Data Intensive Computing
Lab 3 – Data Analytics Pipeline Using Apache Spark
REPORT

Charushi Nanwani / charushi / 50248736
Kaushik Panneerselvam / kpanneer / 50248889

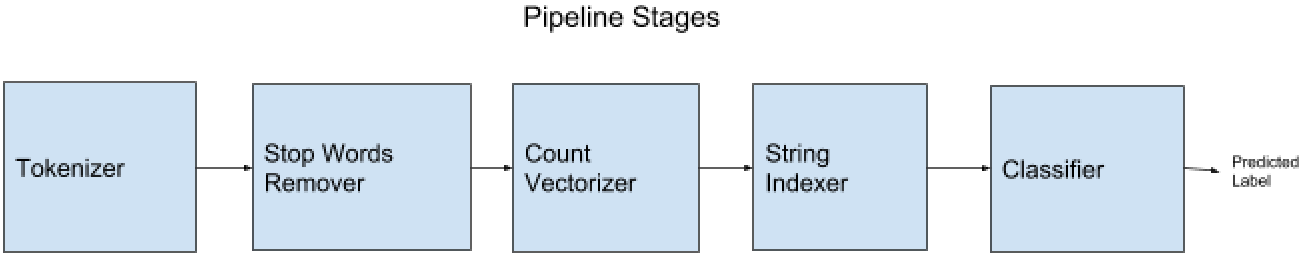
INTRODUCTION

In this lab Apache Spark Infrastructure was used to build a data processing pipeline to classify documents into different categories. The documents were processed and features were extracted, which were used in the classification task.

Infrastructure Used

Apache Spark - 2.1.0
Python 2.7
Mac OSX 10.13

PIPELINE



The pipeline consists of the following stages explained below:-

RegexTokenizer

The first step in the pipeline is to convert the document into a list of words. For this Pyspark's RegexTokenizer module ("pyspark.ml.feature.RegexTokenizer") is used. Here we use regular expression for whitespace to get tokens from the text.

StopWordsRemover

The list of words contains a lot of word like "is", "a", "the" etc, which are pretty common in all kinds of texts and documents. These set of words are called stopwords which when present confuses the model. So these words are removed using the StopWordsRemover module which takes in a list of stopwords and removes the words from the token list.

CountVectorizer

The CountVectorizer selects the top "vocabSize" words across the corpus ordered by the term frequency, i.e. the count of each tokens. We have specified the "minDF" parameter which specifies the minimum number of documents, the token should be present in.

StringIndexer
StringIndexer encodes the string label column to an index label. We have four categories(“business”,”politics”, ”sports”,and “travel”). So each category is mapped to an index value(0 for business, 1 for politics, 2 for sports, and 3 for travel).

Classifiers
We have used Logistic Regression and Naïve Bayes classifier for classifying the articles from NY Times.
Logistic Regression: -
Logistic Regression works by extracting some set of weighted features from the input, taking logs, and combining them linearly (meaning that each feature is multiplied by a weight and then added up)

Naïve Bayes: -
Naive Bayes classifier is a probabilistic classifier based on Bayes Rule and assumes strong independence between the feature variables. Here the probabilities are calculated for each class using Bayes rule and maximum likelihood estimation. The class with the maximum probability is assigned as the label for the document.

Confusion Matrix for Logistic Regression on the test dataset

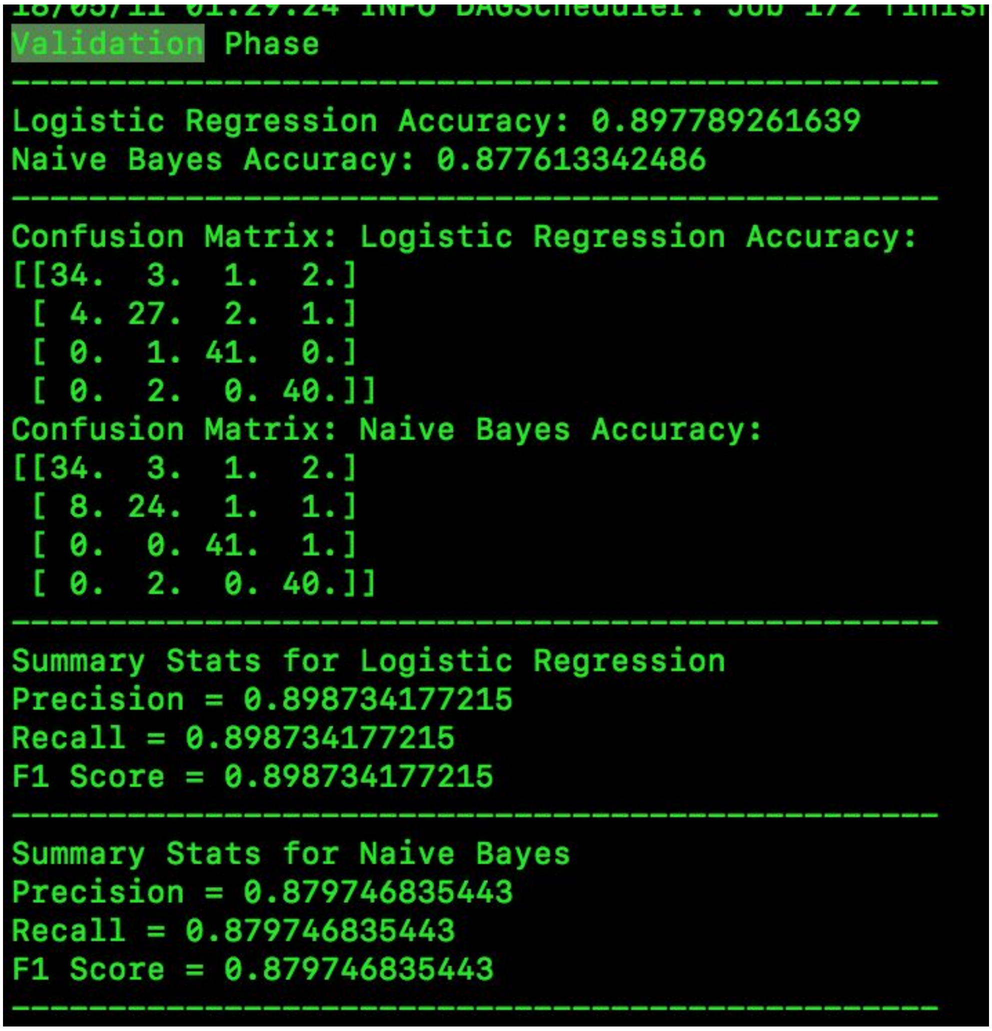
	Business	**Politics**	**Sports**	**Travel**
Business	**35**	**3**	**1**	**1**
Politics	**7**	**24**	**1**	**2**
Sports	**0**	**2**	**40**	**0**
Travel	**0**	**0**	**0**	**42**

Logistic Regression Accuracy: 0.88989131333

Confusion Matrix for Naive Bayes on the test dataset

	Business	**Politics**	**Sports**	**Travel**
Business	**34**	**3**	**2**	**1**
Politics	**8**	**24**	**1**	**1**
Sports	**0**	**2**	**40**	**0**
Travel	**0**	**0**	**1**	**41**

Naive Bayes Accuracy: 0.877613342486



```
18/05/11 01:29:24 INFO DAGScheduler: Job 172 finish
Validation Phase
------------------------------------------------
Logistic Regression Accuracy: 0.897789261639
Naive Bayes Accuracy: 0.877613342486
------------------------------------------------
Confusion Matrix: Logistic Regression Accuracy:
[[34.  3.  1.  2.]
 [ 4. 27.  2.  1.]
 [ 0.  1. 41.  0.]
 [ 0.  2.  0. 40.]]
Confusion Matrix: Naive Bayes Accuracy:
[[34.  3.  1.  2.]
 [ 8. 24.  1.  1.]
 [ 0.  0. 41.  1.]
 [ 0.  2.  0. 40.]]
------------------------------------------------
Summary Stats for Logistic Regression
Precision = 0.898734177215
Recall = 0.898734177215
F1 Score = 0.898734177215
------------------------------------------------
Summary Stats for Naive Bayes
Precision = 0.879746835443
Recall = 0.879746835443
F1 Score = 0.879746835443
------------------------------------------------
```

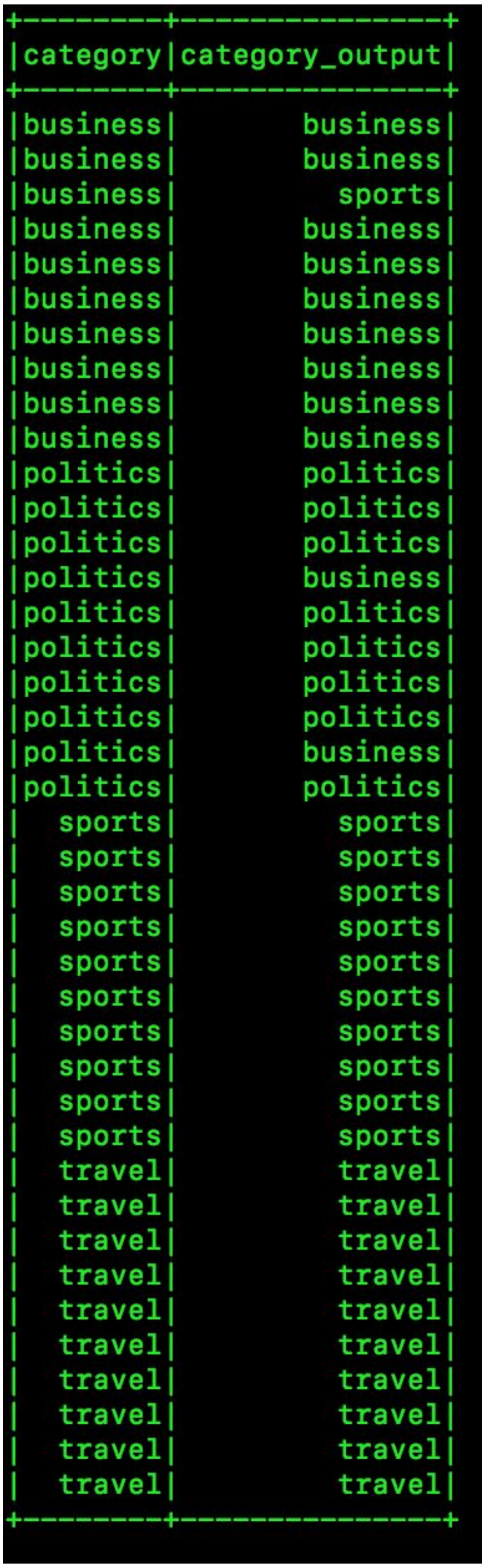
Unknown Test Dataset

We used the above classification models, and then predicted the categories of unknown dataset. The confusion matrix for both the classifiers are shown below:

Logistic Regression

	Business	**Politics**	**Sports**	**Travel**
Business	9	0	0	1
Politics	2	8	0	0
Sports	0	0	10	0
Travel	0	0	0	10

Logistic Regression Accuracy: 0.924603174603

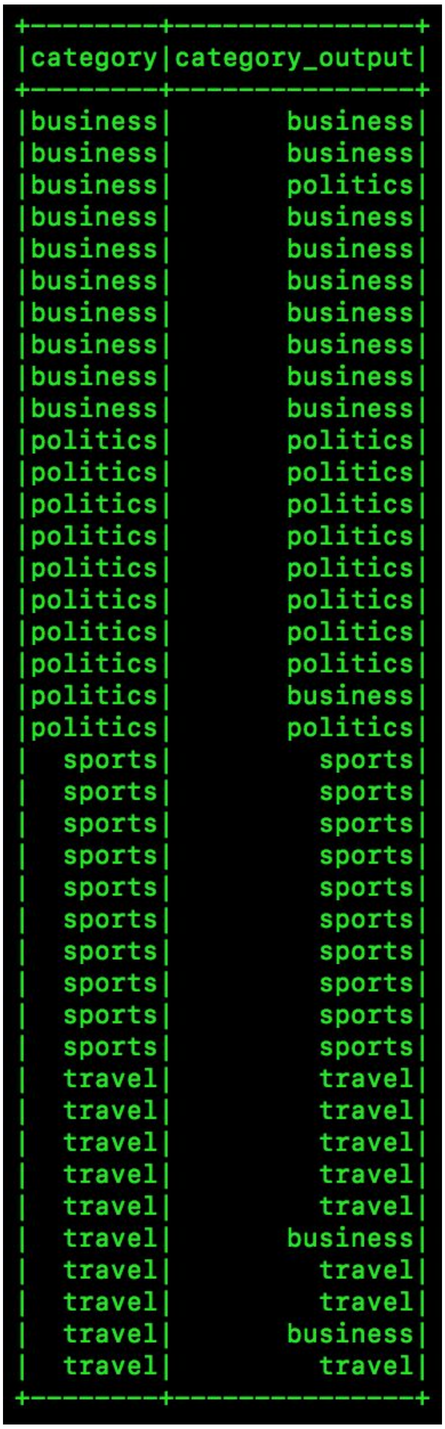


```
+--------+---------------+
|category|category_output|
+--------+---------------+
|business|       business|
|business|       business|
|business|         sports|
|business|       business|
|business|       business|
|business|       business|
|business|       business|
|business|       business|
|business|       business|
|business|       business|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       business|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       business|
|politics|       politics|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
+--------+---------------+
```

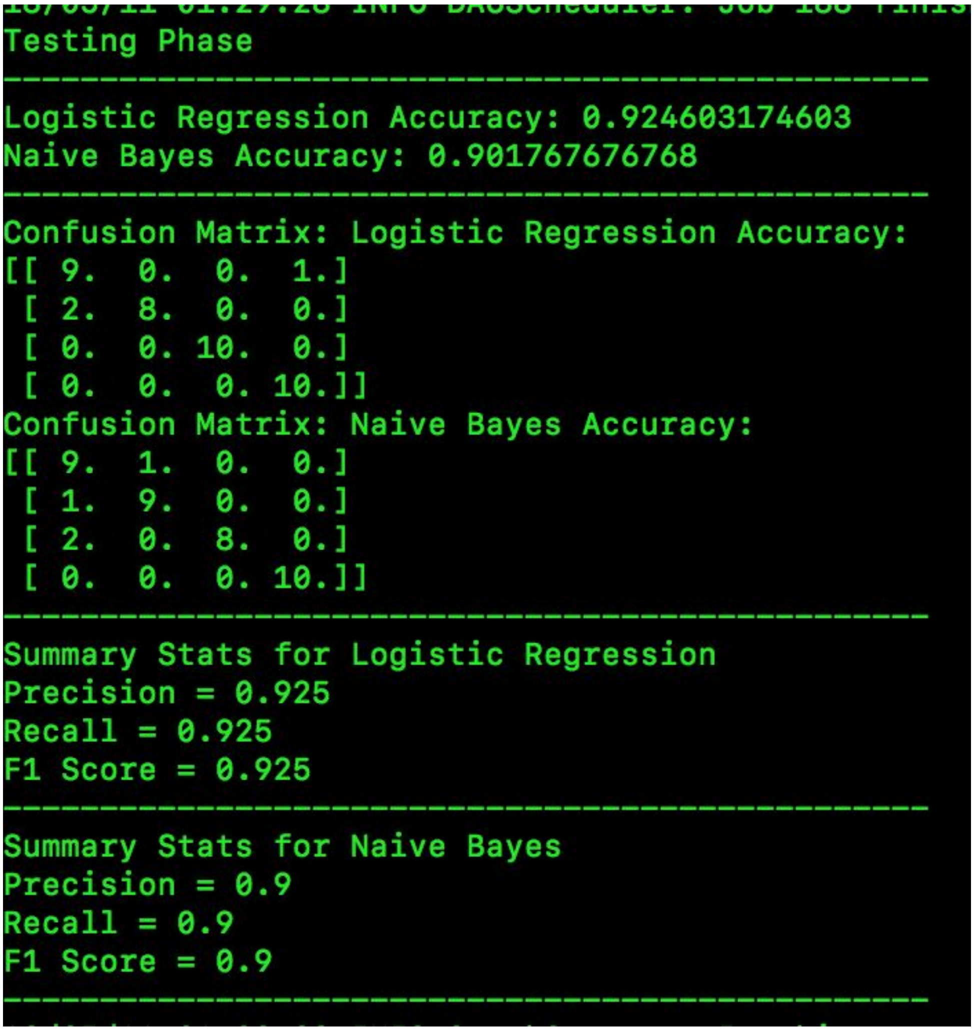
Naive Bayes

	Business	Politics	Sports	Travel
Business	9	0	0	1
Politics	1	9	0	0
Sports	2	0	8	0
Travel	0	0	0	10

Naive Bayes Accuracy: 0.901767676768



```
+--------+---------------+
|category|category_output|
+--------+---------------+
|business|       business|
|business|       business|
|business|       politics|
|business|       business|
|business|       business|
|business|       business|
|business|       business|
|business|       business|
|business|       business|
|business|       business|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       politics|
|politics|       business|
|politics|       politics|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  sports|         sports|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|         travel|
|  travel|       business|
|  travel|         travel|
|  travel|         travel|
|  travel|       business|
|  travel|         travel|
+--------+---------------+
```



```
Testing Phase
------------------------------------------------
Logistic Regression Accuracy: 0.924603174603
Naive Bayes Accuracy: 0.901767676768
------------------------------------------------
Confusion Matrix: Logistic Regression Accuracy:
[[ 9.   0.   0.   1.]
 [ 2.   8.   0.   0.]
 [ 0.   0. 10.   0.]
 [ 0.   0.   0. 10.]]
Confusion Matrix: Naive Bayes Accuracy:
[[ 9.   1.   0.   0.]
 [ 1.   9.   0.   0.]
 [ 2.   0.   8.   0.]
 [ 0.   0.   0. 10.]]
------------------------------------------------
Summary Stats for Logistic Regression
Precision = 0.925
Recall = 0.925
F1 Score = 0.925
------------------------------------------------
Summary Stats for Naive Bayes
Precision = 0.9
Recall = 0.9
F1 Score = 0.9
------------------------------------------------
```

Submission Content

- <LastName>Part1.ipynb
- train.csv
- part2.py
- fetch_content.py
- data

Steps to run the assignment

1. Please run the Part1.ipynb notebook. The data file used is ***"train.csv"***
2. We ran the ***"fetch_content.py"*** script to collect the articles from The New York Times.

   The data is in the **"data"** folder with separate directories for each categories.
3. Change the path of the data folder, in the ***"part2.py"*** file, and run it using ***spark-submit part2.py***