Data Mining with Big Data using Hadoop Technology

Mamatha Balachandra* and Yash Mathur**
*Associate Professor, Dept. of CSE, M.I.T, Manipal, Manipal University
mamtha.bc@manipal.edu
** Student, VIII Sem , B.Tech, CSE, M.I.T, Manipal, Manipal University
mathur.yash@yahoo.com

Abstract: Big Data deals with large-volume, complex, growing data sets with multiple, autonomous sources. With the fast development of networking, data storage, and the data collection capacity, Big Data are now rapidly expanding in all science and engineering domains, including physical, biological and biomedical sciences. The Big Data challenges are broad in the case of accessing, storing, searching, sharing, and transfer. Managing Big Data is not easy by using traditional relational database management systems; it requires instead parallel computing of dataset. This work makes use of a HACE theorem that characterizes the features of the Big Data revolution, and proposes a Big Data processing model, from the data mining perspective. This data-driven model involves demand-driven aggregation of information sources, mining and analysis, user interest modeling, and security and privacy considerations. In this work the challenging issues in the data-driven model and also in the Big Data revolution are analyzed.

Keywords: Big Data, Hace theorem, Data Mining, Hadoop **,** Data Base.

Introduction

Every day, 2.5 quintillion bytes of data are created and 90 percent of the data in the world today were produced within the past two years. Our capability for data generation has never been so powerful and enormous ever since the invention of the information technology in the early 19th century. With the recent development of IT technology, the capacity of data has surpassed the Zetta byte, and improving the efficiency of business by increasing the predictive ability through an efficient analysis on these data has emerged as an issue of the current society.

As an example, on 4 October 2012, the first presidential debate between President Barack Obama and Governor Mitt Romney triggered more than 10 million tweets within 2 hours. Among all these tweets, the specific moments that generated the most discussions actually revealed the public interests, such as the discussions about medicare and vouchers. Such online discussions provide a new means to sense the public interests and generate feedback in real time, and are mostly appealing compared to generic media, such as radio or TV broadcasting.

The above examples show the rise of Big Data applications where data collection has grown extremely and is beyond the capability of commonly used software tools to capture, process and manage within a tolerable elapsed time. The biggest challenge for Big Data applications is explore the large amount of data and take out useful information from system and knowledge for future actions. In many situations the knowledge extraction process has to be very efficient and close to real time because storing all observed data is nearly infeasible. As a result the unmatched data volumes require an effective data analysis and also prediction platform to achieve fast response and real-time classification for such Big Data [1].

Related Works

Generally, data mining (sometimes called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

Data mining is the technology to extract the knowledge from the data. The data to be mined varies from a small data set to a large data set i.e. big data. The data Mining environment produces a large volume of the data. The information retrieved in the data Mining step is transformed into the structure that is easily understood by its user [2][3].

RHadoop is a collection of five R packages that allow users to manage and analyze data with Hadoop. The packages have been tested (and always before a release) on recent releases of the Cloudera and Hortonworks Hadoop distributions and should have broad compatibility with open source Hadoop and mapR's distribution.

Proposed work

Big Data starts with large-volume, heterogeneous, autonomous sources with distributed and decentralized control, and seeks to explore complex and evolving relationships among data. These characteristics make it an extreme challenge for discovering useful knowledge from the Big Data.

Data Mining challenges with Big Data

For an intelligent learning database system to handle Big Data, the essential key is to scale up to the exceptionally large volume of data and provide treatments for the characteristics featured by the aforementioned HACE theorem[4][5]. The data mining in big data mainly divided into three-tier structure, those are shown in Figure 1.

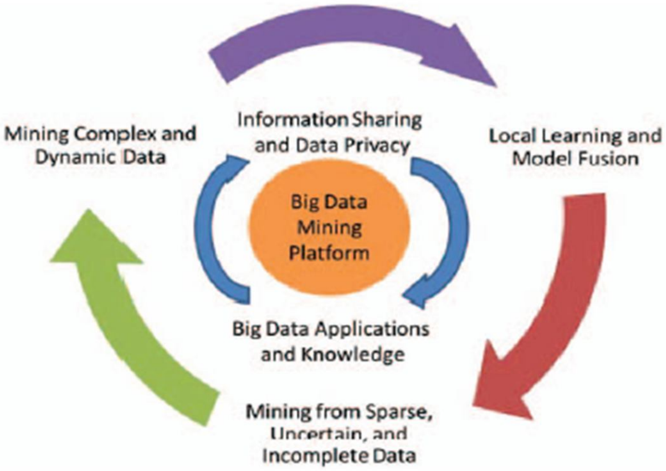

Mining Complex and Dynamic Data
Information Sharing and Data Privacy
Local Learning and Model Fusion
Big Data Mining Platform
Big Data Applications and Knowledge
Mining from Sparse, Uncertain, and Incomplete Data

Figure 1. A Big Data processing framework

The research challenges form a three tier structure and center around the "Big Data mining platform" (Tier I), which focuses on low-level data accessing and computing. Challenges on information sharing and privacy, and Big Data application domains and knowledge form Tier II, which concentrates on high-level semantics, application domain knowledge, and user privacy issues. The outmost circle shows Tier III challenges on actual mining algorithms.

The challenges at Tier I focus on data accessing and arithmetic computing procedures. Because Big Data are often stored at different locations and data volumes may continuously grow, an effective computing platform will have to take distributed large-scale data storage into consideration for computing. For example, typical data mining algorithms require all data to be loaded into the main memory, this, however, is becoming a clear technical barrier for Big Data because moving data across different locations is expensive (e.g., subject to intensive network communication and other IO costs), even if we do have a super large main memory to hold all data for computing.

The challenges at Tier II center around semantics and domain knowledge for different Big Data applications. Such information can provide additional benefits to the mining process, as well as add technical barriers to the Big Data access (Tier I) and mining algorithms (Tier III). For example, depending on different domain applications, the data privacy and information sharing mechanisms between data producers and data consumers can be significantly different. Sharing sensor network data for applications like water quality monitoring may not be discouraged, whereas releasing and sharing mobile users' location information is clearly not acceptable for majority, if not all, applications.

At Tier III, the data mining challenges concentrate on algorithm designs in tackling the difficulties raised by the Big Data volumes, distributed data distributions, and by complex and dynamic data characteristics. The circle at Tier III contains three stages. First, sparse, heterogeneous, uncertain, incomplete, and multisource data are preprocessed by data fusion techniques. Second, complex and dynamic data are mined after preprocessing. Third, the global knowledge obtained by local learning and model fusion is tested and relevant information is fed back to the preprocessing stage. Then, the model and parameters are adjusted according to the feedback. In the whole process, information sharing is not only a promise of smooth development of each stage, but also a purpose of Big Data processing.

Conceptual view of Big data processing framework

Figure 2 shows a conceptual view of the Big Data processing framework, which includes three tiers[6] from inside out with considerations on data accessing and computing (Tier I), data privacy and domain knowledge (Tier II), and Big Data mining algorithms (Tier III).

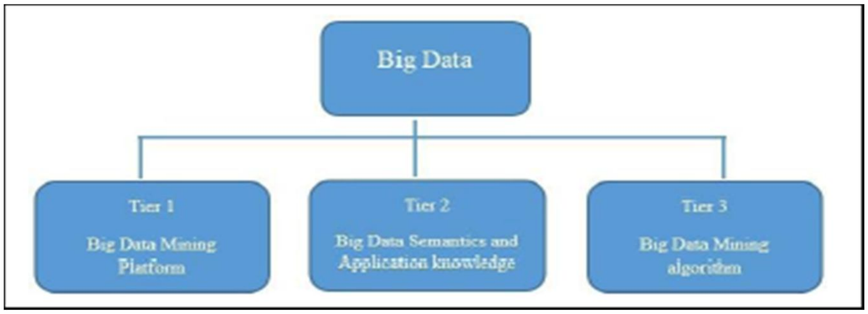

Big Data
Tier 1
Big Data Mining Platform
Tier 2
Big Data Semantics and Application knowledge
Tier 3
Big Data Mining algorithm

Figure 2. Three-Tier structure

At Tier I concentration on data accessing and doing arithmetic operation on them. Big data are not able to store in single place so that it is stored on diverse locations and day by day it constantly increasing, hence effective computing platform will have to take distributed large-scale data storage and also do arithmetic operation on them. So that for doing such type of operation common solutions are depend on parallel computing..

At Tier II focus on semantics and domain knowledge for different Big Data applications. Such information can provide benefits to the mining process and add technical barriers Tier I and data mining algorithms that is in Tier III. For example, rely on various domain applications, the information sharing and data privacy mechanisms between data producers and data consumers can be significantly different.

At Tier III, the data mining challenges concentrate on algorithm designs in tackling the difficulties raised by the Big Data volumes, complex and dynamic data characteristics and distributed data distributions. The circle at Tier III contains three stages. First uncertain and sparse, heterogeneous, and multisource data are preprocessed. Second dynamic and complex data are mined after preprocessing operation. Third the global knowledge get by local learning and relevant information is feedback to the preprocessing stage. Then the model and parameters are adjusted according to the feedback.

Working of overall system

Overall working of the system is shown in Figure 3. It shows all the phases of the data mining concept from the big data[6].

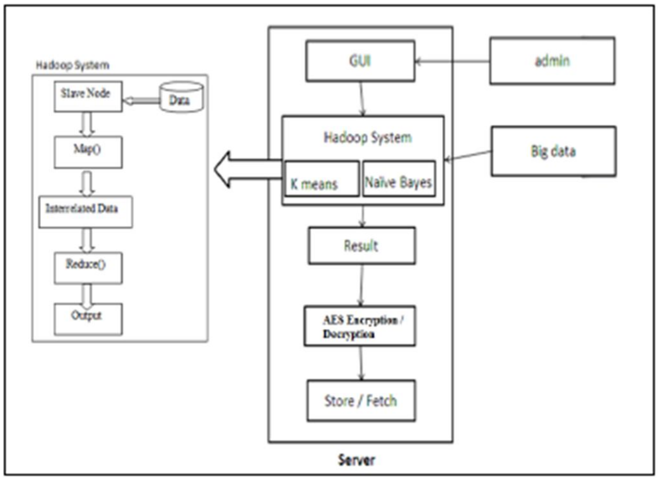


Figure 3. System Architecture

Admin is responsible for the fired the queries according to his need. When admin has fired the query, means admin is interacting with the GUI of the system. After that the Hadoop System is responsible for the processing the mining further.

Hadoop uses MapReduce programming model to mine data. This MapReduce program is used to separate datasets which are sent as input into independent subsets. Those are process parallel map task. Map() procedure that performs filtering and sorting. Reduce() procedure that performs a summary operation. After doing the MapReduce operation then whatever the output system created is given to the K-means or Naive Bayes algorithm for doing clustering and classification.

K-means algorithm or any other algorithm as per requirement can be used. Here implement with K-means algorithm in consideration is explained:

- Partition of a dataset into given k non-empty set.

- Identification of cluster mean point called centroids for the current partition.
- Assign each point to a specific cluster.
- The minimum distance from each point to centroid is computed, and then points are allotted to the cluster
- Computation of distance between each point and allocation of minimum distanced points from mean point to cluster.
- Repeat the above steps for re-allotted points and find the mean point for the new cluster.

Naïve Bayes Algorithm based on Bayes theorem and frequency table. It gives the Estimation, Classification, and Prediction. It is used when large data set. It is very easy to construct. Not using complicated iterative parameter estimations. It solves the zero frequency problems. It uses the following formula as:

$$P(Y|X_1,\dots,X_n) = \frac{\overset{\text{Likelihood}}{P(X_1,\dots,X_n|Y)}\overset{\text{Prior}}{P(Y)}}{\underset{\text{Normalization Constant}}{P(X_1,\dots,X_n)}}$$

AES is symmetric key based encryption algorithm that means the same key is used for both encrypting and decrypting the data. It has mainly three types as AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks. There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys a round consists of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of cipher text. The use of AES algorithm is optional since it just provide security for the accessing of the data set and the system, any other algorithm or methodology can be used to provide security as per the requirement of the system, data set and its intended user.

Figure 4 shows the flow of overall proposed system. Figure 5 shows the complete step wise working of twitter posts analysis. Here we use data set collected from Twitter to do analysis, any other data set can be used as per the requirement of the miner with a few changes made to the mining algorithm so as to perform efficient knowledge discovery from the data. Big Data which is impossible to store on a single workstation is made available through online repositories which can be accessed using Hadoop framework. Thus, the data set used for implementation of analysis is rather small in size but, by the combined usage of Hadoop and online repositories Big Data can be mined for efficient knowledge discovery.

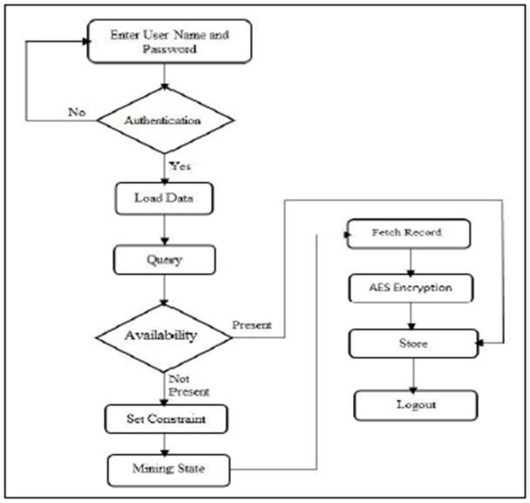


Figure 4. The flow of overall proposed system

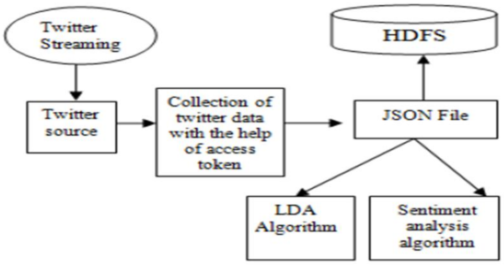

Twitter Streaming
Twitter source
Collection of twitter data with the help of access token
JSON File
HDFS
LDA Algorithm
Sentiment analysis algorithm

Figure 5. Installation and workflow

Twitter and twitter REST API

Twitter is an online social networking service that enables users to send and read short 140-character messages called "tweets". Registered users can read and post tweets, but unregistered users can only read them.

A REST API defines a set of functions which developers can perform requests and receive responses via HTTP protocol such as GET and POST. Twitter provides a REST API which you can query to get the latest tweets, you can provide a search query (or hash tag) and it will return the results in JSON format.

The information can be collected using both forms of Twitter API. Requests to the APIs contain parameters which can include hash tags, keywords, geographic regions, and Twitter user IDs. Responses from Twitter APIs is in JavaScript Object Notation (JSON) format. JSON is a popular format that is widely used as an object notation on the web. Twitter APIs can be accessed only via authenticated requests. Twitter uses Open Authentication and each request must be signed with valid Twitter user credentials. Open Authentication (OAuth) is an open standard for authentication, adopted by Twitter to provide access to protected information. The authentication of API requests on Twitter is carried out using OAuth. Twitter APIs can only be accessed by applications. Below we detail the steps for making an API call from a Twitter application using OAuth:

1) Applications are also known as consumers and all applications are required to register themselves with Twitter4. Through this process the application is issued a consumer key and secret which the application must use to authenticate itself to Twitter.

2) The application uses the consumer key and secret to create a unique Twitter link to which a user is directed for authentication. The user authorizes the application by authenticating himself to Twitter. Twitter verifies the user's identity and issues a OAuth verifier also called a PIN.

3) The user provides this PIN to the application. The application uses the PIN to request an "Access Token" and "Access Secret" unique to the user.

4) Using the "Access Token" and "Access Secret", the application authenticates the user on Twitter and issues API calls on behalf of the user.

The "Access Token" and "Access Secret" for a user do not change and can be cached by the application for future requests. Thus, this process only needs to be performed once, and it can be easily accomplished using the method GetUserAccessKeySecret. Figure 6 shows OAuth workflow. Create Access token [Open Authentication (OAuth)] for collection twitter data:

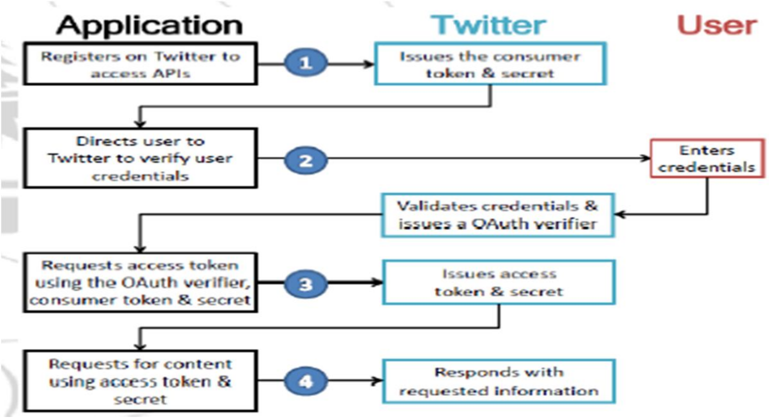

Application
Twitter
User
Registers on Twitter to access APIs
1
Issues the consumer token & secret
Directs user to Twitter to verify user credentials
2
Enters credentials
Validates credentials & issues a OAuth verifier
Requests access token using the OAuth verifier, consumer token & secret
3
Issues access token & secret
Requests for content using access token & secret
4
Responds with requested information

Figure 6. OAuth workflow

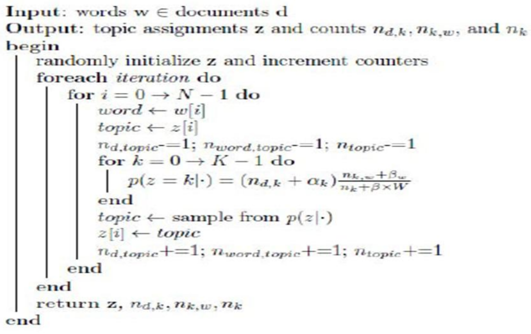
Hadoop

Hadoop is a open source framework that developed by apache software foundation. Hadoop are the most widely used models used today for Big Data processing. Hadoop is an open source large-scale data processing framework that supports distributed processing of large chunks of data using simple programming models. The Apache Hadoop project consists of the HDFS and Hadoop Map Reduce in addition to other modules. The software is modeled to harvest upon the processing power of clustered computing while managing failures at node level. All the modules in Hadoop are designed with a fundamental assumption that hardware failures are common and should be automatically handled by the framework. Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Apache Hadoop MapReduce and HDFS components were inspired by Google papers on their MapReduce and Google File System Hadoop splits files into large blocks and distributes them across nodes in a cluster. To process data, Hadoop transfers packaged code for nodes to process in parallel based on the data that needs to be processed. This approach takes advantage of data locality nodes manipulating the data they have access to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

Latent Dirichlet allocation (LDA)

LDA is a collection of words. Each topic contains all of the words in the corpus with a probability of the word belonging to that topic. It involves setting up the requisite count variables, randomly initializing them, and then running a loop over the desired number of iterations where on each loop a topic is sampled for each word instance in the corpus. Following the Gibbs iterations, the counts can be used to compute the latent distributions _d and _k.

The only required count variables include nd;k, the number of words assigned to topic k in document d; and nk;w, the number of times word w is assigned to topic k. However, for simplicity and e_ciency, we also keep a running count of nk, the total number of times any word is assigned to topic k. Finally, in addition to the obvious variables such as a representation of the corpus (w), we need an array z which will contain the current topic assignment for each of the N words in the corpus. Figure 7 shows Latent Dirichlet Allocation.



```
Input: words w $\in$ documents d
Output: topic assignments z and counts $n_{d,k}, n_{k,w}$, and $n_k$
begin
    randomly initialize z and increment counters
    foreach iteration do
        for $i = 0 \rightarrow N - 1$ do
            $word \leftarrow w[i]$
            $topic \leftarrow z[i]$
            $n_{d,topic}$-=1; $n_{word,topic}$-=1; $n_{topic}$-=1
            for $k = 0 \rightarrow K - 1$ do
                $p(z = k|\cdot) = (n_{d,k} + \alpha_k)\frac{n_{k,w}+\beta_w}{n_k+\beta \times W}$
            end
            $topic \leftarrow$ sample from $p(z|\cdot)$
            $z[i] \leftarrow topic$
            $n_{d,topic}$+=1; $n_{word,topic}$+=1; $n_{topic}$+=1
        end
    end
    return z, $n_{d,k}, n_{k,w}, n_k$
end
```

Figure 7. Latent Dirichlet Allocation

Sentiment analysis algorithm

Sentiment Analysis is to detect the polarity of text in consideration in textual form. It is also known as opinion mining as it derives the opinion of the speaker or the user about some topic. The sentiment analysis algorithm we use in our project is based on a Naive Bayes Classifier. Since Naive Bayes is fast, space efficient, and not sensitive to irrelevant features, in this research we used the Naive Bayes classifier which is based on Bayes' theorem

P(w|T)=(P(w).P(T|w))/P(T)

where w is a sentiment word, T is a Twitter message Bayes's theorem is based on strong independence assumptions. Therefore, the probabilistic model for a classifier can be described as:

R=P(positive|T) – P(negative|T);

R=P(positive)P(T|positive) – P(negative)P(T|negative);

R=P(positive)πP(T|positive) – P(negative)πP(T|negative).

Comparing the probabilities P(positive|T) and P(negative|T), the larger probability indicates that the class label value has a higher probability to be actual label. If R is larger than 0, then predict positive attitude is more likely to be true, otherwise, predict negative attitude has more likely to be true.

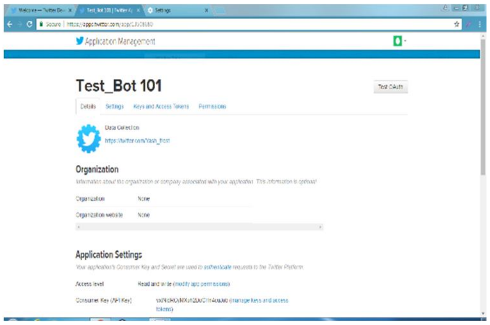
During the sentiment analysis, the Naive Bayes classifier classifies a Tweet into a positive class or a negative class by comparing the words in each Tweet. Each word will be labeled with "positive" and "negative" coming from the lexicon. In the Naive Bayes classification, the number of sentiment words is counted. If more positive words are used than negative in a Tweet, then the Tweet could be labelled as positive, otherwise if less positive words presented in a Tweet than negative ones, the Tweet could be labelled as negative. A neutral label word is ignored in this study since it contains no valuable information for sentiment analysis.

Results and analysis

Create New Twitter[7][8] Application

- Login to Twitter and Create a new Twitter app at http://apps.twitter.com (Figure 8)
- Enter the name of the application, brief description about the application and callback URL. (Figure 9)
    - Name: Name of the application as given by the user.
    - Description: Description of apps' basic functionality.
    - Website: Domain from where application is made available.

- o Callback URL: URL to which application redirects (Figure 9).



Test_Bot 101

Data Collection
https://twitter.com/Yash_frost

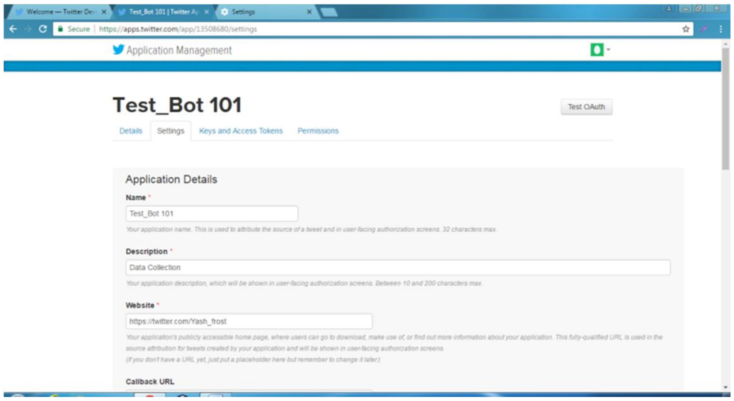
Organization
Information about the organization or company associated with your application. This information is optional.

Organization None
Organization website None

Application Settings
Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.

Access level Read and write (modify app permissions)
Consumer Key (API Key) [illegible] (manage keys and access tokens)

Figure *8.* Create a new Twitter app



https://apps.twitter.com/app/13508680/settings

Test_Bot 101

Application Details

Name *
Test_Bot 101
Your application name. This is used to attribute the source of a tweet and in user-facing authorization screens. 32 characters max.

Description *
Data Collection
Your application description, which will be shown in user-facing authorization screens. Between 10 and 200 characters max.

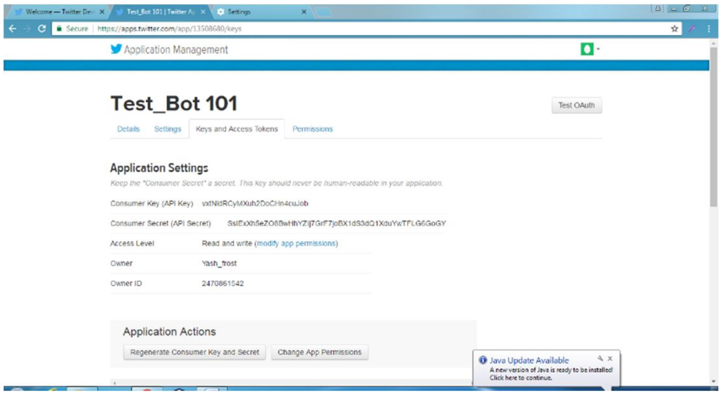
Website *
https://twitter.com/Yash_frost
Your application's publicly accessible home page, where users can go to download, make use of, or find out more information about your application. This fully-qualified URL is used in the source attribution for tweets created by your application and will be shown in user-facing authorization screens.
(If you don't have a URL yet, just put a placeholder here but remember to change it later.)

Callback URL

Figure 9. Callback URL

Getting Twitter API keys Figures and Tables

After creating a new application , the application generates four unique tokens namely API key, API secret, Access token, Access token secret which enables us to collect data by authorizing it as shown in Figure 10.



https://apps.twitter.com/app/13508680/keys

Test_Bot 101

Application Settings
Keep the "Consumer Secret" a secret. This key should never be human-readable in your application.

Consumer Key (API Key) vxtNIdRCyMXuh2DoCHn4cuJob
Consumer Secret (API Secret) SslExXh5eZO8BwHhYZIj7GrF7joBX1dS3dQ1XduYwTFLG6GoGY
Access Level Read and write (modify app permissions)
Owner Yash_frost
Owner ID 2470861542

Application Actions

Java Update Available
A new version of Java is ready to be installed.
Click here to continue.

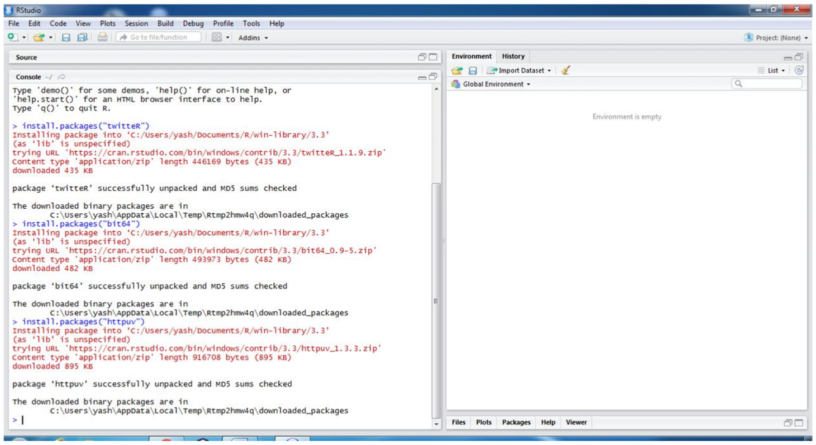
Figure 10. Twitter API keys

Install and Load Packages

Before using the preceding keys, some packages need to be installed and loaded to access the data in R using the application created using following code:

install.packages(“twitter”) //provides access to the Twitter API

install.packages(“bit64”) //provides serializable S3 atomic 64 bit (signed) integers that can be used in vectors, matrices, arrays and data frames install.packages("httpuv") //provides low-level socket and protocol support for handling HTTP and Web Socket requests directly from within R. It is primarily intended as a building block for other packages. Figure 11 shows Connecting to Twitter Streaming API:



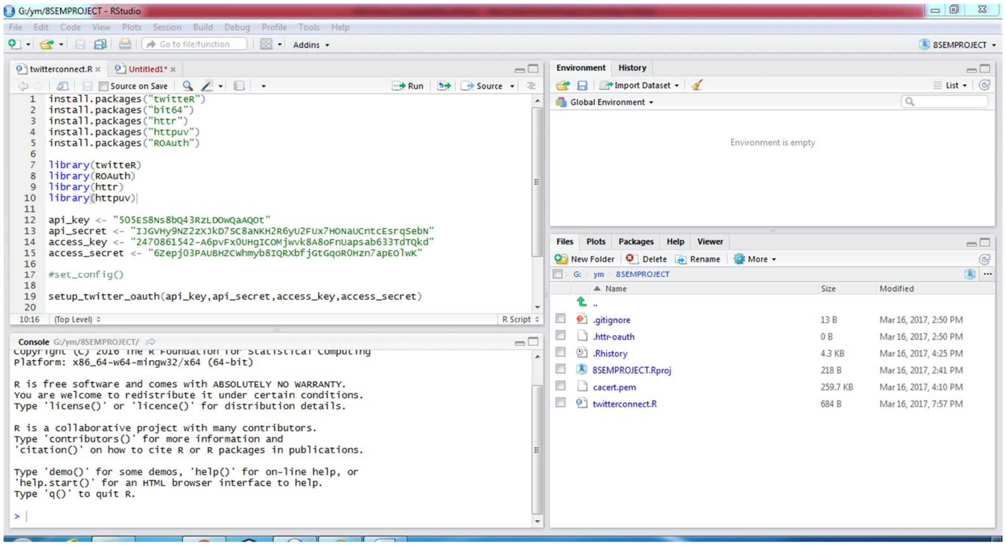
```
Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("twitteR")
Installing package into 'C:/Users/yash/Documents/R/win-library/3.3'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/twitteR_1.1.9.zip'
Content type 'application/zip' length 446169 bytes (435 KB)
downloaded 435 KB

package 'twitteR' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\yash\AppData\Local\Temp\Rtmp2hmw4q\downloaded_packages
> install.packages("bit64")
Installing package into 'C:/Users/yash/Documents/R/win-library/3.3'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/bit64_0.9-5.zip'
Content type 'application/zip' length 493973 bytes (482 KB)
downloaded 482 KB

package 'bit64' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\yash\AppData\Local\Temp\Rtmp2hmw4q\downloaded_packages
> install.packages("httpuv")
Installing package into 'C:/Users/yash/Documents/R/win-library/3.3'
(as 'lib' is unspecified)
trying URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/httpuv_1.3.3.zip'
Content type 'application/zip' length 916708 bytes (895 KB)
downloaded 895 KB

package 'httpuv' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\yash\AppData\Local\Temp\Rtmp2hmw4q\downloaded_packages
> |
```

Figure 11. Installing and Loading Packages

Connecting to Twitter Streaming API(OAuth Interface)

OAuth Interface is used to get an app authorized by the user and access its resources on Twitter. It uses unique key and access tokens generated by the application. Figure 12 shows Connecting to Twitter Streaming API.

api_key = "XXXXXX"
api_secret = "XXXXX"
access_key = "XXXXX"
access_secret = "XXXXXX"
setup_twitter_oauth(api_key,api_secret)



```
install.packages("twitteR")
install.packages("bit64")
install.packages("httr")
install.packages("httpuv")
install.packages("ROAuth")

library(twitteR)
library(ROAuth)
library(httr)
library(httpuv)

api_key <- "505ES8Ns8bQ43RzLDOwQaAQOt"
api_secret <- "IJGVHy9NZ2zXJkD7SC8aNKH2R6yU2FUx7HONaUCntcEsrqSebN"
access_key <- "2470861542-A6pvFx0UHgICOMjwvk8A8oFnUapsab633TdTQkd"
access_secret <- "6Zepj03PAUBHZCWhmyb8IQRXbfjGtGqoR0Hzn7apEOlwK"

#set_config()

setup_twitter_oauth(api_key,api_secret,access_key,access_secret)
```

```
Copyright (C) 2016 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

Name	Size	Modified
..		
.gitignore	13 B	Mar 16, 2017, 2:50 PM
.httr-oauth	0 B	Mar 16, 2017, 2:50 PM
.Rhistory	4.3 KB	Mar 16, 2017, 4:25 PM
8SEMPROJECT.Rproj	218 B	Mar 16, 2017, 2:41 PM
cacert.pem	259.7 KB	Mar 16, 2017, 4:10 PM
twitterconnect.R	684 B	Mar 16, 2017, 7:57 PM

Figure 12. Connecting to Twitter Streaming API

Conclusion

During the sentiment analysis, the Naive Bayes classifier classifies a Tweet into a positive class or a negative class by comparing the words in each Tweet. Each word will be labeled with "positive" and "negative" coming from the lexicon. In the Naive Bayes classification, the number of sentiment words is counted. If more positive words are used than negative in a Tweet, then the Tweet could be labelled as positive, otherwise if less positive words presented in a Tweet than negative ones, the Tweet could be labelled as negative. A neutral label word is ignored in this study since it contains no valuable information for sentiment analysis. The work presented shows the various steps and result analysis of Twitter data.

References

[1] G. Q. Wu, X. Wu, X. Zhu “Data mining with big data,” IEEE Trans. on Knowledge and Data Engineering, vol. 26, no. 1, pp. 97-107, 2014.

[2] Data Mining: What Is Data Mining? Essay - EssaysForStudent.com, https://www.essaysforstudent.com , 2010.

[3] Bogdan BatrincaPhilip C. Treleaven ,“Social media analytics: a survey of techniques, tools and platforms”, Volume 30, pp 89–116, 2015

[4] Prema Gadling, Mahip Bartere “Implementing HACE Theorem for Big Data Processing”, International Journal of Science and Research (IJSR) ISSN (Online): 2319-7064., 2015

[5] Deepak S. Tamhane, Sultana N. Sayyad, “BIG DATA ANALYSIS USING HACE THEOREM”,International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 4 Issue 1, ISSN: 2278 – 1323, January 2015 .

[6] “Data Mining and Information Security in Big Data Using HACE Theorem” International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395 -0056 p-ISSN: 2395-0072 © 2016.

[7] https://www.r-bloggers.com/search/twitter/ for r programming.

[8] https://dev.twitter.com/streaming/overview for API Streaming in twitter