News Classifier Project Report

Objective:

To train machine learning or deep 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.

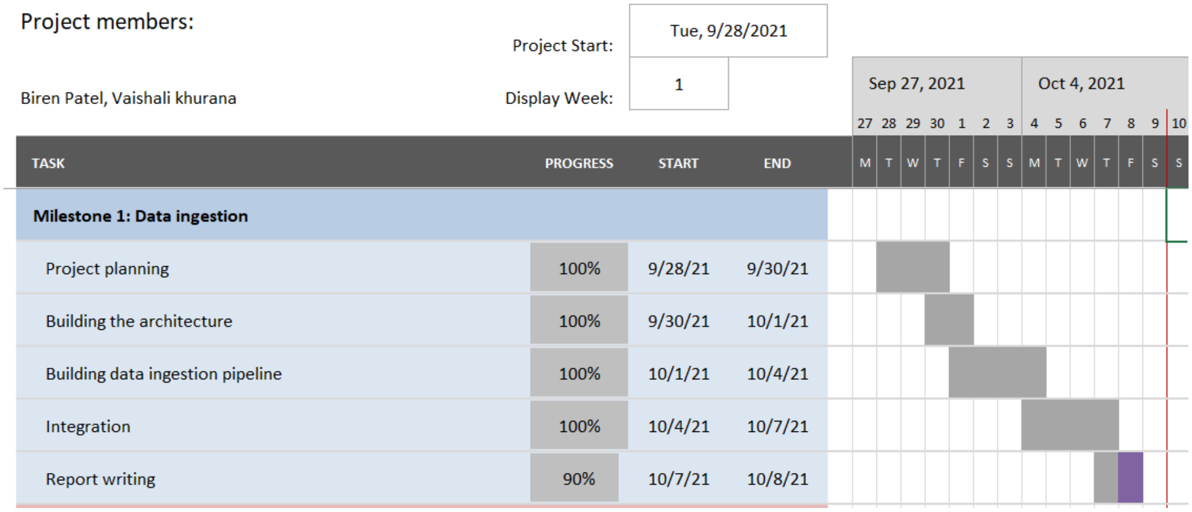
Planning:

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

#Sprint: 4

#People: 2

Sprint:1



Project members:

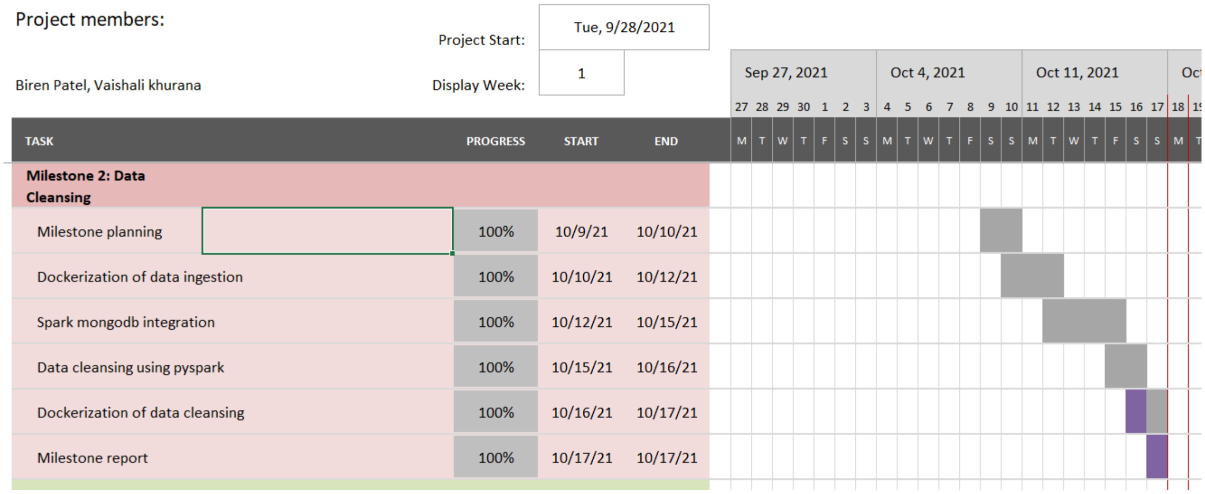
Biren Patel, Vaishali khurana

Project Start: Tue, 9/28/2021

Display Week: 1

TASK	PROGRESS	START	END
Milestone 1: Data ingestion			
Project planning	100%	9/28/21	9/30/21
Building the architecture	100%	9/30/21	10/1/21
Building data ingestion pipeline	100%	10/1/21	10/4/21
Integration	100%	10/4/21	10/7/21
Report writing	90%	10/7/21	10/8/21

Sprint:2



Project members:

Biren Patel, Vaishali khurana

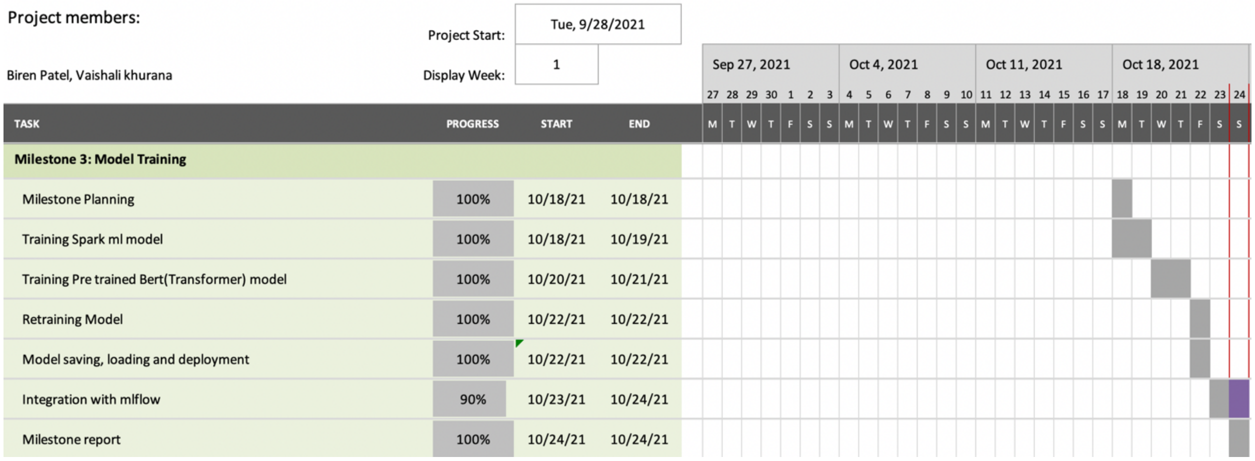
Project Start: Tue, 9/28/2021

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

Sprint:3



Project members:

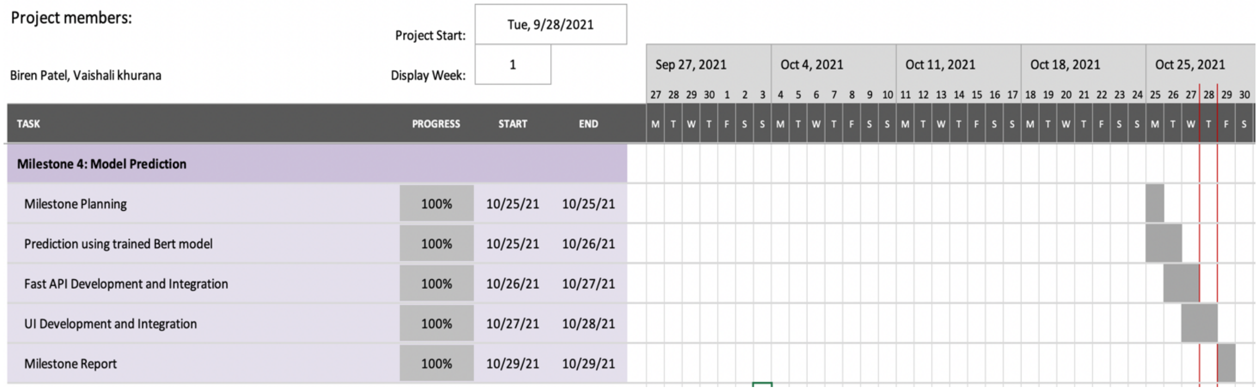
Biren Patel, Vaishali khurana

Project Start: Tue, 9/28/2021

Display Week: 1

TASK	PROGRESS	START	END
Milestone 3: Model Training			
Milestone Planning	100%	10/18/21	10/18/21
Training Spark ml model	100%	10/18/21	10/19/21
Training Pre trained Bert(Transformer) model	100%	10/20/21	10/21/21
Retraining Model	100%	10/22/21	10/22/21
Model saving, loading and deployment	100%	10/22/21	10/22/21
Integration with mlflow	90%	10/23/21	10/24/21
Milestone report	100%	10/24/21	10/24/21

Sprint:4



Project members:

Biren Patel, Vaishali khurana

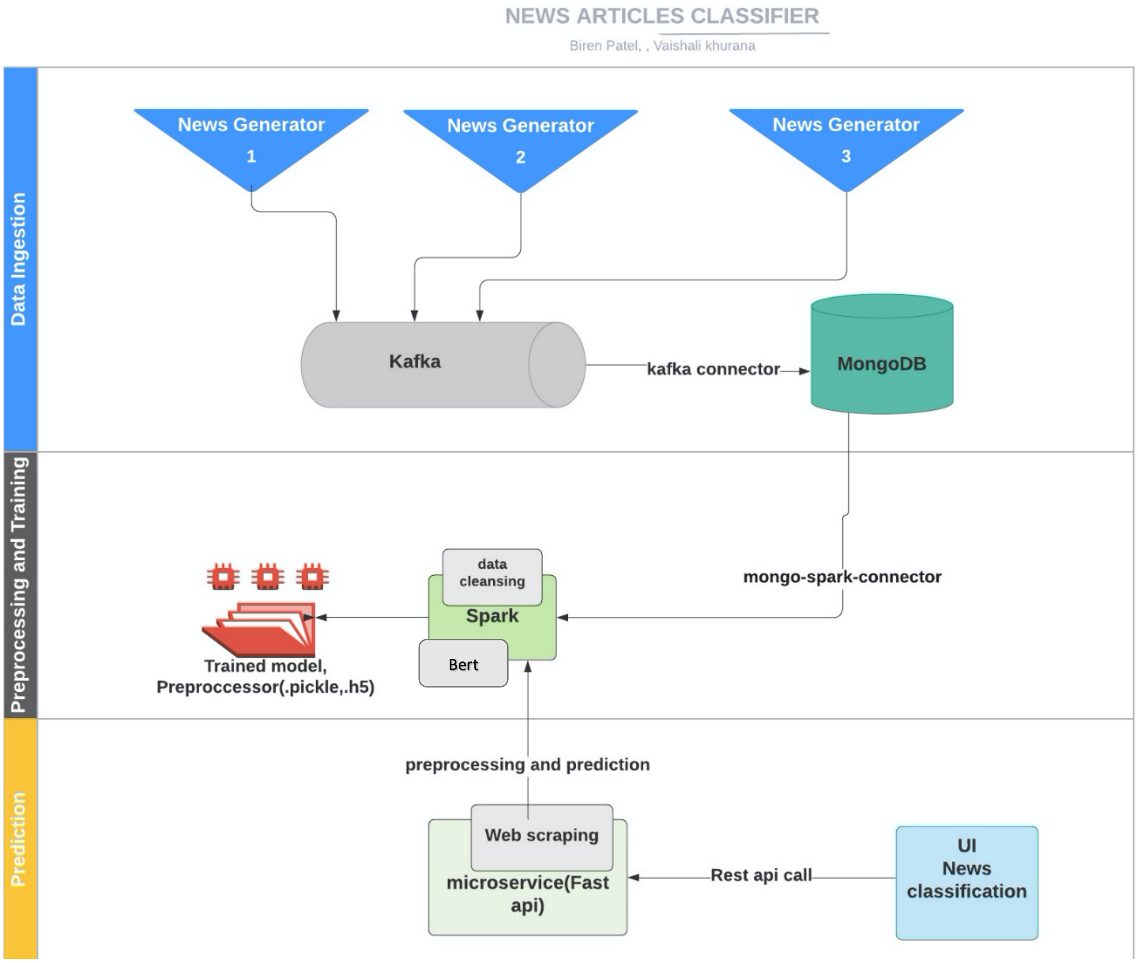
Project Start: Tue, 9/28/2021

Display Week: 1

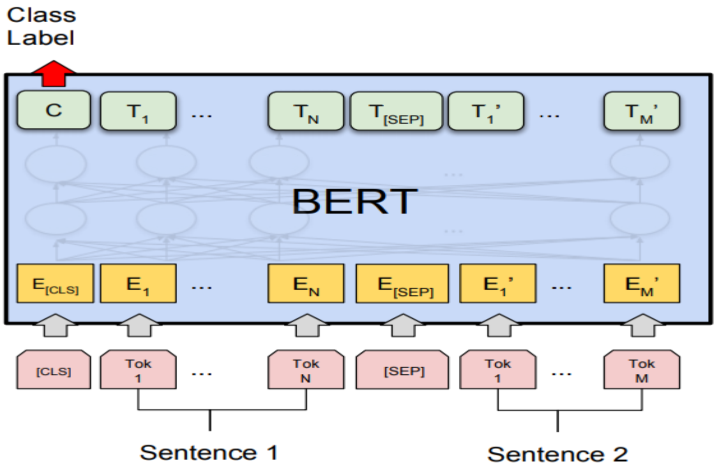
TASK	PROGRESS	START	END
Milestone 4: Model Prediction			
Milestone Planning	100%	10/25/21	10/25/21
Prediction using trained Bert model	100%	10/25/21	10/26/21
Fast API Development and Integration	100%	10/26/21	10/27/21
UI Development and Integration	100%	10/27/21	10/28/21
Milestone Report	100%	10/29/21	10/29/21

News Classifier Project Report

Architecture:



Bert model (Transformer) Architecture:

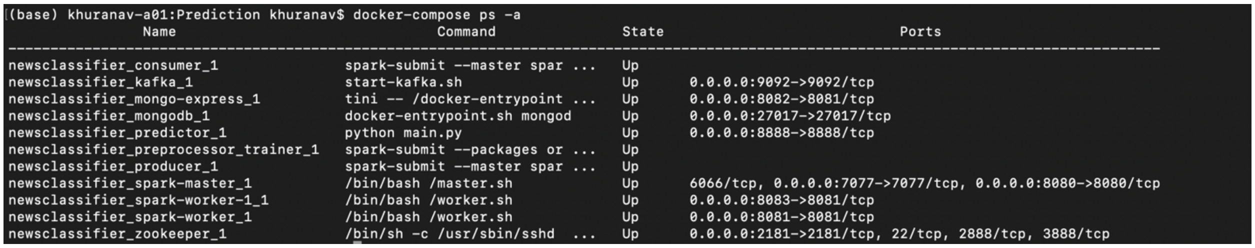


Components and Description:

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 is used for training bert model and achieved training accuracy of 87.8% and test accuracy of 82.9%.
Predictor and UI	Predicting news category based on the news summary entered by user in UI. Clicking "Predict" button calls backend rest API which firstly processes the data using the trained tokenizer and then predict using the trained Bert model.

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, predictor service running on 8888 port.
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 and predictor service has UI integration with Fast API in backend.



```
(base) khuranav-a01:Prediction khuranav$ docker-compose ps -a
                Name                              Command               State                          Ports
---------------------------------------------------------------------------------------------------------------------------------------
newsclassifier_consumer_1             spark-submit --master spar ...   Up
newsclassifier_kafka_1                start-kafka.sh                   Up      0.0.0.0:9092->9092/tcp
newsclassifier_mongo-express_1        tini -- /docker-entrypoint ...   Up      0.0.0.0:8082->8081/tcp
newsclassifier_mongodb_1              docker-entrypoint.sh mongod      Up      0.0.0.0:27017->27017/tcp
newsclassifier_predictor_1            python main.py                   Up      0.0.0.0:8888->8888/tcp
newsclassifier_preprocessor_trainer_1   spark-submit --packages or ...   Up
newsclassifier_producer_1             spark-submit --master spar ...   Up
newsclassifier_spark-master_1         /bin/bash /master.sh             Up      6066/tcp, 0.0.0.0:7077->7077/tcp, 0.0.0.0:8080->8080/tcp
newsclassifier_spark-worker-1_1       /bin/bash /worker.sh             Up      0.0.0.0:8083->8081/tcp
newsclassifier_spark-worker_1         /bin/bash /worker.sh             Up      0.0.0.0:8081->8081/tcp
newsclassifier_zookeeper_1            /bin/sh -c /usr/sbin/sshd  ...   Up      0.0.0.0:2181->2181/tcp, 22/tcp, 2888/tcp, 3888/tcp
```

News Classifier Project Report

- **Hardware Details:**
  - NVIDIA GPU
  - Memory requirements: Minimum 4GB RAM dedicated for docker

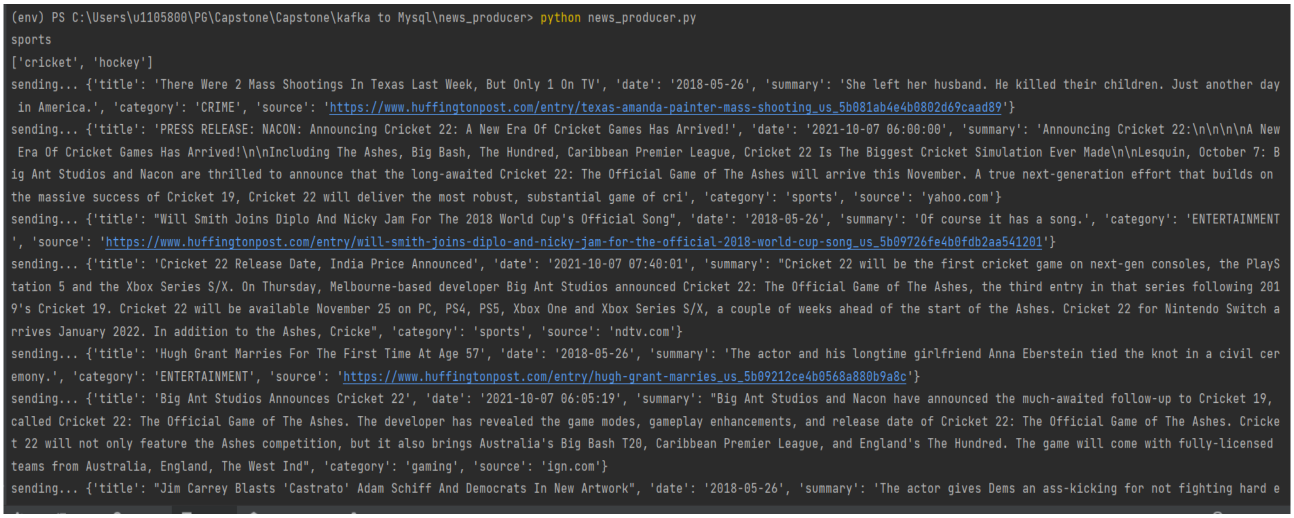
- **Tools/libraries used**:
  - Docker
  - Pycharm
  - Kafka
  - Zookeeper
  - MongoDB
  - MongoExpress
  - Spark
  - Pyspark
  - colab notebook
  - Tensorflow : 2.0.0
  - Seaborn
  - mlflow
  - Pretrained Bert model(Transformer): uncased_L-8_H-512_A-8
  - Fast API
  - HTML
  - CSS
  - Jinja2Templates

- **Coding Standards:**
  - PEP-8 Coding Standards

1. Data Ingestion:

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

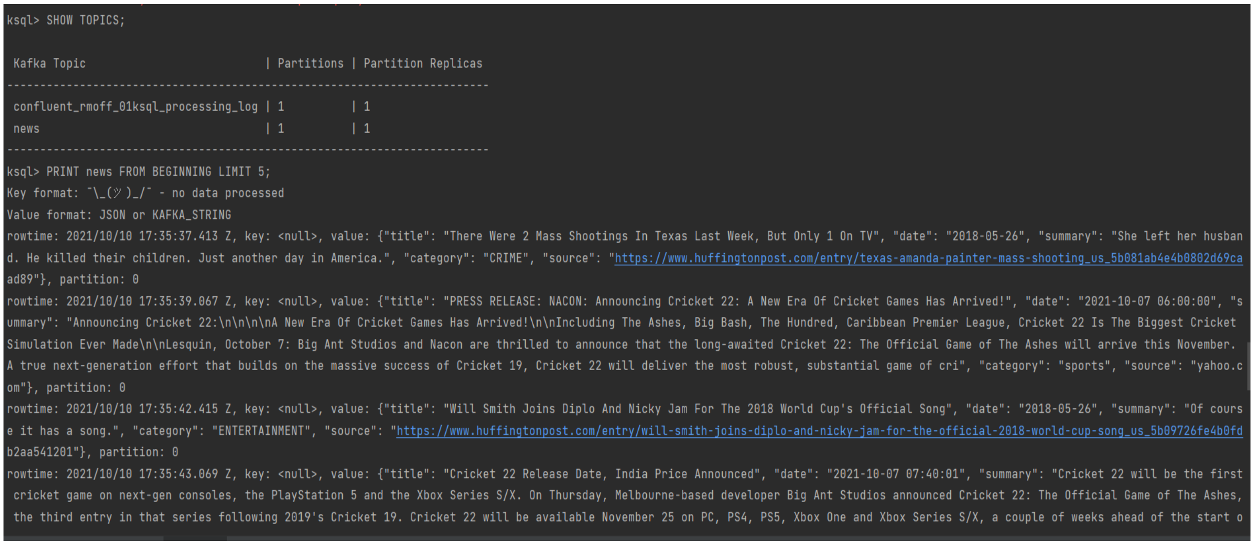
  We have used rapid api and custom news generator as data sources.



```
(env) PS C:\Users\u1105800\PG\Capstone\Capstone\kafka to Mysql\news_producer> python news_producer.py
sports
['cricket', 'hockey']
sending... {'title': 'There Were 2 Mass Shootings In Texas Last Week, But Only 1 On TV', 'date': '2018-05-26', 'summary': 'She left her husband. He killed their children. Just another day
 in America.', 'category': 'CRIME', 'source': 'https://www.huffingtonpost.com/entry/texas-amanda-painter-mass-shooting_us_5b081ab4e4b0802d69caad89'}
sending... {'title': 'PRESS RELEASE: NACON: Announcing Cricket 22: A New Era Of Cricket Games Has Arrived!', 'date': '2021-10-07 06:00:00', 'summary': 'Announcing Cricket 22:\n\n\n\nA New
 Era Of Cricket Games Has Arrived!\n\nIncluding The Ashes, Big Bash, The Hundred, Caribbean Premier League, Cricket 22 Is The Biggest Cricket Simulation Ever Made\n\nLesquin, October 7: B
ig Ant Studios and Nacon are thrilled to announce that the long-awaited Cricket 22: The Official Game of The Ashes will arrive this November. A true next-generation effort that builds on
the massive success of Cricket 19, Cricket 22 will deliver the most robust, substantial game of cri', 'category': 'sports', 'source': 'yahoo.com'}
sending... {'title': "Will Smith Joins Diplo And Nicky Jam For The 2018 World Cup's Official Song", 'date': '2018-05-26', 'summary': 'Of course it has a song.', 'category': 'ENTERTAINMENT
', 'source': 'https://www.huffingtonpost.com/entry/will-smith-joins-diplo-and-nicky-jam-for-the-official-2018-world-cup-song_us_5b09726fe4b0fdb2aa541201'}
sending... {'title': 'Cricket 22 Release Date, India Price Announced', 'date': '2021-10-07 07:40:01', 'summary': "Cricket 22 will be the first cricket game on next-gen consoles, the PlayS
tation 5 and the Xbox Series S/X. On Thursday, Melbourne-based developer Big Ant Studios announced Cricket 22: The Official Game of The Ashes, the third entry in that series following 201
9's Cricket 19. Cricket 22 will be available November 25 on PC, PS4, PS5, Xbox One and Xbox Series S/X, a couple of weeks ahead of the start of the Ashes. Cricket 22 for Nintendo Switch a
rrives January 2022. In addition to the Ashes, Cricke", 'category': 'sports', 'source': 'ndtv.com'}
sending... {'title': 'Hugh Grant Marries For The First Time At Age 57', 'date': '2018-05-26', 'summary': 'The actor and his longtime girlfriend Anna Eberstein tied the knot in a civil cer
emony.', 'category': 'ENTERTAINMENT', 'source': 'https://www.huffingtonpost.com/entry/hugh-grant-marries_us_5b09212ce4b0568a880b9a8c'}
sending... {'title': 'Big Ant Studios Announces Cricket 22', 'date': '2021-10-07 06:05:19', 'summary': "Big Ant Studios and Nacon have announced the much-awaited follow-up to Cricket 19,
called Cricket 22: The Official Game of The Ashes. The developer has revealed the game modes, gameplay enhancements, and release date of Cricket 22: The Official Game of The Ashes. Cricke
t 22 will not only feature the Ashes competition, but it also brings Australia's Big Bash T20, Caribbean Premier League, and England's The Hundred. The game will come with fully-licensed
teams from Australia, England, The West Ind", 'category': 'gaming', 'source': 'ign.com'}
sending... {'title': "Jim Carrey Blasts 'Castrato' Adam Schiff And Democrats In New Artwork", 'date': '2018-05-26', 'summary': 'The actor gives Dems an ass-kicking for not fighting hard e
```

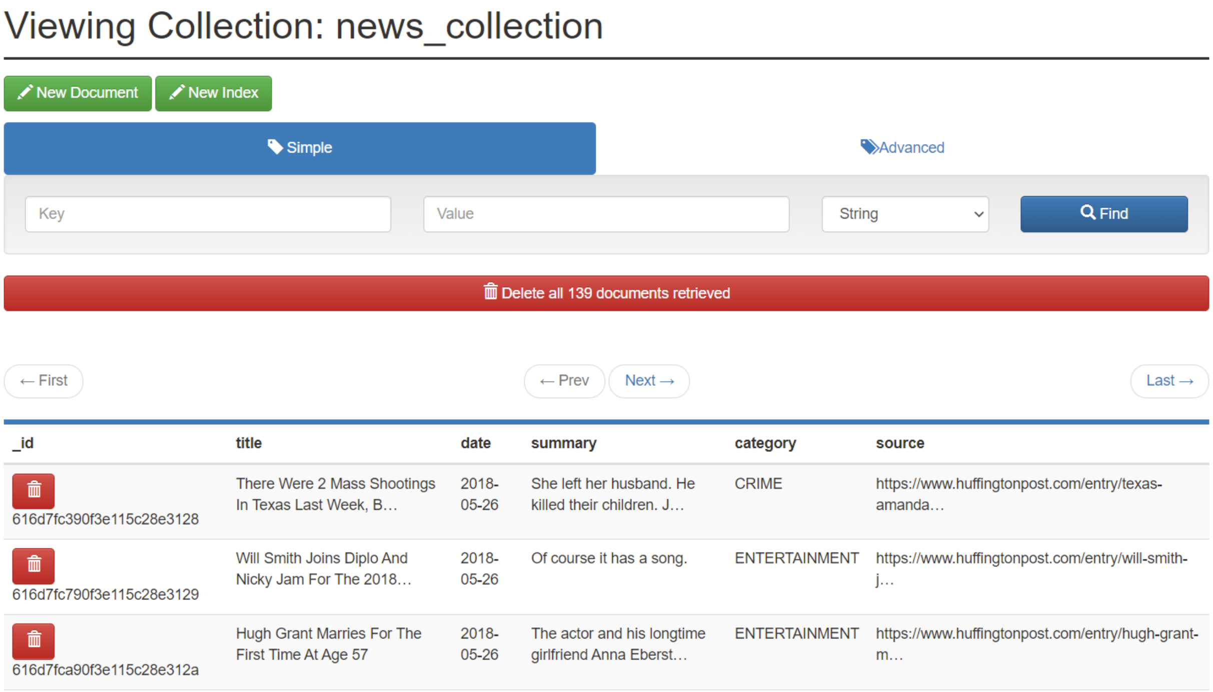
News Classifier Project Report

- **How the input is being processed:**
  Multithreaded application that is collecting the data from web using rapid api and custom news generator, passing it to kafka queue. Kafka consumer is dumping the data into mongo db.



```
ksql> SHOW TOPICS;

 Kafka Topic                        | Partitions | Partition Replicas
-----------------------------------------------------------------------
 confluent_rmoff_01ksql_processing_log | 1          | 1
 news                               | 1          | 1
-----------------------------------------------------------------------
ksql> PRINT news FROM BEGINNING LIMIT 5;
Key format: ¯\_(ツ )_/¯ - no data processed
Value format: JSON or KAFKA_STRING
rowtime: 2021/10/10 17:35:37.413 Z, key: <null>, value: {"title": "There Were 2 Mass Shootings In Texas Last Week, But Only 1 On TV", "date": "2018-05-26", "summary": "She left her husban
d. He killed their children. Just another day in America.", "category": "CRIME", "source": "https://www.huffingtonpost.com/entry/texas-amanda-painter-mass-shooting_us_5b081ab4e4b0802d69ca
ad89"}, partition: 0
rowtime: 2021/10/10 17:35:39.067 Z, key: <null>, value: {"title": "PRESS RELEASE: NACON: Announcing Cricket 22: A New Era Of Cricket Games Has Arrived!", "date": "2021-10-07 06:00:00", "s
ummary": "Announcing Cricket 22:\n\n\n\nA New Era Of Cricket Games Has Arrived!\n\nIncluding The Ashes, Big Bash, The Hundred, Caribbean Premier League, Cricket 22 Is The Biggest Cricket
Simulation Ever Made\n\nLesquin, October 7: Big Ant Studios and Nacon are thrilled to announce that the long-awaited Cricket 22: The Official Game of The Ashes will arrive this November.
A true next-generation effort that builds on the massive success of Cricket 19, Cricket 22 will deliver the most robust, substantial game of cri", "category": "sports", "source": "yahoo.c
om"}, partition: 0
rowtime: 2021/10/10 17:35:42.415 Z, key: <null>, value: {"title": "Will Smith Joins Diplo And Nicky Jam For The 2018 World Cup's Official Song", "date": "2018-05-26", "summary": "Of cours
e it has a song.", "category": "ENTERTAINMENT", "source": "https://www.huffingtonpost.com/entry/will-smith-joins-diplo-and-nicky-jam-for-the-official-2018-world-cup-song_us_5b09726fe4b0fd
b2aa541201"}, partition: 0
rowtime: 2021/10/10 17:35:43.069 Z, key: <null>, value: {"title": "Cricket 22 Release Date, India Price Announced", "date": "2021-10-07 07:40:01", "summary": "Cricket 22 will be the first
 cricket game on next-gen consoles, the PlayStation 5 and the Xbox Series S/X. On Thursday, Melbourne-based developer Big Ant Studios announced Cricket 22: The Official Game of The Ashes,
 the third entry in that series following 2019's Cricket 19. Cricket 22 will be available November 25 on PC, PS4, PS5, Xbox One and Xbox Series S/X, a couple of weeks ahead of the start o
```

- **What comes out as an output:**
  News records saved in news_collection under news database MongoDb.

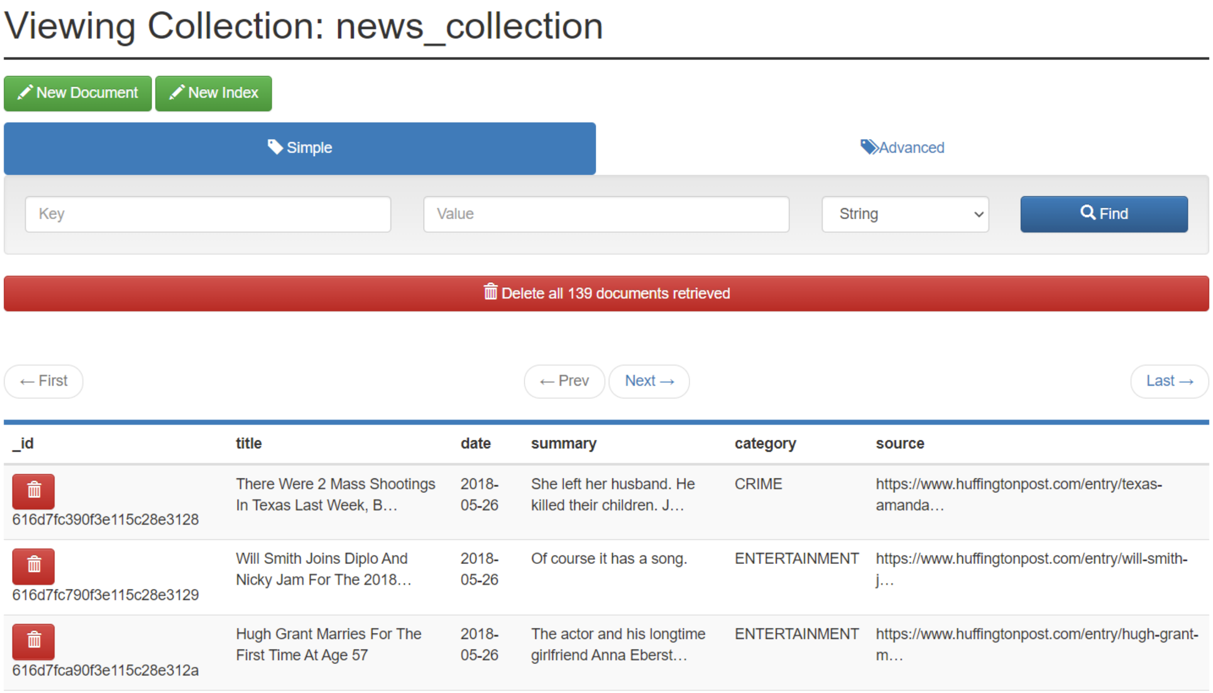


Viewing Collection: news_collection

_id	title	date	summary	category	source
616d7fc390f3e115c28e3128	There Were 2 Mass Shootings In Texas Last Week, B…	2018-05-26	She left her husband. He killed their children. J…	CRIME	https://www.huffingtonpost.com/entry/texas-amanda…
616d7fc790f3e115c28e3129	Will Smith Joins Diplo And Nicky Jam For The 2018…	2018-05-26	Of course it has a song.	ENTERTAINMENT	https://www.huffingtonpost.com/entry/will-smith-j…
616d7fca90f3e115c28e312a	Hugh Grant Marries For The First Time At Age 57	2018-05-26	The actor and his longtime girlfriend Anna Eberst…	ENTERTAINMENT	https://www.huffingtonpost.com/entry/hugh-grant-m…

2. Data Cleansing:

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



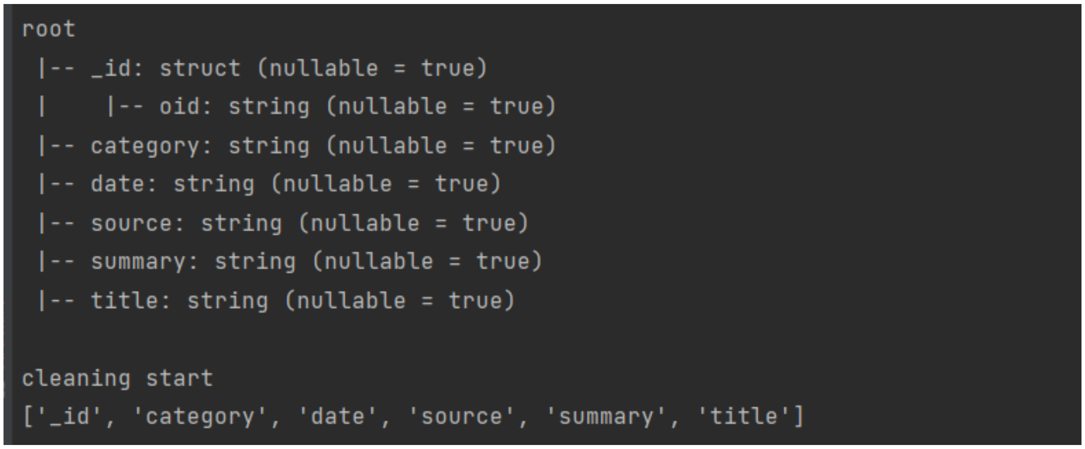
Viewing Collection: news_collection

_id	title	date	summary	category	source
616d7fc390f3e115c28e3128	There Were 2 Mass Shootings In Texas Last Week, B…	2018-05-26	She left her husband. He killed their children. J…	CRIME	https://www.huffingtonpost.com/entry/texas-amanda…
616d7fc790f3e115c28e3129	Will Smith Joins Diplo And Nicky Jam For The 2018…	2018-05-26	Of course it has a song.	ENTERTAINMENT	https://www.huffingtonpost.com/entry/will-smith-j…
616d7fca90f3e115c28e312a	Hugh Grant Marries For The First Time At Age 57	2018-05-26	The actor and his longtime girlfriend Anna Eberst…	ENTERTAINMENT	https://www.huffingtonpost.com/entry/hugh-grant-m…

- **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.

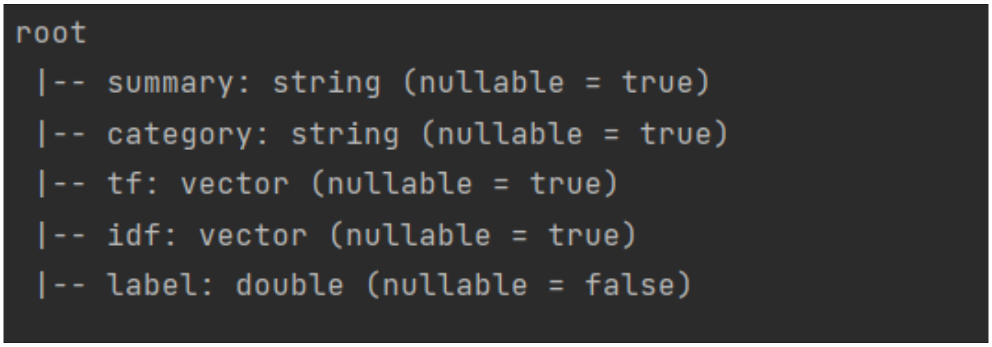
  As a part of feature selection we are using category and summary columns.

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



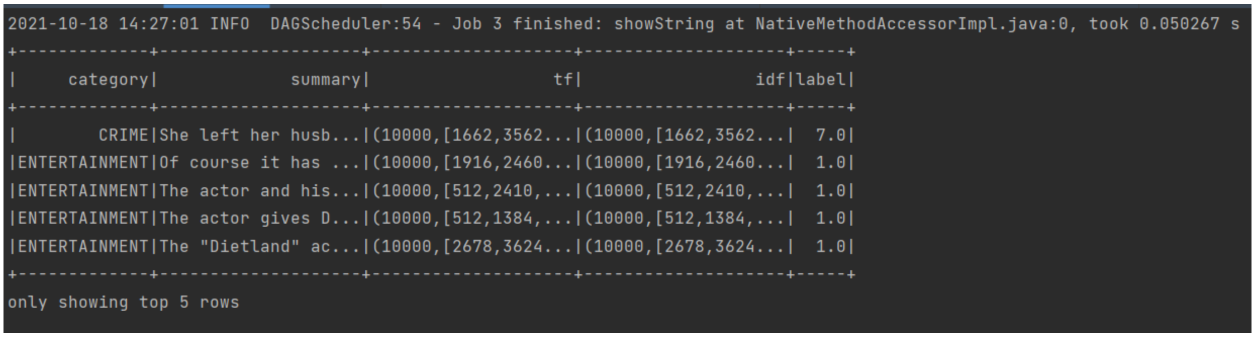
```
root
 |-- _id: struct (nullable = true)
 |    |-- oid: string (nullable = true)
 |-- category: string (nullable = true)
 |-- date: string (nullable = true)
 |-- source: string (nullable = true)
 |-- summary: string (nullable = true)
 |-- title: string (nullable = true)

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)
```

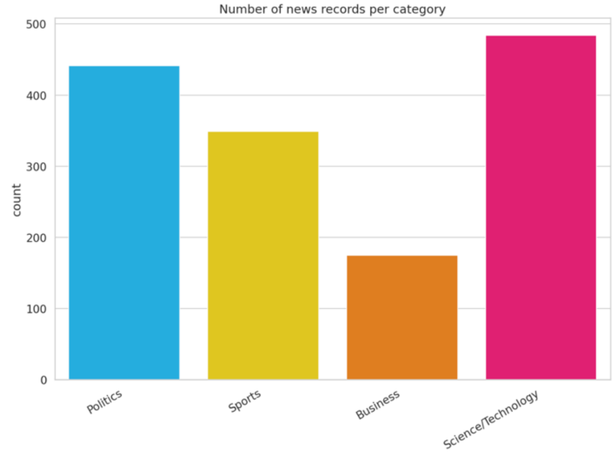
- **What comes out as an output:**
  Processed and cleaned data, saved the preprocessed pipeline



```
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
```

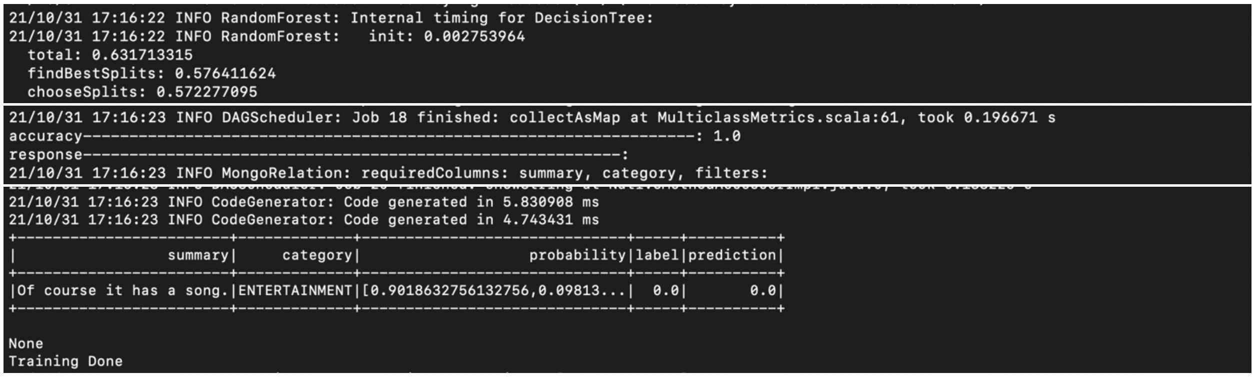
3. Model Training:

- **What goes in as an input:**
  Spark dataframe having cleaned data.



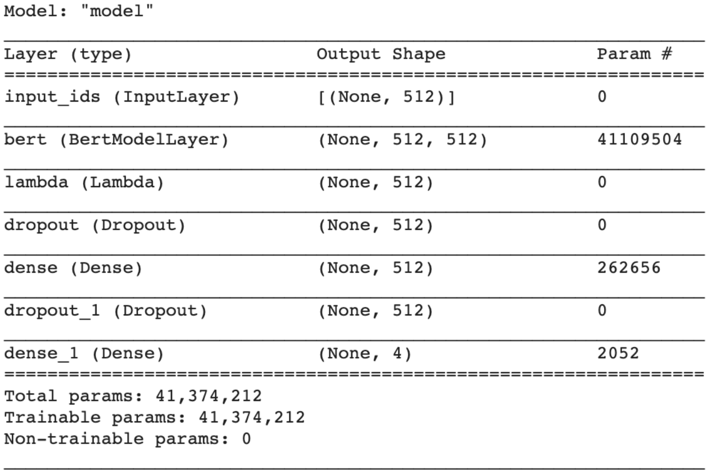
- **How the input is being processed:**

After the records from mongo db collection is processed and cleaned, these records are further used for training using pyspark random forest model over spark node which is connected through mongo_spark_connector.



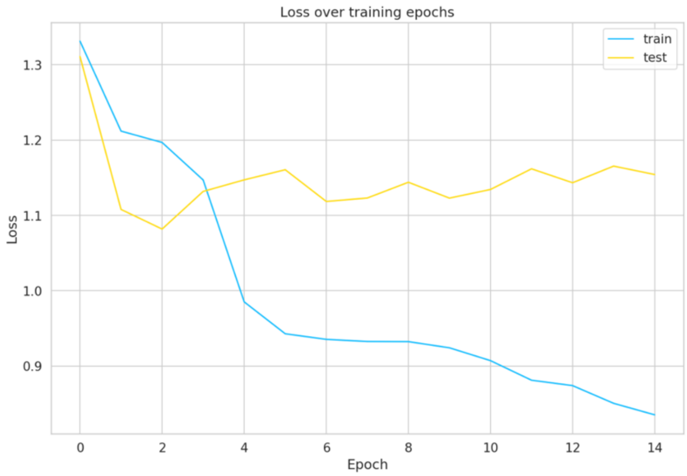
```
21/10/31 17:16:22 INFO RandomForest: Internal timing for DecisionTree:
21/10/31 17:16:22 INFO RandomForest:   init: 0.002753964
  total: 0.631713315
  findBestSplits: 0.576411624
  chooseSplits: 0.572277095
21/10/31 17:16:23 INFO DAGScheduler: Job 18 finished: collectAsMap at MulticlassMetrics.scala:61, took 0.196671 s
accuracy------------------------------------------------------------------: 1.0
response----------------------------------------------------------:
21/10/31 17:16:23 INFO MongoRelation: requiredColumns: summary, category, filters:
21/10/31 17:16:23 INFO CodeGenerator: Code generated in 5.830908 ms
21/10/31 17:16:23 INFO CodeGenerator: Code generated in 4.743431 ms
+------------------------+-------------+-------------------------------+-----+----------+
|                 summary|     category|                    probability|label|prediction|
+------------------------+-------------+-------------------------------+-----+----------+
|Of course it has a song.|ENTERTAINMENT|[0.9018632756132756,0.09813...]|  0.0|       0.0|
+------------------------+-------------+-------------------------------+-----+----------+

None
Training Done
```

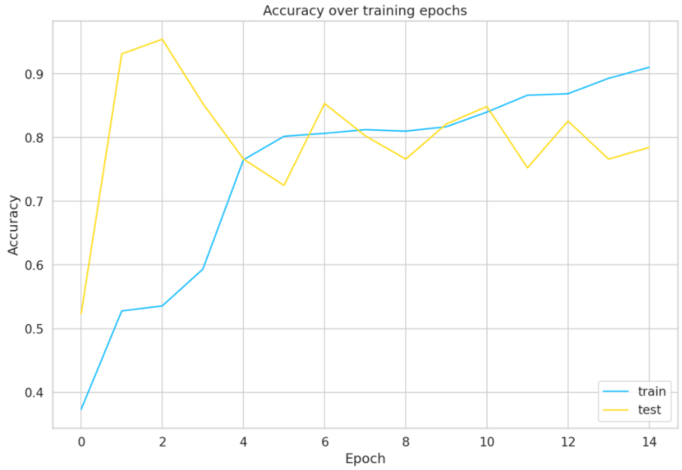
Input data is divided into 80% training and 20% test data.
Pretrained Bert model(uncased_L-8_H-512_A-8) is also used for training for 15 epochs with training accuracy of 87.8% and test accuracy of 82.9%, with following parameters:

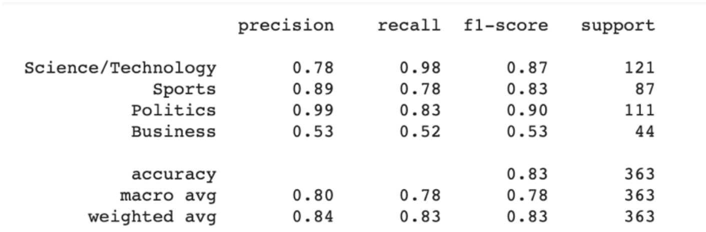


```
Model: "model"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_ids (InputLayer)       [(None, 512)]             0
_________________________________________________________________
bert (BertModelLayer)        (None, 512, 512)          41109504
_________________________________________________________________
lambda (Lambda)              (None, 512)               0
_________________________________________________________________
dropout (Dropout)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 512)               262656
_________________________________________________________________
dropout_1 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 4)                 2052
=================================================================
Total params: 41,374,212
Trainable params: 41,374,212
Non-trainable params: 0
_________________________________________________________________
```

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



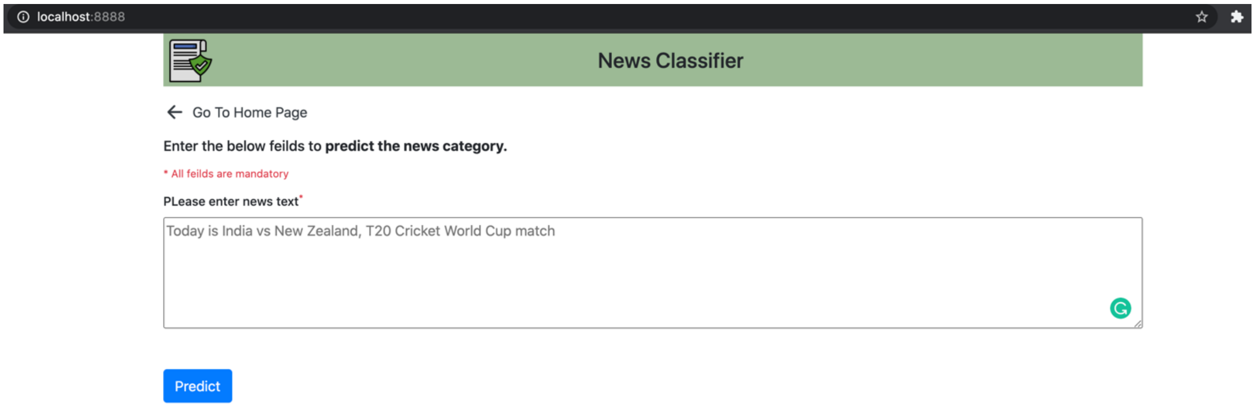




```
                    precision    recall  f1-score   support

Science/Technology       0.78      0.98      0.87       121
            Sports       0.89      0.78      0.83        87
          Politics       0.99      0.83      0.90       111
          Business       0.53      0.52      0.53        44

          accuracy                           0.83       363
         macro avg       0.80      0.78      0.78       363
      weighted avg       0.84      0.83      0.83       363
```

4. Model Prediction:

- **What goes in as an input:**
  News Summary text entered by user in UI



localhost:8888

News Classifier

← Go To Home Page

Enter the below feilds to **predict the news category.**

* All feilds are mandatory

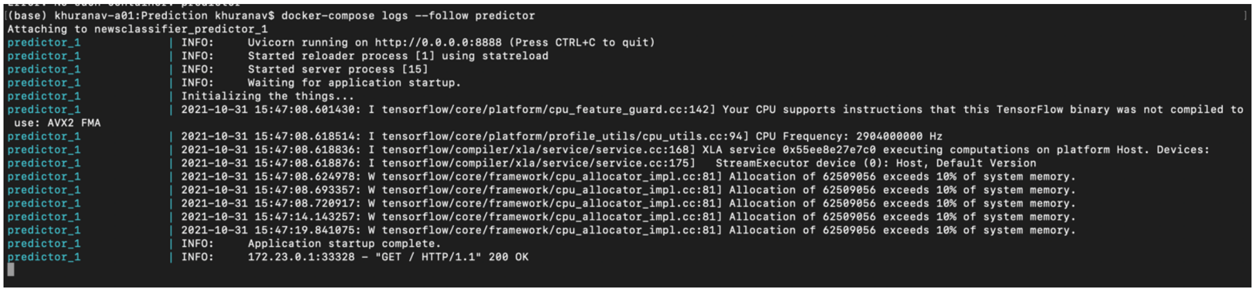
PLease enter news text*

Today is India vs New Zealand, T20 Cricket World Cup match

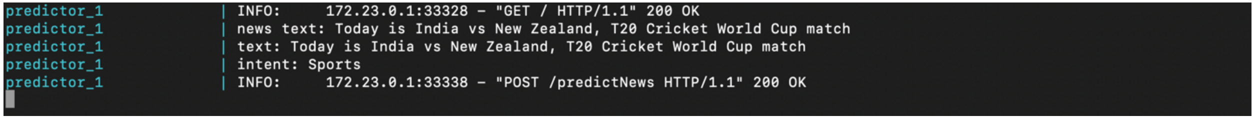
Predict

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

Clicking “Predict” button calls backend rest API which firstly processes the data using the trained tokenizer and then predict using the trained Bert model and returned response is displayed on UI.



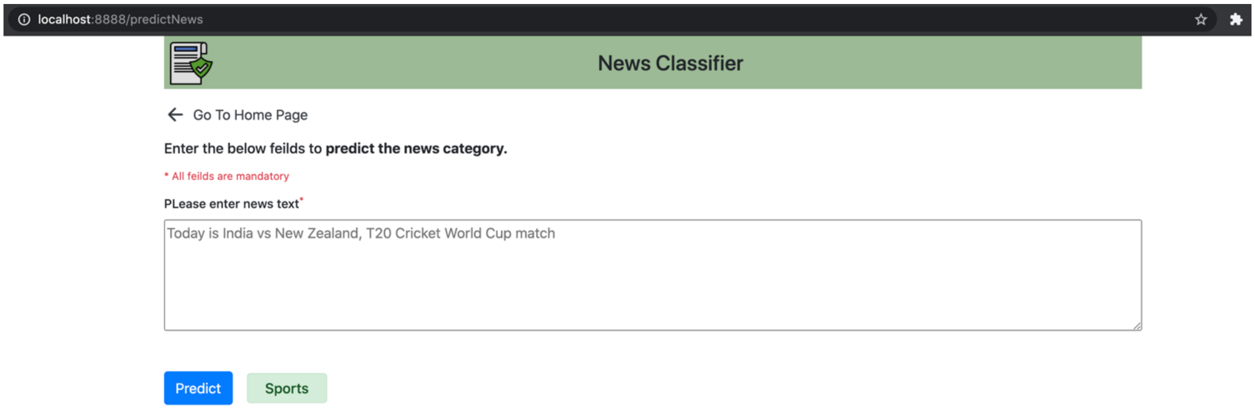
```
(base) khuranav-a01:Prediction khuranav$ docker-compose logs --follow predictor
Attaching to newsclassifier_predictor_1
predictor_1          | INFO:     Uvicorn running on http://0.0.0.0:8888 (Press CTRL+C to quit)
predictor_1          | INFO:     Started reloader process [1] using statreload
predictor_1          | INFO:     Started server process [15]
predictor_1          | INFO:     Waiting for application startup.
predictor_1          | Initializing the things...
predictor_1          | 2021-10-31 15:47:08.601430: I tensorflow/core/platform/cpu_feature_guard.cc:142] Your CPU supports instructions that this TensorFlow binary was not compiled to
 use: AVX2 FMA
predictor_1          | 2021-10-31 15:47:08.618514: I tensorflow/core/platform/profile_utils/cpu_utils.cc:94] CPU Frequency: 2904000000 Hz
predictor_1          | 2021-10-31 15:47:08.618836: I tensorflow/compiler/xla/service/service.cc:168] XLA service 0x55ee8e27e7c0 executing computations on platform Host. Devices:
predictor_1          | 2021-10-31 15:47:08.618876: I tensorflow/compiler/xla/service/service.cc:175]   StreamExecutor device (0): Host, Default Version
predictor_1          | 2021-10-31 15:47:08.624978: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 62509056 exceeds 10% of system memory.
predictor_1          | 2021-10-31 15:47:08.693357: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 62509056 exceeds 10% of system memory.
predictor_1          | 2021-10-31 15:47:08.720917: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 62509056 exceeds 10% of system memory.
predictor_1          | 2021-10-31 15:47:14.143257: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 62509056 exceeds 10% of system memory.
predictor_1          | 2021-10-31 15:47:19.841075: W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 62509056 exceeds 10% of system memory.
predictor_1          | INFO:     Application startup complete.
predictor_1          | INFO:     172.23.0.1:33328 - "GET / HTTP/1.1" 200 OK
```



```
predictor_1          | INFO:     172.23.0.1:33328 - "GET / HTTP/1.1" 200 OK
predictor_1          | news text: Today is India vs New Zealand, T20 Cricket World Cup match
predictor_1          | text: Today is India vs New Zealand, T20 Cricket World Cup match
predictor_1          | intent: Sports
predictor_1          | INFO:     172.23.0.1:33338 - "POST /predictNews HTTP/1.1" 200 OK
```

News Classifier Project Report

- **What comes out as an output:**
  Predicted news category is returned to the UI.

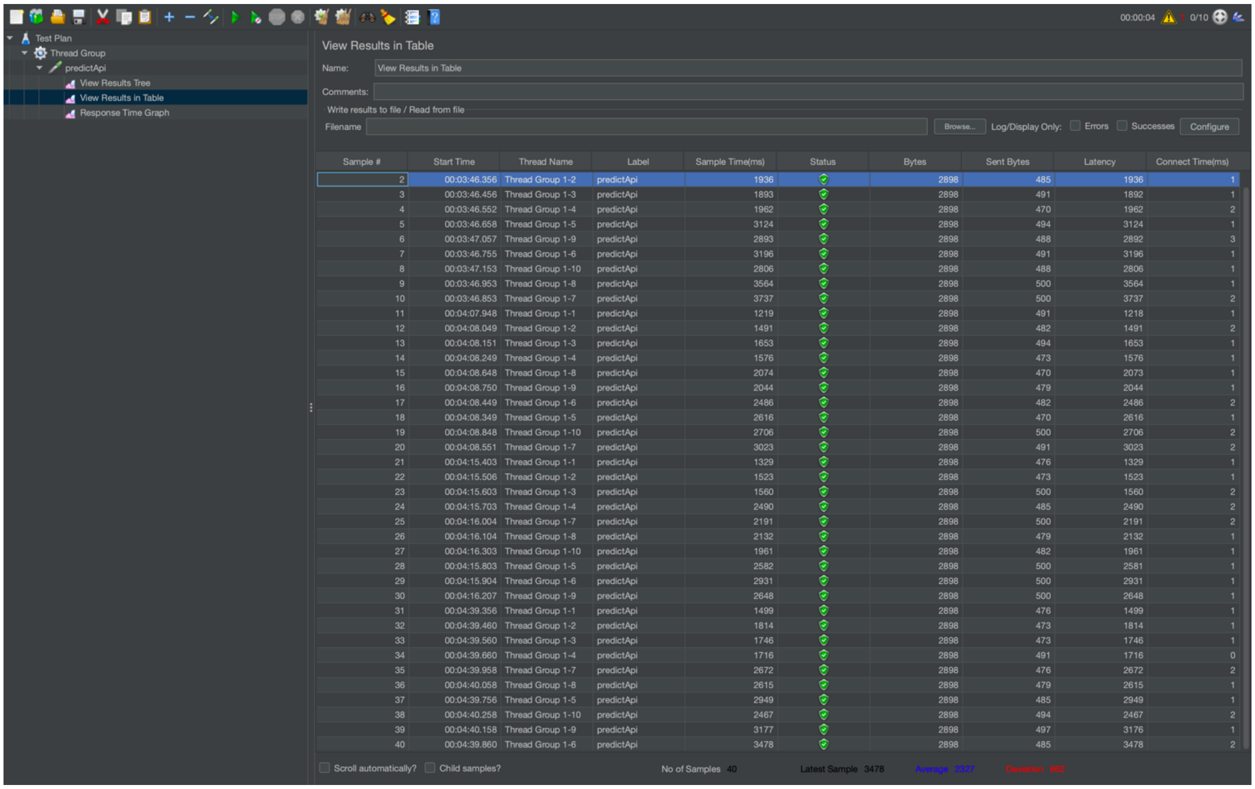


Enter the below feilds to **predict the news category.**

* All feilds are mandatory

PLease enter news text*

Today is India vs New Zealand, T20 Cricket World Cup match



View Results in Table

Name: View Results in Table

Comments:

Write results to file / Read from file

Filename

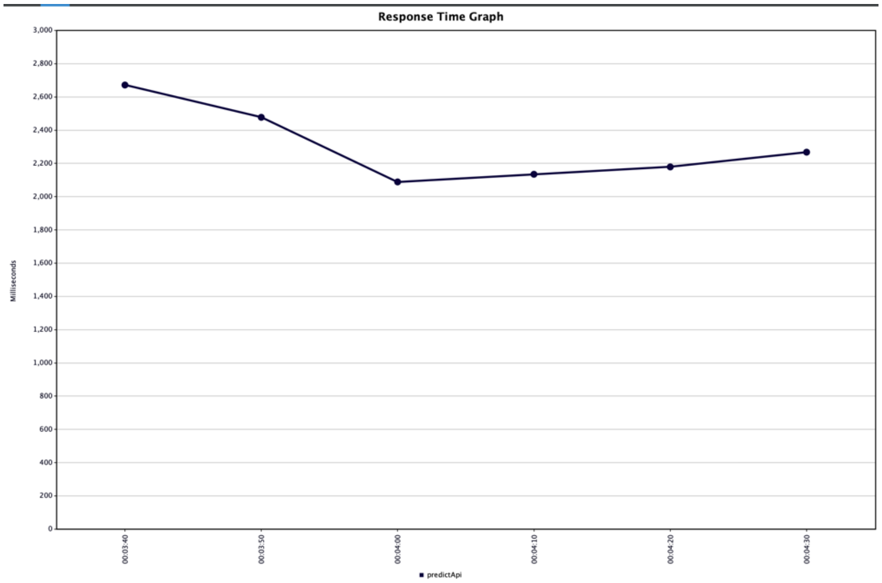
Log/Display Only: □ Errors □ Successes

Sample #	Start Time	Thread Name	Label	Sample Time(ms)	Status	Bytes	Sent Bytes	Latency	Connect Time(ms)
2	00:03:46.356	Thread Group 1-2	predictApi	1936	✓	2898	485	1936	1
3	00:03:46.456	Thread Group 1-3	predictApi	1893	✓	2898	491	1892	1
4	00:03:46.552	Thread Group 1-4	predictApi	1962	✓	2898	470	1962	2
5	00:03:46.658	Thread Group 1-5	predictApi	3124	✓	2898	494	3124	1
6	00:03:47.057	Thread Group 1-9	predictApi	2893	✓	2898	488	2892	3
7	00:03:46.755	Thread Group 1-6	predictApi	3196	✓	2898	491	3196	1
8	00:03:47.153	Thread Group 1-10	predictApi	2806	✓	2898	488	2806	1
9	00:03:46.953	Thread Group 1-8	predictApi	3564	✓	2898	500	3564	1
10	00:03:46.853	Thread Group 1-7	predictApi	3737	✓	2898	500	3737	2
11	00:04:07.948	Thread Group 1-1	predictApi	1219	✓	2898	491	1218	1
12	00:04:08.049	Thread Group 1-2	predictApi	1491	✓	2898	482	1491	2
13	00:04:08.151	Thread Group 1-3	predictApi	1653	✓	2898	494	1653	1
14	00:04:08.249	Thread Group 1-4	predictApi	1576	✓	2898	473	1576	1
15	00:04:08.648	Thread Group 1-8	predictApi	2074	✓	2898	470	2073	1
16	00:04:08.750	Thread Group 1-9	predictApi	2044	✓	2898	479	2044	1
17	00:04:08.449	Thread Group 1-6	predictApi	2486	✓	2898	482	2486	2
18	00:04:08.349	Thread Group 1-5	predictApi	2616	✓	2898	470	2616	1
19	00:04:08.848	Thread Group 1-10	predictApi	2706	✓	2898	500	2706	2
20	00:04:08.551	Thread Group 1-7	predictApi	3023	✓	2898	491	3023	2
21	00:04:15.403	Thread Group 1-1	predictApi	1329	✓	2898	476	1329	1
22	00:04:15.506	Thread Group 1-2	predictApi	1523	✓	2898	473	1523	1
23	00:04:15.603	Thread Group 1-3	predictApi	1560	✓	2898	500	1560	2
24	00:04:15.703	Thread Group 1-4	predictApi	2490	✓	2898	485	2490	2
25	00:04:16.004	Thread Group 1-7	predictApi	2191	✓	2898	500	2191	2
26	00:04:16.104	Thread Group 1-8	predictApi	2132	✓	2898	479	2132	1
27	00:04:16.303	Thread Group 1-10	predictApi	1961	✓	2898	482	1961	1
28	00:04:15.803	Thread Group 1-5	predictApi	2582	✓	2898	500	2581	1
29	00:04:15.904	Thread Group 1-6	predictApi	2931	✓	2898	500	2931	1
30	00:04:16.207	Thread Group 1-9	predictApi	2648	✓	2898	500	2648	1
31	00:04:39.356	Thread Group 1-1	predictApi	1499	✓	2898	476	1499	1
32	00:04:39.460	Thread Group 1-2	predictApi	1814	✓	2898	473	1814	1
33	00:04:39.560	Thread Group 1-3	predictApi	1746	✓	2898	473	1746	1
34	00:04:39.660	Thread Group 1-4	predictApi	1716	✓	2898	491	1716	0
35	00:04:39.958	Thread Group 1-7	predictApi	2672	✓	2898	476	2672	2
36	00:04:40.058	Thread Group 1-8	predictApi	2615	✓	2898	479	2615	1
37	00:04:39.756	Thread Group 1-5	predictApi	2949	✓	2898	485	2949	1
38	00:04:40.258	Thread Group 1-10	predictApi	2467	✓	2898	494	2467	2
39	00:04:40.158	Thread Group 1-9	predictApi	3177	✓	2898	497	3176	1
40	00:04:39.860	Thread Group 1-6	predictApi	3478	✓	2898	485	3478	2

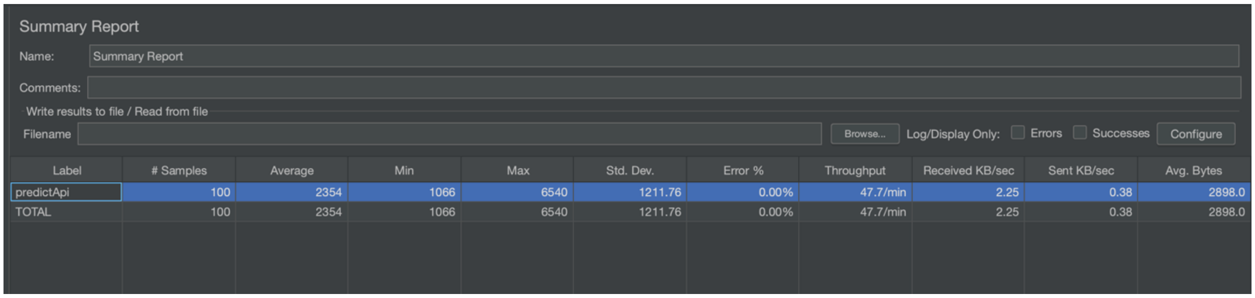
□ Scroll automatically? □ Child samples? No of Samples 40 Latest Sample 3478 Average 2327 Deviation 662

API Stress Testing:

- ### API Response Time



- ### Summary Report (API Throughput)



Name: Summary Report

Comments:

Filename

Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/sec	Sent KB/sec	Avg. Bytes
predictApi	100	2354	1066	6540	1211.76	0.00%	47.7/min	2.25	0.38	2898.0
TOTAL	100	2354	1066	6540	1211.76	0.00%	47.7/min	2.25	0.38	2898.0

Challenges encountered and how we tackled them:

We faced below mentioned challenges:

- Finding better legal data sources
  - We have used Rapid API and custom dataset.
- Rapid API rate limiting makes the pipeline slow
  - For training, we have used the dataset dumped into mongodb
- Data Labeling
  - Annotating correct labels to custom dataset was done manually
- Connecting MongoDb and spark
  - Tried different ways of connecting spark to mongodb then implemented it with mongo-spark-connector
- Authorization issue in MongoDb connection
  - Configured appropriate parameters to resolve it

- Loading model from HDFS
    - Configured HDFS in spark node.
- Python-Tensorflow version compatibility
    - Installed compatible libraries versions
- Slow prediction because of Hardware limitation
    - Increased docker memory allocation and processor cores
- Loading Bert tokenizer in predictor service
    - Used save and load methods of pytorch
- Responsive UI Design and Integration for mobile support
    - Used Jinja2Templates

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