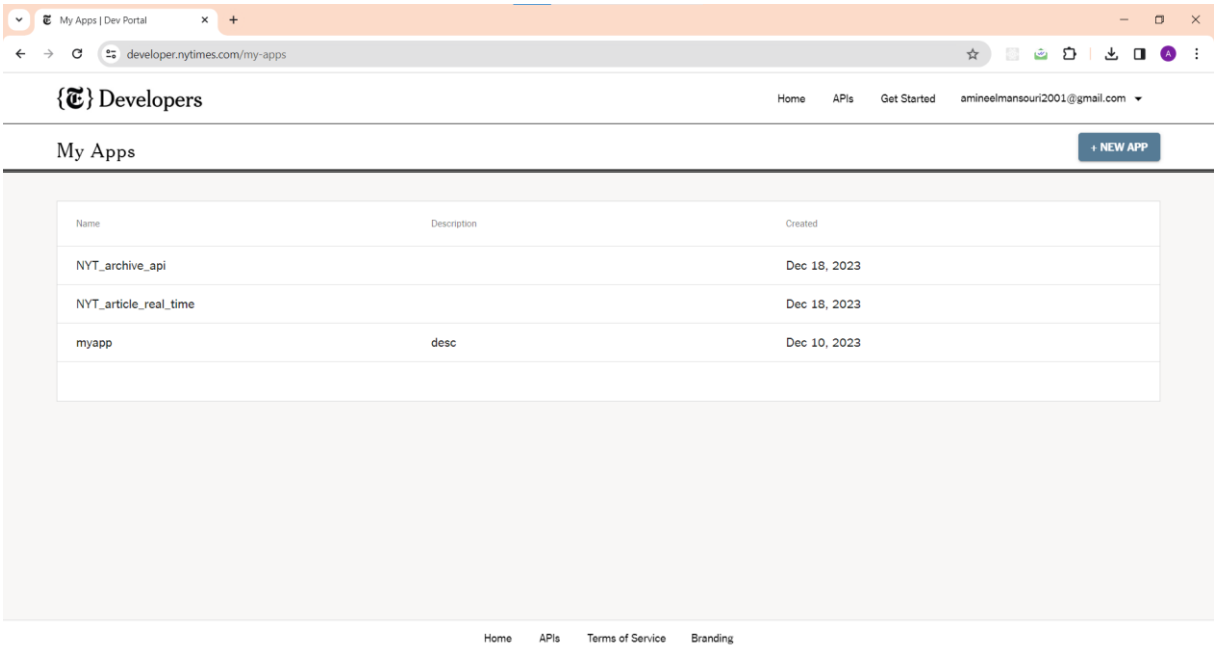


Figure 12:Archive Api pour les articles de New York Times

Vous cliquez sur votre adresse mail et vous choisissez **apps**



Cliquez sur New App



Par la suite vous ecrivez le nom de votre app ainsi qu'une description de cette app et vous cliquez sur l'action enable pour activer l'api **Archive API** Puis cliquez sur save pour enregistrer l'app.

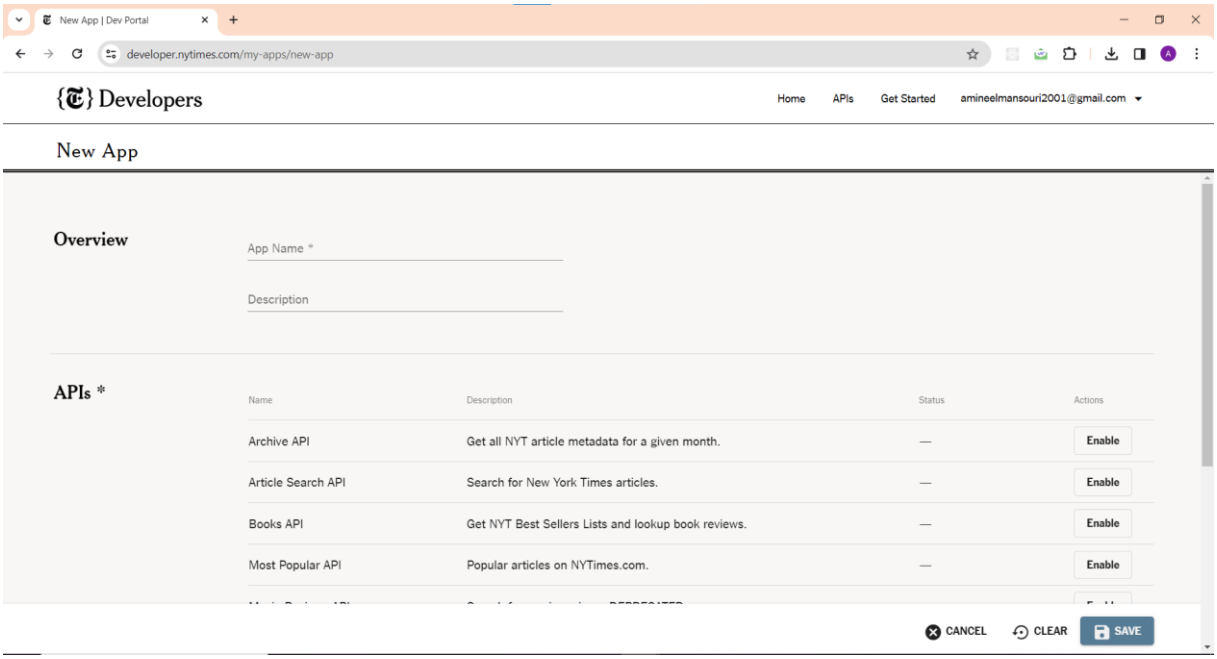
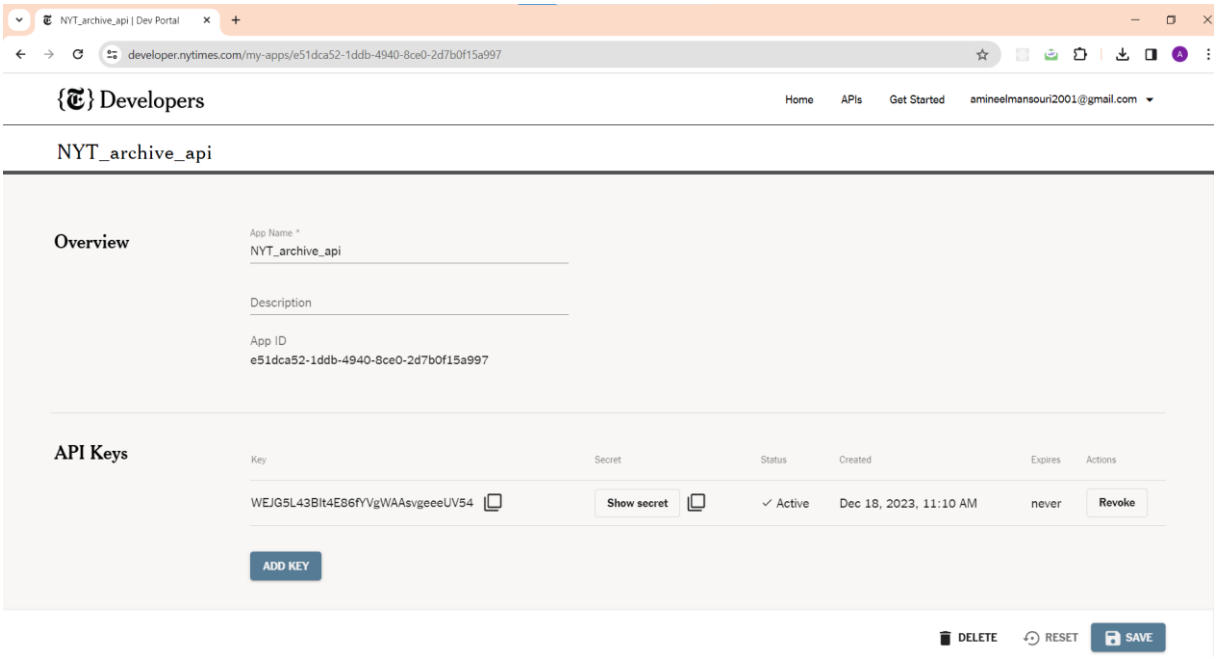


Figure 13:La création d'une nouvelle app

En fin, voilà l'api key a été générer avec succès vous pouvez l'utiliser par la suite pour intercepter les données de New Yourk Times .



2. Yahoo finance Api

Yahoo Finance propose une API qui permet d'accéder à diverses informations financières, notamment les cotations boursières, les informations sur les indices, les données historiques, etc. Pour interagir avec l'API Yahoo Finance en utilisant Python, vous pouvez utiliser des bibliothèques telles que **yfinance**.

Il suffit d'Installer la bibliothèque **yfinance,** via la commande suivante :

⇨ **pip install yfinance**

5. Interception des données des 2 Apis en utilisant Apache Beam

Ce code python utilise Apache Beam pour créer un pipeline de traitement de données. Il définit deux schémas de données, l'un pour les articles d'actualité de New York Times et l'autre pour les informations financières de Yahoo Finance, en utilisant la bibliothèque PyArrow. Le pipeline envisage de récupérer des données d'articles via des requêtes HTTP, ainsi que des données financières de Yahoo Finance, et les traiter simultanément. Les schémas spécifient la structure des données pour garantir une cohérence lors du traitement parallèle dans le pipeline Apache Beam.

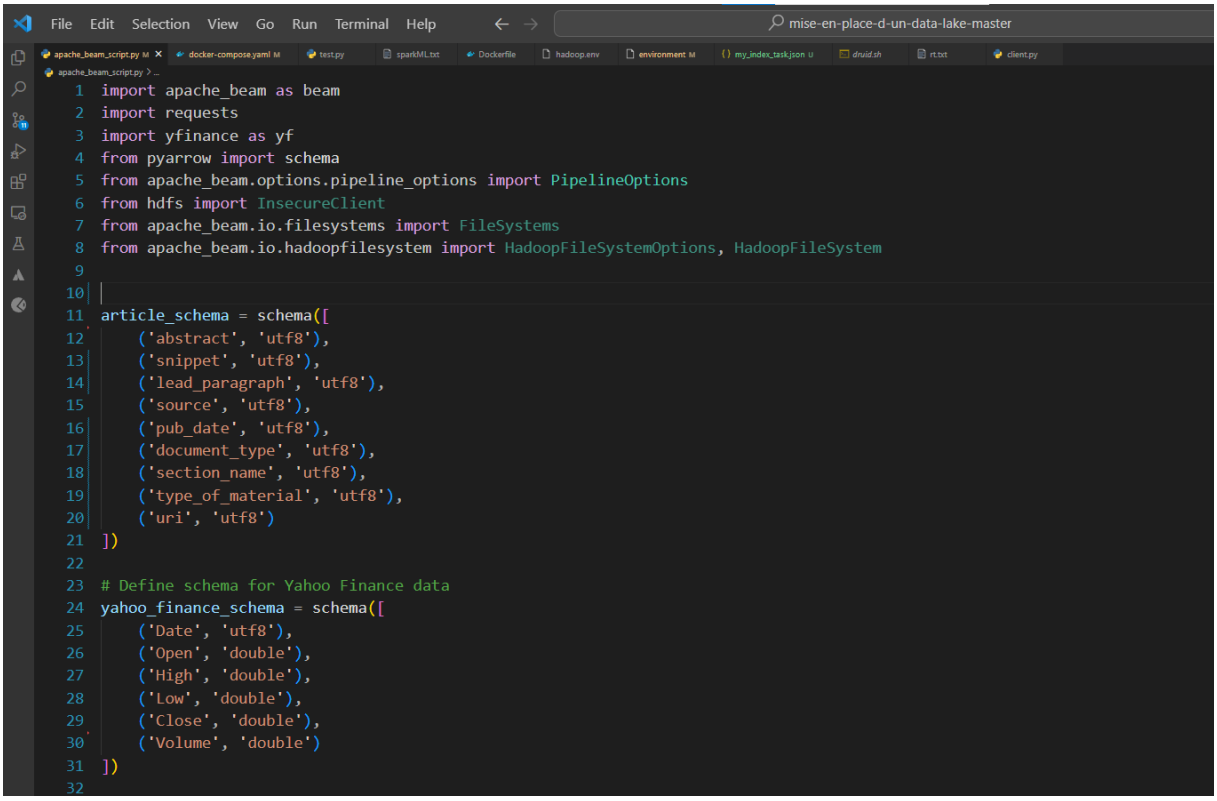


Figure 14:Interception des données-partie1

Ce code utilise Apache Beam pour créer un pipeline de traitement de données.

- Deux classes (**NYTAPIClient** et **YahooFinanceAPIClient**) interrogent respectivement l'API du New York Times et Yahoo Finance pour récupérer des données.
- La fonction **process_article** extrait des champs spécifiques des articles du New York Times.
- Le pipeline traite les données en parallèle, imprime des informations de débogage, puis écrit les résultats dans des fichiers Parquet.
- Les chemins de sortie (**output_path_nyt** et **output_path_yahoo**) et les options du pipeline sont configurés.

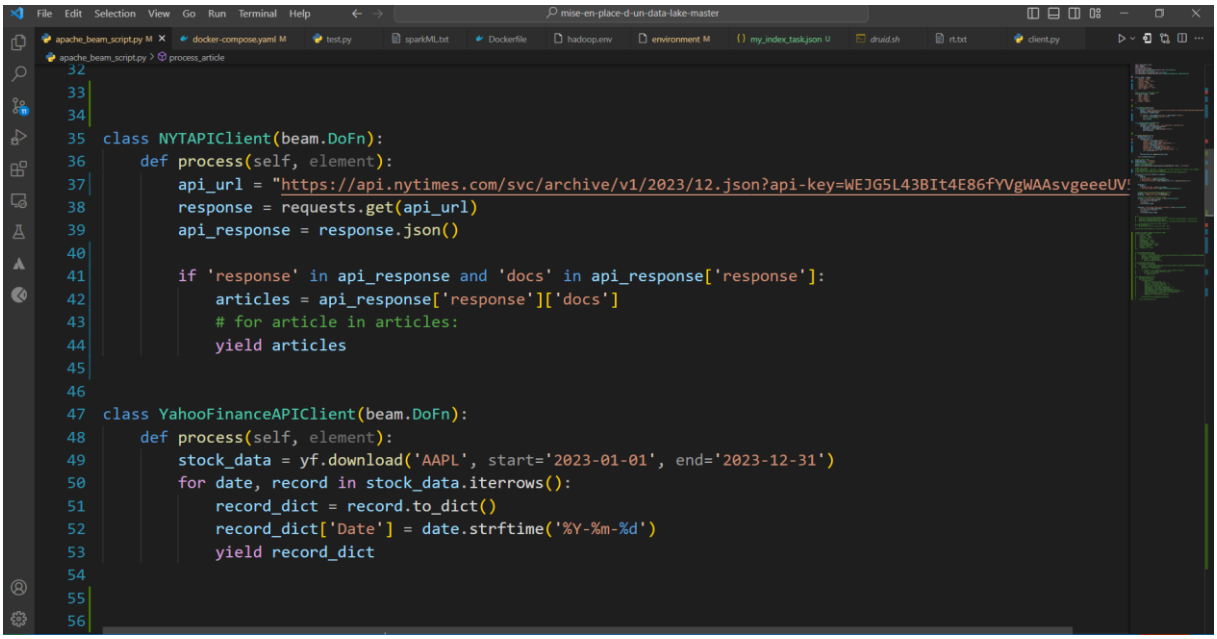
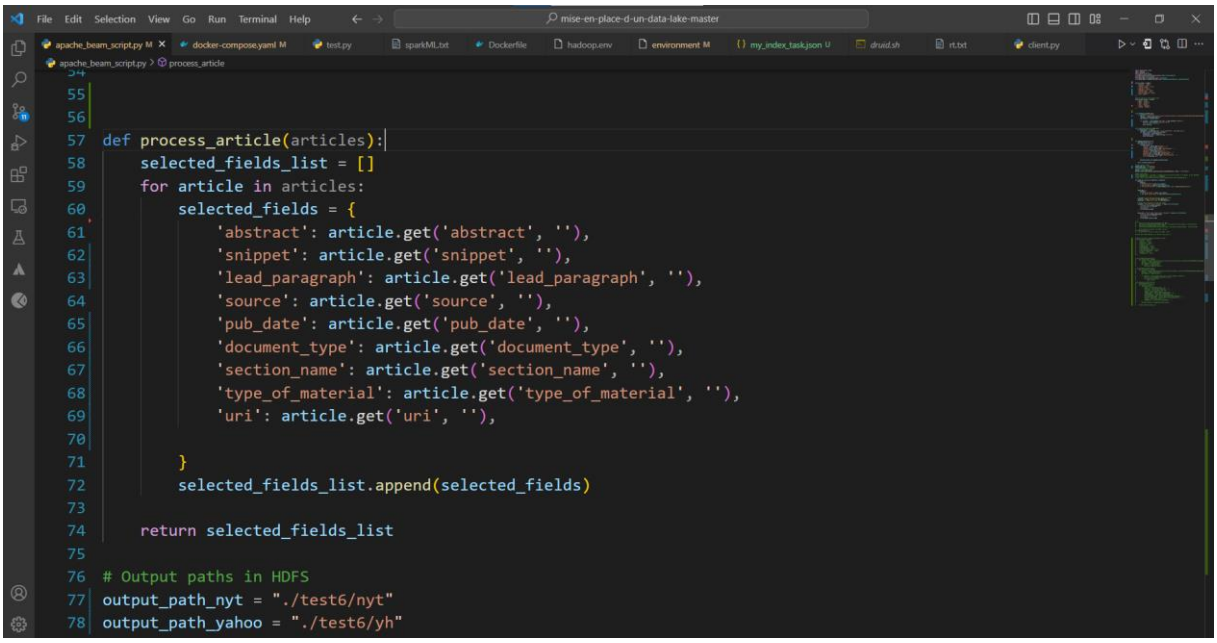
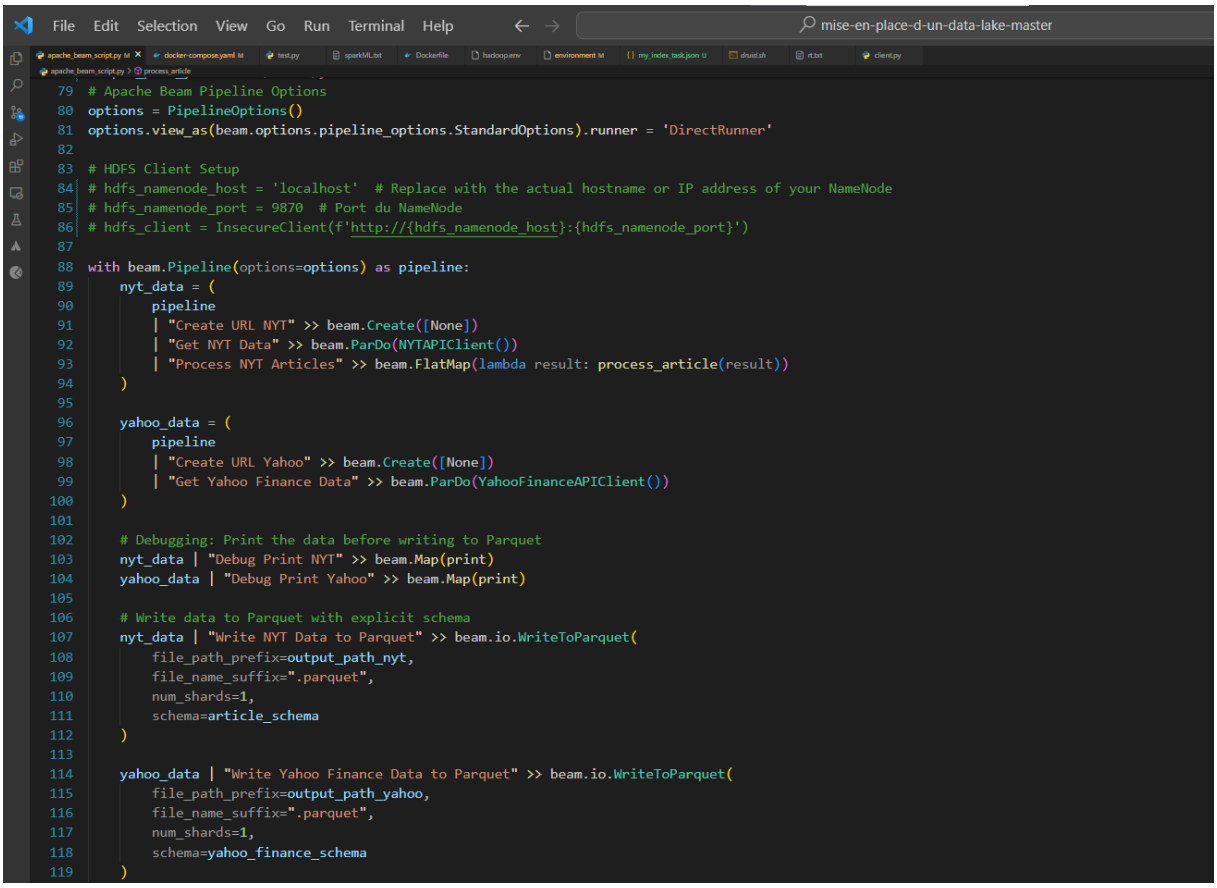


Figure 15:Interception des données-partie2



```python
def process_article(articles):
    selected_fields_list = []
    for article in articles:
        selected_fields = {
            'abstract': article.get('abstract', ''),
            'snippet': article.get('snippet', ''),
            'lead_paragraph': article.get('lead_paragraph', ''),
            'source': article.get('source', ''),
            'pub_date': article.get('pub_date', ''),
            'document_type': article.get('document_type', ''),
            'section_name': article.get('section_name', ''),
            'type_of_material': article.get('type_of_material', ''),
            'uri': article.get('uri', ''),

        }
        selected_fields_list.append(selected_fields)

    return selected_fields_list

# Output paths in HDFS
output_path_nyt = "./test6/nyt"
output_path_yahoo = "./test6/yh"
```

Figure 16:Interception des données-partie3



```python
# Apache Beam Pipeline Options
options = PipelineOptions()
options.view_as(beam.options.pipeline_options.StandardOptions).runner = 'DirectRunner'

# HDFS Client Setup
# hdfs_namenode_host = 'localhost'  # Replace with the actual hostname or IP address of your NameNode
# hdfs_namenode_port = 9870  # Port du NameNode
# hdfs_client = InsecureClient(f'http://{hdfs_namenode_host}:{hdfs_namenode_port}')

with beam.Pipeline(options=options) as pipeline:
    nyt_data = (
        pipeline
        | "Create URL NYT" >> beam.Create([None])
        | "Get NYT Data" >> beam.ParDo(NYTAPIClient())
        | "Process NYT Articles" >> beam.FlatMap(lambda result: process_article(result))
    )

    yahoo_data = (
        pipeline
        | "Create URL Yahoo" >> beam.Create([None])
        | "Get Yahoo Finance Data" >> beam.ParDo(YahooFinanceAPIClient())
    )

    # Debugging: Print the data before writing to Parquet
    nyt_data | "Debug Print NYT" >> beam.Map(print)
    yahoo_data | "Debug Print Yahoo" >> beam.Map(print)

    # Write data to Parquet with explicit schema
    nyt_data | "Write NYT Data to Parquet" >> beam.io.WriteToParquet(
        file_path_prefix=output_path_nyt,
        file_name_suffix=".parquet",
        num_shards=1,
        schema=article_schema
    )

    yahoo_data | "Write Yahoo Finance Data to Parquet" >> beam.io.WriteToParquet(
        file_path_prefix=output_path_yahoo,
        file_name_suffix=".parquet",
        num_shards=1,
        schema=yahoo_finance_schema
    )
```

Figure 17:Interception des données-partie4

6. Enregistrement des données brutes sous format Parquet

Les résultats de l'exécution du pipeline Beam sont écrits dans des fichiers Parquet avec des schémas explicites pour les données du New York Times et de Yahoo Finance.

Figure 18:Les fichiers parquet générés

7. Entraînement de plusieurs modèle ML pour sentiment analysis

Voici une description des étapes que vous pourriez suivre pour entraîner plusieurs modèles de sentiment analysis en utilisant Spark ML sur un notebook Zeppelin et enregistrer le meilleur modèle :

1. **Importation des bibliothèques Spark ML :**
    - Débutez par importer les bibliothèques Spark ML nécessaires.
2. **Définition des modèles et de leurs hyperparamètres :**
    - Définissez une liste de modèles de classification tels que Logistic Regression, Naive Bayes et RandomForest, avec leurs hyperparamètres respectifs.
3. **Division du dataset en ensembles d'entraînement et de test :**
    - Divisez le dataset en ensembles d'entraînement et de test en utilisant la méthode **randomSplit**.

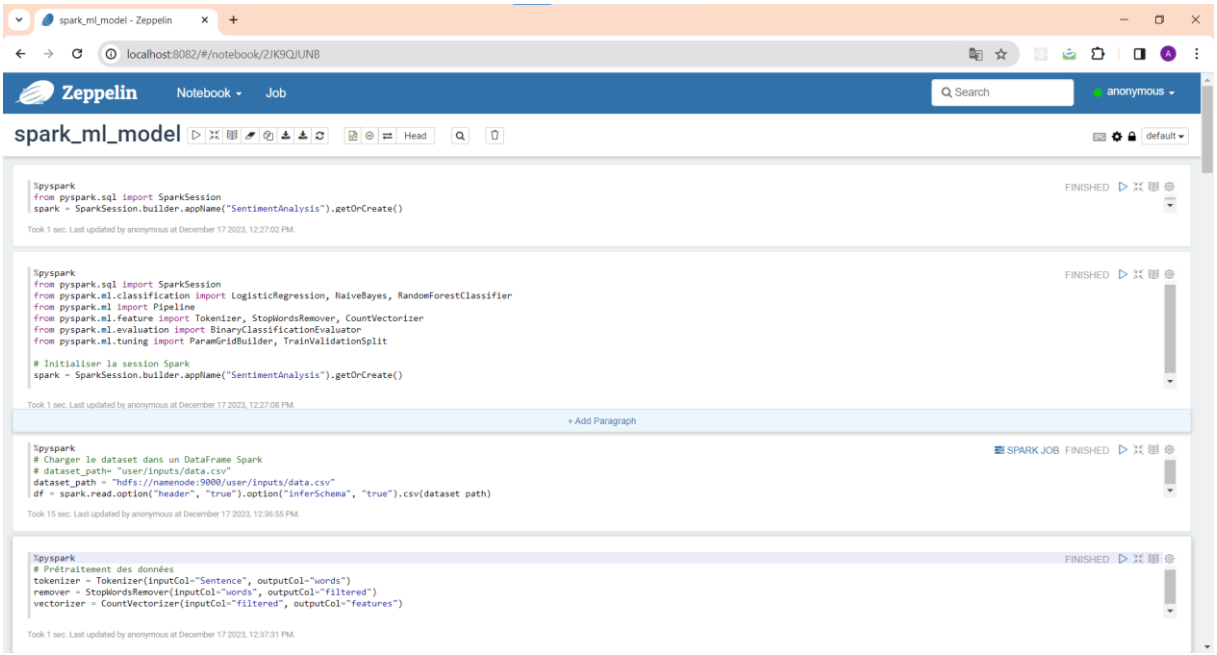


Figure 19:code d'entrainement de plusieurs Modèles ML pour sentiment analysis-partie1

4. **Prétraitement des données avec un pipeline :**

   - Pour chaque modèle, créez un pipeline comprenant plusieurs étapes : Tokenization des phrases, HashingTF pour la conversion des mots en vecteurs, IDF pour pondérer les termes, StringIndexer pour indexer la variable cible "Sentiment", et l'entraînement du modèle spécifique (Logistic Regression, Naive Bayes, RandomForest).

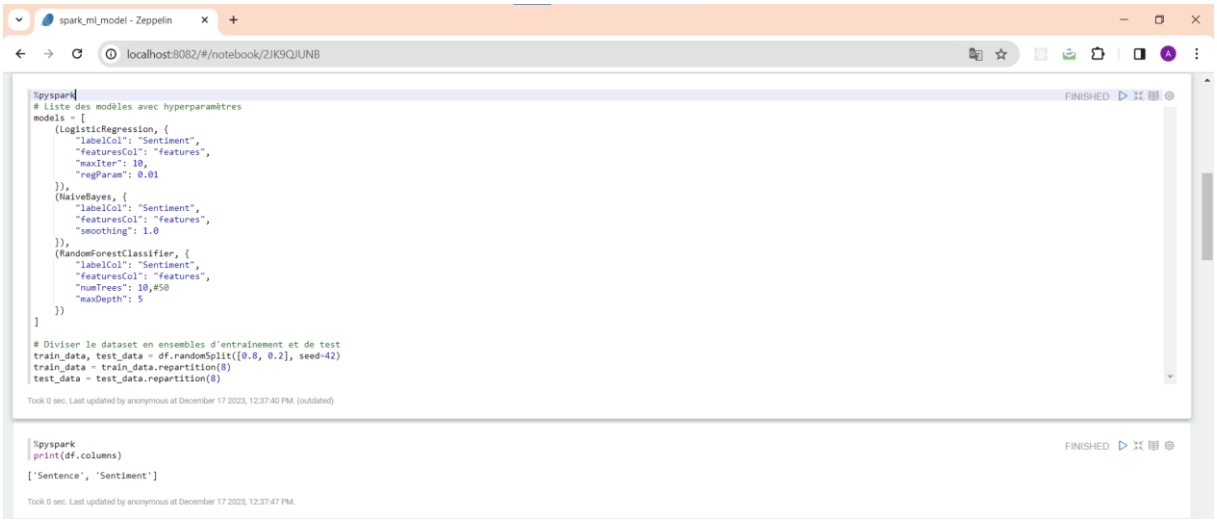


Figure 20:code d'entrainement de plusieurs Modèles ML pour sentiment analysis-partie2

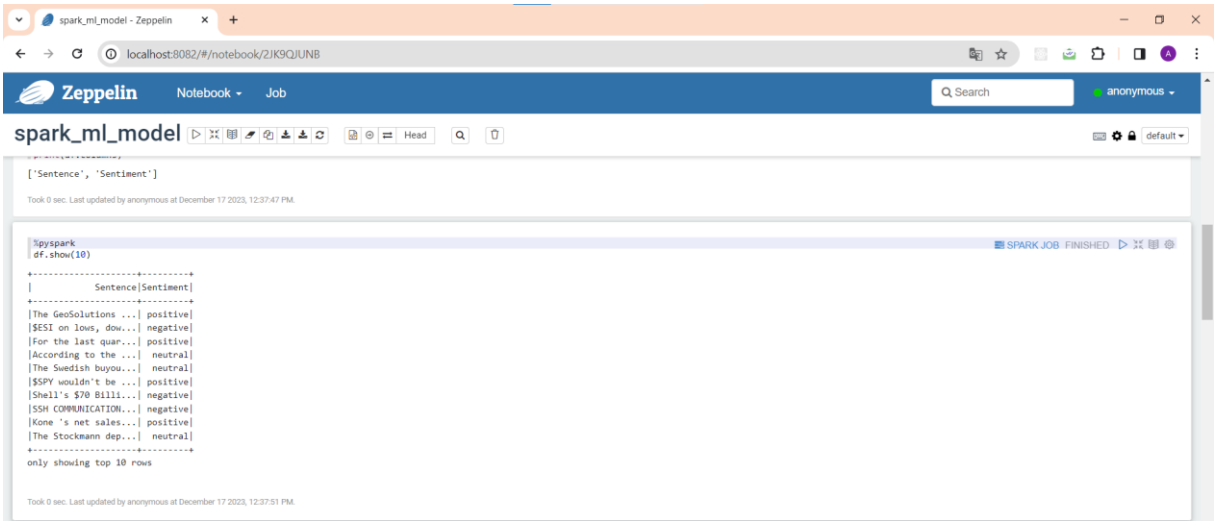


Figure 21:code d'entrainement de plusieurs Modèles ML pour sentiment analysis-partie3

5. **Entraînement et évaluation des modèles :**

   - Utilisez un échantillon de 10% des données d'entraînement pour accélérer le processus. Entraînez et évaluez les modèles sur l'ensemble de test, puis imprimez l'exactitude de chaque modèle.

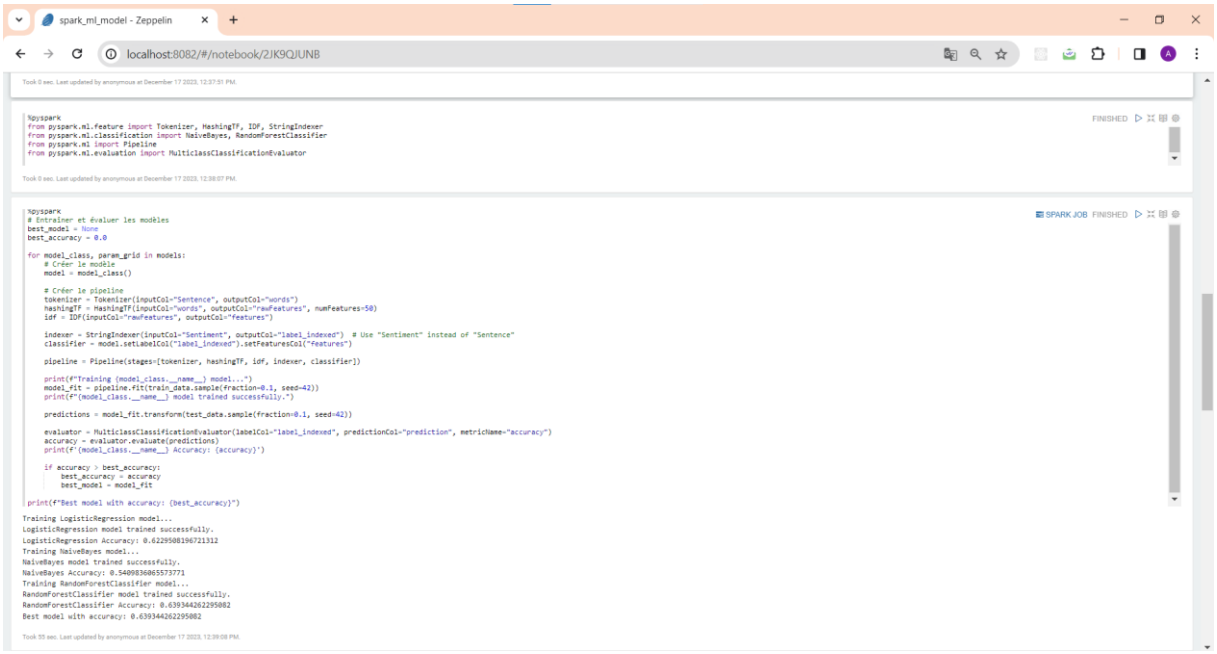


Figure 22:Entraînement et évaluation des modèles

6. **Sélection du meilleur modèle :**

   - Suivez une approche de sélection du meilleur modèle en comparant les exactitudes et enregistrez le modèle avec la meilleure précision.

7. **Enregistrement du meilleur modèle sur HDFS :**
    - Enregistrez le meilleur modèle sur le système de fichiers distribué HDFS en utilisant le chemin spécifié.

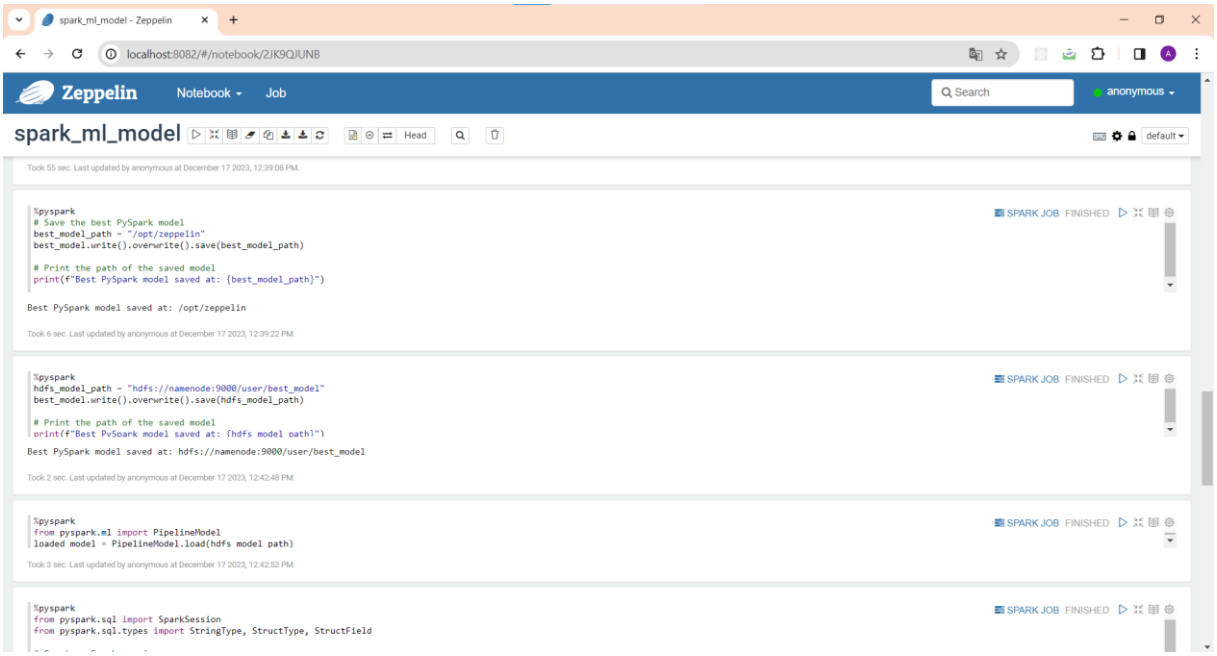


Figure 23:Enregistrement du meilleur modèle sur HDFS

8. **Chargement du modèle enregistré :**
    - Chargez le modèle enregistré à partir de HDFS pour une utilisation ultérieure.
9. **Utilisation du modèle pour faire des prédictions sur de nouvelles données :**
    - Créez un Spark DataFrame avec de nouvelles données de texte et utilisez le modèle chargé pour faire des prédictions sur ces nouvelles données.

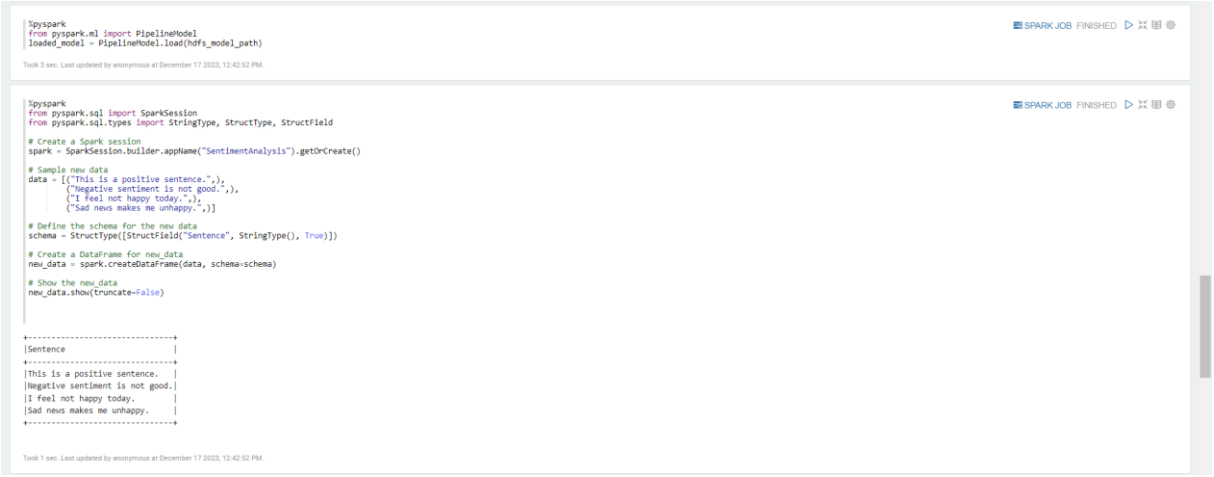


Figure 24:Utilisation du modèle pour faire des prédictions sur de nouvelles données

10. **Affichage des résultats des prédictions :**

- Définissez une fonction pour mapper les prédictions numériques à des libellés de sentiment (positif, négatif, neutre). Appliquez cette fonction aux prédictions et sélectionnez les colonnes pertinentes pour l'affichage des résultats.

Figure 25:Affichage des résultats des prédictions

8. La lecture des données enregistrées avec Spark SQL

Les étapes que vous devez suivre en utilisant PySpark pour lire et traiter des données à partir de fichiers Parquet :

- **Initialisez la session Spark**
- **Lisez les fichiers Parquet :**
  - ✓ Définissez les chemins des fichiers Parquet du New York Times (nyt_path) et de Yahoo Finance (yahoo_path).
  - ✓ Utilisez la méthode spark.read.parquet() pour lire ces fichiers et créer les DataFrames correspondants (nyt_df et yahoo_df).

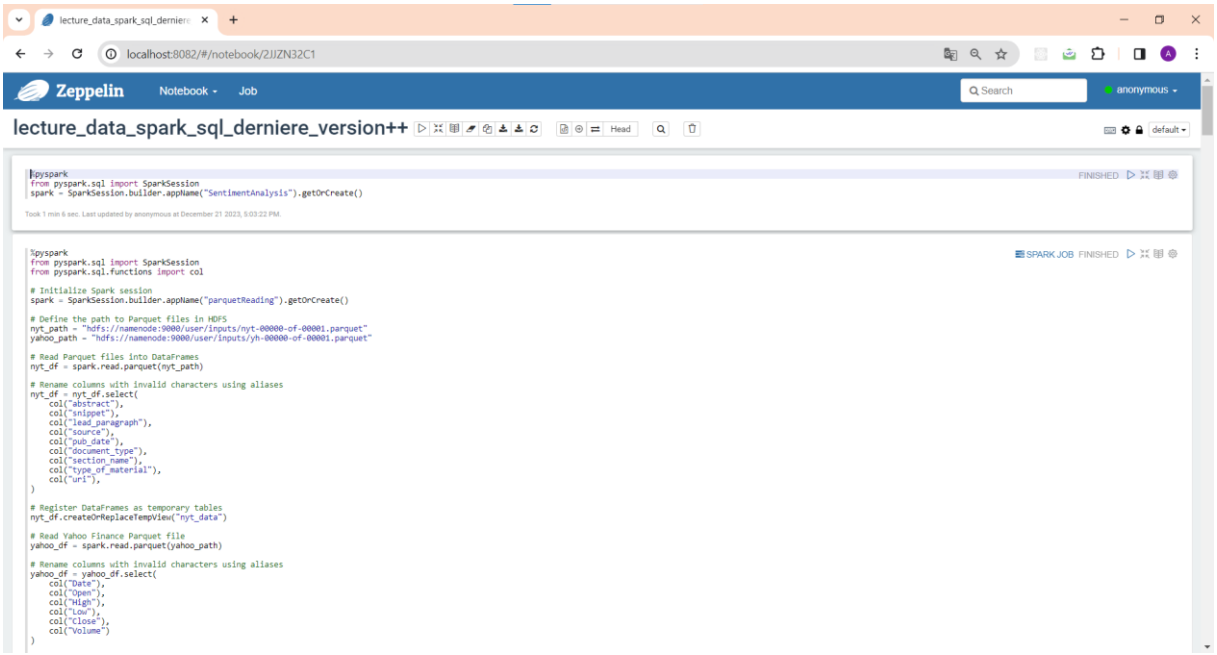


Figure 26:La lecture des données enregistrées avec Spark SQL

- **Enregistrez les DataFrames en tant que tables temporaires :**
  - ✓ Utilisez la méthode **createOrReplaceTempView** pour enregistrer les DataFrames en tant que tables temporaires dans Spark, permettant l'exécution de requêtes SQL.
- **Exécutez des requêtes SQL Spark :**
  - ✓ Utilisez la fonction **spark.sql()** pour exécuter des requêtes SQL sur les tables temporaires. Par exemple, sélectionnez toutes les colonnes des tables **nyt_data** et **yahoo_data**.
- **Affichez les résultats :**
  - ✓ Utilisez la méthode **show()** pour afficher les résultats des requêtes SQL.

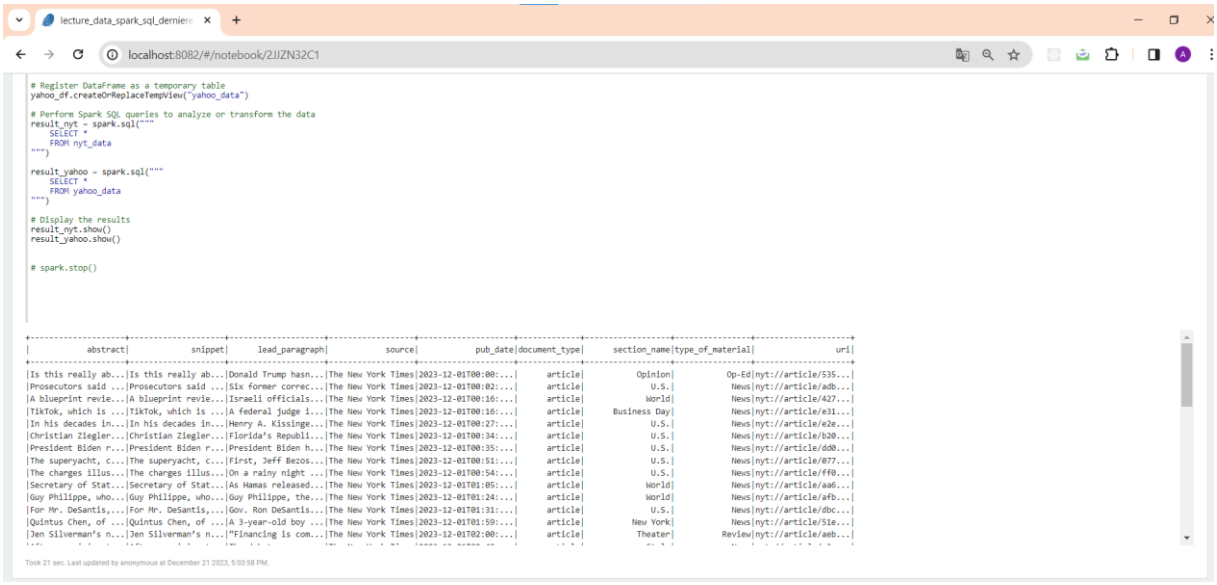
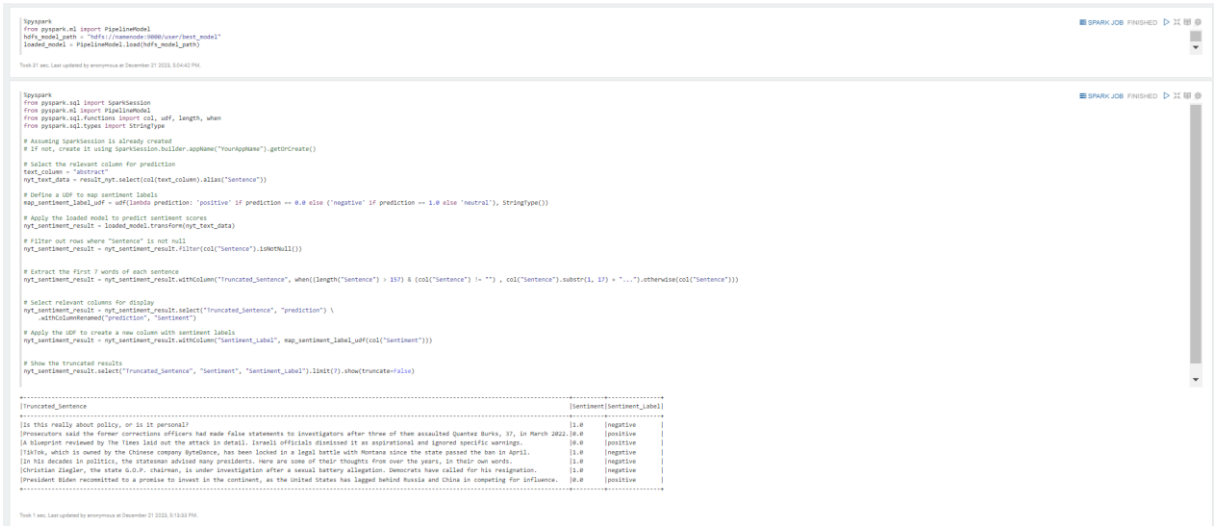


Figure 27:L'affichage des résultats des requêtes SQL-partie1

Figure 28:L'affichage des résultats des requêtes SQL-partie2

9. Prédiction du score de sentiment des textes en utilisant Spark ML

Dans ce script PySpark, on utilise un modèle préalablement entraîné pour prédire les étiquettes de sentiment à partir d'une colonne de texte dans un DataFrame du New York Times. Les résultats sont ensuite filtrés et affichés, montrant les 7 premières lignes avec une colonne de texte tronquée, les scores de sentiment prédits, et les étiquettes associées ("positive", "negative", "neutral").

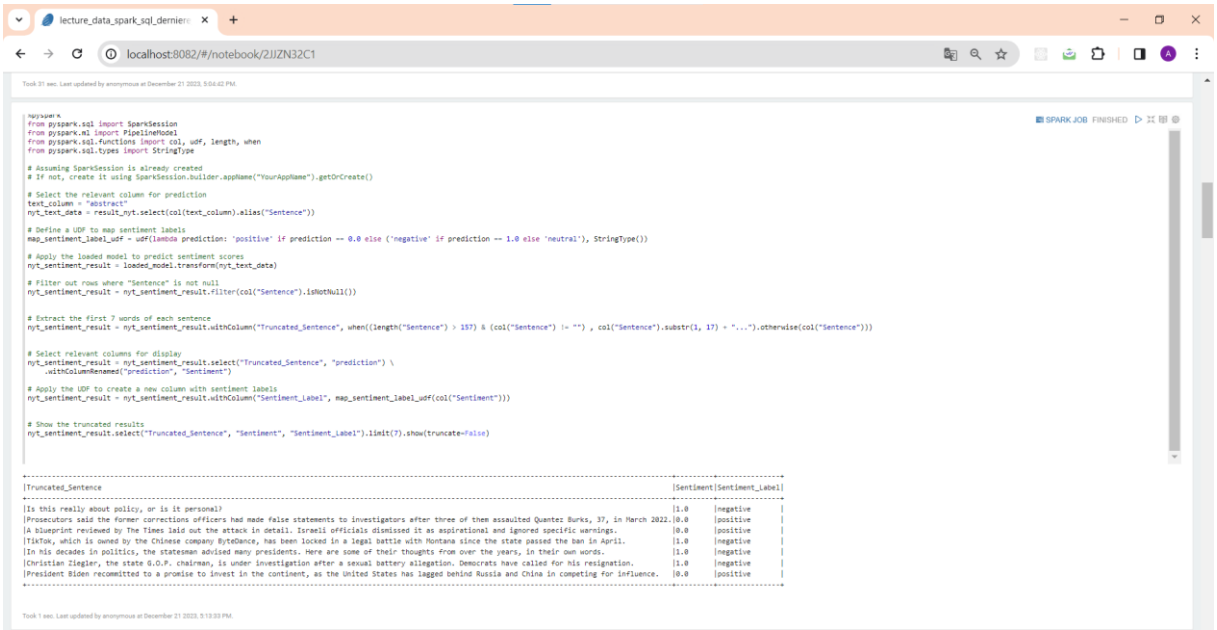


```
%pyspark
from pyspark.ml import PipelineModel
hdfs_model_path = "hdfs://namenode:9000/user/best_model"
loaded_model = PipelineModel.load(hdfs_model_path)
```

```
%pyspark
from pyspark.sql import SparkSession
from pyspark.ml import PipelineModel
from pyspark.sql.functions import col, udf, length, when
from pyspark.sql.types import StringType

# Assuming SparkSession is already created
# If not, create it using SparkSession.builder.appName("YourAppName").getOrCreate()

# Select the relevant column for prediction
text_column = "abstract"
nyt_text_data = result_nyt.select(col(text_column).alias("Sentence"))

# Define a UDF to map sentiment labels
map_sentiment_label_udf = udf(lambda prediction: 'positive' if prediction == 0.0 else ('negative' if prediction == 1.0 else 'neutral'), StringType())

# Apply the loaded model to predict sentiment scores
nyt_sentiment_result = loaded_model.transform(nyt_text_data)

# Filter out rows where "Sentence" is not null
nyt_sentiment_result = nyt_sentiment_result.filter(col("Sentence").isNotNull())


# Extract the first 7 words of each sentence
nyt_sentiment_result = nyt_sentiment_result.withColumn("Truncated_Sentence", when((length("Sentence") > 157) & (col("Sentence") != "") , col("Sentence").substr(1, 17) + "...").otherwise(col("Sentence")))


# Select relevant columns for display
nyt_sentiment_result = nyt_sentiment_result.select("Truncated_Sentence", "prediction") \
    .withColumnRenamed("prediction", "Sentiment")

# Apply the UDF to create a new column with sentiment labels
nyt_sentiment_result = nyt_sentiment_result.withColumn("Sentiment_Label", map_sentiment_label_udf(col("Sentiment")))


# Show the truncated results
nyt_sentiment_result.select("Truncated_Sentence", "Sentiment", "Sentiment_Label").limit(7).show(truncate=False)
```

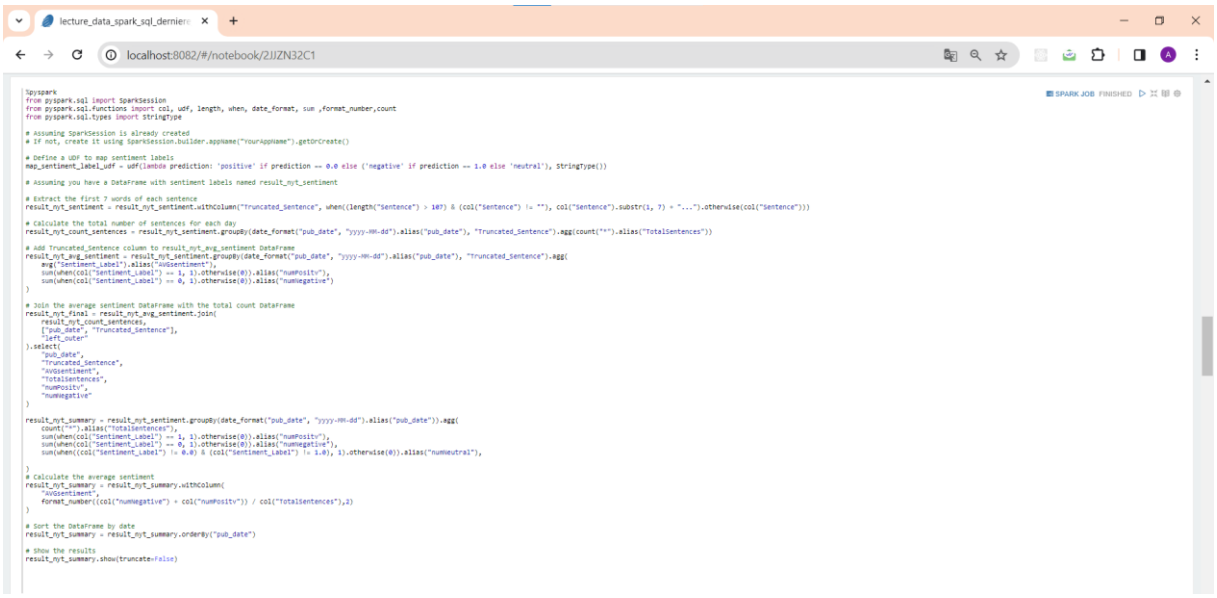
```
+-----------------------------------------------------------------------------------------------------------------------+---------+---------------+
|Truncated_Sentence                                                                                                     |Sentiment|Sentiment_Label|
+-----------------------------------------------------------------------------------------------------------------------+---------+---------------+
|Is this really about policy, or is it personal?                                                                        |1.0      |negative       |
|Prosecutors said the former corrections officers had made false statements to investigators after three of them assaulted Quantez Burks, 37, in March 2022.|0.0      |positive       |
|A blueprint reviewed by The Times laid out the attack in detail. Israeli officials dismissed it as aspirational and ignored specific warnings.        |0.0      |positive       |
|TikTok, which is owned by the Chinese company ByteDance, has been locked in a legal battle with Montana since the state passed the ban in April.      |1.0      |negative       |
|In his decades in politics, the statesman advised many presidents. Here are some of their thoughts from over the years, in their own words.          |1.0      |negative       |
|Christian Ziegler, the state G.O.P. chairman, is under investigation after a sexual battery allegation. Democrats have called for his resignation.    |1.0      |negative       |
|President Biden recommitted to a promise to invest in the continent, as the United States has lagged behind Russia and China in competing for influence.|0.0      |positive       |
+-----------------------------------------------------------------------------------------------------------------------+---------+---------------+
```

Figure 29:Prédiction du score de sentiment des textes en utilisant Spark ML



10. Ajout de nouvelles colonnes dans les données NYT

On analyse les données de sentiments du New York Times en tronquant les abstracts, calculant le nombre total de abstracts et la moyenne des sentiments par jour, fournissant un résumé global des statistiques par jour, et affichant les résultats triés par date.




```
+----------+--------------+---------+-----------+----------+------------+
|pub_date  |TotalSentences|numPositv|numNegative|numNeutral|AVGsentiment|
+----------+--------------+---------+-----------+----------+------------+
|2023-12-01|194           |85       |4          |105       |0.46        |
|2023-12-02|90            |44       |0          |46        |0.49        |
|2023-12-03|64            |30       |1          |33        |0.48        |
|2023-12-04|102           |40       |1          |61        |0.40        |
|2023-12-05|170           |80       |0          |90        |0.47        |
|2023-12-06|163           |72       |1          |90        |0.45        |
|2023-12-07|187           |88       |5          |94        |0.50        |
|2023-12-08|161           |70       |2          |89        |0.45        |
|2023-12-09|82            |39       |0          |43        |0.48        |
|2023-12-10|70            |35       |2          |33        |0.53        |
|2023-12-11|129           |59       |4          |66        |0.49        |
|2023-12-12|154           |73       |1          |80        |0.48        |
|2023-12-13|139           |66       |4          |69        |0.50        |
|2023-12-14|158           |79       |2          |77        |0.51        |
```

Took 2 sec. Last updated by anonymous at December 21 2023, 5:14:15 PM.

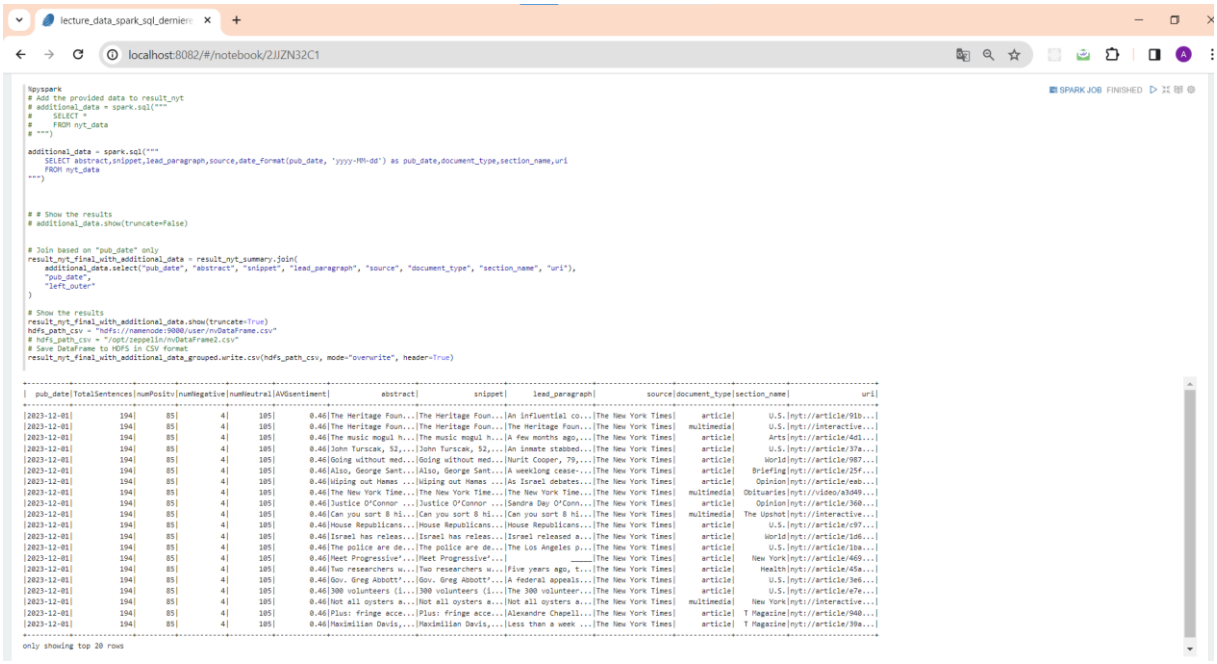


Figure 30:L'ajout de nouvelles colonnes dans les données NYT-partie1

On ajoute les colonnes suivantes dans les données New York Times :

- Une colonne nommée AVGsentiment dans laquelle vous mettez la moyenne des sentiments des textes publiés dans cette journée
- On ajoute une autre colonne appelé numPositv dans laquelle vous mettez le nombre de textes positives et une autre appelée numNegative pour les textes négatifs

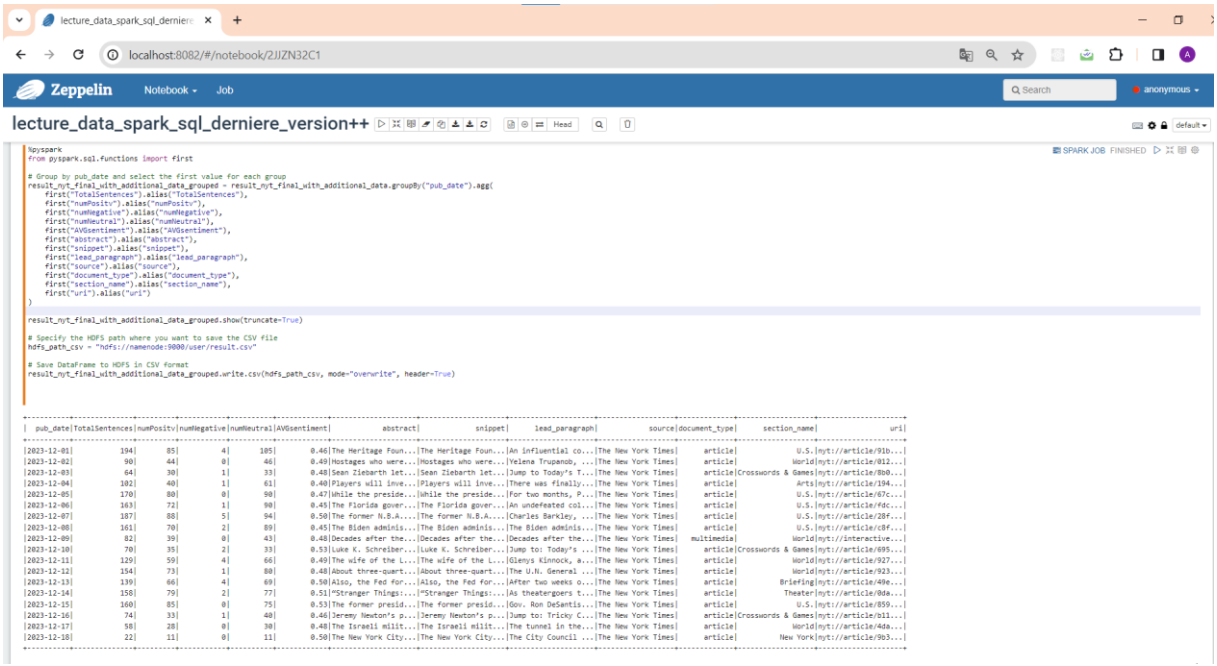
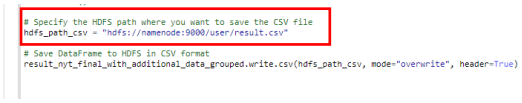


Figure 31:L'ajout de nouvelles colonnes dans les données NYT-partie2

11. Enregistrement du nouveau DataFrame obtenu dans apache Druid



Voilà, les étapes à suivre pour enregistrer le nouveau dataframe dans une nouvelle datastore Druid :

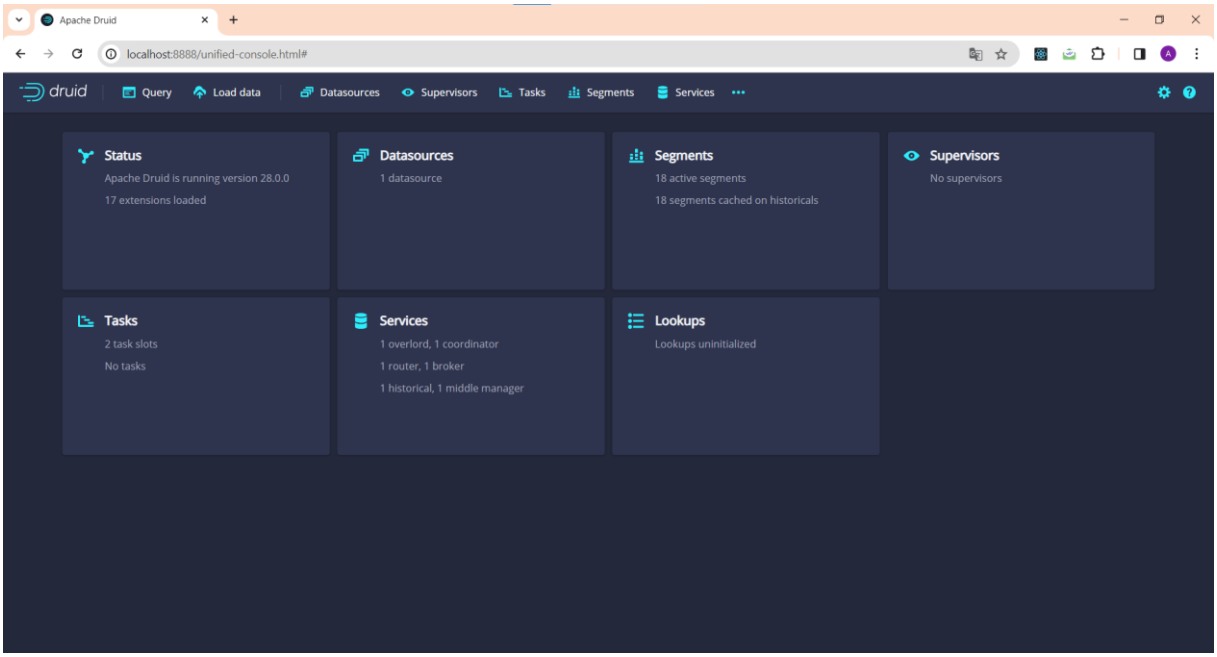


Figure 32:L'interface utilisateur de Apache Druid

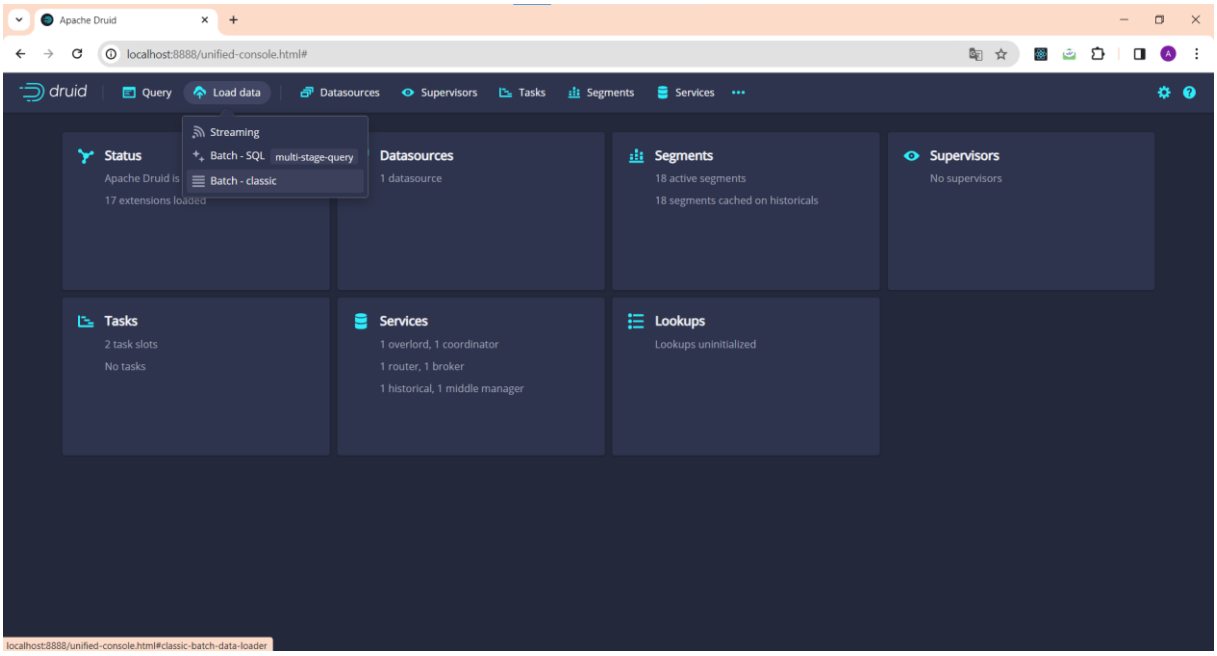


Figure 33:Le choix de Batch-classic

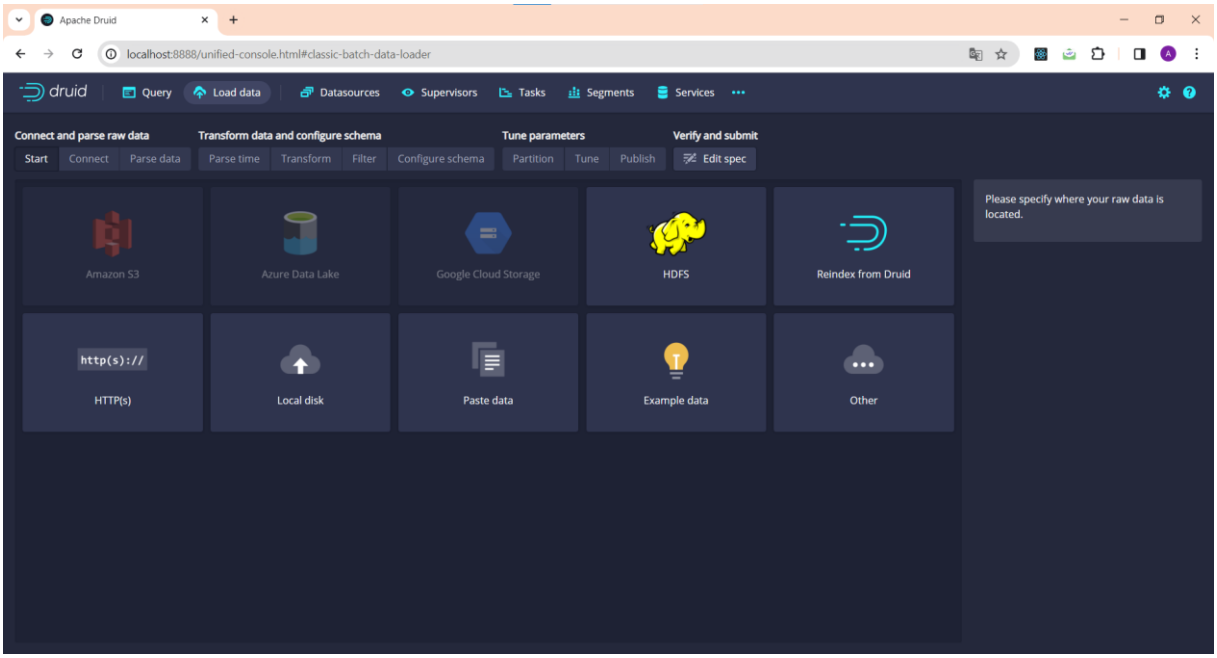


Figure 34:cliquer sur HDFS

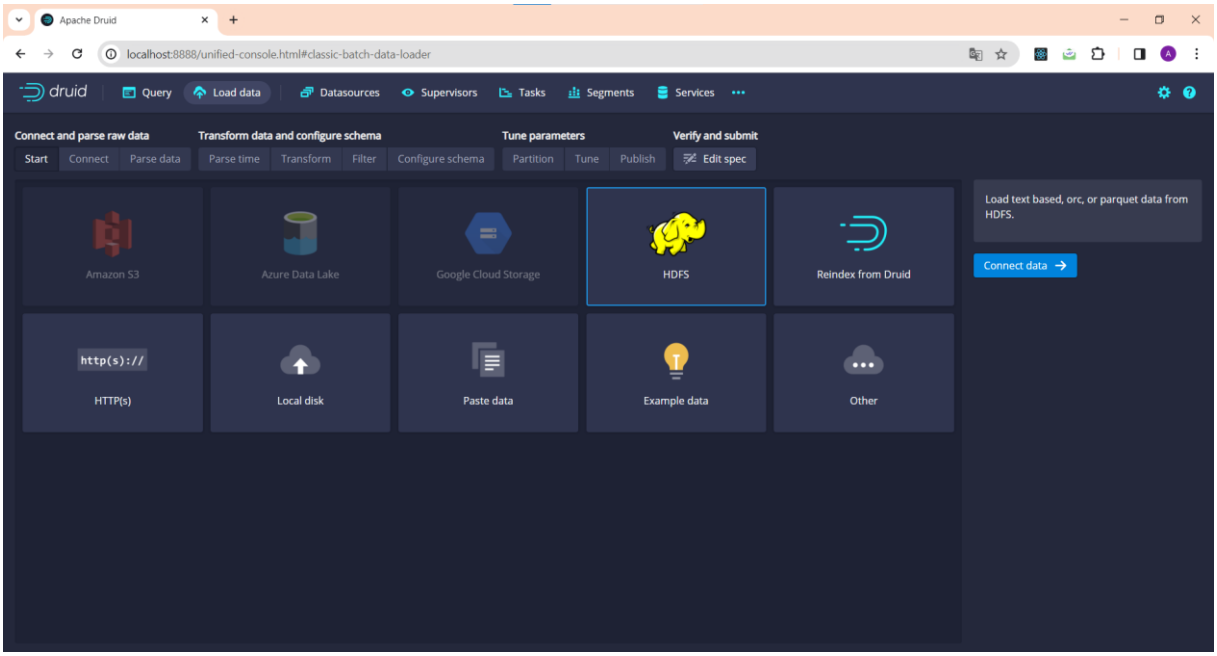


Figure 35:Cliquer sur Connect data

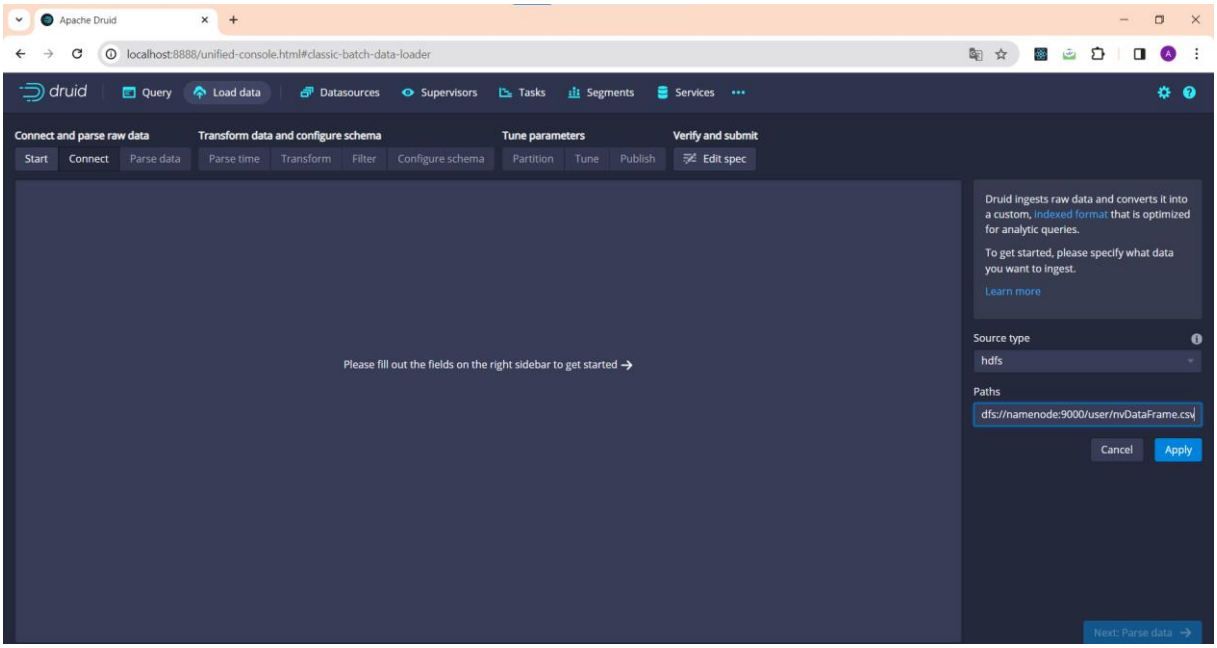
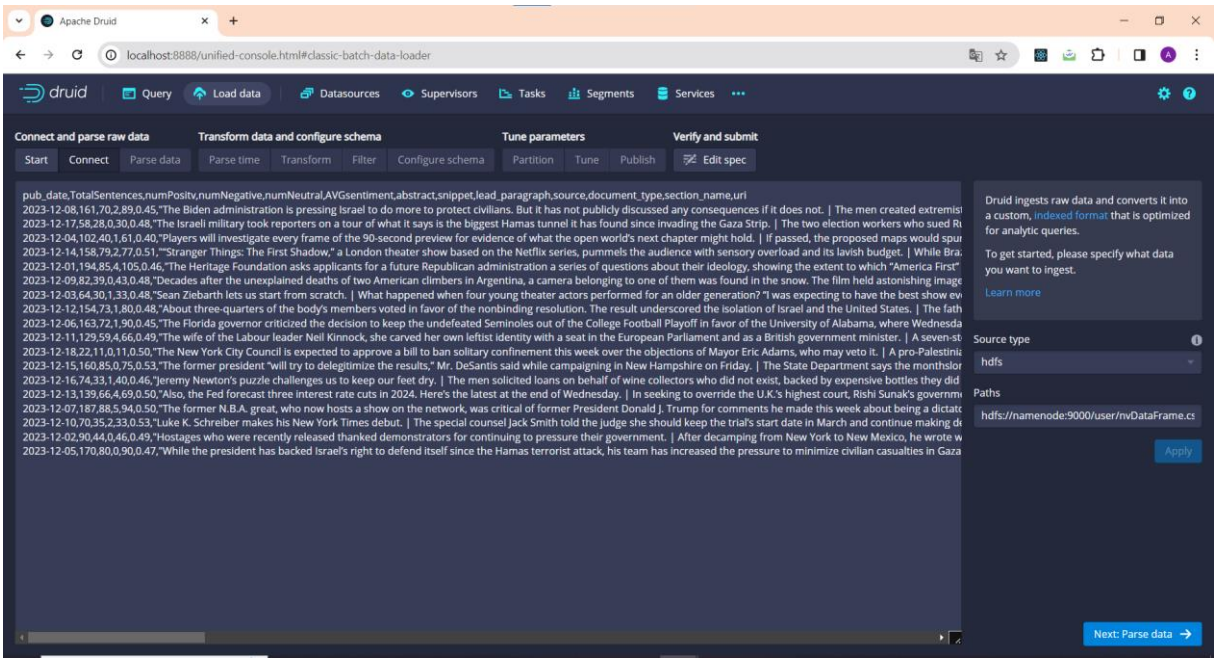
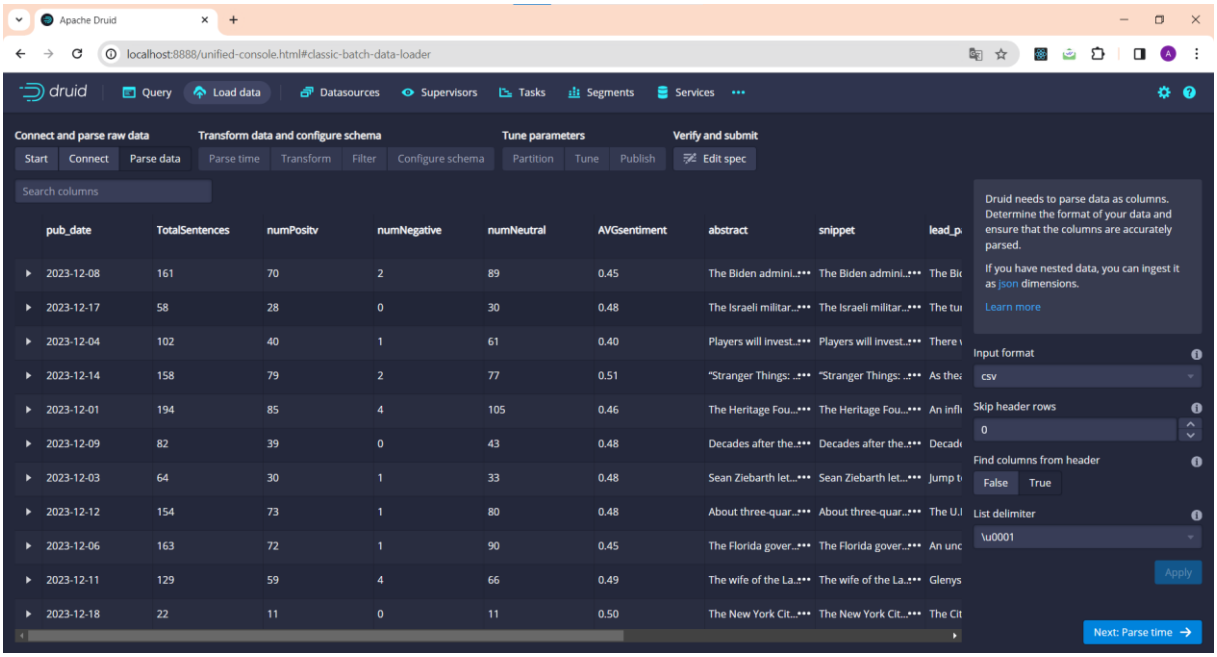
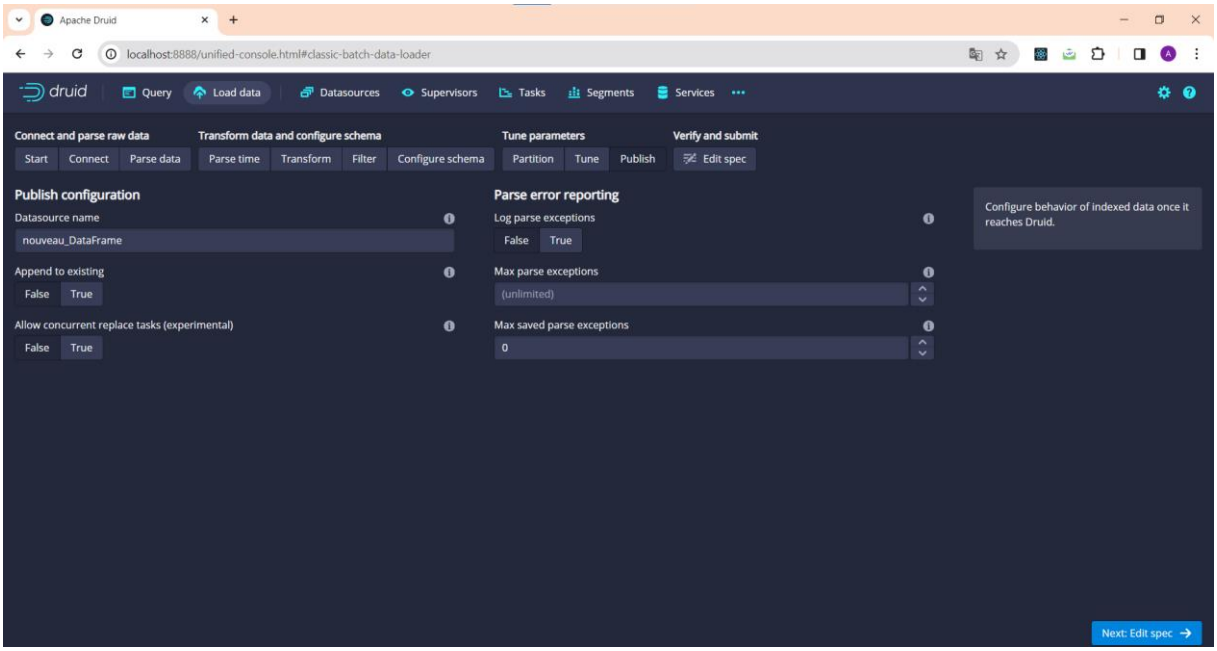


Figure 36:L'insertion du chemin hdfs de nouveau dataframe




Apache Druid
localhost:8888/unified-console.html#classic-batch-data-loader
druid
Query
Load data
Datasources
Supervisors
Tasks
Segments
Services
Connect and parse raw data
Start
Connect
Parse data
Transform data and configure schema
Parse time
Transform
Filter
Configure schema
Tune parameters
Partition
Tune
Publish
Verify and submit
Edit spec
Search columns
pub_date
TotalSentences
numPositv
numNegative
numNeutral
AVGsentiment
abstract
snippet
lead_p
2023-12-08 161 70 2 89 0.45
2023-12-17 58 28 0 30 0.48
2023-12-04 102 40 1 61 0.40
2023-12-14 158 79 2 77 0.51
2023-12-01 194 85 4 105 0.46
2023-12-09 82 39 0 43 0.48
2023-12-03 64 30 1 33 0.48
2023-12-12 154 73 1 80 0.48
2023-12-06 163 72 1 90 0.45
2023-12-11 129 59 4 66 0.49
2023-12-18 22 11 0 11 0.50
Druid needs to parse data as columns. Determine the format of your data and ensure that the columns are accurately parsed.
If you have nested data, you can ingest it as json dimensions.
Learn more
Input format
csv
Skip header rows
0
Find columns from header
False
True
List delimiter
\u0001
Apply
Next: Parse time


Apache Druid
localhost:8888/unified-console.html#classic-batch-data-loader
druid
Query
Load data
Datasources
Supervisors
Tasks
Segments
Services
Connect and parse raw data
Start
Connect
Parse data
Transform data and configure schema
Parse time
Transform
Filter
Configure schema
Tune parameters
Partition
Tune
Publish
Verify and submit
Edit spec
Publish configuration
Datasource name
nouveau_DataFrame
Append to existing
False
True
Allow concurrent replace tasks (experimental)
False
True
Parse error reporting
Log parse exceptions
False
True
Max parse exceptions
(unlimited)
Max saved parse exceptions
0
Configure behavior of indexed data once it reaches Druid.
Next: Edit spec

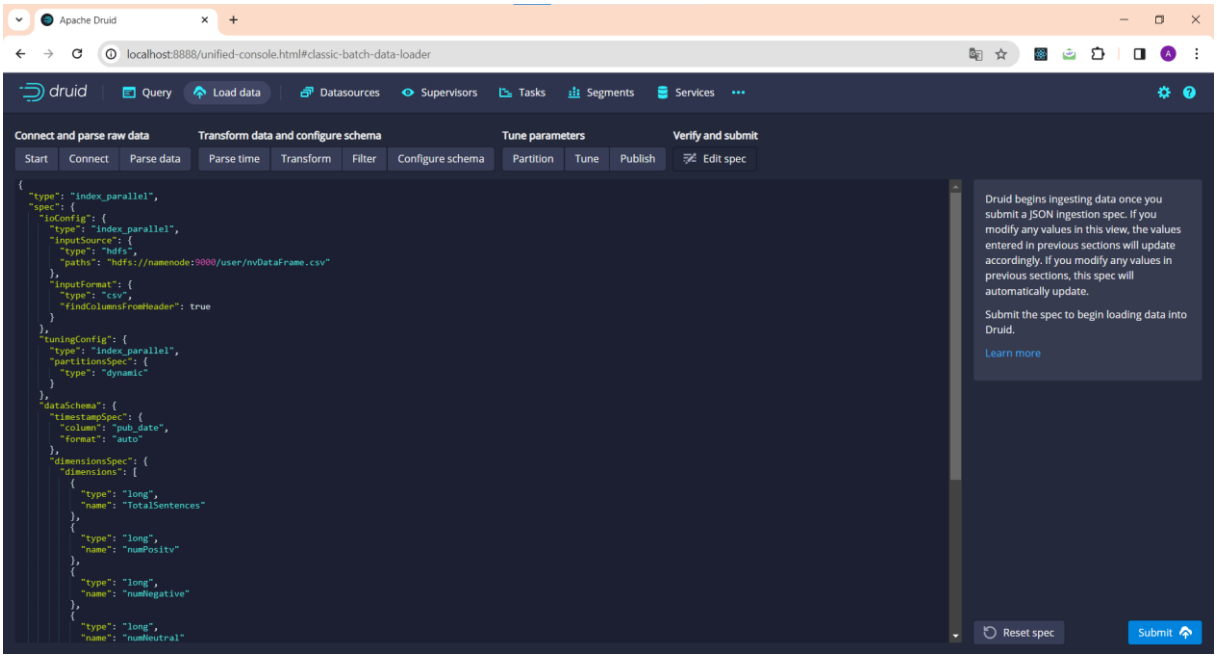


Figure 37:Le schéma de nouveau dataframe

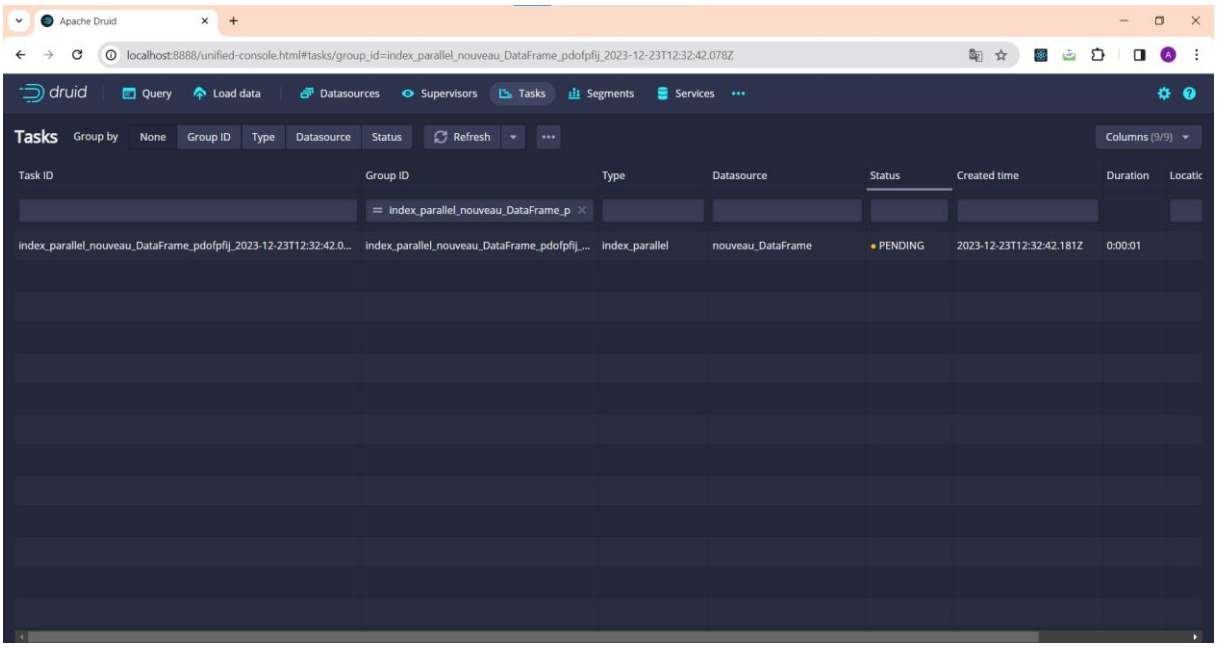


Figure 38:L'ajout de nouveau dataframe est entrain de s'éxécuter

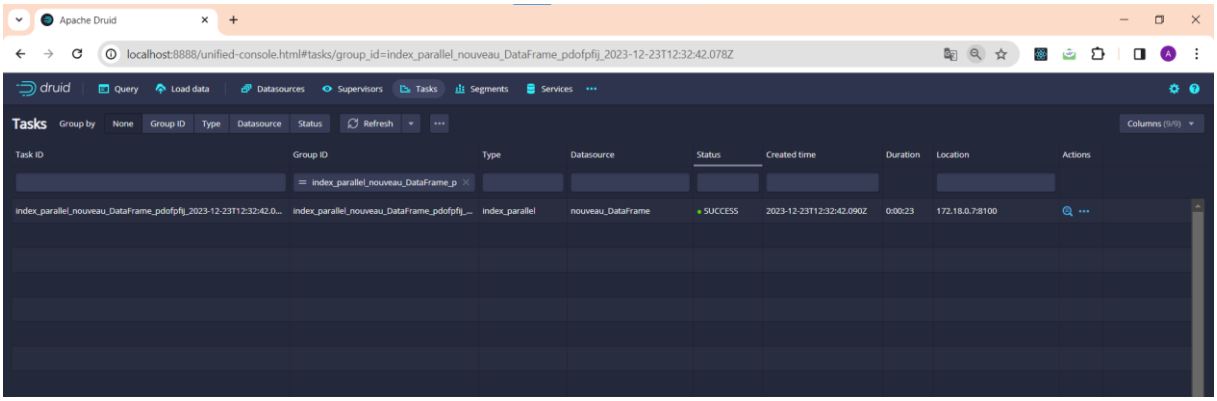


Figure 39:L'ajout de nouveau dataframe est entrain est éxécuté avec succés

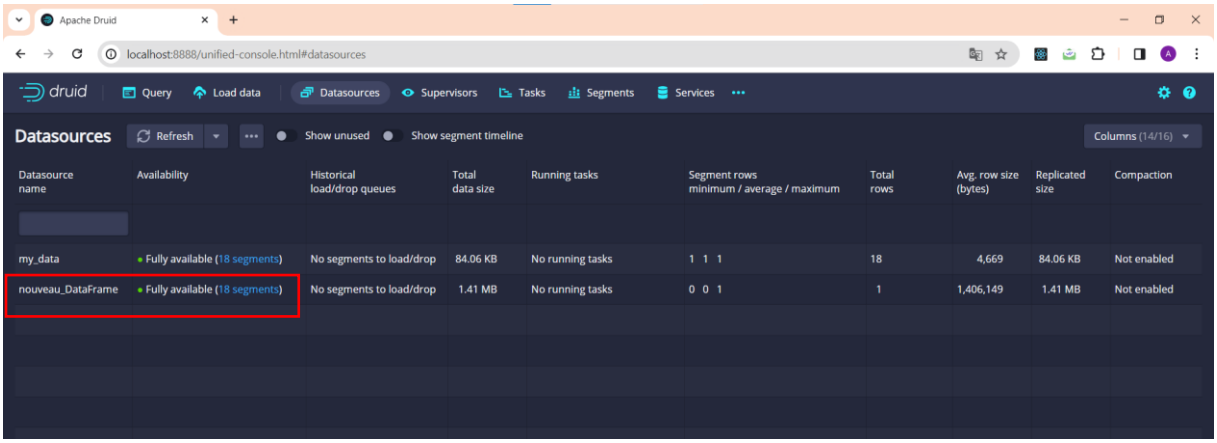
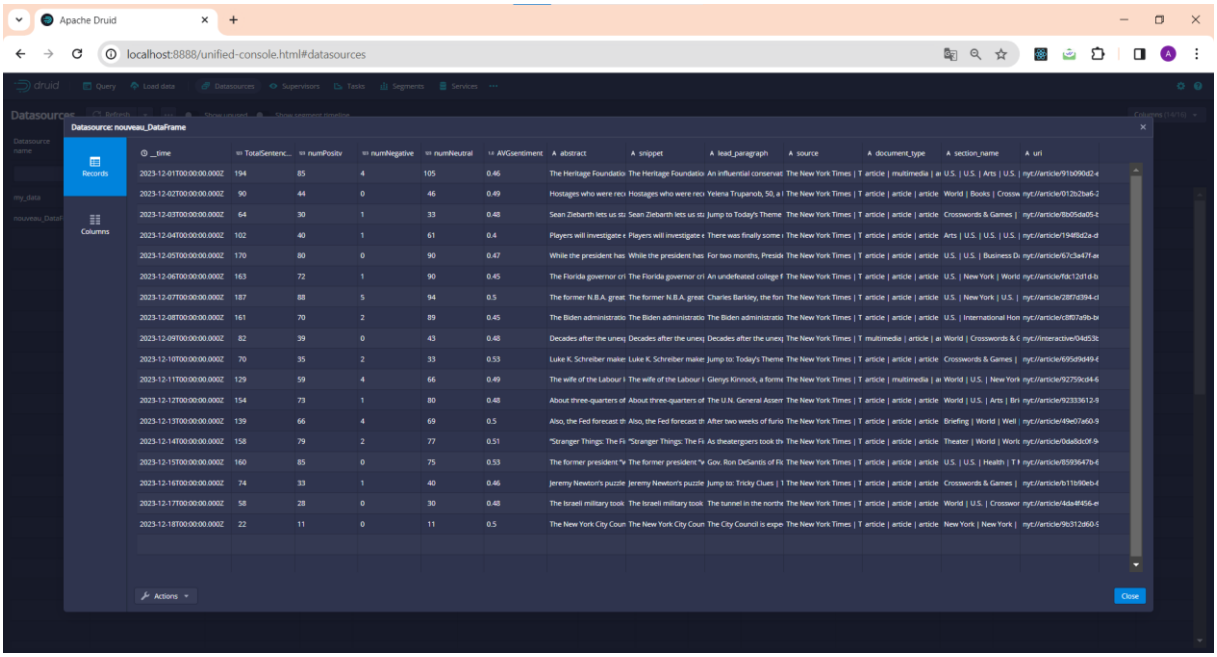
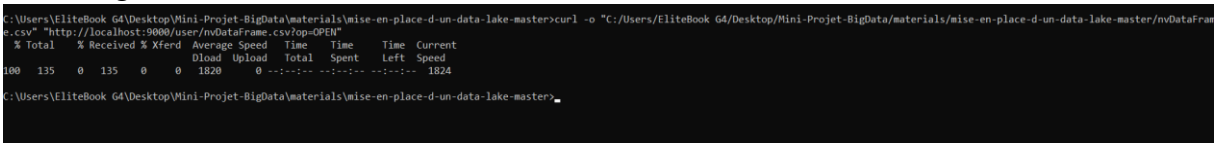


Figure 40:La création de nouveau datasource druid "nouveau_dataframe"



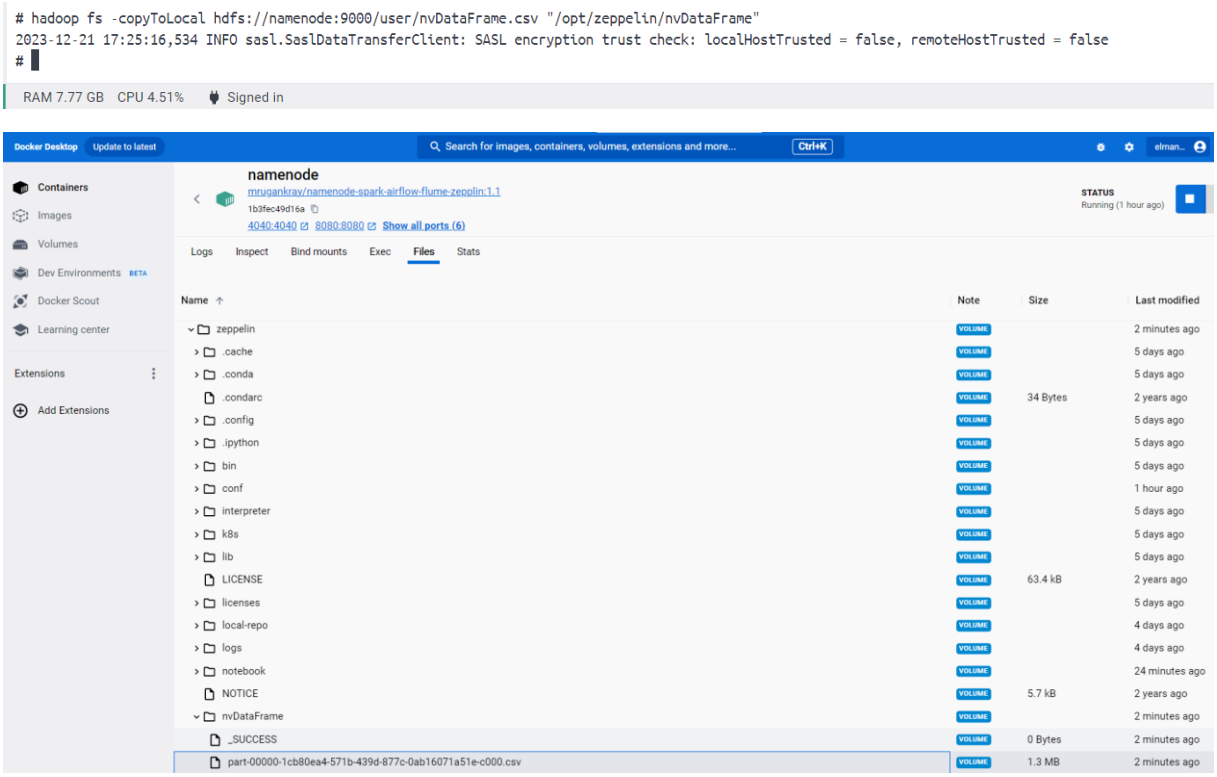
12. Visualisation du résultat de traitement effectue en utilisant Streamlit

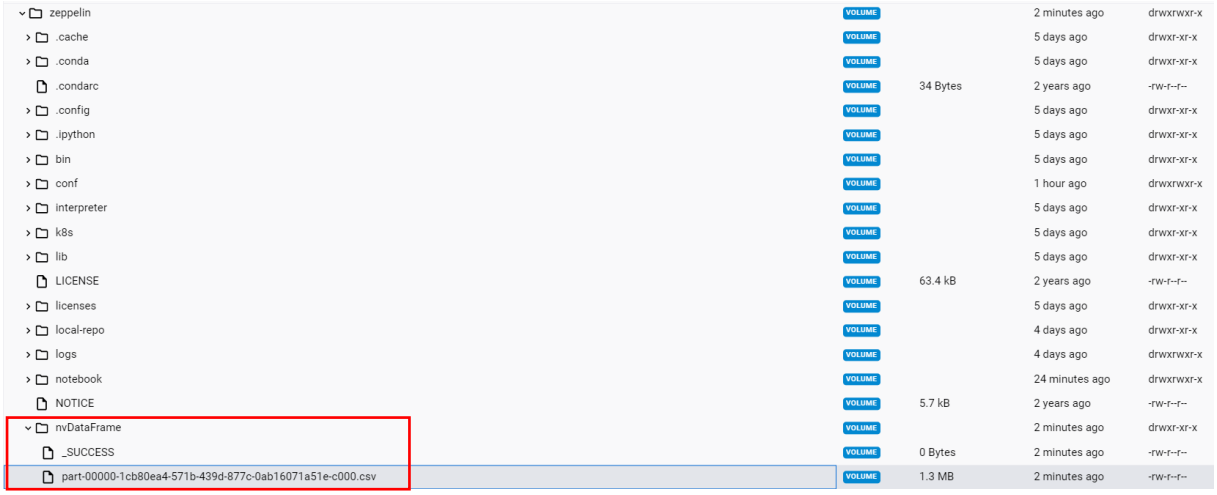
Téléchargement du fichier csv de nouveau dataframe :



⇨ curl -o "C:/Users/EliteBook G4/Desktop/Mini-Projet-BigData/materials/mise-en-place-d-un-data-lake-master/nvDataFrame.csv" http://localhost:9000/user/nvDataFrame.csv?op=OPEN

Ou bien il suffit de taper la commande suivante :





Puis vous pouvez télécharger le fichier qui se trouve dans le chemin **/opt/zeppelin/nvDataFrame** puis importer le dans votre dossier de travail ,afin que vous puissiez l'utiliser par la suite dans la visualisation avec Streamlit.

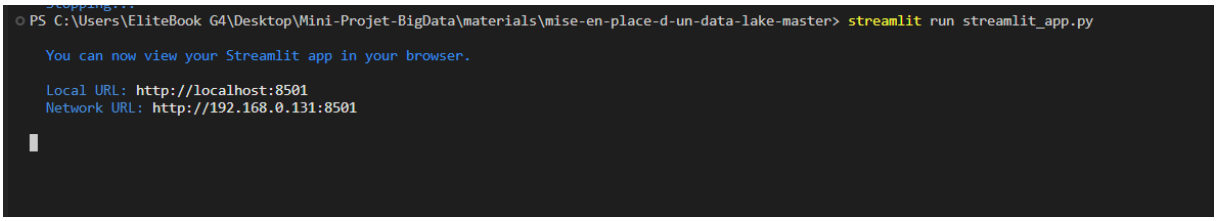
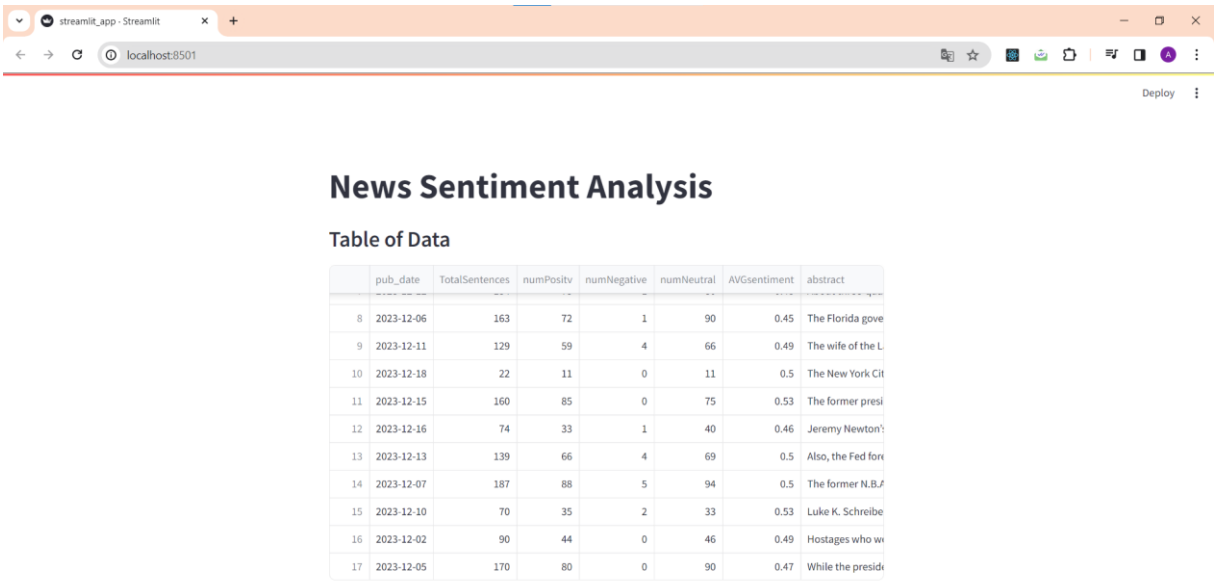


Tableau de données :

- Affiche l'ensemble du jeu de données sous forme de tableau, montrant différentes colonnes telles que la date de publication, la moyenne de sentiments de textes publiés pour chaque jour, le nombre de sentiments positives pour chaque jour et autres.



	pub_date	TotalSentences	numPositv	numNegative	numNeutral	AVGsentiment	abstract
8	2023-12-06	163	72	1	90	0.45	The Florida gove
9	2023-12-11	129	59	4	66	0.49	The wife of the L
10	2023-12-18	22	11	0	11	0.5	The New York Cit
11	2023-12-15	160	85	0	75	0.53	The former presi
12	2023-12-16	74	33	1	40	0.46	Jeremy Newton's
13	2023-12-13	139	66	4	69	0.5	Also, the Fed fore
14	2023-12-07	187	88	5	94	0.5	The former N.B.A
15	2023-12-10	70	35	2	33	0.53	Luke K. Schreibe
16	2023-12-02	90	44	0	46	0.49	Hostages who w
17	2023-12-05	170	80	0	90	0.47	While the presid

Figure 41:Le tableau des données du nouveau_dataframe

Nombre de sentiments positifs par jour :

- Présente un graphique à barres illustrant le nombre total de sentiments positifs agrégés par jour.

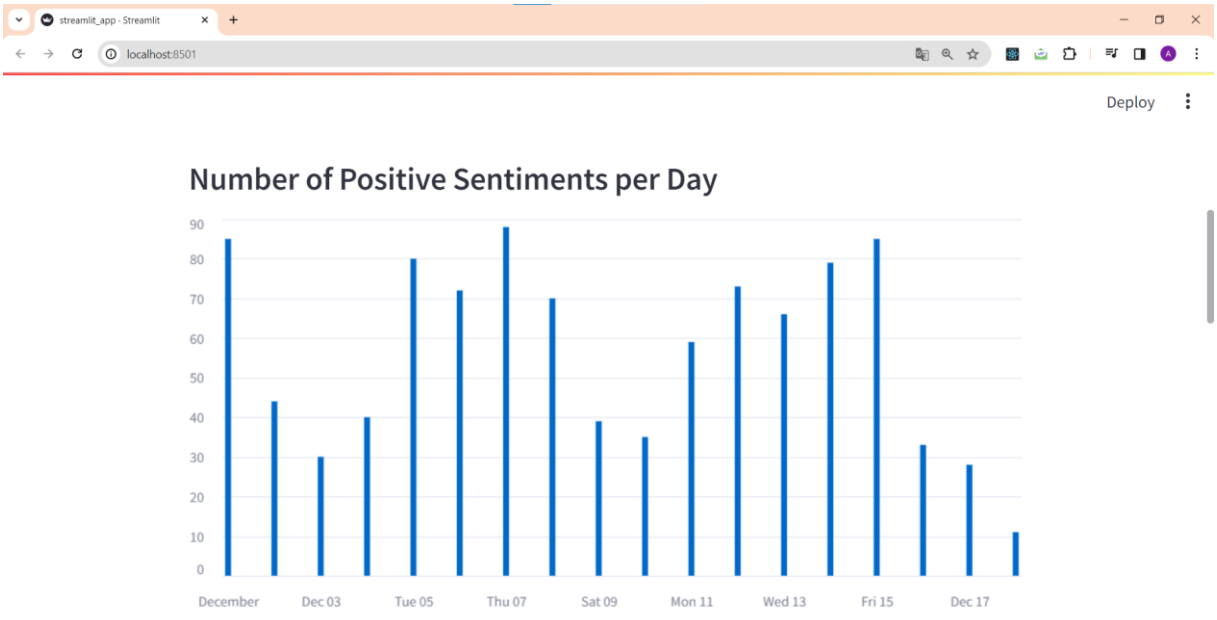


Figure 42:Le nombre de sentiment positifs par jour

Nombre de sentiments négatifs par jour :

- Fournit un graphique à barres montrant le nombre total de sentiments négatifs agrégés par jour.

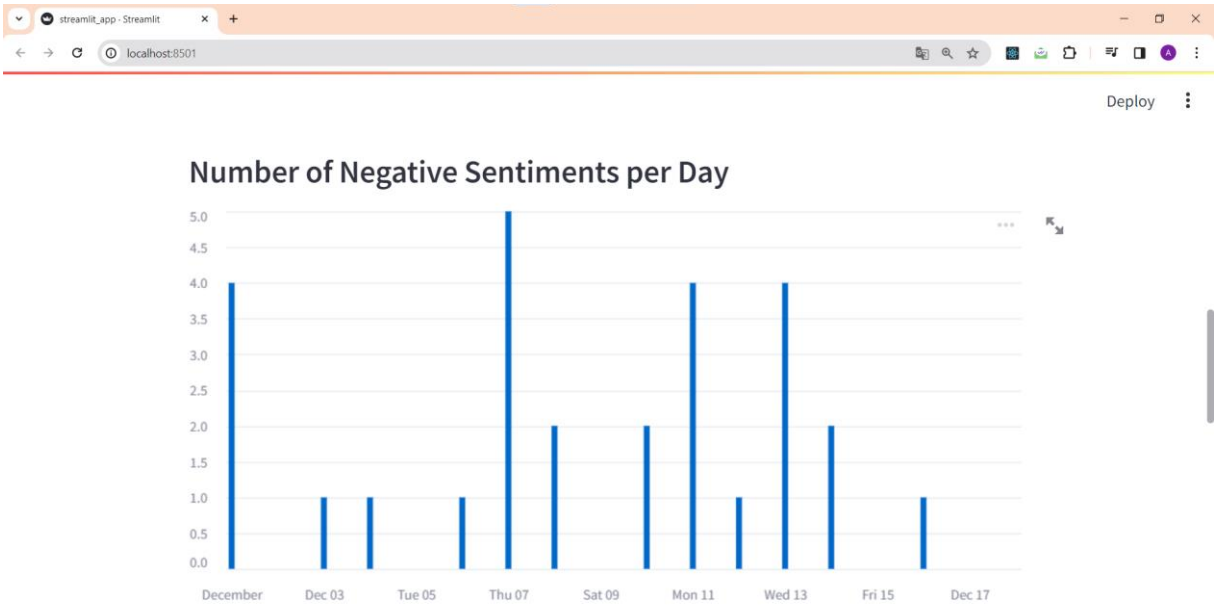


Figure 43:Le nombre de sentiment négatifs par jour

Sentiment moyen par jour :

- Affiche un graphique linéaire représentant le score de sentiment moyen par jour.

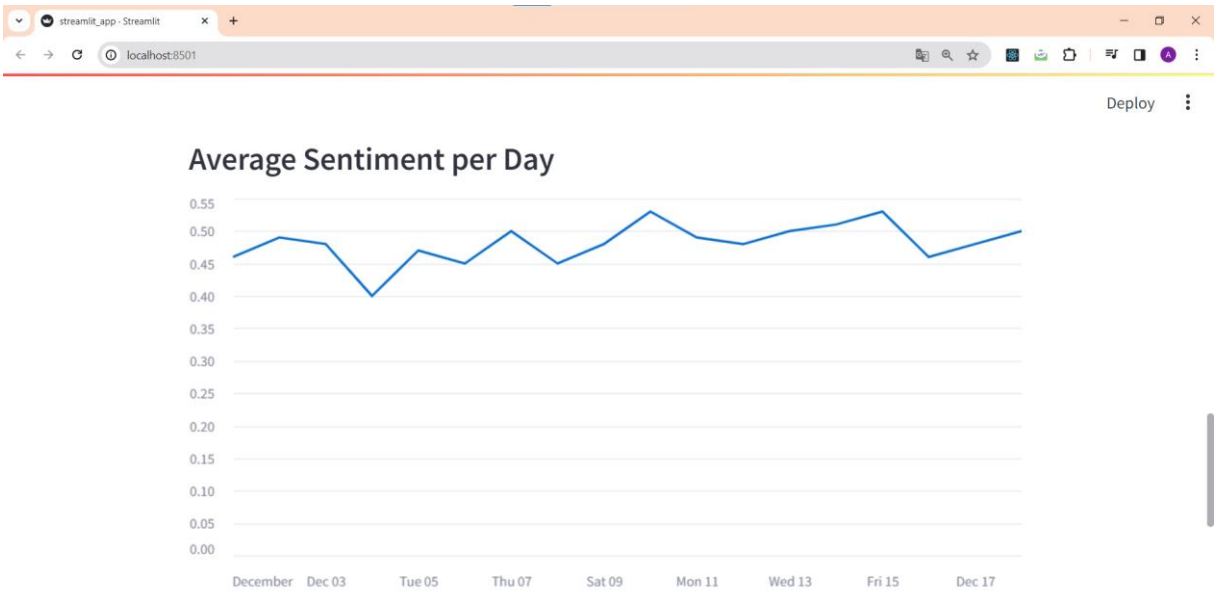


Figure 44:La moyenne des sentiments par jour

Nombre d'articles par jour :

- Affiche un graphique à barres représentant le nombre total d'articles par jour.

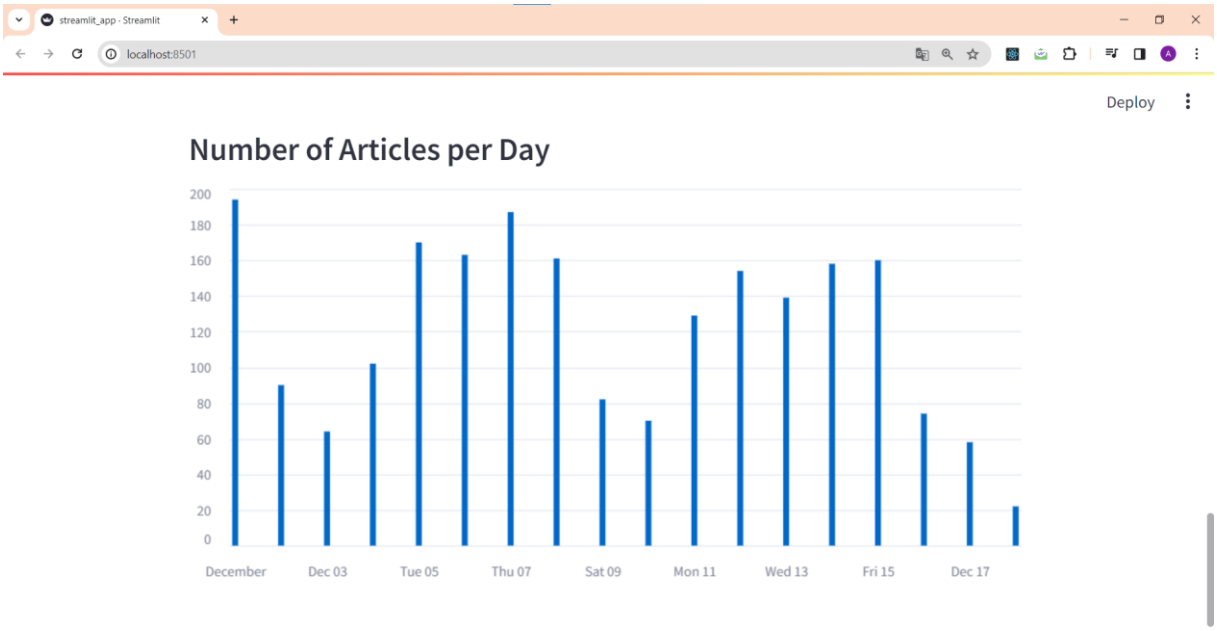


Figure 45:Le nombre des articles par jour

Distribution des scores de sentiment :

- Présente un histogramme affichant la distribution des scores de sentiment moyen à travers tous les points de données.

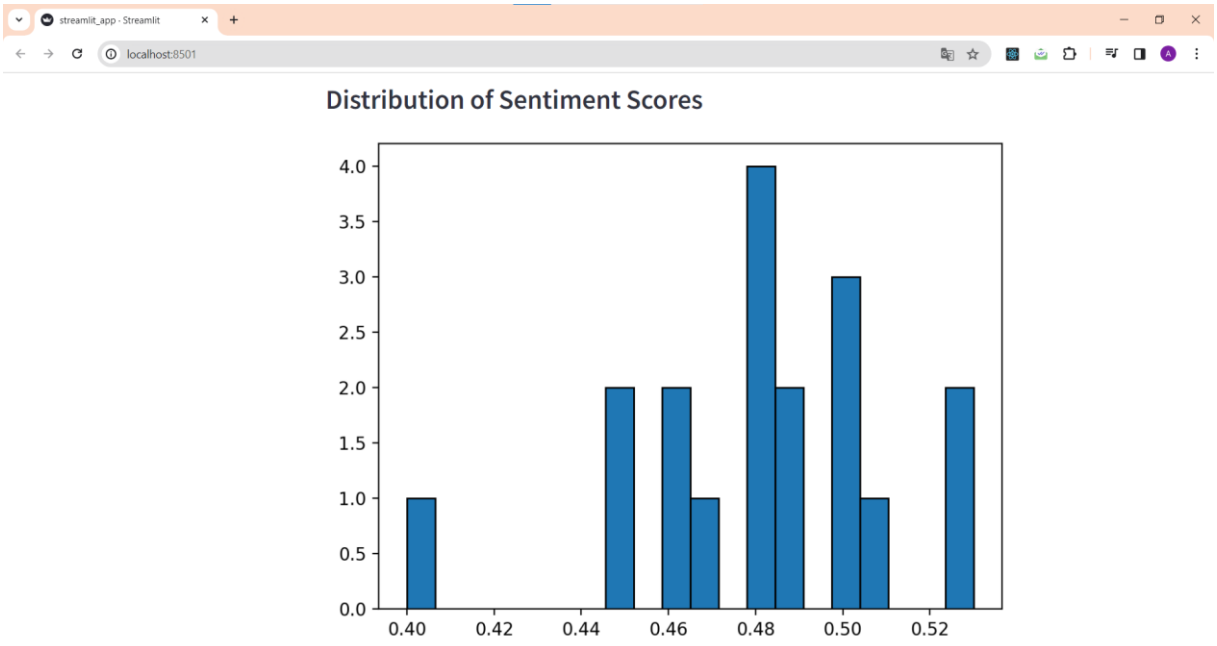
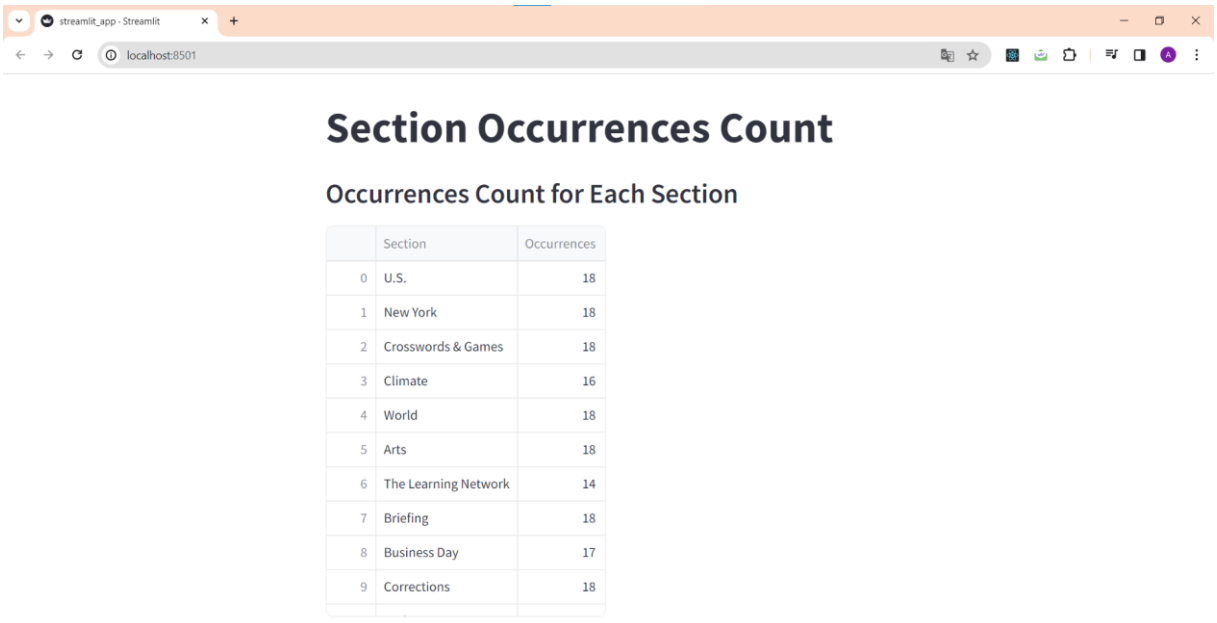


Figure 46:Répartition des scores de sentiments

Comptage des occurrences de sections :

- Affiche un tableau présentant le nombre d'occurrences pour chaque section prédéfinie (par exemple, U.S., World, Arts) dans les données



	Section	Occurrences
0	U.S.	18
1	New York	18
2	Crosswords & Games	18
3	Climate	16
4	World	18
5	Arts	18
6	The Learning Network	14
7	Briefing	18
8	Business Day	17
9	Corrections	18

Figure 47:Le nombre d'occurrences pour chaque section

Diagramme en barres du nombre d'occurrences :

- Fournit un diagramme en barres avec des couleurs différentes pour chaque section, illustrant le nombre d'occurrences pour chaque section.

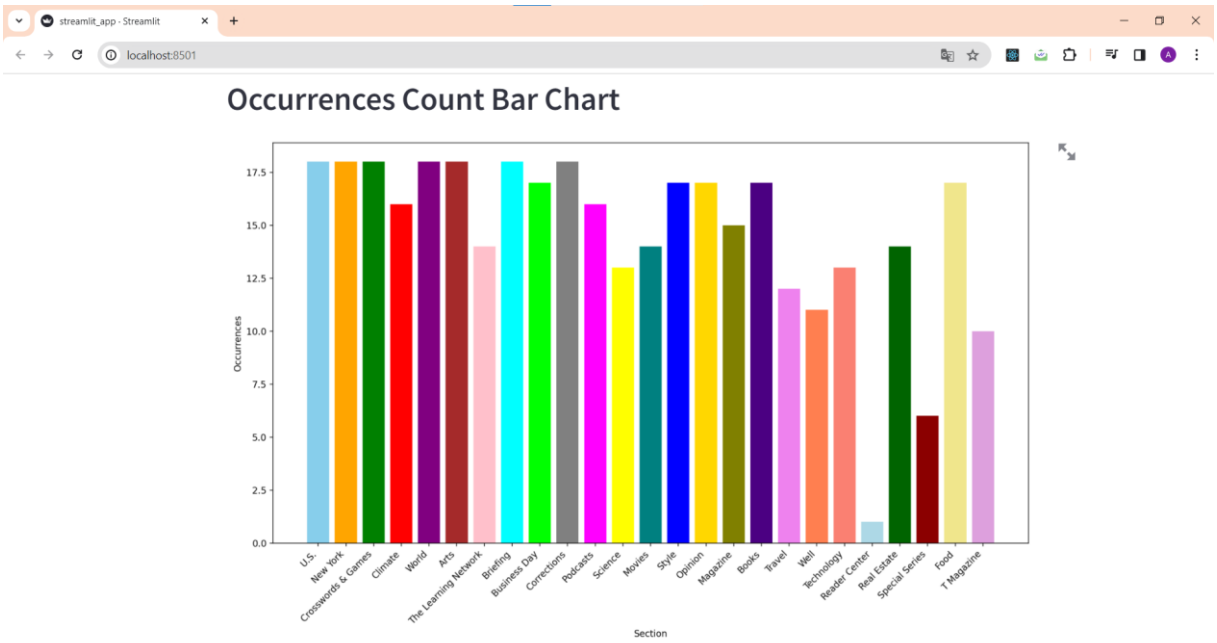


Figure 48:Le graphe à barres pour calculer le nombre d'occurrences pour chaque section

Nuage de mots des résumés(abstracts) :

- Génère un nuage de mots basé sur les résumés(abstracts) des articles, où la taille de chaque mot représente sa fréquence dans le texte.

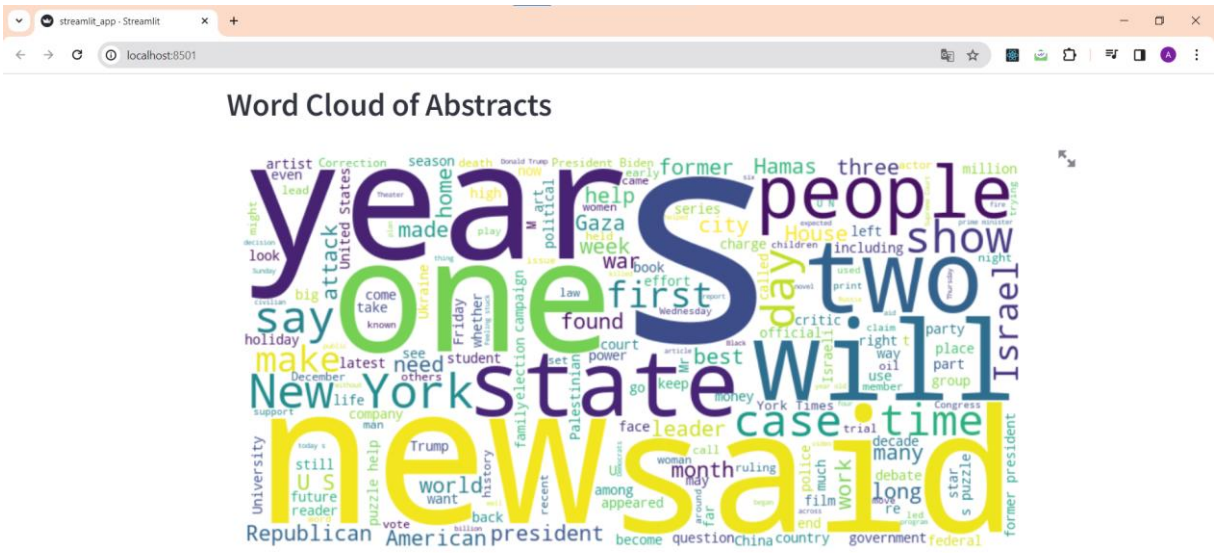


Figure 49:Le résumé de nuage de mots