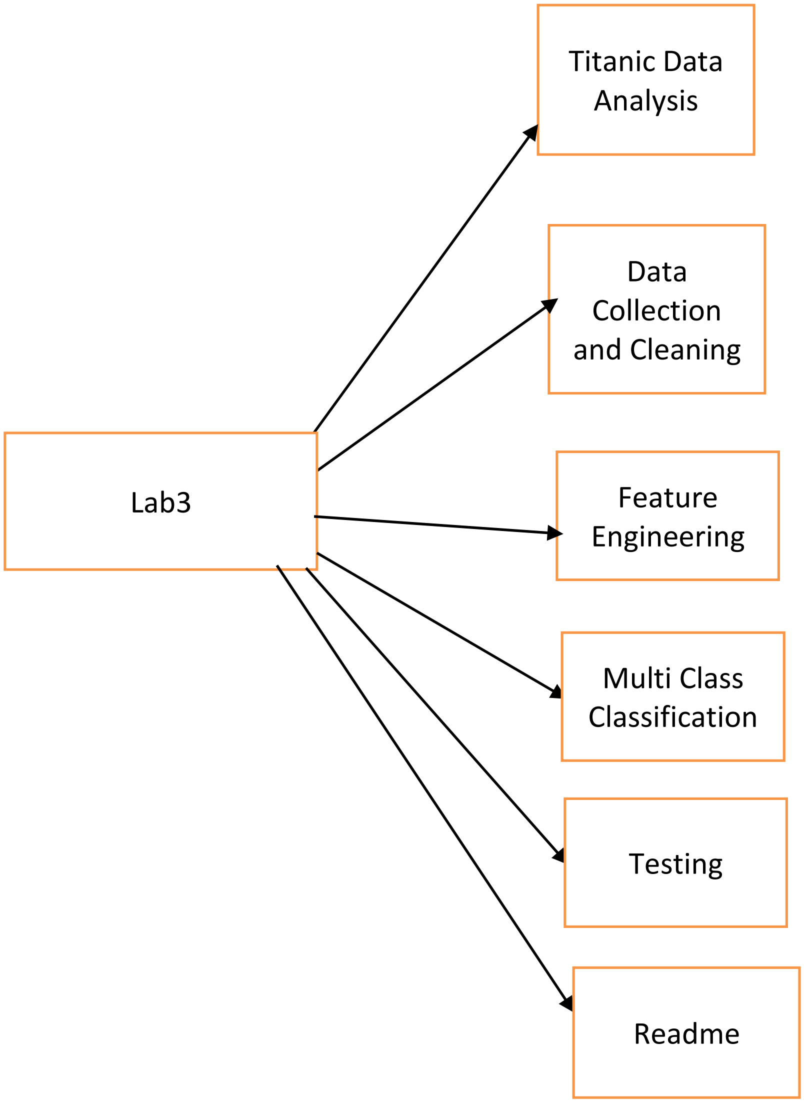
Data-Intensive Computing

LAB 3: DATA ANALYTICS PIPELINE USING APACHE SPARK

Done by:
Charanya Sudharsanan – 50245956 - csudhars
Prachi Shah – 50248748 – pshah

Directory Location:



Environment:

1. **OS - MacOS**
2. **Spark installation for MacOS**
3. **Language used - Python**

Commands:

1. **Navigate to the folder with the file myNaive.py**
2. **Run the following code on terminal/cmd:**

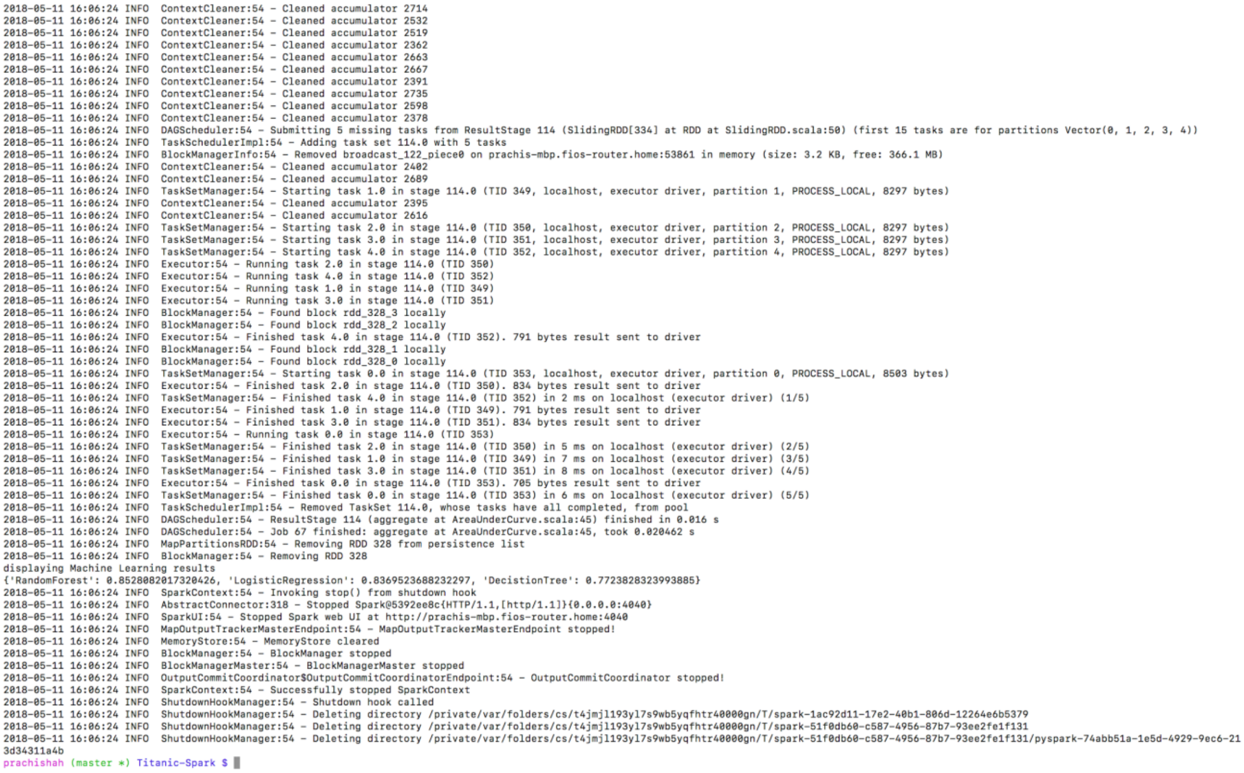
```
spark-submit myNaive.py
```

TITANIC DATA ANALYSIS:

1. **We load the dataset by reading the csv file as text in RDD data format. We load both the training and test data set.**
2. **We then transform RDD into dataframe for further manipulation.**
3. **We proceed by cleaning the data, where we remove non characters and normalize the data by filling in missing data.**
4. **Feature Engineering is performed by extracting their titles that'll be used as labels. We index the categorical labels to passing it to machine learning algorithms .**
5. **Convert the features into vectors to apply MLLIB.**
6. **Split data into training and test data set and apply models from the ML/MLLIB and tune the parameters.**

Input Files: Lab3 - > Titanic Data Analysis -> train.csv, test.csv

Output Screenshot after running it on pyspark:



```
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2714
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2532
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2519
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2362
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2663
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2667
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2391
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2735
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2598
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2378
2018-05-11 16:06:24 INFO  DAGScheduler:54 - Submitting 5 missing tasks from ResultStage 114 (SlidingRDD[334] at RDD at SlidingRDD.scala:50) (first 15 tasks are for partitions Vector(0, 1, 2, 3, 4))
2018-05-11 16:06:24 INFO  TaskSchedulerImpl:54 - Adding task set 114.0 with 5 tasks
2018-05-11 16:06:24 INFO  BlockManagerInfo:54 - Removed broadcast_122_piece0 on prachis-mbp.fios-router.home:53861 in memory (size: 3.2 KB, free: 366.1 MB)
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2402
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2689
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Starting task 1.0 in stage 114.0 (TID 349, localhost, executor driver, partition 1, PROCESS_LOCAL, 8297 bytes)
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2395
2018-05-11 16:06:24 INFO  ContextCleaner:54 - Cleaned accumulator 2616
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Starting task 2.0 in stage 114.0 (TID 350, localhost, executor driver, partition 2, PROCESS_LOCAL, 8297 bytes)
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Starting task 3.0 in stage 114.0 (TID 351, localhost, executor driver, partition 3, PROCESS_LOCAL, 8297 bytes)
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Starting task 4.0 in stage 114.0 (TID 352, localhost, executor driver, partition 4, PROCESS_LOCAL, 8297 bytes)
2018-05-11 16:06:24 INFO  Executor:54 - Running task 2.0 in stage 114.0 (TID 350)
2018-05-11 16:06:24 INFO  Executor:54 - Running task 4.0 in stage 114.0 (TID 352)
2018-05-11 16:06:24 INFO  Executor:54 - Running task 1.0 in stage 114.0 (TID 349)
2018-05-11 16:06:24 INFO  Executor:54 - Running task 3.0 in stage 114.0 (TID 351)
2018-05-11 16:06:24 INFO  BlockManager:54 - Found block rdd_328_3 locally
2018-05-11 16:06:24 INFO  BlockManager:54 - Found block rdd_328_2 locally
2018-05-11 16:06:24 INFO  Executor:54 - Finished task 4.0 in stage 114.0 (TID 352). 791 bytes result sent to driver
2018-05-11 16:06:24 INFO  BlockManager:54 - Found block rdd_328_1 locally
2018-05-11 16:06:24 INFO  BlockManager:54 - Found block rdd_328_0 locally
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Starting task 0.0 in stage 114.0 (TID 353, localhost, executor driver, partition 0, PROCESS_LOCAL, 8503 bytes)
2018-05-11 16:06:24 INFO  Executor:54 - Finished task 2.0 in stage 114.0 (TID 350). 834 bytes result sent to driver
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Finished task 4.0 in stage 114.0 (TID 352) in 2 ms on localhost (executor driver) (1/5)
2018-05-11 16:06:24 INFO  Executor:54 - Finished task 1.0 in stage 114.0 (TID 349). 791 bytes result sent to driver
2018-05-11 16:06:24 INFO  Executor:54 - Finished task 3.0 in stage 114.0 (TID 351). 834 bytes result sent to driver
2018-05-11 16:06:24 INFO  Executor:54 - Running task 0.0 in stage 114.0 (TID 353)
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Finished task 2.0 in stage 114.0 (TID 350) in 5 ms on localhost (executor driver) (2/5)
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Finished task 1.0 in stage 114.0 (TID 349) in 7 ms on localhost (executor driver) (3/5)
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Finished task 3.0 in stage 114.0 (TID 351) in 8 ms on localhost (executor driver) (4/5)
2018-05-11 16:06:24 INFO  Executor:54 - Finished task 0.0 in stage 114.0 (TID 353). 705 bytes result sent to driver
2018-05-11 16:06:24 INFO  TaskSetManager:54 - Finished task 0.0 in stage 114.0 (TID 353) in 6 ms on localhost (executor driver) (5/5)
2018-05-11 16:06:24 INFO  TaskSchedulerImpl:54 - Removed TaskSet 114.0, whose tasks have all completed, from pool
2018-05-11 16:06:24 INFO  DAGScheduler:54 - ResultStage 114 (aggregate at AreaUnderCurve.scala:45) finished in 0.016 s
2018-05-11 16:06:24 INFO  DAGScheduler:54 - Job 67 finished: aggregate at AreaUnderCurve.scala:45, took 0.020462 s
2018-05-11 16:06:24 INFO  MapPartitionsRDD:54 - Removing RDD 328 from persistence list
2018-05-11 16:06:24 INFO  BlockManager:54 - Removing RDD 328
displaying Machine Learning results
{'RandomForest': 0.8528082017320426, 'LogisticRegression': 0.8369523688232297, 'DecistionTree': 0.7723828323993885}
2018-05-11 16:06:24 INFO  SparkContext:54 - Invoking stop() from shutdown hook
2018-05-11 16:06:24 INFO  AbstractConnector:318 - Stopped Spark@5392ee8c{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
2018-05-11 16:06:24 INFO  SparkUI:54 - Stopped Spark web UI at http://prachis-mbp.fios-router.home:4040
2018-05-11 16:06:24 INFO  MapOutputTrackerMasterEndpoint:54 - MapOutputTrackerMasterEndpoint stopped!
2018-05-11 16:06:24 INFO  MemoryStore:54 - MemoryStore cleared
2018-05-11 16:06:24 INFO  BlockManager:54 - BlockManager stopped
2018-05-11 16:06:24 INFO  BlockManagerMaster:54 - BlockManagerMaster stopped
2018-05-11 16:06:24 INFO  OutputCommitCoordinator$OutputCommitCoordinatorEndpoint:54 - OutputCommitCoordinator stopped!
2018-05-11 16:06:24 INFO  SparkContext:54 - Successfully stopped SparkContext
2018-05-11 16:06:24 INFO  ShutdownHookManager:54 - Shutdown hook called
2018-05-11 16:06:24 INFO  ShutdownHookManager:54 - Deleting directory /private/var/folders/cs/t4jmjl193yl7s9wb5yqfhtr40000gn/T/spark-1ac92d11-17e2-40b1-806d-12264e6b5379
2018-05-11 16:06:24 INFO  ShutdownHookManager:54 - Deleting directory /private/var/folders/cs/t4jmjl193yl7s9wb5yqfhtr40000gn/T/spark-51f0db60-c587-4956-87b7-93ee2fe1f131
2018-05-11 16:06:24 INFO  ShutdownHookManager:54 - Deleting directory /private/var/folders/cs/t4jmjl193yl7s9wb5yqfhtr40000gn/T/spark-51f0db60-c587-4956-87b7-93ee2fe1f131/pyspark-74abb51a-1e5d-4929-9ec6-21
3d34311a4b
prachishah (master *) Titanic-Spark $ 
```

DATA COLLECTION AND CLEANING:

Collecting data from NY Times:

- Articles from NY Times API are collected using keyword Politics/Sports/Media/Business for specified period of time. We later divide them into test and train sets.
- Removed urls containing videos and scraped only the content of the articles
- Tweets are cleaned for further data manipulation.. (Cases, special characters, spaces, one letter words, redundant words are removed).
- Put each of the articles into a seperate text file and put them on seperate folders each for one category.

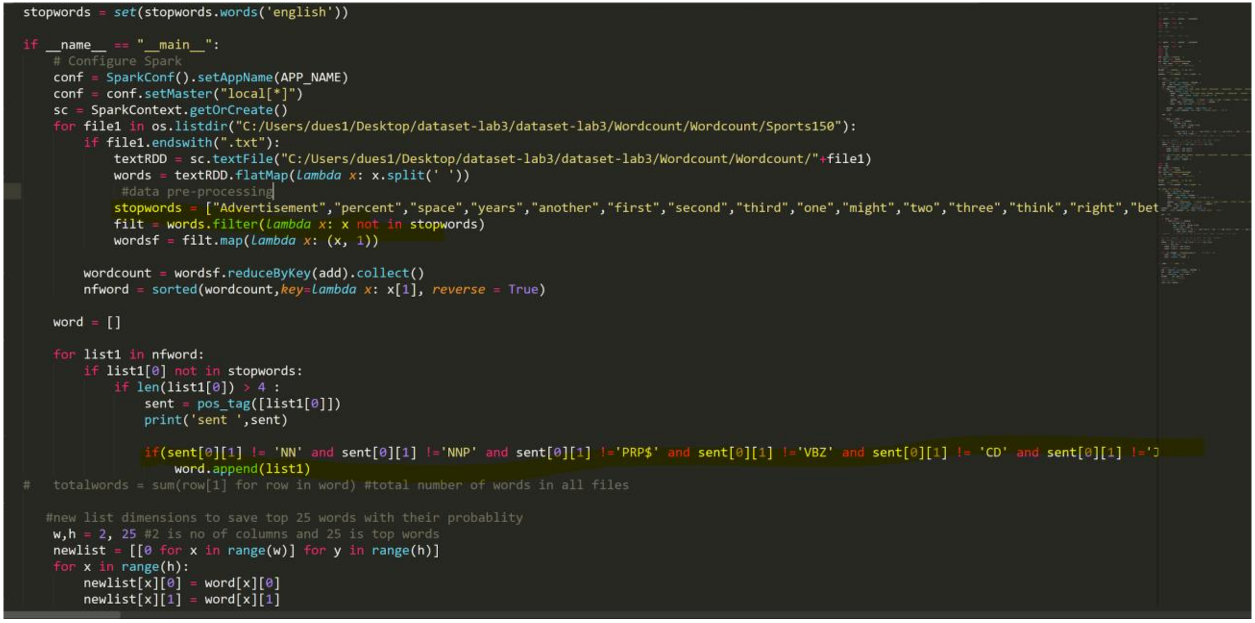
Data Collection Scripts : Lab 3 - > Data Collection and Cleaning -> Data Collection Scripts
Articles Collected : Lab 3 - > Data Collection and Cleaning -> Data -> Politics/Sports/Media/Business

We collect some articles that we will use to test the model on unseen data.

Test Data Collected : Lab3 -> Data Collection and Cleaning -> Test Data

Further Cleaning is done on pyspark to remove stop words , nouns, prepositions, conjuncts etc to get key words using the nltk package.

Data Cleaning on pyspark :



```
stopwords = set(stopwords.words('english'))

if __name__ == "__main__":
    # Configure Spark
    conf = SparkConf().setAppName(APP_NAME)
    conf = conf.setMaster("local[*]")
    sc = SparkContext.getOrCreate()
    for file1 in os.listdir("C:/Users/dues1/Desktop/dataset-lab3/dataset-lab3/Wordcount/Wordcount/Sports150"):
        if file1.endswith(".txt"):
            textRDD = sc.textFile("C:/Users/dues1/Desktop/dataset-lab3/dataset-lab3/Wordcount/Wordcount/"+file1)
            words = textRDD.flatMap(lambda x: x.split(' '))
             #data pre-processing
            stopwords = ["Advertisement","percent","space","years","another","first","second","third","one","might","two","three","think","right","bet
            filt = words.filter(lambda x: x not in stopwords)
            wordsf = filt.map(lambda x: (x, 1))

        wordcount = wordsf.reduceByKey(add).collect()
        nfword = sorted(wordcount,key=lambda x: x[1], reverse = True)

    word = []

    for list1 in nfword:
        if list1[0] not in stopwords:
            if len(list1[0]) > 4 :
                sent = pos_tag([list1[0]])
                print('sent ',sent)

                if(sent[0][1] != 'NN' and sent[0][1] !='NNP' and sent[0][1] !='PRP$' and sent[0][1] !='VBZ' and sent[0][1] != 'CD' and sent[0][1] !='J
                    word.append(list1)
#   totalwords = sum(row[1] for row in word) #total number of words in all files

   #new list dimensions to save top 25 words with their probablity
    w,h = 2, 25 #2 is no of columns and 25 is top words
    newlist = [[0 for x in range(w)] for y in range(h)]
    for x in range(h):
        newlist[x][0] = word[x][0]
        newlist[x][1] = word[x][1]
```

The Stop words are removed and redundant words like modal verbs, prepositions, nouns etc are removed on pyspark.

Data consolidation:

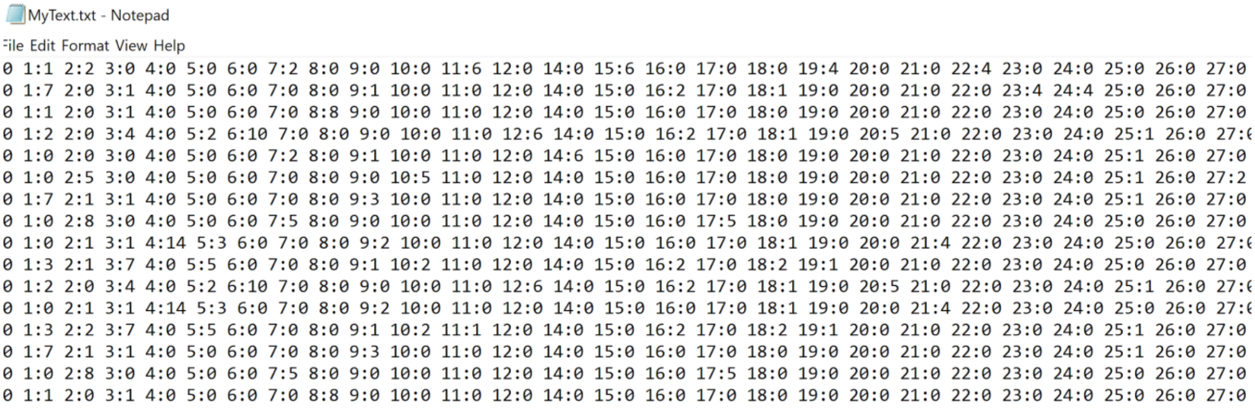
To enable further computations on data, all the articles are consolidated into one csv file. The structure of the file is the following:
Column 1 - The article text
Column 2 - Label of the article [1 - Sports, 2 - Business, 3 - Media, 4 - Politics]
The file is called final.csv

Feature Engineering :

In pyspark , we preprocess the data by looping through each of the folder and collecting top 25 words for each category . The data is cleaned and all redundant words are removed as in above screenshot . We write the top 25 words into a csv file . These serve as features for the category.

Similary we do the same on each category of data and arrive at 25 X 4 features. These features are obtained from clean data.

pyspark file to calculate wordcount and generate csv : Lab 3 - > Feature Engineering -> wordc.py
Feature files : Lab 3 - > Feature Engineering -> Feature ->bus_wc.csv/med_wc.csv/pol_wc.csv/sp_wc.csv
Article list -> Lab3 -> Feature Engineering -> traininglist.csv
Now we create a document Matrix using these csv files and our articles .
Document matrix input -> Lab3 -> Feature Engineering -> MyText.txt



```
0 1:1 2:2 3:0 4:0 5:0 6:0 7:2 8:0 9:0 10:0 11:6 12:0 14:0 15:6 16:0 17:0 18:0 19:4 20:0 21:0 22:4 23:0 24:0 25:0 26:0 27:0
0 1:7 2:0 3:1 4:0 5:0 6:0 7:0 8:0 9:1 10:0 11:0 12:0 14:0 15:0 16:2 17:0 18:1 19:0 20:0 21:0 22:0 23:4 24:4 25:0 26:0 27:0
0 1:1 2:0 3:1 4:0 5:0 6:0 7:0 8:8 9:0 10:0 11:0 12:0 14:0 15:0 16:0 17:0 18:0 19:0 20:0 21:0 22:0 23:0 24:0 25:0 26:0 27:0
0 1:2 2:0 3:4 4:0 5:2 6:10 7:0 8:0 9:0 10:0 11:0 12:6 14:0 15:0 16:2 17:0 18:1 19:0 20:5 21:0 22:0 23:0 24:0 25:1 26:0 27:
0 1:0 2:0 3:0 4:0 5:0 6:0 7:2 8:0 9:1 10:0 11:0 12:0 14:6 15:0 16:0 17:0 18:0 19:0 20:0 21:0 22:0 23:0 24:0 25:1 26:0 27:0
0 1:0 2:5 3:0 4:0 5:0 6:0 7:0 8:0 9:0 10:5 11:0 12:0 14:0 15:0 16:0 17:0 18:0 19:0 20:0 21:0 22:0 23:0 24:0 25:1 26:0 27:2
0 1:7 2:1 3:1 4:0 5:0 6:0 7:0 8:0 9:3 10:0 11:0 12:0 14:0 15:0 16:0 17:0 18:0 19:0 20:0 21:0 22:0 23:0 24:0 25:1 26:0 27:0
0 1:0 2:8 3:0 4:0 5:0 6:0 7:5 8:0 9:0 10:0 11:0 12:0 14:0 15:0 16:0 17:5 18:0 19:0 20:0 21:0 22:0 23:0 24:0 25:0 26:0 27:0
0 1:0 2:1 3:1 4:14 5:3 6:0 7:0 8:0 9:2 10:0 11:0 12:0 14:0 15:0 16:0 17:0 18:1 19:0 20:0 21:4 22:0 23:0 24:0 25:0 26:0 27:
0 1:3 2:1 3:7 4:0 5:5 6:0 7:0 8:0 9:1 10:2 11:0 12:0 14:0 15:0 16:2 17:0 18:2 19:1 20:0 21:0 22:0 23:0 24:0 25:0 26:0 27:0
0 1:2 2:0 3:4 4:0 5:2 6:10 7:0 8:0 9:0 10:0 11:0 12:6 14:0 15:0 16:2 17:0 18:1 19:0 20:5 21:0 22:0 23:0 24:0 25:1 26:0 27:
0 1:0 2:1 3:1 4:14 5:3 6:0 7:0 8:0 9:2 10:0 11:0 12:0 14:0 15:0 16:0 17:0 18:1 19:0 20:0 21:4 22:0 23:0 24:0 25:0 26:0 27:
0 1:3 2:2 3:7 4:0 5:5 6:0 7:0 8:0 9:1 10:2 11:1 12:0 14:0 15:0 16:2 17:0 18:2 19:1 20:0 21:0 22:0 23:0 24:0 25:1 26:0 27:0
0 1:7 2:1 3:1 4:0 5:0 6:0 7:0 8:0 9:3 10:0 11:0 12:0 14:0 15:0 16:0 17:0 18:0 19:0 20:0 21:0 22:0 23:0 24:0 25:1 26:0 27:0
0 1:0 2:8 3:0 4:0 5:0 6:0 7:5 8:0 9:0 10:0 11:0 12:0 14:0 15:0 16:0 17:5 18:0 19:0 20:0 21:0 22:0 23:0 24:0 25:0 26:0 27:0
0 1:1 2:0 3:1 4:0 5:0 6:0 7:0 8:8 9:0 10:0 11:0 12:0 14:0 15:0 16:0 17:0 18:0 19:0 20:0 21:0 22:0 23:0 24:0 25:0 26:0 27:0
```

Scripts used to create the csv , article list and document matrix : Lab3 -> Feature Engineering -> Script -> csvgen.py , DocMatgen.py

Multi Class Classification :

We now use Naive Bayes and Logistic Regression to perform the classification by giving as an input our document matrix in the form of "libsvm". The data is split into training and testing data. The training data is fit to the model and we test the accuracy of the model on the testing data.

PySpark Scripts for Logistic regression and Naive Bayes :

Logistic Regression - Lab3 -> Multi Class Classification -> Logistic regression -> LR.py
Naive Bayes - Lab3 -> Multi Class Classification -> Naive bayes -> MyNaive.ipynb

Data cleaning and Spark processing for Logistic Regression:

Input - final.csv

The input is first loaded in an RDD. This RDD is then converted to a Spark dataframe, with two columns, ‘FileContent’ and ‘label’.

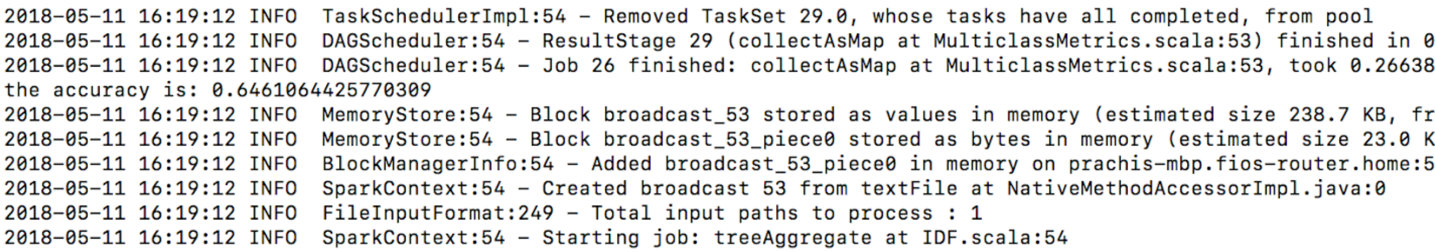
Then, the following steps are done -

1. The FileContent data is Tokenised using RegexTokenizer. This will help in filtering stop words in the next step.
2. The Stop words are removed, using nltk package.
3. The term frequence (TF) of the FileContent are calculated, using the Spark ML package. The number of features are set to 20. This means that each token is hashed to one of the 20 hashcodes. And TF for each hash is calculated.
4. The Inverse Document Frequency is calculated, using the Spark ML package.
5. Each word then has a TF-IDF score.
6. The data is split into train and test data
7. The logistic regression model is made and training data
8. To evaluate the accuracy of the model, the test data is fit on the model. And the accuracy is calculated using the evaluator.

The following part is done for testing the model on unseen/new articles:

1. The steps 1 to 5 are followed as before.
2. The data is then fit in the model from the previous part and the accuracy is calculated.

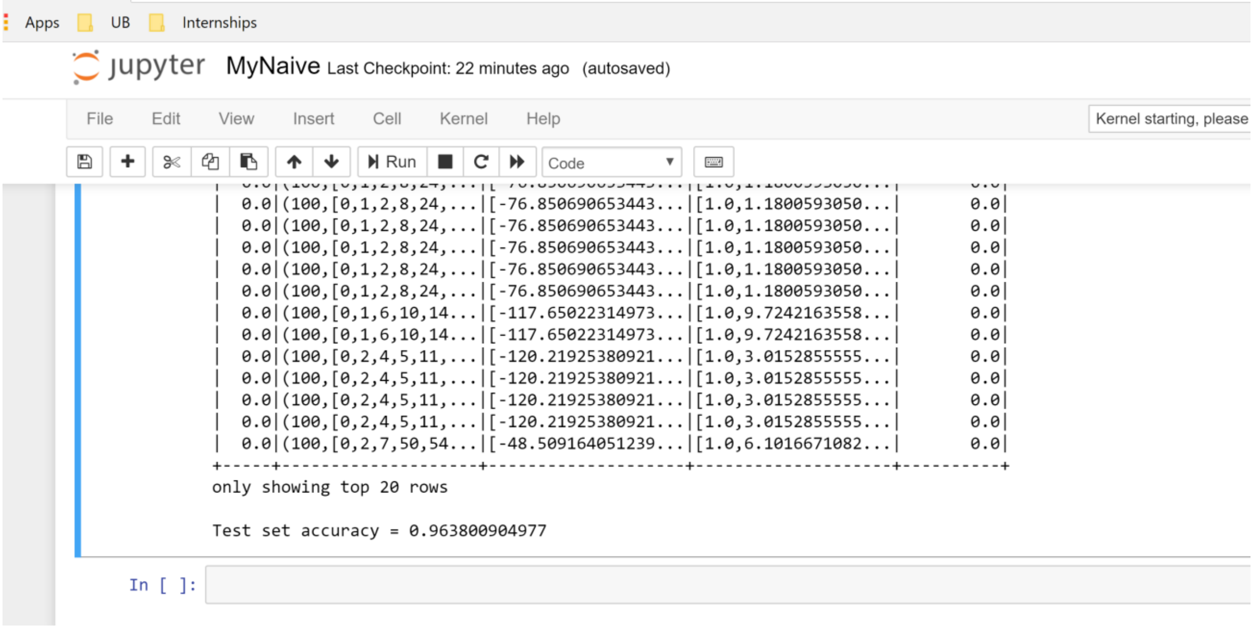
Following are the results:



```
2018-05-11 16:19:12 INFO  TaskSchedulerImpl:54 - Removed TaskSet 29.0, whose tasks have all completed, from pool
2018-05-11 16:19:12 INFO  DAGScheduler:54 - ResultStage 29 (collectAsMap at MulticlassMetrics.scala:53) finished in 0
2018-05-11 16:19:12 INFO  DAGScheduler:54 - Job 26 finished: collectAsMap at MulticlassMetrics.scala:53, took 0.26638
the accuracy is: 0.6461064425770309
2018-05-11 16:19:12 INFO  MemoryStore:54 - Block broadcast_53 stored as values in memory (estimated size 238.7 KB, fr
2018-05-11 16:19:12 INFO  MemoryStore:54 - Block broadcast_53_piece0 stored as bytes in memory (estimated size 23.0 K
2018-05-11 16:19:12 INFO  BlockManagerInfo:54 - Added broadcast_53_piece0 in memory on prachis-mbp.fios-router.home:5
2018-05-11 16:19:12 INFO  SparkContext:54 - Created broadcast 53 from textFile at NativeMethodAccessorImpl.java:0
2018-05-11 16:19:12 INFO  FileInputFormat:249 - Total input paths to process : 1
2018-05-11 16:19:12 INFO  SparkContext:54 - Starting job: treeAggregate at IDF.scala:54
```

Fig1 - Accuracy for testing data

Naïve Bayes :



```
|  0.0|(100,[0,1,2,8,24,...|[-76.850690653443...|[1.0,1.1800593050...|       0.0|
|  0.0|(100,[0,1,2,8,24,...|[-76.850690653443...|[1.0,1.1800593050...|       0.0|
|  0.0|(100,[0,1,2,8,24,...|[-76.850690653443...|[1.0,1.1800593050...|       0.0|
|  0.0|(100,[0,1,2,8,24,...|[-76.850690653443...|[1.0,1.1800593050...|       0.0|
|  0.0|(100,[0,1,2,8,24,...|[-76.850690653443...|[1.0,1.1800593050...|       0.0|
|  0.0|(100,[0,1,2,8,24,...|[-76.850690653443...|[1.0,1.1800593050...|       0.0|
|  0.0|(100,[0,1,6,10,14...|[-117.65022314973...|[1.0,9.7242163558...|       0.0|
|  0.0|(100,[0,1,6,10,14...|[-117.65022314973...|[1.0,9.7242163558...|       0.0|
|  0.0|(100,[0,2,4,5,11,...|[-120.21925380921...|[1.0,3.0152855555...|       0.0|
|  0.0|(100,[0,2,4,5,11,...|[-120.21925380921...|[1.0,3.0152855555...|       0.0|
|  0.0|(100,[0,2,4,5,11,...|[-120.21925380921...|[1.0,3.0152855555...|       0.0|
|  0.0|(100,[0,2,4,5,11,...|[-120.21925380921...|[1.0,3.0152855555...|       0.0|
|  0.0|(100,[0,2,7,50,54...|[-48.509164051239...|[1.0,6.1016671082...|       0.0|
+-----+--------------------+--------------------+--------------------+----------+
only showing top 20 rows

Test set accuracy = 0.963800904977
```

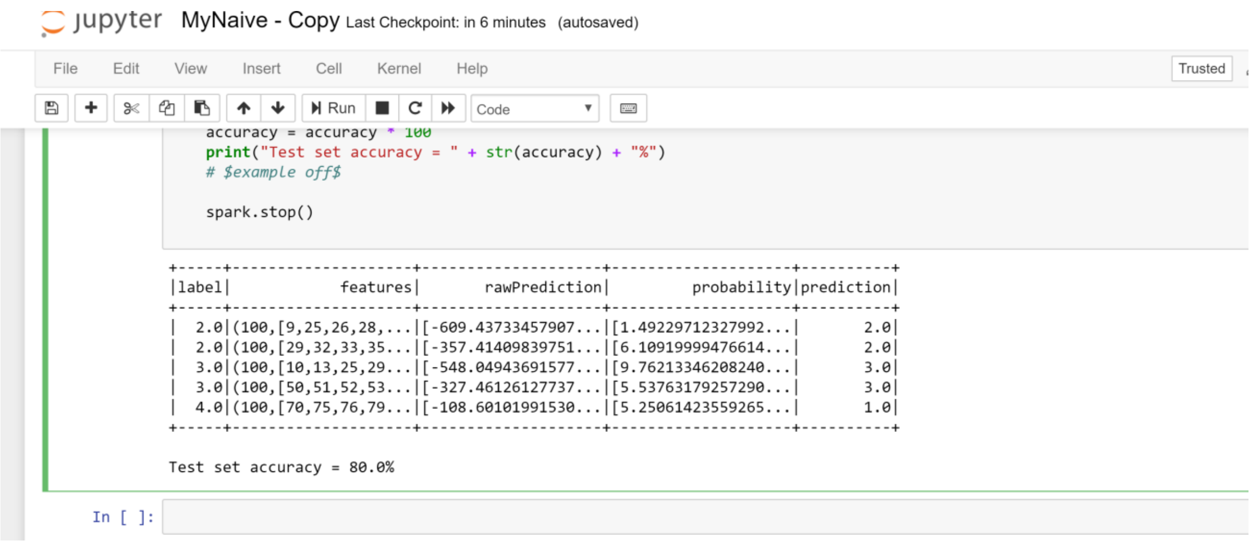
```
In [ ]:
```

Testing:

We test the above model on new unseen data. The test data has already been collected in the above step and kept separately.

This data is unseen and can be used to check the accuracy of the two models on unseen data.

Naïve Bayes : Tested on unseen Data

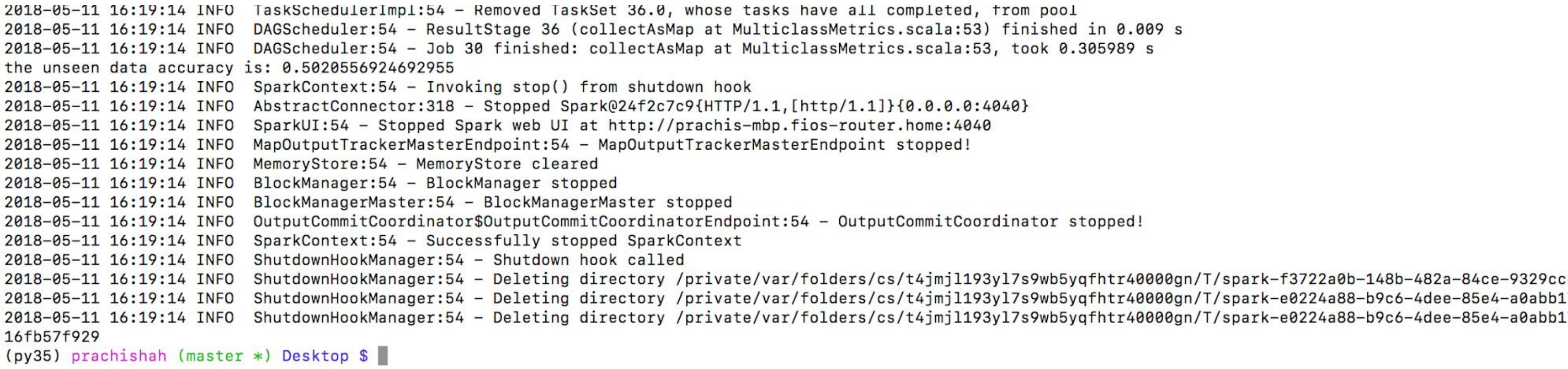


```
accuracy = accuracy * 100
print("Test set accuracy = " + str(accuracy) + "%")
# $example off$

spark.stop()
```

```
+-----+--------------------+--------------------+--------------------+----------+
|label|            features|       rawPrediction|         probability|prediction|
+-----+--------------------+--------------------+--------------------+----------+
|  2.0|(100,[9,25,26,28,...|[-609.43733457907...|[1.49229712327992...|       2.0|
|  2.0|(100,[29,32,33,35...|[-357.41409839751...|[6.10919999476614...|       2.0|
|  3.0|(100,[10,13,25,29...|[-548.04943691577...|[9.76213346208240...|       3.0|
|  3.0|(100,[50,51,52,53...|[-327.46126127737...|[5.53763179257290...|       3.0|
|  4.0|(100,[70,75,76,79...|[-108.60101991530...|[5.25061423559265...|       1.0|
+-----+--------------------+--------------------+--------------------+----------+

Test set accuracy = 80.0%
```

```
In [ ]:
```

Logistic regression : unseen Data Testing



```
2018-05-11 16:19:14 INFO  TaskSchedulerImpl:54 - Removed TaskSet 36.0, whose tasks have all completed, from pool
2018-05-11 16:19:14 INFO  DAGScheduler:54 - ResultStage 36 (collectAsMap at MulticlassMetrics.scala:53) finished in 0.009 s
2018-05-11 16:19:14 INFO  DAGScheduler:54 - Job 30 finished: collectAsMap at MulticlassMetrics.scala:53, took 0.305989 s
the unseen data accuracy is: 0.5020556924692955
2018-05-11 16:19:14 INFO  SparkContext:54 - Invoking stop() from shutdown hook
2018-05-11 16:19:14 INFO  AbstractConnector:318 - Stopped Spark@24f2c7c9{HTTP/1.1,[http/1.1]}{0.0.0.0:4040}
2018-05-11 16:19:14 INFO  SparkUI:54 - Stopped Spark web UI at http://prachis-mbp.fios-router.home:4040
2018-05-11 16:19:14 INFO  MapOutputTrackerMasterEndpoint:54 - MapOutputTrackerMasterEndpoint stopped!
2018-05-11 16:19:14 INFO  MemoryStore:54 - MemoryStore cleared
2018-05-11 16:19:14 INFO  BlockManager:54 - BlockManager stopped
2018-05-11 16:19:14 INFO  BlockManagerMaster:54 - BlockManagerMaster stopped
2018-05-11 16:19:14 INFO  OutputCommitCoordinator$OutputCommitCoordinatorEndpoint:54 - OutputCommitCoordinator stopped!
2018-05-11 16:19:14 INFO  SparkContext:54 - Successfully stopped SparkContext
2018-05-11 16:19:14 INFO  ShutdownHookManager:54 - Shutdown hook called
2018-05-11 16:19:14 INFO  ShutdownHookManager:54 - Deleting directory /private/var/folders/cs/t4jmjl193yl7s9wb5yqfhtr40000gn/T/spark-f3722a0b-148b-482a-84ce-9329cc
2018-05-11 16:19:14 INFO  ShutdownHookManager:54 - Deleting directory /private/var/folders/cs/t4jmjl193yl7s9wb5yqfhtr40000gn/T/spark-e0224a88-b9c6-4dee-85e4-a0abb1
2018-05-11 16:19:14 INFO  ShutdownHookManager:54 - Deleting directory /private/var/folders/cs/t4jmjl193yl7s9wb5yqfhtr40000gn/T/spark-e0224a88-b9c6-4dee-85e4-a0abb1
16fb57f929
(py35) prachishah (master *) Desktop $ 
```

Test Set Accuracy for Naïve Bayes: 80%
Test Set Accuracy for Logistic Regression : 50%

We tested on different sizes of data to get an optimum accuracy.

Testing for naïve bayes - > lab3 -> Testing
Testing for Logistic regression - > Done Along with test data .

Conclusion: Thus , Data has been collected from NYTimes api , cleaned and consolidated in pyspark and passed through Machine learning classification models after feature engineering to get an optimum accuracy.