A DevOps Pipeline for Real Time Social/Web Mining

End Semester Report Submitted
in Fulfillment of the Requirements
for the Course of
Minor Project - II
In
Third year – Fifth Semester of
Bachelor of Technology
specialization
In
DevOps, CCVT and GG.

Under

Dr. Monit Kapoor

By

500060720	**R171217041**	**Nishkarsh Raj**
500060911	**R110217071**	**Karan Nautiyal**
500059999	**R110217083**	**Megha Rawat**
500060227	**R142217022**	**Rishabh Negi**

DEPARTMENT OF CYBERNETICS AND VIRTUALIZATION
SCHOOL OF COMPUTER SCIENCE
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES,
BIDHOLI, DEHRADUN, UTTRAKHAND, INDIA
January, 2020

Abstract

In today's world, social media plays a very crucial role in our society. The Internet revolution has caused the problem of data explosion where data generated on average by every human is more than 1.37 MB per second.This large amount of data is mostly unstructured hence difficult to analyze, but once processed can provide important insight of sentiments of individual consumers.

Users have an opportunity to express their personal opinions about specific topics. The example of such websites include blogs, forums, product reviews sites, and social networks. In this case, twitter data is used. Sites like twitter contain prevalently short comments.This project contributes to the sentiment analysis for customers' review classification which is helpful to analyze the information in the form of the number of tweets where opinions are highly unstructured and are either positive or negative, or somewhere in between these two.

The proposed framework combines three methodologies – descriptive analytics (DA), content analytics (CA) integrating text mining and sentiment analysis, and network analytics (NA) relying on network visualization and metrics – for extracting intelligence from tweets about any hashtag.

Keywords: Twitter, Data analytics, Network analytics, Content analytics, Big data, Social media analytics,Application Programming Interface (API), Hadoop

Introduction

The social web is a great source of information and the best platform for communication. The Platform gives users to connect and communicate freely. Social web mining is the process of representing, analyzing and extracting patterns from raw social media data. Social web mining includes web platforms, network analysis to provide a convenient and consistent platform for the users. Real-time web mining means that the system is subjected to real-time, i.e., the response is delivered within a particular timing constraint or system should meet the confirmed deadline. For example flight control systems, real-time monitors, etc.
Twitter is a social media application that allows the users to send and read short messages('280 characters') called 'tweets'. Users on twitter generate about half-billion tweets on Twitter every day. The project aim is to collect social information and tweets using a set of specified keywords and geolocation-based criteria by using an open-source library called Tweepy and build the Twitter crawler. For accessing the Twitter API easily a python library Tweepy is used.

Twitter API handles a huge amount of data. This data can be used for economic, industrial, social or government approaches by analyzing these tweets as per the demand. Since Twitter contains an enormous volume of data, storing and processing this data is a complex problem.

Twitter APIs can be accessed only through the authenticated requests. Twitter uses Open Authentication and each request must be signed with valid Twitter user credentials. Twitter APIs are also accessible to a limited number of requests within a time window known as the rate limit. These limits are applicable to both the individual user level as well as for the application level.

Open Authentication is an open standard framework for authentication adopted by Twitter to provide access to the private and secure information of the users. It is strongly supported by Twitter, Google and other companies. OAuth supports three-way handshaking to provide safer authentication. Twitter APIs can only be accessed by registered applications (e.g., the crawlers will develop in the project). Python is a programming language for fast Text data processing. Tweepy is an open-source library hosted on GitHub. The library provides an easy way for the python script to communicate with the Twitter API.

Twitter APIs can be accessed only through the authenticated requests. Twitter uses Open Authentication and each request must be signed with valid Twitter user

credentials. Twitter APIs are also accessible to a limited number of requests within a time window known as the rate limit. These limits are applicable to both the individual user level as well as for the application level.

For an enormous data processing Hadoop tool is used for analyzing data. Data with huge volume, variety and velocity is analyzed easily through Hadoop. The framework analyses Big data and it has its own family which supports the processing of different things which are tied up in one umbrella called the Hadoop Ecosystem. Apache Flume is also used to capture real-time tweets. As an analysis, the main objective is to generate a method for finding recent trends in tweets and performing a sentiment analysis on real-time tweets. The analysis is done through the Hadoop ecosystem tool Apache Pig. To ensure the data is secured for developers and users certain authentications are adopted.

By analyzing the enormous Twitter data, can help in understanding the Customers, helps in Influencer Marketing, brand monitoring and Sentiment Analysis.
Sentiment Analysis means the design of the product in such a way that it reaches the right audience. This well helps in gathering the overall experience and opinions of the targeted audiences.

Motivation

Data is invaluable. Every second, there are more than 300k+ tweets made on Twitter and more than 20k+ hashtags are used in the same period. It is necessary that some insights are drawn from this vast amount of data to understand what the audience perspectives are.
In today's world, people blindly post about trending hashtags to support or protest against a cause without knowing the core background of the topics.
The motivation of the project is to extract the top five trending hashtags in real-time and tweet about them via automated pipeline using Google News API to spread awareness about what the hashtags are really about.
Further, this project aims to analyze the information in the form of the number of tweets where opinions are highly unstructured and are either positive or negative, or somewhere in between these two.

Problem Statement

There are three main problems with analysing social media content:

1) People might be unaware of the hashtags and unwilling to do their research before tweeting their opinion for the whole world to see.
2) It is unstructured, the inherent design of social media makes it difficult to analyse.Unstructured data does not have a data model, thus it cannot be used with relational databases. We need to pre-process the data.
3) The sheer amount of data. Every second, there are more than 300k+ tweets made on Twitter and more than 20k+ hashtags are used in the same period.

Objective

This project aims to create a fully automated application that extracts information in real time from the social media and web platforms using API's and capture the general sentiment of the people using the mass media and also share awareness about topics that people talk about regardless of their knowledge on the topic of interest.
The sub objectives needed to be completed to accomplish the main objective are:

- To create a web crawler using Twitter API and OAuth protocol.
- Populate dataset with unstructured data in real time using HDFS
- Create Hadoop clusters and use YARN to manage Big Data pipeline.
- Perform Data Analytics using Pig, Hive, Impala etc.
- Visualize data using Kibana, Matplotlib etc.

Literature Review

Semantic analysis is a very trending topic, a lot of work has been done in this sector in the past few years. Bo Pang and Lee pioneered this sector.

The research paper "Large-Scale Machine Learning at Twitter"[1] using machine learning tools integrated into Hadoop Clusters analysed by pig.They have used a knowledge-poor,data-driven approach. The data set consisted of 1 million English tweets with emoticons having equal positive and negative tweets. They identified stochastic gradient descent techniques and ensemble methods for being highly amenable even for a large amount of data. Their results showed an accuracy in the range 77% to 82% depending on the size of the dataset.

Algorithm:

let's assume a two-class problem with a training set D = {(x1, y1),(x2, y2)...(xn, yn)}, we define a class of linear discriminative functions of the form:

$F(x) : R\ N \rightarrow \{-1, +1\}$

$$F(x) = \begin{cases} +1 \text{ if } w \cdot x \geq t \\ -1 \text{ if } w \cdot x < t \end{cases}$$

where t represents a decision threshold and w is the weight vector.

The modeling process usually optimizes the weight vector based on the training data D, and the decision threshold is subsequently tuned according to various operational constraints (e.g., taking precision, recall, and coverage into account). A range of methods for learning w have been proposed, where generally different methods optimize a different form of a loss or objective function defined over the training data. One particularly well-established technique is known as logistic regression, where the linear function $w \cdot x$ is interpreted as the logarithmic odds of x belonging to class +1 over −1, The objective of regularized logistic regression (using Gaussian smoothing) is to identify the parameter vector w that maximizes the conditional posterior of the data.

In stochastic gradient update where the gradient is computed based on a single training instance, then update the weight vector upon seeing the ith training example. Note that all elements of the weight vector w are decayed at each iteration, but in situations where the feature vectors of training instances are very sparse (as is true for text) we can simply delay the updates for features until they are actually seen

Machine Learning algorithms are the most suitable algorithms for sentiment analysis as they have the ability to "learn". However when working on a huge data set like the one in this project some algorithms might not scale optimally. Liu, Bingwei, Eric Balsch, Yu Chen, Dan Shen, and Genshe Chen have evaluated the scalability of Naive Bayes Classifier in the research paper "Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier"[2]. They performed sentiment mining on large datasets using a Naive Bayes Theorem classifier with the Hadoop framework. The results showed 80.85%accuracy.

Algorithms:

Algorithm 1 : Training Job
Input: The training reviews.
Output: Model.

```
1. define function MAPPER(key, value)
2. s = parse_sentiment(value)
3. for all w belongs to d do
4. emit(w, 1 : s)
5. end for loop
6. end Mapper()
7. define function REDUCER(key, values)
8. declare posSum = 0; negSum = 0;
9. for all value in values do
10. [s, count] = parse(value)
11. if (s == positive) then
12. posSum = posSum + count;
13. else 14: negSum = negSum +  count;
15. end if 16: end for loop[
17. call emit(key, posSum, negSum)
18. end function
```

Algorithm 2 : Combining Job
Input: The test reviews and the model.
Output: Intermediate results from input of job classification.

```
1. function MAPPER(key, value)
2. if value in model then
3. [word, possum, negsum] = parse(value)
4. emit(word, possum, negsum)
5. else
```

```
6. [docid, sentiment] = parse(value)
7. for all word in value do
8. emit(word,1,docid,sentiment)
9. end for loop
10. end if
11. end function
12. function REDUCER(key, values)
13. for all value in values do
14. if value in model then
15. outstr.append(value)
16. else
17. [count, docid] = parse(value)
18. docs[docid].add(count)
19. end if
20. end for loop[
21. for all docid ∈ docs do
22. outstr.append(count,docid)
23. end for loop
24. emit(word, outstr)
25. end function
```

Algorithm 3 : Classify Job
Input: The intermediate results.
Output: Classification results and accuracy.

```
1. function MAPPER(key, value)
2. for all item in value do
3. [docid, count, possum, negsum] = parse(value)
4. emit(docid, count, possum, negsum)
5. end for loop
6. end function
7. function REDUCER(key, values)
8. [docid, trueSentiment]=parse(key)
9. for all value in values do
10. calculate positive probability
11. calculate negative probability
12. end for
13. predict = (pos prob > neg prob) ? pos : neg
14. if predict = trueSentiment then
15. correct = true
```

```
16. correct count + +
17. else
18. correct = false
19. end if
20. emit(docid, predict, correct)
21. end function
```

Proposed Method

A. Extraction of Real-Time Data

This project uses the streaming API provided by twitter to fulfill the real time data need. The API allows us to view 1% of the recent tweets based on the particular keyword. The hashtag on which we want to perform sentiment analysis is input into the crawler which calls twitter API mines it's database and provides the tweets related to that hashtag.

Twitter data is unstructured, people often use abbreviations. A tweet consists of a maximum of 280 characters. Users might also use emoticons which directly indicate their emotions. User location can be used to differentiate trends for various geographical regions.

B. Stemming and Lemmatization

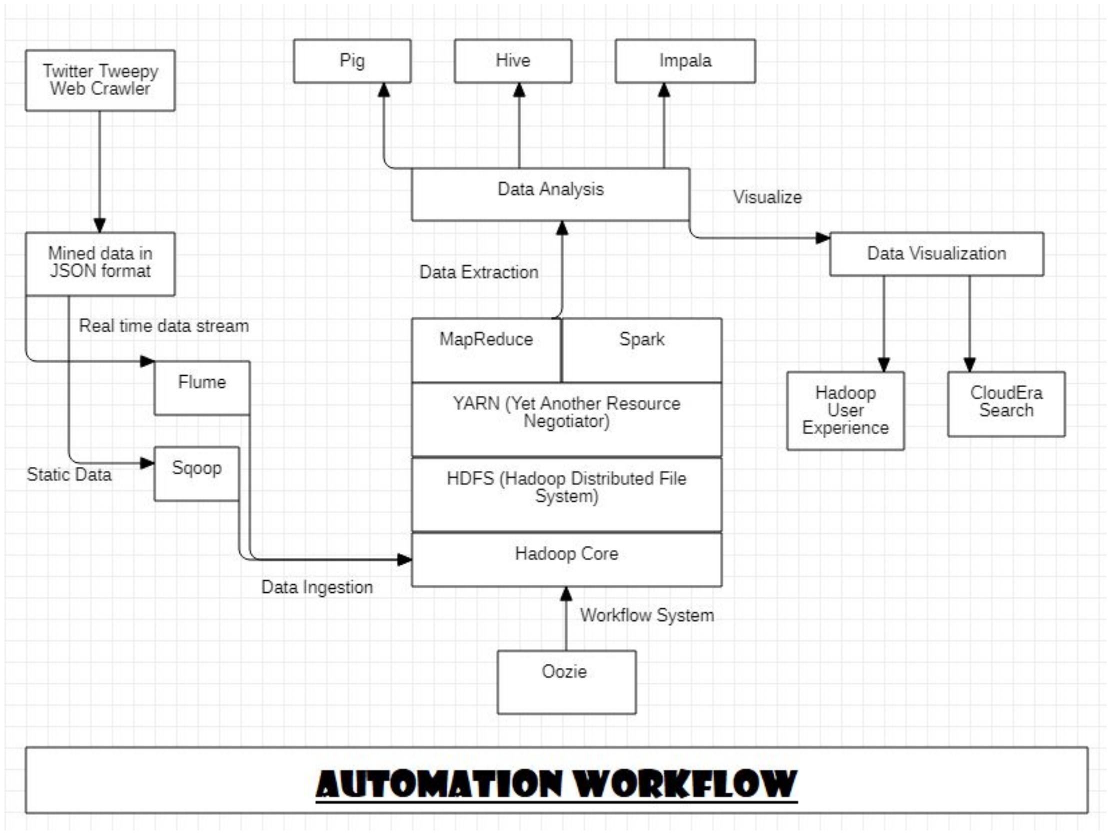
The words in the tweet are converted to their root form to avoid the unwanted extra storage of the derived word's sentiment. The root form dictionary is used to do that which is made local as it is heavily used in the program. This lowers the access time and increases the overall efficiency of the system.

C. Sentiment Directory

This project uses a sentiment Directory that is created using data obtained from sentiwordnet, and uses all possible usage of a particular word, i.e. "good" can be used in many different contexts each way having a different sentiment value each time it is used. So sentiment from all possible usages of good is obtained and stored in a directory which should also be local to the project to save time that would otherwise be wasted in searching the word in the secondary memory storage.

D. Map-reduce

The faster real time processing needed for this project can be obtained by using cluster architecture set up by hadoop. The program uses chained map-reduce structure which processes every tweet and assigns the sentiment to each remaining words of tweet and then sums them up to find the final sentiment. We should specially consider phrases where sentiment of phrase matters rather than sentiment of each word. It can be done by creating a dynamic directory of phrases whose sentiment values will be derived from a standard algorithm.



<u>Methodology</u>

<u>MapReduce for Parallel Computing in Distributed Environment</u>

<u>**Scaling:**</u> Scaling is the increase/decrease of computing resources allocated to the process in execution according to the need.

There are two forms of Scaling:

1. <u>**Vertical Scaling:**</u> Vertical scaling modifies the system resources of the same machine as per load. This is highly expensive and difficult because it requires change in the hardware infrastructure which may lead to some downtime.

2. **Horizontal Scaling:** Horizontal scaling increases the number of processors/machines in the same hierarchy to solve the problem in modules using parallel computing approach.

Distributed Environment: It consists of multiple machines that work in parallel and they work in the same hierarchy, i.e., one machine does not control another machine as in Master-Slave architecture.

Horizontal hierarchy makes it more secure as these environments don't have one-point failure as in the Centralized environment.

MapReduce: MapReduce is a Big Data framework that is used to process huge amounts of data in parallel using Distributed Computing.

One task is divided into multiple sub tasks that are independent. They are processed in parallel and output is created by collating output of subtasks from parallel processors.

MapReduce Process

MapReduce consists of two major tasks which itself consists of two minor tasks each.

1. Map Process

1.1) Split function

1.2) Map function

2. Reduce Process

2.1) Combine/Shuffling function

2.2) Reduce function

Map process

Split function: It divides the task into subtasks and assigns each subtask to different parallel processors using hash functions.

Map function: Map function process occurs in individual processors.

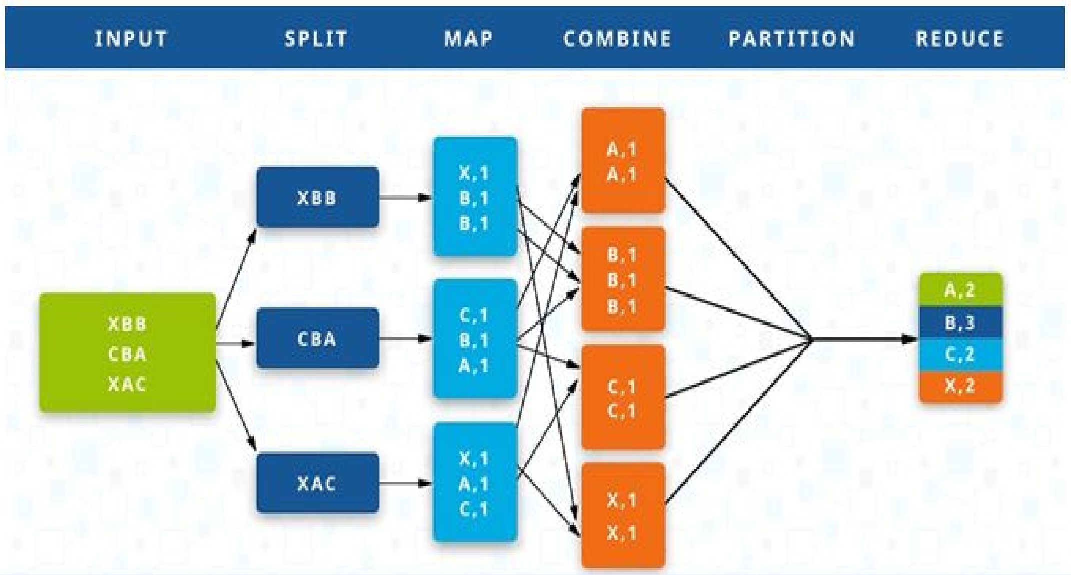
Takes input from split functions, reads the input and creates an intermediate form of computational records in terms of (key,value) pairs.

Reduce process

Combine function: Collects output of individual computation from different processors and pre-processes the information and organizes it.

Reduce function: It collects the combined output, collates the results and processes it to produce final output.

MapReduce Example



MapReduce Process for counting frequency of letters in a file

In this example, MapReduce process is applied on a file to count the occurrences of individual characters in the file.

The split function takes the file as an input and divides it into smaller parts with each part consisting of one line of the original file.

The map function assigns each sub part's character with initial value to make (key,value pair) for further computation.

The combine function pre-processes each sub part and organizes each letter with mapping generated key-value pairs in the same container.

The reduce function collates result from each sub process and gives the final output.

System Requirements:

1. **Software:**

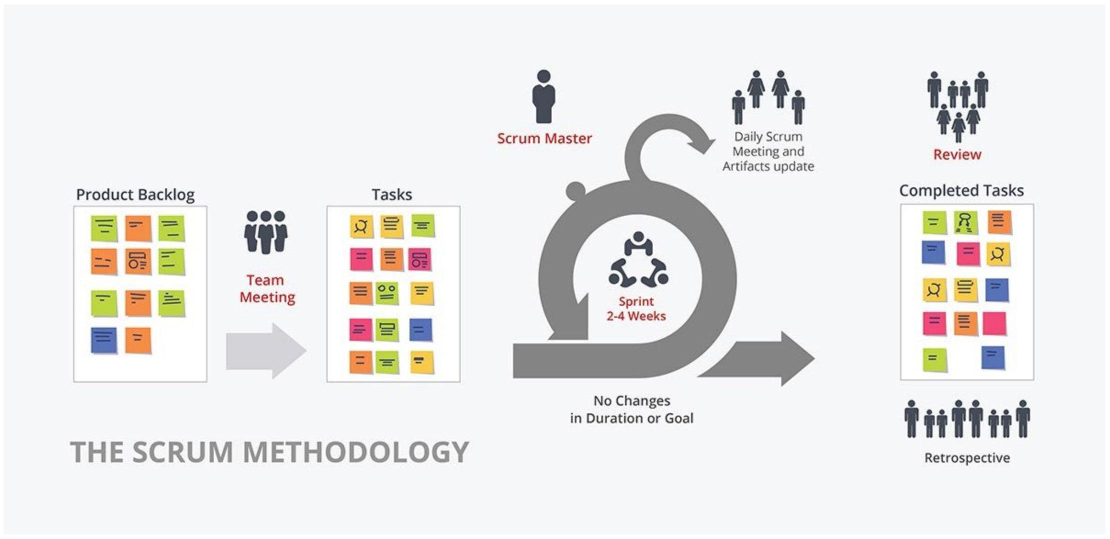
* Python 3.7
* JDK 11
* Hadoop
* Supports Windows XP, Vista, 7, 8, 8.1 and 10

2. **Hardware:**

- 6 GB RAM or above
- I5 5th Generation or above processors

SDLC Model: SCRUM Model

The entire project is divided into individual modules, based on their functionalities. Which will be developed in isolation by the entire team, based on the priority of implementation. Using the Version Control tool Git.



[1] Scrum Model

Efficiency Parameters

1. **Velocity:** Velocity is the speed by which our crawler mines data and stores it into our HDFS cluster.
2. **Throughput:** Throughput is how much data the project can analyze in unit time.
3. **Accuracy:** Accuracy will help us understand how many tweets were successfully based on the sentiments of each tweet.
4. **Storage:** We need to do all this in a realistic amount of space as the data we will be handling is huge.

Plan of Work

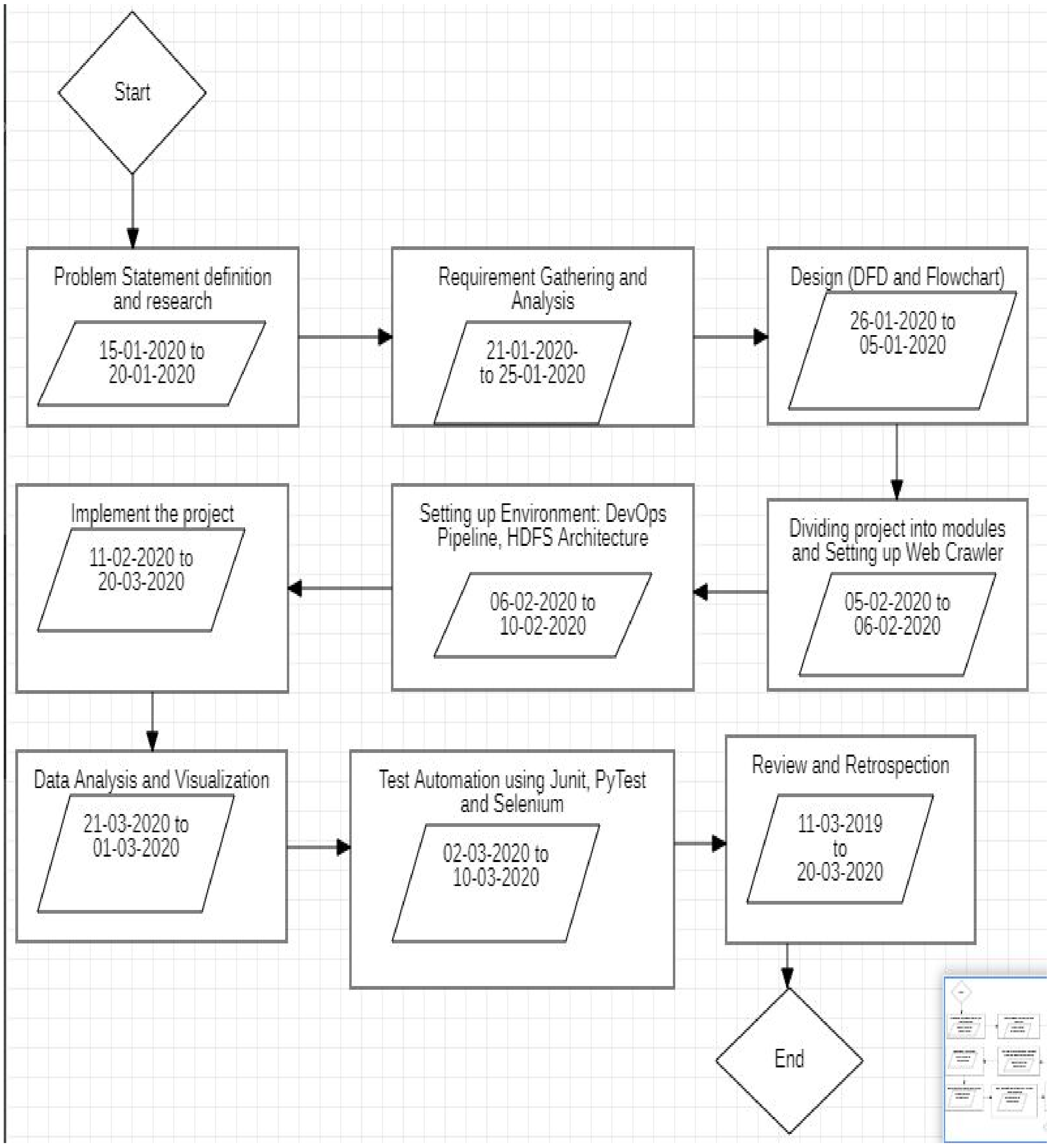

Start
Problem Statement definition and research
15-01-2020 to 20-01-2020
Requirement Gathering and Analysis
21-01-2020- to 25-01-2020
Design (DFD and Flowchart)
26-01-2020 to 05-01-2020
Dividing project into modules and Setting up Web Crawler
05-02-2020 to 06-02-2020
Setting up Environment: DevOps Pipeline, HDFS Architecture
06-02-2020 to 10-02-2020
Implement the project
11-02-2020 to 20-03-2020
Data Analysis and Visualization
21-03-2020 to 01-03-2020
Test Automation using Junit, PyTest and Selenium
02-03-2020 to 10-03-2020
Review and Retrospection
11-03-2019 to 20-03-2020
End

Figure 2: Plan of Work

Project progress

We are using the SCRUM model so we have divided the project into different modules. All these modules are listed in the product log

Product Log :

This is the most important document. It list all the requirements of the project and can be considered as a to-do list. All these modules provide some functionality to the business process. Our product log is as follows

1] Setting up Apache Maven for Java project - User Interface and MapReduce functions

2] Setting up GitHub repository workflow

3] Setting up GitHub Actions for automation

4] Creating a web crawler in Python using Tweepy library to fetch data based on some parameter.

5] Create a User Interface

6] Create a HDFS cluster for MapReduce functionality and program Hadoop MapReduce in Java

7] Setup Hadoop Core and create Job Tracker and Task Trackers for the project

8] Implement MapReduce in HDFS using Java to count the frequency of significant words in Data dictionary, in Twitter string

9] Configure Apache Maven with MapReduce codes and install Apache Hadoop Jar dependency

10] Configure MapReduce code in GitHub Actions for automation

11] Automate the Big Data pipeline till MapReduce using GitHub Actions

12] Use Data Ingestion tools like Flume to send data from crawler to HDFS at real time

13] WAP in Java to implement MapReduce from JSON file extracted from crawler to find the frequency of significant words - Textual Analysis

14] Data Classification - create a multi-class data dictionary for sentiment analysis - currently for words (in future, we might extend it for phrases and sentences for improved accuracy)

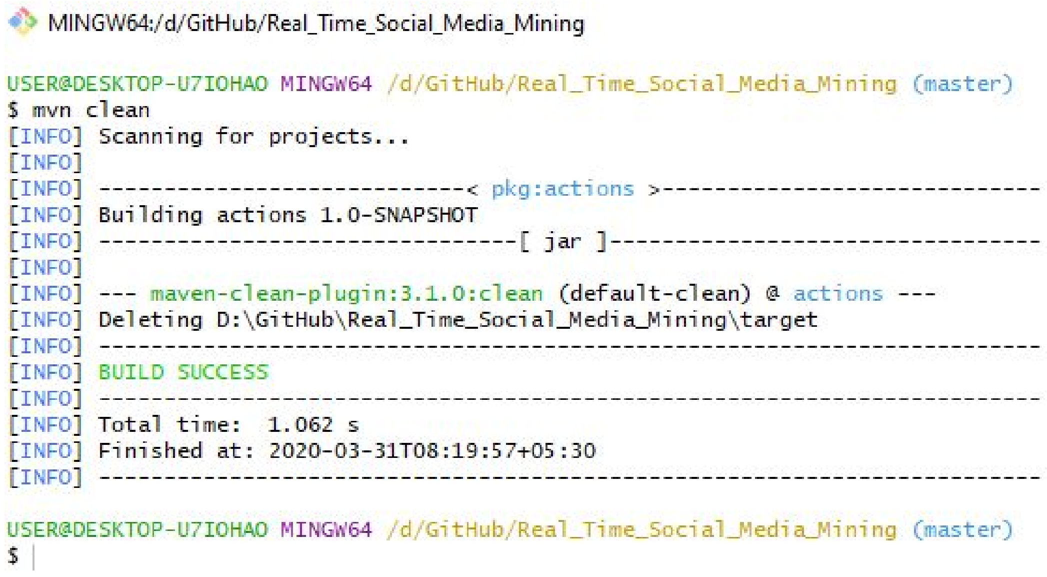
15] Data Prediction - Using the KNN algorithm in Python to find the relation between tweets and their sentiments.

16] Data Visualization - Using the Python matplotlib library to implement visualization.

Increment of Sprint 1 (Sprint Review):

This a list of all the modules that have been completed in the sprints. The modules that we have finished are

1]Setting up Apache Maven for Java project - User Interface and MapReduce functions

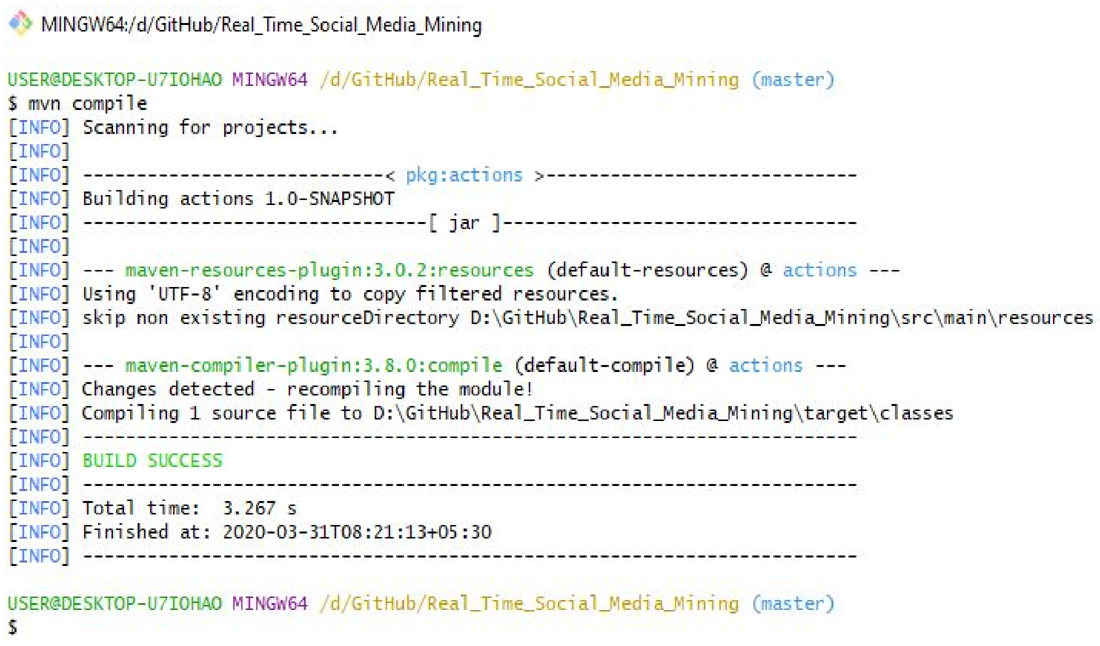


```
MINGW64:/d/GitHub/Real_Time_Social_Media_Mining

USER@DESKTOP-U7IOHAO MINGW64 /d/GitHub/Real_Time_Social_Media_Mining (master)
$ mvn clean
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------------< pkg:actions >----------------------------
[INFO] Building actions 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-clean-plugin:3.1.0:clean (default-clean) @ actions ---
[INFO] Deleting D:\GitHub\Real_Time_Social_Media_Mining\target
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  1.062 s
[INFO] Finished at: 2020-03-31T08:19:57+05:30
[INFO] ------------------------------------------------------------------------

USER@DESKTOP-U7IOHAO MINGW64 /d/GitHub/Real_Time_Social_Media_Mining (master)
$ |
```

Figure: Maven Clean goal

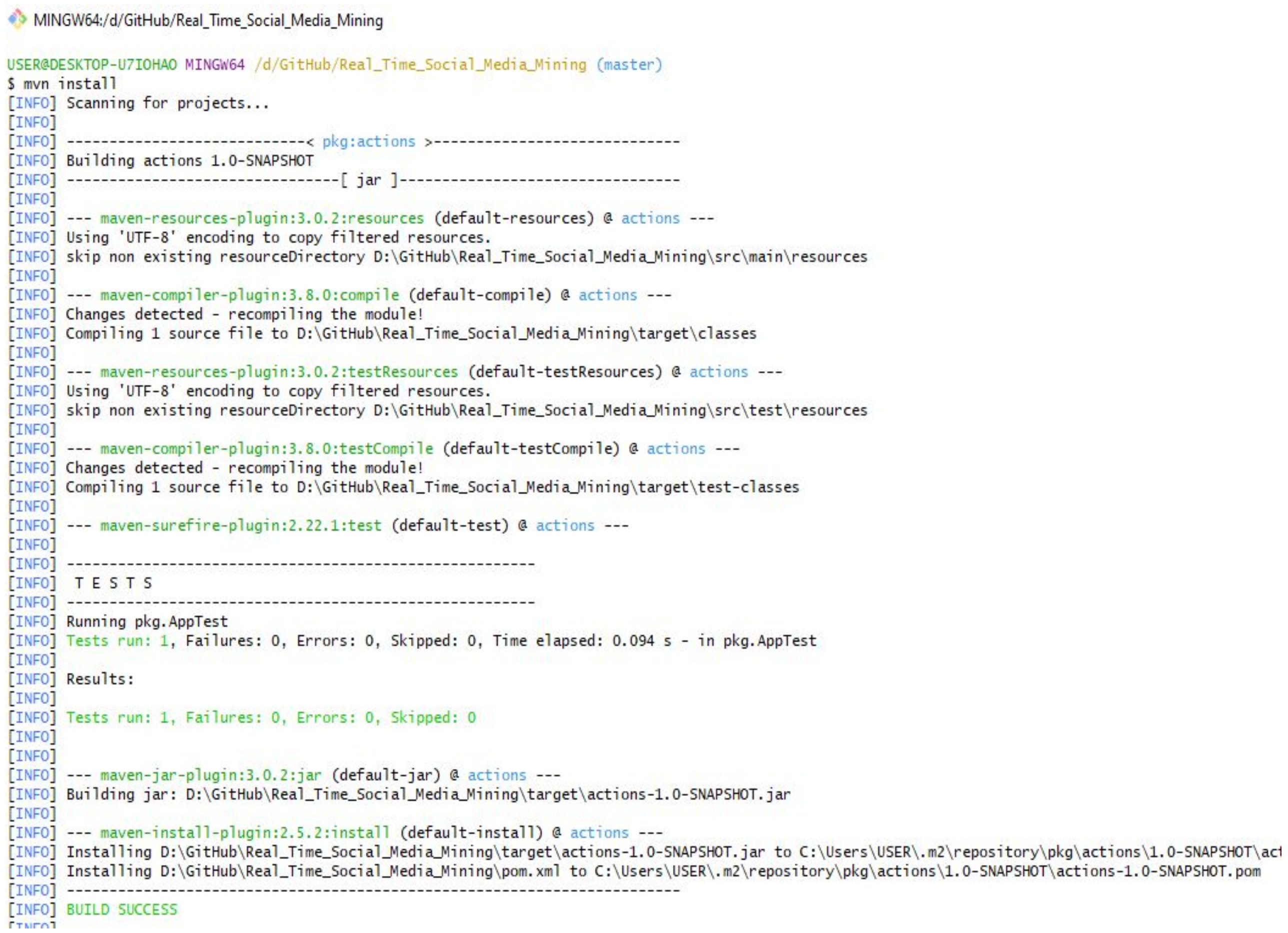


MINGW64:/d/GitHub/Real_Time_Social_Media_Mining

```
USER@DESKTOP-U7IOHAO MINGW64 /d/GitHub/Real_Time_Social_Media_Mining (master)
$ mvn compile
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------------< pkg:actions >-----------------------------
[INFO] Building actions 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ actions ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\GitHub\Real_Time_Social_Media_Mining\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ actions ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\GitHub\Real_Time_Social_Media_Mining\target\classes
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  3.267 s
[INFO] Finished at: 2020-03-31T08:21:13+05:30
[INFO] ------------------------------------------------------------------------

USER@DESKTOP-U7IOHAO MINGW64 /d/GitHub/Real_Time_Social_Media_Mining (master)
$
```

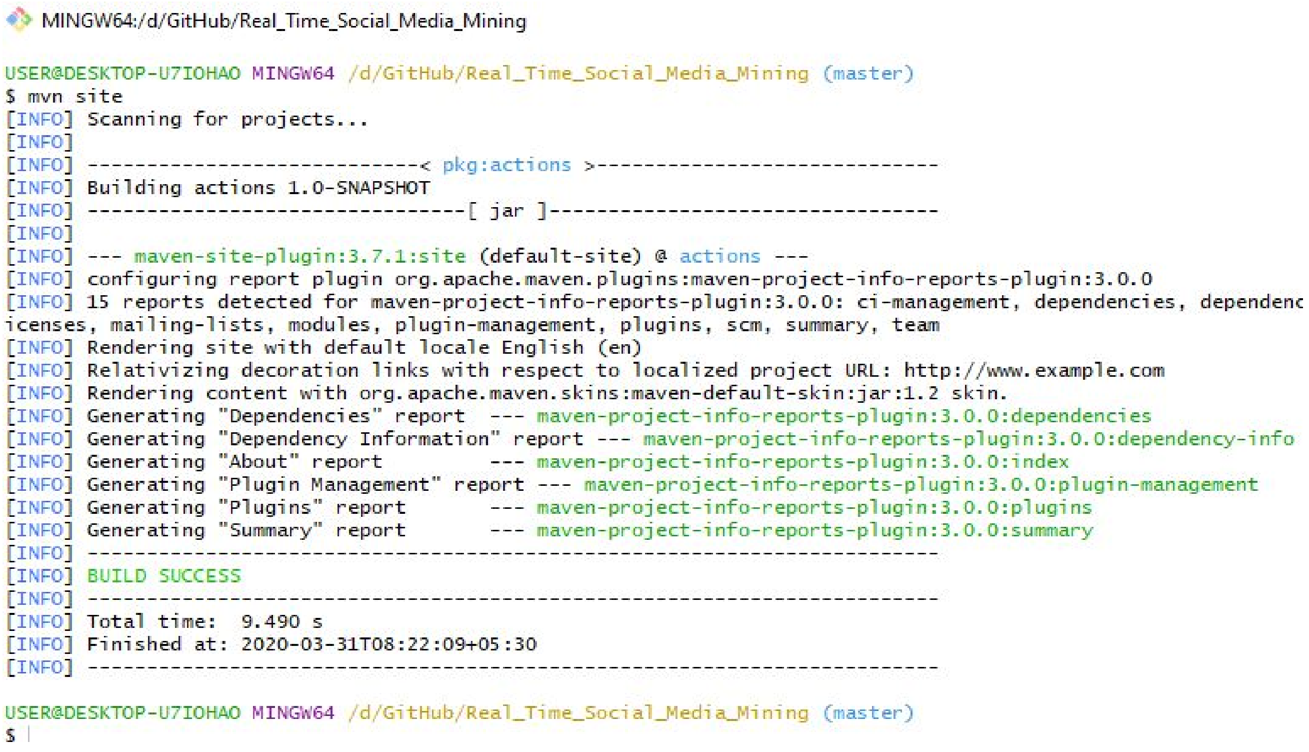
Figure: Maven compile goal



MINGW64:/d/GitHub/Real_Time_Social_Media_Mining

```
USER@DESKTOP-U7IOHAO MINGW64 /d/GitHub/Real_Time_Social_Media_Mining (master)
$ mvn install
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------------< pkg:actions >-----------------------------
[INFO] Building actions 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:resources (default-resources) @ actions ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\GitHub\Real_Time_Social_Media_Mining\src\main\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:compile (default-compile) @ actions ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\GitHub\Real_Time_Social_Media_Mining\target\classes
[INFO]
[INFO] --- maven-resources-plugin:3.0.2:testResources (default-testResources) @ actions ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory D:\GitHub\Real_Time_Social_Media_Mining\src\test\resources
[INFO]
[INFO] --- maven-compiler-plugin:3.8.0:testCompile (default-testCompile) @ actions ---
[INFO] Changes detected - recompiling the module!
[INFO] Compiling 1 source file to D:\GitHub\Real_Time_Social_Media_Mining\target\test-classes
[INFO]
[INFO] --- maven-surefire-plugin:2.22.1:test (default-test) @ actions ---
[INFO]
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running pkg.AppTest
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.094 s - in pkg.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO]
[INFO] --- maven-jar-plugin:3.0.2:jar (default-jar) @ actions ---
[INFO] Building jar: D:\GitHub\Real_Time_Social_Media_Mining\target\actions-1.0-SNAPSHOT.jar
[INFO]
[INFO] --- maven-install-plugin:2.5.2:install (default-install) @ actions ---
[INFO] Installing D:\GitHub\Real_Time_Social_Media_Mining\target\actions-1.0-SNAPSHOT.jar to C:\Users\USER\.m2\repository\pkg\actions\1.0-SNAPSHOT\act
[INFO] Installing D:\GitHub\Real_Time_Social_Media_Mining\pom.xml to C:\Users\USER\.m2\repository\pkg\actions\1.0-SNAPSHOT\actions-1.0-SNAPSHOT.pom
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO]
```

Figure: Maven install goal



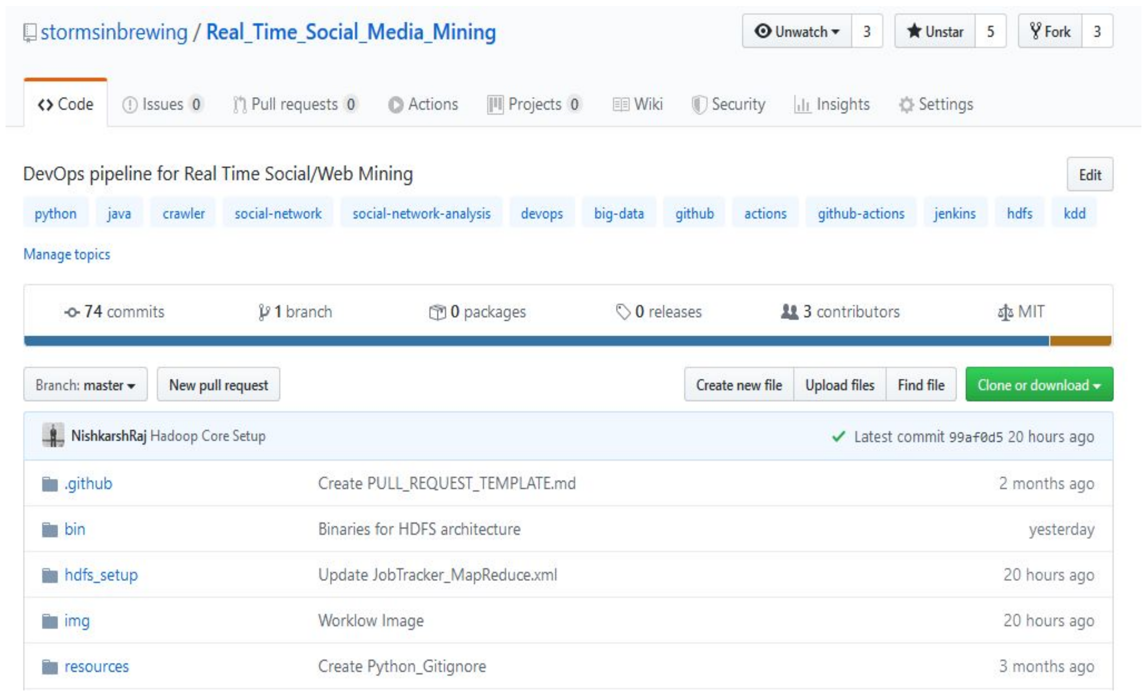
```
MINGW64:/d/GitHub/Real_Time_Social_Media_Mining

USER@DESKTOP-U7IOHAO MINGW64 /d/GitHub/Real_Time_Social_Media_Mining (master)
$ mvn site
[INFO] Scanning for projects...
[INFO]
[INFO] ----------------------------< pkg:actions >----------------------------
[INFO] Building actions 1.0-SNAPSHOT
[INFO] --------------------------------[ jar ]---------------------------------
[INFO]
[INFO] --- maven-site-plugin:3.7.1:site (default-site) @ actions ---
[INFO] configuring report plugin org.apache.maven.plugins:maven-project-info-reports-plugin:3.0.0
[INFO] 15 reports detected for maven-project-info-reports-plugin:3.0.0: ci-management, dependencies, dependenc
icenses, mailing-lists, modules, plugin-management, plugins, scm, summary, team
[INFO] Rendering site with default locale English (en)
[INFO] Relativizing decoration links with respect to localized project URL: http://www.example.com
[INFO] Rendering content with org.apache.maven.skins:maven-default-skin:jar:1.2 skin.
[INFO] Generating "Dependencies" report  --- maven-project-info-reports-plugin:3.0.0:dependencies
[INFO] Generating "Dependency Information" report --- maven-project-info-reports-plugin:3.0.0:dependency-info
[INFO] Generating "About" report         --- maven-project-info-reports-plugin:3.0.0:index
[INFO] Generating "Plugin Management" report --- maven-project-info-reports-plugin:3.0.0:plugin-management
[INFO] Generating "Plugins" report       --- maven-project-info-reports-plugin:3.0.0:plugins
[INFO] Generating "Summary" report       --- maven-project-info-reports-plugin:3.0.0:summary
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  9.490 s
[INFO] Finished at: 2020-03-31T08:22:09+05:30
[INFO] ------------------------------------------------------------------------

USER@DESKTOP-U7IOHAO MINGW64 /d/GitHub/Real_Time_Social_Media_Mining (master)
$
```

Figure: Maven site goal

2] Setting up GitHub repository workflow: This project has a repository on GitHub, to manage workflow and better version control.



stormsinbrewing / Real_Time_Social_Media_Mining

DevOps pipeline for Real Time Social/Web Mining

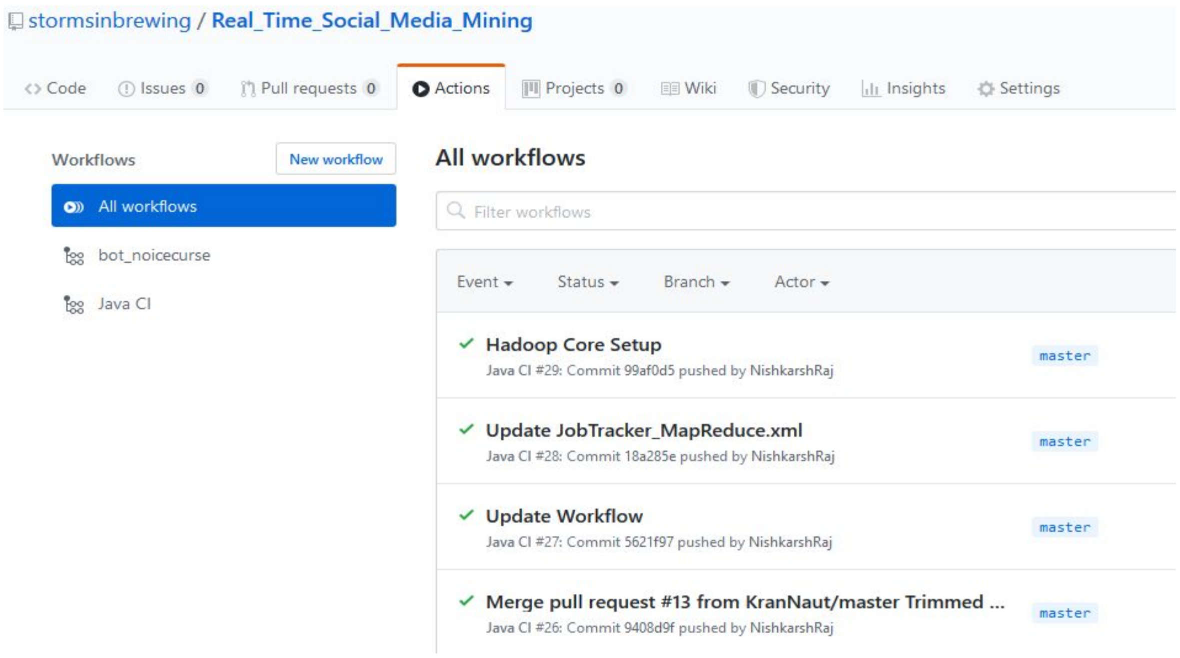
python java crawler social-network social-network-analysis devops big-data github actions github-actions jenkins hdfs kdd

74 commits | 1 branch | 0 packages | 0 releases | 3 contributors | MIT

NishkarshRaj Hadoop Core Setup — Latest commit 99af0d5 20 hours ago

.github	Create PULL_REQUEST_TEMPLATE.md	2 months ago
bin	Binaries for HDFS architecture	yesterday
hdfs_setup	Update JobTracker_MapReduce.xml	20 hours ago
img	Worklow Image	20 hours ago
resources	Create Python_Gitignore	3 months ago

3] Setting up GitHub Actions for automation



Hadoop Core Setup — master
Java CI #29: Commit 99af0d5 pushed by NishkarshRaj

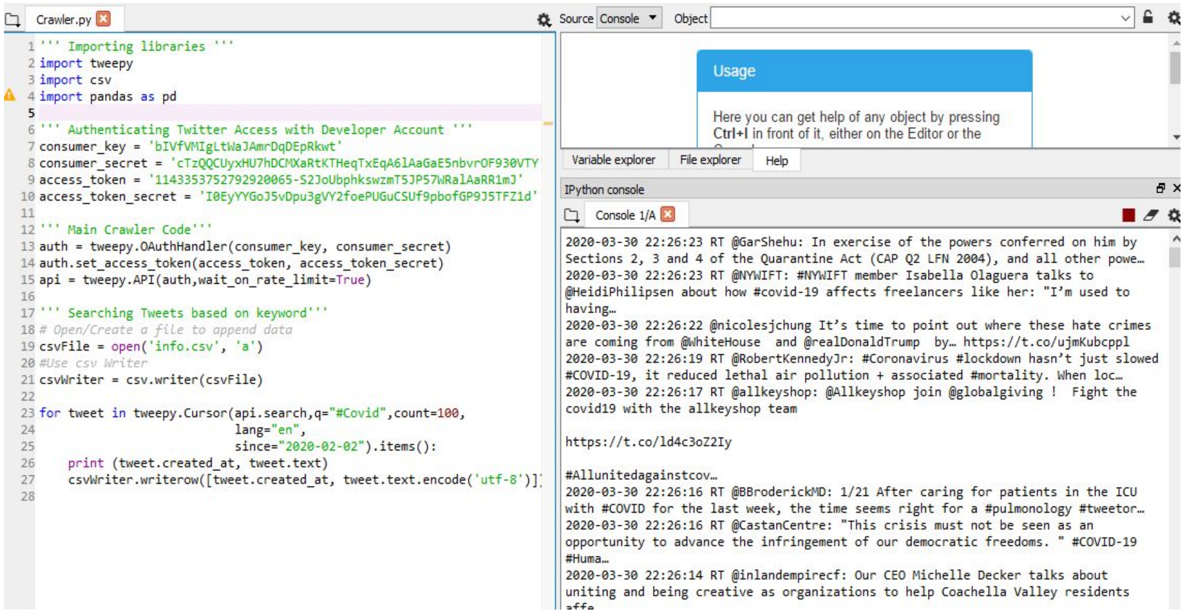
Update JobTracker_MapReduce.xml — master
Java CI #28: Commit 18a285e pushed by NishkarshRaj

Update Workflow — master
Java CI #27: Commit 5621f97 pushed by NishkarshRaj

Merge pull request #13 from KranNaut/master Trimmed ... — master
Java CI #26: Commit 9408d9f pushed by NishkarshRaj

Figure: GitHub Actions setup

4] Creating a web crawler in Python using Tweepy library to fetch data based on Hashtag and Location: This project uses a python script to crawl twitter using the **twitter API** and **tweepy** library and returns a number of tweets based on the tweeters **location** and the **hashtags** used in the tweet.


Crawler.py
Source
Console
Object
Usage
Here you can get help of any object by pressing Ctrl+I in front of it, either on the Editor or the
Variable explorer
File explorer
Help
IPython console
Console 1/A

```python
''' Importing libraries '''
import tweepy
import csv
import pandas as pd

''' Authenticating Twitter Access with Developer Account '''
consumer_key = 'bIVfVMIgLtWaJAmrDqDEpRkwt'
consumer_secret = 'cTzQQCUyxHU7hDCMXaRtKTHeqTxEqA6lAaGaE5nbvrOF930VTY
access_token = '1143353752792920065-S2JoUbphkswzmT5JP57WRalAaRR1mJ'
access_token_secret = 'I0EyYYGoJ5vDpu3gVY2foePUGuCSUf9pbofGP9J5TFZ1d'

''' Main Crawler Code'''
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth,wait_on_rate_limit=True)

''' Searching Tweets based on keyword'''
# Open/Create a file to append data
csvFile = open('info.csv', 'a')
#Use csv Writer
csvWriter = csv.writer(csvFile)

for tweet in tweepy.Cursor(api.search,q="#Covid",count=100,
                           lang="en",
                           since="2020-02-02").items():
    print (tweet.created_at, tweet.text)
    csvWriter.writerow([tweet.created_at, tweet.text.encode('utf-8')])

```

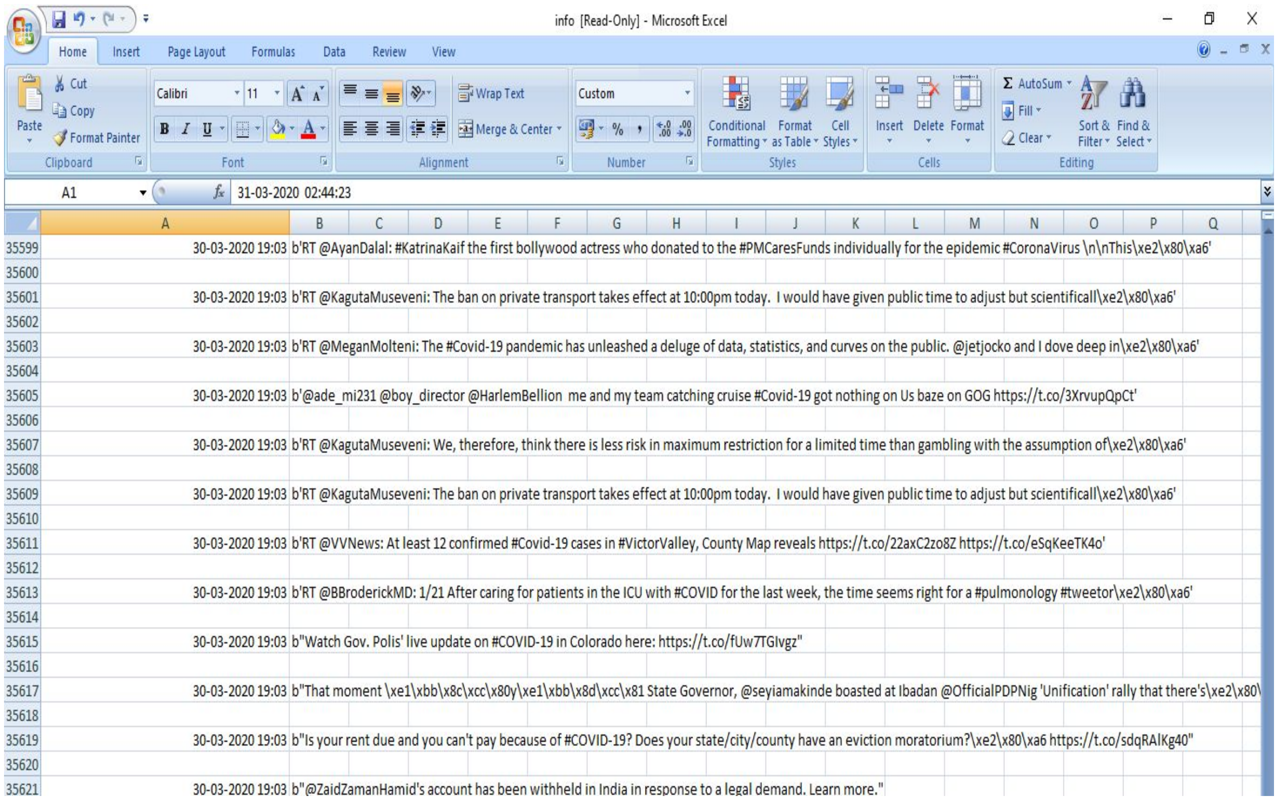
```
2020-03-30 22:26:23 RT @GarShehu: In exercise of the powers conferred on him by
Sections 2, 3 and 4 of the Quarantine Act (CAP Q2 LFN 2004), and all other powe…
2020-03-30 22:26:23 RT @NYWIFT: #NYWIFT member Isabella Olaguera talks to
@HeidiPhilipsen about how #covid-19 affects freelancers like her: "I'm used to
having…
2020-03-30 22:26:22 @nicolesjchung It's time to point out where these hate crimes
are coming from @WhiteHouse  and @realDonaldTrump  by… https://t.co/ujmKubcppl
2020-03-30 22:26:19 RT @RobertKennedyJr: #Coronavirus #lockdown hasn't just slowed
#COVID-19, it reduced lethal air pollution + associated #mortality. When loc…
2020-03-30 22:26:17 RT @allkeyshop: @Allkeyshop join @globalgiving !  Fight the
covid19 with the allkeyshop team

https://t.co/ld4c3oZ2Iy

#Allunitedagainstcov…
2020-03-30 22:26:16 RT @BBroderickMD: 1/21 After caring for patients in the ICU
with #COVID for the last week, the time seems right for a #pulmonology #tweetor…
2020-03-30 22:26:16 RT @CastanCentre: "This crisis must not be seen as an
opportunity to advance the infringement of our democratic freedoms. " #COVID-19
#Huma…
2020-03-30 22:26:14 RT @inlandempirecf: Our CEO Michelle Decker talks about
uniting and being creative as organizations to help Coachella Valley residents
affe
```

Figure: Execution of script

The API response is trimmed to remove special symbols like @,# and links. The tweets and their dates are stored in a csv file.



A1 31-03-2020 02:44:23

	A	B
35599	30-03-2020 19:03	b'RT @AyanDalal: #KatrinaKaif the first bollywood actress who donated to the #PMCaresFunds individually for the epidemic #CoronaVirus \n\nThis\xe2\x80\xa6'
35600		
35601	30-03-2020 19:03	b'RT @KagutaMuseveni: The ban on private transport takes effect at 10:00pm today. I would have given public time to adjust but scientificall\xe2\x80\xa6'
35602		
35603	30-03-2020 19:03	b'RT @MeganMolteni: The #Covid-19 pandemic has unleashed a deluge of data, statistics, and curves on the public. @jetjocko and I dove deep in\xe2\x80\xa6'
35604		
35605	30-03-2020 19:03	b'@ade_mi231 @boy_director @HarlemBellion me and my team catching cruise #Covid-19 got nothing on Us baze on GOG https://t.co/3XrvupQpCt'
35606		
35607	30-03-2020 19:03	b'RT @KagutaMuseveni: We, therefore, think there is less risk in maximum restriction for a limited time than gambling with the assumption of\xe2\x80\xa6'
35608		
35609	30-03-2020 19:03	b'RT @KagutaMuseveni: The ban on private transport takes effect at 10:00pm today. I would have given public time to adjust but scientificall\xe2\x80\xa6'
35610		
35611	30-03-2020 19:03	b'RT @VVNews: At least 12 confirmed #Covid-19 cases in #VictorValley, County Map reveals https://t.co/22axC2zo8Z https://t.co/eSqKeeTK4o'
35612		
35613	30-03-2020 19:03	b'RT @BBroderickMD: 1/21 After caring for patients in the ICU with #COVID for the last week, the time seems right for a #pulmonology #tweetor\xe2\x80\xa6'
35614		
35615	30-03-2020 19:03	b"Watch Gov. Polis' live update on #COVID-19 in Colorado here: https://t.co/fUw7TGIvgz"
35616		
35617	30-03-2020 19:03	b"That moment \xe1\xbb\x8c\xcc\x80y\xe1\xbb\x8d\xcc\x81 State Governor, @seyiamakinde boasted at Ibadan @OfficialPDPNig 'Unification' rally that there's\xe2\x80\
35618		
35619	30-03-2020 19:03	b"Is your rent due and you can't pay because of #COVID-19? Does your state/city/county have an eviction moratorium?\xe2\x80\xa6 https://t.co/sdqRAIKg40"
35620		
35621	30-03-2020 19:03	b"@ZaidZamanHamid's account has been withheld in India in response to a legal demand. Learn more."

Figure: 35000 tweets extracted by the crawler about #COVID19

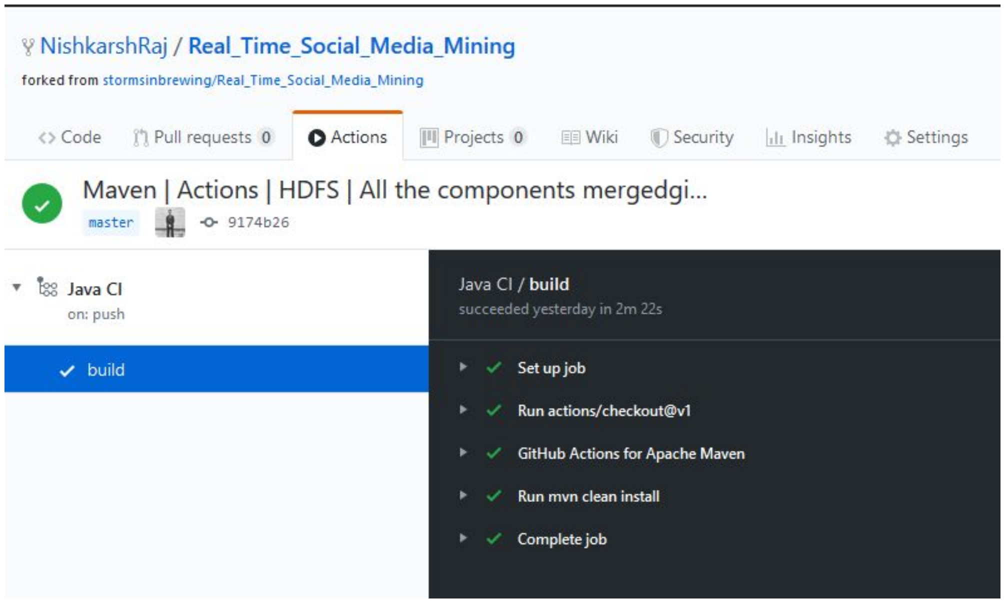
5] Create a HDFS cluster for MapReduce functionality and program Hadoop MapReduce in Java: We store the file in a HDFS cluster and ise MapRed to analyse it. So, we created the clusters and uploaded the configuration files on github for easy replication.

6] Setup Hadoop Core and create Job Tracker and Task Trackers for the project

7] Implemented MapReduce in HDFS using Java to count the frequency of significant words in Data dictionary, in Twitter string

8] Configure Apache Maven with MapReduce codes and install Apache Hadoop Jar dependency

9] Configure MapReduce code in GitHub Actions for automation


NishkarshRaj / Real_Time_Social_Media_Mining
forked from stormsinbrewing/Real_Time_Social_Media_Mining

Maven | Actions | HDFS | All the components mergedgi...
master 9174b26

Java CI
on: push

✓ build

Java CI / build
succeeded yesterday in 2m 22s

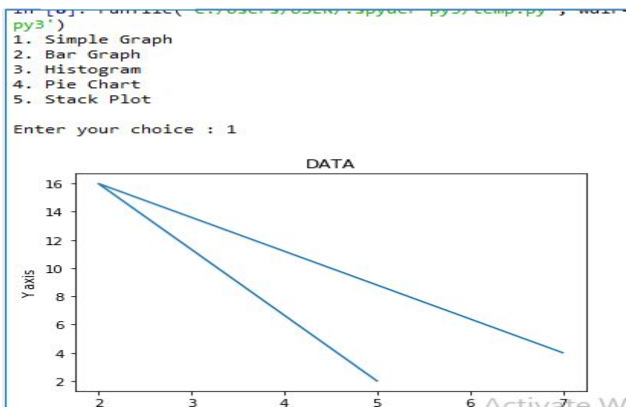
✓ Set up job
✓ Run actions/checkout@v1
✓ GitHub Actions for Apache Maven
✓ Run mvn clean install
✓ Complete job

10] Automate the Big Data pipeline till MapReduce using GitHub Actions

11] WAP in Java to implement MapReduce from JSON file extracted from crawler to find the frequency of significant words - Textual Analysis

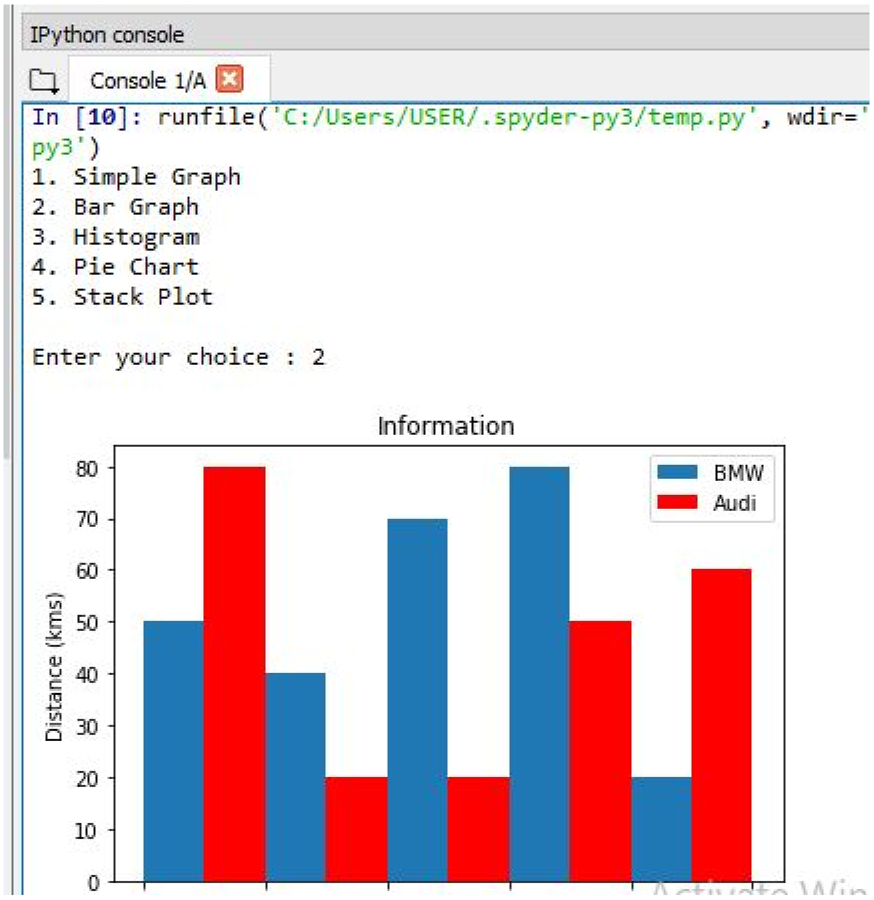
12] Data Prediction - Using the KNN algorithm in Python to find the relation between tweets and their sentiments.

13] Data Visualization - Using the Python matplotlib library to implement visualization on the chart of choice simple graph, histogram, bar graph, etc.


```
[illegible]
py3')
1. Simple Graph
2. Bar Graph
3. Histogram
4. Pie Chart
5. Stack Plot

Enter your choice : 1
```

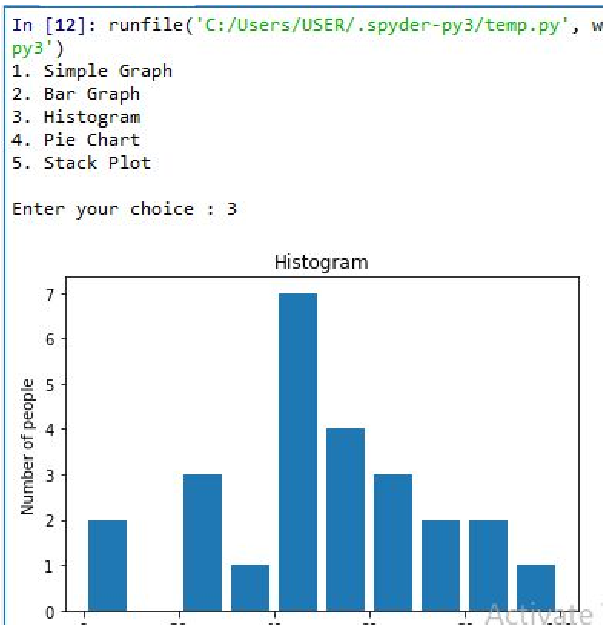
Figure: Simple Graph



```
In [10]: runfile('C:/Users/USER/.spyder-py3/temp.py', wdir='
py3')
1. Simple Graph
2. Bar Graph
3. Histogram
4. Pie Chart
5. Stack Plot

Enter your choice : 2
```

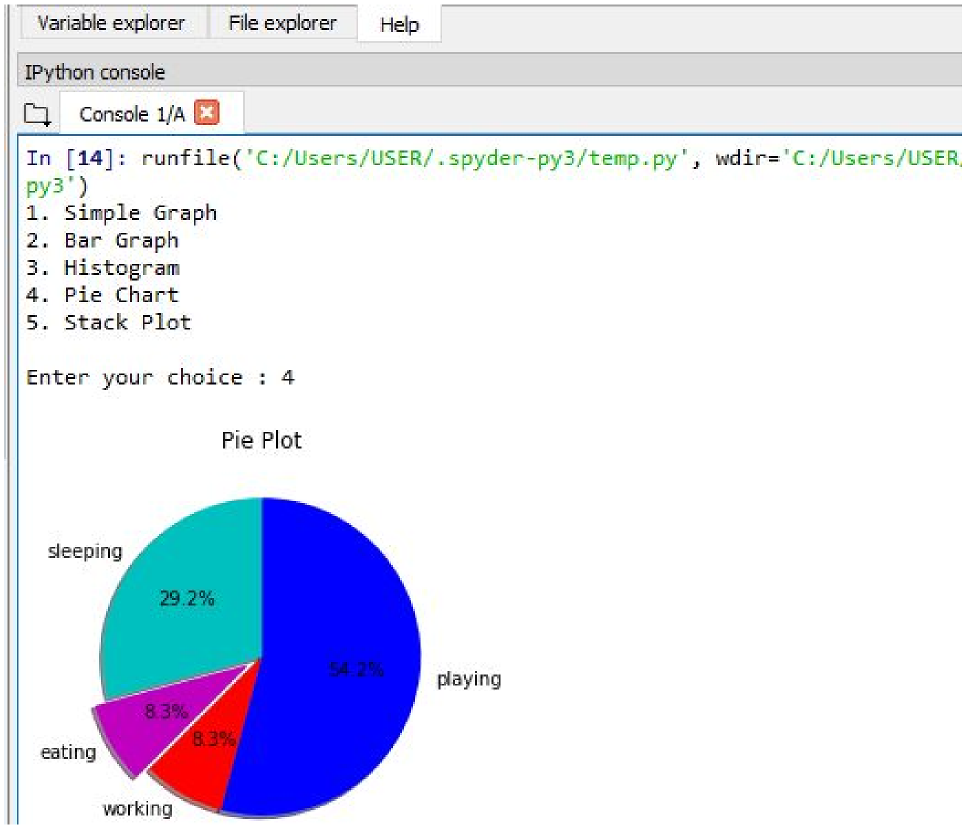
Figure: Bar Graph



```
In [12]: runfile('C:/Users/USER/.spyder-py3/temp.py', w
py3')
1. Simple Graph
2. Bar Graph
3. Histogram
4. Pie Chart
5. Stack Plot

Enter your choice : 3
```

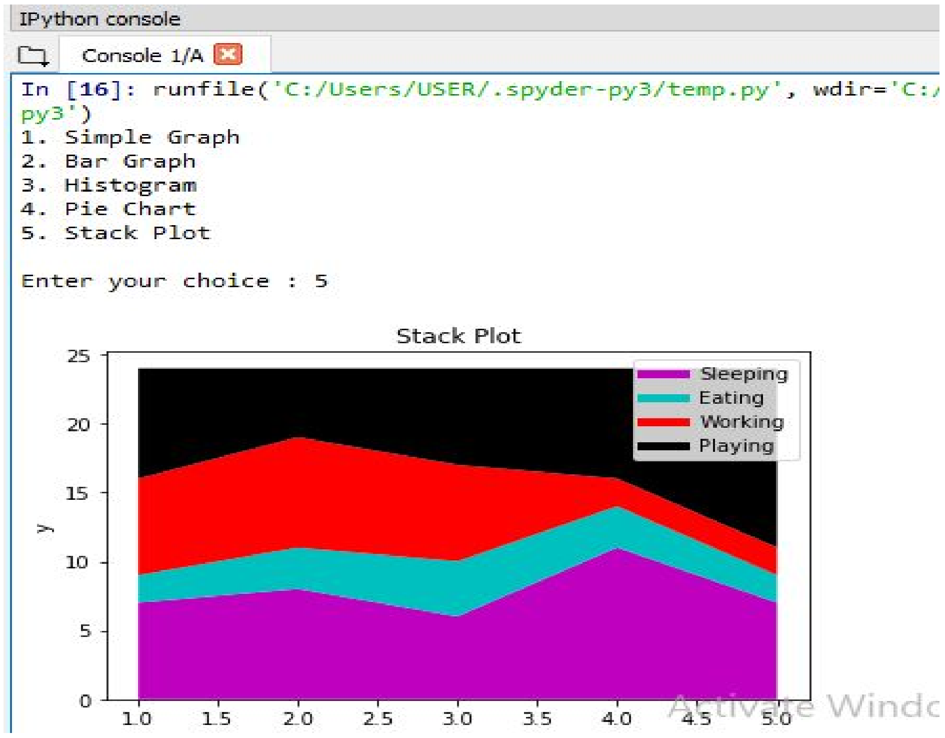
Figure: Histogram



```
In [14]: runfile('C:/Users/USER/.spyder-py3/temp.py', wdir='C:/Users/USER
py3')
1. Simple Graph
2. Bar Graph
3. Histogram
4. Pie Chart
5. Stack Plot

Enter your choice : 4
```

Figure: Pie Chart



```
In [16]: runfile('C:/Users/USER/.spyder-py3/temp.py', wdir='C:/
py3')
1. Simple Graph
2. Bar Graph
3. Histogram
4. Pie Chart
5. Stack Plot

Enter your choice : 5
```

Figure: Stack plot

References

[1] Lin, Jimmy, and Alek Kolcz. "Large-Scale Machine Learning at Twitter." In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, pp. 793-804. ACM, 2012.

[2] Liu, Bingwei, Erik Blasch, Yu Chen, Dan Shen, and Genshe Chen. "Scalable Sentiment Classification for Big Data Analysis Using Naive Bayes Classifier" In Big Data, 2013 IEEE International Conference on, pp. 99-104. IEEE, 2013.

[3] https://www.systemsvalley.com/our-obsession-with-scrum-methodology/

[4] ÁlvaroCuesta, David F., and María D. R-Moreno, "A Framework for Massive Twitter Data Extraction and Analysis", In Malaysian Journal of Computer Science, pp 50-67