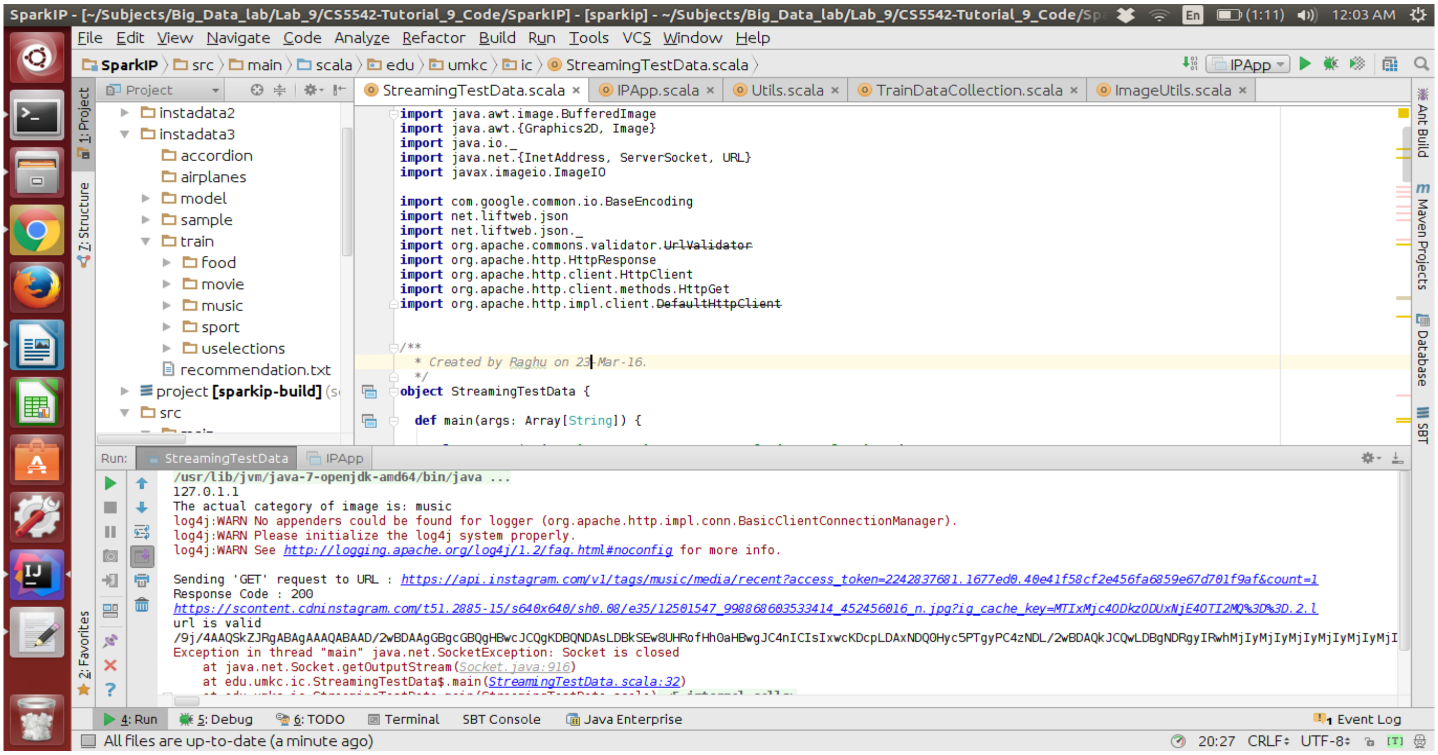
Lab assignment 9

Task 1: The task here was to create an image classification system that could classify the images based on training data collected from Instagram. The following were the steps performed to achieve the needed.

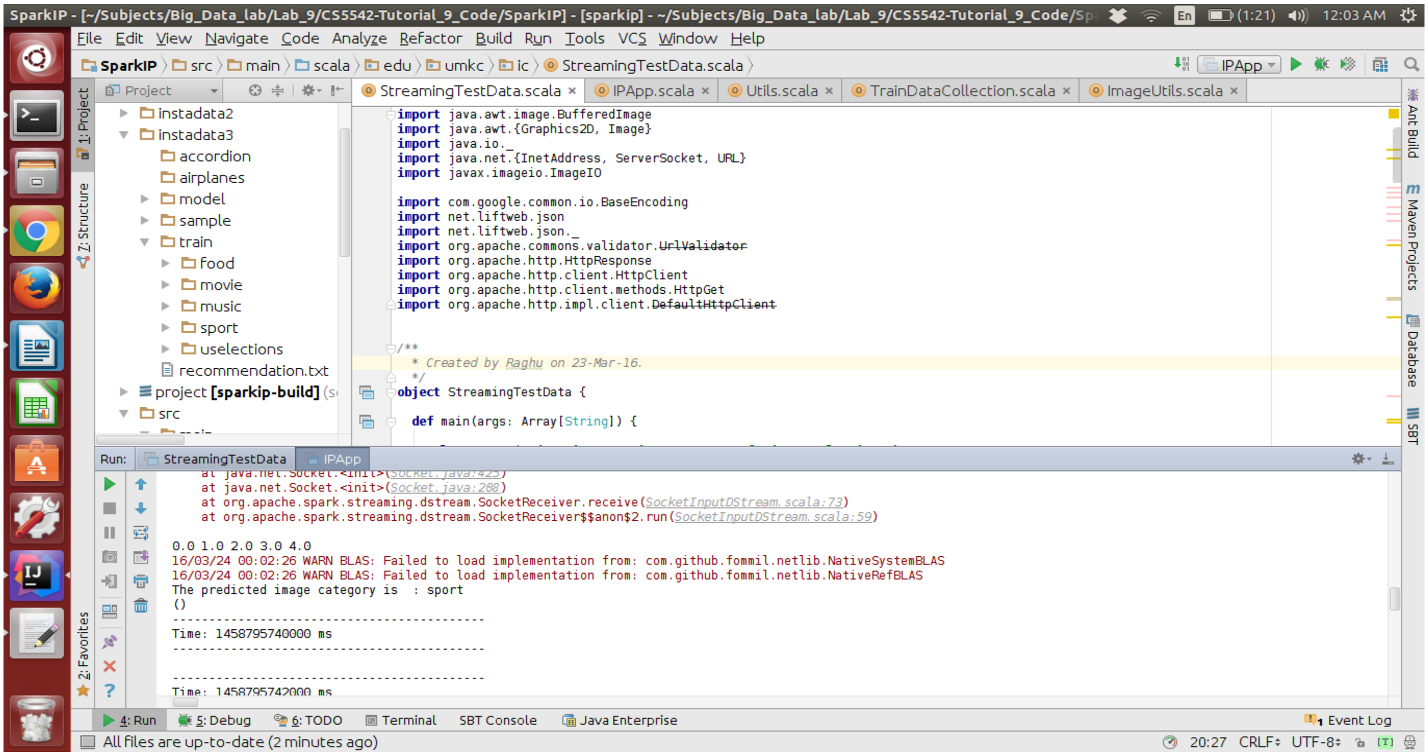
1. Create an access token with instagram.
2. Collect images on the tags of Music, Movie, Sport, Food and USElections.
3. Perform training of the model with the following categories as input.
4. Run the instagram stream to randomly pick an image from one of the above categories.
5. Test the image received and predict its category.

PFB the result for the Image Classification.


SparkIP - [~/Subjects/Big_Data_lab/Lab_9/CS5542-Tutorial_9_Code/SparkIP] - [sparkip] - ~/Subjects/Big_Data_lab/Lab_9/CS5542-Tutorial_9_Code/Sp
En (1:11) 12:03 AM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkIP src main scala edu umkc ic StreamingTestData.scala
IPApp
Project
StreamingTestData.scala IPApp.scala Utils.scala TrainDataCollection.scala ImageUtils.scala
instadata2
instadata3
accordion
airplanes
model
sample
train
food
movie
music
sport
uselections
recommendation.txt
project [sparkip-build]
src
Ant Build
Maven Projects
Database
SBT
1: Project
7: Structure
2: Favorites
Run: StreamingTestData IPApp
4: Run 5: Debug 6: TODO Terminal SBT Console Java Enterprise
Event Log
All files are up-to-date (a minute ago)
20:27 CRLF÷ UTF-8÷

```
import java.awt.image.BufferedImage
import java.awt.{Graphics2D, Image}
import java.io._
import java.net.{InetAddress, ServerSocket, URL}
import javax.imageio.ImageIO

import com.google.common.io.BaseEncoding
import net.liftweb.json
import net.liftweb.json._
import org.apache.commons.validator.UrlValidator
import org.apache.http.HttpResponse
import org.apache.http.client.HttpClient
import org.apache.http.client.methods.HttpGet
import org.apache.http.impl.client.DefaultHttpClient


/**
  * Created by Raghu on 23-Mar-16.
  */
object StreamingTestData {

  def main(args: Array[String]) {
```

```
/usr/lib/jvm/java-7-openjdk-amd64/bin/java ...
127.0.1.1
The actual category of image is: music
log4j:WARN No appenders could be found for logger (org.apache.http.impl.conn.BasicClientConnectionManager).
log4j:WARN Please initialize the log4j system properly.
log4j:WARN See http://logging.apache.org/log4j/1.2/faq.html#noconfig for more info.

Sending 'GET' request to URL : https://api.instagram.com/v1/tags/music/media/recent?access_token=2242837681.1677ed0.40e41f58cf2e456fa6859e67d701f9af&count=1
Response Code : 200
https://scontent.cdninstagram.com/t51.2885-15/s640x640/sh0.08/e35/12501547_998868603533414_452456016_n.jpg?ig_cache_key=MTIxMjc40Dkz0DUxNjE40TI2MQ%3D%3D.2.l
url is valid
/9j/4AAQSkZJRgABAgAAAQABAAD/2wBDAAgGBgcGBQgHBwcJCQgKDBQNDAsLDBkSEw8UHRofHh0aHBwgJC4nICIsIxwcKDcpLDAxNDQ0Hyc5PTgyPC4zNDL/2wBDAQkJCQwLDBgNDRgyIRwhMjIyMjIyMjIyMjIyMjIyMjI
Exception in thread "main" java.net.SocketException: Socket is closed
    at java.net.Socket.getOutputStream(Socket.java:916)
    at edu.umkc.ic.StreamingTestData$.main(StreamingTestData.scala:32)
```


SparkIP - [~/Subjects/Big_Data_lab/Lab_9/CS5542-Tutorial_9_Code/SparkIP] - [sparkip] - ~/Subjects/Big_Data_lab/Lab_9/CS5542-Tutorial_9_Code/Sp
En (1:21) 12:03 AM
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
SparkIP src main scala edu umkc ic StreamingTestData.scala
IPApp
Project
StreamingTestData.scala IPApp.scala Utils.scala TrainDataCollection.scala ImageUtils.scala
instadata2
instadata3
accordion
airplanes
model
sample
train
food
movie
music
sport
uselections
recommendation.txt
project [sparkip-build]
src
Run: StreamingTestData IPApp
4: Run 5: Debug 6: TODO Terminal SBT Console Java Enterprise
Event Log
All files are up-to-date (2 minutes ago)
20:27 CRLF÷ UTF-8÷

```
import java.awt.image.BufferedImage
import java.awt.{Graphics2D, Image}
import java.io._
import java.net.{InetAddress, ServerSocket, URL}
import javax.imageio.ImageIO

import com.google.common.io.BaseEncoding
import net.liftweb.json
import net.liftweb.json._
import org.apache.commons.validator.UrlValidator
import org.apache.http.HttpResponse
import org.apache.http.client.HttpClient
import org.apache.http.client.methods.HttpGet
import org.apache.http.impl.client.DefaultHttpClient


/**
  * Created by Raghu on 23-Mar-16.
  */
object StreamingTestData {

  def main(args: Array[String]) {
```

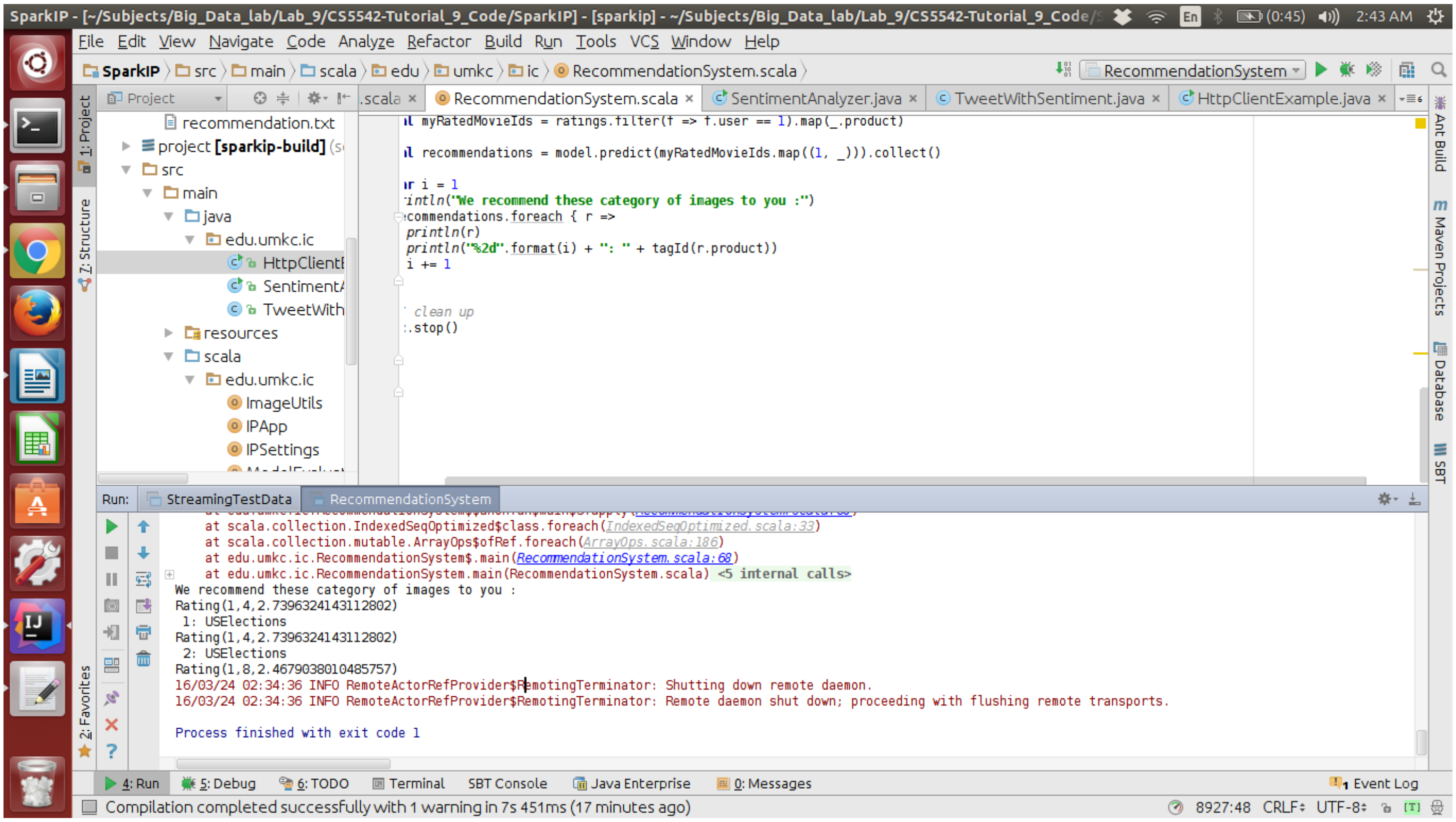
```
    at java.net.Socket.<init>(Socket.java:208)
    at org.apache.spark.streaming.dstream.SocketReceiver.receive(SocketInputDStream.scala:73)
    at org.apache.spark.streaming.dstream.SocketReceiver$$anon$2.run(SocketInputDStream.scala:59)

0.0 1.0 2.0 3.0 4.0
16/03/24 00:02:26 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
16/03/24 00:02:26 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeRefBLAS
The predicted image category is  : sport
()
-------------------------------------------
Time: 1458795740000 ms
-------------------------------------------

-------------------------------------------
Time: 1458795742000 ms
```

Task 2:

The task here was to create a recommendation system that can recommend category of images to the user based on his image from Instagram. The following steps were performed to achieve the given task.

1. Create a server side socket that can get Instagram data based on a random tag that as previously used in the classification process.
2. Get Image and caption from the user.
3. Assign User Id based on the first letter of the user.
4. Assign category id to each of the category selected.
5. Train the model with data collected from Instagram.
6. Then read data from the socket stream of Instagram to get one tag based row of data.
7. For the caption part of the Instagram data calculate the sentiment using NLP sentiment analyzer.
8. Then perform the testing on the data with the format UserId, TagId, SentimentRating
9. This would give the recommendations for the user based on his TagID and SentimentRating.

   PFB the result for the recommendation system.



SparkIP - [~/Subjects/Big_Data_lab/Lab_9/CS5542-Tutorial_9_Code/SparkIP] - [sparkip] - ~/Subjects/Big_Data_lab/Lab_9/CS5542-Tutorial_9_Code/

```
il myRatedMovieIds = ratings.filter(f => f.user == 1).map(_.product)

il recommendations = model.predict(myRatedMovieIds.map((1, _))).collect()

ir i = 1
intln("We recommend these category of images to you :")
commendations.foreach { r =>
 println(r)
 println("%2d".format(i) + ": " + tagId(r.product))
 i += 1

' clean up
:.stop()
```

```
    at scala.collection.IndexedSeqOptimized$class.foreach(IndexedSeqOptimized.scala:33)
    at scala.collection.mutable.ArrayOps$ofRef.foreach(ArrayOps.scala:186)
    at edu.umkc.ic.RecommendationSystem$.main(RecommendationSystem.scala:68)
    at edu.umkc.ic.RecommendationSystem.main(RecommendationSystem.scala) <5 internal calls>
We recommend these category of images to you :
Rating(1,4,2.7396324143112802)
 1: USElections
Rating(1,4,2.7396324143112802)
 2: USElections
Rating(1,8,2.4679038010485757)
16/03/24 02:34:36 INFO RemoteActorRefProvider$RemotingTerminator: Shutting down remote daemon.
16/03/24 02:34:36 INFO RemoteActorRefProvider$RemotingTerminator: Remote daemon shut down; proceeding with flushing remote transports.

Process finished with exit code 1
```

Compilation completed successfully with 1 warning in 7s 451ms (17 minutes ago)