News Classifier Project Report

Objective:

To train machine learning model and classify upcoming news on the fly with good accuracy by building end to end machine learning pipeline.

To make containerized application which is scalable, robust, fault tolerant.

Planning:

We are using agile methodology to build the project. Task level details are mentioned in below gantt chart.

#Sprint: 4

#People: 2

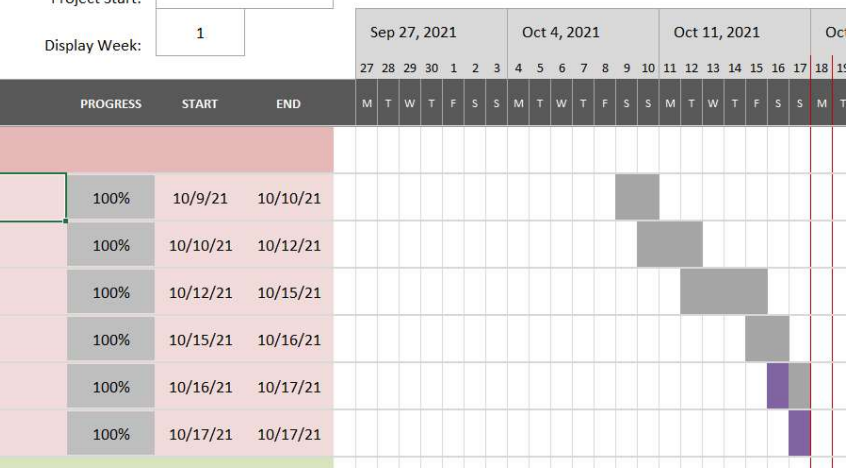
Project members:

Biren Patel, Vaishali khurana

Project Start: Tue, 9/28/2021

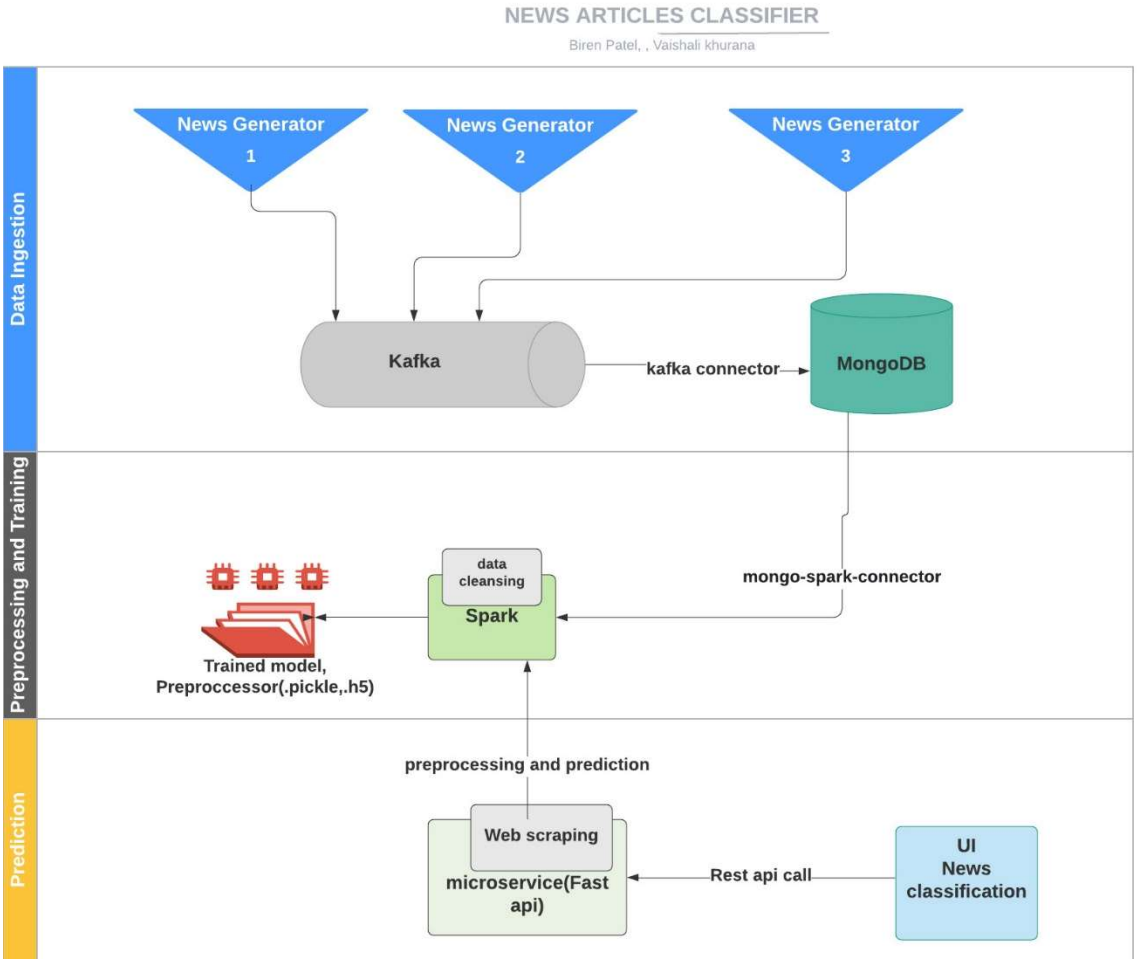
Display Week: 1

TASK	PROGRESS	START	END
Milestone 2: Data Cleansing			
Milestone planning	100%	10/9/21	10/10/21
Dockerization of data ingestion	100%	10/10/21	10/12/21
Spark mongodb integration	100%	10/12/21	10/15/21
Data cleansing using pyspark	100%	10/15/21	10/16/21
Dockerization of data cleansing	100%	10/16/21	10/17/21
Milestone report	100%	10/17/21	10/17/21



News Classifier Project Report

Architecture:



Components and description:

Component	Description
Data Ingestion	Multithreaded service that is collecting the data from web using rapid api and custom news generator, passing it to kafka queue. Kafka consumer is used to sync the data between kafka and database. Finally, data is dumped into MongoDb database
Preprocessor and trainer	Reading data from mongo db to spark session using mongo-spark-connector. As a part of feature selection we are using category and summary columns.

	Data cleansing including stop word removal, tokenization, tf-idf vectorization using pyspark. This cleaned data will be used for training.
Prediction	Scraping the weblink provided by user in UI, clean it and predict the news using rest api.

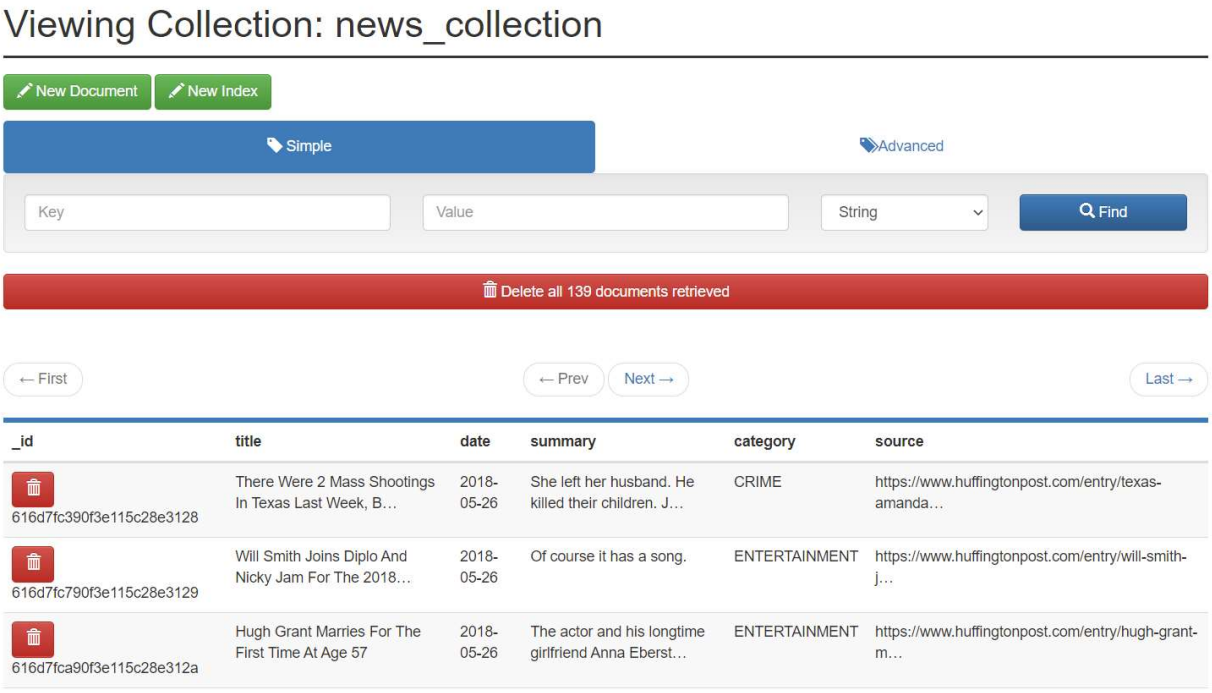
Since data cleansing is the second milestone we have explained it in detail here.

Data Cleansing:

- **Environment details:**

  Docker environment with kafka broker running on 9092 port, zookeeper running on 2181, mongo-db running on 27017, mongo-express running on 8082, spark-master running on 8080, spark-worker running on 8081, producer is connecting to kafka and producing news records to kafka queue, consumer is consuming news records from kafka and dumping to mongo db, preprocessor is cleansing the news data on spark node.

- **What goes in as an input:**
  News records saved in news_collection under news database MongoDb



News Classifier Project Report

- **How the input is being processed:**
  Records from mongo db collection is processed using pyspark over spark node which is connected through mongo_spark_connector.

  As a part of feature selection we are using category and summary columns.

  Data cleansing including stop word removal, tokenization, tf-idf vectorization using pyspark

```
root
 |-- _id: struct (nullable = true)
 |    |-- oid: string (nullable = true)
 |-- category: string (nullable = true)
 |-- date: string (nullable = true)
 |-- source: string (nullable = true)
 |-- summary: string (nullable = true)
 |-- title: string (nullable = true)

cleaning start
['_id', 'category', 'date', 'source', 'summary', 'title']
```

After feature selection and tf-idf vectorization:

```
root
 |-- summary: string (nullable = true)
 |-- category: string (nullable = true)
 |-- tf: vector (nullable = true)
 |-- idf: vector (nullable = true)
 |-- label: double (nullable = false)
```

- **What comes out as an output:** Processed and cleaned data, saved the preprocessed pipeline

```
2021-10-18 14:27:01 INFO  DAGScheduler:54 - Job 3 finished: showString at NativeMethodAccessorImpl.java:0, took 0.050267 s
+-------------+--------------------+--------------------+--------------------+-----+
|     category|             summary|                  tf|                 idf|label|
+-------------+--------------------+--------------------+--------------------+-----+
|        CRIME|She left her husb...|(10000,[1662,3562...|(10000,[1662,3562...|  7.0|
|ENTERTAINMENT|Of course it has ...|(10000,[1916,2460...|(10000,[1916,2460...|  1.0|
|ENTERTAINMENT|The actor and his...|(10000,[512,2410,...|(10000,[512,2410,...|  1.0|
|ENTERTAINMENT|The actor gives D...|(10000,[512,1384,...|(10000,[512,1384,...|  1.0|
|ENTERTAINMENT|The "Dietland" ac...|(10000,[2678,3624...|(10000,[2678,3624...|  1.0|
+-------------+--------------------+--------------------+--------------------+-----+
only showing top 5 rows
```

- **Tools/libraries used**: Docker, Pycharm, kafka, MongoDB, zookeeper, Spark, kafka, pyspark, MongoExpress

Challenges encountered and how we tackled them:

We faced below mentioned challenges. However, we have resolved them.

- Connecting MongoDb and spark
  - Tried different ways of connecting spark to mongodb then implemented it with mongo-spark-connector
- Docker networking challenges
  - Had to explore different docker-compose versions and related configuration for mac-os and windows.
- Authorization issue in MongoDb connection
  - Configured appropriate parameters to resolve it
- We switched to Mongodb to support auto scalability in future for large volume of text and unstructured data.
  - Developed consumer service to connect kafka broker to mongoDb

Future Scope:

- Further scale optimizations
- Implementing re-training mechanism using feedback feature
- Once we have large volume of labeled data, we will train our own model in place of transfer learning
- Adding multi language support

Github link:

https://github.com/biren162/Capstone/tree/master/Milestone-2