INDUSTRIAL TRAINING REPORT – II

Done by

ROHITH.V
Register Number: RA1411003010245

At

National Informatics Centre- Open Technology Group.
Software Technology Parks of India, Taramani
Chennai- 90.

Submitted to

Department of Computer Science Engineering
Faculty of Engineering and Technology
SRM University (Under section 3 of UGC Act, 1956)
SRM Nagar, Kattankulathur – 603203
Kanchipuram District.

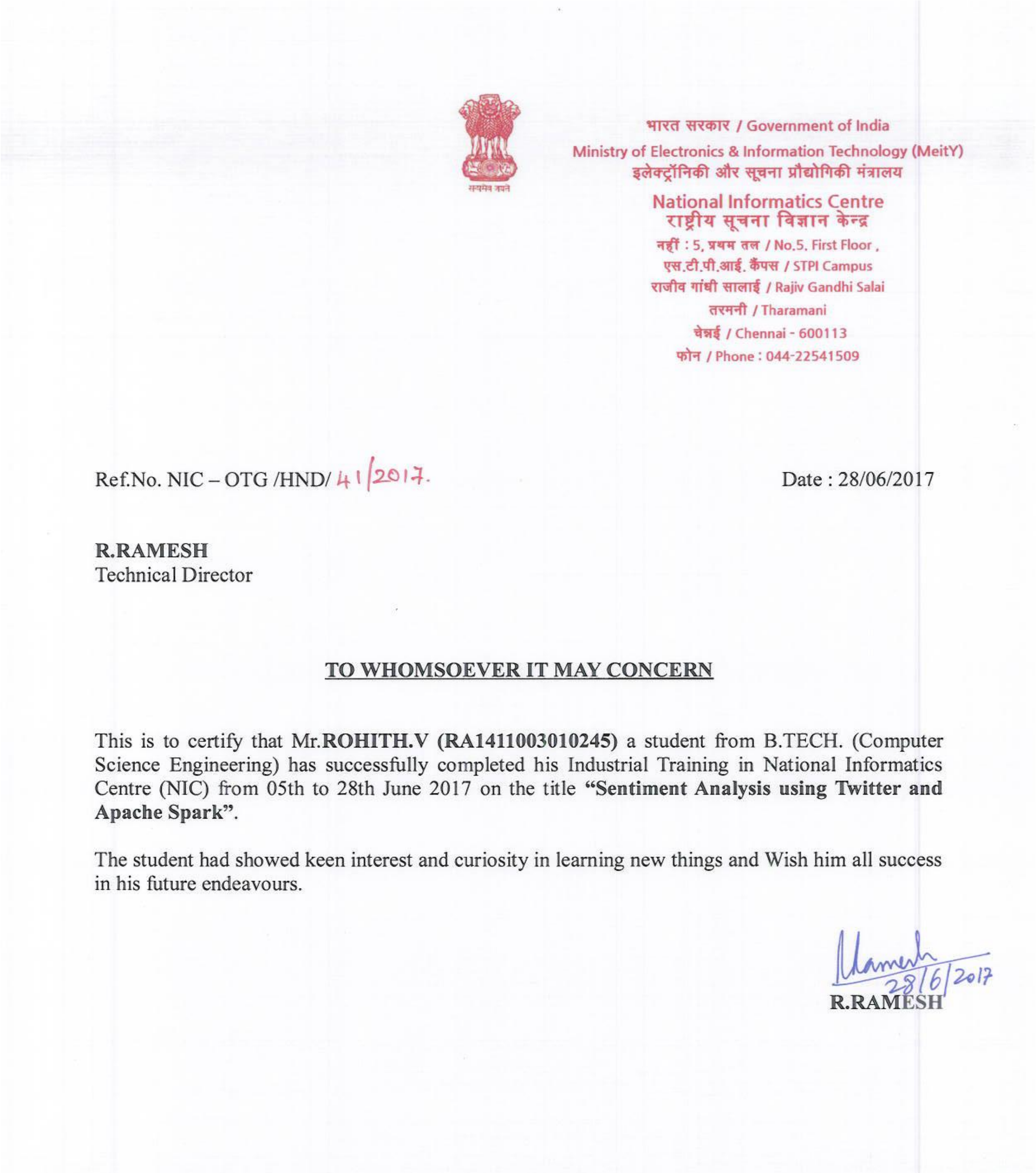
June 2017

BONAFIDE CERTIFICATE

Certified that this Industrial Training Report – II is a work of **ROHITH.V** (Reg.No.**RA1411003010245**) who carried out the work at **NATIONAL INFORMATICS CENTRE- OPEN TECHNOLOGY GROUP**, Software Technology Parks of India, Taramani, Chennai, Tamil Nadu.

Class In charge	Year-Coordinator
(Mr. C. RAJESH BABU)	(Mrs. PUSHPALATHA)

Certificate



सत्यमेव जयते

भारत सरकार / Government of India
Ministry of Electronics & Information Technology (MeitY)
इलेक्ट्रॉनिकी और सूचना प्रौद्योगिकी मंत्रालय

National Informatics Centre
राष्ट्रीय सूचना विज्ञान केन्द्र
नहीं : 5, प्रथम तल / No.5. First Floor ,
एस.टी.पी.आई. कैंपस / STPI Campus
राजीव गांधी सालाई / Rajiv Gandhi Salai
तरमनी / Tharamani
चेन्नई / Chennai - 600113
फोन / Phone : 044-22541509

Ref.No. NIC – OTG /HND/ 41/2017.

Date : 28/06/2017

R.RAMESH
Technical Director

TO WHOMSOEVER IT MAY CONCERN

This is to certify that Mr.**ROHITH.V (RA1411003010245)** a student from B.TECH. (Computer Science Engineering) has successfully completed his Industrial Training in National Informatics Centre (NIC) from 05th to 28th June 2017 on the title **"Sentiment Analysis using Twitter and Apache Spark"**.

The student had showed keen interest and curiosity in learning new things and Wish him all success in his future endeavours.

Ramesh
28/6/2017
R.RAMESH

Acknowledgement

It is always a pleasure to remind the fine people in the Engineering program for their sincere guidance I received to uphold my practical as well as theoretical skills in engineering.

Firstly I would like to thank Dr. C.Muthamizhchelvan (Director of E&T, SRM University) for meticulously planning academic curriculum in such a way that students are not only academically sound but also industry ready by including such industrial training patterns.

I would also like to thanks Mr. C.Rajesh Babu (Class In-charge) for the positive attitude he showed for my work, always allowing me to question him and giving prompt replies for my uncertainties in all the fields including educational, social and managerial work.

I express my immense pleasure and deep sense of gratitude to Mr. R.Ramesh (Technical Director) and to my guide Ms.Agalya (Developer) for spending her valuable time with me and also helped me in completion of task.

Table of Contents

List of Figures

Company Profile:

The **National Informatics Centre (NIC)** is the premier science & technology organisation of India's Union Government in informatics services and information-and-communication- technology (ICT) applications. The NIC is a part of the Indian Ministry of Communications and Information Technology's Department of Electronics & Information Technology.

It has played a pivotal role in steering e-governance applications in the governmental departments at national, state and district levels, enabling the improvement in, and a wider transparency of, government services. Almost all Indian-government websites are developed / managed by NIC.

Field of work

NIC offers telecommunications-networking services including K_u band (TDMA, FDMA, SCPC & satellite broadband) VSATs, wireless metropolitan-area networks (MANs) and local-area networks (LANs) with gateways for Internet- and Intranet-resource sharing.

It is the network infrastructure and e-governance support to India's central government and state governments, union-territory administrations, administrative divisions and other government bodies. The NIC assists in implementing information-technology projects, in collaboration with central and state governments, in the areas of: Communication & Information Technology.

Organisation

NIC is a part of the Indian Ministry of Communications & Information Technology's Department of Electronics & Information Technology and is headquartered in New Delhi. It has offices in all 29 state capitals and 7 union-territory headquarters and almost all districts. At New Delhi Headquarters, Mean Head a large number of Application Divisions exist which provide total Informatics Support to the Ministries and Departments of the Central Government. To cater to the ICT needs at the grassroots level, the NIC has also opened offices in almost all district collectorates. NIC Extends Technical Coordination and IT support to District Administration.

Services

The NIC offers a host of services including:

- Computer-aided design (CAD)
- Geographical-Information system (GIS)
- Domain-Name Registration for gov.in and nic.in domain
- Informatics
  - Biomedical informatics
  - Patent informatics
  - Rural informatics
  - Agriculture informatics, including hydrography
- Internet Data Centre (IDC)
- Mathematical Modelling and Simulation
- Computer networking
- office-procedure automation (OPA)
- Training
- Cyber Security
- Videoconferencing
- Website hosting & website development
- Internet Services
- Wi-Fi

Open Technology Group

NIC has established Open Technology Group (OTG) to spearhead the technology exploration and provision support services for adoption of OSS in various e-Governance Projects and applications under NIC and NeGP program of DeitY. NIC-OTG is mandated to facilitate strategic control of Open Technology within NIC and spearhead the knowledge centric activities in e-Governance Projects all over India.

The Open Technology Group (OTG), National Informatics Centre (NIC) has implemented the Open Technology Centre Project (OTC) (A Grant-in-Aid Project) of Department of Electronics & (DeitY), Government of India (GoI) during the period (April 2007- March 2015).

Key Technology Services supported by OTG

1. CMS/Portal using Drupal
2. Open Web Platform
3.Database Replication using Symmetric DS
4. Database Migration to PostgreSQL
5. Single Sign on Solution using CAS
6. Verification Services based on 2D Barcode
7. Platform independent Digital Signature Certificate
8. Recommendation of Open Source Stack after due evaluation
9. Bundled OSS Stack for Development, Staging & Deployment
10. Capacity Building & Hand holding on Recommended Open Source Stack
11. Provisioning Support for recommended stack-List of empanelled Vendors-Quick Reference
12.Performance Tuning of Open Source Application Servers

Also the Group participates in the Open Source related activities of NeST project of DeitY implemented by NICSI with STQC as technology partner.

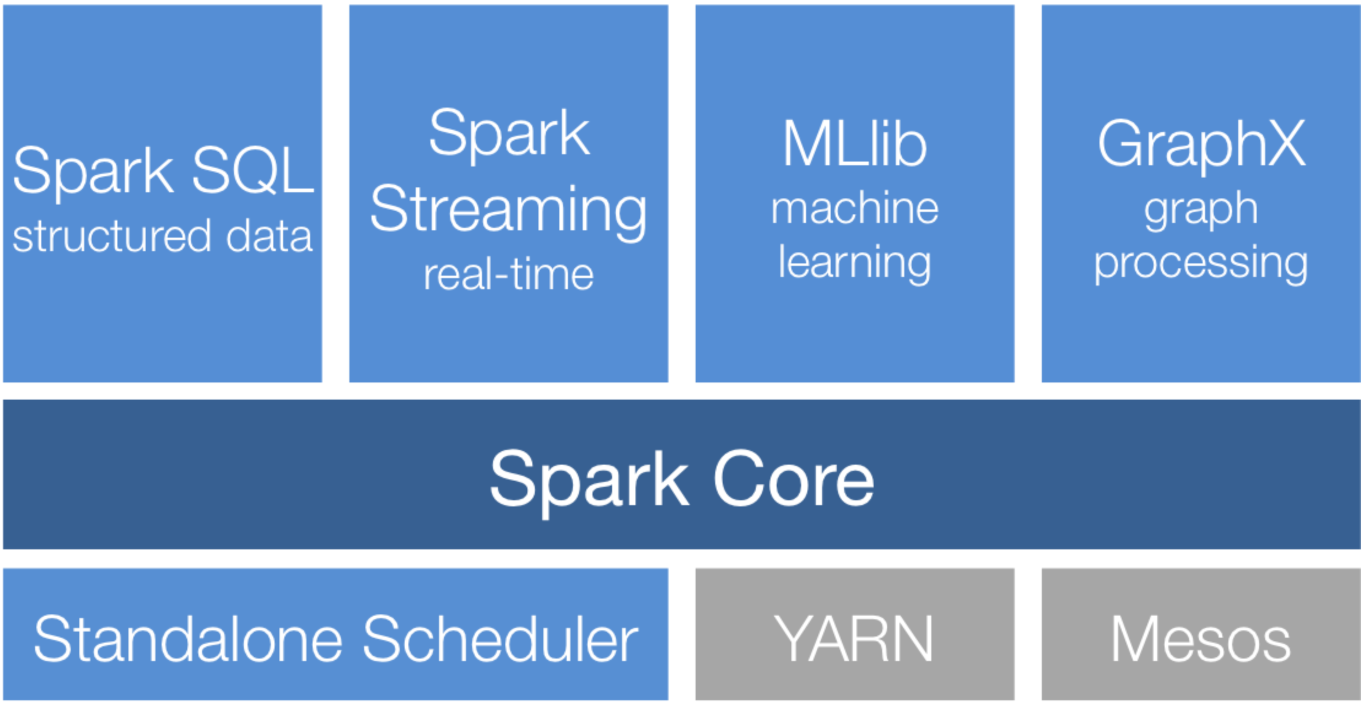
Platforms used:

1.	Apache Spark
2.	Apache Open NLP Classifier
3.	Eclipse IDE
4.	Scala
5.	Java
6.	Maven Repository
7.	Twitter API

APACHE SPARK:

Apache Spark is an open-source cluster-computing framework. Originally developed at the University of California, Berkeley's AMPLab, the Spark codebase was later donated to the Apache Software Foundation, which has maintained it since. Spark provides an interface for programming entire clusters with implicit data parallelism and fault-tolerance.

Overview



Apache Spark provides programmers with an application programming interface centered on a data structure called the resilient distributed dataset (RDD), a read-only multiset of data items distributed over a cluster of machines, that is maintained in a fault-tolerant way. It was developed in response to limitations in the MapReduce cluster computing paradigm, which forces a particular linear dataflow structure on distributed programs: MapReduce programs read input data from disk, map a function across the data, reduce the results of the map, and store reduction results on disk. Spark's RDDs function as a working set for distributed programs that offers a (deliberately) restricted form of distributed shared memory.

The availability of RDDs facilitates the implementation of both iterative algorithms, that visit their dataset multiple times in a loop, and interactive/exploratory data analysis, i.e., the repeated database-style querying of data. The latency of such applications (compared to a MapReduce implementation, as was common in Apache Hadoop stacks) may be reduced by several orders of magnitude. Among the class of iterative algorithms are the training algorithms for machine learning systems, which formed the initial impetus for developing Apache Spark.

Apache Spark requires a cluster manager and a distributed storage system. For cluster management, Spark supports standalone (native Spark cluster), Hadoop YARN, or Apache Mesos. For distributed storage, Spark can interface with a wide variety, including Hadoop Distributed File System (HDFS), MapR File System (MapR-FS), Cassandra, OpenStack Swift, Amazon S3, Kudu, or a custom solution can be implemented. Spark also supports a pseudo-distributed local mode, usually used only for development or testing purposes, where distributed storage is not required and the local file system can be used instead; in such a scenario, Spark is run on a single machine with one executor per CPU core.

Spark Core

Spark Core is the foundation of the overall project. It provides distributed task dispatching, scheduling, and basic I/O functionalities, exposed through an application programming interface (for Java, Python, Scala, and R) centered on the RDD abstraction). This interface mirrors a functional/higher-order model of programming: a "driver" program invokes parallel operations such as map, filter or reduce on an RDD by passing a function to Spark, which then schedules the function's execution in parallel on the cluster. These operations, and additional ones such as joins, take RDDs as input and produce new RDDs. RDDs are immutable and their operations are lazy; fault-tolerance is achieved by keeping track of the "lineage" of each RDD (the sequence of operations that produced it) so that it can be reconstructed in the case of data loss. RDDs can contain any type of Python, Java, or Scala objects.

Spark SQL

It is a component on top of Spark Core that introduced a data abstraction called DataFrames, which provides support for structured and semi-structured data. Spark SQL provides a domain-specific language (DSL) to manipulate DataFrames in Scala, Java, or Python. It also provides SQL language support, with command-line interfaces and ODBC/JDBC server. Although DataFrames lack the compile-time type-checking afforded by RDDs, as of Spark 2.0, the strongly typed DataSet is fully supported by Spark SQL as well.

Spark Streaming

Spark Streaming leverages Spark Core's fast scheduling capability to perform streaming analytics. It ingests data in mini-batches and performs RDD transformations on those mini-batches of data. This design enables the same set of application code written for batch analytics to be used in streaming analytics, thus facilitating easy implementation of lambda architecture. However, this convenience comes with the penalty of latency equal to the mini-batch duration. Other streaming data engines that process event by event rather than in minibatches include Storm and the streaming component of Flink. Spark Streaming has support built-in to consume from Kafka, Flume, Twitter, ZeroMQ, Kinesis, and TCP/IP sockets.

MLlib Machine Learning Library

Spark MLlib is a distributed machine learning framework on top of Spark Core that, due in large part to the distributed memory-based Spark architecture, is as much as nine times as fast as the disk-based implementation used by Apache Mahout (according to benchmarks done by the MLlib developers against the Alternating Least Squares (ALS) implementations, and before Mahout itself gained a Spark interface), and scales better than Vowpal Wabbit. Many common machine learning and statistical algorithms have been implemented and are shipped with MLlib which simplifies large scale machine learning pipelines, including:

- summary statistics, correlations, stratified sampling, hypothesis testing, random data generation
- classification and regression: support vector machines, logistic regression, linear regression, decision trees, naive Bayes classification
- collaborative filtering techniques including alternating least squares (ALS)
- cluster analysis methods including k-means, and Latent Dirichlet Allocation (LDA)
- dimensionality reduction techniques such as singular value decomposition (SVD), and principal component analysis (PCA)
- feature extraction and transformation functions optimization algorithms such as stochastic gradient descent, limited-memory BFGS (L-BFGS)

GraphX

GraphX is a distributed graph processing framework on top of Apache Spark. Because it is based on RDDs, which are immutable, graphs are immutable and thus GraphX is unsuitable for graphs that need to be updated, let alone in a transactional manner like a graph database. GraphX provides two separate APIs for implementation of massively parallel algorithms (such as PageRank): a Pregel abstraction, and a more general MapReduce style API. Unlike its predecessor Bagel, which was formally deprecated in Spark 1.6, GraphX has full support for property graphs (graphs where properties can be attached to edges and vertices). GraphX can be viewed as being the Spark in-memory version of Apache Giraph, which utilized Hadoop disk based MapReduce. Like Apache Spark, GraphX initially started as a research project at UC Berkeley's AMPLab and Databricks, and was later donated to the Apache Software Foundation and the Spark project.

APACHE OPEN NLP:

The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and co-reference resolution. These tasks are usually required to build more advanced text processing services. OpenNLP also included maximum entropy and perceptron based machine learning.

The goal of the OpenNLP project will be to create a mature toolkit for the abovementioned tasks. An additional goal is to provide a large number of prebuilt models for a variety of languages, as well as the annotated text resources that those models are derived from.

General Library Structure

The Apache OpenNLP library contains several components, enabling one to build a full natural language processing pipeline. These components include: sentence detector, tokenizer, name finder, document categorizer, part-of-speech tagger, chunker, parser, co reference resolution. Components contain parts which enable one to execute the respective natural language processing task, to train a model and often also to evaluate a model. Each of these facilities is accessible via its application program interface (API). In addition, a command line interface (CLI) is provided for convenience of experiments and training.

ECLIPSE IDE:

Eclipse is an integrated development environment (IDE) used in computer programming, and is the most widely used Java IDE. It contains a base workspace and an extensible plug-in system for customizing the environment. Eclipse is written mostly in Java and its primary use is for developing Java applications, but it may also be used to develop applications in other programming languages via plug-ins, including Ada, ABAP, C, C++, COBOL, D, Fortran, Haskell, JavaScript, Julia, Lasso, Lua, NATURAL, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Rust, Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop documents with LaTeX (via a TeXlipse plug-in) and packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++, and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM VisualAge. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules. Since Equinox, plug-ins can be plugged-stopped dynamically and are termed (OSGI) bundles.

Eclipse software development kit (SDK) is free and open source software, released under the terms of the Eclipse Public License, although it is incompatible with the GNU General Public License. It was one of the first IDEs to run under GNU Classpath and it runs without problems under IcedTea.

SCALA:

Scala is a general-purpose programming language providing support for functional programming and a strong static type system. Designed to be concise, many of Scala's design decisions aimed to address criticisms of Java.

Scala source code is intended to be compiled to Java bytecode, so that the resulting executable code runs on a Java virtual machine. Scala provides language interoperability with Java, so that libraries written in both languages may be referenced directly in Scala or Java code. Like Java, Scala is object-oriented, and uses a curly-brace syntax reminiscent of the C programming language. Unlike Java, Scala has many features of functional programming languages like Scheme, Standard ML and Haskell, including currying, type inference, immutability, lazy evaluation, and pattern matching. It also has an advanced type system supporting algebraic data types, covariance and contravariance, higherorder types (but not higher-rank types), and anonymous types. Other features of Scala not present in Java include operator overloading, optional parameters, named parameters, raw strings, and no checked exceptions.

The name Scala is a portmanteau of scalable and language, signifying that it is designed to grow with the demands of its users.

JAVA:

Java is a general-purpose computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA),meaning that compiled Java code can run on all platforms that support Java without the need for recompilation.[16] Java applications are typically compiled to bytecode that can run on any Java virtual machine (JVM) regardless of computer architecture. As of 2016, Java is one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since been acquired by Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

TWITTER :

Twitter is an online news and social networking service where users post and interact with messages, "tweets", restricted to 140 characters. Registered users can post tweets, but those who are unregistered can only read them. Users access Twitter through its website interface, SMS or a mobile device app. Twitter Inc. is based in San Francisco, California, United States, and has more than 25 offices around the world.

Developers

Twitter is recognized for having one of the most open and powerful developer APIs of any major technology company. Developer interest in Twitter began immediately following its launch, prompting the company to release the first version of its public API in September 2006. The API quickly became iconic as a reference implementation for public REST APIs and is widely cited in programming tutorials.

From 2006 until 2010, Twitter's developer platform experienced strong growth and a highly favorable reputation. Developers built upon the public API to create the first Twitter mobile phone clients as well as the first URL shortener. Between 2010 and 2012, however, Twitter made a number of decisions that were received unfavorably by the developer community. In 2010, Twitter mandated that all developers adopt OAuth authentication with just 9 weeks of notice. Later that year, Twitter launched its own URL shortener, in direct competition with some of its most well-known 3rd-party developers. And in 2012, Twitter introduced strict usage limits for its API, "completely crippling" some developers. While these moves successfully increased the stability and security of the service, they were broadly perceived as hostile to developers, causing them to lose trust in the platform.

In an effort to reset its relationship with developers, Twitter acquired Crashlytics on January 28, 2013 for over USD $100 million, its largest acquisition to date. Founded by Jeff Seibert and Wayne Chang, Crashlytics had rapidly gained popularity as a tool to help mobile developers identify and fix bugs in their apps. Twitter committed to continue supporting and expanding the service.

In an effort to reset its relationship with developers, Twitter acquired Crashlytics on January 28, 2013 for over USD $100 million, its largest acquisition to date. Founded by Jeff Seibert and Wayne Chang, Crashlytics had rapidly gained popularity as a tool to help mobile developers identify and fix bugs in their apps. Twitter committed to continue supporting and expanding the service.

In early 2016, Twitter announced that Fabric was installed on more than 2 billion active devices and used by more than 225,000 developers. Fabric is recognized as the #1 most popular crash reporting and also the #1 mobile analytics solution among the top 200 iOS apps, beating out Google Analytics, Flurry, and MixPanel.

NECESSITY:

Spark Streaming: Real time twitter sentiment analysis

Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. Spark Streaming can be used to stream live data and processing can happen in real time. Spark Streaming's ever-growing user base consists of household names like Uber, Netflix and Pinterest.

When it comes to Real Time Data Analytics, Spark Streaming provides a single platform to ingest data for fast and live processing in Apache Spark. Through this blog, I will introduce you to this new exciting domain of Spark Streaming and we will go through a complete use case, *Twitter Sentiment Analysis* using Spark Streaming.

Spark Streaming API can consume from sources like Kafka, Flume, and Twitter source to name a few. It can then apply transformations on the data to get the desired result which can be pushed further downstream.

What is Streaming?

Data Streaming is a technique for transferring data so that it can be processed as a steady and continuous stream. Streaming technologies are becoming increasingly important with the growth of the Internet.

Why Spark Streaming?

We can use Spark Streaming to stream real-time data from various sources like Twitter, Stock Market and Geographical Systems and perform powerful analytics to help businesses.

Spark Streaming Overview

Spark Streaming is used for processing real-time streaming data. It is a useful addition to the core Spark API. Spark Streaming enables high-throughput and fault-tolerant stream processing of live data streams.

The fundamental stream unit is DStream which is basically a series of RDDs to process the real-time data.

Spark Streaming Features

1. *Scaling:* Spark Streaming can easily scale to hundreds of nodes.
2. *Speed:* It achieves low latency.
3. *Fault Tolerance:* Spark has the ability to efficiently recover from failures.
4. *Integration:* Spark integrates with batch and real-time processing.
5. *Business Analysis:* Spark Streaming is used to track the behavior of customers which can be used in business analysis.

Spark Streaming Workflow

Spark Streaming workflow has four high-level stages. The first is to stream data from various sources. These sources can be streaming data sources like Akka, Kafka, Flume, AWS or Parquet for real-time streaming. The second type of sources includes HBase, MySQL, PostgreSQL, Elastic Search, Mongo DB and Cassandra for static/batch streaming. Once this happens, Spark can be used to perform Machine Learning on the data through its MLlib API. Further, Spark SQL is used to perform further operations on this data. Finally, the streaming output can be stored into various data storage systems like HBase, Cassandra, MemSQL, Kafka, Elastic Search, HDFS and local file system.

Spark Streaming Fundamentals

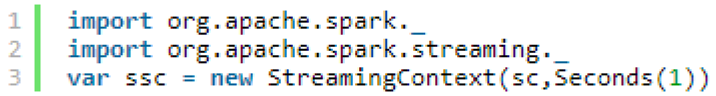
1. Streaming Context
2. DStream
3. Caching
4. Accumulators, Broadcast Variables and Checkpoints

Streaming Context

Streaming Context consumes a stream of data in Spark. It registers an *Input DStream* to produce a *Receiver* object. It is the main entry point for Spark functionality. Spark provides a number of default implementations of sources like Twitter, Akka Actor and ZeroMQ that are accessible from the context.

A StreamingContext object can be created from a SparkContext object. A SparkContext represents the connection to a Spark cluster and can be used to create RDDs, accumulators and broadcast variables on that cluster.

Example:



```
import org.apache.spark._
import org.apache.spark.streaming._
var ssc = new StreamingContext(sc,Seconds(1))
```

DStream

Discretized Stream (DStream) is the basic abstraction provided by Spark Streaming. It is a continuous stream of data. It is received from a data source or a processed data stream generated by transforming the input stream.

Internally, a DStream is represented by a continuous series of RDDs and each RDD contains data from a certain interval.

Input DStreams:

Input DStreams are DStreams representing the stream of input data received from streaming sources.

Every input DStream is associated with a Receiver object which receives the data from a source and stores it in Spark's memory for processing.

Transformations on DStreams:

Any operation applied on a DStream translates to operations on the underlying RDDs. Transformations allow the data from the input DStream to be modified similar to RDDs. DStreams support many of the transformations available on normal Spark RDDs.

The following are some of the popular transformations on DStreams:

map(*func*)	map(*func*) returns a new DStream by passing each element of the source DStream through a function *func*.
flatMap(*func*)	flatMap(*func*) is similar to map(*func*) but each input item can be mapped to 0 or more output items and returns a new DStream by passing each source element through a function *func*.
filter(*func*)	filter(*func*) returns a new DStream by selecting only the records of the source DStream on which *func* returns true.

reduce(*func*)	reduce(*func*) returns a new DStream of single-element RDDs by aggregating the elements in each RDD of the source DStream using a function *func*.
groupBy(*func*)	groupBy(*func*) returns the new RDD which basically is made up with a key and corresponding list of items of that group.

Output DStreams:

Output operations allow DStream's data to be pushed out to external systems like databases or file systems. Output operations trigger the actual execution of all the DStream transformations.

Caching

DStreams allow developers to cache/ persist the stream's data in memory. This is useful if the data in the DStream will be computed multiple times. This can be done using the *persist()* method on a DStream.

For input streams that receive data over the network (such as Kafka, Flume, Sockets, etc.), the default persistence level is set to replicate the data to two nodes for fault-tolerance.

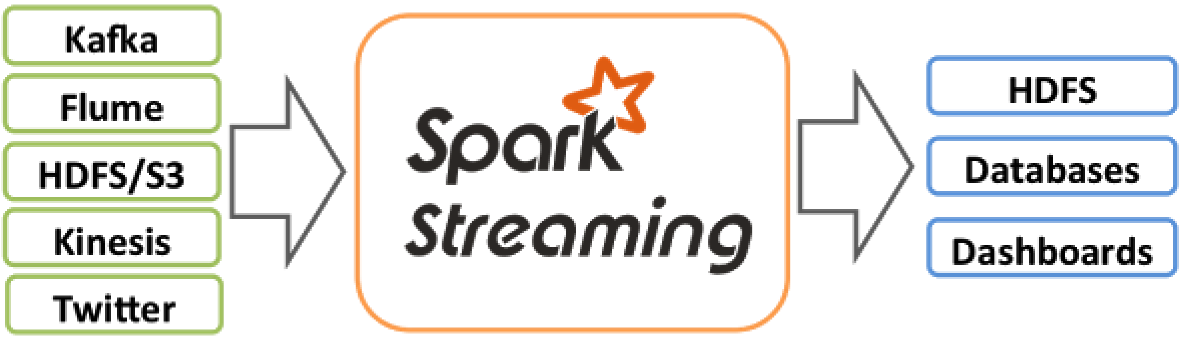
Problem Statement

To design a Twitter Sentiment Analysis System where we populate real-time sentiments for crisis management, service adjusting and target marketing.

Applications of Sentiment Analysis:

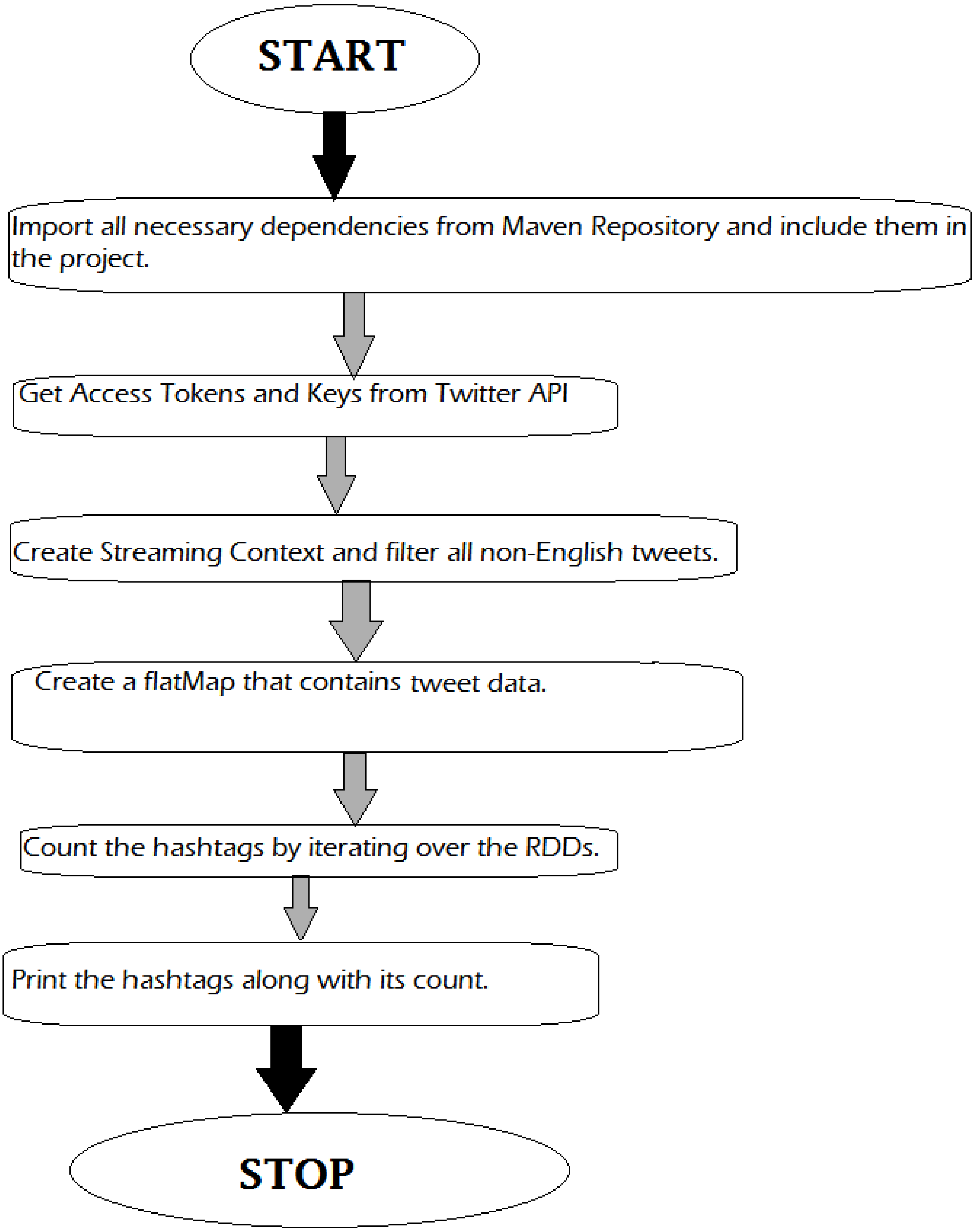
- Predict the success of a movie
- Predict political campaign success
- Decide whether to invest in a certain company
- Targeted advertising
- Review products and services

Architecture:



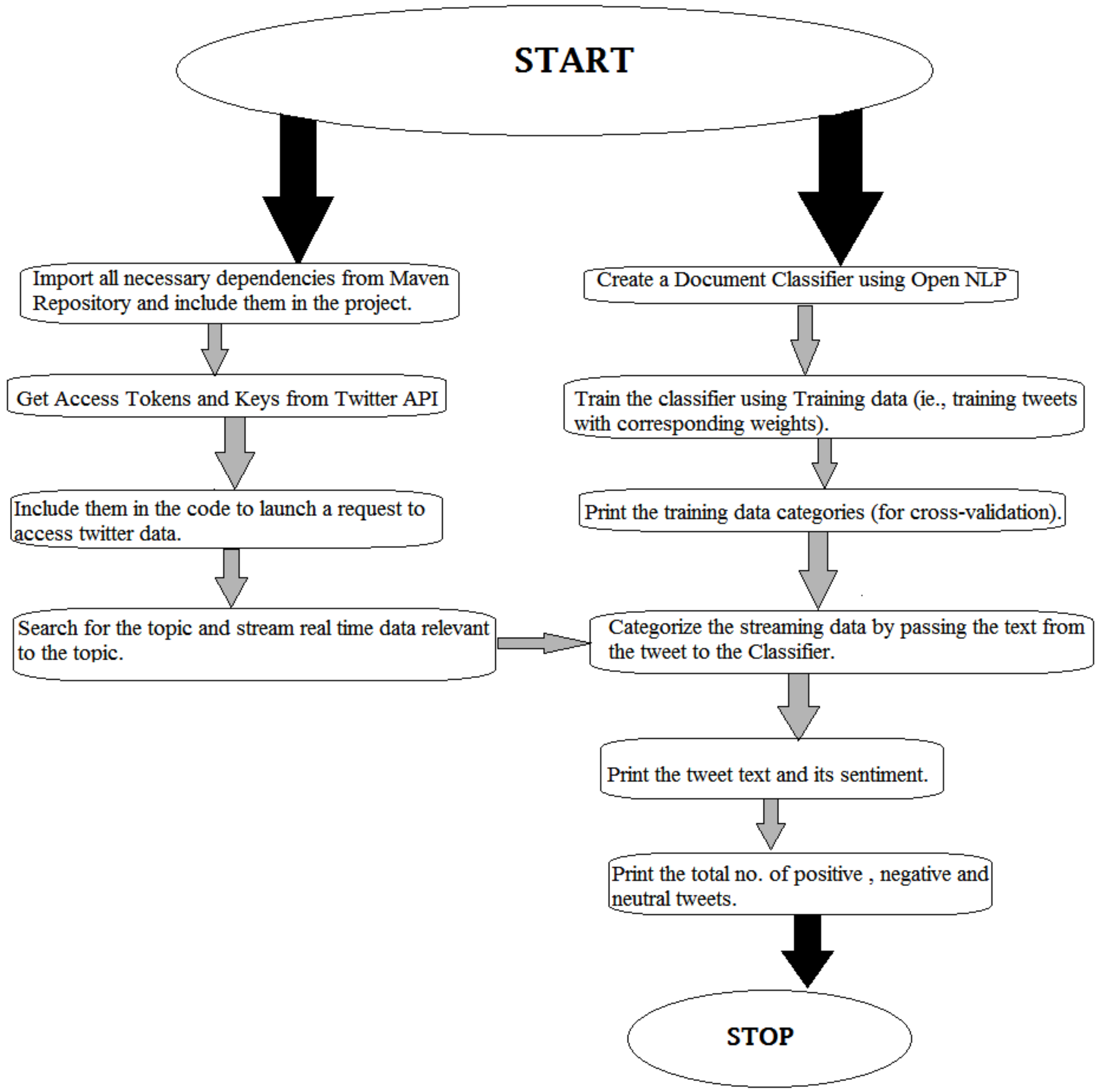
WORKFLOW:

Spark Popular HashTags



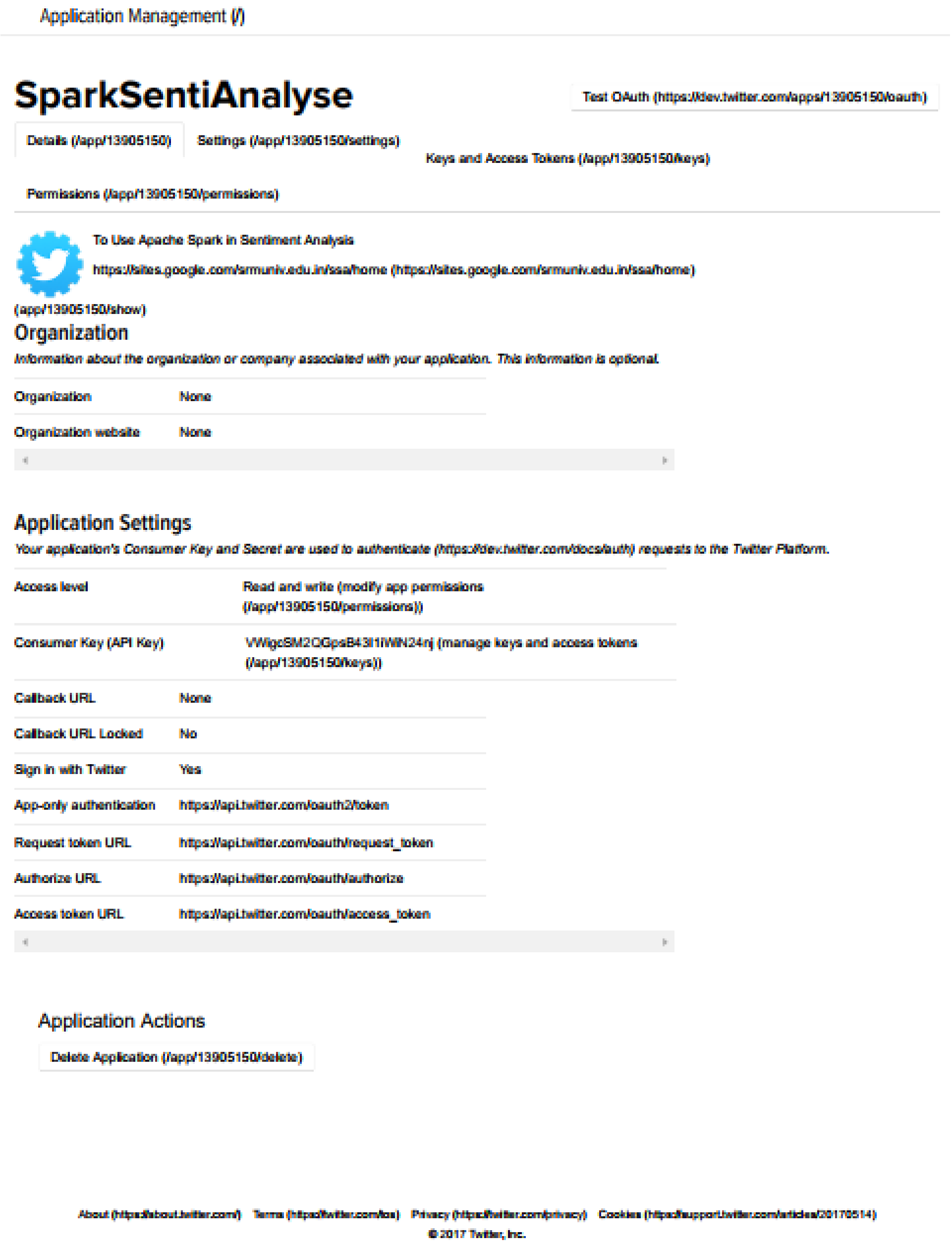
START
Import all necessary dependencies from Maven Repository and include them in the project.
Get Access Tokens and Keys from Twitter API
Create Streaming Context and filter all non-English tweets.
Create a flatMap that contains tweet data.
Count the hashtags by iterating over the RDDs.
Print the hashtags along with its count.
STOP

Sentiment Analysis with count:


START
Import all necessary dependencies from Maven Repository and include them in the project.
Get Access Tokens and Keys from Twitter API
Include them in the code to launch a request to access twitter data.
Search for the topic and stream real time data relevant to the topic.
Create a Document Classifier using Open NLP
Train the classifier using Training data (ie., training tweets with corresponding weights).
Print the training data categories (for cross-validation).
Categorize the streaming data by passing the text from the tweet to the Classifier.
Print the tweet text and its sentiment.
Print the total no. of positive , negative and neutral tweets.
STOP

Twitter API:

i. Log on to https://www.apps.twitter.com/ with Twitter Credentials.



Application Management (/)

SparkSentiAnalyse

Test OAuth (https://dev.twitter.com/apps/13905150/oauth)

Details (/app/13905150) Settings (/app/13905150/settings) Keys and Access Tokens (/app/13905150/keys) Permissions (/app/13905150/permissions)

To Use Apache Spark in Sentiment Analysis

https://sites.google.com/srmuniv.edu.in/ssa/home (https://sites.google.com/srmuniv.edu.in/ssa/home)

(app/13905150/show)

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 (https://dev.twitter.com/docs/auth) requests to the Twitter Platform.

Access level	Read and write (modify app permissions (/app/13905150/permissions))
Consumer Key (API Key)	VWigcSM2OGpsB43I1IWiN24nj (manage keys and access tokens (/app/13905150/keys))
Callback URL	None
Callback URL Locked	No
Sign in with Twitter	Yes
App-only authentication	https://api.twitter.com/oauth2/token
Request token URL	https://api.twitter.com/oauth/request_token
Authorize URL	https://api.twitter.com/oauth/authorize
Access token URL	https://api.twitter.com/oauth/access_token

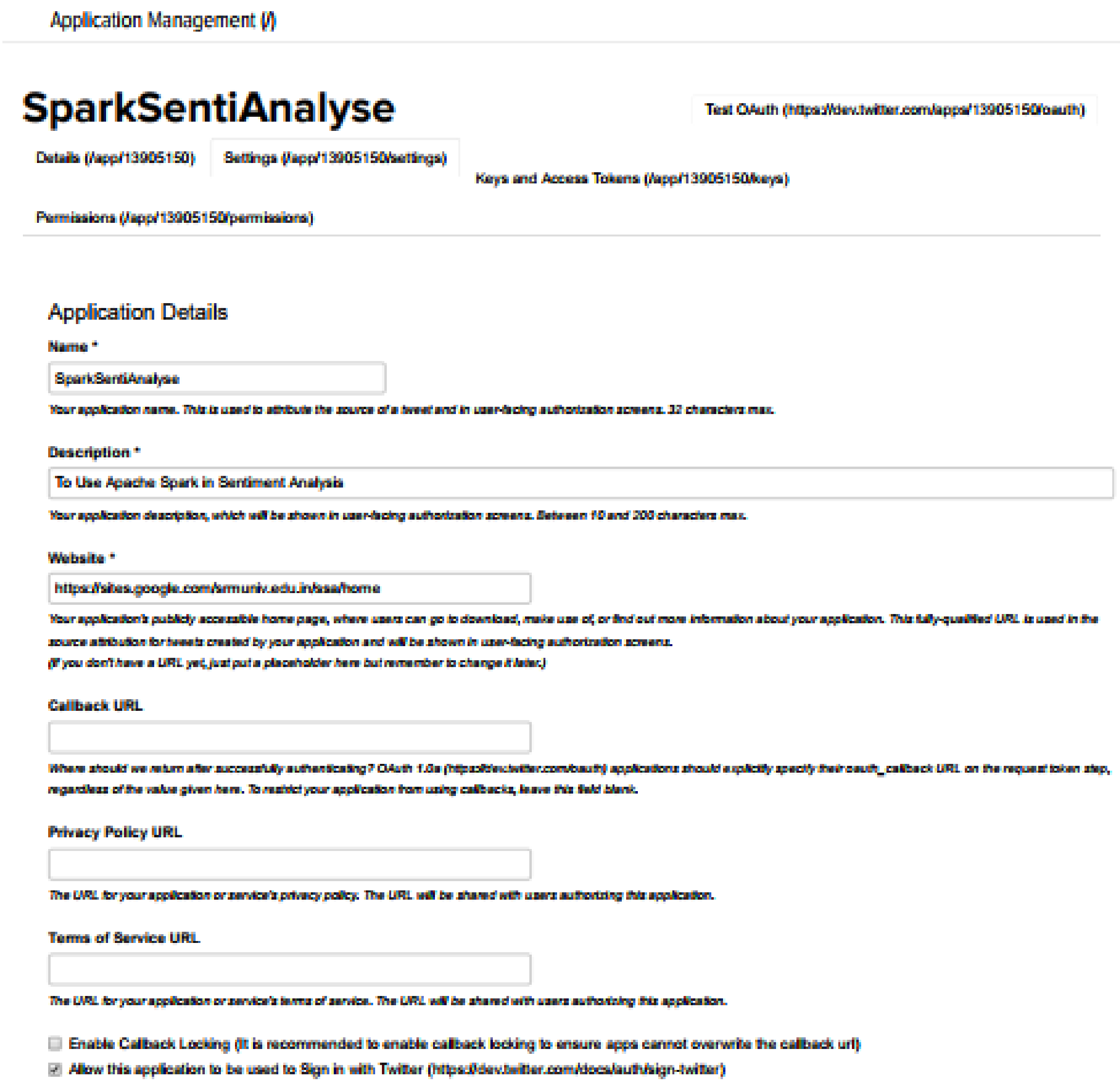
Application Actions

Delete Application (/app/13905150/delete)

About (https://about.twitter.com/) Terms (https://twitter.com/tos) Privacy (https://twitter.com/privacy) Cookies (https://support.twitter.com/articles/20170514)

© 2017 Twitter, Inc.

ii. Set the app name and the website URL.



Application Management (/)

SparkSentiAnalyse

Test OAuth (https://dev.twitter.com/apps/13905150/oauth)

Details (/app/13905150) Settings (/app/13905150/settings) Keys and Access Tokens (/app/13905150/keys)

Permissions (/app/13905150/permissions)

Application Details

Name *

SparkSentiAnalyse

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

Description *

To Use Apache Spark in Sentiment Analysis

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

Website *

https://sites.google.com/srmuniv.edu.in/ssa/home

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

Where should we return after successfully authenticating? OAuth 1.0a (https://dev.twitter.com/oauth) applications should explicitly specify their oauth_callback URL on the request token step, regardless of the value given here. To restrict your application from using callbacks, leave this field blank.

Privacy Policy URL

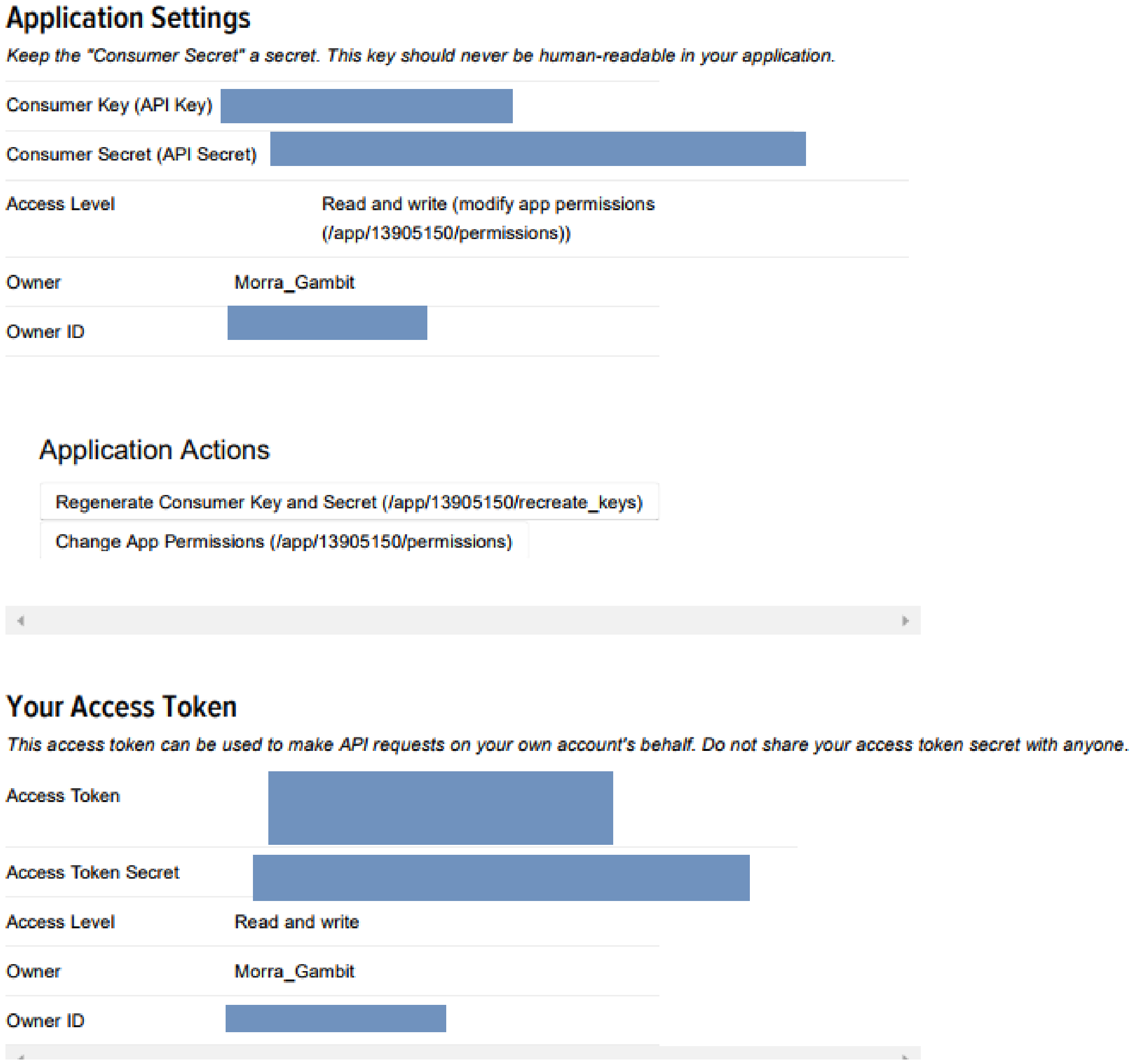
The URL for your application or service's privacy policy. The URL will be shared with users authorizing this application.

Terms of Service URL

The URL for your application or service's terms of service. The URL will be shared with users authorizing this application.

☐ Enable Callback Locking (It is recommended to enable callback locking to ensure apps cannot overwrite the callback url)
☑ Allow this application to be used to Sign in with Twitter (https://dev.twitter.com/docs/auth/sign-twitter)

iii. Take note of the Consumer Key, Consumer Token, Access Key and Access token in the Keys



Application Settings

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

Consumer Key (API Key)	[redacted]
Consumer Secret (API Secret)	[redacted]
Access Level	Read and write (modify app permissions (/app/13905150/permissions))
Owner	Morra_Gambit
Owner ID	[redacted]

Application Actions

Regenerate Consumer Key and Secret (/app/13905150/recreate_keys)

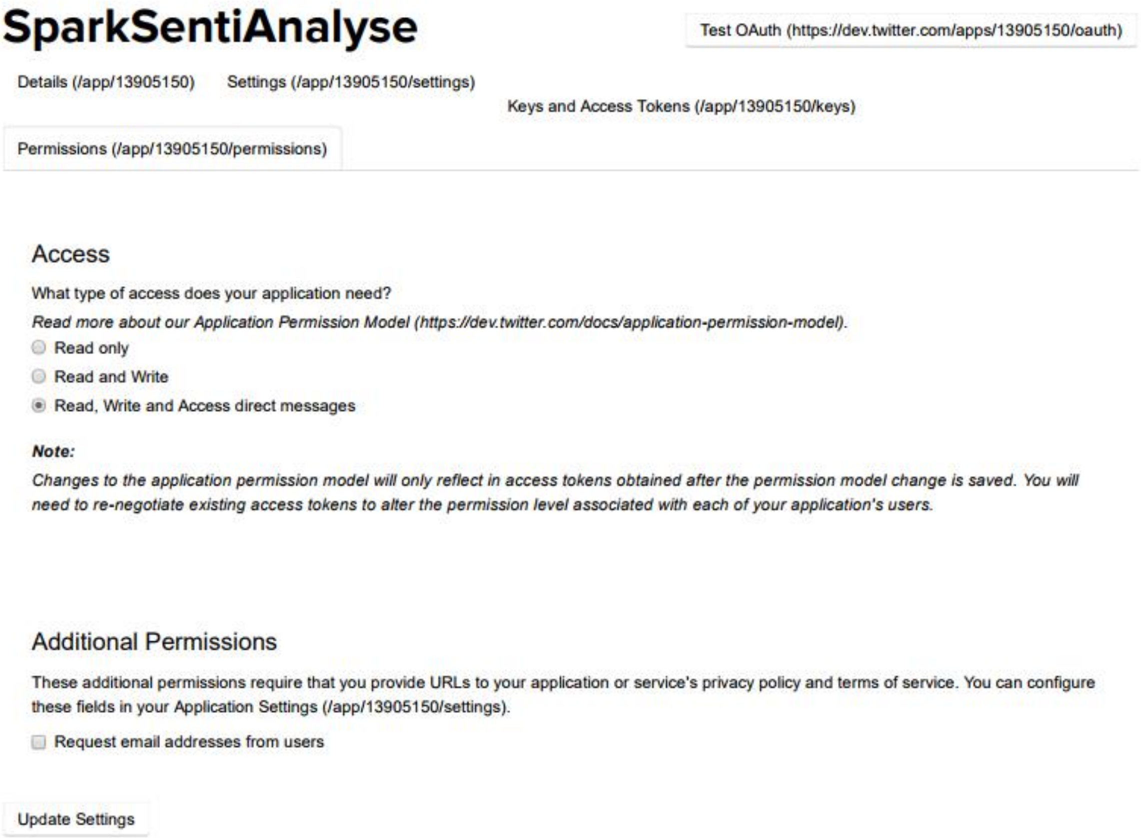
Change App Permissions (/app/13905150/permissions)

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	[redacted]
Access Token Secret	[redacted]
Access Level	Read and write
Owner	Morra_Gambit
Owner ID	[redacted]

iv. Set app permissions, if necessary.



SparkSentiAnalyse

Test OAuth (https://dev.twitter.com/apps/13905150/oauth)
Details (/app/13905150)
Settings (/app/13905150/settings)
Keys and Access Tokens (/app/13905150/keys)
Permissions (/app/13905150/permissions)

Access

What type of access does your application need?

Read more about our Application Permission Model (https://dev.twitter.com/docs/application-permission-model).

○ Read only

○ Read and Write

● Read, Write and Access direct messages

Note:

Changes to the application permission model will only reflect in access tokens obtained after the permission model change is saved. You will need to re-negotiate existing access tokens to alter the permission level associated with each of your application's users.

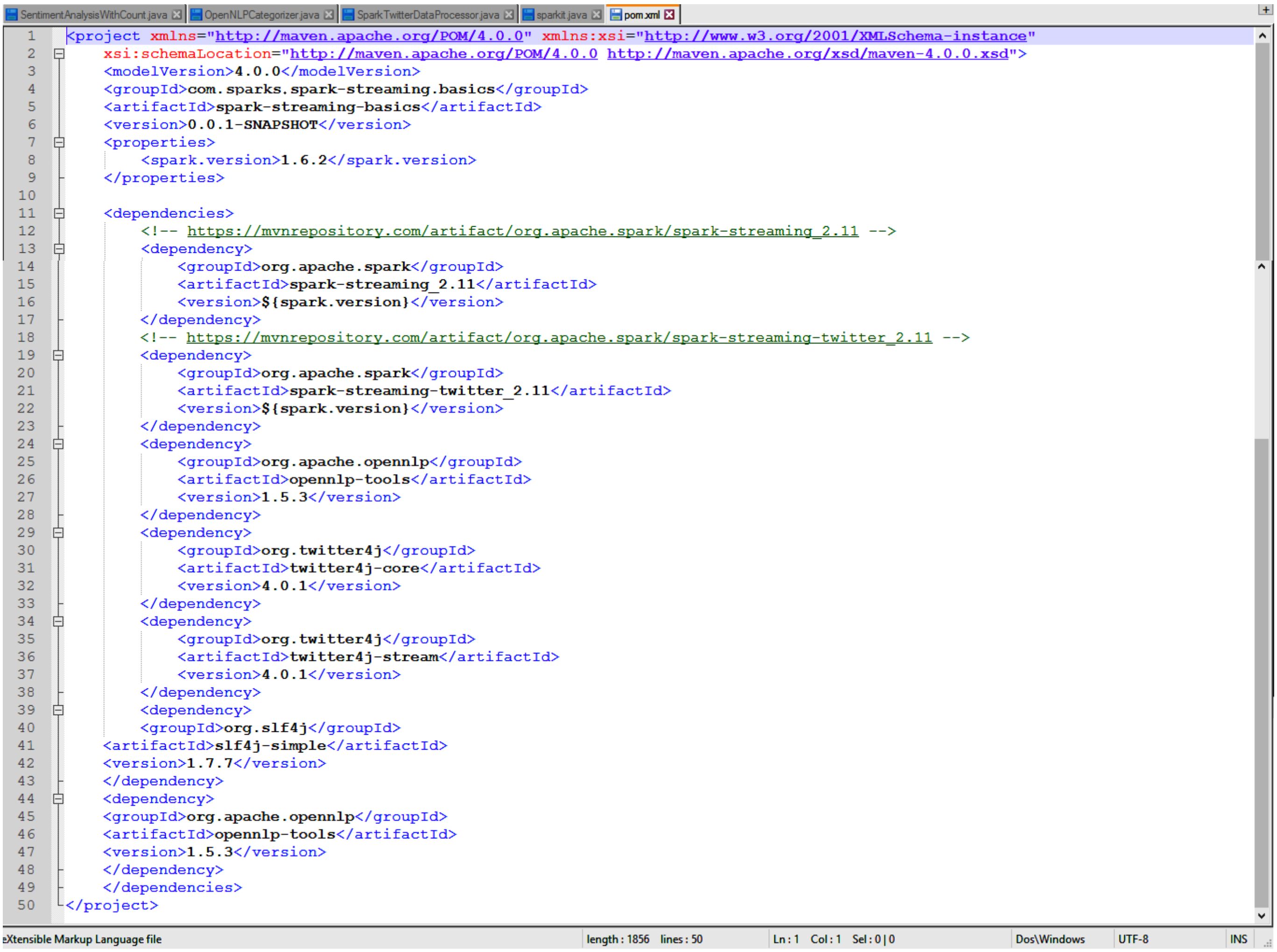
Additional Permissions

These additional permissions require that you provide URLs to your application or service's privacy policy and terms of service. You can configure these fields in your Application Settings (/app/13905150/settings).

□ Request email addresses from users

CODE:

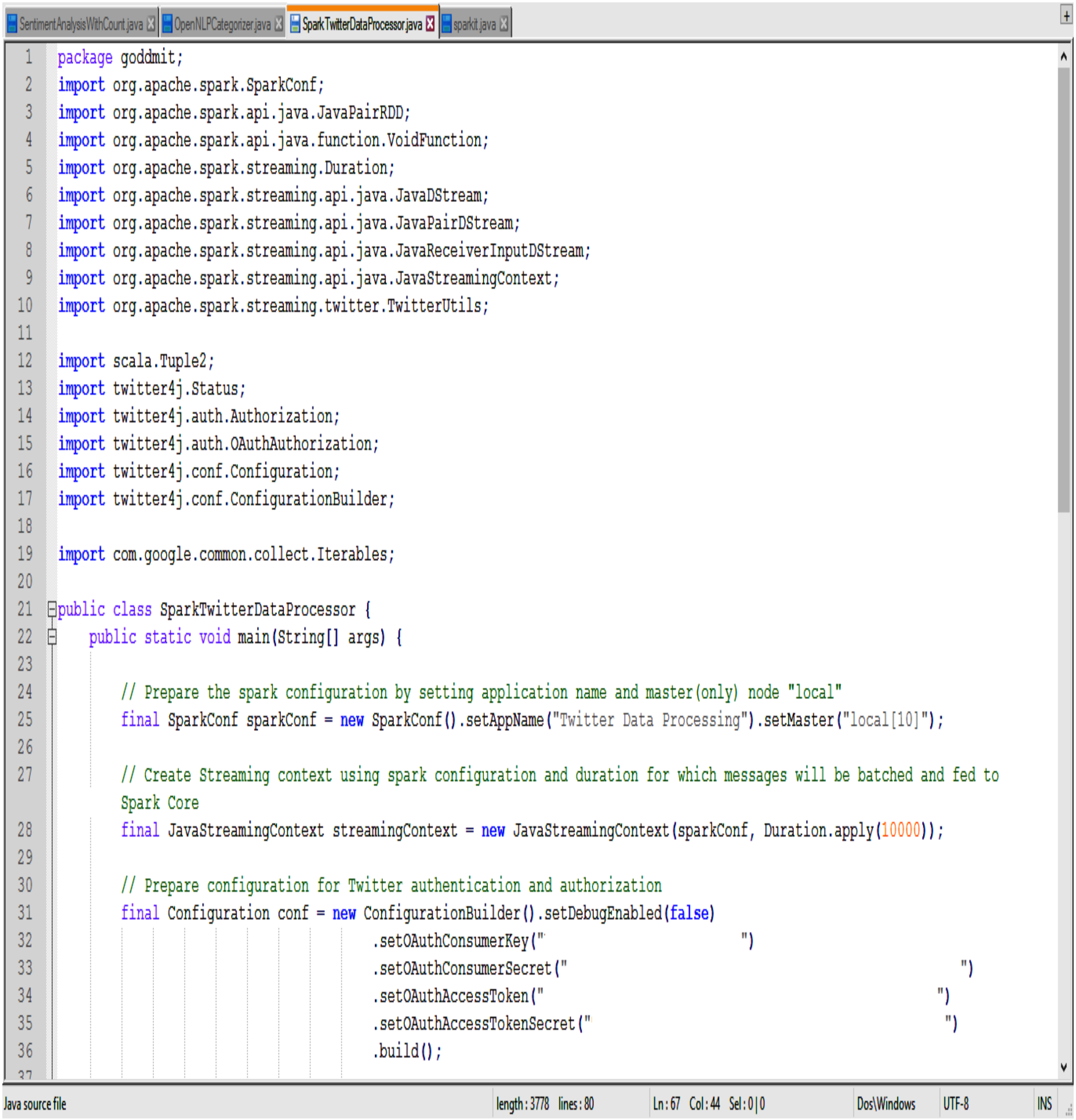
Maven Dependencies:



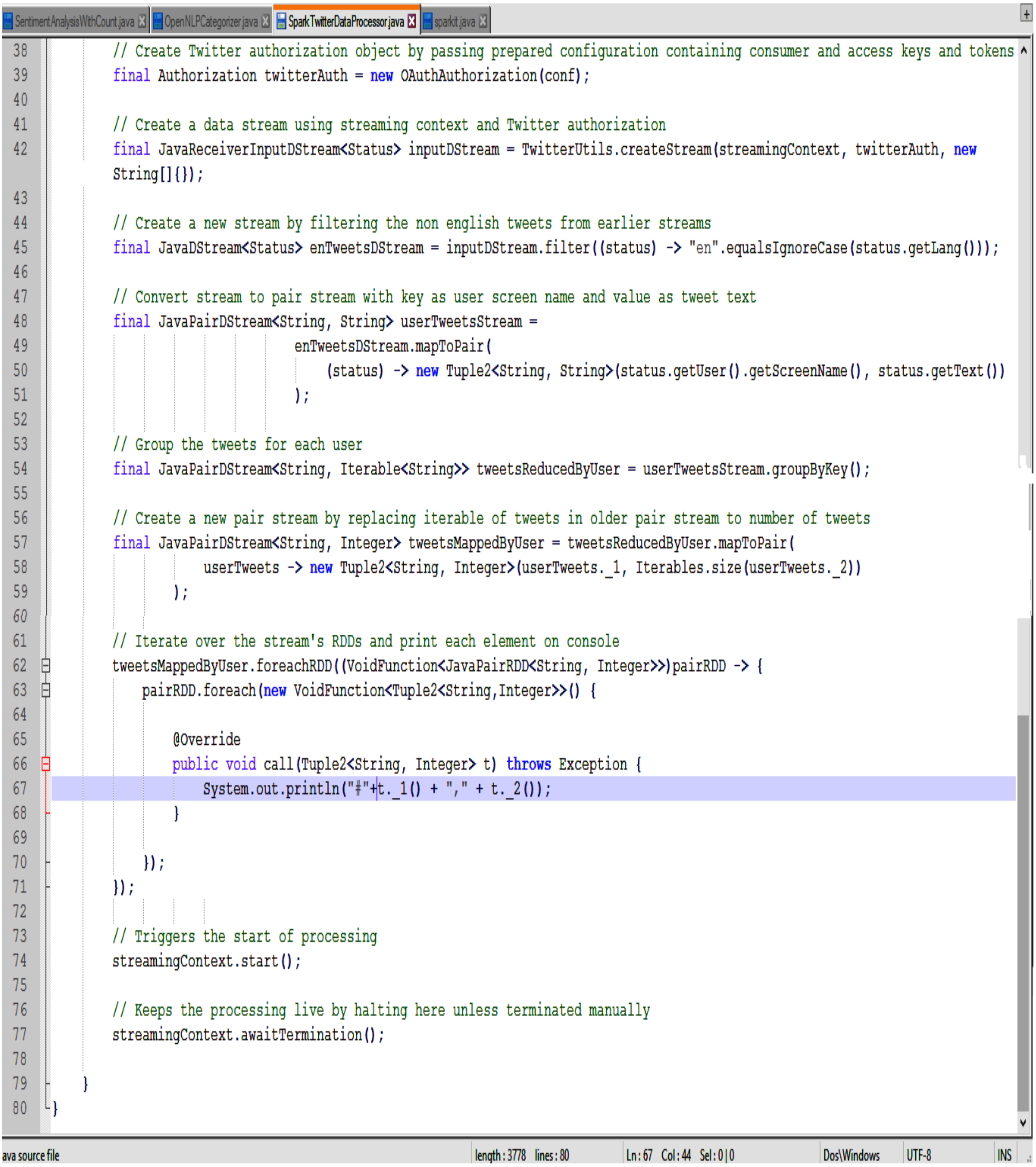
SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java
pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
    <modelVersion>4.0.0</modelVersion>
    <groupId>com. sparks. spark-streaming.basics</groupId>
    <artifactId>spark-streaming-basics</artifactId>
    <version>0.0.1-SNAPSHOT</version>
    <properties>
        <spark.version>1.6.2</spark.version>
    </properties>

    <dependencies>
        <!-- https://mvnrepository.com/artifact/org.apache.spark/spark-streaming_2.11 -->
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-streaming_2.11</artifactId>
            <version>${spark.version}</version>
        </dependency>
        <!-- https://mvnrepository.com/artifact/org.apache.spark/spark-streaming-twitter_2.11 -->
        <dependency>
            <groupId>org.apache.spark</groupId>
            <artifactId>spark-streaming-twitter_2.11</artifactId>
            <version>${spark.version}</version>
        </dependency>
        <dependency>
            <groupId>org.apache.opennlp</groupId>
            <artifactId>opennlp-tools</artifactId>
            <version>1.5.3</version>
        </dependency>
        <dependency>
            <groupId>org.twitter4j</groupId>
            <artifactId>twitter4j-core</artifactId>
            <version>4.0.1</version>
        </dependency>
        <dependency>
            <groupId>org.twitter4j</groupId>
            <artifactId>twitter4j-stream</artifactId>
            <version>4.0.1</version>
        </dependency>
        <dependency>
        <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <version>1.7.7</version>
    </dependency>
    <dependency>
    <groupId>org.apache.opennlp</groupId>
    <artifactId>opennlp-tools</artifactId>
    <version>1.5.3</version>
    </dependency>
    </dependencies>
</project>
```

Spark Twitter Data Processor.java

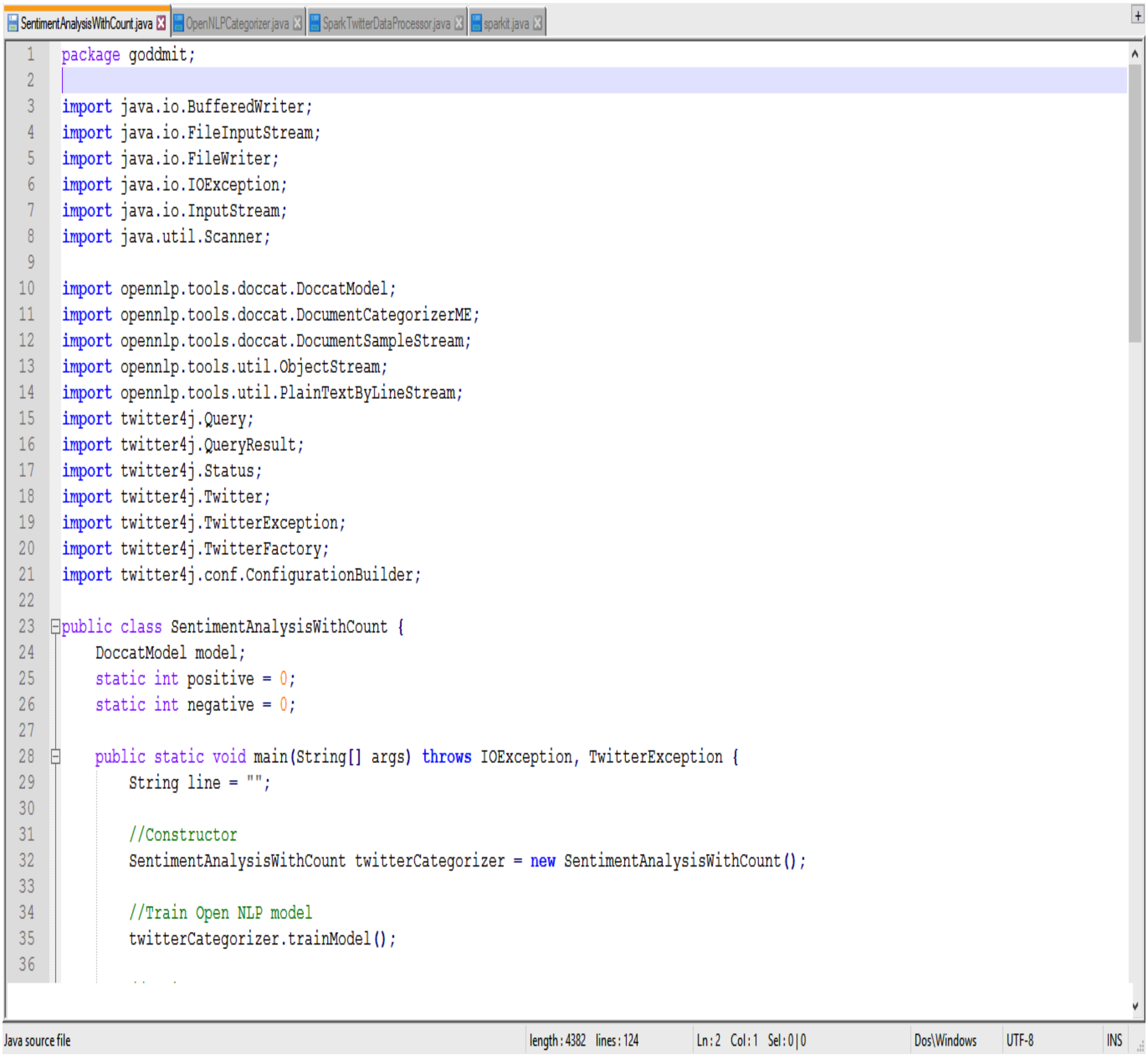


```java
package goddmit;
import org.apache.spark.SparkConf;
import org.apache.spark.api.java.JavaPairRDD;
import org.apache.spark.api.java.function.VoidFunction;
import org.apache.spark.streaming.Duration;
import org.apache.spark.streaming.api.java.JavaDStream;
import org.apache.spark.streaming.api.java.JavaPairDStream;
import org.apache.spark.streaming.api.java.JavaReceiverInputDStream;
import org.apache.spark.streaming.api.java.JavaStreamingContext;
import org.apache.spark.streaming.twitter.TwitterUtils;

import scala.Tuple2;
import twitter4j.Status;
import twitter4j.auth.Authorization;
import twitter4j.auth.OAuthAuthorization;
import twitter4j.conf.Configuration;
import twitter4j.conf.ConfigurationBuilder;

import com.google.common.collect.Iterables;

public class SparkTwitterDataProcessor {
    public static void main(String[] args) {

        // Prepare the spark configuration by setting application name and master(only) node "local"
        final SparkConf sparkConf = new SparkConf().setAppName("Twitter Data Processing").setMaster("local[10]");

        // Create Streaming context using spark configuration and duration for which messages will be batched and fed to
        Spark Core
        final JavaStreamingContext streamingContext = new JavaStreamingContext(sparkConf, Duration.apply(10000));

        // Prepare configuration for Twitter authentication and authorization
        final Configuration conf = new ConfigurationBuilder().setDebugEnabled(false)
                                        .setOAuthConsumerKey("                         ")
                                        .setOAuthConsumerSecret("                                              ")
                                        .setOAuthAccessToken("                                              ")
                                        .setOAuthAccessTokenSecret("                                         ")
                                        .build();
```

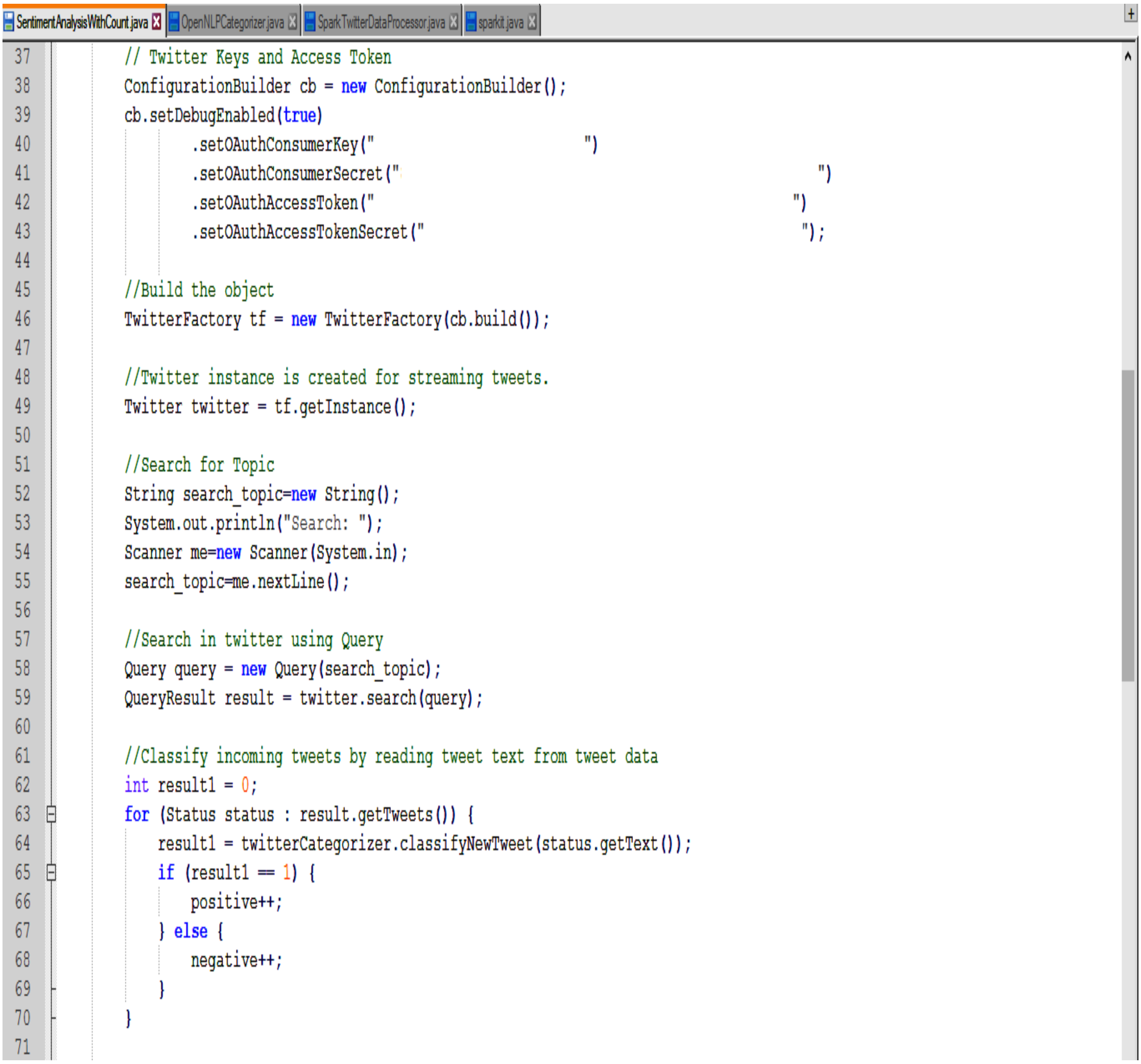

SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java

```
        // Create Twitter authorization object by passing prepared configuration containing consumer and access keys and tokens
        final Authorization twitterAuth = new OAuthAuthorization(conf);

        // Create a data stream using streaming context and Twitter authorization
        final JavaReceiverInputDStream<Status> inputDStream = TwitterUtils.createStream(streamingContext, twitterAuth, new
        String[]{});

        // Create a new stream by filtering the non english tweets from earlier streams
        final JavaDStream<Status> enTweetsDStream = inputDStream.filter((status) -> "en".equalsIgnoreCase(status.getLang()));

        // Convert stream to pair stream with key as user screen name and value as tweet text
        final JavaPairDStream<String, String> userTweetsStream =
                                enTweetsDStream.mapToPair(
                                    (status) -> new Tuple2<String, String>(status.getUser().getScreenName(), status.getText())
                                );

        // Group the tweets for each user
        final JavaPairDStream<String, Iterable<String>> tweetsReducedByUser = userTweetsStream.groupByKey();

        // Create a new pair stream by replacing iterable of tweets in older pair stream to number of tweets
        final JavaPairDStream<String, Integer> tweetsMappedByUser = tweetsReducedByUser.mapToPair(
                    userTweets -> new Tuple2<String, Integer>(userTweets._1, Iterables.size(userTweets._2))
                );

        // Iterate over the stream's RDDs and print each element on console
        tweetsMappedByUser.foreachRDD((VoidFunction<JavaPairRDD<String, Integer>>)pairRDD -> {
            pairRDD.foreach(new VoidFunction<Tuple2<String,Integer>>() {

                @Override
                public void call(Tuple2<String, Integer> t) throws Exception {
                    System.out.println("#"+t._1() + "," + t._2());
                }

            });
        });

        // Triggers the start of processing
        streamingContext.start();

        // Keeps the processing live by halting here unless terminated manually
        streamingContext.awaitTermination();

    }
}
```

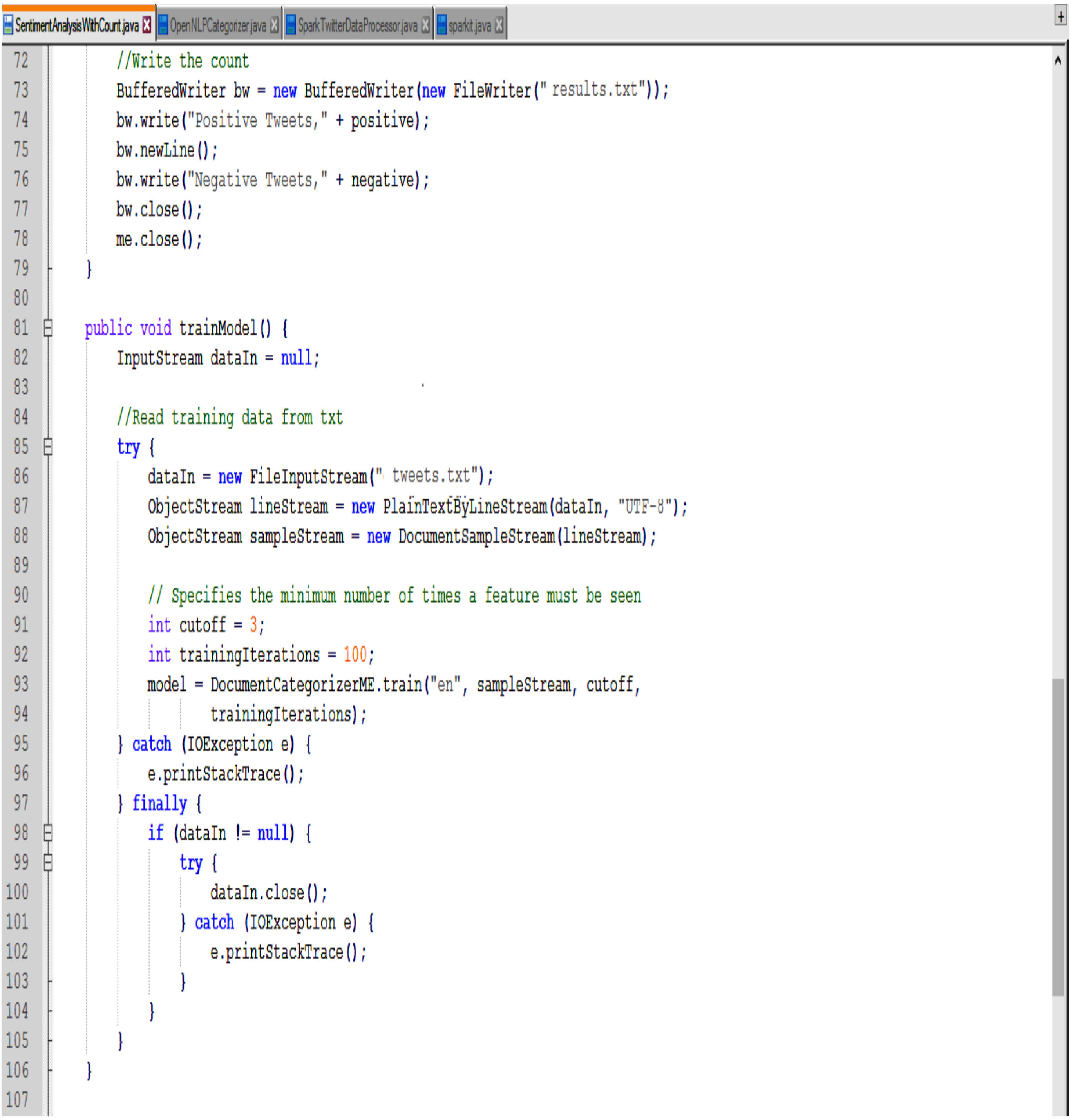
Sentiment Analysis with count.java


SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java
Java source file
length : 4382 lines : 124
Ln : 2 Col : 1 Sel : 0 | 0
Dos\Windows
UTF-8
INS

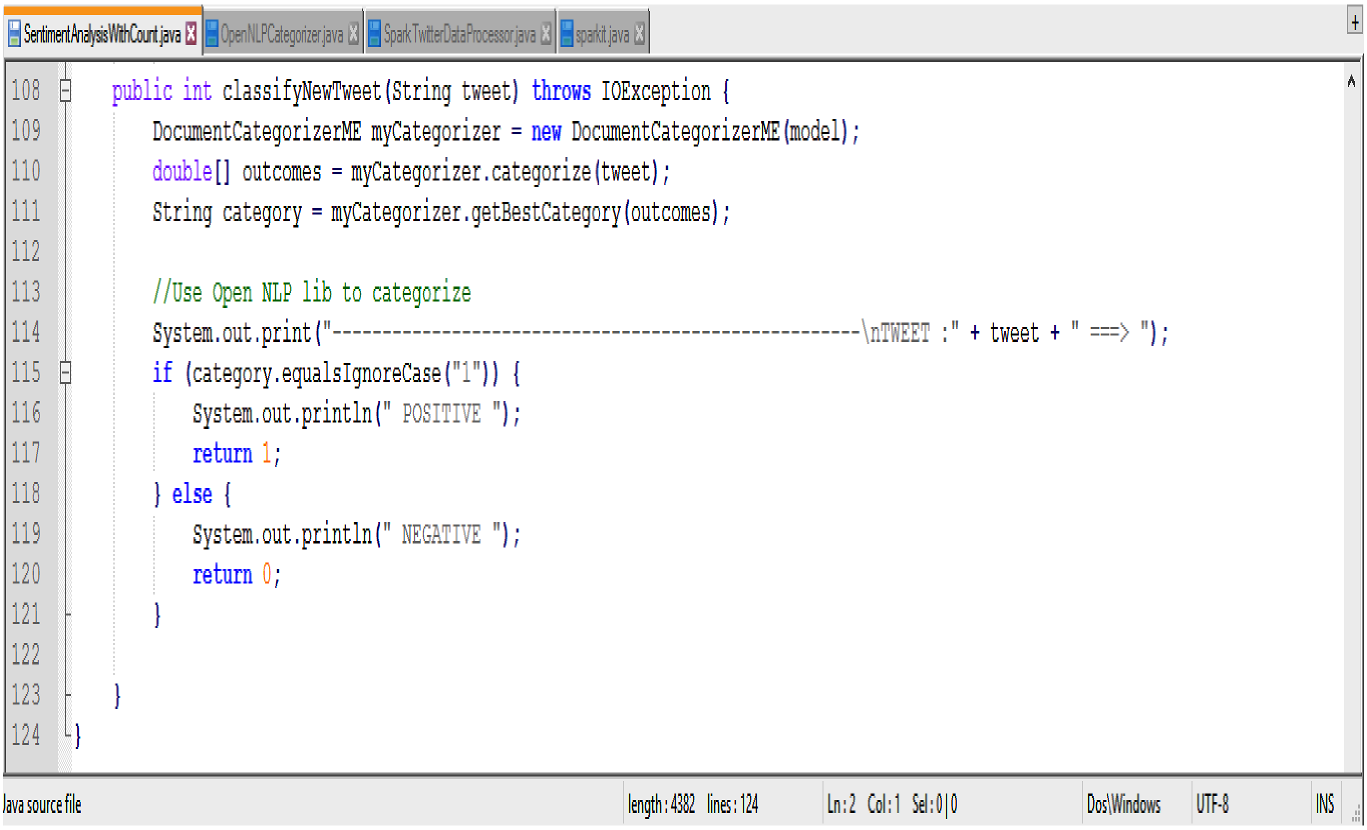
```java
package goddmit;

import java.io.BufferedWriter;
import java.io.FileInputStream;
import java.io.FileWriter;
import java.io.IOException;
import java.io.InputStream;
import java.util.Scanner;

import opennlp.tools.doccat.DoccatModel;
import opennlp.tools.doccat.DocumentCategorizerME;
import opennlp.tools.doccat.DocumentSampleStream;
import opennlp.tools.util.ObjectStream;
import opennlp.tools.util.PlainTextByLineStream;
import twitter4j.Query;
import twitter4j.QueryResult;
import twitter4j.Status;
import twitter4j.Twitter;
import twitter4j.TwitterException;
import twitter4j.TwitterFactory;
import twitter4j.conf.ConfigurationBuilder;

public class SentimentAnalysisWithCount {
    DoccatModel model;
    static int positive = 0;
    static int negative = 0;

    public static void main(String[] args) throws IOException, TwitterException {
        String line = "";

        //Constructor
        SentimentAnalysisWithCount twitterCategorizer = new SentimentAnalysisWithCount();

        //Train Open NLP model
        twitterCategorizer.trainModel();

```


SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java

```
        // Twitter Keys and Access Token
        ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true)
                .setOAuthConsumerKey("                          ")
                .setOAuthConsumerSecret("                                            ")
                .setOAuthAccessToken("                                            ")
                .setOAuthAccessTokenSecret("                                         ");

        //Build the object
        TwitterFactory tf = new TwitterFactory(cb.build());

        //Twitter instance is created for streaming tweets.
        Twitter twitter = tf.getInstance();

        //Search for Topic
        String search_topic=new String();
        System.out.println("Search: ");
        Scanner me=new Scanner(System.in);
        search_topic=me.nextLine();

        //Search in twitter using Query
        Query query = new Query(search_topic);
        QueryResult result = twitter.search(query);

        //Classify incoming tweets by reading tweet text from tweet data
        int result1 = 0;
        for (Status status : result.getTweets()) {
            result1 = twitterCategorizer.classifyNewTweet(status.getText());
            if (result1 == 1) {
                positive++;
            } else {
                negative++;
            }
        }

```

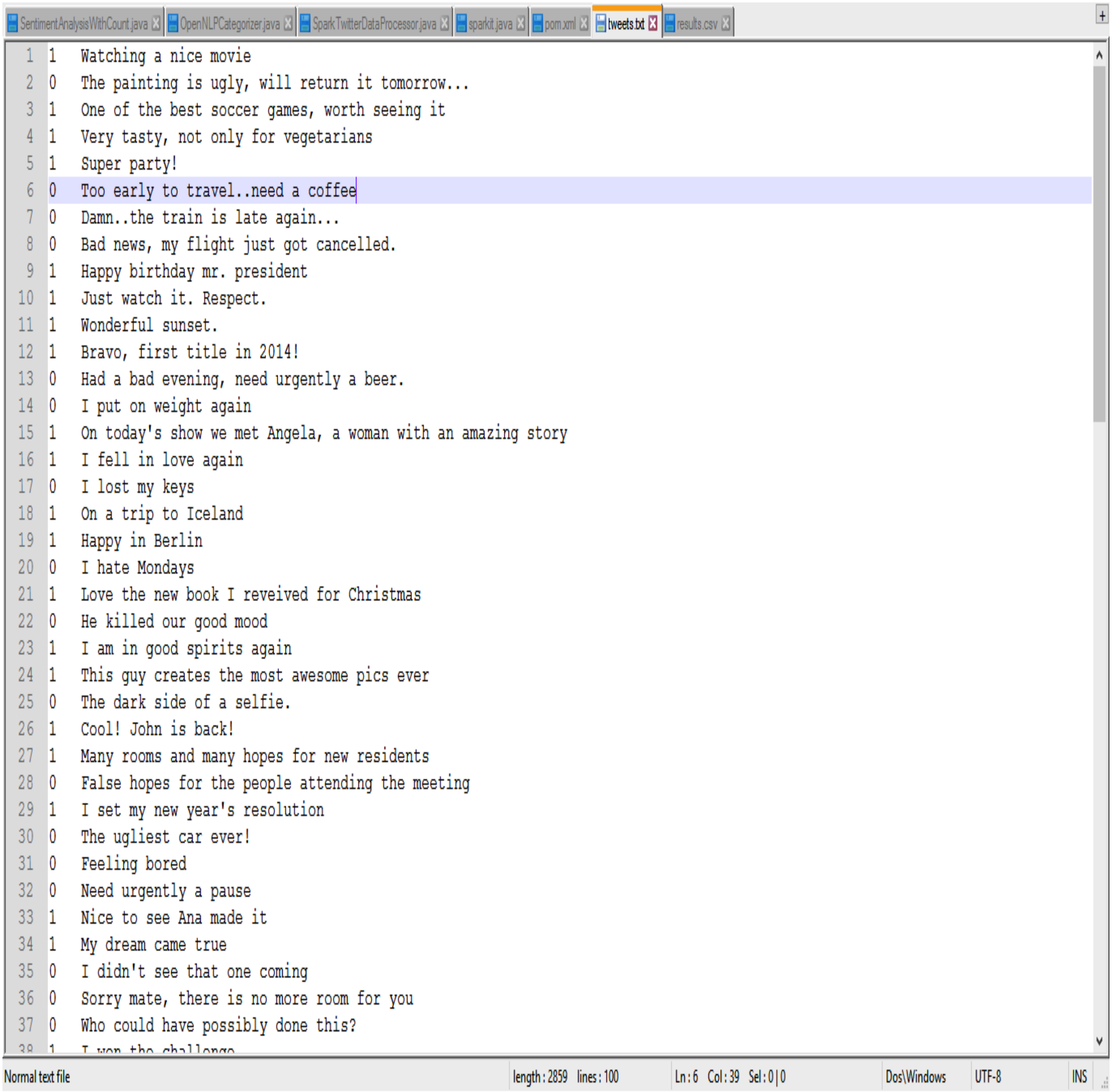

SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java

```
        //Write the count
        BufferedWriter bw = new BufferedWriter(new FileWriter(" results.txt"));
        bw.write("Positive Tweets," + positive);
        bw.newLine();
        bw.write("Negative Tweets," + negative);
        bw.close();
        me.close();
    }

    public void trainModel() {
        InputStream dataIn = null;

        //Read training data from txt
        try {
            dataIn = new FileInputStream(" tweets.txt");
            ObjectStream lineStream = new PlainTextByLineStream(dataIn, "UTF-8");
            ObjectStream sampleStream = new DocumentSampleStream(lineStream);

            // Specifies the minimum number of times a feature must be seen
            int cutoff = 3;
            int trainingIterations = 100;
            model = DocumentCategorizerME.train("en", sampleStream, cutoff,
                    trainingIterations);
        } catch (IOException e) {
            e.printStackTrace();
        } finally {
            if (dataIn != null) {
                try {
                    dataIn.close();
                } catch (IOException e) {
                    e.printStackTrace();
                }
            }
        }
    }
```

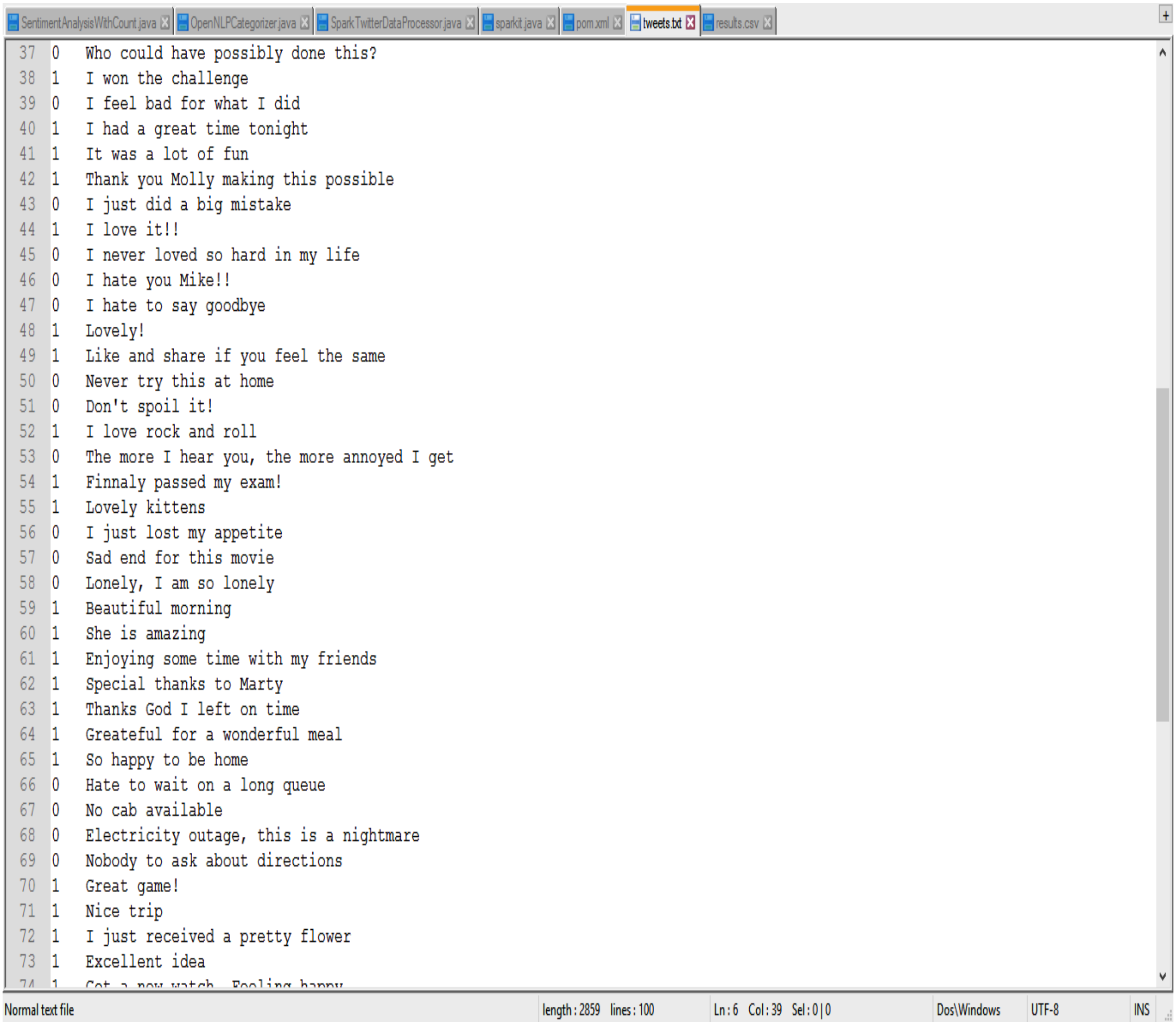

SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java
Java source file
length : 4382 lines : 124
Ln : 2 Col : 1 Sel : 0 | 0
Dos\Windows
UTF-8
INS

```
    public int classifyNewTweet(String tweet) throws IOException {
        DocumentCategorizerME myCategorizer = new DocumentCategorizerME(model);
        double[] outcomes = myCategorizer.categorize(tweet);
        String category = myCategorizer.getBestCategory(outcomes);

        //Use Open NLP lib to categorize
        System.out.print("-----------------------------------------------------\nTWEET :" + tweet + " ===> ");
        if (category.equalsIgnoreCase("1")) {
            System.out.println(" POSITIVE ");
            return 1;
        } else {
            System.out.println(" NEGATIVE ");
            return 0;
        }

    }
}
```

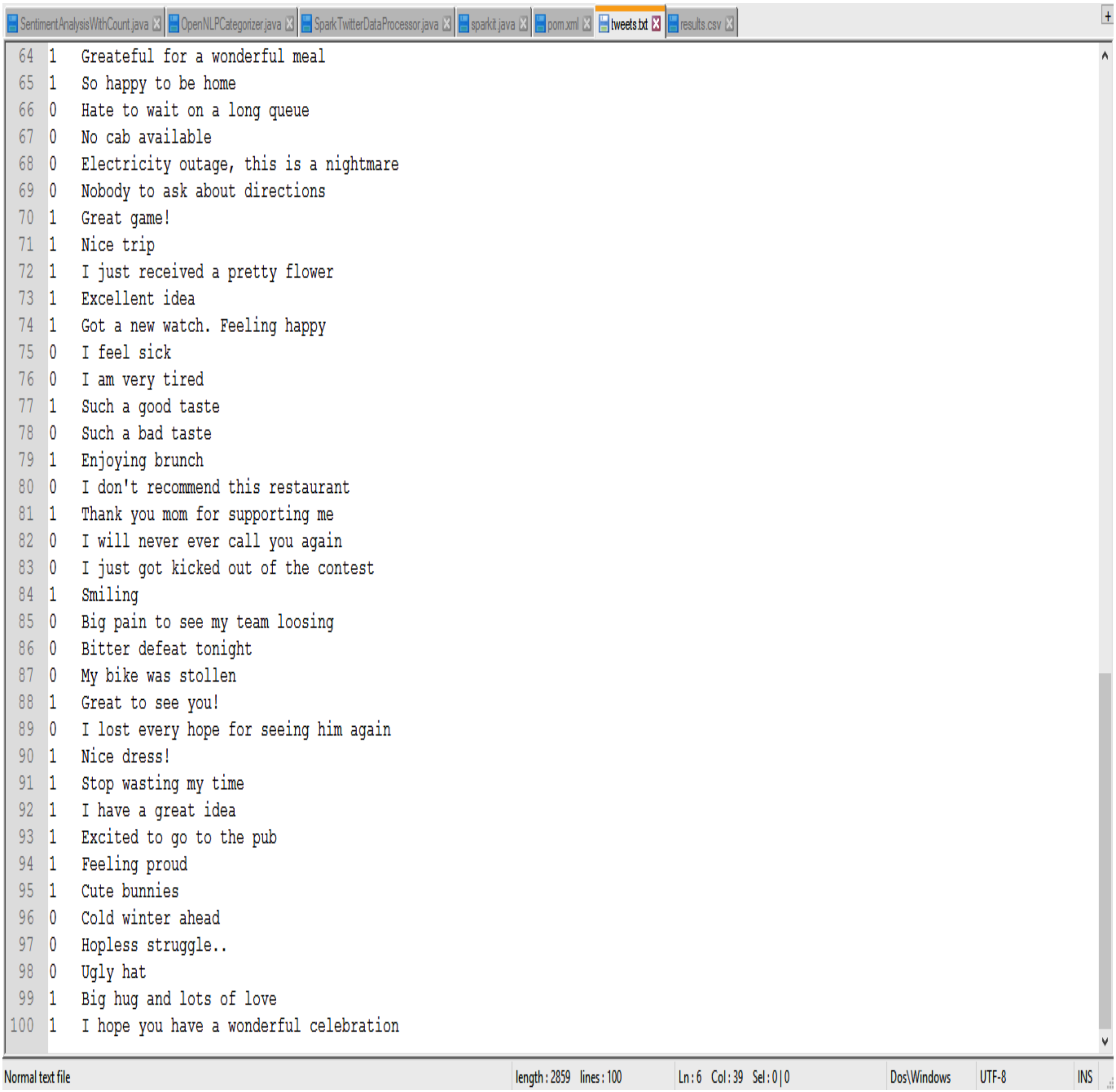
Training Data: tweets.txt


SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java
pom.xml
tweets.txt
results.csv
Normal text file
length : 2859 lines : 100
Ln : 6 Col : 39 Sel : 0 | 0
Dos\Windows
UTF-8
INS

```
1  Watching a nice movie
0  The painting is ugly, will return it tomorrow...
1  One of the best soccer games, worth seeing it
1  Very tasty, not only for vegetarians
1  Super party!
0  Too early to travel..need a coffee
0  Damn..the train is late again...
0  Bad news, my flight just got cancelled.
1  Happy birthday mr. president
1  Just watch it. Respect.
1  Wonderful sunset.
1  Bravo, first title in 2014!
0  Had a bad evening, need urgently a beer.
0  I put on weight again
1  On today's show we met Angela, a woman with an amazing story
1  I fell in love again
0  I lost my keys
1  On a trip to Iceland
1  Happy in Berlin
0  I hate Mondays
1  Love the new book I reveived for Christmas
0  He killed our good mood
1  I am in good spirits again
1  This guy creates the most awesome pics ever
0  The dark side of a selfie.
1  Cool! John is back!
1  Many rooms and many hopes for new residents
0  False hopes for the people attending the meeting
1  I set my new year's resolution
0  The ugliest car ever!
0  Feeling bored
0  Need urgently a pause
1  Nice to see Ana made it
1  My dream came true
0  I didn't see that one coming
0  Sorry mate, there is no more room for you
0  Who could have possibly done this?
1  I won the challenge
```


SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java
pom.xml
tweets.txt
results.csv
Normal text file
length : 2859 lines : 100
Ln : 6 Col : 39 Sel : 0 | 0
Dos\Windows
UTF-8
INS

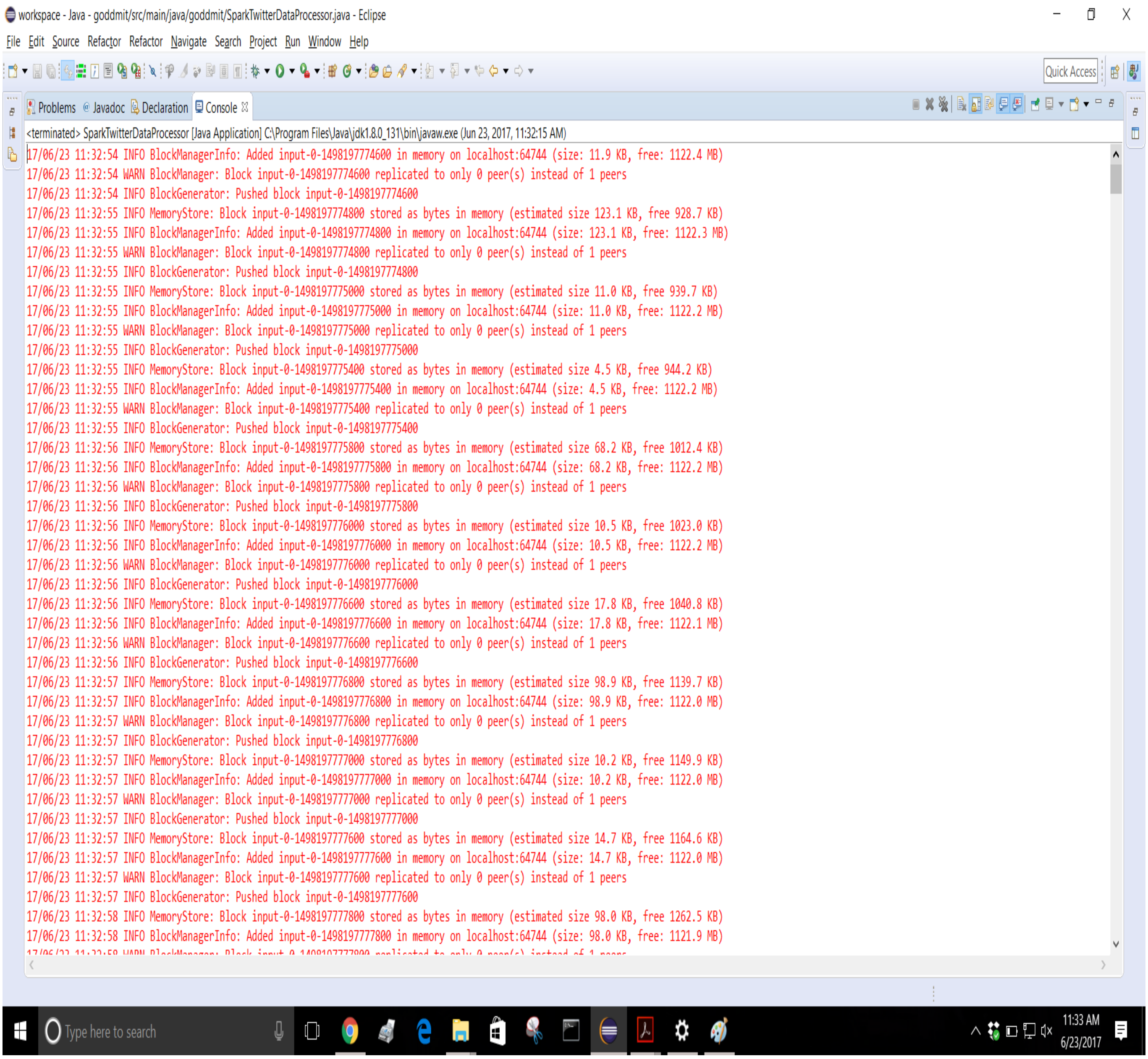
```
0   Who could have possibly done this?
1   I won the challenge
0   I feel bad for what I did
1   I had a great time tonight
1   It was a lot of fun
1   Thank you Molly making this possible
0   I just did a big mistake
1   I love it!!
0   I never loved so hard in my life
0   I hate you Mike!!
0   I hate to say goodbye
1   Lovely!
1   Like and share if you feel the same
0   Never try this at home
0   Don't spoil it!
1   I love rock and roll
0   The more I hear you, the more annoyed I get
1   Finnaly passed my exam!
1   Lovely kittens
0   I just lost my appetite
0   Sad end for this movie
0   Lonely, I am so lonely
1   Beautiful morning
1   She is amazing
1   Enjoying some time with my friends
1   Special thanks to Marty
1   Thanks God I left on time
1   Greateful for a wonderful meal
1   So happy to be home
0   Hate to wait on a long queue
0   No cab available
0   Electricity outage, this is a nightmare
0   Nobody to ask about directions
1   Great game!
1   Nice trip
1   I just received a pretty flower
1   Excellent idea
1   Got a new watch. Feeling happy
```


SentimentAnalysisWithCount.java
OpenNLPCategorizer.java
SparkTwitterDataProcessor.java
sparkit.java
pom.xml
tweets.txt
results.csv
Normal text file
length : 2859 lines : 100
Ln : 6 Col : 39 Sel : 0 | 0
Dos\Windows
UTF-8
INS

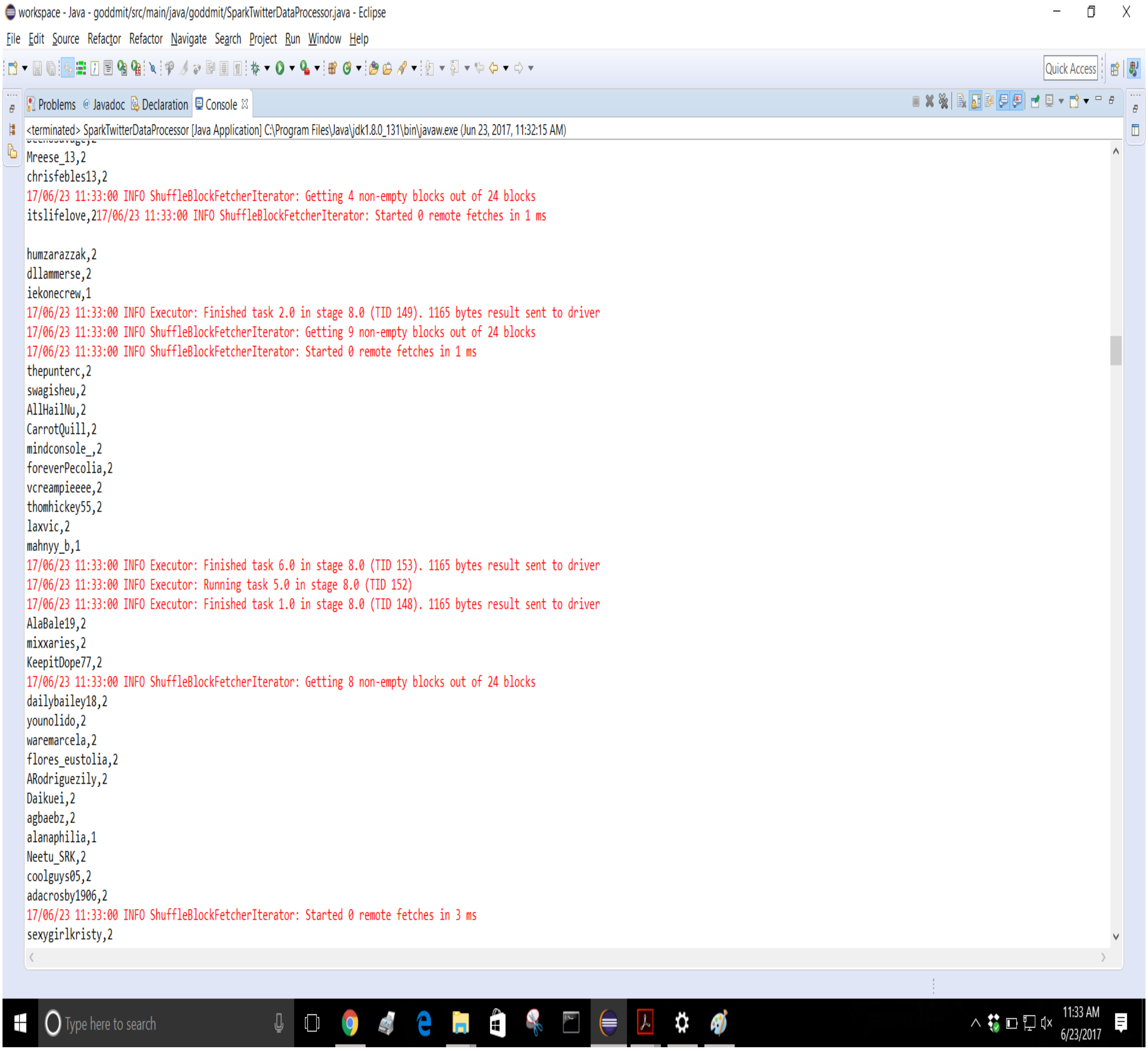
```
1	Greateful for a wonderful meal
1	So happy to be home
0	Hate to wait on a long queue
0	No cab available
0	Electricity outage, this is a nightmare
0	Nobody to ask about directions
1	Great game!
1	Nice trip
1	I just received a pretty flower
1	Excellent idea
1	Got a new watch. Feeling happy
0	I feel sick
0	I am very tired
1	Such a good taste
0	Such a bad taste
1	Enjoying brunch
0	I don't recommend this restaurant
1	Thank you mom for supporting me
0	I will never ever call you again
0	I just got kicked out of the contest
1	Smiling
0	Big pain to see my team loosing
0	Bitter defeat tonight
0	My bike was stollen
1	Great to see you!
0	I lost every hope for seeing him again
1	Nice dress!
1	Stop wasting my time
1	I have a great idea
1	Excited to go to the pub
1	Feeling proud
1	Cute bunnies
0	Cold winter ahead
0	Hopless struggle..
0	Ugly hat
1	Big hug and lots of love
1	I hope you have a wonderful celebration
```

RESULTS:

SparkTwitterDataProcessor:

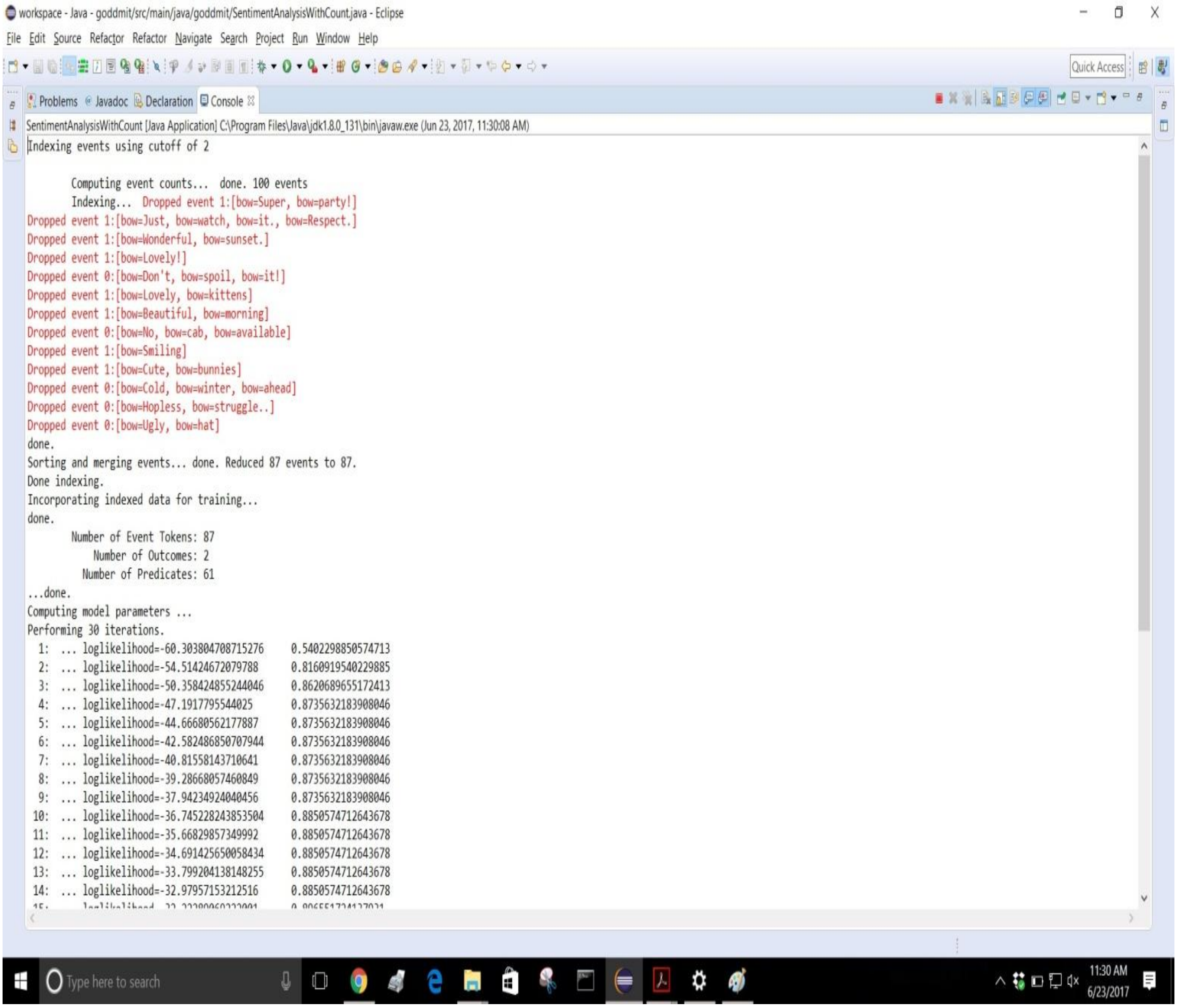

workspace - Java - goddmit/src/main/java/goddmit/SparkTwitterDataProcessor.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Quick Access
Problems
Javadoc
Declaration
Console
Type here to search
11:33 AM
6/23/2017

```
<terminated> SparkTwitterDataProcessor [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (Jun 23, 2017, 11:32:15 AM)
17/06/23 11:32:54 INFO BlockManagerInfo: Added input-0-1498197774600 in memory on localhost:64744 (size: 11.9 KB, free: 1122.4 MB)
17/06/23 11:32:54 WARN BlockManager: Block input-0-1498197774600 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:54 INFO BlockGenerator: Pushed block input-0-1498197774600
17/06/23 11:32:55 INFO MemoryStore: Block input-0-1498197774800 stored as bytes in memory (estimated size 123.1 KB, free 928.7 KB)
17/06/23 11:32:55 INFO BlockManagerInfo: Added input-0-1498197774800 in memory on localhost:64744 (size: 123.1 KB, free: 1122.3 MB)
17/06/23 11:32:55 WARN BlockManager: Block input-0-1498197774800 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:55 INFO BlockGenerator: Pushed block input-0-1498197774800
17/06/23 11:32:55 INFO MemoryStore: Block input-0-1498197775000 stored as bytes in memory (estimated size 11.0 KB, free 939.7 KB)
17/06/23 11:32:55 INFO BlockManagerInfo: Added input-0-1498197775000 in memory on localhost:64744 (size: 11.0 KB, free: 1122.2 MB)
17/06/23 11:32:55 WARN BlockManager: Block input-0-1498197775000 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:55 INFO BlockGenerator: Pushed block input-0-1498197775000
17/06/23 11:32:55 INFO MemoryStore: Block input-0-1498197775400 stored as bytes in memory (estimated size 4.5 KB, free 944.2 KB)
17/06/23 11:32:55 INFO BlockManagerInfo: Added input-0-1498197775400 in memory on localhost:64744 (size: 4.5 KB, free: 1122.2 MB)
17/06/23 11:32:55 WARN BlockManager: Block input-0-1498197775400 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:55 INFO BlockGenerator: Pushed block input-0-1498197775400
17/06/23 11:32:56 INFO MemoryStore: Block input-0-1498197775800 stored as bytes in memory (estimated size 68.2 KB, free 1012.4 KB)
17/06/23 11:32:56 INFO BlockManagerInfo: Added input-0-1498197775800 in memory on localhost:64744 (size: 68.2 KB, free: 1122.2 MB)
17/06/23 11:32:56 WARN BlockManager: Block input-0-1498197775800 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:56 INFO BlockGenerator: Pushed block input-0-1498197775800
17/06/23 11:32:56 INFO MemoryStore: Block input-0-1498197776000 stored as bytes in memory (estimated size 10.5 KB, free 1023.0 KB)
17/06/23 11:32:56 INFO BlockManagerInfo: Added input-0-1498197776000 in memory on localhost:64744 (size: 10.5 KB, free: 1122.2 MB)
17/06/23 11:32:56 WARN BlockManager: Block input-0-1498197776000 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:56 INFO BlockGenerator: Pushed block input-0-1498197776000
17/06/23 11:32:56 INFO MemoryStore: Block input-0-1498197776600 stored as bytes in memory (estimated size 17.8 KB, free 1040.8 KB)
17/06/23 11:32:56 INFO BlockManagerInfo: Added input-0-1498197776600 in memory on localhost:64744 (size: 17.8 KB, free: 1122.1 MB)
17/06/23 11:32:56 WARN BlockManager: Block input-0-1498197776600 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:56 INFO BlockGenerator: Pushed block input-0-1498197776600
17/06/23 11:32:57 INFO MemoryStore: Block input-0-1498197776800 stored as bytes in memory (estimated size 98.9 KB, free 1139.7 KB)
17/06/23 11:32:57 INFO BlockManagerInfo: Added input-0-1498197776800 in memory on localhost:64744 (size: 98.9 KB, free: 1122.0 MB)
17/06/23 11:32:57 WARN BlockManager: Block input-0-1498197776800 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:57 INFO BlockGenerator: Pushed block input-0-1498197776800
17/06/23 11:32:57 INFO MemoryStore: Block input-0-1498197777000 stored as bytes in memory (estimated size 10.2 KB, free 1149.9 KB)
17/06/23 11:32:57 INFO BlockManagerInfo: Added input-0-1498197777000 in memory on localhost:64744 (size: 10.2 KB, free: 1122.0 MB)
17/06/23 11:32:57 WARN BlockManager: Block input-0-1498197777000 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:57 INFO BlockGenerator: Pushed block input-0-1498197777000
17/06/23 11:32:57 INFO MemoryStore: Block input-0-1498197777600 stored as bytes in memory (estimated size 14.7 KB, free 1164.6 KB)
17/06/23 11:32:57 INFO BlockManagerInfo: Added input-0-1498197777600 in memory on localhost:64744 (size: 14.7 KB, free: 1122.0 MB)
17/06/23 11:32:57 WARN BlockManager: Block input-0-1498197777600 replicated to only 0 peer(s) instead of 1 peers
17/06/23 11:32:57 INFO BlockGenerator: Pushed block input-0-1498197777600
17/06/23 11:32:58 INFO MemoryStore: Block input-0-1498197777800 stored as bytes in memory (estimated size 98.0 KB, free 1262.5 KB)
17/06/23 11:32:58 INFO BlockManagerInfo: Added input-0-1498197777800 in memory on localhost:64744 (size: 98.0 KB, free: 1121.9 MB)
17/06/23 11:32:58 WARN BlockManager: Block input-0-1498197777800 replicated to only 0 peer(s) instead of 1 peers
```


workspace - Java - goddmit/src/main/java/goddmit/SparkTwitterDataProcessor.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Quick Access
Problems Javadoc Declaration Console
Type here to search
11:33 AM
6/23/2017

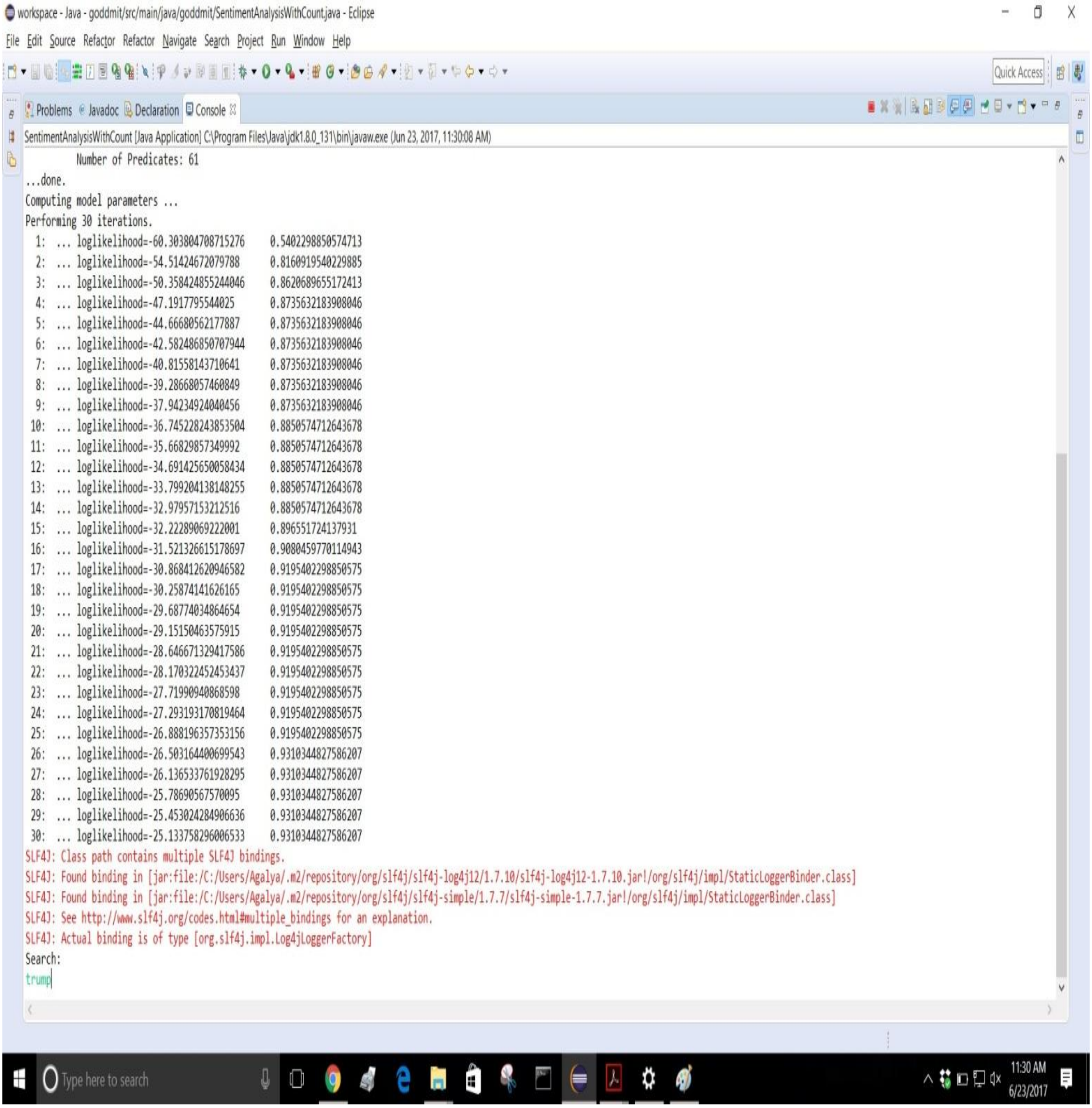
```
<terminated> SparkTwitterDataProcessor [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (Jun 23, 2017, 11:32:15 AM)
Mreese_13,2
chrisfebles13,2
17/06/23 11:33:00 INFO ShuffleBlockFetcherIterator: Getting 4 non-empty blocks out of 24 blocks
itslifelove,217/06/23 11:33:00 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms

humzarazzak,2
dllammerse,2
iekonecrew,1
17/06/23 11:33:00 INFO Executor: Finished task 2.0 in stage 8.0 (TID 149). 1165 bytes result sent to driver
17/06/23 11:33:00 INFO ShuffleBlockFetcherIterator: Getting 9 non-empty blocks out of 24 blocks
17/06/23 11:33:00 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 1 ms
thepunterc,2
swagisheu,2
AllHailNu,2
CarrotQuill,2
mindconsole_,2
foreverPecolia,2
vcreampieeee,2
thomhickey55,2
laxvic,2
mahnyy_b,1
17/06/23 11:33:00 INFO Executor: Finished task 6.0 in stage 8.0 (TID 153). 1165 bytes result sent to driver
17/06/23 11:33:00 INFO Executor: Running task 5.0 in stage 8.0 (TID 152)
17/06/23 11:33:00 INFO Executor: Finished task 1.0 in stage 8.0 (TID 148). 1165 bytes result sent to driver
AlaBale19,2
mixxaries,2
KeepitDope77,2
17/06/23 11:33:00 INFO ShuffleBlockFetcherIterator: Getting 8 non-empty blocks out of 24 blocks
dailybailey18,2
younolido,2
waremarcela,2
flores_eustolia,2
ARodriguezily,2
Daikuei,2
agbaebz,2
alanaphilia,1
Neetu_SRK,2
coolguys05,2
adacrosby1906,2
17/06/23 11:33:00 INFO ShuffleBlockFetcherIterator: Started 0 remote fetches in 3 ms
sexygirlkristy,2
```

Sentiment Analysis with count:


workspace - Java - goddmit/src/main/java/goddmit/SentimentAnalysisWithCount.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Quick Access
Problems Javadoc Declaration Console
Type here to search
11:30 AM
6/23/2017

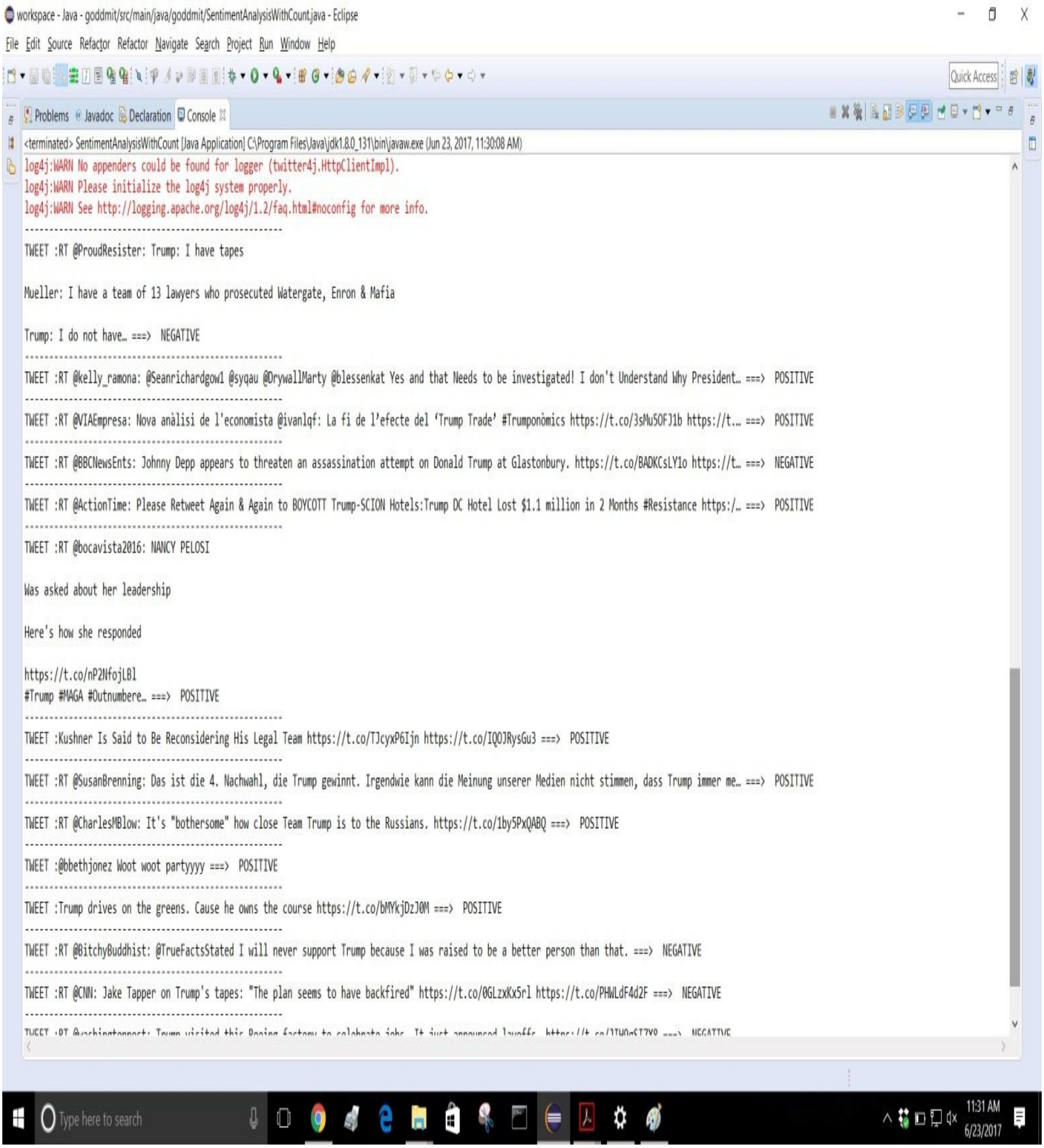
SentimentAnalysisWithCount [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (Jun 23, 2017, 11:30:08 AM)

```
Indexing events using cutoff of 2

	Computing event counts...  done. 100 events
	Indexing...  Dropped event 1:[bow=Super, bow=party!]
Dropped event 1:[bow=Just, bow=watch, bow=it., bow=Respect.]
Dropped event 1:[bow=Wonderful, bow=sunset.]
Dropped event 1:[bow=Lovely!]
Dropped event 0:[bow=Don't, bow=spoil, bow=it!]
Dropped event 1:[bow=Lovely, bow=kittens]
Dropped event 1:[bow=Beautiful, bow=morning]
Dropped event 0:[bow=No, bow=cab, bow=available]
Dropped event 1:[bow=Smiling]
Dropped event 1:[bow=Cute, bow=bunnies]
Dropped event 0:[bow=Cold, bow=winter, bow=ahead]
Dropped event 0:[bow=Hopless, bow=struggle..]
Dropped event 0:[bow=Ugly, bow=hat]
done.
Sorting and merging events... done. Reduced 87 events to 87.
Done indexing.
Incorporating indexed data for training...
done.
	Number of Event Tokens: 87
	    Number of Outcomes: 2
	  Number of Predicates: 61
...done.
Computing model parameters ...
Performing 30 iterations.
  1:  ... loglikelihood=-60.303804708715276	0.5402298850574713
  2:  ... loglikelihood=-54.51424672079788	0.8160919540229885
  3:  ... loglikelihood=-50.358424855244046	0.8620689655172413
  4:  ... loglikelihood=-47.1917795544025	0.8735632183908046
  5:  ... loglikelihood=-44.66680562177887	0.8735632183908046
  6:  ... loglikelihood=-42.582486850707944	0.8735632183908046
  7:  ... loglikelihood=-40.81558143710641	0.8735632183908046
  8:  ... loglikelihood=-39.28668057460849	0.8735632183908046
  9:  ... loglikelihood=-37.94234924040456	0.8735632183908046
 10:  ... loglikelihood=-36.745228243853504	0.8850574712643678
 11:  ... loglikelihood=-35.66829857349992	0.8850574712643678
 12:  ... loglikelihood=-34.691425650058434	0.8850574712643678
 13:  ... loglikelihood=-33.799204138148255	0.8850574712643678
 14:  ... loglikelihood=-32.97957153212516	0.8850574712643678
```

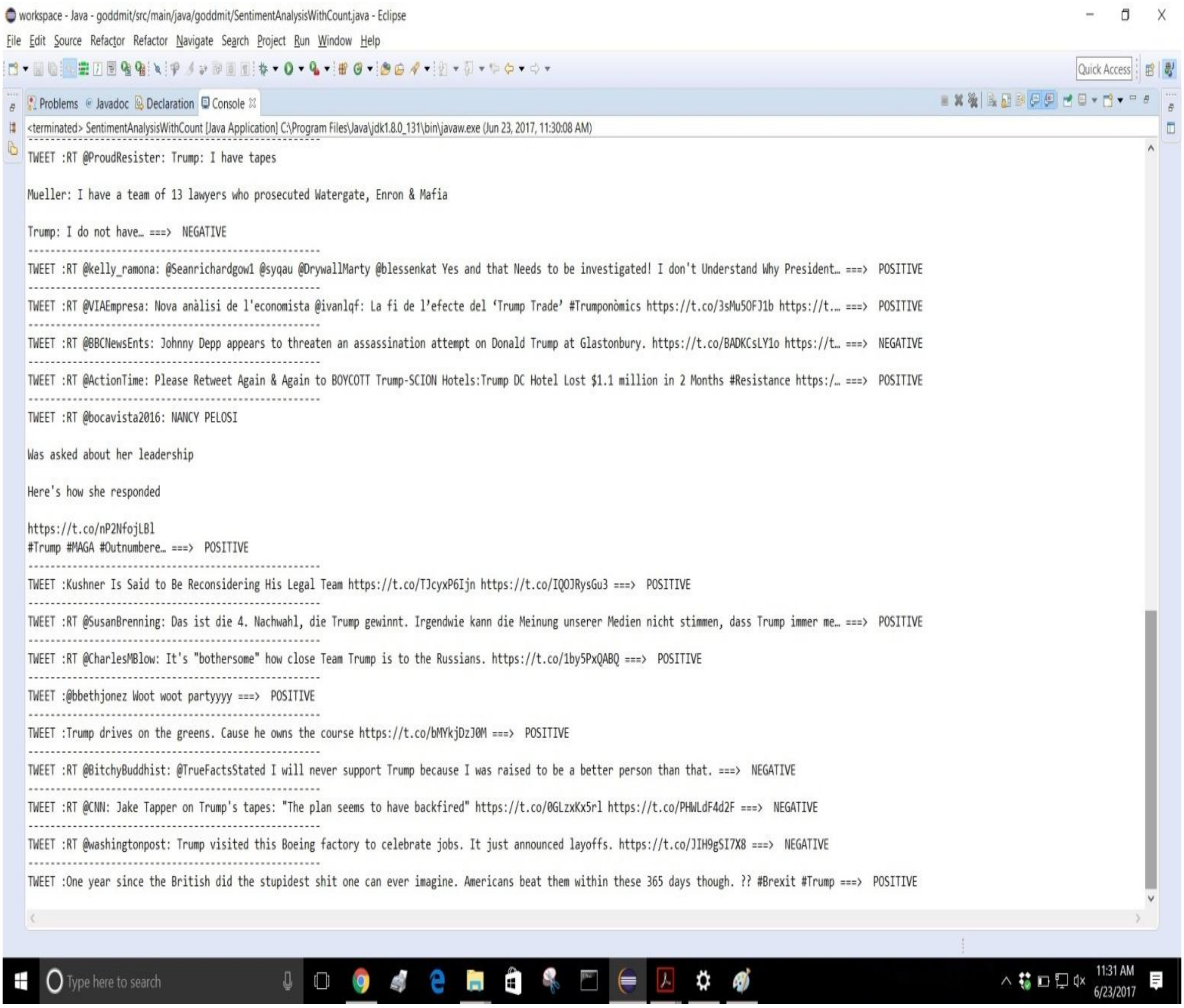

workspace - Java - goddmit/src/main/java/goddmit/SentimentAnalysisWithCount.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Quick Access
Problems
Javadoc
Declaration
Console
Type here to search
11:30 AM
6/23/2017

SentimentAnalysisWithCount [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (Jun 23, 2017, 11:30:08 AM)

```
        Number of Predicates: 61
...done.
Computing model parameters ...
Performing 30 iterations.
  1:  ... loglikelihood=-60.303804708715276    0.5402298850574713
  2:  ... loglikelihood=-54.51424672079788     0.8160919540229885
  3:  ... loglikelihood=-50.358424855244046    0.8620689655172413
  4:  ... loglikelihood=-47.1917795544025      0.8735632183908046
  5:  ... loglikelihood=-44.66680562177887     0.8735632183908046
  6:  ... loglikelihood=-42.582486850707944    0.8735632183908046
  7:  ... loglikelihood=-40.81558143710641     0.8735632183908046
  8:  ... loglikelihood=-39.28668057460849     0.8735632183908046
  9:  ... loglikelihood=-37.94234924040456     0.8735632183908046
 10:  ... loglikelihood=-36.745228243853504    0.8850574712643678
 11:  ... loglikelihood=-35.66829857349992     0.8850574712643678
 12:  ... loglikelihood=-34.691425650058434    0.8850574712643678
 13:  ... loglikelihood=-33.799204138148255    0.8850574712643678
 14:  ... loglikelihood=-32.97957153212516     0.8850574712643678
 15:  ... loglikelihood=-32.22289069222001     0.896551724137931
 16:  ... loglikelihood=-31.521326615178697    0.9080459770114943
 17:  ... loglikelihood=-30.868412620946582    0.9195402298850575
 18:  ... loglikelihood=-30.25874141626165     0.9195402298850575
 19:  ... loglikelihood=-29.68774034864654     0.9195402298850575
 20:  ... loglikelihood=-29.15150463575915     0.9195402298850575
 21:  ... loglikelihood=-28.646671329417586    0.9195402298850575
 22:  ... loglikelihood=-28.170322452453437    0.9195402298850575
 23:  ... loglikelihood=-27.71990940868598     0.9195402298850575
 24:  ... loglikelihood=-27.293193170819464    0.9195402298850575
 25:  ... loglikelihood=-26.888196357353156    0.9195402298850575
 26:  ... loglikelihood=-26.503164400699543    0.9310344827586207
 27:  ... loglikelihood=-26.136533761928295    0.9310344827586207
 28:  ... loglikelihood=-25.78690567570095     0.9310344827586207
 29:  ... loglikelihood=-25.453024284906636    0.9310344827586207
 30:  ... loglikelihood=-25.133758296006533    0.9310344827586207
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/C:/Users/Agalya/.m2/repository/org/slf4j/slf4j-log4j12/1.7.10/slf4j-log4j12-1.7.10.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/C:/Users/Agalya/.m2/repository/org/slf4j/slf4j-simple/1.7.7/slf4j-simple-1.7.7.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
Search:
trump
```


workspace - Java - goddmit/src/main/java/goddmit/SentimentAnalysisWithCount.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Quick Access
Problems Javadoc Declaration Console
Type here to search
11:31 AM
6/23/2017

```
<terminated> SentimentAnalysisWithCount [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (Jun 23, 2017, 11:30:08 AM)
log4j:WARN No appenders could be found for logger (twitter4j.HttpClientImpl).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.
----------------------------------------------------
TWEET :RT @ProudResister: Trump: I have tapes

Mueller: I have a team of 13 lawyers who prosecuted Watergate, Enron & Mafia

Trump: I do not have… ===>  NEGATIVE
----------------------------------------------------
TWEET :RT @kelly_ramona: @Seanrichardgow1 @syqau @DrywallMarty @blessenkat Yes and that Needs to be investigated! I don't Understand Why President… ===>  POSITIVE
----------------------------------------------------
TWEET :RT @VIAEmpresa: Nova anàlisi de l'economista @ivanlqf: La fi de l'efecte del 'Trump Trade' #Trumponòmics https://t.co/3sMu5OFJ1b https://t.… ===>  POSITIVE
----------------------------------------------------
TWEET :RT @BBCNewsEnts: Johnny Depp appears to threaten an assassination attempt on Donald Trump at Glastonbury. https://t.co/BADKCsLY1o https://t… ===>  NEGATIVE
----------------------------------------------------
TWEET :RT @ActionTime: Please Retweet Again & Again to BOYCOTT Trump-SCION Hotels:Trump DC Hotel Lost $1.1 million in 2 Months #Resistance https:/… ===>  POSITIVE
----------------------------------------------------
TWEET :RT @bocavista2016: NANCY PELOSI

Was asked about her leadership

Here's how she responded

https://t.co/nP2NfojLBl
#Trump #MAGA #Outnumbere… ===>  POSITIVE
----------------------------------------------------
TWEET :Kushner Is Said to Be Reconsidering His Legal Team https://t.co/TJcyxP6Ijn https://t.co/IQOJRysGu3 ===>  POSITIVE
----------------------------------------------------
TWEET :RT @SusanBrenning: Das ist die 4. Nachwahl, die Trump gewinnt. Irgendwie kann die Meinung unserer Medien nicht stimmen, dass Trump immer me… ===>  POSITIVE
----------------------------------------------------
TWEET :RT @CharlesMBlow: It's "bothersome" how close Team Trump is to the Russians. https://t.co/1by5PxQABQ ===>  POSITIVE
----------------------------------------------------
TWEET :@bbethjonez Woot woot partyyyy ===>  POSITIVE
----------------------------------------------------
TWEET :Trump drives on the greens. Cause he owns the course https://t.co/bMYkjDzJ0M ===>  POSITIVE
----------------------------------------------------
TWEET :RT @BitchyBuddhist: @TrueFactsStated I will never support Trump because I was raised to be a better person than that. ===>  NEGATIVE
----------------------------------------------------
TWEET :RT @CNN: Jake Tapper on Trump's tapes: "The plan seems to have backfired" https://t.co/0GLzxKx5rl https://t.co/PHWLdF4d2F ===>  NEGATIVE
----------------------------------------------------
TWEET :RT @washingtonpost: Trump visited this Boeing factory to celebrate jobs. It just announced layoffs. https://t.co/1JHQqSI7Y8 ===>  NEGATIVE
```


workspace - Java - goddmit/src/main/java/goddmit/SentimentAnalysisWithCount.java - Eclipse
File Edit Source Refactor Refactor Navigate Search Project Run Window Help
Quick Access
Problems Javadoc Declaration Console
Type here to search
11:31 AM
6/23/2017

```
<terminated> SentimentAnalysisWithCount [Java Application] C:\Program Files\Java\jdk1.8.0_131\bin\javaw.exe (Jun 23, 2017, 11:30:08 AM)
----------------------------------------------------
TWEET :RT @ProudResister: Trump: I have tapes

Mueller: I have a team of 13 lawyers who prosecuted Watergate, Enron & Mafia

Trump: I do not have… ===>  NEGATIVE
----------------------------------------------------
TWEET :RT @kelly_ramona: @Seanrichardgow1 @syqau @DrywallMarty @blessenkat Yes and that Needs to be investigated! I don't Understand Why President… ===>  POSITIVE
----------------------------------------------------
TWEET :RT @VIAEmpresa: Nova anàlisi de l'economista @ivanlqf: La fi de l'efecte del ‘Trump Trade’ #Trumponòmics https://t.co/3sMu5OFJ1b https://t.… ===>  POSITIVE
----------------------------------------------------
TWEET :RT @BBCNewsEnts: Johnny Depp appears to threaten an assassination attempt on Donald Trump at Glastonbury. https://t.co/BADKCsLY1o https://t… ===>  NEGATIVE
----------------------------------------------------
TWEET :RT @ActionTime: Please Retweet Again & Again to BOYCOTT Trump-SCION Hotels:Trump DC Hotel Lost $1.1 million in 2 Months #Resistance https:/… ===>  POSITIVE
----------------------------------------------------
TWEET :RT @bocavista2016: NANCY PELOSI

Was asked about her leadership

Here's how she responded

https://t.co/nP2NfojLBl
#Trump #MAGA #Outnumbere… ===>  POSITIVE
----------------------------------------------------
TWEET :Kushner Is Said to Be Reconsidering His Legal Team https://t.co/TJcyxP6Ijn https://t.co/IQOJRysGu3 ===>  POSITIVE
----------------------------------------------------
TWEET :RT @SusanBrenning: Das ist die 4. Nachwahl, die Trump gewinnt. Irgendwie kann die Meinung unserer Medien nicht stimmen, dass Trump immer me… ===>  POSITIVE
----------------------------------------------------
TWEET :RT @CharlesMBlow: It's "bothersome" how close Team Trump is to the Russians. https://t.co/1by5PxQABQ ===>  POSITIVE
----------------------------------------------------
TWEET :@bbethjonez Woot woot partyyyy ===>  POSITIVE
----------------------------------------------------
TWEET :Trump drives on the greens. Cause he owns the course https://t.co/bMYkjDzJ0M ===>  POSITIVE
----------------------------------------------------
TWEET :RT @BitchyBuddhist: @TrueFactsStated I will never support Trump because I was raised to be a better person than that. ===>  NEGATIVE
----------------------------------------------------
TWEET :RT @CNN: Jake Tapper on Trump's tapes: "The plan seems to have backfired" https://t.co/0GLzxKx5rl https://t.co/PHWLdF4d2F ===>  NEGATIVE
----------------------------------------------------
TWEET :RT @washingtonpost: Trump visited this Boeing factory to celebrate jobs. It just announced layoffs. https://t.co/JIH9gSI7X8 ===>  NEGATIVE
----------------------------------------------------
TWEET :One year since the British did the stupidest shit one can ever imagine. Americans beat them within these 365 days though. ?? #Brexit #Trump ===>  POSITIVE
```

FUTURE SCOPES

This application can be easily implemented under various situations. We can add new features as and when we require. Reusability is possible as and when require in this application. There is flexibility in all the modules.

SOFTWARE SCOPE:

- *Extensibility*: This software is extendable in ways that its original developers may not expect. The following principles enhance extensibility like updating data locally while being offline and then syncing the server when there is internet connection. We can also delete values either offline or online and then update it as and when there is an internet connection.
- *Reusability:* Reusability is possible as and when require in this application.
- *Understandability:* A method is understandable if someone other than the creator of the method can understand the code (as well as the creator after a time lapse). We use the method, which small and coherent helps to accomplish this.
- *Cost-effectiveness*: Its cost is under the budget and make within given time period. It is desirable to aim for a system with a minimum cost subject to the condition that it must satisfy the entire requirement. Scope of this document is to put down the requirements, clearly identifying the information needed by the user, the source of the information and outputs expected from the system.

REFERENCES:

i. Wikipedia

ii. Apache Spark Docs:

https://spark.apache.org/docs/latest/sql-programming-guide.html#overview

iii. Apache Open NLP

http://opennlp.apache.org/docs/1.8.0/manual/opennlp.html

iv. http://www.stdatalabs.in/

v. InfoQ blog on Big Data Processing using Apache Spark

http://www.infoq.com/articles/apache-spark-streaming?utm_source=apachesparkseries&utm_medium=link&utm_campaign=internal

vi. Spark Tutorial by Edureka

https://www.edureka.co/blog/spark-streaming/?utm_source=blog&utm_medium=left-menu&utm_campaign=spark-tutorial

vii. Tutorialspoint

https://www.tutorialspoint.com/apache_spark/apache_spark_quick_guide.htm