Soumya Venkatesan & Shri Sai Sadhana Natarajan

LAB 3: DATA ANALYTICS PIPELINE USING APACHE SPARK

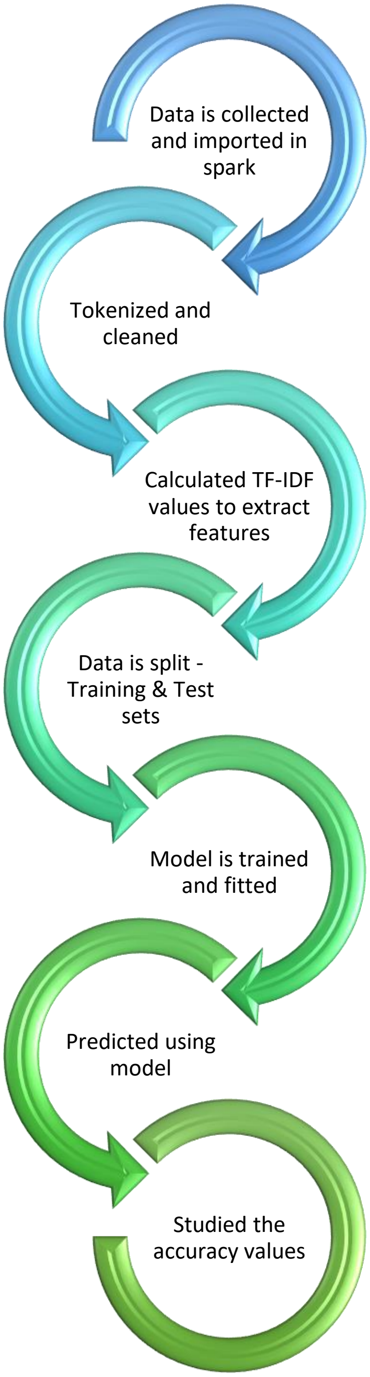

Data is collected and imported in spark
Tokenized and cleaned
Calculated TF-IDF values to extract features
Data is split - Training & Test sets
Model is trained and fitted
Predicted using model
Studied the accuracy values

Fig 1: Flow Chart

1. Data Collection:

News articles data required for the lab is collected from NYTimes.com. In order to do this, we use nytimesAPI key and collect the url for the articles based on a title and save it in a csv file. Then for each url in the csv file, we perform an automated collection of articles and store it in separate text files.

2. Data Cleaning:

Data is cleaned by splitting each sentence into words and removing unnecessary words. This process involves the following two steps:

- **Tokenize:** Tokenization is the process in which sentences are take and broken into individual words. Several tokenizer classes are available for this purpose. Here we use **regexTokenizer ().** This tokenizer extracts the tokens either using the pattern that is provided to split the tokens or by repeatedly matching regular expressions.
- **Stop words Removal:** Stopwords are words such as 'the', 'a',.…etc. which appear frequently in the document and are of not any use in the analysis process. In order to remove these words, we use **StopWordsRemover ().** This takes the list of stopwords specified by the stopwords parameter and drops all of them from the input sequence. A function called loadDefaultStopwords () can be used to remove the default stopwords that are present for each language.

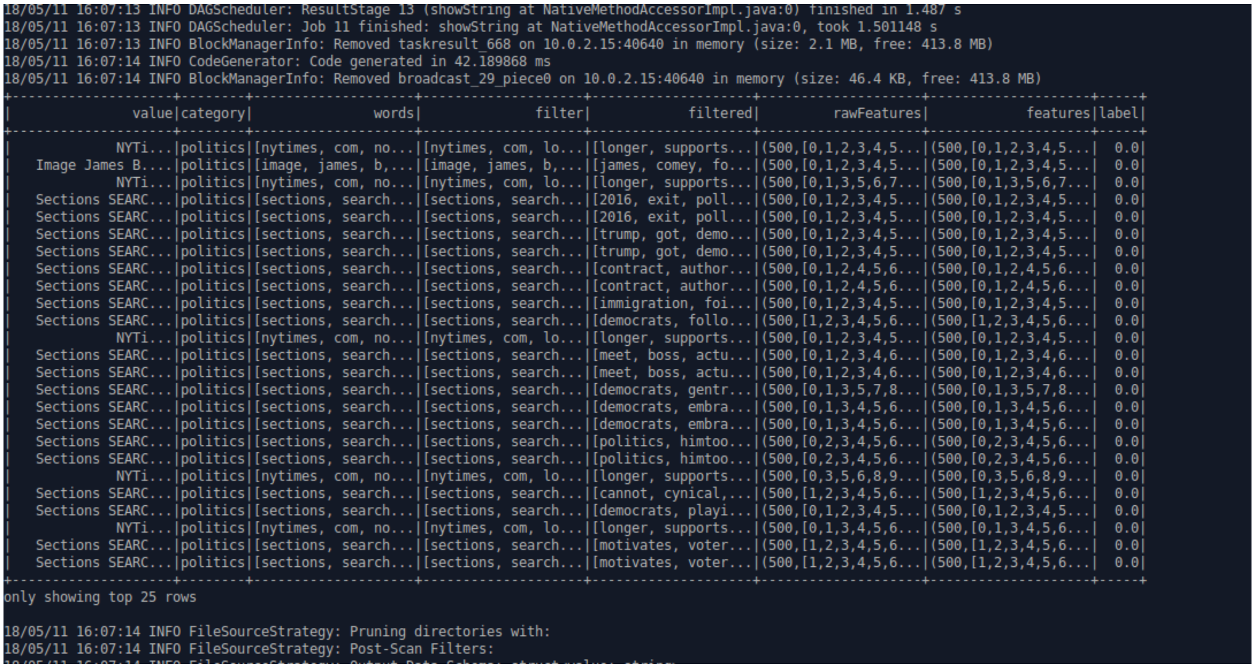
3. Feature Engineering:

Feature is a property or phenomenon that could be used to construct a model. In our model we extract words (or features) that help us characterize the category and compute the probability of the word frequency to the total words in the article. In order to perform this, we use TF-IDF (Term Frequency–Inverse Document Frequency). TF-IDF helps identify the importance of a word in a collection. The working is as follows:

- TF helps convert word to vector. HashingTF or CountVectorizer can be used for this purpose. We use HashingTF to generate frequency vector. HashingTF takes set of terms, maps it to raw feature using hash function and gives fixed length feature vectors.
- IDF is like an estimator. It takes the feature vectors produced by TF and scales the columns based on their category.

The results obtained after feature engineering is shown in Fig 2

Using Pipelines: The process of cleaning data and feature extraction is carried out within a pipeline. Pipeline is a set of data processing elements connected together in such a way that output of one element is the input of another. MLlib is used in order to perform this.



```
18/05/11 16:07:13 INFO DAGScheduler: ResultStage 13 (showString at NativeMethodAccessorImpl.java:0) finished in 1.487 s
18/05/11 16:07:13 INFO DAGScheduler: Job 11 finished: showString at NativeMethodAccessorImpl.java:0, took 1.501148 s
18/05/11 16:07:13 INFO BlockManagerInfo: Removed taskresult_668 on 10.0.2.15:40640 in memory (size: 2.1 MB, free: 413.8 MB)
18/05/11 16:07:14 INFO CodeGenerator: Code generated in 42.189868 ms
18/05/11 16:07:14 INFO BlockManagerInfo: Removed broadcast_29_piece0 on 10.0.2.15:40640 in memory (size: 46.4 KB, free: 413.8 MB)
+--------------------+--------+--------------------+--------------------+--------------------+--------------------+--------------------+-----+
|               value|category|               words|              filter|            filtered|         rawFeatures|            features|label|
+--------------------+--------+--------------------+--------------------+--------------------+--------------------+--------------------+-----+
|             NYTi...|politics|[nytimes, com, no...|[nytimes, com, lo...|[longer, supports...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|   Image James B....|politics|[image, james, b,...|[image, james, b,...|[james, comey, fo...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|             NYTi...|politics|[nytimes, com, no...|[nytimes, com, lo...|[longer, supports...|(500,[0,1,3,5,6,7...|(500,[0,1,3,5,6,7...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[2016, exit, poll...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[2016, exit, poll...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[trump, got, demo...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[trump, got, demo...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[contract, author...|(500,[0,1,2,4,5,6...|(500,[0,1,2,4,5,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[contract, author...|(500,[0,1,2,4,5,6...|(500,[0,1,2,4,5,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[immigration, foi...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[democrats, follo...|(500,[1,2,3,4,5,6...|(500,[1,2,3,4,5,6...|  0.0|
|             NYTi...|politics|[nytimes, com, no...|[nytimes, com, lo...|[longer, supports...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[meet, boss, actu...|(500,[0,1,2,3,4,6...|(500,[0,1,2,3,4,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[meet, boss, actu...|(500,[0,1,2,3,4,6...|(500,[0,1,2,3,4,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[democrats, gentr...|(500,[0,1,3,5,7,8...|(500,[0,1,3,5,7,8...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[democrats, embra...|(500,[0,1,3,4,5,6...|(500,[0,1,3,4,5,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[democrats, embra...|(500,[0,1,3,4,5,6...|(500,[0,1,3,4,5,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[politics, himtoo...|(500,[0,2,3,4,5,6...|(500,[0,2,3,4,5,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[politics, himtoo...|(500,[0,2,3,4,5,6...|(500,[0,2,3,4,5,6...|  0.0|
|             NYTi...|politics|[nytimes, com, no...|[nytimes, com, lo...|[longer, supports...|(500,[0,3,5,6,8,9...|(500,[0,3,5,6,8,9...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[cannot, cynical,...|(500,[1,2,3,4,5,6...|(500,[1,2,3,4,5,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[democrats, playi...|(500,[0,1,2,3,4,5...|(500,[0,1,2,3,4,5...|  0.0|
|             NYTi...|politics|[nytimes, com, no...|[nytimes, com, lo...|[longer, supports...|(500,[0,1,3,4,5,6...|(500,[0,1,3,4,5,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[motivates, voter...|(500,[1,2,3,4,5,6...|(500,[1,2,3,4,5,6...|  0.0|
|   Sections SEARC...|politics|[sections, search...|[sections, search...|[motivates, voter...|(500,[1,2,3,4,5,6...|(500,[1,2,3,4,5,6...|  0.0|
+--------------------+--------+--------------------+--------------------+--------------------+--------------------+--------------------+-----+
only showing top 25 rows

18/05/11 16:07:14 INFO FileSourceStrategy: Pruning directories with:
18/05/11 16:07:14 INFO FileSourceStrategy: Post-Scan Filters:
18/05/11 16:07:14 INFO FileSourceStrategy: Output Data Schema: struct<value: string
```

Fig 2: Feature Engineering Result

- The pipeline does data cleaning on raw data and converts it to words/vectors.
- It then performs feature engineering on the previous output(vectors) and generates features.
- It passes these features on to the multiclass classification model

The flow of the process using pipeline is shown in Fig 3.

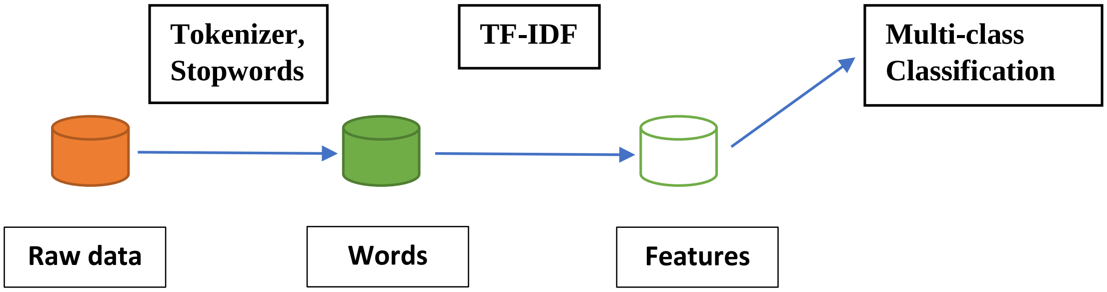


Fig 3: Pipeline Working

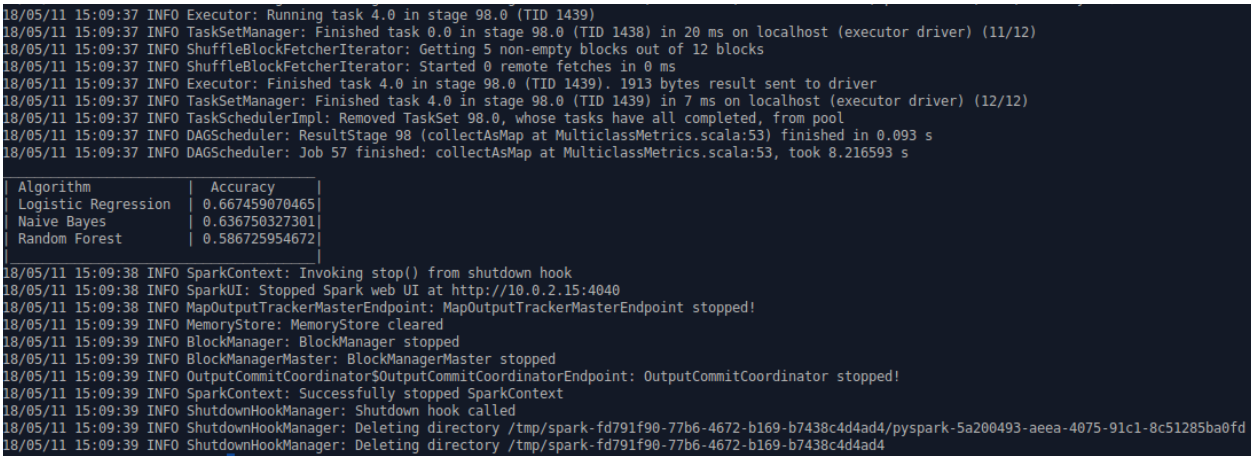
4. Multi-class Classification:

This involves, classifying various instances and finding the accuracy based on the test and train sets. We use the following three methods and determine the accuracy for each.

- **Logistic Regression:** Logistic regression is a type of predictive analysis where we use one or more independent variables to determine the outcome.

- **Naïve Bayes classification:** Naïve Bayes is a probabilistic classifier that uses Bayes theorem to classify data making strong independence assumptions between features.
- **Random Forest:** Random forest is a supervised learning method which classifies data by constructing decision trees and outputs the mode of classes or mean of the individual trees.

The accuracy we obtained using these three algorithms for classification are as shown in the Fig4

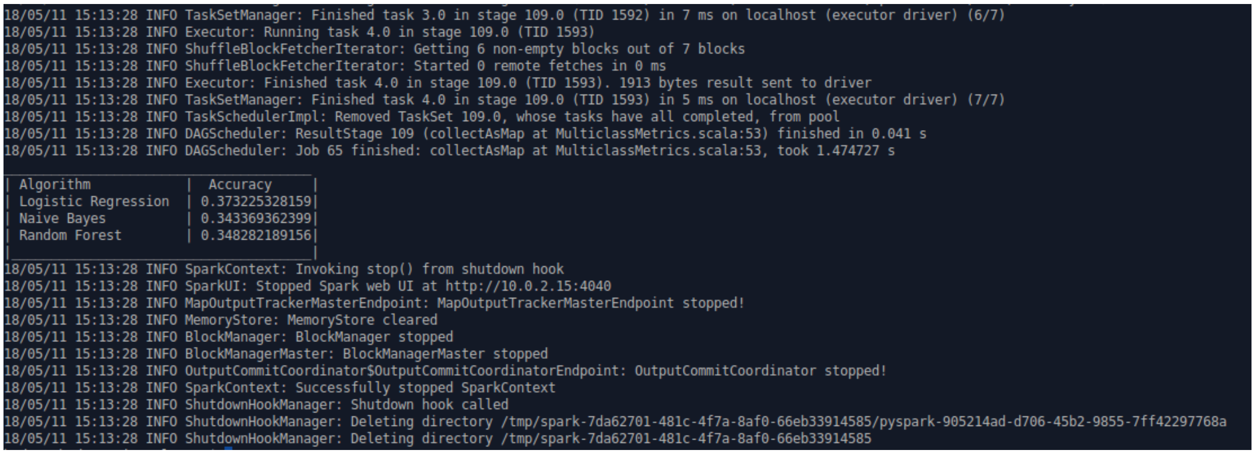


```
18/05/11 15:09:37 INFO Executor: Running task 4.0 in stage 98.0 (TID 1439)
18/05/11 15:09:37 INFO TaskSetManager: Finished task 0.0 in stage 98.0 (TID 1438) in 20 ms on localhost (executor driver) (11/12)
18/05/11 15:09:37 INFO ShuffleBlockFetcherIterator: Getting 5 non-empty blocks out of 12 blocks
18/05/11 15:09:37 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
18/05/11 15:09:37 INFO Executor: Finished task 4.0 in stage 98.0 (TID 1439). 1913 bytes result sent to driver
18/05/11 15:09:37 INFO TaskSetManager: Finished task 4.0 in stage 98.0 (TID 1439) in 7 ms on localhost (executor driver) (12/12)
18/05/11 15:09:37 INFO TaskSchedulerImpl: Removed TaskSet 98.0, whose tasks have all completed, from pool
18/05/11 15:09:37 INFO DAGScheduler: ResultStage 98 (collectAsMap at MulticlassMetrics.scala:53) finished in 0.093 s
18/05/11 15:09:37 INFO DAGScheduler: Job 57 finished: collectAsMap at MulticlassMetrics.scala:53, took 8.216593 s
 ______________________________________
| Algorithm           |  Accuracy      |
| Logistic Regression | 0.667459070465|
| Naive Bayes         | 0.636750327301|
| Random Forest       | 0.586725954672|
|______________________________________|
18/05/11 15:09:38 INFO SparkContext: Invoking stop() from shutdown hook
18/05/11 15:09:38 INFO SparkUI: Stopped Spark web UI at http://10.0.2.15:4040
18/05/11 15:09:38 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
18/05/11 15:09:39 INFO MemoryStore: MemoryStore cleared
18/05/11 15:09:39 INFO BlockManager: BlockManager stopped
18/05/11 15:09:39 INFO BlockManagerMaster: BlockManagerMaster stopped
18/05/11 15:09:39 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
18/05/11 15:09:39 INFO SparkContext: Successfully stopped SparkContext
18/05/11 15:09:39 INFO ShutdownHookManager: Shutdown hook called
18/05/11 15:09:39 INFO ShutdownHookManager: Deleting directory /tmp/spark-fd791f90-77b6-4672-b169-b7438c4d4ad4/pyspark-5a200493-aeea-4075-91c1-8c51285ba0fd
18/05/11 15:09:39 INFO ShutdownHookManager: Deleting directory /tmp/spark-fd791f90-77b6-4672-b169-b7438c4d4ad4
```

Fig 4: Classification Accuracy

5.Testing:

Now we perform testing by collecting an unknown set of data (not testing set) and repeating the classification process for the three classification algorithms. The accuracy we obtained using Logistic Regression, Naïve Bayes and Random Forest algorithms for testing are as shown in the Fig 5



```
18/05/11 15:13:28 INFO TaskSetManager: Finished task 3.0 in stage 109.0 (TID 1592) in 7 ms on localhost (executor driver) (6/7)
18/05/11 15:13:28 INFO Executor: Running task 4.0 in stage 109.0 (TID 1593)
18/05/11 15:13:28 INFO ShuffleBlockFetcherIterator: Getting 6 non-empty blocks out of 7 blocks
18/05/11 15:13:28 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 0 ms
18/05/11 15:13:28 INFO Executor: Finished task 4.0 in stage 109.0 (TID 1593). 1913 bytes result sent to driver
18/05/11 15:13:28 INFO TaskSetManager: Finished task 4.0 in stage 109.0 (TID 1593) in 5 ms on localhost (executor driver) (7/7)
18/05/11 15:13:28 INFO TaskSchedulerImpl: Removed TaskSet 109.0, whose tasks have all completed, from pool
18/05/11 15:13:28 INFO DAGScheduler: ResultStage 109 (collectAsMap at MulticlassMetrics.scala:53) finished in 0.041 s
18/05/11 15:13:28 INFO DAGScheduler: Job 65 finished: collectAsMap at MulticlassMetrics.scala:53, took 1.474727 s
 ______________________________________
| Algorithm           |  Accuracy      |
| Logistic Regression | 0.373225328159|
| Naive Bayes         | 0.343369362399|
| Random Forest       | 0.348282189156|
|______________________________________|
18/05/11 15:13:28 INFO SparkContext: Invoking stop() from shutdown hook
18/05/11 15:13:28 INFO SparkUI: Stopped Spark web UI at http://10.0.2.15:4040
18/05/11 15:13:28 INFO MapOutputTrackerMasterEndpoint: MapOutputTrackerMasterEndpoint stopped!
18/05/11 15:13:28 INFO MemoryStore: MemoryStore cleared
18/05/11 15:13:28 INFO BlockManager: BlockManager stopped
18/05/11 15:13:28 INFO BlockManagerMaster: BlockManagerMaster stopped
18/05/11 15:13:28 INFO OutputCommitCoordinator$OutputCommitCoordinatorEndpoint: OutputCommitCoordinator stopped!
18/05/11 15:13:28 INFO SparkContext: Successfully stopped SparkContext
18/05/11 15:13:28 INFO ShutdownHookManager: Shutdown hook called
18/05/11 15:13:28 INFO ShutdownHookManager: Deleting directory /tmp/spark-7da62701-481c-4f7a-8af0-66eb33914585/pyspark-905214ad-d706-45b2-9855-7ff42297768a
18/05/11 15:13:28 INFO ShutdownHookManager: Deleting directory /tmp/spark-7da62701-481c-4f7a-8af0-66eb33914585
```

Fig 5: Testing Accuracy