**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.

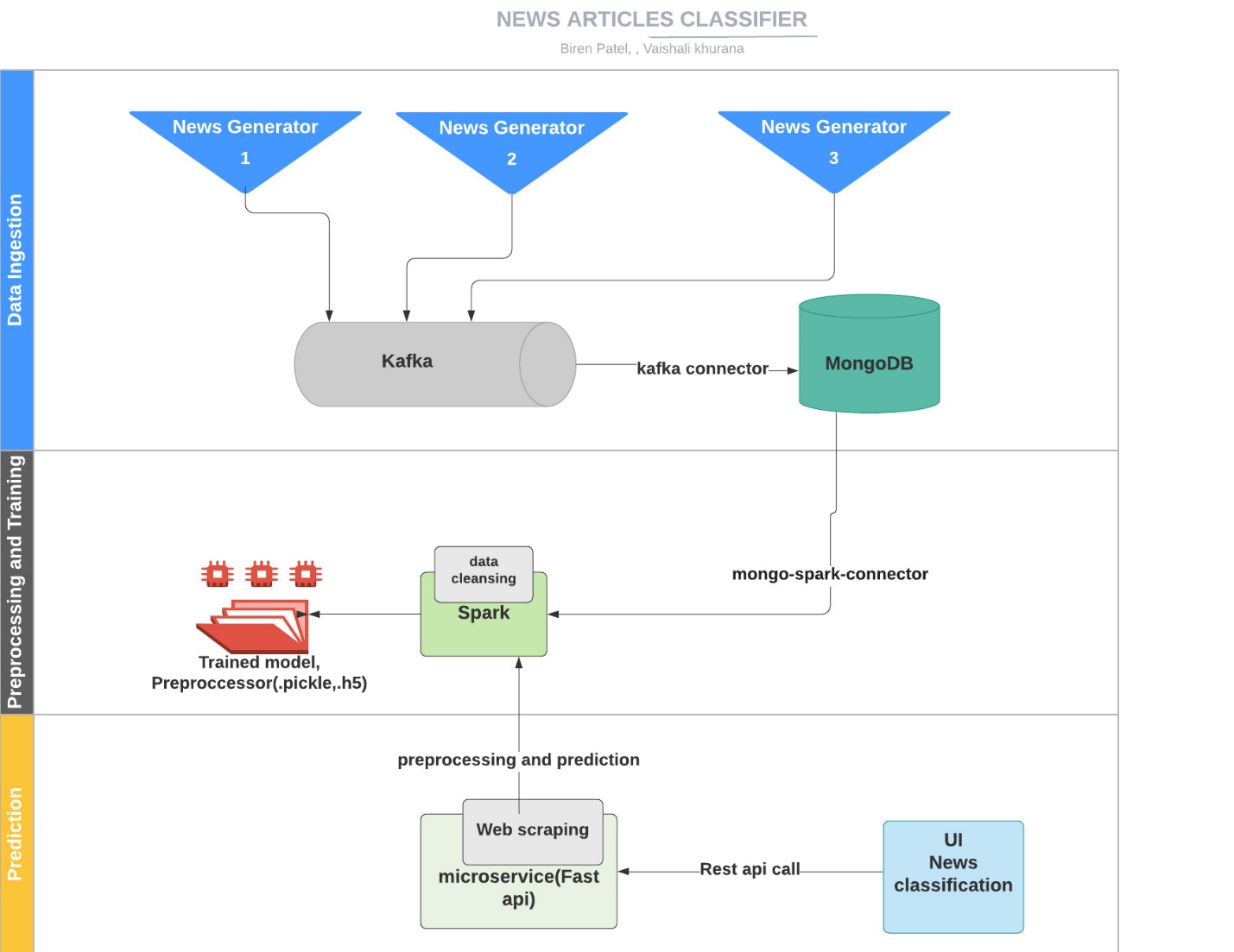
#Sprint: 4

#People: 2

**Graphical user interface

Description automatically generated with medium confidence**

**Architecture:**

****

Bert

|  |
| --- |
|  |

**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 is used for training bert model and achieved training accuracy of 87.8% and test accuracy of 82.9%. |
| Prediction | Scraping the weblink provided by user in UI, clean it and predict the news using rest api. |

Since model training is the third milestone we have explained it in detail here.

**Model Training:**

* **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\_trainer is cleansing the news data and training machine learning model on spark node using pyspark.

Hardware Details: NVIDIA GPU

Tensorflow: 2.0.0

Pretrained Bert model(Transformer): uncased\_L-8\_H-512\_A-8

Diagram

Description automatically generated

* **What goes in as an input:**

Spark dataframe having cleaned data.

Chart, bar chart

Description automatically generated

* **How the input is being processed:**

After the records from mongo db collection is processed and cleaned, these records are further used for training using pyspark over spark node which is connected through mongo\_spark\_connector.

Input data is divided into 80% training and 20% test data.

Pretrained Bert model(uncased\_L-8\_H-512\_A-8) is also used for training for 15 epochs with training accuracy of 87.8% and test accuracy of 82.9%, with following parameters:

**Table

Description automatically generated**

* **What comes out as an output:** The trained model and tokenizer is saved in registry.

Chart, line chart

Description automatically generated

Chart, line chart

Description automatically generated

**Table

Description automatically generated**

* **Tools/libraries used**: Docker, Pycharm, kafka, MongoDB, zookeeper, Spark, pyspark, MongoExpress, colab notebook, Tensorflow, seaborn, mlflow.

**Challenges encountered and how we tackled them:**

We faced below mentioned challenges:

* Loading model from HDFS
  + Configured HDFS in spark node.
* Python-Tensorflow version compatibility
  + Installed compatible libraries versions
* Configuring mlflow with docker container

**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