Lab2 Report

List of Activities:

1. As required in Part 1 and in order to get our hands dirty with Python libraries like Pandas, Scikitlearns, numPy etc. we completed all the code expositions in chapters 4 and 5 of the textbook and ran the snippets of Python code in separate Jupyter notebooks for chapters 4 and 5.

2. We wrote a Python script to collect full tweets related to our topic i.e. 'Shootings' and stored the tweets in a text file. Search API and tweepy library was used. We collected tweets of a single day and then for a single week.

3. Similarly we scraped New York Times articles using the NYTArticles API. We wrote a Python script to collect the articles, scrape entire articles from the URLs and then copy the articles in a text file. We used nytimesarticles library and NYT article API key for that. We collected articles related to "shooting" of a single day and then for a single week.

4. We downloaded the required files and installed Hadoop on our local Linux machine.

5. MapReduce was coded in Python. MRJob library was used. The text files obtained in previous steps were given as input to the program. Finally the output of the reducer was stored in a txt file.

6. We then wrote a script to extract the top 10 words in terms of count and saved it in a txt and csv file. This text file was used as one of the inputs in the Cooccurence program and csv file is used for visualisation.

7. MapReduce with cooccurence for top 10 words was coded in Python. MRJob library was used. The text files obtained in previous steps were given as input to the program. Finally the output of the reducer was stored in a txt file. This step is done for single day of data.

8. Mapper can't understand that <word1,word2> and <word2,word1> is the same pair. So displays different results for these two pairs. To nullify the effect of order on the pair count, we wrote a script to add their results and remove other pair of same words from the file.

9. Finally, the outputs in steps 6 and 8 were converted to csv files which were then visualized in d3.js using a wordcloud. We used top 50 words and top 10 pairs in the word cloud.

Our script's function is given below:

Part2→
DataCollection:
1) getArticle.py: To collect nytimes articles. Begindate, enddate and topic is given in the script.
It stores collected articles in the txt file.

2) getTweets.py: This script is used to collect the tweets for given hashtag and given dates. It stores collected tweets in the txt file. We have also retweeted tweets so that tweets wont repeat multiple times.

WordCount:

3) mrword: This script uses hadoop mapreduce to give word count for provided txt file and stores result in another txt file.

4) top10extract.py: This script sorts collected tweets according to count, and stores top 10 words in a txt file according to the data. Used dont have to enter name of the file where words are to be stored. Used just need to give the choice according to the data. This script also creates csv file which contain top 50 words with their counts.

Co-occurance:
5) cooc1.py: This script takes tweets in txt file as a input, divides data by tweet and emits result if there is co-occurance of top 10 words. It stores result in txt file with count.

6)coocNYT.py: This script takes article data and does the same functionality as above by dividing data by to paragraph.

7) removeRepeatPairs.py: This script does functionality as explained in step 7. It also created wordpair cloud and stores result in txt and csv respectively.

*
8) script.py: Instead of running these 7 scripts and providing different user details, we have created a script which runs these methods sequencially and gives result accordingly.