TWITTER DATA ANALYSIS USING SPARK

A Project

Presented to the faculty of the Department of Computer Science

California State University, Sacramento

Submitted in partial satisfaction of
the requirements for the degree of

MASTER OF SCIENCE

in

Computer Science

by

Yash Bopardikar

FALL
2016

TWITTER DATA ANALYSIS USING SPARK

A Project

by

Yash Bopardikar

Approved by:

_____, Committee Chair
Dr. Jun Dai

_____, Second Reader
Dr. Ying Jin

_____
Date

Student: Yash Bopardikar

I certify that this student has met the requirements for format contained in the University format manual, and that this project is suitable for shelving in the Library and credit is to be awarded for the project.

_____, Graduate Coordinator _____
Dr. Ying Jin Date

Department of Computer Science

Abstract

of

TWITTER DATA ANALYSIS USING SPARK

by

Yash Bopardikar

The idea is to solve people's dilemma for choosing a certain product or service over other. The objective is to collect data from tweeter feeds on trending topics like various OS, new technologies, different Products etc. and to categories them according to need. This Project would use Scala and Spark cluster. All the Big Data that would be fetched would be collected in MySQL database. Then using the MapReduce API's and Data Mining algorithms data will be classified accordingly. Here we would be using Naïve Bayes Algorithm for Sentimental analysis over the data. Twitter provides a streaming API to stream real time twitter data. This library named Twitter4j is available in python to access Streaming API and download twitter data. The filter on this data is based on a list of keywords supplied. This analysis will employee a distributed data processing system known as Apache Spark using several worker and master nodes. This cluster is scalable and can handle millions of records.

To filter out the huge number of data we will use Map reduce technique on Spark. The input file will contain a JSON object for each tweet in data. This file will be uploaded on Spark frame structure. The Spark frame structure will have replicated the structure and distribute to multiple nodes, thus the mapper takes all files presented in the directory and

classifies them according to the filter set. These filtered tweets will pass through the data mining algorithms and thus we could observe one to one comparison of data which would be helpful to take tough decisions.

_____, Committee Chair
Dr. Jun Dai

_____
Date

ACKNOLEDGEMENTS

I would like to thank Dr. Jun Dai, my advisor for providing me an opportunity to work on this project and also motivate me to do better and better. His constant encouragement and interest in making best out of such a powerful platform helped me throughout the project. In addition, I would also like to thank Dr. Ying Jin for her willingness to serve on the committee. Last but not the least, I would like to thank the entire faculty and staff of Department of Computer Science Engineering at California State University, Sacramento.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter 1

INTRODUCTION

1.1 Overview

In today's world there is a lot of informative data available in the social media. This data has been a powerful resource for data miners to get deep insights. Collecting this data and analyzing it might give us a lot of useful information which can help solve a lot of our day to day problems. Taking this idea into consideration, I have harnessed a tool which collects data from Twitter and analyses it to produce intelligent information. Suppose we use this tool according to a consumer/buyer perspective, it can be used in the following way. This app collects the Tweets that relate to brands, products or topics set as filter and gives us a review as to which is the best for a given geographical location, what are the sentiments associated with it at a particular location, which brand, product or topic is most trending, what the trend looks like on different dates, how popular it is and how many followers it has and thus helping the consumer to make wise choices. Consider buying a gaming console as our target mission, we want to analyze gaming consoles that belong to three brands say Xbox, PlayStation, Nintendo. This tool is not just limited to gaming consoles but can be used to get reviews and insights for any product or any trending topic for example phones, cars, electronic devices. The results can be visualized it various graphs like heat map, bar graph, trending graph, density graph, etc.

1.2 Introduction to Big Data

Today, we live in the digital world. Because of the growing digitization and innovations in technologies, the amount of data (structured or unstructured) and data collection have also tremendously increased. The data is deposited in databases which grow substantially and become difficult to collect, manage, share, examine and visualize via traditional database software tools [1]. This is the reason for emergence of big data and a recent area of strategic investment for IT organizations.

Big Data can be defined as a collection of data which may be structured or unstructured and so large and complicated that it may become complex to process it using simple systems or traditional data processing applications [2]

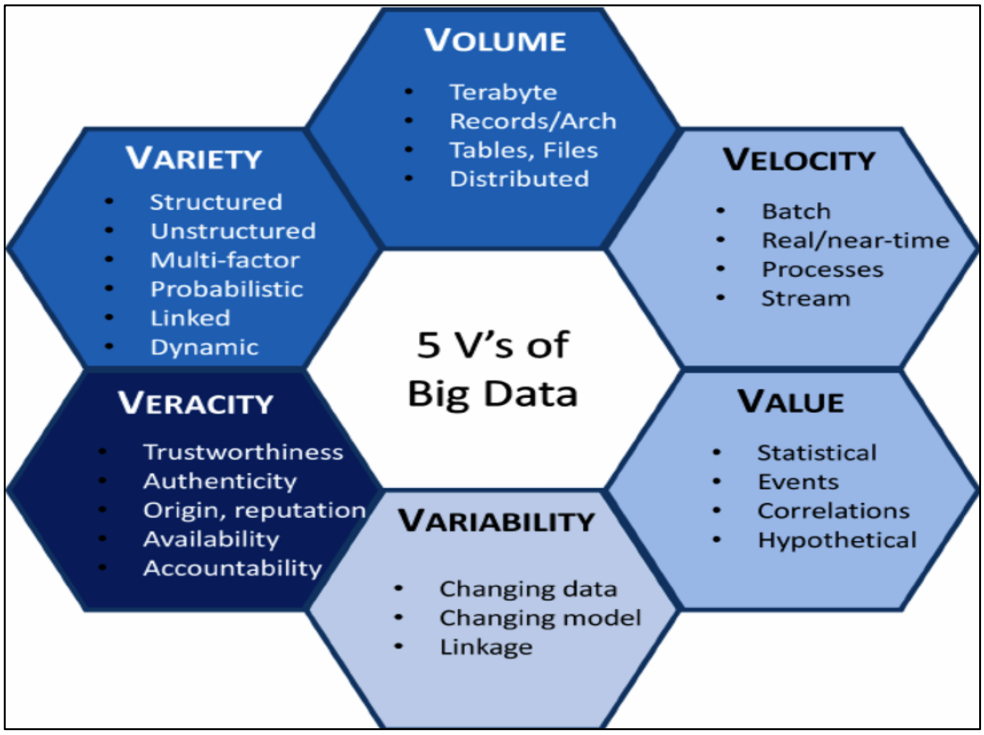


Figure 1 V's of Big Data [3]

The characteristics of Big Data are the following [3]:

- Volume

Volume [3] refers to huge amount of data transformed every second which is not possible for traditional systems to manage. For example, consider a social networking website like Facebook where people upload pictures, send messages, comments and click the Like button billion times on daily basis.

- Variety

The data like structured or unstructured, videos, audios, images, text messages, tables, graphs etc., are referred to as Variety. These different forms of data can be collected via various sources like social media, sensors, mobile devices, government datasets, user based etc. This data can have many dimensions and also various forms.

- Velocity

The speed to generate new data and move around are stated as Velocity. For example, the messages or videos on social media that's going viral within a minute and quickly detect the suspicious activity on credit card transactions. Systems designed to handle big data can handle and process millions of rows per second, which makes it easier to get desired insights out of this data.

- Veracity

Veracity refers to the reliability or disorderliness of the data [3]. The quality and accuracy are not easy to manage with many forms of big data. For example, the post on Twitter or Facebook with many hashtags, typos, and informal speech. The system has many worker

nodes and threads running to complete a desired task which work in synchronization to maintain the reliability of data.

- Value

Value refers to the business cost which is obtained through managing big data. For example, managing very large unstructured data from blogs or social media stream. The value of data depends on how accurate are the insights obtained from the data, the processing time, how deep the mining is done and how it is visualized.

1.3 Sources of Big Data

With the evolution and innovation of technology, the amount of generated data is also increased. Also, the data needs to be collected into variety of formats. Sources of Big Data can be generally classified into six different categories:

- Enterprise Data

There is humongous amount of data spread across the various businesses, companies, and institutions in different formats known as Enterprise Data. The formats include emails, word documents, spreadsheets, presentations, HTML pages, pdf files, XMLs, flat files, legacy formats, Json, csv, tables, graphs, etc.

- Transactional Data

There are countless applications like web applications, mobile applications, CRM systems, banking systems, service providing platforms, etc., in every organization which

involve in different kinds of transactions. So, the relational databases can be used as a backend to reinforce these applications.

- Social Media

Social networks like Facebook, Twitter, etc., produce vast amount of data every day which are unstructured data which can be text, images, video, and others. This data can be cleansed, ordered and transformed into a structured data to make it useful data.

- Activity Generated

The data being generated by machines are referred to as Activity generated data. The origin of these kinds of data comes from satellites, medical devices, sensor data, industrial machinery, surveillance videos, cell phone towers, etc. This data may be present in Log files, log tables, Json format and can be used to get future predictions of the system.

- Public Data

The publicly available data like Wikipedia, data from weather department, data published by research institutes, open data-sets provided by government and other types of data which are easily available and accessible to public at no cost. This type of data is called Public Data.

- Archives

In today's world, all insignificant data are being achieved by enterprises. As hardware getting cheaper day-by-day, the enterprises can pile up the data. This kind of data includes scanned documents and agreements, records of ex-employees, project documents, and all banking records.

1.4 Why we need Big Data?

A normal system hardware and software are not capable to deal with very large amount of various types of data which are created and collected at such a high speed. Big data is the term for large and complicated data sets that it becomes difficult for traditional data warehousing to store, analyze, manage, process and work on them and visualize. The insights gained by processing big data can be significant for a business, can help consumers, can be used to get predictive models for natural calamities and avoid them, and can be used to predict behavioral patterns and trends. That is the reason of big data and its analysis is the main focus modern science and business.

1.5 What is Big Data Analytics?

“Big data analytics is the process of examining large data sets containing a variety of data types, like big data, to uncover hidden patterns, unknown correlations, market trends, customer preferences and other useful business information.” [4]

Using Big Data Analysis important insights can be gained. For this project we use Big Data Analytic for getting insights from twitter data.

Chapter 2

OVERVIEW TECHNOLOGIES AND PLATFORM USED

2.1 Twitter

Twitter [5] is a social networking platform used to share small bits of information across the world. It is the most popular micro blogging website in today's world. The 140 characters' messages/posts are called tweet and people can follow each other to receive other person's tweets. They can tag other using @ symbol.

Like all social media forum these tweets can include URLs, photographs, and hashtags. A person can share another person's tweet (re-tweet) if he is following the later. Hashtag (#) is used as a trendsetter in twitter, and used to easily look for information on a particular topic while collecting data from twitter.

Twitter has more than billion registered user accounts and around 317 million monthly active Twitter users [5]. It contains massive amount of data and included users from all fields like movie stars, brand reviews, news, sportsmen, common people, politicians, etc. thus giving us the global outlook of people around the world. Thus collecting the expressions of people from different walks of life, can give us a clear picture on a particular topic.

2.2 Apache Spark

Spark [6] is a frame work which provides cluster computing platform to perform various tasks like data analytics, machine learning, data streaming, database management, parallel computing, graph operations, etc. Spark can run as a standalone cluster as well as in cluster mode.

Spark uses RDD (resilient distributed dataset) to distribute items over the cluster. RDD's are read only datasets and are handled by spark for the purpose of maintaining its fault tolerant behavior. Aside from RDD's spark also has DataFrames which are table like structures used by spark to store data in table format, for manipulation of data in dataframes spark also provides SQL libraries and functions. Spark has Spark SQL library which can be used to perform SQL queries on dataframes.

Apache Spark supports Java, Scala, Python and R [6]. Spark core is the main engine which manages input output operations, scheduling, memory management, networking interfaces and dataset (RDD based).

The spark structure can be understood from the diagram below:

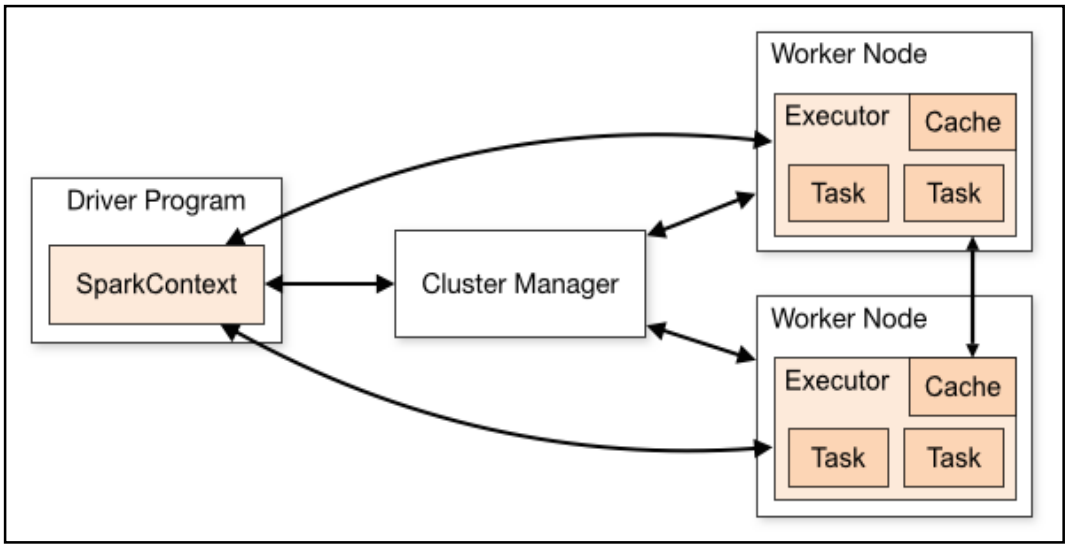


Figure 2: Spark Architecture [7]

- Driver Program

Driver program [7] manages the allocation of tasks to the worker nodes. It is also called as Master. The driver does the task of listening to worker nodes for possible incoming messages. It also keeps the task isolated so there is not data leak between the tasks. It prioritizes the task of managing the jobs submitted to the spark cluster.

- Cluster Manager

Cluster manager [7] distributes the task to various worker nodes. It is like the intermediator between the driver program and worker nodes which manages the cluster when it is distributed. The cluster manager handles the request when either od driver or

worker node is busy. This makes spark capable of fault tolerant. The cluster manager supports various applications and package handlers

- Worker Node

Each worker node [7] has an executor which can perform many tasks. Every worker has a cache memory allocated which is configurable. Every executor isolates the task so there is no memory leak between the multiple tasks submitted. The only way share memory between two tasks is to write it to an external memory.

- Executor

Executor [7] is the process initiated for an execution of an application. Each application running on the cluster has its own executors. The executor is responsible for data keeping the data and doing input output operations between the applications.

- Task

Task [7] is a unit of work assigned by an executor.

A spark cluster can be configured with various parameter settings. The number of executors, worker nodes, memory allocation, cache memory, worker cores, executor cores everything can be configured according to the system requirements.

Apache spark provides various other functionalities like:

2.2.1 Spark Streaming

Spark Streaming is an add on package to the core Spark API which is scalable, high-throughput and also fault-tolerant. It provides functionality of processing live data streams. [8]. For streaming Apache spark can have flume, HDFS, apache Kafka, twitter, kinesis data sources. This data can be then cleaned and structured in spark itself and used to do further processing.

Figure 3: Spark Streaming [8]

2.2.2 Spark SQL

Spark core has an SQL extension which supports more optimization on datasets(RDD) and is in structured format to retrieve data using the SQL queries. Spark SQL provides most convenient way to perform several transitions on the data. Spark SQL [9] uses dataframes for data manipulations.

2.2.3 Spark ML Lib (Machine learning libraries)

Spark ML Lib [10] provides wide range of advanced machine learning libraries. Spark has two kinds of libraries which perform operations on RDDs and DataFrames. Spark can process large amount of data and perform advanced machine learning algorithms on it. Spark supports clustering, classification, reduction, regression, etc.

2.3 SBT

"SBT is an open source build tool for Scala and Java projects." [11]. It is tool for compiling Scala code and integrating various Scala frameworks. SBT builds a jar file of the project which includes all the dependencies needed for project to run. This jar then can be deployed on various frameworks as a complete application with dependencies.

2.4 Tableau

Tableau [12] is data visualization software. Data visualization is one of the important features of Big Data. Visualization views can project data in different dimensions which can make an impact. Tableau is a powerful tool which can project data in many formats and graphs. It has the power to extract data from various data sources as well as stream data from data sources to provide a live view. Tableau has custom settings for color, toot-tips, text representation, scale, etc. It has various versions which can be accessed online as well as of-line as a desktop version.

Chapter3

SYSTEM DESIGN

3.1 Spark Cluster Setup

I have set up spark cluster in standalone mode on my laptop which has 8GB memory(RAM), 512GB storage and intel i5 processor. This spark cluster is build using maven which is included in the spark package. Below are the spark environment variables set for this cluster.

```
export SPARK_WORKER_INSTANCES="4"
export SPARK_EXECUTOR_INSTANCES="5"
export SPARK_WORKER_MEMORY="8g"
export SPARK_EXECUTOR_MEMORY="8g"
export SPARK_EXECUTOR_CORES="6"
export SPARK_WORKER_CORES="10"
```

Here I have set spark worker instances to 4, which initiates 4 worker instances are soon as the spark cluster is started. The Executor instances are set to 5 and each worker is allocated 8 GB of memory i.e. the spark cluster can use the total amount of memory available on the system. The spark Executor and Worker cores are set to 6 and 10 which are virtual cores.

Spark has a WebUI interface which provides visualizations for the processing and resources allocated. I have set the WebUI port to 8080 on localhost.

The WebUI looks as follows for the system configured.

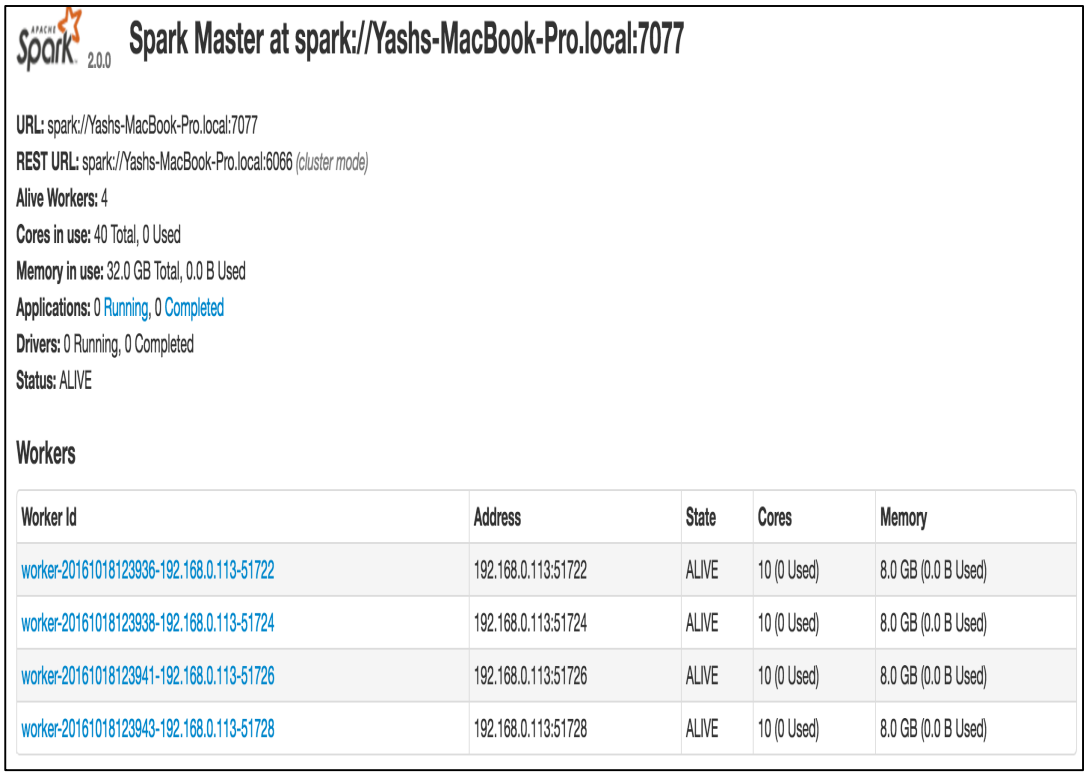


Figure 4: Spark Master WebUI

Spark 2.0.0 **Spark Worker at 10.0.0.14:49848**

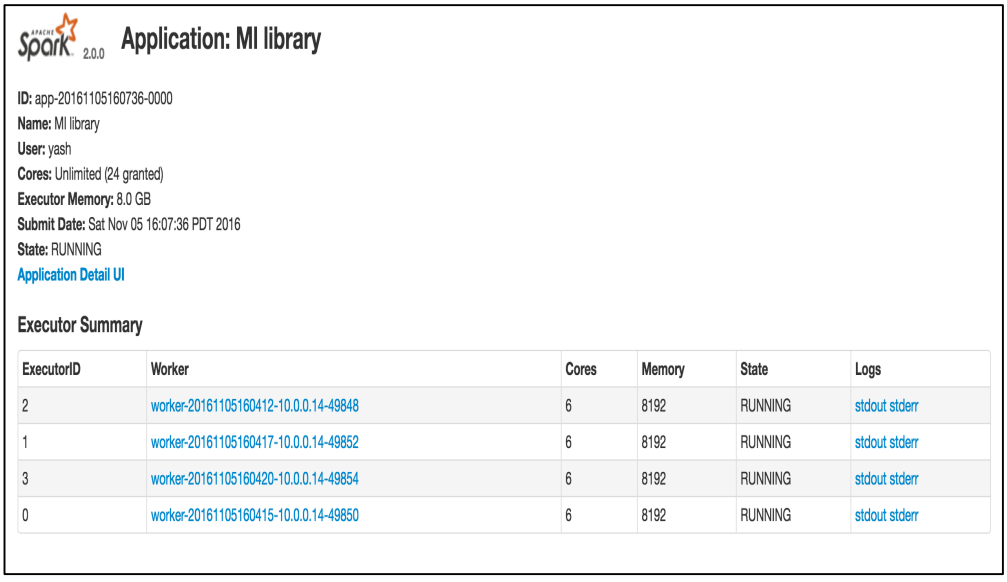
ID: worker-20161105160412-10.0.0.14-49848
Master URL: spark://Yashs-MacBook-Pro.local:7077
Cores: 10 (6 Used)
Memory: 8.0 GB (8.0 GB Used)

Back to Master

Running Executors (1)

ExecutorID	Cores	State	Memory	Job Details	Logs
2	6	RUNNING	8.0 GB	**ID:** app-20161105160736-0000 **Name:** MI library **User:** yash	stdout stderr

Figure 5: Spark Worker UI



Spark 2.0.0 **Application: MI library**

ID: app-20161105160736-0000
Name: MI library
User: yash
Cores: Unlimited (24 granted)
Executor Memory: 8.0 GB
Submit Date: Sat Nov 05 16:07:36 PDT 2016
State: RUNNING
Application Detail UI

Executor Summary

ExecutorID	Worker	Cores	Memory	State	Logs
2	worker-20161105160412-10.0.0.14-49848	6	8192	RUNNING	stdout stderr
1	worker-20161105160417-10.0.0.14-49852	6	8192	RUNNING	stdout stderr
3	worker-20161105160420-10.0.0.14-49854	6	8192	RUNNING	stdout stderr
0	worker-20161105160415-10.0.0.14-49850	6	8192	RUNNING	stdout stderr

Figure 6: Spark Application Window

3.2 System Flow Diagram

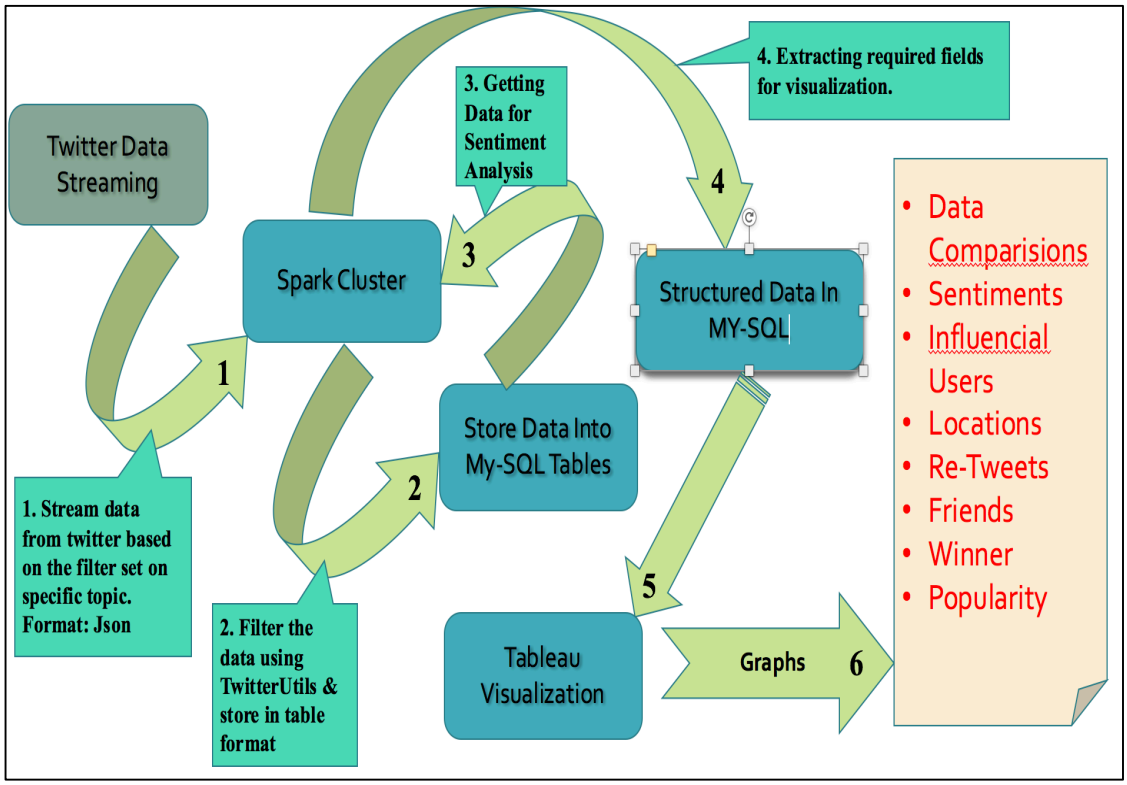


Figure 7: System Flow Diagram

Chapter 4

IMPLEMENTATION

4.1 Streaming Twitter Data Using Spark

We use spark streaming to stream live spark data. To load Twitter data into Apache Spark twitter provides an interface to developers which can be used to access twitter data. Visit twitters applications site to register your application "https://apps.twitter.com/" [13]. I have noted twitter tokens into TwitterKeys.txt which are needed to initialize spark streaming context.

The streaming data is captured into batches. The interval for downloading batches can be set, I have set the interval to be 10 seconds. Therefore, every 10 seconds a new batch of data is streamed and captured into batches.

We set the filter which contains set of keywords to filter the tweets. Only tweets with those keywords present in it would be streamed.

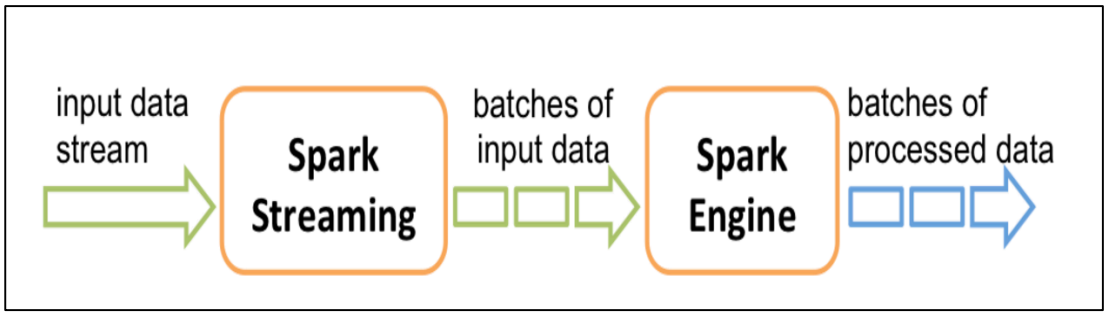


Figure 8: Spark Streaming [8]

```
object DhruvTwitter extends App {

  var sc: SparkContext = new SparkContext(new SparkConf())
  val sqlContext = new SQLContext(sc)

  def loadTwitterKeys() = {
    val lines: Iterator[String] = Source.fromFile("/Users/yash/Downloads/spark-1.6.1/twitter/TwitterKeys.txt").getLines()
    val props = lines.map(line => line.split("=")).map { case (scala.Array(k, v)) => (k, v)}
    props.foreach {
      case (k: String, v: String) => System.setProperty(k, v)
    }
  }

  def configureStreamingContext() = {
    val conf = new SparkConf().setMaster("local[*]")
    sc = new SparkContext(conf)
    new StreamingContext(sc, Seconds(10))
  }

  def startStream() = {

    val duration: Duration = Seconds(10)
    val filters = Seq("ps4","ps4k","ps4neo","ps4pro","psneo","playstation4","playstation4k","playstation4neo",
    "Xbox","Xbonone","xbox1","Xbox","xbox360","nintendo","nintendowii","nintendowiiu","wiiu","nintendostill",
     "#nes","wii","nintendo3ds")

    val ssc = configureStreamingContext()
    val tweets = TwitterUtils.createStream(ssc,None,filters,StorageLevel.MEMORY_ONLY_SER_2)
```

4.2 Understanding Data

The far most and initial stage in data mining is to perceive and recognize the problem and to note down the objectives [14]. We have equipped ourselves with domain knowledge to recognize the problem complication, it will tremendously improve data mining effectiveness and potency. We have thoroughly understood the sources and types of related data and we have gathered, collected some useful data. We are focusing on finding the right insights from data.

The data we receive from twitter streaming is in Json format. This data needs to be understood before using it, selecting the fields form this data is one of the important

aspects. We use twitter4j library which is embedded into TwitterUtils library for getting specific fields out of the data. The raw Json looks as follows:

```
StatusJSONImpl
{
  createdAt=Mon Oct 10 09:25:20 PDT 2016,
  id=785516575143190528,
  text='Sick infographic! https://t.co/5q1WyCQ1DW',
  source='<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>',
  isTruncated=false,
  inReplyToStatusId=-1,
  inReplyToUserId=-1,
   isFavorited=false,
   isRetweeted=false,
   favoriteCount=0,
   inReplyToScreenName='null',
   geoLocation=null,
   place=PlaceJSONImpl
                {name='Buckhall',
                 streetAddress='null',
                 countryCode='US',
                 id='015ac8fc158ff38e',
                 country='United States',
                 placeType='city',
                 url='https://api.twitter.com/1.1/geo/id/015ac8fc158ff38e.json',
                 fullName='Buckhall,
                 VA',
                 boundingBoxType='Polygon',
                 boundingBoxCoordinates=[[Ltwitter4j.GeoLocation;@6af7d3c4],
                 geometryType='null',
                 geometryCoordinates=null,
                 containedWithIn=null},
        retweetCount=0,
        isPossiblySensitive=false,
        lang='en',
        contributorsIDs=[],
        retweetedStatus=null,
        userMentionEntities=[],
        urlEntities=[URLEntityJSONImpl
                      {url='https://t.co/5q1WyCQ1DW',
                       expandedURL='https://twitter.com/johngreen/status/785295173110599680',
                       displayURL='twitter.com/johngreen/stat…'}],
                       hashtagEntities=[],
                       mediaEntities=[],
                       symbolEntities=[],
                       currentUserRetweetId=-1,
```

```
user=UserJSONImpl{
        id=281533233,
        name='Caroline Lemp',
        screenName='clempsophia',
        location='STL',
        description='try to travel, learn, smile, and be passionate about life.
        living in STL, prior DC girl and all-around world traveler',
        isContributorsEnabled=false,
        profileImageUrl='http://pbs.twimg.com/profile_images/734422924615094272/l_u3wNjE_normal.jpg',
        profileImageUrlHttps='https://pbs.twimg.com/profile_images/734422924615094272/l_u3wNjE_normal.jpg',
        isDefaultProfileImage=false,
        url='http://clempsophia.tumblr.com/',
        isProtected=false,
        followersCount=97,
        status=null,
        profileBackgroundColor='B86800',
        profileTextColor='0C6D70',
        profileLinkColor='0084B4',
        profileSidebarFillColor='C0DFEC',
        profileSidebarBorderColor='A8C7F7',
        profileUseBackgroundImage=true,
        isDefaultProfile=false,
        showAllInlineMedia=false,
        friendsCount=212,
        createdAt=Wed Apr 13 06:29:32 PDT 2011,
        favouritesCount=89,
        utcOffset=-14400,
        timeZone='Eastern Time (US & Canada)',
        profileBackgroundImageUrl='http://pbs.twimg.com/profile_background_images/236939565/free-vector-world-map.gif',
        profileBackgroundImageUrlHttps='https://pbs.twimg.com/profile_background_images/236939565/free-vector-world-map.gif',
        profileBackgroundTiled=true,
        lang='en',
        statusesCount=1172,
        isGeoEnabled=true,
        isVerified=false,
        translator=false,
        listedCount=2,
        isFollowRequestSent=false,
        withheldInCountries=null
    },
    withHeldInCountries=null,
    quotedStatusId=785295173110599680,
quotedStatus=null}}
```

4.3 Pre- Processing Data

The collected data were noisy, missing useful info and inconsistent [14]. We have almost finished the Data preparation process. In this process, we have to check if there are empty values or inconsistency in the data. The data should be in a consistent state to be analyzed. To improve the efficiency of our analysis the data should be in a simple format. Data mining is done on this data so to get efficient results the data must be processed by removing redundancies. The data is made meaningful by deriving information from it like deriving dates out of months, days and year. For example, Deriving the age of the tweet by its date.

A schema of organized data can be seen as follows

```
root
 |-- console: string (nullable = true)
 |-- location: string (nullable = false)
 |-- sentiment: string (nullable = false)
 |-- retweet: integer (nullable = true)
 |-- follower: integer (nullable = true)
 |-- favourite: integer (nullable = true)
 |-- date_s: string (nullable = true)
 |-- month: integer (nullable = true)
 |-- day: integer (nullable = true)
 |-- sensative: boolean (nullable = true)
```

Figure 9: DataFrame Schema

4.4 Algorithm and Sentiment Analysis

Sentiment analysis is performed using Stanford's Natural Language Processing library [15]. It takes text input and sends this library to get the sentiment in return. This library constructs a tree like structure out of the plain text passed to it, this structured is created after cleaning the data and removing all the stop words.

```
val df = tweets.cache().map { value => {
  val sentiment = SentimentAnalysisUtils.detectSentiment(value.getText())
  println(sentiment.toString + "SENTIMENTS")
  val tags = value.getHashtagEntities.map(_.getText.toLowerCase)
  (value, sentiment.toString, tags)
}
}
```

Calling function SentimentAnalysisUtils with parameters as plain text from the tweet. The returned value is saved into map so that the status id tagged with its sentiment and can be used further. This is the code for filtering the

```
def onlyWords(text: String) : String = {
  text.split(" ").filter(_.matches("^[a-zA-Z0-9 ]+$")).fold("")((a,b) => a + " " + b).trim
}
```

After we filter the code and get the plain text we pass it to the function which creates a tree out of it for getting the sentiment score. The function is defined as follows:

```
def detectSentiment(message: String): SENTIMENT_TYPE = {


  val pipeline = new edu.stanford.nlp.pipeline.StanfordCoreNLP(nlpProps)

  val filteredTweet=onlyWords(message)

  val annotation = pipeline.process(filteredTweet)
  var sentiments: ListBuffer[Double] = ListBuffer[Double]()
  var sizes: ListBuffer[Int] = ListBuffer[Int]()

  var longest = 0
  var mainSentiment = 0

  for (sentence <- annotation.get(classOf[CoreAnnotations.SentencesAnnotation])) {
    val tree = sentence.get(classOf[edu.stanford.nlp.sentiment.SentimentCoreAnnotations.SentimentAnnotatedTree])
    val sentiment = RNNCoreAnnotations.getPredictedClass(tree)
    val partText = sentence.toString

    if (partText.length() > longest) {
      mainSentiment = sentiment
      longest = partText.length()
    }

    sentiments += sentiment.toDouble
    sizes += partText.length

  }
```

The score is counted by taking calculating the average i.e. by dividing the sum of score by the size of sentence as each sub tree and each word has a score associated to it.

```
  val averageSentiment:Double = {
    if(sentiments.size > 0) sentiments.sum / sentiments.size
    else -1
  }

  val weightedSentiments = (sentiments, sizes).zipped.map((sentiment, size) => sentiment * size)
  var weightedSentiment = weightedSentiments.sum / (sizes.fold(0)(_ + _))

  if(sentiments.size == 0) {
    mainSentiment = -1
    weightedSentiment = -1
  }


     println("debug: main: " + mainSentiment)
     println("debug: avg: " + averageSentiment)
     println("debug: weighted: " + weightedSentiment)
  /*
   0 -> very negative
   1 -> negative
   2 -> neutral
   3 -> positive
   4 -> very positive
   */
```

The score is then used to get the weighted sentiment out of the status and then mapped to the sentiment based on the score. Here we have stated that 0 score is very negative, 1 is associated to negative, 2 is neutral, 3 is positive and 4 is very positive.

4.4 Developed Model

The model should be reviewed and checked before using it for obtaining results. The results obtained after mining the data should be analyzed meticulously and interpreted by experts in order to perform efficient data analysis.

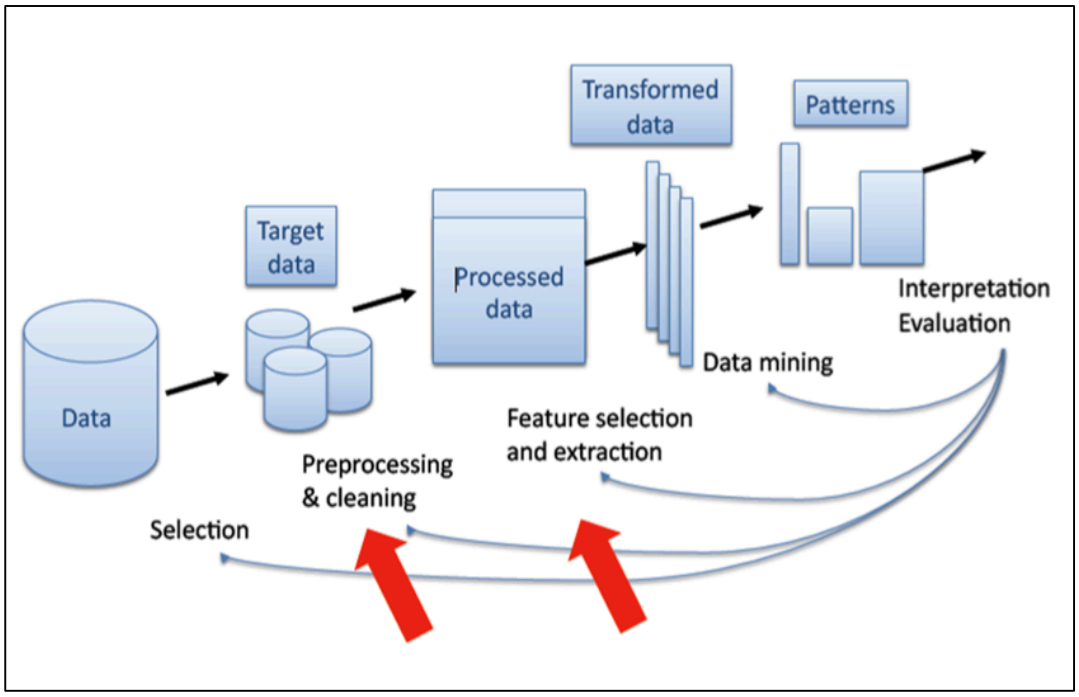


Figure 10: Basic Workflow Model [16]

Above figure describes the steps involved in our project. Basic flow diagram of how we use our raw data to extract relevant information. The extracted information is saved into case classes and then saved into data frames. The data frame is created using the structure we need.

Case Class:

```
case class Store(phone:String,sentiment:String,location:String,retweet:Integer,follower:Integer,favourite:Integer,
                 date_s:String,
                 month:Integer,day:Integer,sensetive:Boolean)
val storeMap=new scala.collection.mutable.HashMap[String,Store]()
```

Constructing DataFrames from schema string. The data types of each column needs to be specified while creating the schema of for data frame.

```
// Generate the schema based on the string of schema
val schemaString = "console location sentiment retweet follower favourite date_s month day sensative"
val fields = schemaString.split(" ")
  .map(fieldName => {
   if(fieldName.contentEquals("retweet")||fieldName.contentEquals("follower") || fieldName.contentEquals("favourite")
   ||fieldName.contentEquals("month") || fieldName.contentEquals("day") ){
     StructField(fieldName, IntegerType, nullable = true)
   }
   else if(fieldName.contentEquals("sensative")){
     StructField(fieldName, BooleanType, nullable = true)
   }
   else {
     StructField(fieldName, StringType, nullable = true)
   }
  })
```

A dataframe looks like a table. When we use show () method on a dataframe it looks like.

```
Sat Nov 05 16:06:49 PDT 2016
+----------+--------------------+---------+-------+--------+---------+---------+-----+---+---------+
|   console|            location|sentiment|retweet|follower|favourite|   date_s|month|day|sensative|
+----------+--------------------+---------+-------+--------+---------+---------+-----+---+---------+
|  nintendo|               Empty|  NEUTRAL|     14|     114|        0|5/11/2016|   11|  5|    false|
|playstation|               Empty| NEGATIVE|    490|      74|        0|5/11/2016|   11|  5|    false|
|playstation|           Austin TX|  NEUTRAL|      0|       4|        0|5/11/2016|   11|  5|    false|
|  nintendo|               Empty| NEGATIVE|      0|       5|        0|5/11/2016|   11|  5|    false|
|  nintendo|      West Yorkshire| NEGATIVE|     16|     244|        0|5/11/2016|   11|  5|    false|
|       xbox|       New York, USA|  NEUTRAL|      0|       3|        0|5/11/2016|   11|  5|    false|
|playstation|             Denmark| POSITIVE|      0|   11778|        0|5/11/2016|   11|  5|    false|
|playstation|       US and Canada|  NEUTRAL|      0|     517|        0|5/11/2016|   11|  5|    false|
|playstation|               Empty| POSITIVE|      0|      54|        0|5/11/2016|   11|  5|    false|
|       xbox|i follow everyone...|  NEUTRAL|      0|      94|        0|5/11/2016|   11|  5|    false|
|  nintendo|                横須賀| POSITIVE|      0|      92|        0|5/11/2016|   11|  5|    false|
|playstation|             Ireland|  NEUTRAL|      0|     172|        0|5/11/2016|   11|  5|    false|
|playstation|               Empty|  NEUTRAL|      0|       0|        0|5/11/2016|   11|  5|    false|
|       xbox|          Yakima, WA|  NEUTRAL|      0|    1260|        0|5/11/2016|   11|  5|    false|
|playstation|               Empty|  NEUTRAL|    157|     194|        0|5/11/2016|   11|  5|    false|
|playstation|Are we legit? Lin...|  NEUTRAL|      0|   12892|        0|5/11/2016|   11|  5|    false|
|playstation|          lucky land| NEGATIVE|      0|     153|        0|5/11/2016|   11|  5|    false|
|  nintendo|               Empty| POSITIVE|      0|    1618|        0|5/11/2016|   11|  5|    false|
|  nintendo|               Empty| NEGATIVE|   2807|       2|        0|5/11/2016|   11|  5|    false|
|  nintendo|                 513|  NEUTRAL|      0|     549|        0|5/11/2016|   11|  5|    false|
+----------+--------------------+---------+-------+--------+---------+---------+-----+---+---------+
only showing top 20 rows
```

Figure 11: DataFrame

4.5 Saving to Database

I am using MySQL to store the huge data. MySQL is widely used and is very efficient in storing tabular data. It is easy to store data into MySQL using JDBC connection. The connection needs a driver which then makes a connection to the database and pushes the data into desired table.

```
val schema = StructType(fields)
val sqlContext = new SQLContext(sc)
import sqlContext.implicits._
locDS.foreachRDD(rdd=>{
  try{
    var newRDD=rdd.map( x => Row(x.phone,x.location,x.sentiment,x.retweet,x.follower,x.favourite,x.date_s,x.month,x.day,x.sensetive))
    val df=sqlContext.createDataFrame(newRDD, schema).na.fill("Empty", Seq("location","sentiment"))
    df.printSchema()
    df.cache()
    df.show()
    val url="jdbc:mysql://localhost:3306/yashtest?user=yash&password=password"
    val prop = new java.util.Properties()
    prop.setProperty("JDBC_TXN_ISOLATION_LEVEL","READ_COMMITTED")
    prop.setProperty("driver","com.mysql.jdbc.Driver")
    val df1 = df.write.mode("append")
    val dfwriter = df.write.mode("append")
    dfwriter.jdbc(url,"yashtest.games",prop)

  }catch{
    case e:Exception=>e.printStackTrace()
  }
}
)
```

Every time a new data frame is created it gets pushed and appended to the existing table in the database. The driver needs MySQL server URL and password along with table and database name to store data. In order for the data to be isolated we need to set the isolation property to read committed.

Chapter 5

EXECUTION

5.1 Building Project

Launching the project on spark cluster needs a jar file. I use SBT as the package builder for generating a .JAR of the project. SBT is widely used for building Scala and Java projects. This JAR downloads all the dependencies needed for execution. This jar is called FAT JAR or UBER JAR which contains all the code segments and dependencies.

```
Yashs-MacBook-Pro:DhruvTwitter yash$ sbt clean assembly
[info] Loading project definition from /Users/yash/Downloads/spark-2.0.0/DhruvTwitter/project
[info] Updating {file:/Users/yash/Downloads/spark-2.0.0/DhruvTwitter/project/}dhruvtwitter-build...
[info] Resolving org.fusesource.jansi#jansi;1.4 ...
[info] Done updating.
[info] Set current project to DhruvTwitter (in build file:/Users/yash/Downloads/spark-2.0.0/DhruvTwitter/)
[success] Total time: 2 s, completed Nov 6, 2016 1:03:10 AM
[info] Updating {file:/Users/yash/Downloads/spark-2.0.0/DhruvTwitter/}dhruvtwitter...
[info] Resolving jline#jline;2.12.1 ...
[info] Done updating.
[info] Compiling 2 Scala sources to /Users/yash/Downloads/spark-2.0.0/DhruvTwitter/target/scala-2.11/classes...
[warn] there were 18 deprecation warnings; re-run with -deprecation for details
[warn] one warning found
[info] Including: ejml-0.23.jar
[info] Including: xercesImpl-2.8.0.jar
[info] Including: javax.json-api-1.0.jar
[info] Including: scalatest_2.11-2.2.6.jar
[info] Including: scala-library-2.11.8.jar
[info] Including: slf4j-api-1.7.12.jar
[info] Including: spark-streaming-twitter_2.11-2.0.0.jar
[info] Including: spark-tags_2.11-2.0.0.jar
[info] Including: unused-1.0.0.jar
[info] Including: twitter4j-stream-4.0.4.jar
[info] Run completed in 228 milliseconds.
[info] Total number of tests run: 0
[info] Suites: completed 0, aborted 0
[info] Tests: succeeded 0, failed 0, canceled 0, ignored 0, pending 0
[info] No tests were executed.
[info] Including: twitter4j-core-4.0.4.jar
[info] Including: scala-reflect-2.11.8.jar
[info] Including: xalan-2.7.0.jar
[info] Including: joda-time-2.9.jar
[info] Including: jollyday-0.4.7.jar
[info] Including: jaxb-api-2.2.7.jar
[info] Including: stanford-corenlp-3.6.0.jar
[info] Including: scala-xml_2.11-1.0.2.jar
[info] Including: xom-1.2.10.jar
[info] Including: xml-apis-1.3.03.jar
[info] Including: stanford-corenlp-3.6.0-models.jar
[info] Checking every *.class/*.jar file's SHA-1.
[info] Merging files...
```

5.2 Submitting Job

Spark cluster needs to be started before submitting any job to the cluster. All worker nodes to be initiated while starting the cluster. Spark contains shell code for starting master, starting workers, stopping master, stopping workers, starting all, stopping all, etc.

```
Yashs-MacBook-Pro:sbin yash$ sh start-all.sh
org.apache.spark.deploy.master.Master running as process 38792.  Stop it first.
Password:
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /Users/yash/Downloads/spark-2.0.0/logs/spark-yash-org.apache.spark.deploy.worker.Worker-1-Yashs-MacBook-Pro.local.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /Users/yash/Downloads/spark-2.0.0/logs/spark-yash-org.apache.spark.deploy.worker.Worker-2-Yashs-MacBook-Pro.local.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /Users/yash/Downloads/spark-2.0.0/logs/spark-yash-org.apache.spark.deploy.worker.Worker-3-Yashs-MacBook-Pro.local.out
localhost: starting org.apache.spark.deploy.worker.Worker, logging to /Users/yash/Downloads/spark-2.0.0/logs/spark-yash-org.apache.spark.deploy.worker.Worker-4-Yashs-MacBook-Pro.local.out
Yashs-MacBook-Pro:sbin yash$
```

```
Yashs-MacBook-Pro:sbin yash$ sh stop-all.sh
Password:
localhost: stopping org.apache.spark.deploy.worker.Worker
localhost: stopping org.apache.spark.deploy.worker.Worker
localhost: stopping org.apache.spark.deploy.worker.Worker
localhost: stopping org.apache.spark.deploy.worker.Worker
stopping org.apache.spark.deploy.master.Master
Yashs-MacBook-Pro:sbin yash$
```

```
Yashs-MacBook-Pro:bin yash$ sh spark-submit --master "spark://Yashs-MacBook-Pro.local:7077" /Users/yash/Downloads/spark-2.0.0/DhruvTwitter/target/scala-2.11/DhruvTwitter-assembly-1.0.jar --class DhruvTwitter
root
 |-- console: string (nullable = true)
 |-- location: string (nullable = false)
 |-- sentiment: string (nullable = false)
 |-- retweet: integer (nullable = true)
 |-- follower: integer (nullable = true)
 |-- favourite: integer (nullable = true)
 |-- date_s: string (nullable = true)
 |-- month: integer (nullable = true)
 |-- day: integer (nullable = true)
 |-- sensative: boolean (nullable = true)

+-------+--------+---------+-------+--------+---------+------+-----+---+---------+
|console|location|sentiment|retweet|follower|favourite|date_s|month|day|sensative|
+-------+--------+---------+-------+--------+---------+------+-----+---+---------+
+-------+--------+---------+-------+--------+---------+------+-----+---+---------+
```

5.3 Tableau

Setting up tableau with the data source [12]. Tableau has two modes which are live and extract.

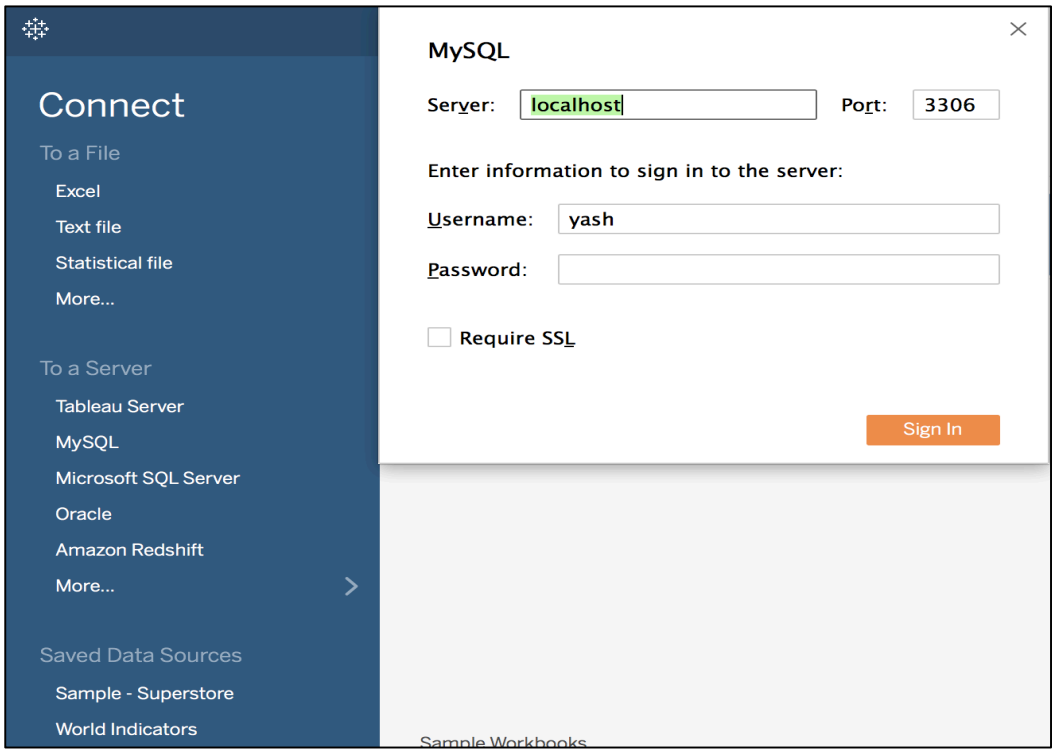


Figure 12: Tableau Connection

Chapter 6

USE-CASES AND RESULTS

6.1 IPhone vs Samsung Note7

One of the most observed dilemmas for common people is comparison between Apple IPhone and Samsung (android phones). [17] The application would filter the live tweets based on the filters set. Getting all the features from the clean data sentiment analysis would be done on the data set. This comparison is made based on sentiments base on the tweet, their locations and other factors. This comparison gives us a brief idea of how, what and where people think on these products (Apple and Samsung)

This would answer questions like:

1. Where is IPhone most Popular?
2. Where is android most Popular?
3. What do peoples sentiments about IPhone and android?
4. How many people on an average from 50,000(Tweets collected) use which phone?
5. How many influential people follow either of them?
6. Are there any trend changes after the recent changes in models?

The observations below are based upon data collected on different dates and the total size of data base is 15000 rows plus.

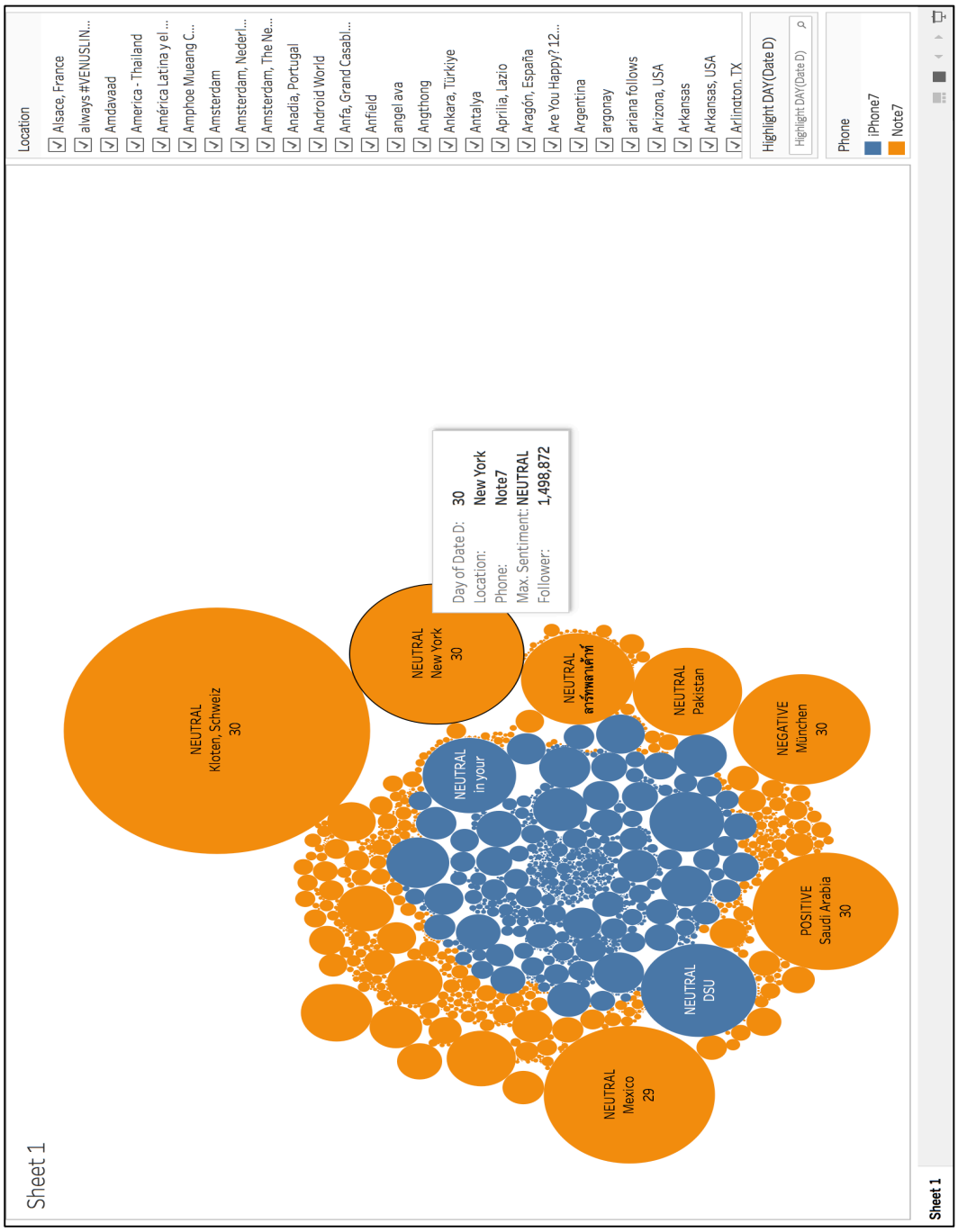


Figure 13:Density, Sentiment and Location Graph for Iphone vs Samsung (New York, USA)

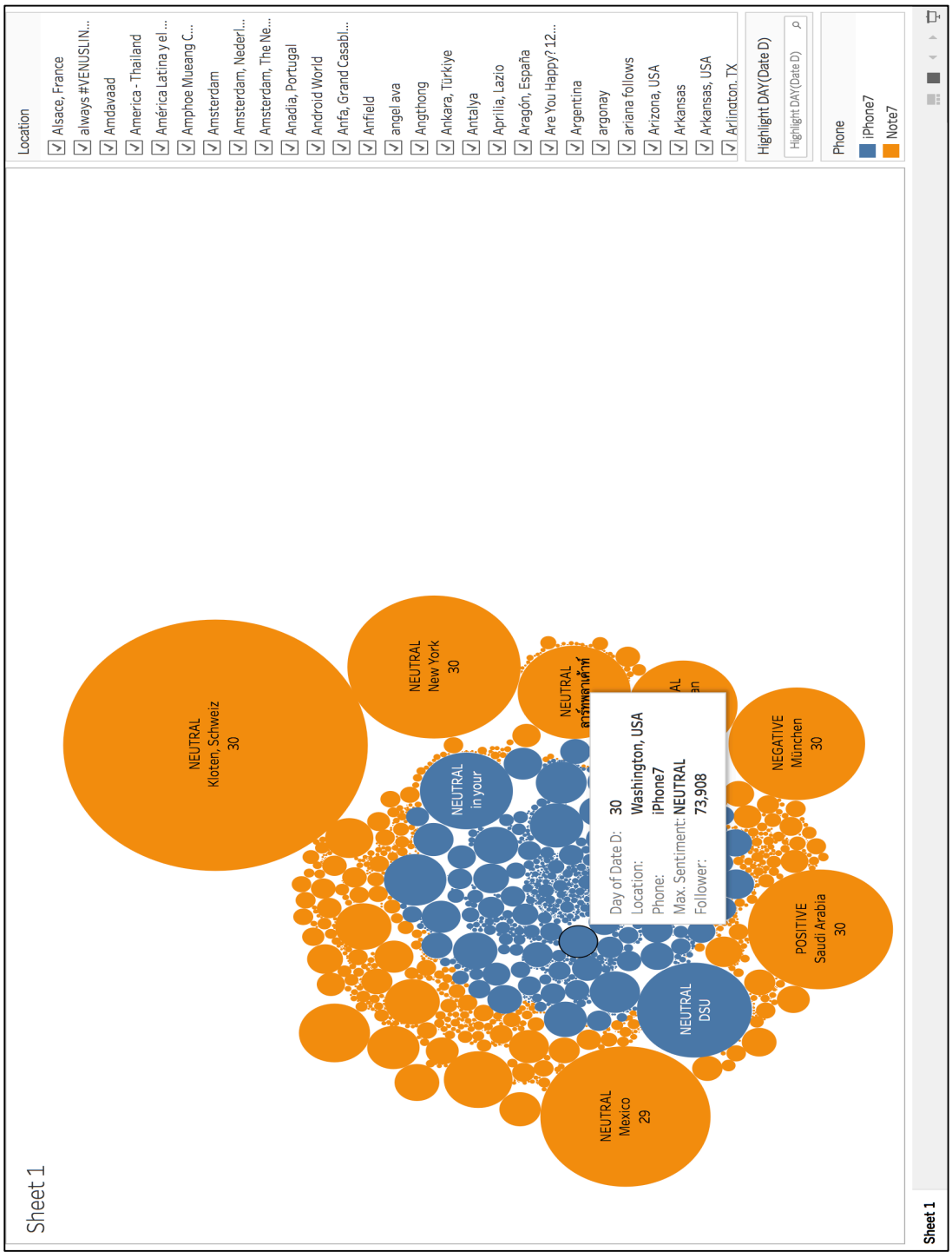


Figure 14: Density, Sentiment and Location Graph for Iphone vs Samsung (Washington, USA)

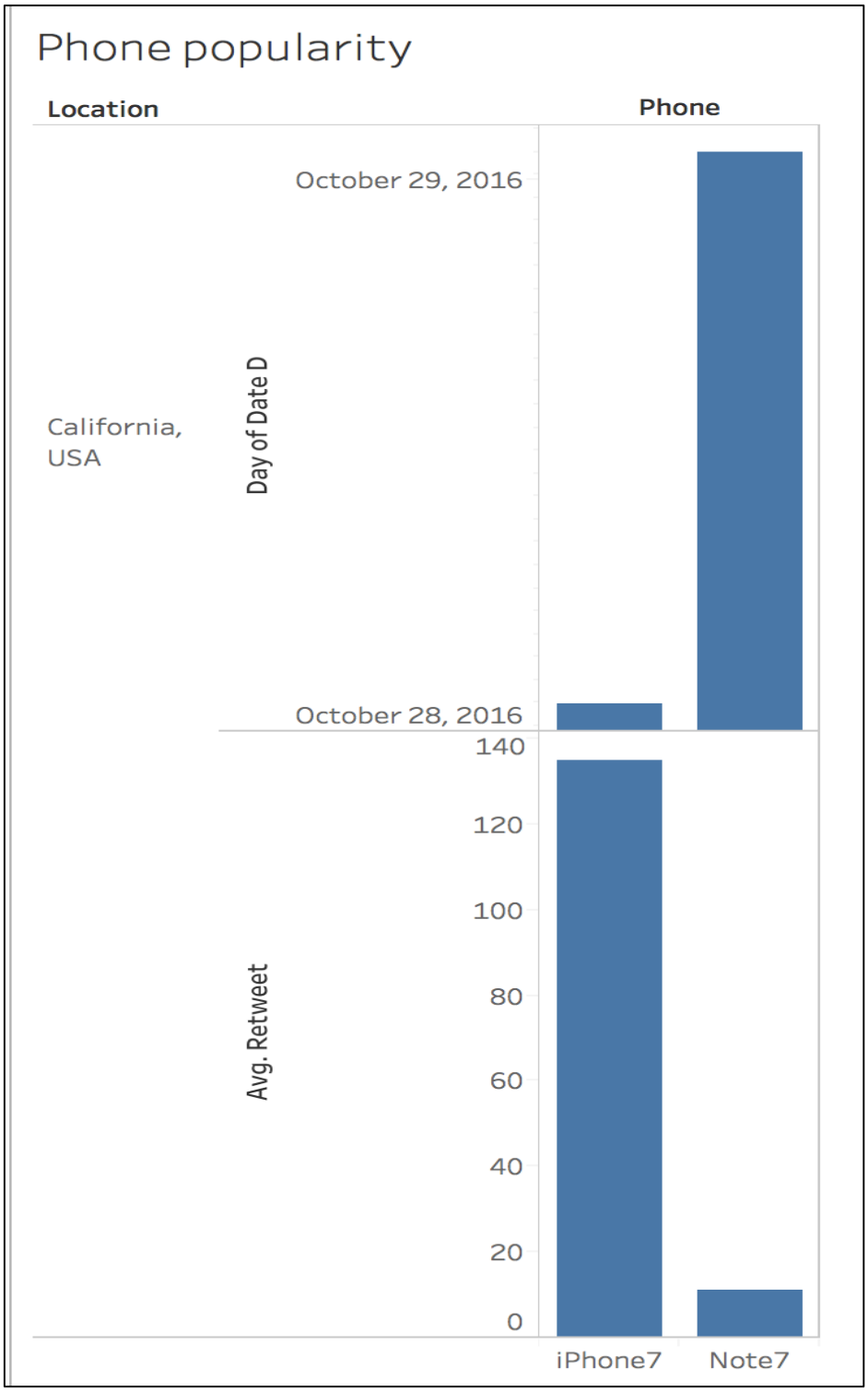


Figure 15: Popularity Graph Iphone v/s Samsung Note7

Figure 16: Location Based Sentimental Analysis

6.2 PlayStation vs Xbox vs Nintendo

Other confusing task found amongst young generations is game and console selections. There are many game stations available in today's world but the Sony PlayStation, Windows Xbox and Nintendo are the market leaders. All of them have their own follower base and come with their own specialties. To solve this dilemma, I picked up this use case for getting heat map based on location and sentiment.

The total size of dataset collected for this use case is 200000 plus.

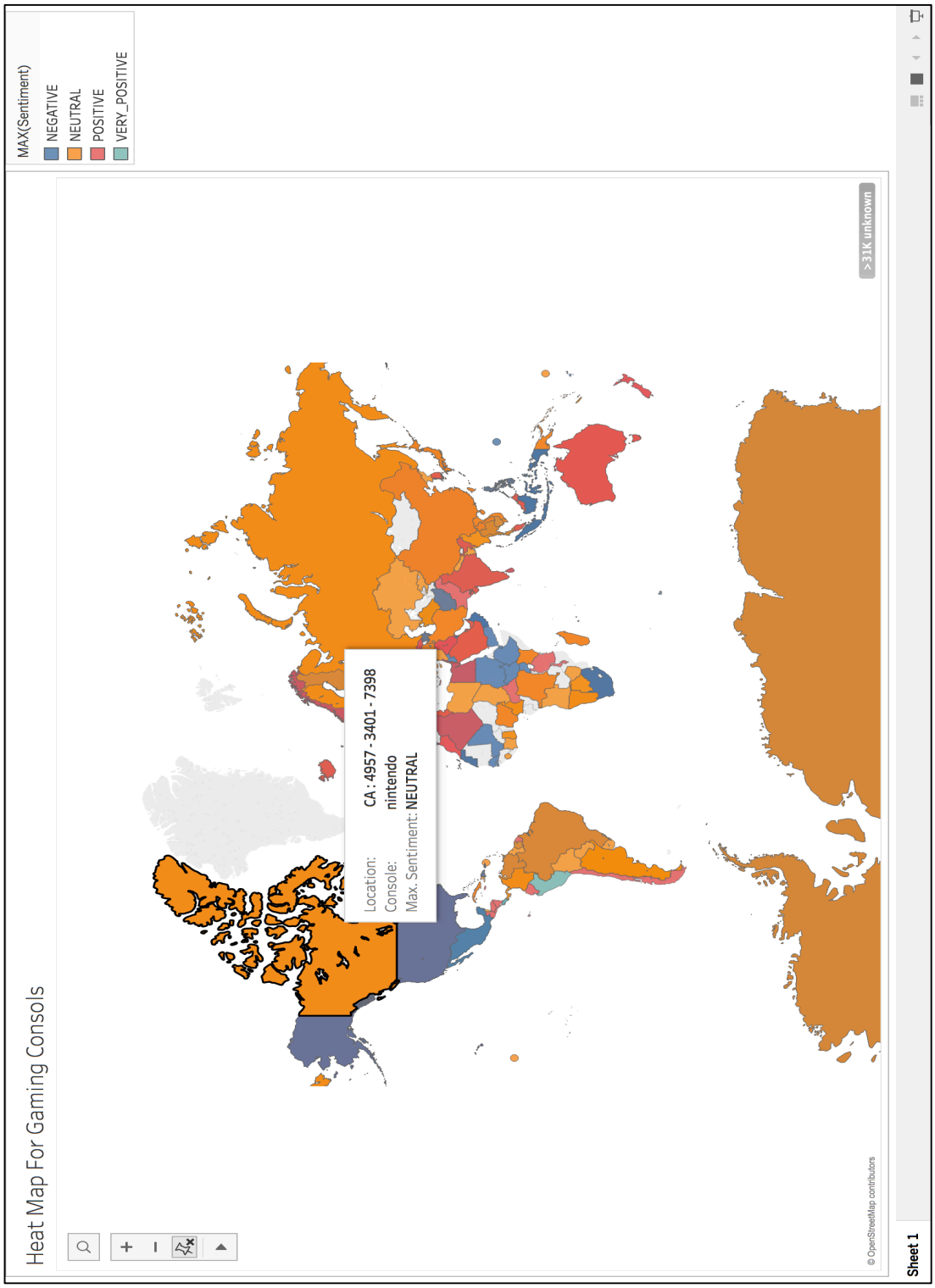


Figure 17: Heat Map for Gaming Consoles (Canada)

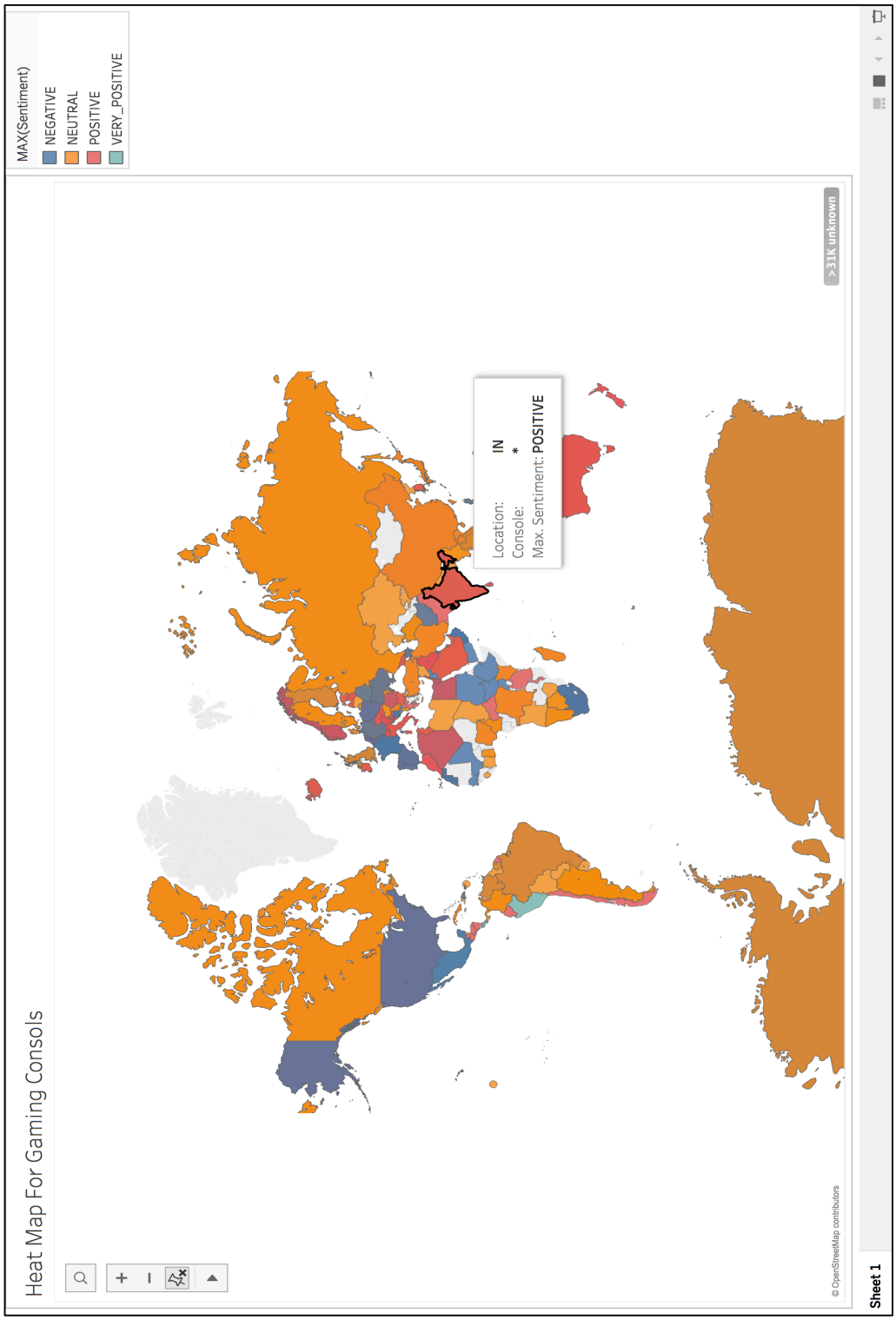


Figure 18: Heat Map for Gaming Consoles (India)

Chapter 7

FUTURE WORK

The world is currently moving towards mobile computing. Mobile applications have become an important factor for a products success. It's also very convenient to access mobile applications on the Go. In future a mobile application could be developed which would give instant reviews as soon as you set the filers as you desire. The task for mobile computing would be optimization of the cluster on such a small device with minimal hardware support and visualization.

Chapter 8

CONCLUSION

I have learnt a lot of technologies while developing this project. The problems I faced made me think deeper and out of the box. The vibrant and multidimensional visualization would help consumers select a better product and avail the benefits of real time review system based on location, public view and trend. It will also benefit businesses to get feedback based on it there would be a huge scope for development of their product or service.

During the course of implementation, I have gained in-depth knowledge in field of big data analytics. I got to learn concepts like cluster computing on Apache spark which is a very powerful platform and a trending platform for cluster computing, language like Scala which is object oriented as well as functional language provides a lot more flexibility for writing optimized code in smaller number of lines and has great capability with the spark cluster. Tools like SBT and Tableau which are used wide across the technological companies.

At last but not the least is I have learned how to overcome problems and develop a product that would help people in their lives. The experience has made me technically skilled and developed my thinking power around the problems.

BIBLIOGRAPHY

[1] Forbes. 12 Big Data Definitions. [Online]. Available: http://www.forbes.com/sites/gilpress/2014/09/03/12-big-data-definitions-whats-yours/#6490f23421a9. Accessed in September 2014.

[2] Wikipedia. Big Data. [Online]. Available: https://en.wikipedia.org/wiki/Big_data. Accessed in November 2013.

[3] Research Gate. The five V's of Big Data. [Online]. Available: https://www.researchgate.net/figure/281404634_fig1_Figure-1-The-five-V's-of-Big-Data-Adapted-from-IBM-big-data-platform-Bringing-big. Accessed in September 2015.

[4] Slide Share. Big Data Characteristics. [Online]. Available: http://www.slideshare.net/venturehire/what-is-big-data-and-its-characteristics. Accessed in July 2013.

[5] Wikipedia. Twitter. [Online]. Available: https://en.wikipedia.org/wiki/Twitter. Accessed in November 2012.

[6] Spark. Spark Overview. [Online]. Available: http://spark.apache.org/docs/2.0.1/. Accessed in October 2013.

[7] Spark. Cluster Mode Overview. [Online]. Available: http://spark.apache.org/docs/latest/cluster-overview.html. Accessed in October 2013.

[8] Spark. Spark Streaming Programming Guide. [Online]. Available: http://spark.apache.org/docs/2.0.1/streaming-programming-guide.html. Accessed in October 2013.

[9] Spark. Spark SQL, DataFrames and Datasets Guide. [Online]. Available: http://spark.apache.org/docs/2.0.1/sql-programming-guide.html. Accessed in October 2013.

[10] Spark. Spark SQL, DataFrames and Datasets Guide. [Online]. Available: http://spark.apache.org/docs/2.0.1/sql-programming-guide.html. Accessed in October 2013.

[11] Wikipedia. SBT. [Online]. Available: https://en.wikipedia.org/wiki/SBT_(software). Accessed in September 2013.

[12] Wikipedia. Tableau. [Online]. Available: https://en.wikipedia.org/wiki/Tableau_Software. Accessed in April 2013.

[13] Spring by Pivotal. Registering an application with Twitter. [Online]. Available: https://spring.io/guides/gs/register-twitter-app/. Accessed in May 2014.

[14] Big Sky. The Data Analysis Process. [Online]. Available: http://www.bigskyassociates.com/blog/bid/372186/The-Data-Analysis-Process-5-Steps-To-Better-Decision-Making. Accessed in March 2013.

[15] Stanford CoreNLP. A Suite of core NLP Tools. [Online]. Available: http://stanfordnlp.github.io/CoreNLP/. Accessed in August 2014.

[16] Recommender Systems. Data Mining. [Online]. Available: http://recommender-systems.readthedocs.io/en/latest/datamining.html. Accessed in March 2013.