# DATA ANALYTICS PIPELINE USING APACHE SPARK

*Data Intensive Computing- CSE-587*


## Jayant Solanki

UBIT Name: jayantso
Person Number: 50246821
SPRING 2018 CSE

## Anant Gupta

UBIT Name: anantram
Person Number: 50249127
SPRING 2018 CSE

# INTRODUCTION

This report contains the implementation details for the LAB 3 of DIC-CSE 587. We have successfully completed all the parts of Lab 3.

# OBJECTIVES ACHIEVED

We explored the Apache Spark framework and programming: sparkcontext (sc), dataflow operations in transformations, actions, pipelines and MLib. We applied our data analytics knowledge and machine learning skills to perform multi-class classification of text data using Apache Spark. We build a data pipeline using data from sources such as NY Times articles using the APIs provided by the data sources. We assessed the accuracy of our model using a new article for each news “category” and then compared the classification accuracy of three well-known classification algorithms, for a given test data set.

Lastly we applied the knowledge and skills learned to solve classification problems in other domains.

**Note: All the data pre-processing has been done in Spark framework using PySpark.**

# IMPLEMENTATION

## Part 1

Understand Apache Spark with Titanic data analysis.

We have loaded the training and testing dataset available from the kaggle website. Few entries from the training data loaded.

```
In [4]: 1 train_rdd.take(3)

Out[4]: ['PassengerId,Survived,Pclass,Name,Sex,Age,SibSp,Parch,Ticket,Fare,Cabin,Embarked',
         '1,0,3,"Braund, Mr. Owen Harris",male,22,1,0,A/5 21171,7.25,,S',
         '2,1,1,"Cumings, Mrs. John Bradley (Florence Briggs Thayer)",female,38,1,0,PC 17599,71.2833,C85,C']
```

Few entry from the training dataframe.

```
1 train_df.show(3)
```

```
+-----------+--------+------+---------+--------------------+------+---+-----+-----+----------------+-------+-----+--------+
|PassengerId|Survived|Pclass|FirstName|                Name|   Sex|Age|SibSp|Parch|          Ticket|   Fare|Cabin|Embarked|
+-----------+--------+------+---------+--------------------+------+---+-----+-----+----------------+-------+-----+--------+
|          1|       0|     3|   Braund|     Mr. Owen Harris|  male| 22|    1|    0|       A/5 21171|   7.25|     |       S|
|          2|       1|     1|  Cumings| Mrs. John Bradle...|female| 38|    1|    0|        PC 17599|71.2833|  C85|       C|
|          3|       1|     3|Heikkinen|         Miss. Laina|female| 26|    0|    0|STON/O2. 3101282|  7.925|     |       S|
+-----------+--------+------+---------+--------------------+------+---+-----+-----+----------------+-------+-----+--------+
only showing top 3 rows
```

Schema of the dataframe:

```
10 df.printSchema()
```

```
root
 |-- PassengerId: string (nullable = true)
 |-- Survived: double (nullable = true)
 |-- Pclass: string (nullable = true)
 |-- FirstName: string (nullable = true)
 |-- Name: string (nullable = true)
 |-- Sex: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- SibSp: double (nullable = true)
 |-- Parch: double (nullable = true)
 |-- Ticket: string (nullable = true)
 |-- Fare: double (nullable = true)
 |-- Cabin: string (nullable = true)
 |-- Embarked: string (nullable = true)
 |-- Mark: string (nullable = false)
```

Accuracy of logistic regression:

```
15 print ('AUC ROC of Logistic Regression model is: '+str(testModel(lr)))
```

```
AUC ROC of Logistic Regression model is: 0.8369523688232298
```

Accuracy of other models and comparison:

```
13 print (modelPerf)
```

```
{'LogisticRegression': 0.8369523688232297, 'DecisionTree': 0.7723828323993885, 'RandomForest': 0.8585392256749873}
```

So, **Random Forest model has better accuracy (85.85 %)** compared to other 2 models.

# Part 2

Here we will use a feature extraction technique along with different supervised machine learning algorithms in Spark.

The overall workflow for the part 2 implementation has been shown in the Figure 1 below.

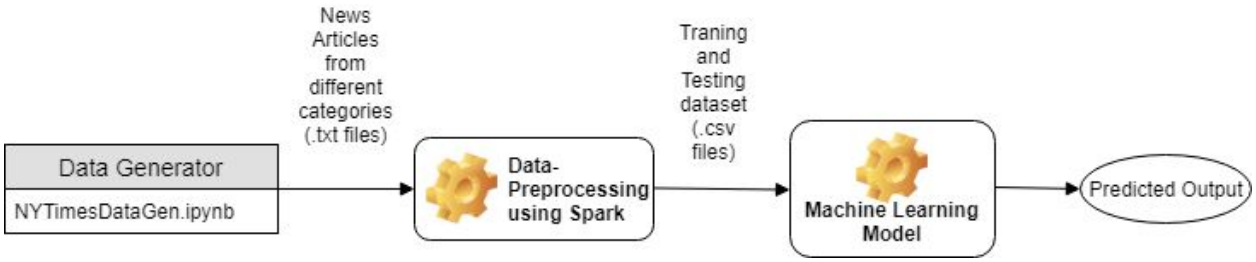


**Figure 1**

**INPUT:**

Input will be a text file which contains the article of a particular category.

**EXAMPLE:**

moviespage1articles120180506.txt

**OUTPUT:**

Output should be the category prediction from our model for this article.

**EXAMPLE:**

ENTERTAINMENT

**STEP 1 (Collect data):**

First we started with the data collection using the NYT API. We have collected the news articles for 4 popular categories (Sports, Business, Politics and of course entertainment). For each category we used multiple keywords to fire the query to the API.

Some of the keywords used are :

Politics Category – Trump, Election, Democrats, Republican etc.

Business Category – Economy, Trade, Market etc.

Sports Category – Football, Baseball, World Cup, Volleyball etc.

Entertainment Category – Movie, Art, Fashion, etc.

After multiple iteration we finally collected around 1000 articles for each category which are stored in different folders as per the category they belong to.
So total articles for first part if 5000 (4000: training set; 1000: testing set).
For 2nd part i.e collecting random data for testing the model we collected 1000 articles.

**Step 2 (Data Cleaning using Apache Spark):**

In Step 1 we collected the articles in text file. Here we have done the cleaning operation by tokenizing the text file data and removing the noisy words (stop words, punctuation, etc).
In order to make it easier to train our machine learning model, we have created csv files for each category.

This csv files has all the articles for a particular news category in the below format. Category is the category name to which the corresponding article entry (Body) belongs.

| Dates | Category | Topic | Page | Body |
|---|---|---|---|---|
| 20180501 | business | business | 2 | SEOUL WITH A FALTERING VOICE AND HER HEAD DOWN THE YOUNGEST DAUGHTER |
| 20180502 | business | finance | 4 | JEFFERSON CITY MO THE LATEST ON THE INVESTIGATION OF MISSOURI GOV ERIC |
| 20180503 | business | business | 8 | WASHINGTON THE TRUMP ADMINISTRATION S SEVEN MAN DELEGATION TO BEIJING F |
| 20180502 | business | business | 10 | REUTERS METLIFE INC REPORTED AN 8 PERCENT RISE IN ADJUSTED FIRST QUARTE |
| 20180504 | sports | hockey | 1 | DALLAS WHEN DALLAS STARS GENERAL MANAGER JIM NILL STARTED HIS THIRD COA |
| 20180501 | business | market | 3 | SOFIA BULGARIA COULD EVENTUALLY BE IN THE MARKET FOR UP TO 10 NEW OR USE |
| 20180505 | entertainment | fashion | 10 | PHOENIX WHEN FORMER VICE PRESIDENT JOSEPH R BIDEN JR TRAVELED TO SENA |
| 20180506 | entertainment | entertainment | 3 | SARA MELINDA BEESLEY AND JOSHUA DAVID SHERMAN WERE MARRIED MAY 5 AT CO |
| 20180504 | sports | soccer | 4 | ON A WARM SATURDAY IN APRIL MOST OF THE BASEBALL DIAMONDS AND SOCCER F |
| 20180507 | sports | baseball | 5 | FEW WOULD DISPUTE THE VALUE TO CHILDREN OF PARTICIPATING IN SPORTS ORGA |
| 20180503 | business | money | 8 | REMARKS BY PRESIDENT DONALD TRUMP AND HIS LAWYER RUDY GIULIANI DEFER O |

**Step 3 ( Feature Engineering) :**

Here our aim is to extract the “words” or the “features” characterizing the category. For doing this we have used TF-IDF by importing this HashingTF and IDF from the feature package of pyspark.
TFIDF, short for term frequency–inverse document frequency, is a numerical statistic that is intended to reflect how important a word is to a document in a collection or corpus.
The tf-idf value increases proportionally to the number of times a word appears in the document and is offset by the frequency of the word in the corpus, which helps to adjust for the fact that some words appear more frequently in general.

**Step 4 (Data Partitioning) :**

Next we have partitioned the collected data into training and testing dataset.

80% as training data and 20% as testing.

Final training and test data looks as below:

**Train.csv:**

| Dates | Category | Topic | Page | Body |
|---|---|---|---|---|
| 20180501 | business | business | 2 | SEOUL WITH A FALTERING VOICE AND HER HEAD DOWN THE YOUNGEST DAUGHTER |
| 20180502 | business | finance | 4 | JEFFERSON CITY MO THE LATEST ON THE INVESTIGATION OF MISSOURI GOV ERIC |
| 20180503 | business | business | 8 | WASHINGTON THE TRUMP ADMINISTRATION S SEVEN MAN DELEGATION TO BEIJING F |
| 20180502 | business | business | 10 | REUTERS METLIFE INC REPORTED AN 8 PERCENT RISE IN ADJUSTED FIRST QUARTE |
| 20180504 | sports | hockey | 1 | DALLAS WHEN DALLAS STARS GENERAL MANAGER JIM NILL STARTED HIS THIRD COA |
| 20180501 | business | market | 3 | SOFIA BULGARIA COULD EVENTUALLY BE IN THE MARKET FOR UP TO 10 NEW OR USE |
| 20180505 | entertainment | fashion | 10 | PHOENIX WHEN FORMER VICE PRESIDENT JOSEPH R BIDEN JR TRAVELED TO SENA |
| 20180506 | entertainment | entertainment | 3 | SARA MELINDA BEESLEY AND JOSHUA DAVID SHERMAN WERE MARRIED MAY 5 AT CO |
| 20180504 | sports | soccer | 4 | ON A WARM SATURDAY IN APRIL MOST OF THE BASEBALL DIAMONDS AND SOCCER F |
| 20180507 | sports | baseball | 5 | FEW WOULD DISPUTE THE VALUE TO CHILDREN OF PARTICIPATING IN SPORTS ORGA |
| 20180503 | business | money | 8 | REMARKS BY PRESIDENT DONALD TRUMP AND HIS LAWYER RUDY GIULIANI DEFER O |

**Test.csv:**

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| Dates | Category | Topic | Page | Body | | |
| 20180505 | business | business | 2 | CAIRO EGYPT WILL HAVE TO DEEPEN ITS IMF E | | |
| 20180502 | business | money | 5 | WANT TO GET THIS BRIEFING BY EMAIL HERES | | |
| 20180501 | politics | trump | 6 | WASHINGTON U S PRESIDENT DONALD TRUMI | | |
| 20180501 | business | economy | 3 | WITH THE ARRIVAL IN BEIJING THIS WEEK OF A | | |
| 20180504 | entertainment | music | 9 | ON THURSDAY SOON AFTER A JURY CONVICTE | | |
| 20180504 | entertainment | entertainment | 9 | FRIDAY PUZZLE SOMETIMES THERE ARE TRAD | | |
| 20180504 | business | business | 8 | HONG KONG PING AN HEALTHCARE AND TECH | | |
| 20180502 | politics | election | 10 | WASHINGTON PRESIDENT DONALD TRUMP IS A | | |
| 20180504 | entertainment | fashion | 2 | LOS ANGELES IF LESLEY MANVILLE S OSCAR N | | |
| 20180504 | business | business | 2 | HONG KONG LONDON HSBC S NEW CHIEF EXE | | |
| 20180507 | business | business | 8 | HOUSTON CONOCOPHILLIPS IS TRYING TO SEI | | |
| 20180501 | politics | election | 3 | WANT TO GET THIS BRIEFING BY EMAIL HERES | | |

**Step 5 (Multi-class Classification) :**

Here we implement a machine learning model and then train it using the training dataset. After training our model we use the testing dataset to evaluate our model performance.

**Output table for the prediction made by Logistic Regression:**

```
predictions = lrModel.transform(testData)
predictions.filter(predictions['prediction'] == 0) \
    .select("Body","Category","probability","label","prediction") \
    .orderBy("probability", ascending=False) \
    .show(n = 20, truncate = 30)
```

```
+------------------------------+-------------+------------------------------+-----+----------+
|                          Body|     Category|                   probability|label|prediction|
+------------------------------+-------------+------------------------------+-----+----------+
|DALIAN MANILA  CHINA PLANS ...|     business|[0.9993406144602194,1.38312...|  0.0|       0.0|
|LONDON FRANKFURT  GERMANY S...|     business|[0.9980119859619365,4.94516...|  0.0|       0.0|
|SYDNEY  AUSTRALIA S BIGGEST...|     business|[0.997918087143643,0.001395...|  0.0|       0.0|
|PARIS  SOCGEN S CHIEF EXECU...|     business|[0.997250963812292,0.001054...|  0.0|       0.0|
|ADEN  YEMEN  THE YOUNG MOTH...|     business|[0.9964283193746383,8.25149...|  0.0|       0.0|
|ADEN  YEMEN  THE YOUNG MOTH...|     business|[0.9964283193746383,8.25149...|  0.0|       0.0|
|HONG KONG  A GADGET MAKER  ...|     business|[0.9952530049599432,0.00187...|  0.0|       0.0|
|HONG KONG  A GADGET MAKER  ...|     business|[0.9952530049599432,0.00187...|  0.0|       0.0|
|HONG KONG  A GADGET MAKER  ...|entertainment|[0.9952530049599432,0.00187...|  3.0|       0.0|
|LPC    THE SIZE OF FUND FIN...|     business|[0.9947368764797554,0.00136...|  0.0|       0.0|
|LPC    THE SIZE OF FUND FIN...|     business|[0.9947368764797554,0.00136...|  0.0|       0.0|
|LOS ANGELES LONDON  SWISS B...|     business|[0.9943281572841745,0.00255...|  0.0|       0.0|
|TORONTO  FOR ABOUT AN HOUR ...|     business|[0.9940307922171391,0.00146...|  0.0|       0.0|
|BEIJING HONG KONG  CHINESE ...|     business|[0.9933015793572394,0.00104...|  0.0|       0.0|
|PARIS  THE FRENCH FINANCE M...|     business|[0.9916311890367605,0.00552...|  0.0|       0.0|
|REUTERS    TYSON FOODS INC ...|     business|[0.991278126599585,0.004102...|  0.0|       0.0|
|FRANKFURT  THE EUROPEAN CEN...|     business|[0.9905817826204851,0.00543...|  0.0|       0.0|
|LONDON  BRITISH MANUFACTURI...|     business|[0.9902443588753802,0.00720...|  0.0|       0.0|
|PARIS  PRESSURE ON AIR FRAN...|     business|[0.9902183367998199,0.00498...|  0.0|       0.0|
|PARIS  PRESSURE ON AIR FRAN...|     business|[0.9902183367998199,0.00498...|  0.0|       0.0|
+------------------------------+-------------+------------------------------+-----+----------+
only showing top 20 rows
```

## Model Performance:

## Logistic Regression Performance:

In [9]:
```
#printing the accuracy for test data, internal partition

evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
print(str(evaluator.evaluate(predictions)*100)+"%")
```

```
65.66287185364578%
```

## Naive Bayes Performance:

In [18]:
```
#printing the accuracy for test data, internal partition
evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
print(str(evaluator.evaluate(predictions)*100)+"%")
```

```
71.43992853147192%
```

## Random Forest Performance:

```
In [23]: #printing the accuracy for test data, internal partition
         evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
         print(str(evaluator.evaluate(predictions)*100)+"%")

         48.32796582928188%
```

**So we can see from the above performance output, the Naive Bayes Prediction model has the best accuracy which is 71.44 %**

**Step 6 (Testing) :**

Here we repeat Step 1 to collect a new dataset for each category. Next we perform the preprocessing using Apache Spark and then pass this dataset to our implemented model for its prediction.

Below are the model performance for the new test dataset:

**Logistic Regression Performance:**

```
In [16]: #printing accuracy for test data
         evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
         print(str(evaluator.evaluate(predictions)*100)+"%")

         79.79084922846847%
```

**Naive Bayes Performance:**

```
In [21]: evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
         print(str(evaluator.evaluate(predictions)*100)+"%")

         76.94682115429076%
```

**Random Forest Performance:**

```
In [25]: evaluator = MulticlassClassificationEvaluator(predictionCol="prediction")
         print(str(evaluator.evaluate(predictions)*100)+"%")

         51.71334098816666%
```

**So we can see from the above performance output, the Naive Bayes Prediction model has the best accuracy which is 79.80 %**

**Step 7 (Documentation) :**
The **Part-2-Spark-Text-Classifier.ipynb** notebook can be found in the code directory.

## DIRECTORY LAYOUT

There are 2 folders under **lab 3** directory:

- **code :**
    1. **Part 1:** Contains the data and the jupyter notebook for part 1.
    2. **Part 2**: Contains the data and the jupyter notebook for part 2.
       **A. articles-test.csv** and **articles-train.csv** contains the processed articles data which is later for for model prediction.
       **B. textcorpus** folder contains the raw articles in text format, which was collected in data collection phase using NYT API. Data has been stored is subdirectory where the subdirectory names represents the category of the article.

**C. NYTimesDataGen.ipynb:** used for NY Times article fetching and generation of the textcorpus data in the form of csv file.

**D. Part-2-Spark-Text-Classifier.ipynb:** contains the main python file for parsing the csv file and performing the article classification.

- **report:** Contains the ipynb files for Lab 3 and the report.

## SOFTWARE/HARDWARE USED

- Anaconda
- Jupyter Notebook
- Apache Spark
- Python 3.6  Environment based on Anaconda
- Ubuntu 17, Intel core i7 processor
- PySpark library

## REFERENCES

1. http://spark.apache.org/
2. https://benfradet.github.io/blog/2015/12/16/Exploring-spark.ml-with-the-Titanic-Kaggle-competition
3. https://6chaoran.wordpress.com/2016/08/13/__trashed/
4. https://datascienceplus.com/multi-class-text-classification-with-pyspark/