Fake News Detection

Mythbusters

Outline

1. Introduction
2. Non-Functional Requirement
3. Data Source
4. Data Preprocessing
5. System Architecture
6. Planned for Further
7. Conclusion
8. Reference

Introduction

The fake news on social media and various other media is wide spreading and is a matter of serious concern due to its ability to cause a lot of social and national damage with destructive impacts. A lot of research is already focused on detecting it. In this project, we will implement Intelligent System to detect the Fake News Detection.

Non-Functional Requirement

1. **Efficiency:** deploying a reliable classification model
2. **Maintainability:** assuring the continuous functioning of the platform
3. **Ergonomy:** creating an efficient front-end able to deal with a large range of queries
4. **Security:** ensuring that the platform can’t be deflected by someone willing to decredibilize real information or propagate fake news
5. **Ethics:** ensuring that the errors made by the platform will not falsely harm the reputation of a journalist or politician or wrongly support and spread manipulated information

Data sources

- Labeled batch data
  - Fakeddit -> classification of 1M reddit posts
  - LIAR -> classification of 12,8K social media statements
  - ISOT -> classification of 40K articles
- Streaming data
  - Twitter
  - Reddit

Data Preprocessing (PySpark) Con’t

1. Text Cleaning
    a. Handle Missing Value
    b. Convert Text into Lowercase letter
    c. Remove Punctuation from Text
    d. Remove Urls from Text
    e. Remove @tags
    f. Remove Special Characters
2. Preprocess Operations
    a. Tokenization
    b. Removing Stop Words

Data Preprocessing (PySpark)

3. Feature Extraction
    a. CountVectorizer
    b. TF-IDF Model
4. Split the Dataset
    c. Training Data (80%)
    d. Test Data (20%)

System Architecture

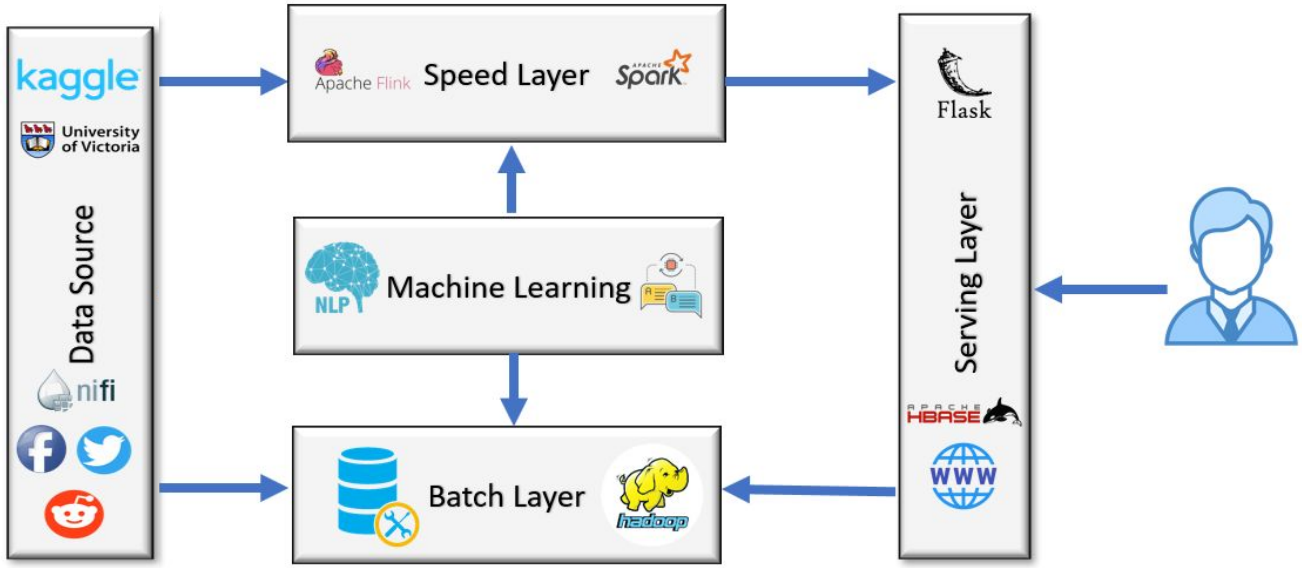


Figure 1: Lambda Architecture

Technical platform

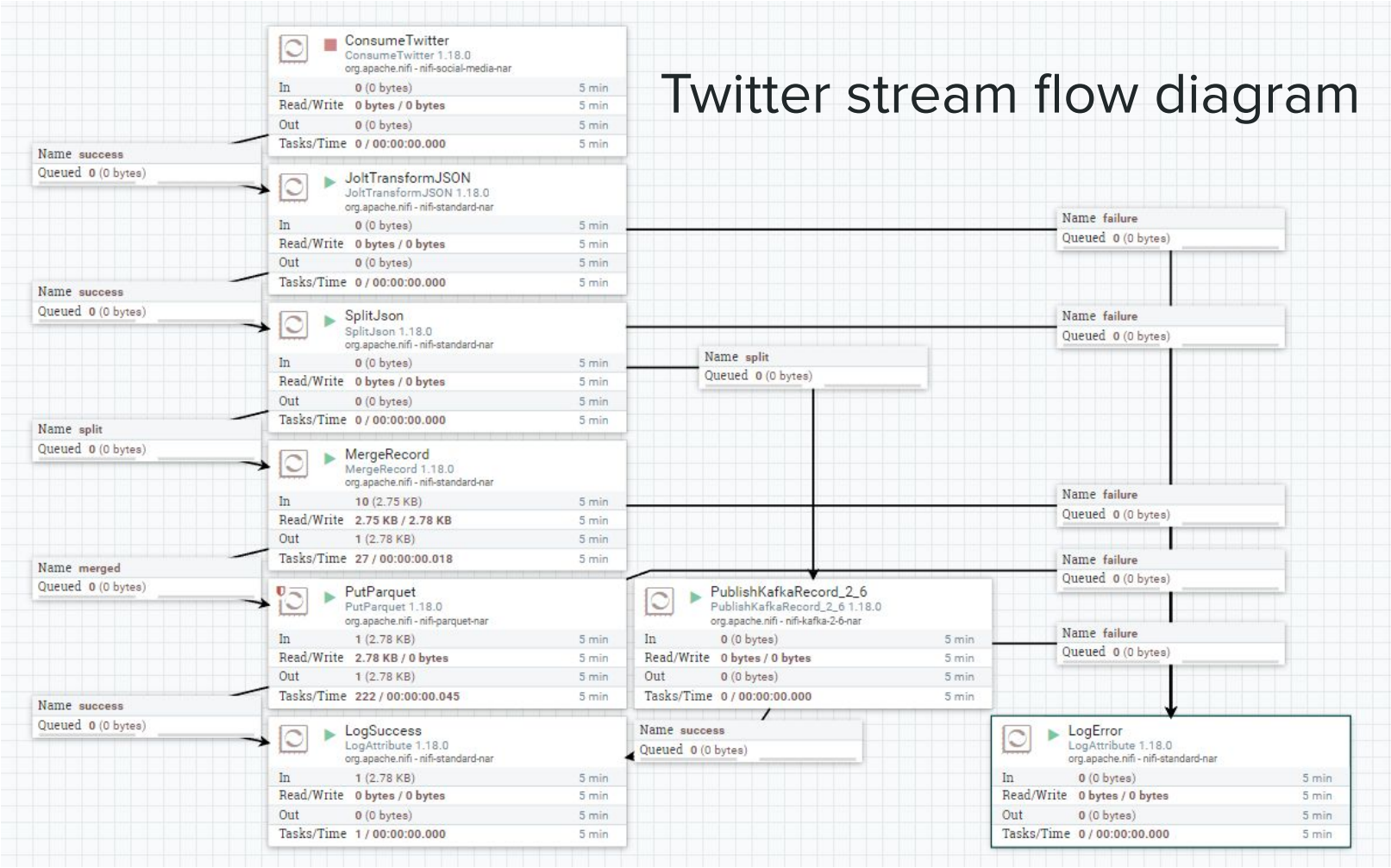
- Hardware Solution : access to a remote private server
- Software tools:
  - Hadoop 3.3.4
  - Hive 3.1.2
  - NiFi 1.18
  - Kafka 2.6
  - Twitter API
  - Reddit API

Stream Ingestion

Extraction of data from Twitter to nifi:

We use the ConsumeTwitter processor to extract stream of tweets filtered by the hashtag “#news”. The stream is then stripped of redundant parts, and split into individual tweets. The flow then splits, publishing each individual tweet separately a kafka queue for the streaming layer, as well as merging the tweets into batches and inserting into a parquet file for the batch layer. Retries with backoff are used where applicable, and all other errors are logged.

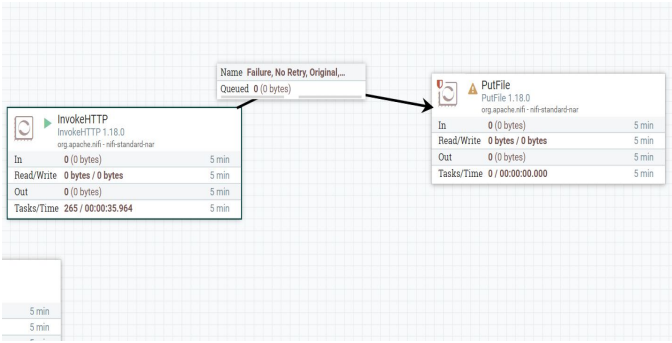
Twitter stream flow diagram

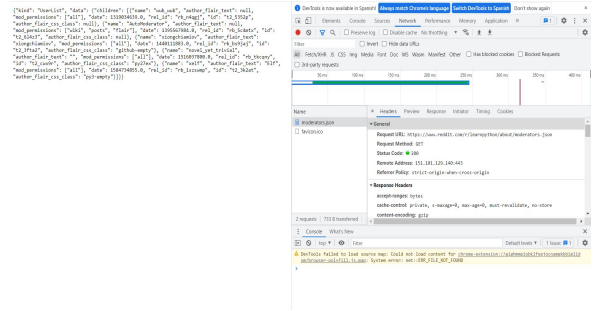


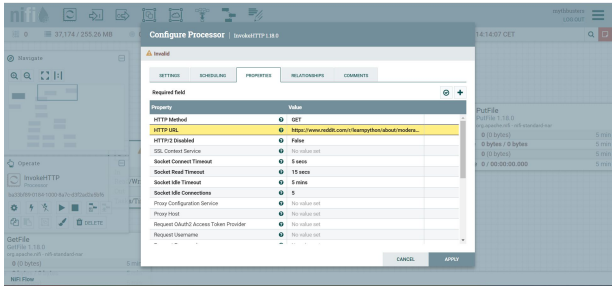
Stream Ingestion

Extraction of data from Reddit API to nifi:

Put two processors, InvokeHTTP and PutFile in nifi. Connect them from InvokeHTTP to PutFile and specify some things of their configuration. Access the Reddit API and specify the URL to the InvokeHTTP processor, and export it certificate using Java keystore. Finally specify other things of the configuration of the processors.



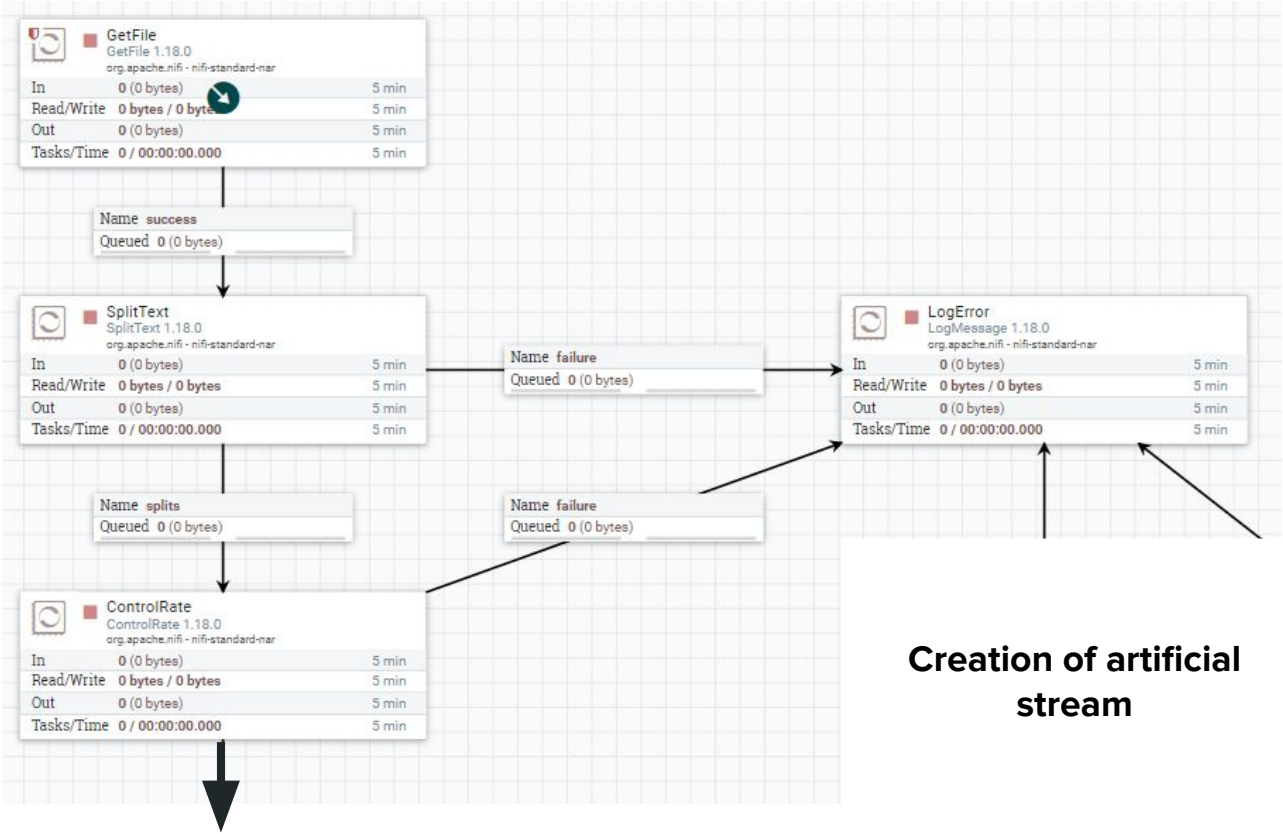




Batch Ingestion

- Conversion of batch data into an artificial stream
  - Labeled data only in batch form
  - Daily/weekly update of the model
- Conversion of each arriving chunk into parquet on HDFS
- Insertion of the new data into Hive

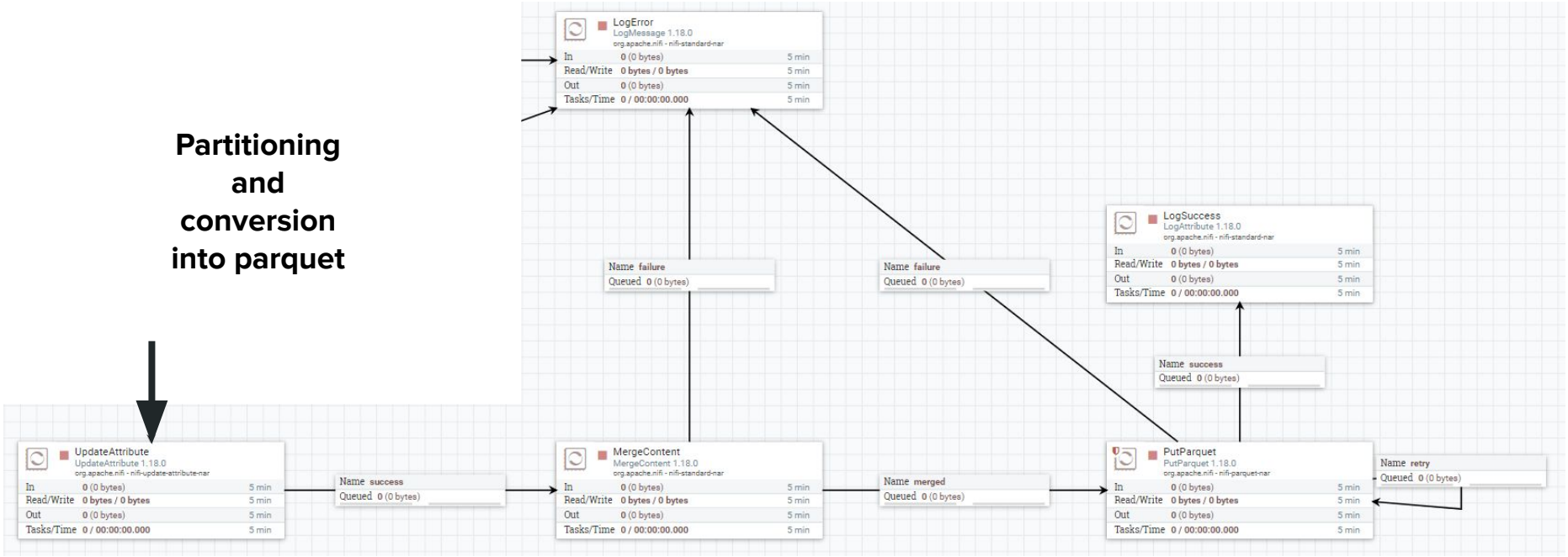
Batch Ingestion



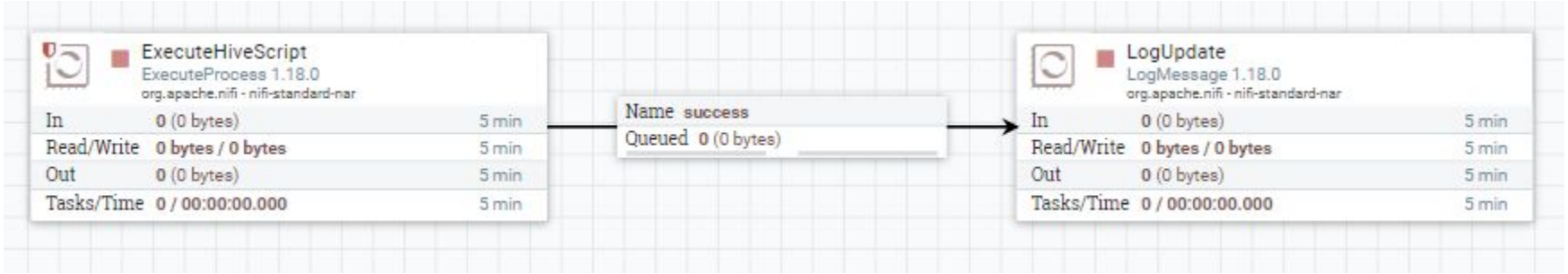
Creation of artificial stream

Batch Ingestion

Partitioning and conversion into parquet



Batch Ingestion



Update of the metadata of the external Hive table

Plan for next

- Exploratory Data Analysis
  - Matplotlib
  - Seaborn
- Binary Classification Problem
  - Support Vector Machine
  - Multilayer Perceptrons (mlps)
- Result Evaluation
  - Confusion Matrix
  - Accuracy and Precision
- Front End Application
  - Python Flask Framework

Questions