

Question 1 and 3

Sentiment analysis using twitter streaming and Trend Notification to smart phone.

The screenshot shows the IntelliJ IDEA 15.0.1 interface. The main editor displays the `TwitterSentimentalAnalysis.java` file. The code defines a `TwitterSentimentalAnalysis` class with a `search(String keyword)` method. This method uses the Twitter4J library to search for tweets containing the keyword, filters out retweets, links, replies, and images, and returns a list of statuses. The `Run` console at the bottom shows the execution output, including log messages and tweet analysis results. The output indicates that the application successfully processed tweets and classified their sentiment as negative or positive.

```

public class TwitterSentimentalAnalysis {
    public List<Status> search(String keyword) {
        ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true).setOAuthConsumerKey("c0QRcGriM5HCbyhbE8HDO2RW5")
        .setOAuthConsumerSecret("41ZbQLsAI6V8HCFX0RFSFmSWiPaAYqaaEomfTyte598FsrU")
        .setOAuthAccessToken("4565837185-yKlZgmTW3Es2r4po5bRgpp53pipeTjv58Q4")
        .setOAuthAccessTokenSecret("MsR84cLkRkFBgjcnalH6waE6on5OR7DeoYPr6H6RpOjR0");
        TwitterFactory tf = new TwitterFactory(cb.build());
        Twitter twitter = tf.getInstance();
        Query query = new Query(keyword + " -filter:retweets -filter:links -filter:replies -filter:images");
        query.setCount(50);
        query.setLocale("en");
        query.setLang("en");
        try {
            QueryResult queryResult = twitter.search(query);

```

Run TwitterSentimentalAnalysis

```

"C:\Program Files\Java\jdk8\bin\java" ...
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/1.2faq.html#noconfig for more info.
Adding annotator tokenize
Adding annotator split
Adding annotator parse
Loading parser from serialized file edu.stanford.nlp.models.lexparser/englishPCFG.ser.gz ... done [4.6 sec].
Adding annotator sentiment
TweetWithSentiment: [line=Steph curry is a robot. I'm convinced he isn't real, cssClass=sentiment : negative]
Adding annotator tokenize
Adding annotator split
Adding annotator parse
Adding annotator sentiment
TweetWithSentiment: [line=Someone please come get this man @StephenCurry30 this man is for real a robot #WatchingGreatness, cssClass=sentiment : positive]
Adding annotator tokenize
Adding annotator split
Adding annotator parse

```

Compilation completed successfully in 11s 505ms (a minute ago)

The screenshot shows the IntelliJ IDEA 15.0.1 interface. The main editor displays the `TwitterSentimentalAnalysis.java` file. The code defines a `TwitterSentimentalAnalysis` class with a `search(String keyword)` method. This method uses the Twitter4J library to search for tweets containing the keyword, filters out retweets, links, replies, and images, and returns a list of statuses. The `Run` console at the bottom shows the execution output, including log messages and tweet analysis results. The output indicates that the application successfully processed tweets and classified their sentiment as negative or positive.

```

import twitter4j.conf.ConfigurationBuilder;
import java.util.Collections;
import java.util.List;

/**
 * Created by Ting on 3/7/16.
 */

public class TwitterSentimentalAnalysis {
    public List<Status> search(String keyword) {
        ConfigurationBuilder cb = new ConfigurationBuilder();
        cb.setDebugEnabled(true).setOAuthConsumerKey("R2v2M4McF7UGip1fRc1K0Yj71")
        .setOAuthConsumerSecret("InkVx1JfU5aJP0yA17GzGkS9uzF5vDnRY9HqsR9m4v25Et3sW2d")
        .setOAuthAccessToken("3630687739-9y2qv6YKXmpeApmg09DKDuYosm2piadUy8aa96n")
        .setOAuthAccessTokenSecret("IBj0d218TbaXomJ13jy2A0hOPaYzCYHmNRxRhLJLJong");
        TwitterFactory tf = new TwitterFactory(cb.build());
        Twitter twitter = tf.getInstance();

```

Run TwitterSentimentalAnalysis

```

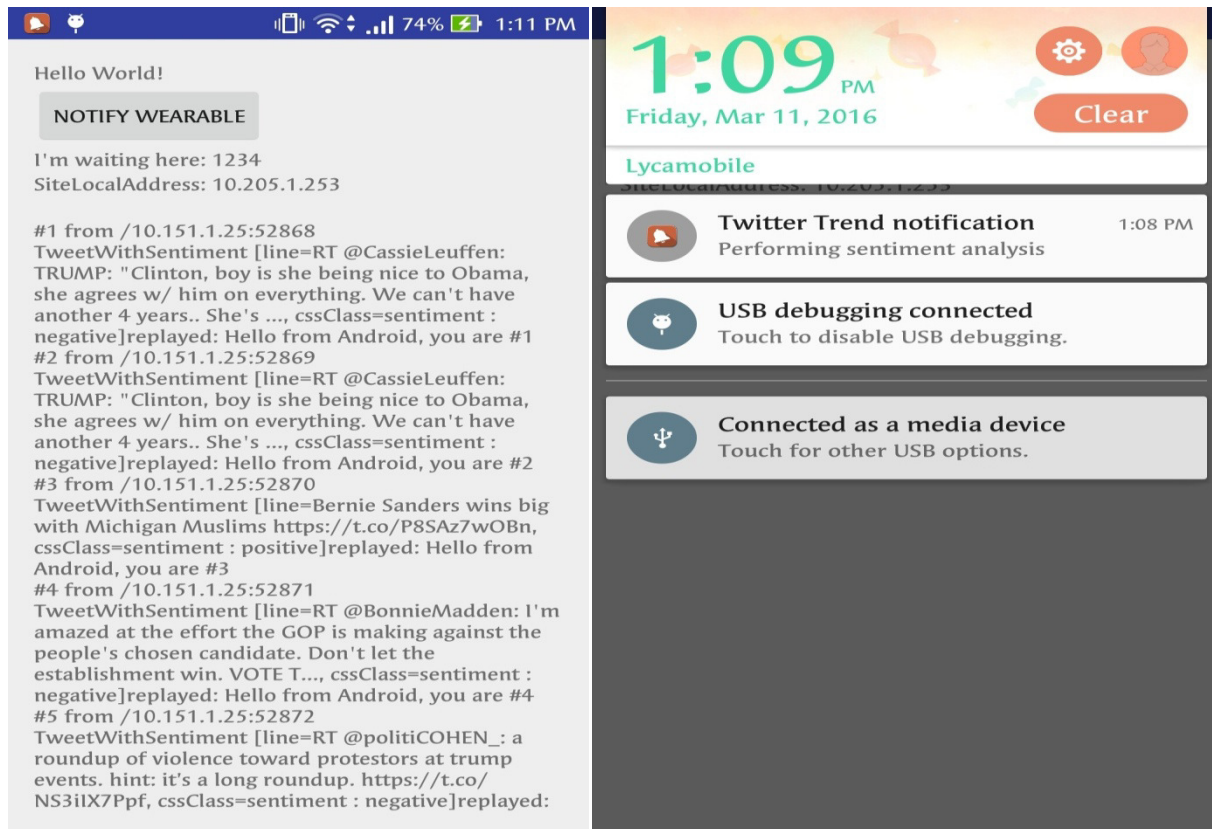
Adding annotator parse
Adding annotator sentiment
TweetWithSentiment: [line=TONIGHT Playhouse District Block Party with food trucks, live music wine 6-11pm., cssClass=sentiment : positive]
Adding annotator tokenize
Adding annotator split
Adding annotator parse
Adding annotator sentiment
Adding annotator tokenize
Adding annotator split
Adding annotator parse
Adding annotator sentiment
TweetWithSentiment: [line=When we sin, we wouldn't want Allah SWT to immediately punish.

But when we make dua, we want its fruits immediately.

Food for thought., cssClass=sentiment : negative]
Adding annotator tokenize

```

Compilation completed successfully in 29s 638ms (a minute ago)



Question 2 and 4

Make recommendations

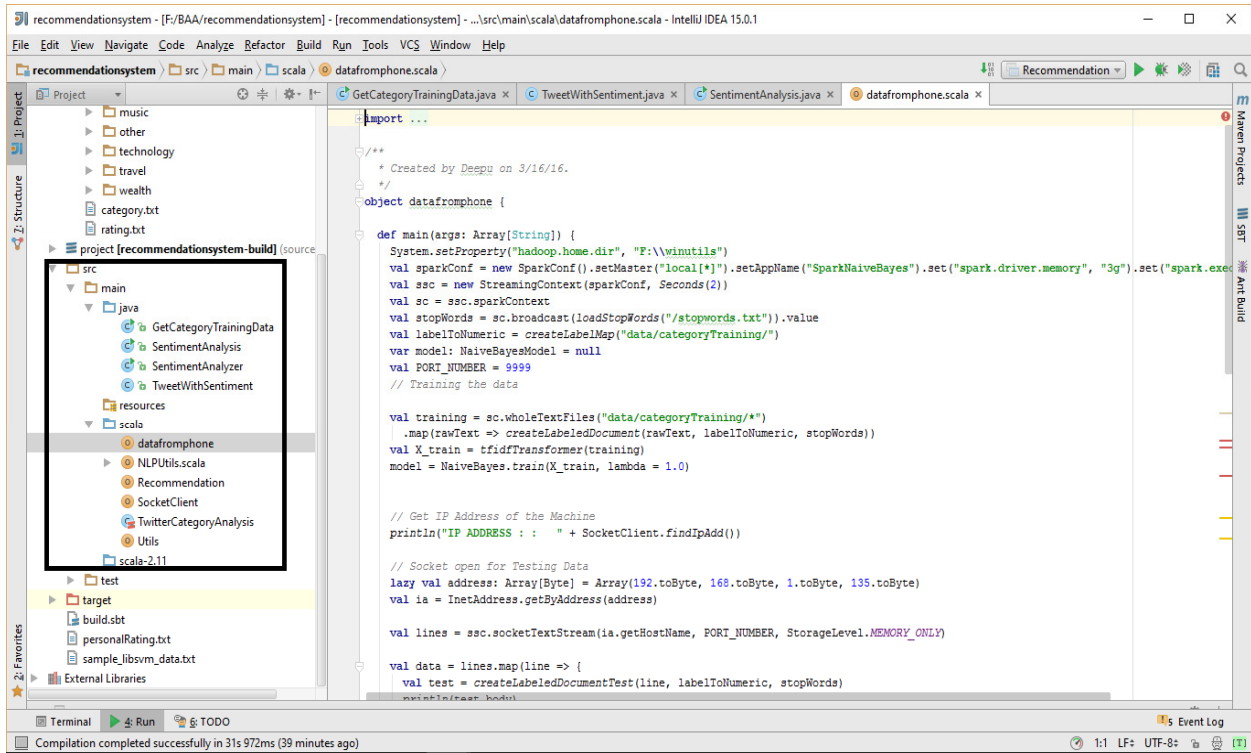
- Training Datasets: Twitter Streaming/categorized data (The categorization here would be from your previous lab 5&6).
- Testing Datasets e.g., UserId, Category, Rating (Twitter Streaming & Smart device data)
- The rating based on sentiment analysis, retweet count would be interesting.
- Expected outcome is to make a recommendation based on user profile (e.g., preferences, location, gender, age)

And notification to smart phone about the recommended category of tweets to a particular user(user id based).

Description:

- With GetCateroryTrainingData.java we collected tweets based on keywords like music, movies, technology etc which are categorized into 10 categories as shown in screenshot below. And each category consists of 100 tweets, with each tweet in a separate text file.
- Now with SentimentAnalysis.java we collected tweets for test dataset and for each tweet we got -> (user id, category and rating). The rating is based on sentiment analysis. And these features are appended to rating.txt file.
- The data is collected from smart phone too. The features(user id 12345,category,rating) gets appended to rating.txt as shown in screenshot below.

4. Next we have got category.txt with category id and category name.
5. Now finally the recommendation is done to the user(user id) about the recommended category of tweets.



```
public static void main(String[] args) throws IOException {
    SentimentAnalysis twitterSentimentalAnalysis = new SentimentAnalysis();
    List<Status> statuses = twitterSentimentalAnalysis.getTestingData();
    String a = "";
    int i = 0;
    for (Status status : statuses) {
        if (status.getText() != null) {
            i++;
            File newTextFile = new File("data/testing/1.txt");

            FileWriter fw = new FileWriter(newTextFile);
            fw.write(status.getText());
            fw.close();

            SentimentAnalyzer doAnalysis = new SentimentAnalyzer();

            int rate = doAnalysis.findSentiment(status.getText()).getRate();

            TwitterCategoryAnalysis twitterCategoryAnalysis = new TwitterCategoryAnalysis();
            int category = twitterCategoryAnalysis.CategoryAnalysis();

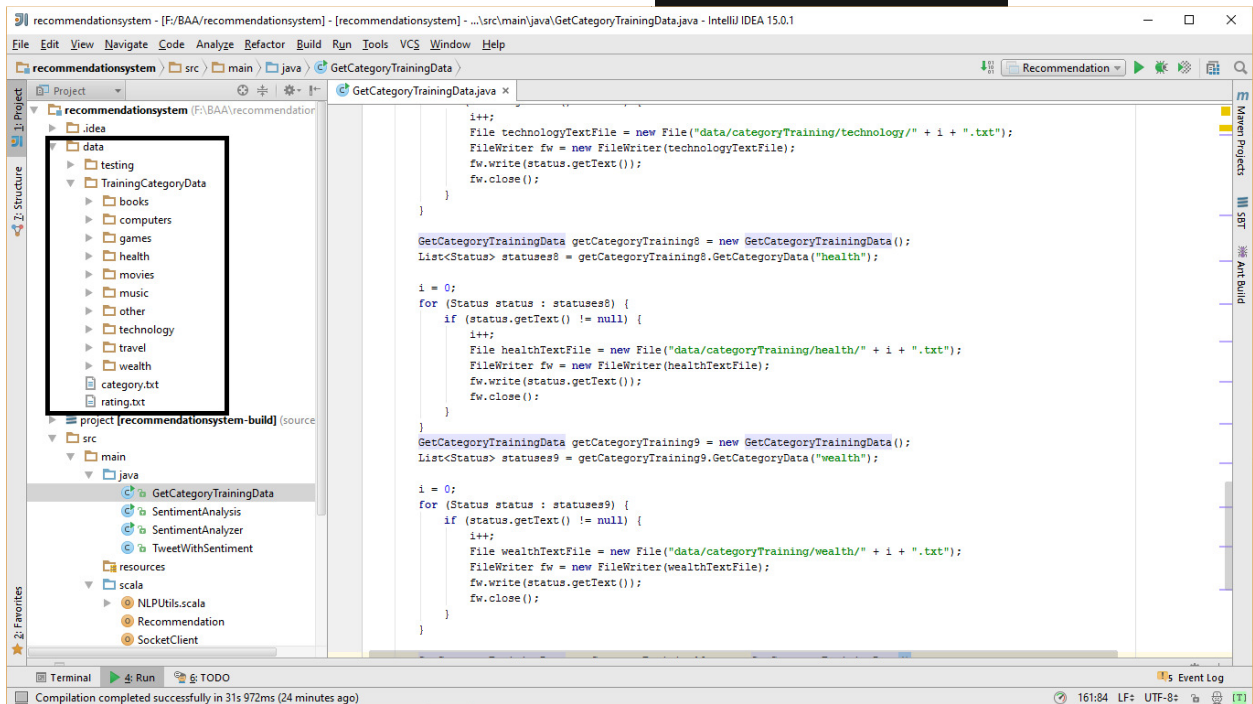
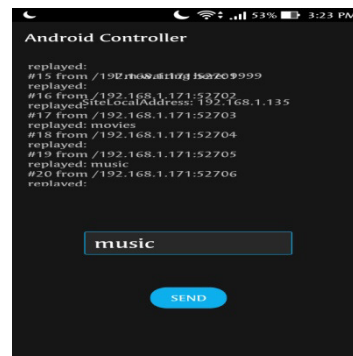
            int userId = (int)((status.getId() >>> 32) ^ status.getId());
            int time = (int)((status.getCreatedAt().getTime() >>> 32) ^ status.getCreatedAt().getTime());
            a += userId + "::-" + Integer.toString(category) + "::-" + Integer.toString(rate) + "::-" + time + "\n";
            System.out.println(a);
        }
    }
}
```

Category mapping file:

```
2::books
3::computers
4::games
5::health
6::movies
7::music
8::other
9::technology
10::travel
11::wealth
```

Data added to rating.txt with user id 12345(smart phone data)

```
1787078521::2::5::1596646489
1547995001::2::5::1596646489
1254360952::8::2::1596645459
1048856442::2::2::1596645452
12345::2::0::1458332570
12345::2::0::1458332572
12345::2::0::1458332581
12345::2::0::1458332585
12345::2::0::1458332586
12345::2::0::1458332599
12345::2::0::1458332624
12345::2::0::1458332628
```




```

// make personalized recommendations
val myRatedCategoryIds = myRatings.map(_._product).toSet
val candidates = sc.parallelize(categories.keys.filter(!myRatedCategoryIds.contains(_)).toSeq)
val recommendations = bestModel.get
    .predict(candidates.map((1090812500, _)))
    .collect()
    .sortBy(_._rating)
    .take(50)

var i = 1
println("Categories recommended for you:")

recommendations.foreach { r =>

```

Run Recommendation

```

16/03/18 13:51:06 WARN BLAS: Failed to load implementation from: com.github.fommil.netlib.NativeSystemBLAS
[Stage 25:]
16/03/18 13:51:08 WARN LAPACK: Failed to load implementation from: com.github.fommil.netlib.NativeSystemLAPACK
16/03/18 13:51:08 INFO UnLoader: already loaded netlib-native_ref-win-x86_64.dll
RMSE (validation) = 0.16449933842172285 for the model trained with rank = 8, lambda = 0.1, and numIter = 10.
RMSE (validation) = 0.17834359678311762 for the model trained with rank = 8, lambda = 0.1, and numIter = 20.
RMSE (validation) = 3.692744729379982 for the model trained with rank = 8, lambda = 10.0, and numIter = 10.
RMSE (validation) = 3.692744729379982 for the model trained with rank = 8, lambda = 10.0, and numIter = 20.
RMSE (validation) = 0.15610231352078593 for the model trained with rank = 12, lambda = 0.1, and numIter = 10.
RMSE (validation) = 0.17817593603738776 for the model trained with rank = 12, lambda = 0.1, and numIter = 20.
RMSE (validation) = 3.692744729379982 for the model trained with rank = 12, lambda = 10.0, and numIter = 10.
RMSE (validation) = 3.692744729379982 for the model trained with rank = 12, lambda = 10.0, and numIter = 20.
The best model was trained with rank = 12 and lambda = 0.1, and numIter = 10, and its RMSE on the test set is 1.8784802537528067.
The best model improves the baseline by -67.63%.
Categories recommended for you:
1: computers
2: other
3: travel

```

Compilation completed successfully in 31s 972ms (17 minutes ago)

57% 1:53 PM	57% 1:46 PM
<p>Hello World!</p> <p>NOTIFY WEARABLE</p> <p>I'm waiting here: 1234 SiteLocalAddress: 192.168.1.135</p> <p>#1 from /192.168.1.171:52014 Categories recommended for you:User Id: 1090812500 2: computers replayed: Hello from Android, you are #1 #2 from /192.168.1.171:52015 Categories recommended for you:User Id: 1090812500 3: other replayed: Hello from Android, you are #2 #3 from /192.168.1.171:52016 Categories recommended for you:User Id: 1090812500 4: travel replayed: Hello from Android, you are #3</p>	<p>Hello World!</p> <p>NOTIFY WEARABLE</p> <p>I'm waiting here: 1234 SiteLocalAddress: 192.168.1.135</p> <p>#1 from /192.168.1.171:51890 Categories recommended for you:User Id: 1363512652 2: other replayed: Hello from Android, you are #1 #2 from /192.168.1.171:51891 Categories recommended for you:User Id: 1363512652 3: computers replayed: Hello from Android, you are #2 #3 from /192.168.1.171:51892 Categories recommended for you:User Id: 1363512652 4: travel replayed: Hello from Android, you are #3</p>