Deep Narayan Mishra

Data Analytics Pipeline using Apache Spark

Introduction/Purpose

News articles can be from different categories like sports, business, etc. In this project I am using machine learning to predict the category of articles. The first step is to train our model using the training set, test it, and finally evaluate the performance on some unknown article set.

Environment

The following environment was used:

- I have used Jupyter Notebook for the purpose of this project.
- Python is the primary programming language used.
- PySpark was used for using Python in the Spark environment
- PySparkSQL was used for creating a SparkSession, creating a data frame and manipulating the data frame. PySparkSQL provides a range of function to manage the data frame.
- mllib library was used extensively for various step involving machine learning. This includes the implementation of the models used, namely Logistic Regression and Random Forest Classfification

Implementation

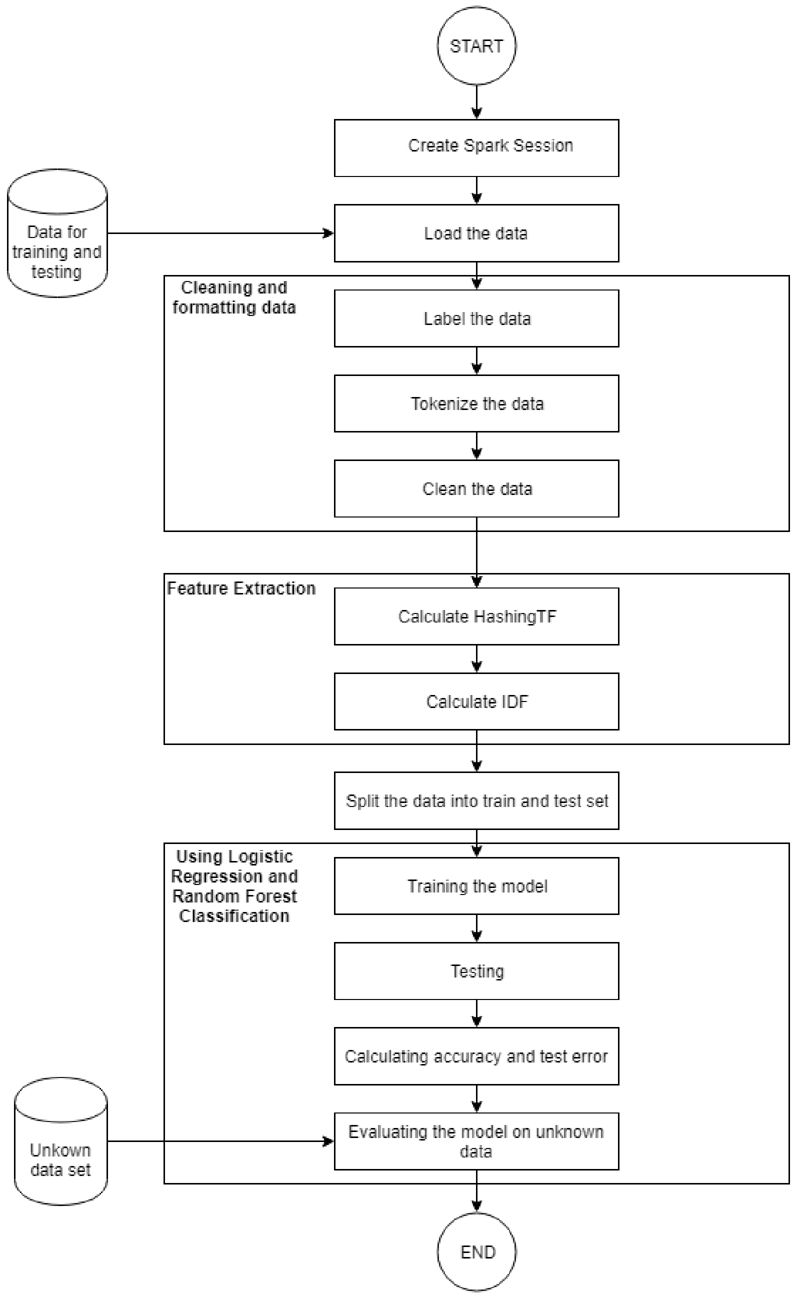
I started by gathering articles from NY Times. These are stored in different directories corresponding to the category in which the article belong to. The categories considered are:

- Business
- Movies
- Politics
- Science
- Sports

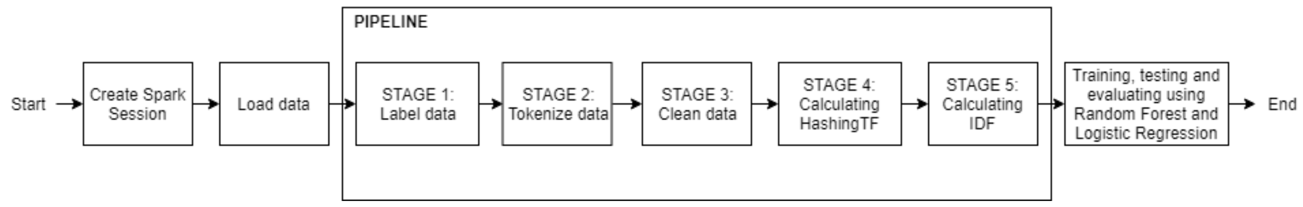
I tokenize the sentences in the articles into words. This is followed by cleaning the data to separate the meaningful data from the noise. I have done this by removing all the stop words in the text. Then process the data to find the TF and IDF of each of these articles. After that separate the articles into training and testing data. I have used the following machine learning models in our project to train and predict the article category type

- Random Forest Classification
- Logistic Regression

The following is the flow diagram for the project:


START
Create Spark Session
Data for training and testing
Load the data
Cleaning and formatting data
Label the data
Tokenize the data
Clean the data
Feature Extraction
Calculate HashingTF
Calculate IDF
Split the data into train and test set
Using Logistic Regression and Random Forest Classification
Training the model
Testing
Calculating accuracy and test error
Unkown data set
Evaluating the model on unknown data
END

The diagram below shows the pipeline structure that is used:



NY Times articles data collection:

To gather the articles, I used the API provided by NY Times. I used Python for achieving this. Using the API Key, search category, and date, I gathered the URL of the articles and saveed it. The response of the API call is in JSON format. I parsed the response to extract only the useful information (URL of the articles) while discarding the rest of the information. I store the URLs of the articles in five different files, one for each category.

After storing the URLs in the files, I scrape the articles using the links. Again, I have used Python for achieving this. The library BeautifulSoup used to get the HTML Page of the articles. I studied the structure of the HTML page returned to identify the body of the article. Then parsed the HTML page to get the body of the article. I store this information in a file. Each article is stored in a different file so that I can efficiently use Spark framework to process these files. Similar to the URLs, I store the articles in five different sets, one for each category.

Cleaning and formatting the data

I start by creating a Spark session. After this generated the data frame using the articles. The data frame has the following structure:

- file_name: this contains the name of the file in which the article was stored
- text: this is the actual content of the articles
- category: one of the five categories to which the article belongs

Then add labels to this data. These labels help in categorizing the article using the machine learning models.

This is followed by tokenization. Tokenization converts the sentences in the ‘text’ field of the data frame to a list of words.

Filtering of the words is done by removing all the stop words from the data. After doing this, I can focus on the meaningful data.

Feature Extraction

This is an important step in the process. In this step I rate the words to generate their importance in the document. This is done by using Term Frequency. HashingTF is a transformer which takes sets of terms (words) and converts them into fixed-length feature vectors. It uses hashing for the calculation of term frequency.

This is followed by calculating the inverse document frequency (IDF). The IDFModel takes feature vectors (created using HashingTF) and scales each column. That is, it down-weights columns which appear frequently in a text.

Data Partitioning

This is a simple step to split our data set into training and test sets. I have used the following parameters for the same:

- 70% training data set
- 30% testing data set

Training and testing

This is the most important step where I train and test models. The first model which I have chosen is Random Forest Classification. I train this model so that it can predict unknown data sets.

After this step, I test the model with our testing data set. The model predicts the labelsof the articles in the testing set and then I compare them to their actual labels. This process helps us in finding out the accuracy and error of the model using Multi class classification evaluator.

The entire procedure is applied to the second model, logistic regression. I train, test and then find the accuracy and error in the model.

Evaluating the model on unknown data set which is different from the test set

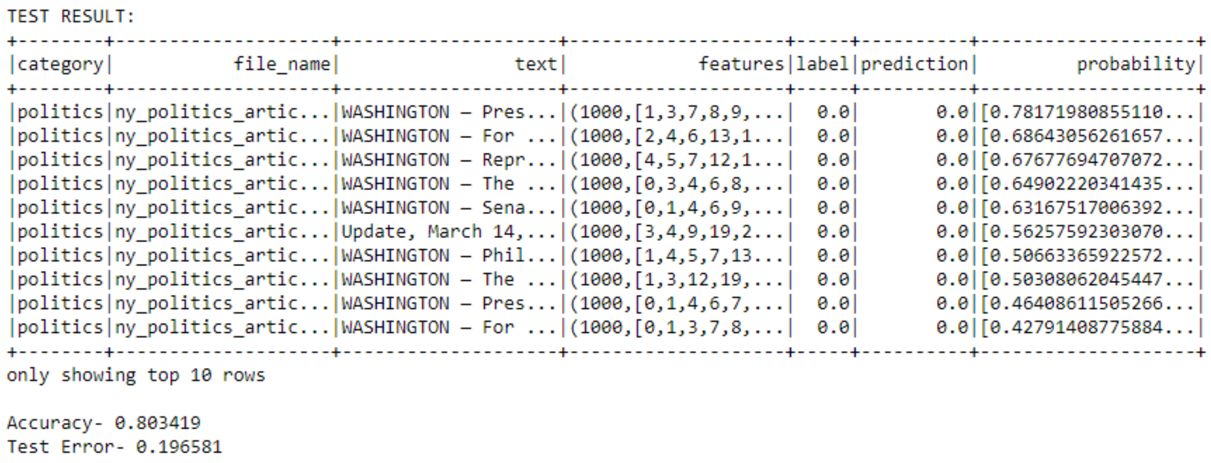
Now selected some unknown data set and try our model on this data set. This is done of each of the two models.

Using Pipeline

All the above steps can be automated using a pipeline. The pipeline will process the steps sequentially and produce the output.

Analysis and Results

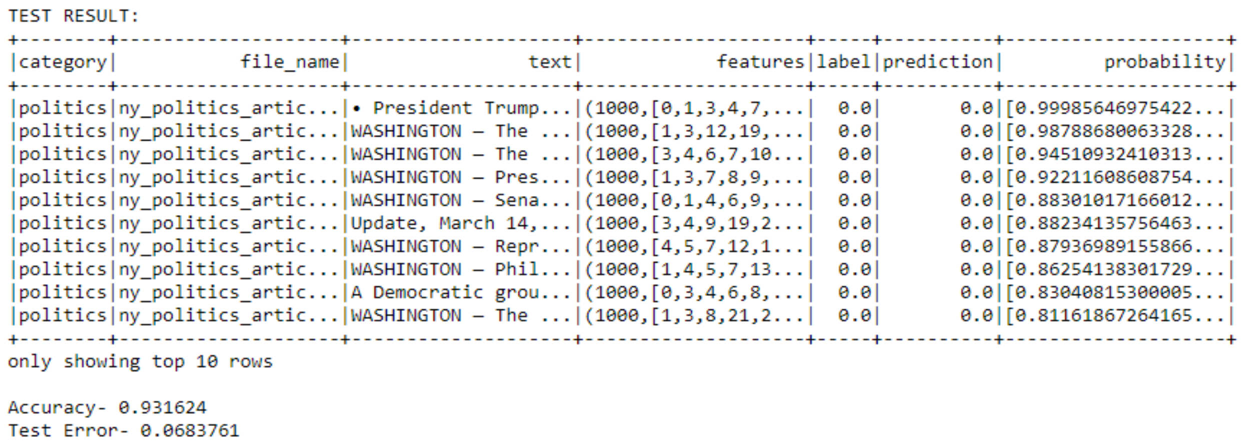
When running the random forest classification model on the test data, accuracy of 80.3% was achieved



```
TEST RESULT:
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
|category|           file_name|                text|            features|label|prediction|         probability|
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
|politics|ny_politics_artic...|WASHINGTON – Pres...|(1000,[1,3,7,8,9,...|  0.0|       0.0|[0.78171980855110...|
|politics|ny_politics_artic...|WASHINGTON – For ...|(1000,[2,4,6,13,1...|  0.0|       0.0|[0.68643056261657...|
|politics|ny_politics_artic...|WASHINGTON – Repr...|(1000,[4,5,7,12,1...|  0.0|       0.0|[0.67677694707072...|
|politics|ny_politics_artic...|WASHINGTON – The ...|(1000,[0,3,4,6,8,...|  0.0|       0.0|[0.64902220341435...|
|politics|ny_politics_artic...|WASHINGTON – Sena...|(1000,[0,1,4,6,9,...|  0.0|       0.0|[0.63167517006392...|
|politics|ny_politics_artic...|Update, March 14,...|(1000,[3,4,9,19,2...|  0.0|       0.0|[0.56257592303070...|
|politics|ny_politics_artic...|WASHINGTON – Phil...|(1000,[1,4,5,7,13...|  0.0|       0.0|[0.50663365922572...|
|politics|ny_politics_artic...|WASHINGTON – The ...|(1000,[1,3,12,19,...|  0.0|       0.0|[0.50308062045447...|
|politics|ny_politics_artic...|WASHINGTON – Pres...|(1000,[0,1,4,6,7,...|  0.0|       0.0|[0.46408611505266...|
|politics|ny_politics_artic...|WASHINGTON – For ...|(1000,[0,1,3,7,8,...|  0.0|       0.0|[0.42791408775884...|
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
only showing top 10 rows

Accuracy- 0.803419
Test Error- 0.196581
```

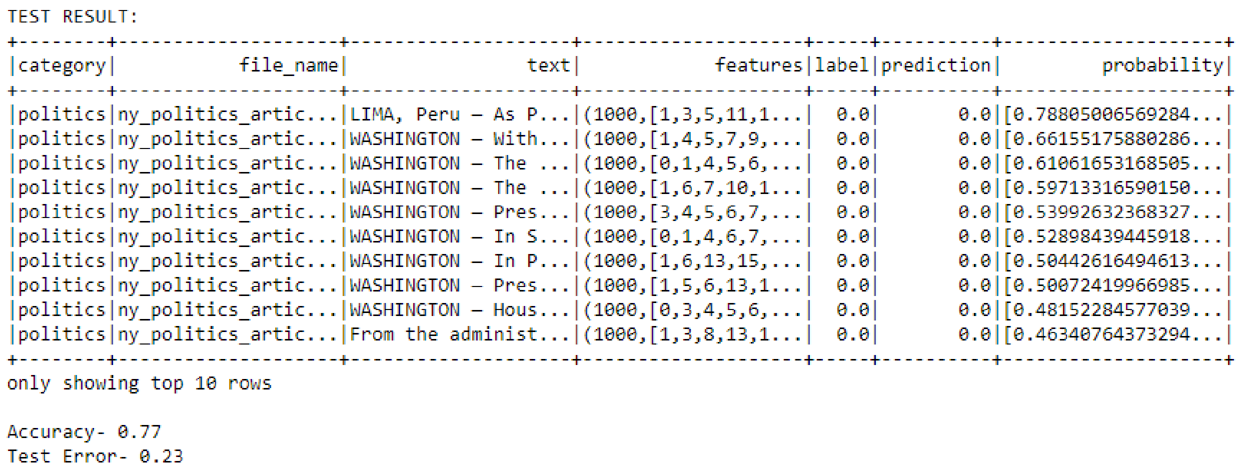
When running the logistic regression model on the test data, logistic accuracy of 93% was achieved



```
TEST RESULT:
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
|category|           file_name|                text|            features|label|prediction|         probability|
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
|politics|ny_politics_artic...|• President Trump...|(1000,[0,1,3,4,7,...|  0.0|       0.0|[0.99985646975422...|
|politics|ny_politics_artic...|WASHINGTON — The ...|(1000,[1,3,12,19,...|  0.0|       0.0|[0.98788680063328...|
|politics|ny_politics_artic...|WASHINGTON — The ...|(1000,[3,4,6,7,10...|  0.0|       0.0|[0.94510932410313...|
|politics|ny_politics_artic...|WASHINGTON — Pres...|(1000,[1,3,7,8,9,...|  0.0|       0.0|[0.92211608608754...|
|politics|ny_politics_artic...|WASHINGTON — Sena...|(1000,[0,1,4,6,9,...|  0.0|       0.0|[0.88301017166012...|
|politics|ny_politics_artic...|Update, March 14,...|(1000,[3,4,9,19,2...|  0.0|       0.0|[0.88234135756463...|
|politics|ny_politics_artic...|WASHINGTON — Repr...|(1000,[4,5,7,12,1...|  0.0|       0.0|[0.87936989155866...|
|politics|ny_politics_artic...|WASHINGTON — Phil...|(1000,[1,4,5,7,13...|  0.0|       0.0|[0.86254138301729...|
|politics|ny_politics_artic...|A Democratic grou...|(1000,[0,3,4,6,8,...|  0.0|       0.0|[0.83040815300005...|
|politics|ny_politics_artic...|WASHINGTON — The ...|(1000,[1,3,8,21,2...|  0.0|       0.0|[0.81161867264165...|
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
only showing top 10 rows

Accuracy- 0.931624
Test Error- 0.0683761
```

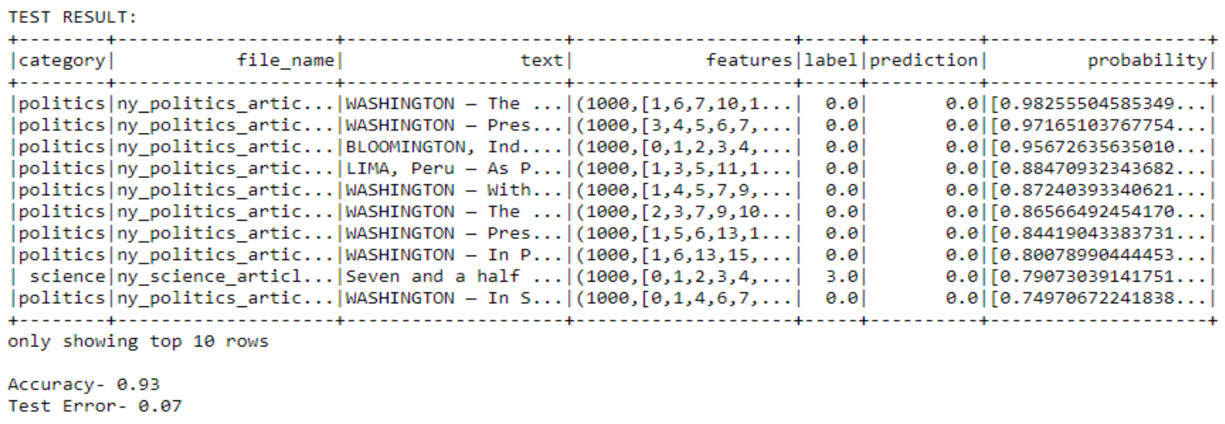
When running the random forest classification model on the unknown data, accuracy of 77% was achieved



```
TEST RESULT:
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
|category|           file_name|                text|            features|label|prediction|         probability|
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
|politics|ny_politics_artic...|LIMA, Peru — As P...|(1000,[1,3,5,11,1...|  0.0|       0.0|[0.78805006569284...|
|politics|ny_politics_artic...|WASHINGTON — With...|(1000,[1,4,5,7,9,...|  0.0|       0.0|[0.66155175880286...|
|politics|ny_politics_artic...|WASHINGTON — The ...|(1000,[0,1,4,5,6,...|  0.0|       0.0|[0.61061653168505...|
|politics|ny_politics_artic...|WASHINGTON — The ...|(1000,[1,6,7,10,1...|  0.0|       0.0|[0.59713316590150...|
|politics|ny_politics_artic...|WASHINGTON — Pres...|(1000,[3,4,5,6,7,...|  0.0|       0.0|[0.53992632368327...|
|politics|ny_politics_artic...|WASHINGTON — In S...|(1000,[0,1,4,6,7,...|  0.0|       0.0|[0.52898439445918...|
|politics|ny_politics_artic...|WASHINGTON — In P...|(1000,[1,6,13,15,...|  0.0|       0.0|[0.50442616494613...|
|politics|ny_politics_artic...|WASHINGTON — Pres...|(1000,[1,5,6,13,1...|  0.0|       0.0|[0.50072419966985...|
|politics|ny_politics_artic...|WASHINGTON — Hous...|(1000,[0,3,4,5,6,...|  0.0|       0.0|[0.48152284577039...|
|politics|ny_politics_artic...|From the administ...|(1000,[1,3,8,13,1...|  0.0|       0.0|[0.46340764373294...|
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
only showing top 10 rows

Accuracy- 0.77
Test Error- 0.23
```

When running the logistic regression model on the unknown data, accuracy of 93% was achieved



```
TEST RESULT:
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
|category|           file_name|                text|            features|label|prediction|         probability|
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
|politics|ny_politics_artic...|WASHINGTON — The ...|(1000,[1,6,7,10,1...|  0.0|       0.0|[0.98255504585349...|
|politics|ny_politics_artic...|WASHINGTON — Pres...|(1000,[3,4,5,6,7,...|  0.0|       0.0|[0.97165103767754...|
|politics|ny_politics_artic...|BLOOMINGTON, Ind....|(1000,[0,1,2,3,4,...|  0.0|       0.0|[0.95672635635010...|
|politics|ny_politics_artic...|LIMA, Peru — As P...|(1000,[1,3,5,11,1...|  0.0|       0.0|[0.88470932343682...|
|politics|ny_politics_artic...|WASHINGTON — With...|(1000,[1,4,5,7,9,...|  0.0|       0.0|[0.87240393340621...|
|politics|ny_politics_artic...|WASHINGTON — The ...|(1000,[2,3,7,9,10...|  0.0|       0.0|[0.86566492454170...|
|politics|ny_politics_artic...|WASHINGTON — Pres...|(1000,[1,5,6,13,1...|  0.0|       0.0|[0.84419043383731...|
|politics|ny_politics_artic...|WASHINGTON — In P...|(1000,[1,6,13,15,...|  0.0|       0.0|[0.80078990444453...|
| science|ny_science_articl...|Seven and a half ...|(1000,[0,1,2,3,4,...|  3.0|       0.0|[0.79073039141751...|
|politics|ny_politics_artic...|WASHINGTON — In S...|(1000,[0,1,4,6,7,...|  0.0|       0.0|[0.74970672241838...|
+--------+--------------------+--------------------+--------------------+-----+----------+--------------------+
only showing top 10 rows

Accuracy- 0.93
Test Error- 0.07
```

References:

https://creativedata.atlassian.net/wiki/spaces/SAP/pages/83237142/Pyspark+-+Tutorial+based+on+Titanic+Dataset
https://towardsdatascience.com/multi-class-text-classification-with-pyspark-7d78d022ed35
https://spark.apache.org/docs/2.2.0/ml-features.html
https://stackoverflow.com/questions/35205865/what-is-the-difference-between-hashingtf-and-countvectorizer-in-spark
https://www.tutorialkart.com/apache-spark/spark-mllib-tf-idf/
https://github.com/apache/spark/blob/master/examples/src/main/python/mllib/tf_idf_example.py