

Rapport de projet - Big Data

Réputation des informations



Groupe :
G4

Membres du groupe :
NANA Dylan
NGOUNGOU Lilian
NGOUNOU Yann
NZOPET Luc

Table des matières

Étape 1 : Étude des besoins	2
1. Description des objectifs d'analyse de données	2
2. Étude de cas	2
3. Identification des sources de données adéquates	6
Étape 2 : Mise en place	7
1. Préparation d'un environnement de stockage et traitement de données	7
2. Construction d'un data lake	11
3. Traitement de données	13
4. Analyse des données	13
Étape 3 : Visualisation des résultats	14

Étape 1 : Étude des besoins

1. Description des objectifs d'analyse de données

Les fausses nouvelles (ou "fake news" en anglais) sont des informations qui sont délibérément créées pour tromper ou induire en erreur les gens. Ces informations peuvent être présentées sous la forme d'articles, de vidéos, de photos, de messages sur les réseaux sociaux ou de tout autre type de contenu.

Les fausses nouvelles peuvent avoir un impact négatif sur la société, car elles peuvent causer de la confusion et de la méfiance envers les médias et les institutions gouvernementales. Les personnes qui propagent des fausses nouvelles peuvent avoir des motivations diverses, comme la recherche d'attention, la manipulation politique, la promotion de produits et services, ou simplement le désir de causer des troubles.

Ainsi pour pouvoir pallier cela, nous est-il possible de différencier parmi des nouvelles, celles qui sont vraies de celles qui sont fausses ? Quels sont les outils que nous pouvons mettre en place pour y parvenir ?

2. Étude de cas

Pour notre étude de cas, nous avons eu recours à l'analyse de deux jeux de données. Le premier recense les réponses au sondage sur les fausses nouvelles par type et par ville. Le second, quant à lui, est constitué du total des réponses au sondage sur les fausses nouvelles par type et par État. Ces sondages ont été soumis à des dizaines de milliers de lycéens des 50 États américains et ils ont été interrogés sur ce qu'ils savent et pensent des "fake news". Il était donc question pour nous de ressortir des tableaux et des graphiques à partir de ces jeux de données et puis d'en déduire une conclusion des résultats obtenus.

```
from pyspark.sql import SparkSession
spark = SparkSession.builder.appName('ops').getOrCreate()
```

Exécuter la commande "%pip install pyspark" si ce n'est pas déjà fait afin d'installer la librairie "pyspark".

```
df1 = spark.read.csv('Fake News Poll Responses by Type and City.csv', inferSchema = True, header = True)
df1.show()
```

Event Category	City	Total Events
Yes, I have heard...	Chicago	926
Yes, I have heard...	(not set)	797
Yes, I have heard...	Dallas	702
Yes, I'm good at ...	Chicago	581
Yes, I have heard...	New York	569
Ignore it	Chicago	527
Yes, I'm good at ...	(not set)	497
Ignore it	(not set)	470
Yes, I'm good at ...	Dallas	457
Yes, I have heard...	Atlanta	442
Ignore it	Dallas	377
Yes, I'm good at ...	New York	348
Yes, I have heard...	Los Angeles	343
Yes, I have heard...	Detroit	332
Yes, I have heard...	Washington	331
Ignore it	New York	311
No, I'm bad at sp...	Chicago	298
No, I'm bad at sp...	(not set)	290
Yes, I have heard...	Houston	289
Yes, I have heard...	San Francisco	282

only showing top 20 rows

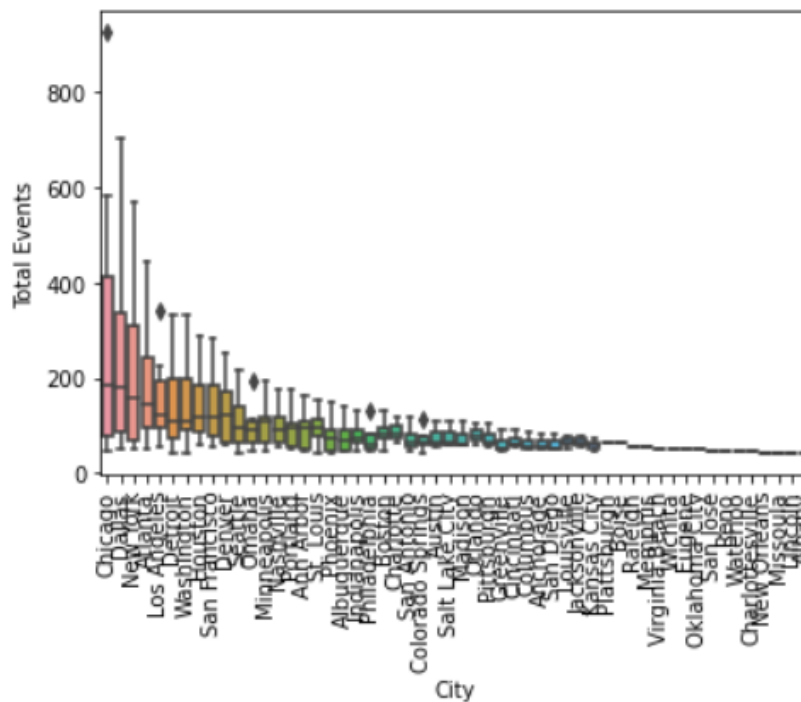
```
df1.createOrReplaceTempView('table1')
query1 = spark.sql("select * from table1 where City != '(not set)'")
query1.show()
```

Event Category	City	Total Events
Yes, I have heard...	Chicago	926
Yes, I have heard...	Dallas	702
Yes, I'm good at ...	Chicago	581
Yes, I have heard...	New York	569
Ignore it	Chicago	527
Yes, I'm good at ...	Dallas	457
Yes, I have heard...	Atlanta	442
Ignore it	Dallas	377
Yes, I'm good at ...	New York	348
Yes, I have heard...	Los Angeles	343
Yes, I have heard...	Detroit	332
Yes, I have heard...	Washington	331
Ignore it	New York	311
No, I'm bad at sp...	Chicago	298
Yes, I have heard...	Houston	289
Yes, I have heard...	San Francisco	282
Yes, I'm good at ...	Atlanta	279
Yes, I have heard...	Denver	254
Call them out	Chicago	232
Ignore it	Atlanta	228

only showing top 20 rows

```
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df1_pandas = query1.toPandas()
dfone = df1_pandas[0:200]
sns.boxplot(x = dfone['City'], y = dfone['Total Events'], data = dfone)
plt.xticks(rotation = 90)
plt.show()
```

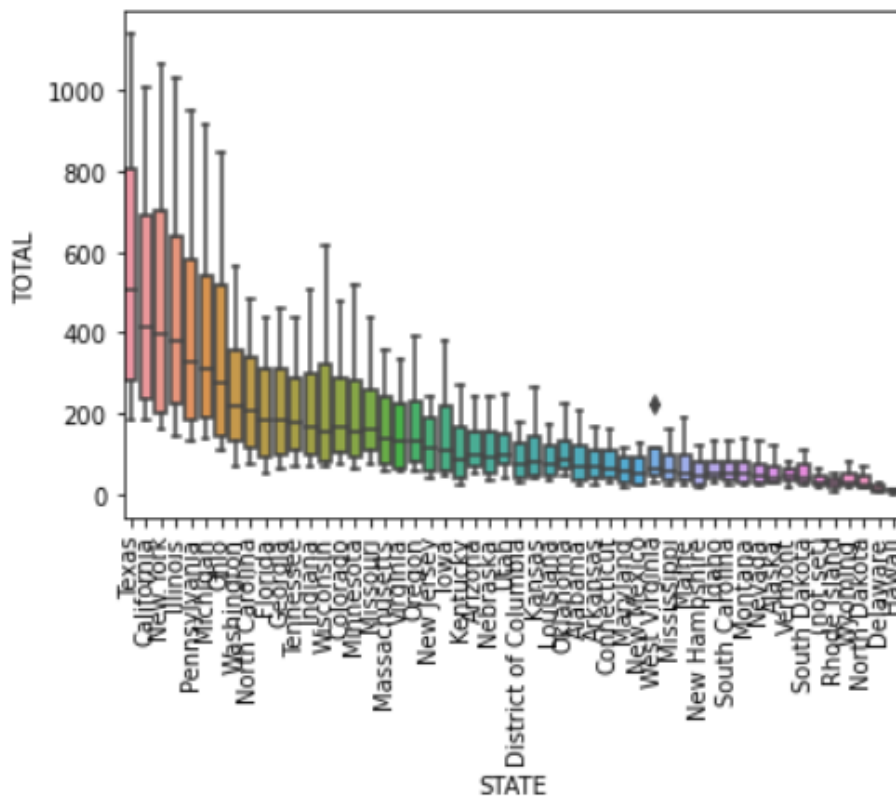


```
df2 = spark.read.csv('Fake News Poll Total Responses by Type and State.csv', inferSchema = True, header = True)
df2.show()
```

RESPONSE	STATE	TOTAL
Call them out	Texas	692
Call them out	California	583
Call them out	New York	577
Call them out	Illinois	509
Call them out	Pennsylvania	461
Call them out	Michigan	417
Call them out	Ohio	405
Call them out	Washington	290
Call them out	North Carolina	288
Call them out	Florida	268
Call them out	Georgia	258
Call them out	Tennessee	239
Call them out	Indiana	228
Call them out	Wisconsin	226
Call them out	Colorado	225
Call them out	Minnesota	207
Call them out	Missouri	203
Call them out	Massachusetts	201
Call them out	Virginia	190
Call them out	Oregon	176

only showing top 20 rows

```
df2_pandas = df2.toPandas()
dftwo = df2_pandas[0:200]
sns.boxplot(x = dftwo['STATE'], y = dftwo['TOTAL'], data = dftwo)
plt.xticks(rotation = 90)
plt.show()
```



Nous constatons que dans chacun de ces deux diagrammes en boîte, nous observons à chaque fois des points isolés du reste des boîtes. Ils sont appelés “points aberrants supérieurs”. Ils indiquent en fait des valeurs extrêmes et qui, pour la plupart du temps, ne sont pas fiables. D’où l’existence de notre étude qui révèle donc là, des fausses nouvelles. En effet, ces types de points extrêmes doivent être exclus du jeu de données afin d’assurer la fiabilité de celui-ci. Toutefois, il peut s’avérer que cette valeur soit normale. Dans ces cas, elle peut fournir des informations précieuses sur la distribution des données.

3. Identification des sources de données adéquates

Parmi les sources de données, nous avons un dataset nommé “news.csv” contenant un ensemble d’articles divers. On a :

- un ID
- un titre
- un texte
- la véracité

	A	B	C	D
1	ID	title	text	label
2		2 Study: women had to drive 4 times farther after Texas laws closed abortion clinics	Ever since Texas laws closed about half of the state's abortion clinics in 2013, researchers have be	REAL
3		3 Trump, Clinton clash in dueling DC speeches	Donald Trump and Hillary Clinton, now at the starting line of a general election race, traded shots	REAL
4		5 As Reproductive Rights Hang In The Balance, Debate Moderators Drop The Ball	WASHINGTON -- Forty-three years after the Supreme Court established the right to a safe and leg	REAL
5		6 Despite Constant Debate, Americans' Abortion Opinions Rarely Change	It's been a big week for abortion news. Carly Fiorina's passionate (if inaccurate) depiction of a Pla	REAL
6		7 Obama Argues Against Government Shutdown Over Planned Parenthood	President Barack Obama said Saturday night that Congress should not shut down the federal gov	REAL
7		9 Planned Parenthood's lobbying effort; pay raises for federal workers; and the future Fed rates	PLANNED PARENTHOOD'S LOBBYING GETS AGGRESSIVE. Congress may have spent August away fr	REAL
8		10 Scalia's death comes just a month before the court's biggest abortion case in years	The unexpected death of Justice Antonin Scalia comes less than a month before the Supreme Cou	REAL
9		12 Fact Check: Was Planned Parenthood Started To 'Control' The Black Population?	Fact Check: Was Planned Parenthood Started To 'Control' The Black Population? Ben Carson allege	REAL
10		14 How Planned Parenthood hoax avoids the truth	Errol Louis is the host of "Inside City Hall," a nightly political show on NY1, a New York all-news ch	REAL
11		16 P. Parenthood Chief Goes Toe-to-Toe with Attackers	WASHINGTON -- Planned Parenthood President Cecile Richards withstood nearly five hours of Re	REAL
12		17 Stop the vendetta against Planned Parenthood	THE STING videos targeting Planned Parenthood are hard to watch. Doctors talk clinically, some s	REAL
13		18 How Planned Parenthood could shut down the government	A verdict in 2017 could have sweeping consequences for tech startups.	REAL
14		19 Planned Parenthood does damage control as GOP demands answers	A verdict in 2017 could have sweeping consequences for tech startups.	REAL

Étape 2 : Mise en place

1. Préparation d'un environnement de stockage et traitement de données

Pour préparer notre environnement de stockage et de traitement de données, nous allons d'abord construire une image pour utiliser Hadoop HDFS, Jupyter Notebook et NiFi avec l'aide de Docker et un fichier de configuration .yaml contenant les services dont nous aurons besoin comme suit :

```
version: "3.9"

services:
  nifi:
    image: apache/nifi:1.20.0
    environment:
      NIFI_WEB_HTTP_PORT: 8080
      NIFI_WEB_HTTP_HOST: 0.0.0.0
    ports:
      - 8080:8080
    volumes:
      - ./data:/data
      - ./driver:/driver
      - ./extensions:/opt/nifi/nifi-current/extensions
      - ./content_repository:/opt/nifi/nifi-current/content_repository
      - ./database_repository:/opt/nifi/nifi-current/database_repository
      - ./flowfile_repository:/opt/nifi/nifi-current/flowfile_repository
      - ./provenance_repository:/opt/nifi/nifi-current/provenance_repository
    namenode:
```



```

image: bde2020/hadoop-namenode:2.0.0-hadoop3.2.1-java8
container_name: namenode
restart: always
ports:
  - 9870:9870
  - 9000:9000
volumes:
  - hadoop_namenode:/hadoop/dfs/name
environment:
  - CLUSTER_NAME=test
env_file:
  - ./hadoop.env

datanode:
image: bde2020/hadoop-datanode:2.0.0-hadoop3.2.1-java8
container_name: datanode
restart: always
volumes:
  - hadoop_datanode:/hadoop/dfs/data
environment:
  SERVICE_PRECONDITION: "namenode:9870"
env_file:
  - ./hadoop.env

resourcemanager:
image: bde2020/hadoop-resourcemanager:2.0.0-hadoop3.2.1-java8
container_name: resourcemanager
restart: always
environment:
  SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864"
env_file:
  - ./hadoop.env

nodemanager1:
image: bde2020/hadoop-nodemanager:2.0.0-hadoop3.2.1-java8
container_name: nodemanager
restart: always
environment:
  SERVICE_PRECONDITION: "namenode:9000 namenode:9870 datanode:9864
resourcemanager:8088"
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"
  volumes:
    - hadoop_historyserver:/hadoop/yarn/timeline
  env_file:
    - ./hadoop.env
notebook:
  image: jupyter/pyspark-notebook:latest
  container_name: python_notebook
  labels:
    name: jupyter notebook
  ports:
    - "8888:8888"
  volumes:
    - C:/projet Big Data/code:/home/jovyan
  build: .

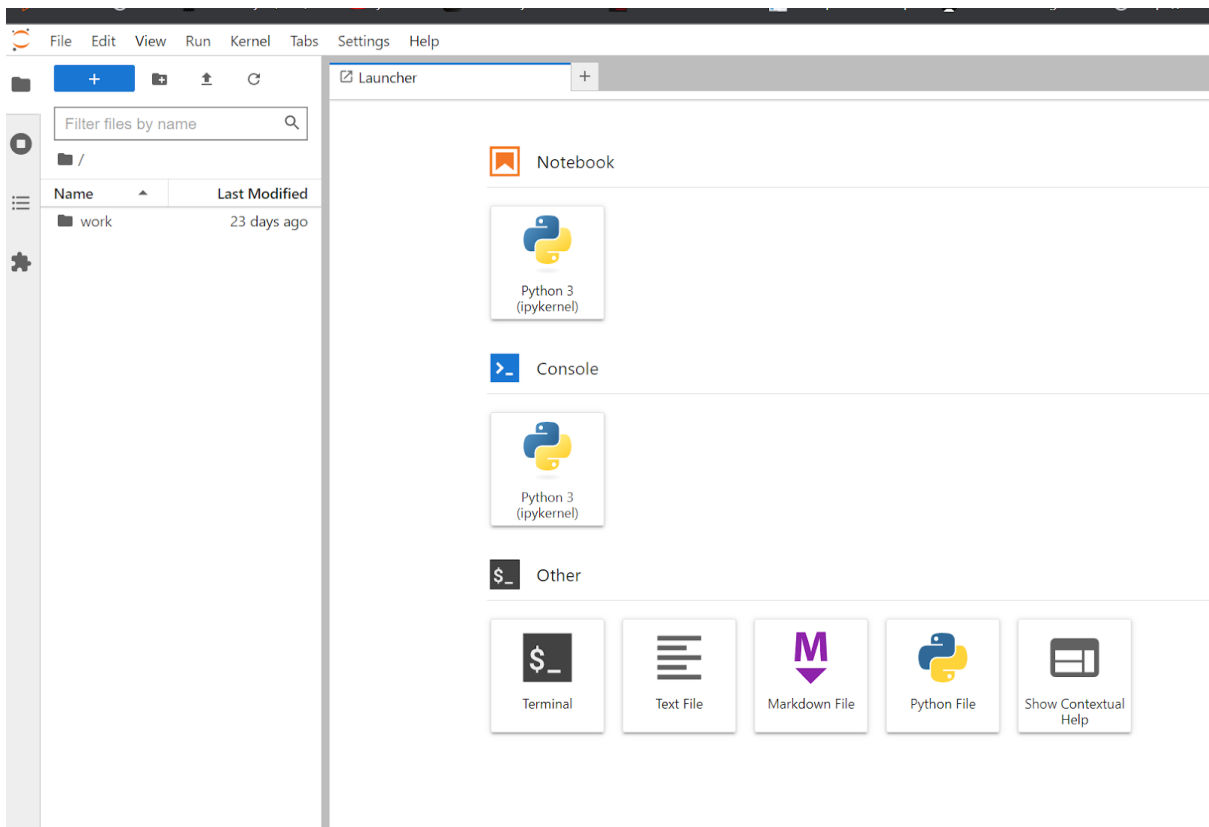
volumes:
  hadoop_namenode:
  hadoop_datanode:
  hadoop_historyserver:

```

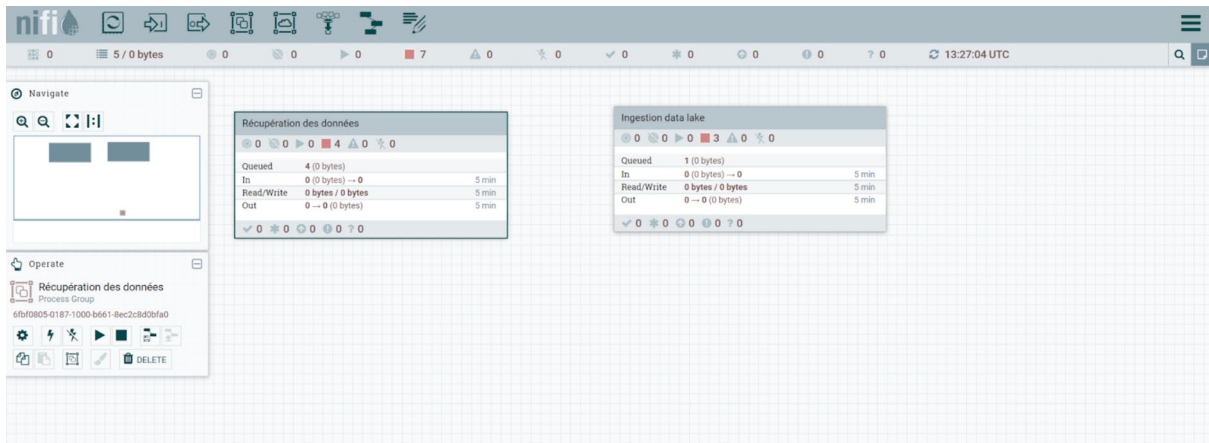
Après construction des images et des conteneurs.

<input type="checkbox"/>	Name	Image	Status	Port(s)	Started	Actions
<input type="checkbox"/>	 projetbigdata	-	Running (6/7)			  
<input type="checkbox"/>	 python_notebook 97c51d075484 	jupyter/pyspark-notebook:late	Running	8888:8888 	2 hours ago	  
<input type="checkbox"/>	 nodemanager 097bd2aef855 	bde2020/hadoop-nodemanager	Running		2 hours ago	  
<input type="checkbox"/>	 datanode 644f87d54ac9 	bde2020/hadoop-datanode:2.0	Running		2 hours ago	  
<input type="checkbox"/>	 nifi-1 be02de224f58 	apache/nifi:1.20.0	Exited	8080:8080 		  
<input type="checkbox"/>	 historyserver 239dfef252fe 	bde2020/hadoop-historyserver	Running		2 hours ago	  
<input type="checkbox"/>	 resourcemanager a1bec3b787cb 	bde2020/hadoop-resourcemanager	Running		2 hours ago	  
<input type="checkbox"/>	 namenode 29c2eceb9365 	bde2020/hadoop-namenode:2.0	Running	9000:9000  9870:9870 	2 hours ago	  

- Jupyter Notebook : Jupyter Notebook est un outil qui permet aux utilisateurs du langage Python de créer et de partager des documents interactifs contenant du code dynamique et exécutable, des visualisations de contenus, des textes de documentation et des équations.



- NiFi : NiFi est un logiciel libre de gestion de flux de données. Il permet de gérer et d'automatiser des flux de données entre plusieurs systèmes informatiques, à partir d'une interface web et dans un environnement distribué. On s'en servira comme ETL.



- Hadoop HDFS : HDFS (Hadoop Distributed File System) est un système de fichier distribué permettant de stocker et de récupérer des fichiers en un temps record. Il s'agit de l'un des composants basiques du framework Hadoop Apache, et plus précisément de son système de stockage.



Overview 'namenode:9000' (active)

Started:	Mon Apr 17 13:21:49 +0200 2023
Version:	3.2.1, rb3cbbb467e22ea829b3808f4b7b01d07e0bf3842
Compiled:	Tue Sep 10 17:56:00 +0200 2019 by rohithsharmaks from branch-3.2.1
Cluster ID:	CID-b405fb1d-d295-4e28-aad3-d1d6e7634190
Block Pool ID:	BP-357464366-172.18.0.7-1681121902117

Summary

Security is off.

Safemode is off.

36 files and directories, 28 blocks (28 replicated blocks, 0 erasure coded block groups) = 64 total filesystem object(s).

Heap Memory used 125.34 MB of 299 MB Heap Memory. Max Heap Memory is 1.64 GB.

Non Heap Memory used 59.73 MB of 61.13 MB Committed Non Heap Memory. Max Non Heap Memory is <unbounded>.

Configured Capacity:	1006.85 GB
Configured Remote Capacity:	0 B
DFS Used:	32.8 MB (0%)

2. Construction d'un data lake

Pour construire notre data lake, nous chargerons les fichiers en local pour les stocker dans HDFS(, **ensuite reçu**). Nous nous servirons ainsi de l'espace dédié que NiFi peut se référencer en local dans le fichier .yaml :

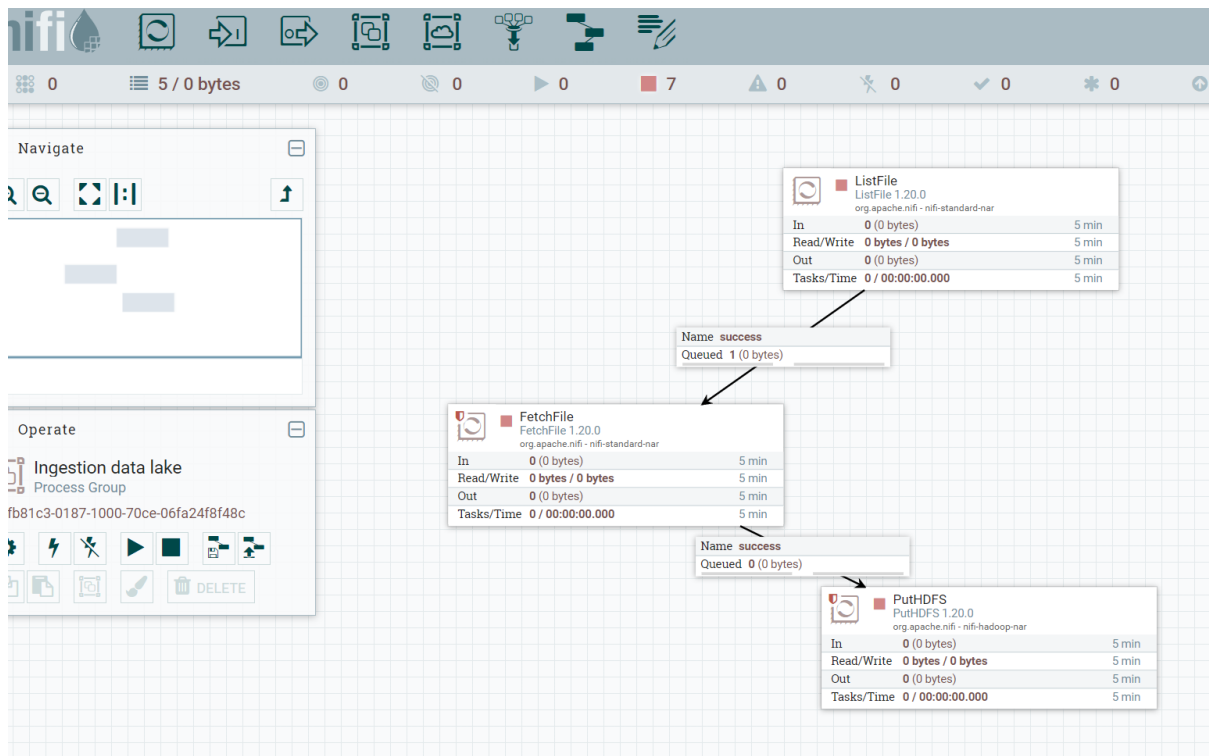
- Espace NiFi entré :

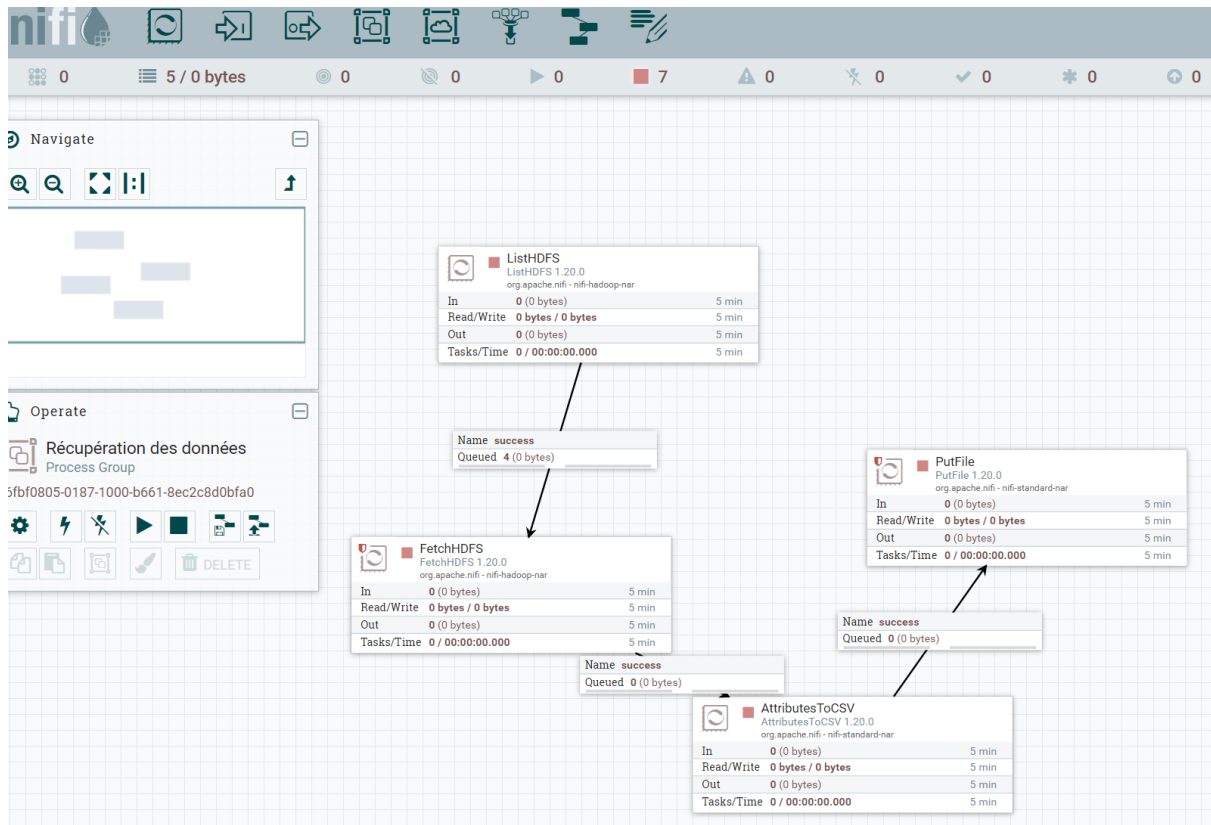
```
volumes:  
  - ./data:/data
```

- Espace NiFi sortie et Jupyter entré :

```
volumes:  
  - C:/projet Big Data/code:/home/jovyan
```

Nous construisons des groupes de processus dont “Récupération de données” et “Ingestion de data lake”.





Après ingestion, nous pouvons observer les données stockées dans HDFS.

Permission	Owner	Group	Size	Last Modified	Replication	Block Size	Name
-rw-r--r--	nifi	supergroup	1.5 MB	Apr 10 13:34	3	128 MB	Fake News Poll Responses by Type and City.csv
-rw-r--r--	nifi	supergroup	28.83 KB	Apr 10 13:34	3	128 MB	Fake News Poll Total Responses by Type and State.csv
-rw-r--r--	nifi	supergroup	60.63 KB	Apr 10 13:34	3	128 MB	State Aggregates.xlsx
-rw-r--r--	nifi	supergroup	89.71 KB	Apr 10 13:34	3	128 MB	data.json
-rw-r--r--	nifi	supergroup	29.27 MB	Apr 10 13:34	3	128 MB	news.csv
-rw-r--r--	nifi	supergroup	22.21 KB	Apr 10 13:34	3	128 MB	sample_submission.csv
-rw-r--r--	nifi	supergroup	1.18 KB	Apr 10 13:34	3	128 MB	script.js
-rw-r--r--	nifi	supergroup	410.92 KB	Apr 10 13:34	3	128 MB	test.csv
-rw-r--r--	nifi	supergroup	964.56 KB	Apr 10 13:34	3	128 MB	train.csv

Showing 1 to 9 of 9 entries

Previous 1 Next

3. Traitement de données

Pour traiter nos données, nous avons opté pour une fonction en Python qui permet de nettoyer les textes du dataset.

```
def wordopt (text):
    text = text.lower()
    text = re.sub('[.*?\\]', '', text)
    text = re.sub("\\W", '', text)
    text = re.sub('https?://\\S+/www\\.\\S+', '', text)
    text = re.sub('<.*?>+', '', text)
    text = re.sub('[%s] % re.escape(string.punctuation)', '', text)
    text = re.sub('\\n', '', text)
    text = re.sub('\\ w*\\d\\w*', '', text)
    return text
```

4. Analyse des données

Scikit-learn, encore appelé “sklearn”, est la bibliothèque pour le machine learning en Python. Elle fournit une sélection d’outils efficaces pour l’apprentissage automatique et la modélisation statistique, notamment la classification, la régression et le clustering via une interface cohérente en Python. Cette bibliothèque, qui est en grande partie écrite en Python, s’appuie sur NumPy, SciPy et Matplotlib.

Nous utiliserons le TfidfVectorizer de Scikit-learn pour représenter nos documents à l’aide des scores TF-IDF calculés en fonction de leur contenu. Un TfidfVectorizer transforme une collection de documents bruts en une matrice de fonctionnalités TF-IDF. Le TF-IDF est une méthode d’analyse qui peut être utilisée dans une stratégie de référencement pour déterminer les mots-clés et les termes qui augmentent la pertinence des textes publiés.

Toujours de la même bibliothèque, nous allons utiliser différents algorithmes d’apprentissage automatique en se servant des données d’entraînement issues du dataset. Il permet de pouvoir construire des modèles de prédiction par rapport au modèle d’entraînement des données. Les algorithmes utilisés sont les suivants :

- La régression logistique : avec une précision de 91.55%.

```
from sklearn.linear_model import LogisticRegression
```

- L’algorithme basé sur l’arbre de décision : avec une précision de 91.55%.

```
from sklearn import tree
```

- Le gradient boosting machine : avec une précision de 81.22%.

```
from sklearn.ensemble import GradientBoostingClassifier
```

- La forêt d’arbres décisionnels : avec une précision de 90.21%.

```
from sklearn.ensemble import RandomForestClassifier
```

Étape 3 : Visualisation des résultats

Maintenant, nous pouvons détecter les “fake news” avec nos modèles prédictifs.

```
] : news = str(input())  
test_fake_news(news)
```

In a stunning election night, the Republican nominee for president, Donald Trump, secured victory after a string of wins in Florida, Ohio, Wisconsin, Iowa and Michigan all turned red. Nationally, Donald Trump won 47% of the vote to Hillary Clinton. He won 304 electoral college votes for the Republicans and 232 for the Democrats.

LR Prediction: Fake News
DT Prediction: Fake News
GB Prediction: Fake News
RF Prediction: Fake News

```
] : news = str(input())  
test_fake_news(news)
```

Donald Trump has secured the Republican nomination for US president on day two of the Republican National Convention in Cleveland, Ohio, behind Mr Trump, a day after splits in the party were evident as the convention opened. The Trump campaign also won the endorsement of the party's former vice president, Mike Pence. Tuesday's speakers focused almost exclusively on attacking Hillary Clinton, the likely Democratic nominee, who was plagiarised. Tuesday's speakers focused almost exclusively on attacking Hillary Clinton, the likely Democratic nominee, held a mock trial for Mrs Clinton as the crowd chanted "lock her up". Mr Christie and others criticised her for serving as secretary of state. An FBI investigation said she was "extremely careless" but found her actions didn't constitute a crime. The crowd disagreed as Mr Christie repeatedly yelled "guilty". He said she has "selfish, awful judgement" and was "a disaster" and elsewhere.

LR Prediction: Not A Fake News
DT Prediction: Not A Fake News
GB Prediction: Not A Fake News
RF Prediction: Not A Fake News

