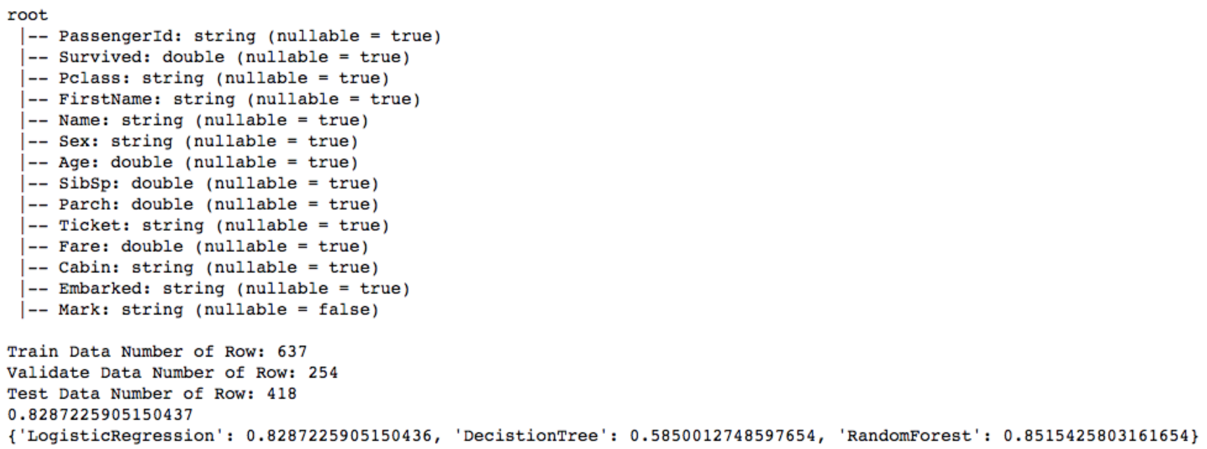
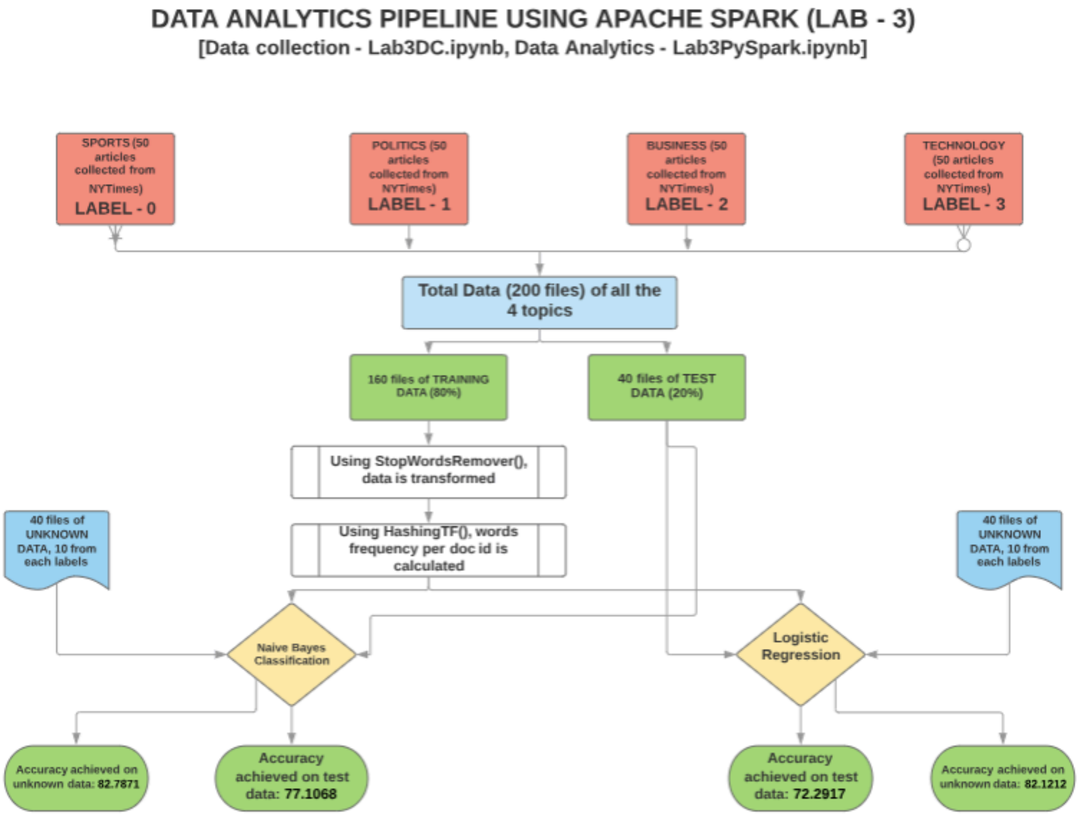
# DATA ANALYTICS PIPELINE USING APACHE SPARK (LAB – 3)
# CSE 587 – DATA INTENSIVE COMPUTING

**Part 1 –** This has been understood in python notebook Lab3TitanicData.ipynb. Please find below the screenshot taken when the notebook was executed.



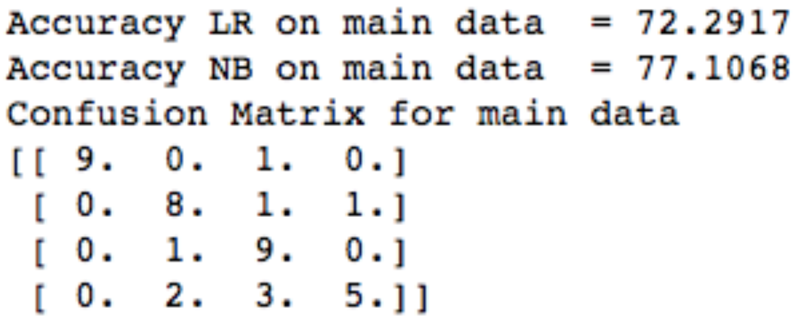
```
root
 |-- PassengerId: string (nullable = true)
 |-- Survived: double (nullable = true)
 |-- Pclass: string (nullable = true)
 |-- FirstName: string (nullable = true)
 |-- Name: string (nullable = true)
 |-- Sex: string (nullable = true)
 |-- Age: double (nullable = true)
 |-- SibSp: double (nullable = 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)

Train Data Number of Row: 637
Validate Data Number of Row: 254
Test Data Number of Row: 418
0.8287225905150437
{'LogisticRegression': 0.8287225905150436, 'DecistionTree': 0.5850012748597654, 'RandomForest': 0.8515425803161654}
```

**Part 2 –** Data has been taken using NYTimes API key. Code has been provided in python notebook Lab3DC.ipynb. Cleaning of the data has been done in PySpark using StopWordsRemover feature available. Please see the below block diagram to understand the whole process. Code of PySpark is available in notebook Lab3PySpark.ipynb.


DATA ANALYTICS PIPELINE USING APACHE SPARK (LAB - 3)
[Data collection - Lab3DC.ipynb, Data Analytics - Lab3PySpark.ipynb]
SPORTS (50 articles collected from NYTimes) LABEL - 0
POLITICS (50 articles collected from NYTimes) LABEL - 1
BUSINESS (50 articles collected from NYTimes) LABEL - 2
TECHNOLOGY (50 articles collected from NYTimes) LABEL - 3
Total Data (200 files) of all the 4 topics
160 files of TRAINING DATA (80%)
40 files of TEST DATA (20%)
Using StopWordsRemover(), data is transformed
Using HashingTF(), words frequency per doc id is calculated
40 files of UNKNOWN DATA, 10 from each labels
40 files of UNKNOWN DATA, 10 from each labels
Naive Bayes Classification
Logistic Regression
Accuracy achieved on unknown data: 82.7871
Accuracy achieved on test data: 77.1068
Accuracy achieved on test data: 72.2917
Accuracy achieved on unknown data: 82.1212

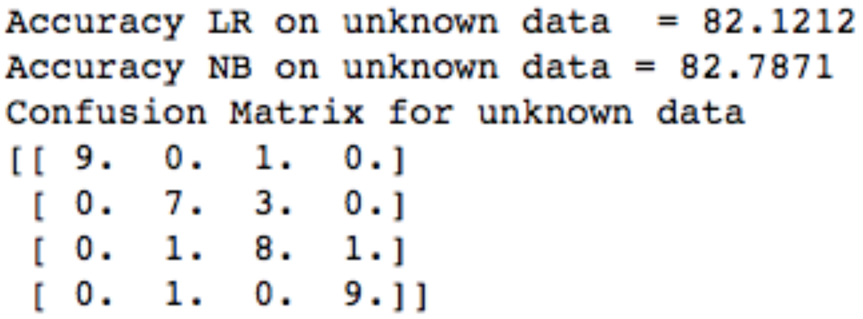
**Part 3 –** Feature engineering has been done in training data of 160 files as shown in the block diagram. This is done using TF-IDF feature provided by Spark. Word frequency is calculated according to it's importance in each document.

**Part 4 –** Training has been done using Logistic Regression and Naïve Bayes Classifications since these are the two models which provide the best accuracy on the dataset. Accuracies provided by these models are **72.2917** and **77.1068** respectively. Accuracy has been calculated by providing input as a test dataset to both the models. Confusion matrix has also been printed below. The code is available in notebook Lab3PySpark.ipynb.



```
Accuracy LR on main data  = 72.2917
Accuracy NB on main data  = 77.1068
Confusion Matrix for main data
[[ 9.  0.  1.  0.]
 [ 0.  8.  1.  1.]
 [ 0.  1.  9.  0.]
 [ 0.  2.  3.  5.]]
```

**Part 5 –** 10 files of each label has been collected as an unknown data set used to check if the model is robust or not. Total 40 files are put inside folder "UnknownArticlesFolder". The input is provided to both the models and results have been noted. Accuracies for Logistic Regression and Naïve Bayes Classification are **82.1212** and **82.7871**. Clearly, both the models are robust since accuracies are in match. Confusion matrix has also been printed below. The code is available in notebook Lab3PySpark.ipynb.



```
Accuracy LR on unknown data  = 82.1212
Accuracy NB on unknown data = 82.7871
Confusion Matrix for unknown data
[[ 9.  0.  1.  0.]
 [ 0.  7.  3.  0.]
 [ 0.  1.  8.  1.]
 [ 0.  1.  0.  9.]]
```