CSE 587 Lab 3 Report

Suhit Datta

Sourav Ranu

Data has been collected from NY Times using a Python code (***Data Collection.ipynb***) that has been created as a part of Lab2.

Four categories of data have been considered:

1) Politics
2) Sports
3) Business
4) Technology (own choice)

Multiple articles (greater than 50 articles) have been collected and are distributed into different folders with the same name.

This is contained in a folder called **DIC_Project_Data**

Name	Date modified	Type	Size
Business	5/5/2018 8:40 PM	File folder	
Politics	5/5/2018 8:40 PM	File folder	
Sports	5/5/2018 8:40 PM	File folder	
Technology	5/5/2018 8:40 PM	File folder	

Code has been written in PySpark (Python) (***Lab3_Code.ipynb***)

Data has been collected from the directory and stored in the form of a dictionary with key being the category name and the value being the content.

The elements of the dictionary are placed in a list in a specific format.

The code snippet exemplifies the process :

In [5]:

```
# Directory which consists the data
dir = "DIC_Project_Data"
```

In [6]:

```
#creates a subdirectory list
subDirNameList =[]
for root, dirs, files in os.walk(dir, topdown=False):
    for name in dirs:
        subDirNameList.append(os.path.join(root, name))

#creates a list of dictionary elements of each category type
listRDD = []
dictRDDElementsAsMap ={}
for eachFolder in subDirNameList:
    folderName = os.path.basename(eachFolder)
    rdd = sc.wholeTextFiles(eachFolder)
    listRDD.append(rdd)
    dictElement = rdd.collectAsMap()
    dictRDDElementsAsMap[folderName] = dictElement
```

In [7]:

```
# method to check if a string is blank or not
def isNotBlank (myString):
    if myString and myString.strip():
        #myString is not None AND myString is not empty or blank
        return True
    #myString is None OR myString is empty or blank
    return False
```

In [8]:

```
"""
this creates a list. Each element of a list is basically a dictionary :
    key = category and
    value = text content
"""
listAll = []
for name,v1 in dictRDDElementsAsMap.items():
    for key,value in v1.items():
        dataDic = {}
        if isNotBlank(value):
            dataDic['category'] = name
            dataDic['text'] = value
            listAll.append(dataDic)

```

In this case, **listAll** consists of the contents.

```
# this creates the dataframe from the list
FULLdf = spark.createDataFrame(listAll)
```

Once the data has been extracted from these folders, it is collected to form a Spark Dataframe with two columns:

1) Text
2) Category

This dataframe is split into Train dataframe and Test dataframe in the ratio 80:20

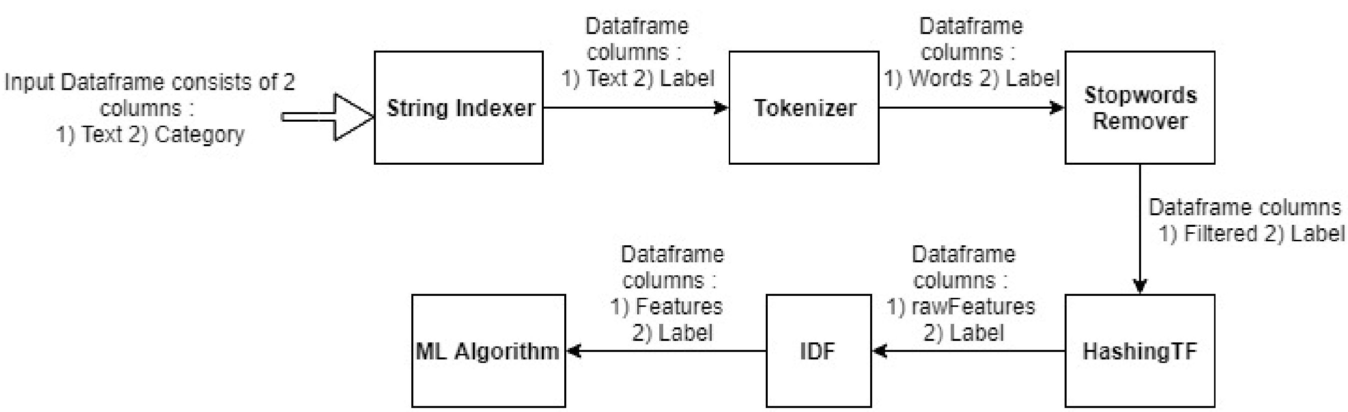
```
#splitting into training and test set
training, test = FULLdf.randomSplit([0.8, 0.2], seed=7)
training.cache()

```

```
DataFrame[category: string, text: string]
```

The model is trained using the Training Dataframe.

The various processes in the pipeline is mentioned as below:

Pipeline Flowchart:



The model thus created is used to test on the Test dataframe that has been separated earlier from the initial dataframe.

The following are the **ML Algorithms** used:

1) Random Forest

2) Naïve Bayes

3) Multiclass Logistic Regression

1) Code snippet for Random Forest

In [13]:

```
# applying Random Forest

from pyspark.ml import Pipeline
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.feature import HashingTF, Tokenizer , IDF , StringIndexer ,StopWordsRemover
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
from pyspark.mllib.util import MLUtils
from pyspark.mllib.evaluation import MulticlassMetrics
from pyspark.ml.classification import RandomForestClassifier
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Configure an ML pipeline
indexer = StringIndexer(inputCol="category", outputCol="label")
tokenizer = Tokenizer(inputCol="text", outputCol="words")
stopwordsRemover = StopWordsRemover(inputCol=tokenizer.getOutputCol(), outputCol="filtered")
hashingTF = HashingTF(inputCol=stopwordsRemover.getOutputCol(), outputCol="rawFeatures")
idf = IDF(inputCol=hashingTF.getOutputCol(), outputCol="features")
rf = RandomForestClassifier(labelCol="label", featuresCol="features", numTrees=15,  maxDepth=12)
pipeline = Pipeline(stages=[indexer, tokenizer,stopwordsRemover, hashingTF,idf, rf])


paramGrid = ParamGridBuilder() \
    .addGrid(hashingTF.numFeatures, [10, 100, 1000]) \
    .build()

crossval = CrossValidator(estimator=pipeline,
                          estimatorParamMaps=paramGrid,
                          evaluator=MulticlassClassificationEvaluator(),
                          numFolds=4)  # use 3+ folds in practice

# Run cross-validation, and choose the best set of parameters.
cvModelRF = crossval.fit(training)

# Make predictions on test documents. cvModel uses the best model found (lrModel).
predictions = cvModelRF.transform(test)
predictions.show()

# compute accuracy on the test set
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

The output obtained is as follows:

```
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
|category|                text|label|               words|            filtered|         rawFeatures|            features|      r
awPrediction|         probability|prediction|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
|Business| Good Tuesday. He...|  2.0|[, good, tuesday....|[, good, tuesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...| [2.0,
5.0,6.0,2.0]|[0.13333333333333...|       2.0|
|Business|Brady Hill used h...|  2.0|[brady, hill, use...|[brady, hill, use...|(1000,[0,1,2,3,8,...|(1000,[0,1,2,3,8,...| [4.0,
4.0,4.0,3.0]|[0.26666666666666...|       0.0|
|Business|Good Wednesday. H...|  2.0|[good, wednesday....|[good, wednesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...| [6.0,
5.0,4.0,0.0]|[0.4,0.3333333333...|       0.0|
|Business|I didn’t mean to ...|  2.0|[i, didn’t, mean,...|[didn’t, mean, it...|(1000,[0,2,4,6,7,...|(1000,[0,2,4,6,7,...| [6.0,
7.0,2.0,0.0]|[0.4,0.4666666666...|       1.0|
|Business|Long before dawn ...|  2.0|[long, before, da...|[long, dawn, wind...|(1000,[0,1,2,4,5,...|(1000,[0,1,2,4,5,...| [4.0,
5.0,3.0,3.0]|[0.26666666666666...|       1.0|
|Business|Microsoft, more t...|  2.0|[microsoft,, more...|[microsoft,, trad...|(1000,[0,1,8,13,1...|(1000,[0,1,8,13,1...| [2.0,
1.0,8.0,4.0]|[0.13333333333333...|       2.0|
|Business|SAN JUAN, P.R. — ...|  2.0|[san, juan,, p.r....|[san, juan,, p.r....|(1000,[1,8,10,12,...|(1000,[1,8,10,12,...| [4.0,
3.0,3.0,5.0]|[0.26666666666666...|       3.0|
|Business|SHANGHAI — Want t...|  2.0|[shanghai, —, wan...|[shanghai, —, wan...|(1000,[0,3,4,8,15...|(1000,[0,3,4,8,15...| [4.0,
3.0,7.0,1.0]|[0.26666666666666...|       2.0|
|Business|SÃO PAULO, Brazil...|  2.0|[são, paulo,, bra...|[são, paulo,, bra...|(1000,[3,4,7,10,2...|(1000,[3,4,7,10,2...| [3.0,
2.0,7.0,3.0]|[0.2,0.1333333333...|       2.0|
|Business|TOKYO — The Japan...|  2.0|[tokyo, —, the, j...|[tokyo, —, japane...|(1000,[8,10,16,18...|(1000,[8,10,16,18...| [1.0,
2.0,9.0,3.0]|[0.06666666666666...|       2.0|
|Business|The economy grew ...|  2.0|[the, economy, gr...|[economy, grew, a...|(1000,[3,7,8,10,1...|(1000,[3,7,8,10,1...| [4.0,
4.0,5.0,2.0]|[0.26666666666666...|       2.0|
|Business|Wall Street was p...|  2.0|[wall, street, wa...|[wall, street, pr...|(1000,[3,25,34,36...|(1000,[3,25,34,36...| [1.0,
2.0,7.0,5.0]|[0.06666666666666...|       2.0|
|Politics|On Day 10 of a sc...|  0.0|[on, day, 10, of,...|[day, 10, scandal...|(1000,[0,1,2,5,6,...|(1000,[0,1,2,5,6,...|[10.0,
4.0,0.0,1.0]|[0.66666666666666...|       0.0|
|Politics|On Saturday, Rebe...|  0.0|[on, saturday,, r...|[saturday,, rebec...|(1000,[0,1,4,6,7,...|(1000,[0,1,4,6,7,...| [8.0,
3.0,3.0,1.0]|[0.53333333333333...|       0.0|
|Politics|PALM BEACH, Fla. ...|  0.0|[palm, beach,, fl...|[palm, beach,, fl...|(1000,[0,3,6,7,10...|(1000,[0,3,6,7,10...| [6.0,
3.0,6.0,0.0]|   [0.4,0.2,0.4,0.0]|       0.0|
|Politics|PHOENIX — Melinda...|  0.0|[phoenix, —, meli...|[phoenix, —, meli...|(1000,[0,1,3,4,7,...|(1000,[0,1,3,4,7,...| [8.0,
2.0,4.0,1.0]|[0.53333333333333...|       0.0|
|Politics|The driver was dr...|  0.0|[the, driver, was...|[driver, drunk,, ...|(1000,[1,2,3,4,5,...|(1000,[1,2,3,4,5,...| [8.0,
1.0,5.0,1.0]|[0.53333333333333...|       0.0|
|Politics|WASHINGTON — A fe...|  0.0|[washington, —, a...|[washington, —, h...|(1000,[0,1,3,4,5,...|(1000,[0,1,3,4,5,...| [4.0,
4.0,3.0,4.0]|[0.26666666666666...|       0.0|
|Politics|WASHINGTON — Even...|  0.0|[washington, —, e...|[washington, —, e...|(1000,[1,2,5,7,15...|(1000,[1,2,5,7,15...| [3.0,
3.0,6.0,3.0]|   [0.2,0.2,0.4,0.2]|       2.0|
|Politics|WASHINGTON — Harr...|  0.0|[washington, —, h...|[washington, —, h...|(1000,[0,2,3,13,1...|(1000,[0,2,3,13,1...| [5.0,
6.0,1.0,3.0]|[0.33333333333333...|       1.0|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
only showing top 20 rows

Test set accuracy = 0.7021276595744681
```

Accuracy of **70.2127 %** is obtained for **Random Forest** on the Test dataframe.

2) Code snippet for Naive Bayes :

In [14]:

```
# applying Naive Bayes

from pyspark.ml import Pipeline
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.feature import HashingTF, Tokenizer , IDF , StringIndexer ,StopWordsRemover
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
from pyspark.mllib.util import MLUtils
from pyspark.mllib.evaluation import MulticlassMetrics
from pyspark.ml.classification import NaiveBayes
from pyspark.ml.evaluation import MulticlassClassificationEvaluator

# Configure an ML pipeline
indexer = StringIndexer(inputCol="category", outputCol="label")
tokenizer = Tokenizer(inputCol="text", outputCol="words")
stopwordsRemover = StopWordsRemover(inputCol=tokenizer.getOutputCol(), outputCol="filtered")
hashingTF = HashingTF(inputCol=stopwordsRemover.getOutputCol(), outputCol="rawFeatures")
idf = IDF(inputCol=hashingTF.getOutputCol(), outputCol="features")
nb = NaiveBayes(smoothing=1.0, modelType="multinomial")
pipeline = Pipeline(stages=[indexer, tokenizer,stopwordsRemover, hashingTF,idf, nb])


paramGrid = ParamGridBuilder() \
    .addGrid(hashingTF.numFeatures, [10, 100, 1000]) \
    .build()

crossval = CrossValidator(estimator=pipeline,
                          estimatorParamMaps=paramGrid,
                          evaluator=MulticlassClassificationEvaluator(),
                          numFolds=4)  # use 3+ folds in practice

# Run cross-validation, and choose the best set of parameters.
cvModelNB = crossval.fit(training)

# Make predictions on test documents. cvModel uses the best model found (lrModel).
predictions = cvModelNB.transform(test)
predictions.show()

# compute accuracy on the test set
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))

```

The output obtained is as follows:

```
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
|category|                text|label|               words|            filtered|         rawFeatures|            features|
rawPrediction|         probability|prediction|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
|Business| Good Tuesday. He...|  2.0|[, good, tuesday....|[, good, tuesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...|[-750
6.8131448156...|[1.61820182577732...|       2.0|
|Business|Brady Hill used h...|  2.0|[brady, hill, use...|[brady, hill, use...|(1000,[0,1,2,3,8,...|(1000,[0,1,2,3,8,...|[-478
5.4573792874...|[2.64678005613299...|       1.0|
|Business|Good Wednesday. H...|  2.0|[good, wednesday....|[good, wednesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...|[-1073
2.631692611...|[3.65314603578155...|       2.0|
|Business|I didn’t mean to ...|  2.0|[i, didn’t, mean,...|[didn’t, mean, it...|(1000,[0,2,4,6,7,...|(1000,[0,2,4,6,7,...|[-386
0.7644305603...|[5.58602358666721...|       1.0|
|Business|Long before dawn ...|  2.0|[long, before, da...|[long, dawn, wind...|(1000,[0,1,2,4,5,...|(1000,[0,1,2,4,5,...|[-703
6.6996978978...|[5.16260372878484...|       1.0|
|Business|Microsoft, more t...|  2.0|[microsoft,, more...|[microsoft,, trad...|(1000,[0,1,8,13,1...|(1000,[0,1,8,13,1...|[-196
2.6681078204...|[1.58569753097225...|       2.0|
|Business|SAN JUAN, P.R. — ...|  2.0|[san, juan,, p.r....|[san, juan,, p.r....|(1000,[1,8,10,12,...|(1000,[1,8,10,12,...|[-583
8.3449909497...|[5.13954664577655...|       1.0|
|Business|SHANGHAI — Want t...|  2.0|[shanghai, —, wan...|[shanghai, —, wan...|(1000,[0,3,4,8,15...|(1000,[0,3,4,8,15...|[-385
4.7002334727...|[2.04971544668225...|       2.0|
|Business|SÃO PAULO, Brazil...|  2.0|[são, paulo,, bra...|[são, paulo,, bra...|(1000,[3,4,7,10,2...|(1000,[3,4,7,10,2...|[-207
4.6773081437...|[1.87977674991846...|       2.0|
|Business|TOKYO — The Japan...|  2.0|[tokyo, —, the, j...|[tokyo, —, japane...|(1000,[8,10,16,18...|(1000,[8,10,16,18...|[-128
0.7713860671...|[3.65926739649113...|       2.0|
|Business|The economy grew ...|  2.0|[the, economy, gr...|[economy, grew, a...|(1000,[3,7,8,10,1...|(1000,[3,7,8,10,1...|[-353
0.7408637991...|[1.38550476759380...|       2.0|
|Business|Wall Street was p...|  2.0|[wall, street, wa...|[wall, street, pr...|(1000,[3,25,34,36...|(1000,[3,25,34,36...|[-898.
67269335400...|[0.00715648644266...|       2.0|
|Politics|On Day 10 of a sc...|  0.0|[on, day, 10, of,...|[day, 10, scandal...|(1000,[0,1,2,5,6,...|(1000,[0,1,2,5,6,...|[-402
8.5399298969...|[1.0,8.6713991623...|       0.0|
|Politics|On Saturday, Rebe...|  0.0|[on, saturday,, r...|[saturday,, rebec...|(1000,[0,1,4,6,7,...|(1000,[0,1,4,6,7,...|[-472
8.5781032164...|[0.99999999995944...|       0.0|
|Politics|PALM BEACH, Fla. ...|  0.0|[palm, beach,, fl...|[palm, beach,, fl...|(1000,[0,3,6,7,10...|(1000,[0,3,6,7,10...|[-378
7.1217369171...|[0.99999930512375...|       0.0|
|Politics|PHOENIX — Melinda...|  0.0|[phoenix, —, meli...|[phoenix, —, meli...|(1000,[0,1,3,4,7,...|(1000,[0,1,3,4,7,...|[-550
7.4545474465...|[1.0,1.0859097859...|       0.0|
|Politics|The driver was dr...|  0.0|[the, driver, was...|[driver, drunk,, ...|(1000,[1,2,3,4,5,...|(1000,[1,2,3,4,5,...|[-400
4.0763441880...|[0.99999998917304...|       0.0|
|Politics|WASHINGTON — A fe...|  0.0|[washington, —, a...|[washington, —, h...|(1000,[0,1,3,4,5,...|(1000,[0,1,3,4,5,...|[-502
8.4888714803...|[1.0,1.0514520189...|       0.0|
|Politics|WASHINGTON — Even...|  0.0|[washington, —, e...|[washington, —, e...|(1000,[1,2,5,7,15...|(1000,[1,2,5,7,15...|[-379
5.4997762411...|[1.0,8.2206181086...|       0.0|
|Politics|WASHINGTON — Harr...|  0.0|[washington, —, h...|[washington, —, h...|(1000,[0,2,3,13,1...|(1000,[0,2,3,13,1...|[-296
6.0111529850...|[1.0,1.5051879605...|       0.0|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
only showing top 20 rows

Test set accuracy = 0.851063829787234
```

Accuracy of **85.1063 %** is obtained for **Naïve Bayes** on the Test dataframe.

3) Code snippet for Logistic Regression

In [16]:

```
# applying Logistic Regression

from pyspark.ml import Pipeline
from pyspark.ml.evaluation import BinaryClassificationEvaluator
from pyspark.ml.feature import HashingTF, Tokenizer , IDF , StringIndexer ,StopWordsRemover
from pyspark.ml.tuning import CrossValidator, ParamGridBuilder
from pyspark.mllib.util import MLUtils
from pyspark.mllib.evaluation import MulticlassMetrics
from pyspark.ml.evaluation import MulticlassClassificationEvaluator
from pyspark.ml.classification import LogisticRegression

# Configure an ML pipeline
indexer = StringIndexer(inputCol="category", outputCol="label")
tokenizer = Tokenizer(inputCol="text", outputCol="words")
stopwordsRemover = StopWordsRemover(inputCol=tokenizer.getOutputCol(), outputCol="filtered")
hashingTF = HashingTF(inputCol=stopwordsRemover.getOutputCol(), outputCol="rawFeatures")
idf = IDF(inputCol=hashingTF.getOutputCol(), outputCol="features")
lr = LogisticRegression(maxIter=10, regParam=0.3, elasticNetParam=0.8)
pipeline = Pipeline(stages=[indexer, tokenizer,stopwordsRemover, hashingTF,idf, lr])


paramGrid = ParamGridBuilder() \
    .addGrid(hashingTF.numFeatures, [10, 100, 1000]) \
    .addGrid(lr.regParam, [0.1, 0.01]) \
    .build()

crossval = CrossValidator(estimator=pipeline,
                          estimatorParamMaps=paramGrid,
                          evaluator=MulticlassClassificationEvaluator(),
                          numFolds=4)  # use 3+ folds in practice

# Run cross-validation, and choose the best set of parameters.
cvModelLR = crossval.fit(training)

# Make predictions on test documents. cvModel uses the best model found (lrModel).
predictions = cvModelLR.transform(test)
predictions.show()

# compute accuracy on the test set
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Test set accuracy = " + str(accuracy))
```

The output obtained is as follows:

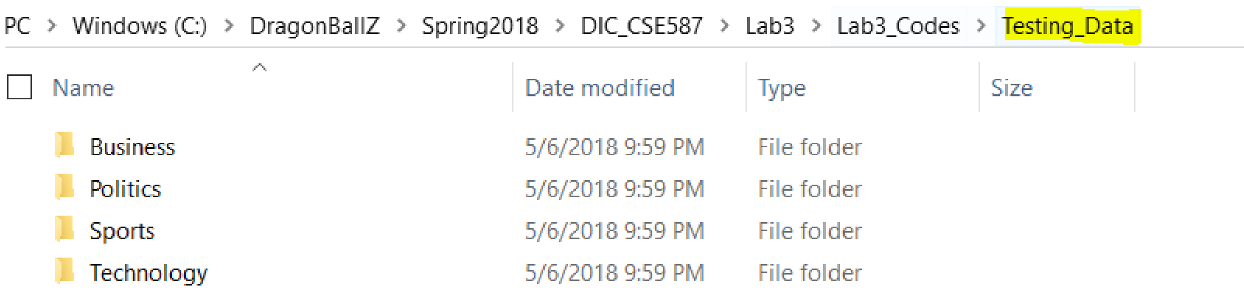
```
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
|category|                text|label|               words|            filtered|         rawFeatures|            features|
rawPrediction|         probability|prediction|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
|Business| Good Tuesday. He...|  2.0|[, good, tuesday....|[, good, tuesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...|[-0.74
06603218313...|[0.00401937982547...|       2.0|
|Business|Brady Hill used h...|  2.0|[brady, hill, use...|[brady, hill, use...|(1000,[0,1,2,3,8,...|(1000,[0,1,2,3,8,...|[-0.11
55582931381...|[0.08386980169683...|       2.0|
|Business|Good Wednesday. H...|  2.0|[good, wednesday....|[good, wednesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...|[0.511
01972775904...|[0.01807309236991...|       2.0|
|Business|I didn’t mean to ...|  2.0|[i, didn’t, mean,...|[didn’t, mean, it...|(1000,[0,2,4,6,7,...|(1000,[0,2,4,6,7,...|[-0.36
44393046674...|[0.06680863860525...|       1.0|
|Business|Long before dawn ...|  2.0|[long, before, da...|[long, dawn, wind...|(1000,[0,1,2,4,5,...|(1000,[0,1,2,4,5,...|[0.131
12565541822...|[0.07124476241056...|       1.0|
|Business|Microsoft, more t...|  2.0|[microsoft,, more...|[microsoft,, trad...|(1000,[0,1,8,13,1...|(1000,[0,1,8,13,1...|[-0.66
30936518893...|[0.05464246643182...|       1.0|
|Business|SAN JUAN, P.R. — ...|  2.0|[san, juan,, p.r....|[san, juan,, p.r....|(1000,[1,8,10,12,...|(1000,[1,8,10,12,...|[0.651
30541633843...|[0.14458847592453...|       1.0|
|Business|SHANGHAI — Want t...|  2.0|[shanghai, —, wan...|[shanghai, —, wan...|(1000,[0,3,4,8,15...|(1000,[0,3,4,8,15...|[0.078
11656860139...|[0.02531448555066...|       2.0|
|Business|SÃO PAULO, Brazil...|  2.0|[são, paulo,, bra...|[são, paulo,, bra...|(1000,[3,4,7,10,2...|(1000,[3,4,7,10,2...|[-0.03
17095050174...|[0.09078047173183...|       2.0|
|Business|TOKYO — The Japan...|  2.0|[tokyo, —, the, j...|[tokyo, —, japane...|(1000,[8,10,16,18...|(1000,[8,10,16,18...|[-0.44
01049258937...|[0.13687667173840...|       2.0|
|Business|The economy grew ...|  2.0|[the, economy, gr...|[economy, grew, a...|(1000,[3,7,8,10,1...|(1000,[3,7,8,10,1...|[0.154
29064916140...|[0.04279056313077...|       2.0|
|Business|Wall Street was p...|  2.0|[wall, street, wa...|[wall, street, pr...|(1000,[3,25,34,36...|(1000,[3,25,34,36...|[0.264
61591218756...|[0.25414026762762...|       2.0|
|Politics|On Day 10 of a sc...|  0.0|[on, day, 10, of,...|[day, 10, scandal...|(1000,[0,1,2,5,6,...|(1000,[0,1,2,5,6,...|[3.899
12028926673...|[0.95397863981493...|       0.0|
|Politics|On Saturday, Rebe...|  0.0|[on, saturday,, r...|[saturday,, rebec...|(1000,[0,1,4,6,7,...|(1000,[0,1,4,6,7,...|[1.824
22692665061...|[0.59151645481913...|       0.0|
|Politics|PALM BEACH, Fla. ...|  0.0|[palm, beach,, fl...|[palm, beach,, fl...|(1000,[0,3,6,7,10...|(1000,[0,3,6,7,10...|[1.883
53047710433...|[0.54602602592313...|       0.0|
|Politics|PHOENIX — Melinda...|  0.0|[phoenix, —, meli...|[phoenix, —, meli...|(1000,[0,1,3,4,7,...|(1000,[0,1,3,4,7,...|[3.331
61028291339...|[0.85195987171896...|       0.0|
|Politics|The driver was dr...|  0.0|[the, driver, was...|[driver, drunk,, ...|(1000,[1,2,3,4,5,...|(1000,[1,2,3,4,5,...|[0.953
98252991273...|[0.24521930687820...|       2.0|
|Politics|WASHINGTON — A fe...|  0.0|[washington, —, a...|[washington, —, h...|(1000,[0,1,3,4,5,...|(1000,[0,1,3,4,5,...|[4.623
68863310841...|[0.97344060495980...|       0.0|
|Politics|WASHINGTON — Even...|  0.0|[washington, —, e...|[washington, —, e...|(1000,[1,2,5,7,15...|(1000,[1,2,5,7,15...|[3.378
81161366650...|[0.92797444781456...|       0.0|
|Politics|WASHINGTON — Harr...|  0.0|[washington, —, h...|[washington, —, h...|(1000,[0,2,3,13,1...|(1000,[0,2,3,13,1...|[2.541
05806776414...|[0.80748258843656...|       0.0|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
only showing top 20 rows

Test set accuracy = 0.851063829787234
```

Accuracy of **85.10638 %** is obtained for Logistic Regression on the Test dataframe.

Running the Model on Unknown Test Data

A similar approach for data collection has been taken to collect Test data. Only this time lesser number of articles have been collected as compared to the previous step.

The data has been randomly collected. A folder Testing_Data has been created which consists of the subfolders as shown below.



RUNNING THE MODELS GENERATED ON A TEST DATA WHICH HAS BEEN SEPARATELY OBTAINED

In [17]:

```
# Directory which consists the test data
test_dir = "Testing_Data"
```

In [18]:

```
#creates a subdirectory list
subDirNameList =[]
for root, dirs, files in os.walk(dir, topdown=False):
    for name in dirs:
        subDirNameList.append(os.path.join(root, name))

#creates a list of dictionary elements of each category type
listRDD = []
dictRDDElementsAsMap ={}
for eachFolder in subDirNameList:
    folderName = os.path.basename(eachFolder)
    rdd = sc.wholeTextFiles(eachFolder)
    listRDD.append(rdd)
    dictElement = rdd.collectAsMap()
    dictRDDElementsAsMap[folderName] = dictElement
```

In [19]:

```
"""
this creates a test list. Each element of a list is basically a dictionary :
    key = category and
    value = text content
"""
testlistAll = []
for name,v1 in dictRDDElementsAsMap.items():
    for key,value in v1.items():
        dataDic = {}
        if isNotBlank(value):
            dataDic['category'] = name
            dataDic['text'] = value
            testlistAll.append(dataDic)

```

In [20]:

```
# this creates the test dataframe from the testlistAll
testDf = spark.createDataFrame(testlistAll)
```

The same ML algorithms have been run in the following order:

1) Random Forest

Input snippet is as follows:

```
In [24]:
# Running Random Forest on this new Test Data

predictions = cvModelRF.transform(testDf)
predictions.show()

# compute accuracy on the test set
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Accuracy for Random Forest on new test data = " + str(accuracy))

```

Output obtained is as follows:

```
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
-----------+--------------------+----------+
|category|                text|label|               words|            filtered|         rawFeatures|            features|     r
awPrediction|         probability|prediction|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
-----------+--------------------+----------+
|Business|One of the few re...|  2.0|[one, of, the, fe...|[one, remaining, ...|(1000,[0,2,3,9,11...|(1000,[0,2,3,9,11...| [1.0,
1.0,9.0,4.0]|[0.06666666666666...|       2.0|
|Business|Deutsche Bank sai...|  2.0|[deutsche, bank, ...|[deutsche, bank, ...|(1000,[2,8,12,20,...|(1000,[2,8,12,20,...|[0.0,
2.0,12.0,1.0]|[0.0,0.1333333333...|       2.0|
|Business|Spotify is a hit....|  2.0|[spotify, is, a, ...|[spotify, hit.on,...|(1000,[0,5,10,16,...|(1000,[0,5,10,16,...|[0.0,
1.0,13.0,1.0]|[0.0,0.0666666666...|       2.0|
|Business| Good Tuesday. He...|  2.0|[, good, tuesday....|[, good, tuesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...| [2.0,
5.0,6.0,2.0]|[0.13333333333333...|       2.0|
|Business|After days of som...|  2.0|[after, days, of,...|[days, sometimes,...|(1000,[1,12,13,14...|(1000,[1,12,13,14...|[0.0,
1.0,13.0,1.0]|[0.0,0.0666666666...|       2.0|
|Business|Good Thursday. He...|  2.0|[good, thursday.,...|[good, thursday.,...|(1000,[0,1,2,3,5,...|(1000,[0,1,2,3,5,...| [2.0,
3.0,8.0,2.0]|[0.13333333333333...|       2.0|
|Business|ABINGTON, Pa. — S...|  2.0|[abington,, pa., ...|[abington,, pa., ...|(1000,[0,1,6,8,13...|(1000,[0,1,6,8,13...| [0.0,
1.0,9.0,5.0]|[0.0,0.0666666666...|       2.0|
|Business|Wall Street was p...|  2.0|[wall, street, wa...|[wall, street, pr...|(1000,[3,25,34,36...|(1000,[3,25,34,36...| [1.0,
2.0,7.0,5.0]|[0.06666666666666...|       2.0|
|Business|SAN JUAN, P.R. — ...|  2.0|[san, juan,, p.r....|[san, juan,, p.r....|(1000,[1,8,10,12,...|(1000,[1,8,10,12,...| [4.0,
3.0,3.0,5.0]|[0.26666666666666...|       3.0|
|Business|The Labor Departm...|  2.0|[the, labor, depa...|[labor, departmen...|(1000,[0,1,2,11,1...|(1000,[0,1,2,11,1...|[0.0,
1.0,13.0,1.0]|[0.0,0.0666666666...|       2.0|
|Business|The sell-off in s...|  2.0|[the, sell-off, i...|[sell-off, stocks...|(1000,[8,25,30,34...|(1000,[8,25,30,34...| [0.0,
1.0,9.0,5.0]|[0.0,0.0666666666...|       2.0|
|Business|SHENZHEN, China —...|  2.0|[shenzhen,, china...|[shenzhen,, china...|(1000,[3,4,6,10,1...|(1000,[3,4,6,10,1...| [2.0,
3.0,9.0,1.0]|[0.13333333333333...|       2.0|
|Business|The stock market ...|  2.0|[the, stock, mark...|[stock, market, f...|(1000,[1,3,13,16,...|(1000,[1,3,13,16,...|[0.0,
2.0,12.0,1.0]|[0.0,0.1333333333...|       2.0|
|Business|I didn’t mean to ...|  2.0|[i, didn’t, mean,...|[didn’t, mean, it...|(1000,[0,2,4,6,7,...|(1000,[0,2,4,6,7,...| [6.0,
7.0,2.0,0.0]|[0.4,0.4666666666...|       1.0|
|Business|Brady Hill used h...|  2.0|[brady, hill, use...|[brady, hill, use...|(1000,[0,1,2,3,8,...|(1000,[0,1,2,3,8,...| [4.0,
4.0,4.0,3.0]|[0.26666666666666...|       0.0|
|Business|FRANKFURT — Mario...|  2.0|[frankfurt, —, ma...|[frankfurt, —, ma...|(1000,[1,3,5,11,1...|(1000,[1,3,5,11,1...|[3.0,
1.0,10.0,1.0]|[0.2,0.0666666666...|       2.0|
|Business|The value of the ...|  2.0|[the, value, of, ...|[value, dollar, f...|(1000,[8,10,13,14...|(1000,[8,10,13,14...|[0.0,
3.0,11.0,1.0]|[0.0,0.2,0.733333...|       2.0|
|Business|LONDON — It is th...|  2.0|[london, —, it, i...|[london, —, close...|(1000,[0,1,2,3,5,...|(1000,[0,1,2,3,5,...|[4.0,
1.0,10.0,0.0]|[0.26666666666666...|       2.0|
|Business|A start-up is tak...|  2.0|[a, start-up, is,...|[start-up, taking...|(1000,[13,15,23,2...|(1000,[13,15,23,2...|[0.0,
2.0,11.0,2.0]|[0.0,0.1333333333...|       2.0|
|Business|Target agreed on ...|  2.0|[target, agreed, ...|[target, agreed, ...|(1000,[0,7,13,17,...|(1000,[0,7,13,17,...| [2.0,
4.0,9.0,0.0]|[0.13333333333333...|       2.0|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
-----------+--------------------+----------+
only showing top 20 rows

Accuracy for Random Forest on new test data = 0.9318181818181818
```

Accuracy % for this new test data = 93.1818 using Random Forest

2) Naïve Bayes

Input snippet is as follows:

```
In [25]:
# Running Naive Bayes on this new Test Data

predictions = cvModelNB.transform(testDf)
predictions.show()

# compute accuracy on the test set
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Accuracy for Naive Bayes on new test data = " + str(accuracy))
```

Output obtained is as follows:

```
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
|category|                text|label|               words|            filtered|         rawFeatures|            features|
rawPrediction|         probability|prediction|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
|Business|One of the few re...|  2.0|[one, of, the, fe...|[one, remaining, ...|(1000,[0,2,3,9,11...|(1000,[0,2,3,9,11...|[-301
1.9941663186...|[5.26919189748328...|       2.0|
|Business|Deutsche Bank sai...|  2.0|[deutsche, bank, ...|[deutsche, bank, ...|(1000,[2,8,12,20,...|(1000,[2,8,12,20,...|[-210
6.1071354629...|[3.46778455863961...|       2.0|
|Business|Spotify is a hit....|  2.0|[spotify, is, a, ...|[spotify, hit.on,...|(1000,[0,5,10,16,...|(1000,[0,5,10,16,...|[-252
1.2715558083...|[5.24177077111416...|       2.0|
|Business| Good Tuesday. He...|  2.0|[, good, tuesday....|[, good, tuesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...|[-750
6.8131448156...|[1.61820182577732...|       2.0|
|Business|After days of som...|  2.0|[after, days, of,...|[days, sometimes,...|(1000,[1,12,13,14...|(1000,[1,12,13,14...|[-219
1.7680300123...|[4.17544377601364...|       2.0|
|Business|Good Thursday. He...|  2.0|[good, thursday.,...|[good, thursday.,...|(1000,[0,1,2,3,5,...|(1000,[0,1,2,3,5,...|[-873
0.0431577621...|[1.80408137245065...|       2.0|
|Business|ABINGTON, Pa. — S...|  2.0|[abington,, pa., ...|[abington,, pa., ...|(1000,[0,1,6,8,13...|(1000,[0,1,6,8,13...|[-486
3.7543345193...|[1.94455957850438...|       2.0|
|Business|Wall Street was p...|  2.0|[wall, street, wa...|[wall, street, pr...|(1000,[3,25,34,36...|(1000,[3,25,34,36...|[-898.
67269335400...|[0.00715648644266...|       2.0|
|Business|SAN JUAN, P.R. — ...|  2.0|[san, juan,, p.r....|[san, juan,, p.r....|(1000,[1,8,10,12,...|(1000,[1,8,10,12,...|[-583
8.3449909497...|[5.13954664577655...|       1.0|
|Business|The Labor Departm...|  2.0|[the, labor, depa...|[labor, departmen...|(1000,[0,1,2,11,1...|(1000,[0,1,2,11,1...|[-319
0.5216782319...|[5.28421050630945...|       2.0|
|Business|The sell-off in s...|  2.0|[the, sell-off, i...|[sell-off, stocks...|(1000,[8,25,30,34...|(1000,[8,25,30,34...|[-716.
68137543547...|[2.15449959628509...|       2.0|
|Business|SHENZHEN, China —...|  2.0|[shenzhen,, china...|[shenzhen,, china...|(1000,[3,4,6,10,1...|(1000,[3,4,6,10,1...|[-370
9.6720756224...|[3.68395548400811...|       2.0|
|Business|The stock market ...|  2.0|[the, stock, mark...|[stock, market, f...|(1000,[1,3,13,16,...|(1000,[1,3,13,16,...|[-221
3.9265913997...|[3.59359840019174...|       2.0|
|Business|I didn’t mean to ...|  2.0|[i, didn’t, mean,...|[didn’t, mean, it...|(1000,[0,2,4,6,7,...|(1000,[0,2,4,6,7,...|[-386
0.7644305603...|[5.58602358666721...|       1.0|
|Business|Brady Hill used h...|  2.0|[brady, hill, use...|[brady, hill, use...|(1000,[0,1,2,3,8,...|(1000,[0,1,2,3,8,...|[-478
5.4573792874...|[2.64678005613299...|       1.0|
|Business|FRANKFURT — Mario...|  2.0|[frankfurt, —, ma...|[frankfurt, —, ma...|(1000,[1,3,5,11,1...|(1000,[1,3,5,11,1...|[-331
5.8439144248...|[4.76783878500862...|       2.0|
|Business|The value of the ...|  2.0|[the, value, of, ...|[value, dollar, f...|(1000,[8,10,13,14...|(1000,[8,10,13,14...|[-240
6.5359729508...|[3.84554733685896...|       2.0|
|Business|LONDON — It is th...|  2.0|[london, —, it, i...|[london, —, close...|(1000,[0,1,2,3,5,...|(1000,[0,1,2,3,5,...|[-429
5.1571068717...|[5.88172441745027...|       2.0|
|Business|A start-up is tak...|  2.0|[a, start-up, is,...|[start-up, taking...|(1000,[13,15,23,2...|(1000,[13,15,23,2...|[-158
9.2867134807...|[7.47630977571436...|       2.0|
|Business|Target agreed on ...|  2.0|[target, agreed, ...|[target, agreed, ...|(1000,[0,7,13,17,...|(1000,[0,7,13,17,...|[-268
4.6288165668...|[1.19133345808314...|       2.0|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
--------------+--------------------+----------+
only showing top 20 rows

Accuracy for Naive Bayes on new test data = 0.9545454545454546
```

Accuracy % for this new test data = 95.4545 using Naïve Bayes

3) Logistic Regression (Multiclass)

Input snippet is as follows:

```python
In [26]:
# Running LR on this new Test Data

predictions = cvModelLR.transform(testDf)
predictions.show()

# compute accuracy on the test set
evaluator = MulticlassClassificationEvaluator(labelCol="label", predictionCol="prediction",
                                              metricName="accuracy")
accuracy = evaluator.evaluate(predictions)
print("Accuracy for Logistic Regression on new test data = " + str(accuracy))
```

Output obtained is as follows:

```
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
-------------+--------------------+----------+
|category|                text|label|               words|            filtered|         rawFeatures|            features|
rawPrediction|         probability|prediction|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
-------------+--------------------+----------+
|Business|One of the few re...|  2.0|[one, of, the, fe...|[one, remaining, ...|(1000,[0,2,3,9,11...|(1000,[0,2,3,9,11...|[-0.31
72092509106...|[0.04569309638105...|       2.0|
|Business|Deutsche Bank sai...|  2.0|[deutsche, bank, ...|[deutsche, bank, ...|(1000,[2,8,12,20,...|(1000,[2,8,12,20,...|[-0.40
55738077416...|[0.06206982346279...|       2.0|
|Business|Spotify is a hit....|  2.0|[spotify, is, a, ...|[spotify, hit.on,...|(1000,[0,5,10,16,...|(1000,[0,5,10,16,...|[-0.87
95030717582...|[0.01198762837706...|       2.0|
|Business| Good Tuesday. He...|  2.0|[, good, tuesday....|[, good, tuesday....|(1000,[0,1,2,3,4,...|(1000,[0,1,2,3,4,...|[-0.74
06603218313...|[0.00401937982547...|       2.0|
|Business|After days of som...|  2.0|[after, days, of,...|[days, sometimes,...|(1000,[1,12,13,14...|(1000,[1,12,13,14...|[-0.13
37464066476...|[0.01928446558270...|       2.0|
|Business|Good Thursday. He...|  2.0|[good, thursday.,...|[good, thursday.,...|(1000,[0,1,2,3,5,...|(1000,[0,1,2,3,5,...|[-0.01
98318623180...|[0.00223920729712...|       2.0|
|Business|ABINGTON, Pa. — S...|  2.0|[abington,, pa., ...|[abington,, pa., ...|(1000,[0,1,6,8,13...|(1000,[0,1,6,8,13...|[0.058
52725101781...|[0.01848723712352...|       2.0|
|Business|Wall Street was p...|  2.0|[wall, street, wa...|[wall, street, pr...|(1000,[3,25,34,36...|(1000,[3,25,34,36...|[0.264
61591218756...|[0.25414026762762...|       2.0|
|Business|SAN JUAN, P.R. — ...|  2.0|[san, juan,, p.r....|[san, juan,, p.r....|(1000,[1,8,10,12,...|(1000,[1,8,10,12,...|[0.651
30541633843...|[0.14458847592453...|       1.0|
|Business|The Labor Departm...|  2.0|[the, labor, depa...|[labor, departmen...|(1000,[0,1,2,11,1...|(1000,[0,1,2,11,1...|[-0.47
10796609528...|[0.00716170624164...|       2.0|
|Business|The sell-off in s...|  2.0|[the, sell-off, i...|[sell-off, stocks...|(1000,[8,25,30,34...|(1000,[8,25,30,34...|[-0.26
17319971911...|[0.11287955191852...|       2.0|
|Business|SHENZHEN, China —...|  2.0|[shenzhen,, china...|[shenzhen,, china...|(1000,[3,4,6,10,1...|(1000,[3,4,6,10,1...|[-0.09
76212170915...|[0.02796134045170...|       2.0|
|Business|The stock market ...|  2.0|[the, stock, mark...|[stock, market, f...|(1000,[1,3,13,16,...|(1000,[1,3,13,16,...|[-0.94
66755644878...|[0.01670182023814...|       2.0|
|Business|I didn’t mean to ...|  2.0|[i, didn’t, mean,...|[didn’t, mean, it...|(1000,[0,2,4,6,7,...|(1000,[0,2,4,6,7,...|[-0.36
44393046674...|[0.06680863860525...|       1.0|
|Business|Brady Hill used h...|  2.0|[brady, hill, use...|[brady, hill, use...|(1000,[0,1,2,3,8,...|(1000,[0,1,2,3,8,...|[-0.11
55582931381...|[0.08386980169683...|       2.0|
|Business|FRANKFURT — Mario...|  2.0|[frankfurt, —, ma...|[frankfurt, —, ma...|(1000,[1,3,5,11,1...|(1000,[1,3,5,11,1...|[0.108
62511965586...|[0.05212680493397...|       2.0|
|Business|The value of the ...|  2.0|[the, value, of, ...|[value, dollar, f...|(1000,[8,10,13,14...|(1000,[8,10,13,14...|[-0.38
31358682494...|[0.04575677078512...|       2.0|
|Business|LONDON — It is th...|  2.0|[london, —, it, i...|[london, —, close...|(1000,[0,1,2,3,5,...|(1000,[0,1,2,3,5,...|[0.016
84002502551...|[0.01986400736163...|       2.0|
|Business|A start-up is tak...|  2.0|[a, start-up, is,...|[start-up, taking...|(1000,[13,15,23,2...|(1000,[13,15,23,2...|[-0.27
27928214479...|[0.08170405119475...|       2.0|
|Business|Target agreed on ...|  2.0|[target, agreed, ...|[target, agreed, ...|(1000,[0,7,13,17,...|(1000,[0,7,13,17,...|[-0.08
85195097157...|[0.06253463473132...|       2.0|
+--------+--------------------+-----+--------------------+--------------------+--------------------+--------------------+------
-------------+--------------------+----------+
only showing top 20 rows

Accuracy for Logistic Regression on new test data = 0.9636363636363636
```

Accuracy % for this new test data = 96.3636 using Logistic Regression