**Cloud Computing with Map Reduce Application on Nation Branding twitter data for sentiment analysis**

**User Manual**

**Required Softwares:**

1. R Studio to verify script
2. Virtual Box/VMvare to run Hadoop jobs
3. Eclipse IDE

**Steps to run the project:**

1. Initially load ExtractTweetData.R file into R studio and run the script, this fetches data from twitter. But this is not necessary as I have supplied data for execution, if you want to execute on different data you can run the script.
2. Once we have data in xls format, convert this xls into csv by using java program (xls\_to\_csv.java) submitted. But you don’t need to use this program as the csv files are converted already, but to handle new data f xls you need to use this program.
3. Once you have all the csv data for the Excels, then run virtual box because Hadoop jobs can only be run on virtual box but we have run on cloud using EC2, S3 and EMR of AWS.
4. Once you have installed Virtual Box, then open it and install Eclipse IDE to run Map Reduce jobs.
5. In IDE create a project and package with naivebayes, copy all the naivebayes java files to this package.
6. In these java files run NBController.java program by giving all the inputs. Inputs include:

* Bag of positive (total\_pos\_words.txt)
* Bag of negative (total\_neg\_words.txt)
* Emojis - Positive & Negative (total\_epos\_words.txt/ total\_eneg\_words.txt)
* Slang words (total\_slpos\_words.txt/ total\_slneg\_words.txt)
* Sarcastic words (total\_sarpos\_words.txt/ total\_sarneg\_words.txt)

1. In the above inputs, positive and negative bag of words is compulsory, if you fail to give rest of the inputs it just ignores.
2. In the actual Hadoop naives bayes sentiment analysis, there are 7 java files. Each java file functionality is explained below:

* NBController.java – Main class which takes inputs and triggers other classes
* NBTrainingMapper.java – Mapper for generating Training model
* NBTrainingReducer.java - Reducer for generating Training model
* NBTestPrepareMapper – Mapper that takes model and test data to generate intermediate data for classification job.
* NBTestPrepareReducer.java - Mapper that takes model and test data to generate intermediate data for classification job.
* NBClassifyMapper.java – Mapper that takes intermediate results from combiner/prepare job to generate results.
* NBClassifyReducer.java - Reducer that takes intermediate results from combiner/prepare job to generate results.

1. Final output of the program is results.txt which has metrics of accuracy percentage, positive sentiment percentage and negative sentiment percentage of csv document (All CSV files are present in data folder, Mexican country data is used).
2. These metrics are documented in IEEE paper submitted.