Increment 4

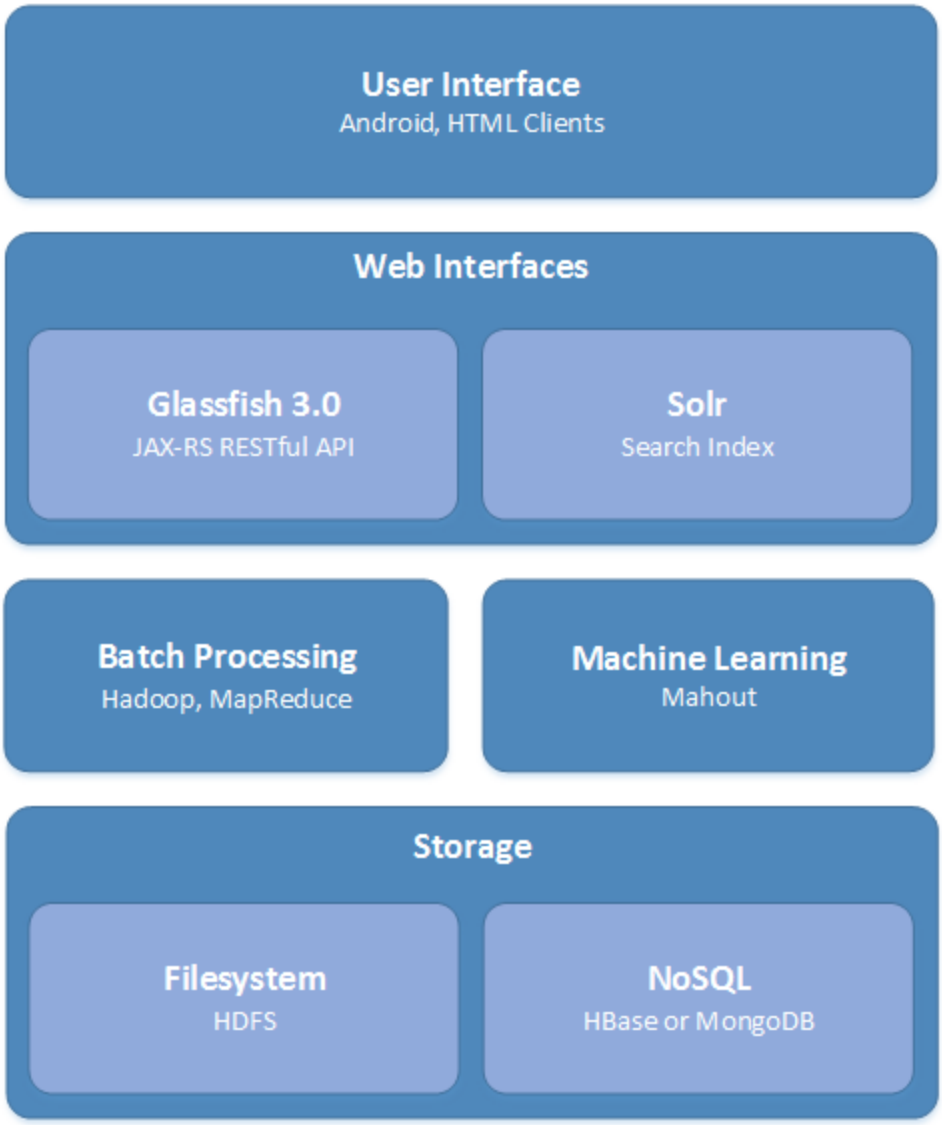
Group 6

James Clark
Tirumala Konireddy
Jagadish Tirumalasetty
Wang Zhang

University of Missouri - Kansas City

CS 560: Knowledge Discovery and Management
Dr. Yugyung Lee
May 2, 2014

System Architecture Diagram



Overall Process

1. supervised learning
2. manual data inspection and classification
3. most general classification possible
4. apply classification to entire organization
5. reinspect
6. reclassify, ex: (mislabeled, contamination) > (salmonella, child, pet)

Training the data

Data is extracted from digital gov. It's collected and pre-processed to TSV. Now, using Excel, I can copy rows into a table and manually categorize them. The benefit of Excel is that copy and paste results in an automatically tab separated values format, immediately compatible for training Mahout.

This is called assisted learning where a human interacts with computer, monitoring the application of algorithms, accuracy of results, and correspondingly approves or declines before arranging the data for production release.

Tutorial 7 covers classification of twitter data. The same TwitterTSVToSeq class can classify our files for loading into Hadoop.

Generally, this is an outline of the process. The actual source code for this outline is coded in the appendix.

1. Convert training set to Hadoop sequence file format
2. Upload sequence file to HDFS
3. Transform training sets to vectors with Mahout (using term frequency by document frequency weights)
4. Split data into training set and testing set
5. Use training set to train classifier
6. Test the classifier on the training set
7. Test the classifier on the testing set
8. To use classifier and classify new documents, copy several files from HDFS
9. Classify the new file (optional section in bold red writes to text file)

The final output document is printed in a uniform set of 2-line data objects following this format. Note the items in bold are relevant to our purpose of classification:

Tweet: 3d13ad94d4 Nutriom LLC, a Lacey, Wash. establishment, is recalling an additional 82,884 pounds of processed egg products that may be contaminated with Salmonella Washington Firm Recalls Dried Egg Products Due To Possible Salmonella Contamination
contamination: -273.64178143939307 mislabeled: -325.0381960862568 **=> contamination**

Classification Results

FDA

2 labels: contamination, mislabeled
5 labels: allergen, animal, debris, illness, medecine
Test the classifier on the training set
Correctly Classified Instances : 149 98.6755%

Incorrectly Classified Instances : 2 1.3245%
Test the classifier on the testing set
Correctly Classified Instances : 100 83.3333%
Incorrectly Classified Instances : 20 16.6667%

Testing the training data is between 92% and 100% accurate
Testing the testing data is approximately 85% accurate

Observed Data Anomalies

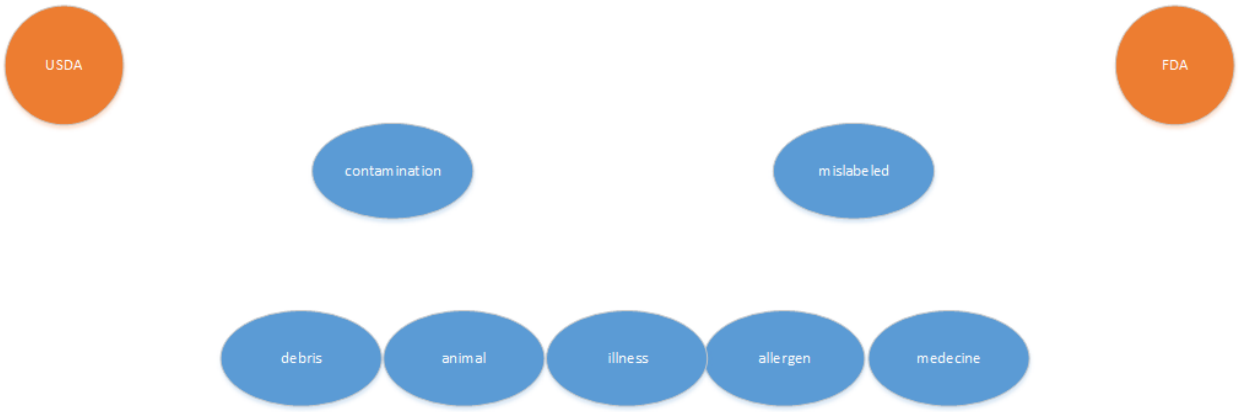
FDA recall_id cb7b35d997, 6266a8f724, 5f2fc28def, 8758f480f8 are a examples of voluntary recall with no expanation as to why. Categorized as contamination.

Moving Data into Solr

The challenge with the UMKC managed Solr instance is a lack of flexibility to update fields for our application. While technically feasible, it would be high-risk to assume any changes on the production server this close to final presentations. A critical mistake would be a massive takedown for all the groups relying on Solr for their presentation and general app functionality. Therefore, we are left mapping our data to the available fields in Solr. This is a list of fields that we can use in our application.

FDA & USDA

Solr Field	**Class Field**
id	recall_number
name	organization
features[]	
title[]	
subject	
description	description
comments	summary
author	
keywords	classification2
category	organization
resourcename	classification1
url	recall_url
last_modified	recall_date



NHTSA

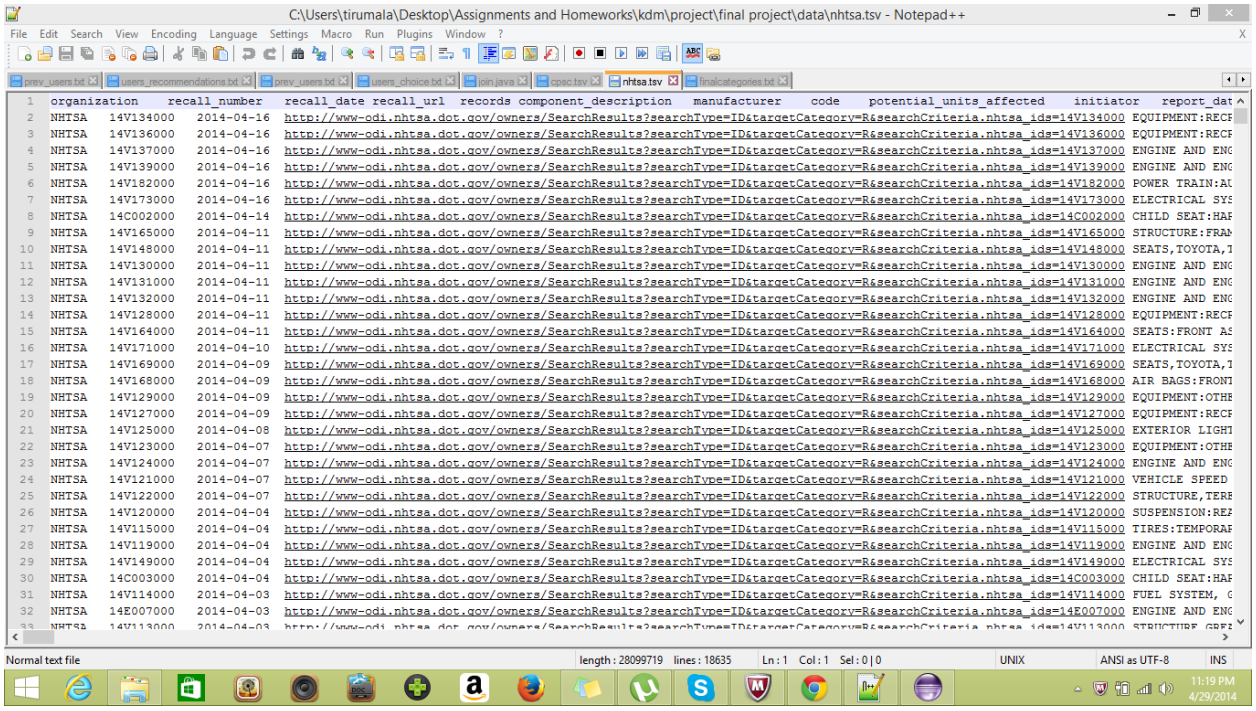
Solr Field	Class Field
id	recall_number
name	organization
features[]	
title[]	
subject	recall_subject
description	consequence_summary, defect_summary, corrective_summary
comments	notes
author	component_description
keywords	manufacturer
category	organization
resourcename	classification1
url	recall_url
last_modified	recall_date
popularity	potential_units_affected

Now, These results are available to retrieve from solr server through android application.

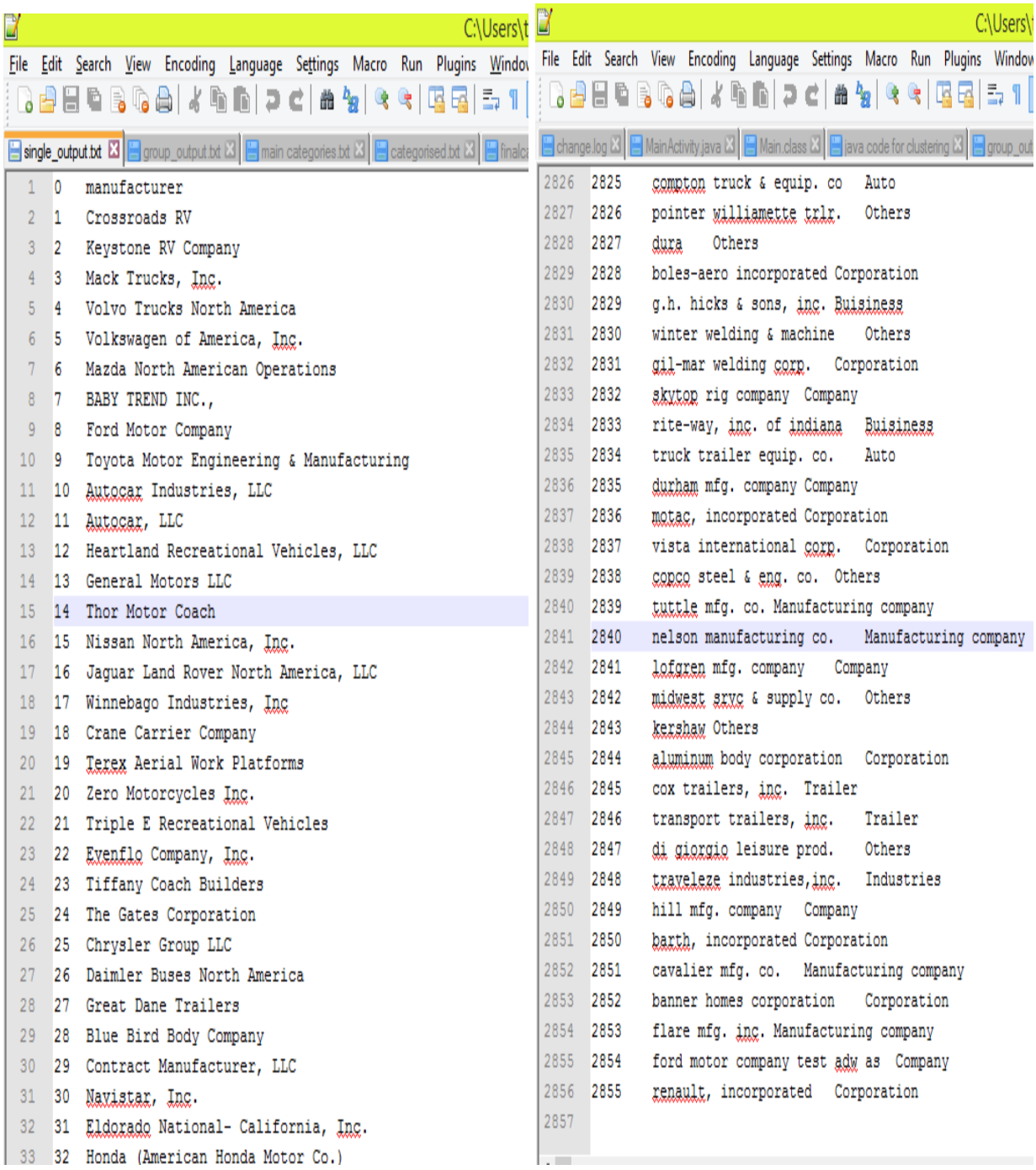
Classification For Nhtsa Data

Raw Data

The data collected from National Highway Transport Safety Administration API is as below.



The nhtsa data file contains 2856 recall categories regarding Cars, Buses, trailers, childcare etc.,



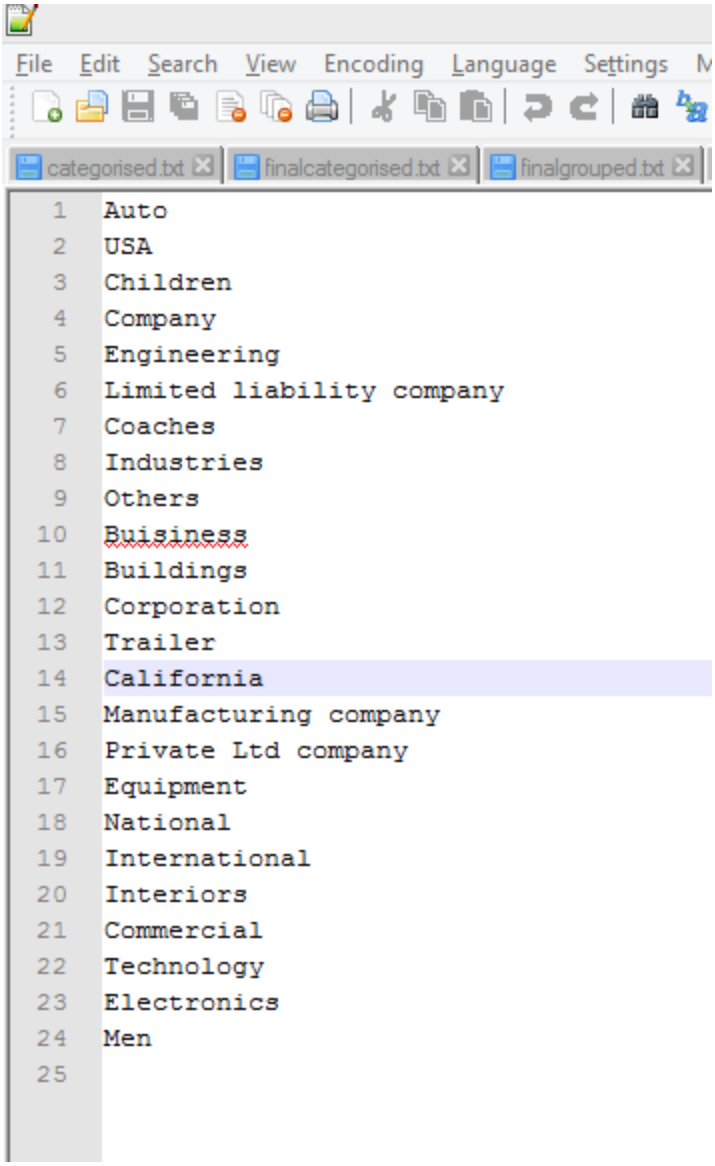
single_output.txt

```
0  manufacturer
1  Crossroads RV
2  Keystone RV Company
3  Mack Trucks, Inc.
4  Volvo Trucks North America
5  Volkswagen of America, Inc.
6  Mazda North American Operations
7  BABY TREND INC.,
8  Ford Motor Company
9  Toyota Motor Engineering & Manufacturing
10 Autocar Industries, LLC
11 Autocar, LLC
12 Heartland Recreational Vehicles, LLC
13 General Motors LLC
14 Thor Motor Coach
15 Nissan North America, Inc.
16 Jaguar Land Rover North America, LLC
17 Winnebago Industries, Inc
18 Crane Carrier Company
19 Terex Aerial Work Platforms
20 Zero Motorcycles Inc.
21 Triple E Recreational Vehicles
22 Evenflo Company, Inc.
23 Tiffany Coach Builders
24 The Gates Corporation
25 Chrysler Group LLC
26 Daimler Buses North America
27 Great Dane Trailers
28 Blue Bird Body Company
29 Contract Manufacturer, LLC
30 Navistar, Inc.
31 Eldorado National- California, Inc.
32 Honda (American Honda Motor Co.)
```

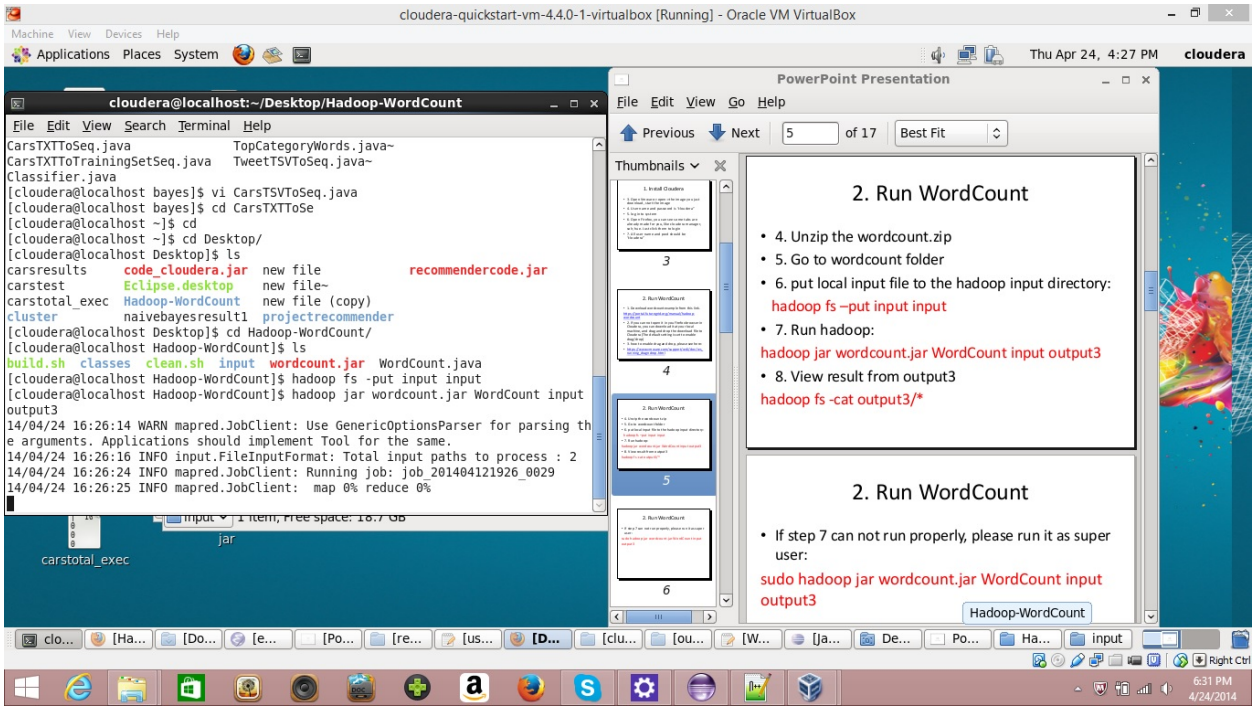
```
2825 compton truck & equip. co   Auto
2826 pointer williamette trlr.   Others
2827 dura    Others
2828 boles-aero incorporated Corporation
2829 g.h. hicks & sons, inc. Buisiness
2830 winter welding & machine    Others
2831 gil-mar welding corp.   Corporation
2832 skytop rig company  Company
2833 rite-way, inc. of indiana   Buisiness
2834 truck trailer equip. co.    Auto
2835 durham mfg. company Company
2836 motac, incorporated Corporation
2837 vista international corp.   Corporation
2838 copco steel & eng. co.  Others
2839 tuttle mfg. co. Manufacturing company
2840 nelson manufacturing co.    Manufacturing company
2841 lofgren mfg. company    Company
2842 midwest srvc & supply co.   Others
2843 kershaw Others
2844 aluminum body corporation   Corporation
2845 cox trailers, inc.  Trailer
2846 transport trailers, inc.    Trailer
2847 di giorgio leisure prod.    Others
2848 traveleze industries,inc.   Industries
2849 hill mfg. company   Company
2850 barth, incorporated Corporation
2851 cavalier mfg. co.   Manufacturing company
2852 banner homes corporation    Corporation
2853 flare mfg. inc. Manufacturing company
2854 ford motor company test adw as  Company
2855 renault, incorporated   Corporation
```

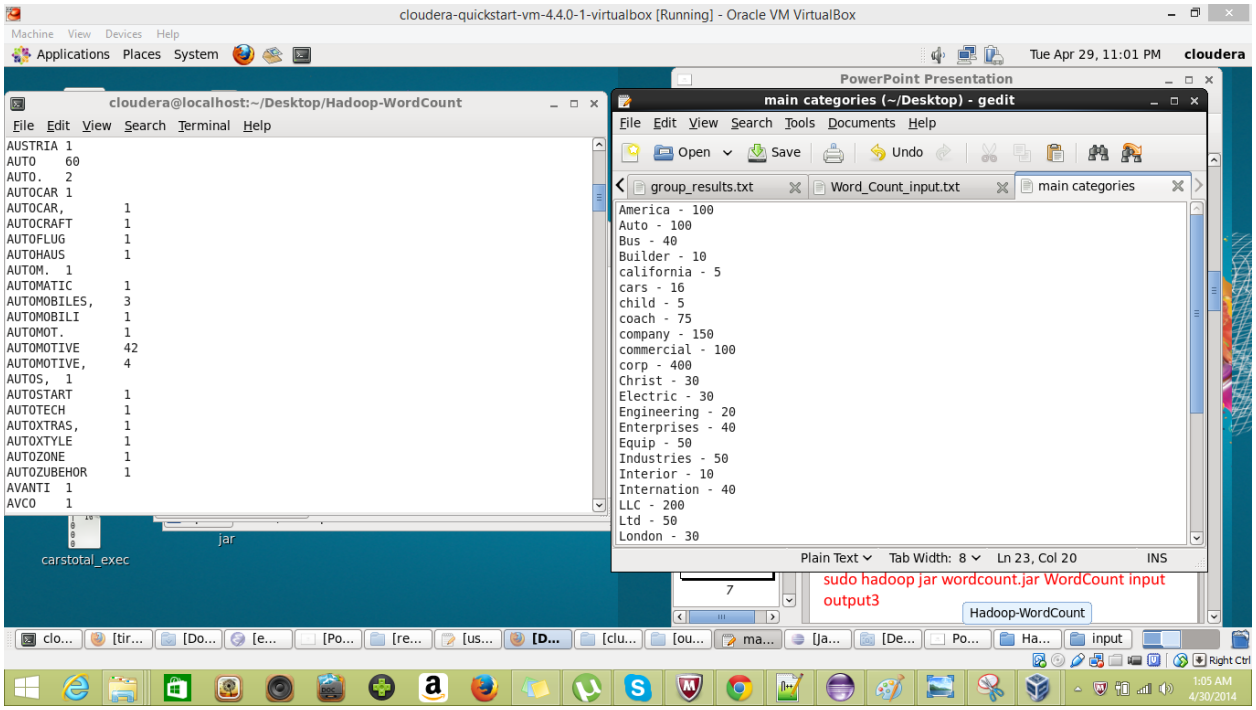
It is difficult for a user to choose a category from a list of 2856 very specific categories like above.

Then, I had grouped or classified the relevant specific categories of 2856 into 24 general categories like Auto, Childcare, Industry etc ., like below

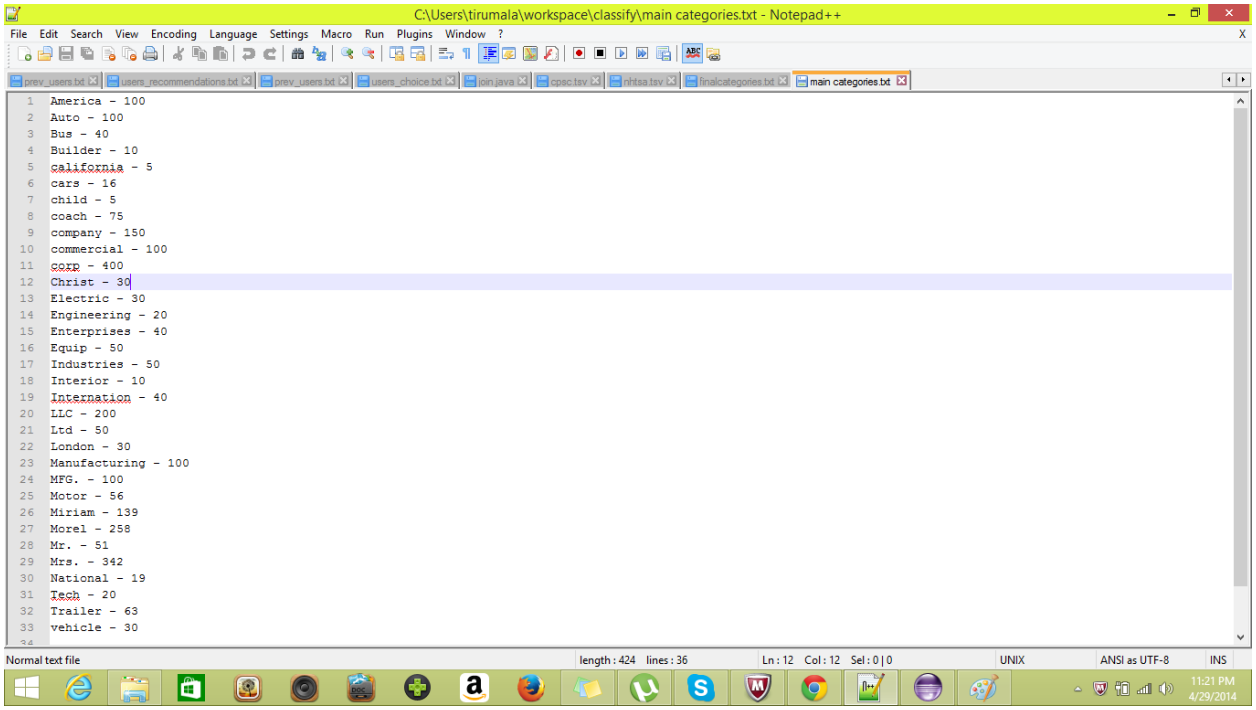


To do the classification for above results i had used a word count program to find the key words or peculiar words in the above 2856 specific categories like below to classify a specific category to a generalized category.

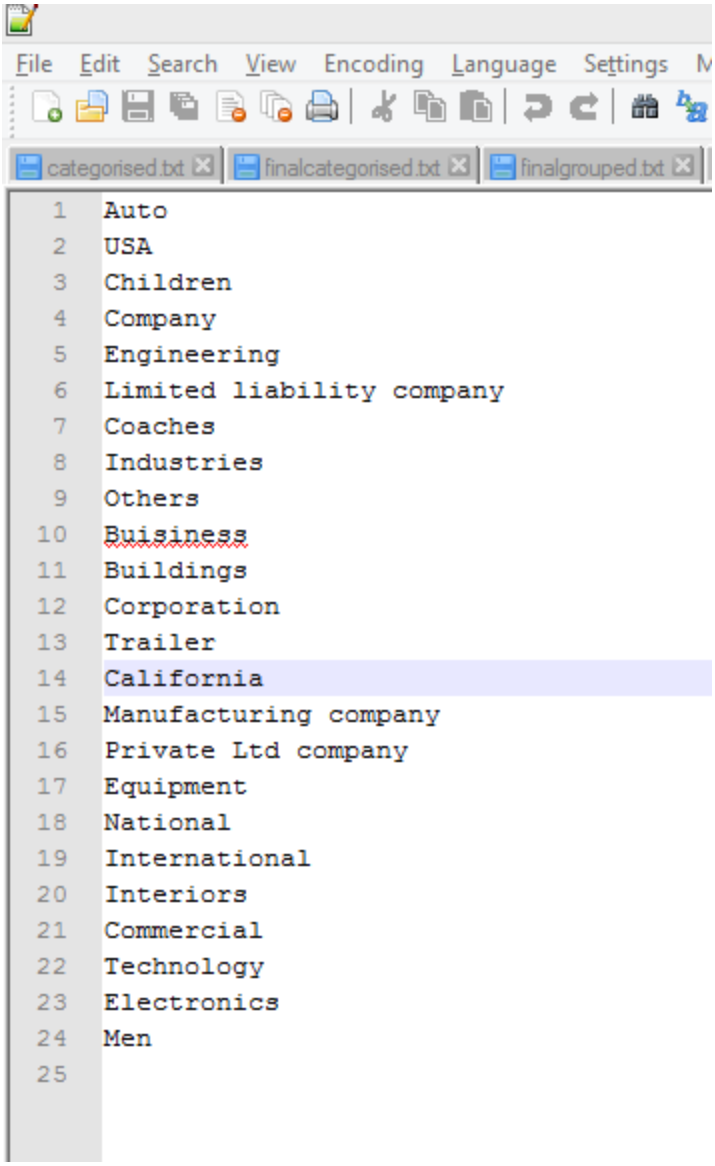

cloudera-quickstart-vm-4.4.0-1-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Thu Apr 24, 4:27 PM cloudera
cloudera@localhost:~/Desktop/Hadoop-WordCount
File Edit View Search Terminal Help
CarsTXTToSeq.java TopCategoryWords.java~
CarsTXTToTrainingSetSeq.java TweetTSVToSeq.java~
Classifier.java
[cloudera@localhost bayes]$ vi CarsTSVToSeq.java
[cloudera@localhost bayes]$ cd CarsTXTToSe
[cloudera@localhost ~]$ cd
[cloudera@localhost ~]$ cd Desktop/
[cloudera@localhost Desktop]$ ls
carsresults code_cloudera.jar new file recommendercode.jar
carstest Eclipse.desktop new file~
carstotal_exec Hadoop-WordCount new file (copy)
cluster naivebayesresult1 projectrecommender
[cloudera@localhost Desktop]$ cd Hadoop-WordCount/
[cloudera@localhost Hadoop-WordCount]$ ls
build.sh classes clean.sh input wordcount.jar WordCount.java
[cloudera@localhost Hadoop-WordCount]$ hadoop fs -put input input
[cloudera@localhost Hadoop-WordCount]$ hadoop jar wordcount.jar WordCount input output3
14/04/24 16:26:14 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Applications should implement Tool for the same.
14/04/24 16:26:16 INFO input.FileInputFormat: Total input paths to process : 2
14/04/24 16:26:24 INFO mapred.JobClient: Running job: job_201404121926_0029
14/04/24 16:26:25 INFO mapred.JobClient: map 0% reduce 0%
PowerPoint Presentation
File Edit View Go Help
Previous Next 5 of 17 Best Fit
Thumbnails
2. Run WordCount
4. Unzip the wordcount.zip
5. Go to wordcount folder
6. put local input file to the hadoop input directory:
hadoop fs –put input input
7. Run hadoop:
hadoop jar wordcount.jar WordCount input output3
8. View result from output3
hadoop fs -cat output3/*
2. Run WordCount
If step 7 can not run properly, please run it as super user:
sudo hadoop jar wordcount.jar WordCount input output3
Hadoop-WordCount
carstotal_exec
jar
input
Right Ctrl
6:31 PM
4/24/2014


cloudera-quickstart-vm-4.4.0-1-virtualbox [Running] - Oracle VM VirtualBox
Applications Places System
Tue Apr 29, 11:01 PM cloudera
cloudera@localhost:~/Desktop/Hadoop-WordCount
File Edit View Search Terminal Help
AUSTRIA 1
AUTO 60
AUTO. 2
AUTOCAR 1
AUTOCAR, 1
AUTOCRAFT 1
AUTOFLUG 1
AUTOHAUS 1
AUTOM. 1
AUTOMATIC 1
AUTOMOBILES, 3
AUTOMOBILI 1
AUTOMOT. 1
AUTOMOTIVE 42
AUTOMOTIVE, 4
AUTOS, 1
AUTOSTART 1
AUTOTECH 1
AUTOXTRAS, 1
AUTOXTYLE 1
AUTOZONE 1
AUTOZUBEHOR 1
AVANTI 1
AVCO 1
PowerPoint Presentation
main categories (~/Desktop) - gedit
File Edit View Search Tools Documents Help
Open Save Undo
group_results.txt
Word_Count_input.txt
main categories
America - 100
Auto - 100
Bus - 40
Builder - 10
california - 5
cars - 16
child - 5
coach - 75
company - 150
commercial - 100
corp - 400
Christ - 30
Electric - 30
Engineering - 20
Enterprises - 40
Equip - 50
Industries - 50
Interior - 10
Internation - 40
LLC - 200
Ltd - 50
London - 30
Plain Text Tab Width: 8 Ln 23, Col 20 INS
sudo hadoop jar wordcount.jar WordCount input output3
Hadoop-WordCount
carstotal_exec
jar
input
Right Ctrl
1:05 AM
4/30/2014

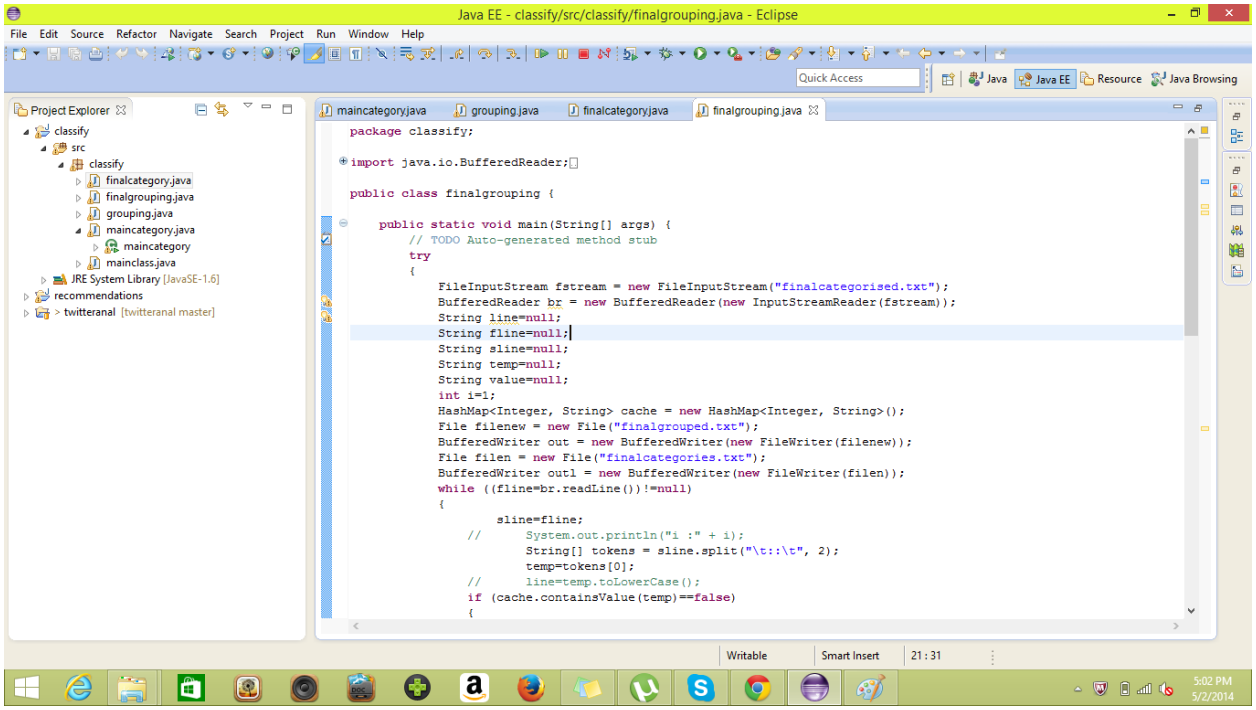
I got the word count of all words in the specific categories list.



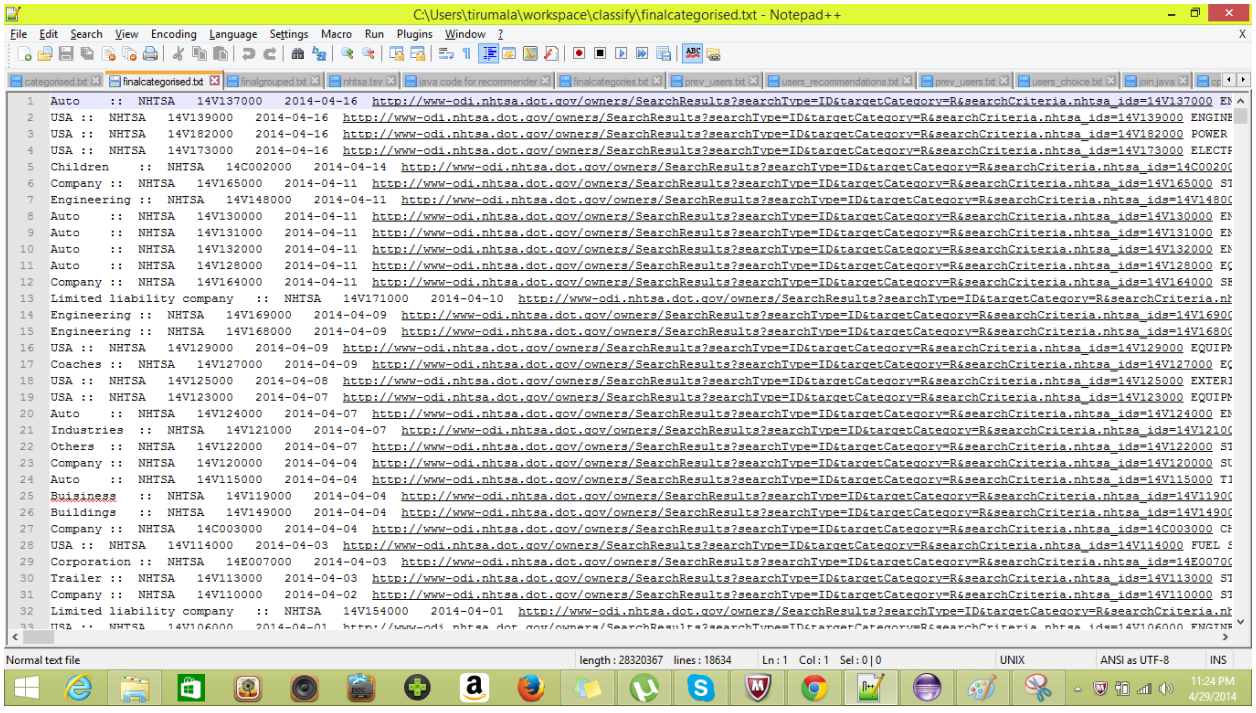
Then, i had identified some particular words those can be used to classify the specific categories into generalised categories as below. With the above words i found that i am able to classify upto 2500 specific categories into one of the generalized category. I had introduced a new category called “Others” category for those 280+ unclassified categories to limit the generalized categories to “Number 24” and those 24 categories are as below.


File Edit Search View Encoding Language Settings
categorised.txt
finalcategorised.txt
finalgrouped.txt
1 Auto
2 USA
3 Children
4 Company
5 Engineering
6 Limited liability company
7 Coaches
8 Industries
9 Others
10 Buisiness
11 Buildings
12 Corporation
13 Trailer
14 California
15 Manufacturing company
16 Private Ltd company
17 Equipment
18 National
19 International
20 Interiors
21 Commercial
22 Technology
23 Electronics
24 Men
25

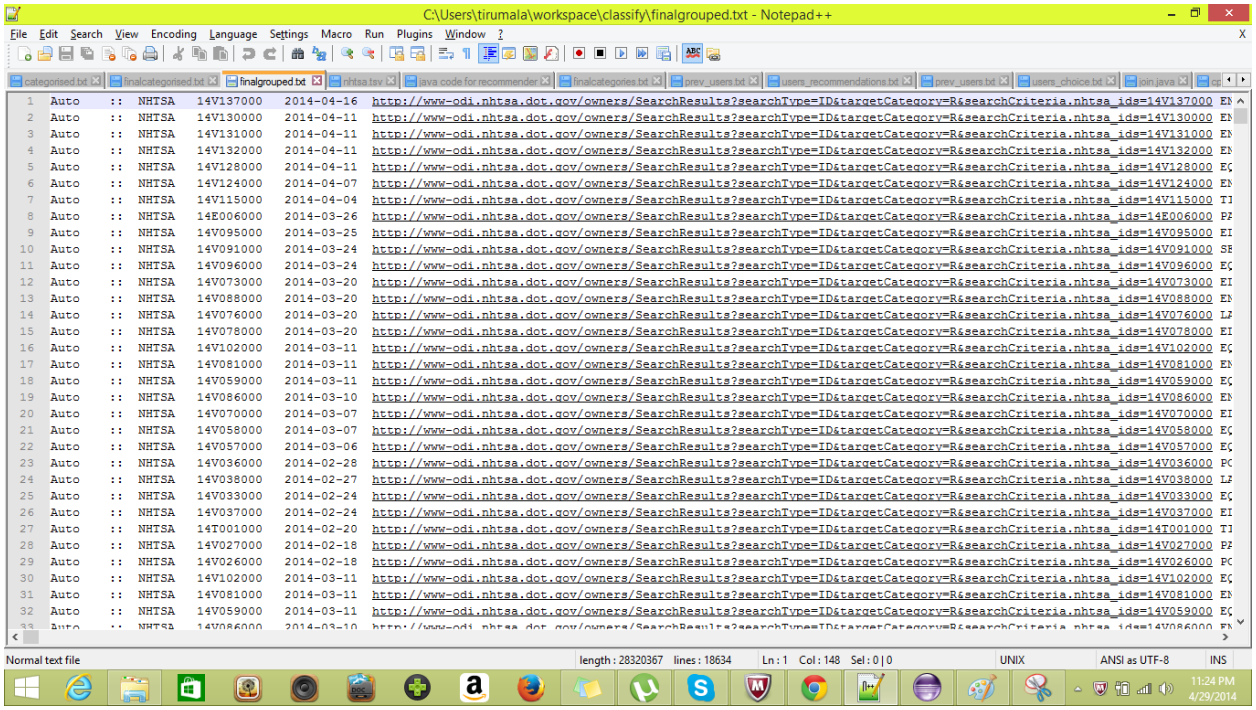
Then, I had developed a java program to classify the data entries containing specific categories into generalised categories.



I had written a new file with appending new generalized category in first column as below.



I had rewritten this file again by grouping all the entries based on the main category as below.



Then, I had uploaded this file to our group 6 solr server group6_shard1_replica1 collection with the following field names.

Recommendation Algorithm

Recommending a category is different than recommending a product. A particular product may bought more than once, whereas in the case of subscriptions for a category, subscription to a particular category is done only once. Also, we need very little complexity and less number of calculations in run time to produce instant dynamic recommendations. Keeping the above criteria in mind, I had developed a new algorithm to work efficiently for our data by considering the likeliness of both interest and disinterest on a particular category with present subscriptions of user based on previous user's subscriptions.

First, I am collecting all the categories present in the previous user's subscriptions.

Then, I am calculating the number of occurrences of every item in the whole user's data base. This number itself denotes the importance or priority of a particular item in recommending it.

Next, I am providing a index for all individual items with all other items as below.

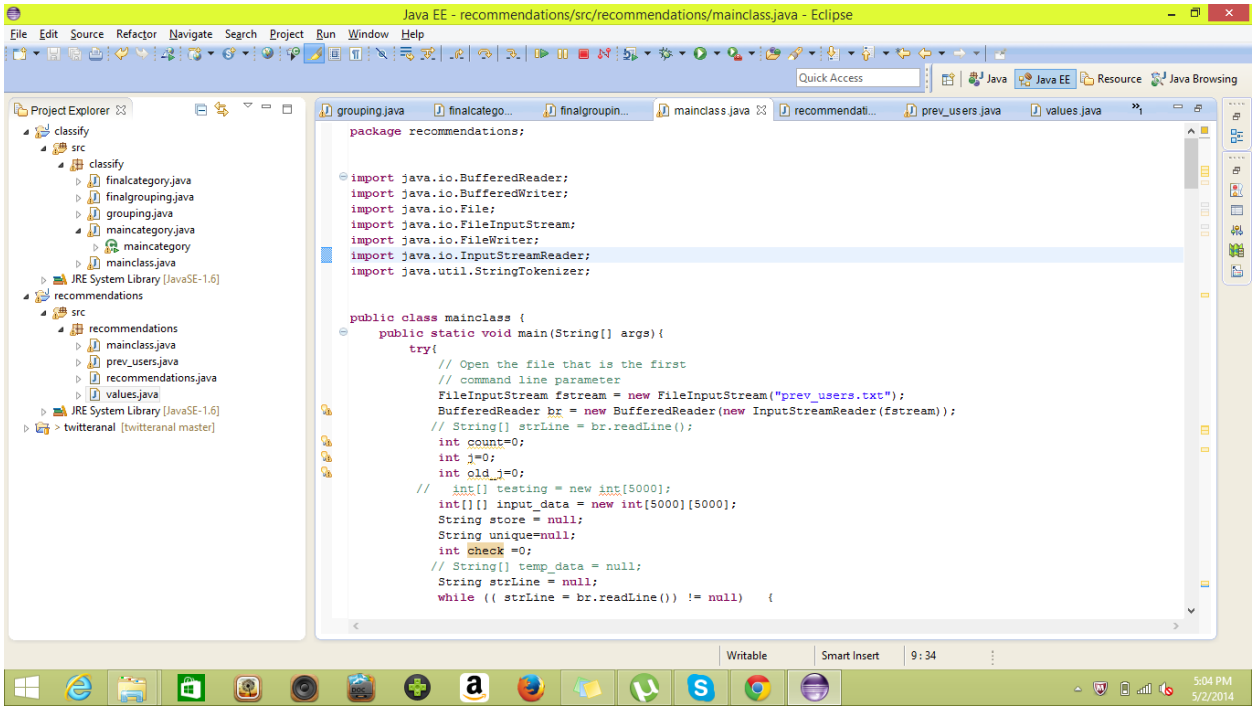
Index for a recommending item B to user who subscribed to A = (sum of number of occurrences of B along with A in previous user's subscriptions list) - (sum of number of disappearances of B when A is present in previous user's subscriptions list)

Then, We recommend a particular category in the entire list to a user subscribed for A and C which has the highest sum of indexes in A and C other than A and C.

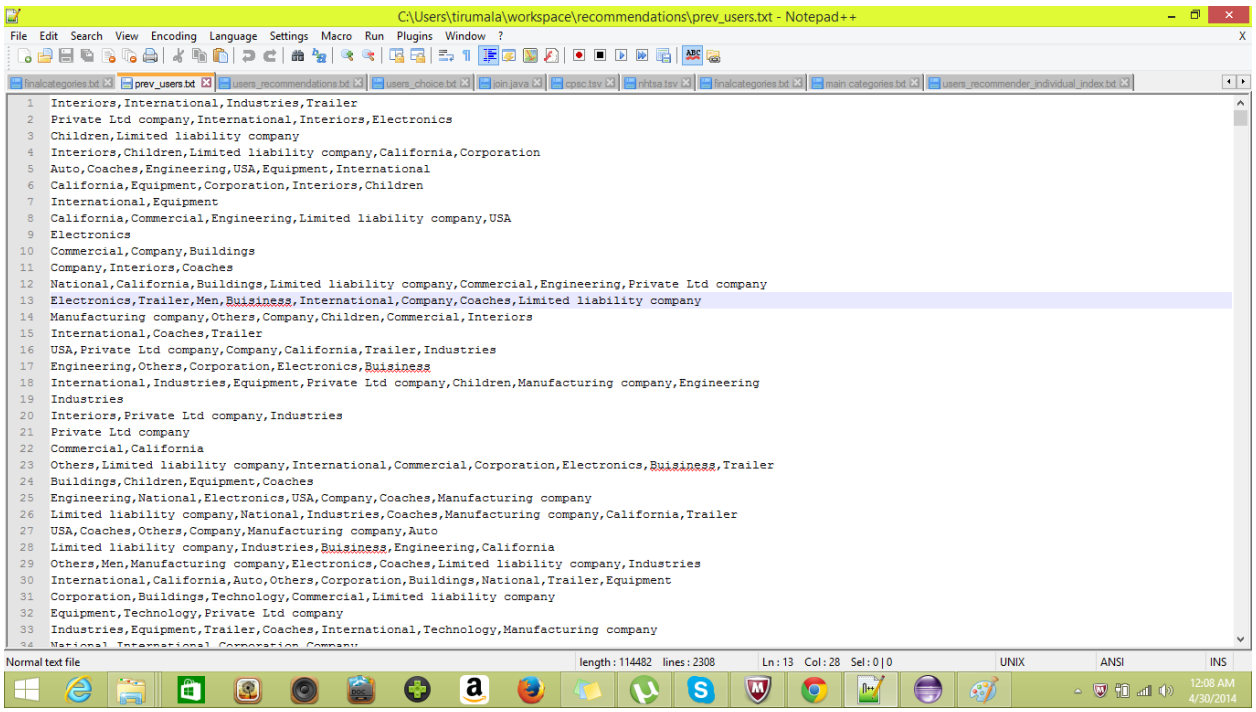
Index of recommending B for user subscribed to A and C categories is
Index B (A and C)=Index B(A) + Index B(C)
Similarly, If i had A,B,C,D,E categories in my data base, then
Index D (A and C)=Index D(A) + Index D(C)
Index E (A and C)=Index E(A) + Index E(C)

Then, I recommend the category in B, D, E which has highest Index.
Using this algorithm, we can eliminate the unlikely combinations in providing recommendations.
For Example :
In amazon.com
if, user 1 had bought a Sony Laptop and a back pack bag
user 2 had bought a Samsung Laptop and a back pack bag too
Here, recommending both Sony Laptop and Samsung Laptop is somewhat good for user who bought a back pack bag.
Recommender algorithms which uses only the appearance of a object with the common object regardless the object type. This will provide a recommendation of a Samsung laptop to a user who had bought a Sony Laptop as it is bought by a user who had bought a back pack bag.
But, it is useless to recommend a Samsung Laptop for a user immediately after buying a Sony Laptop as it is atleast 2 years investment. This type of recommendations can be avoided by calculating the number disappearances along with the number of appearances when a particular item is present.

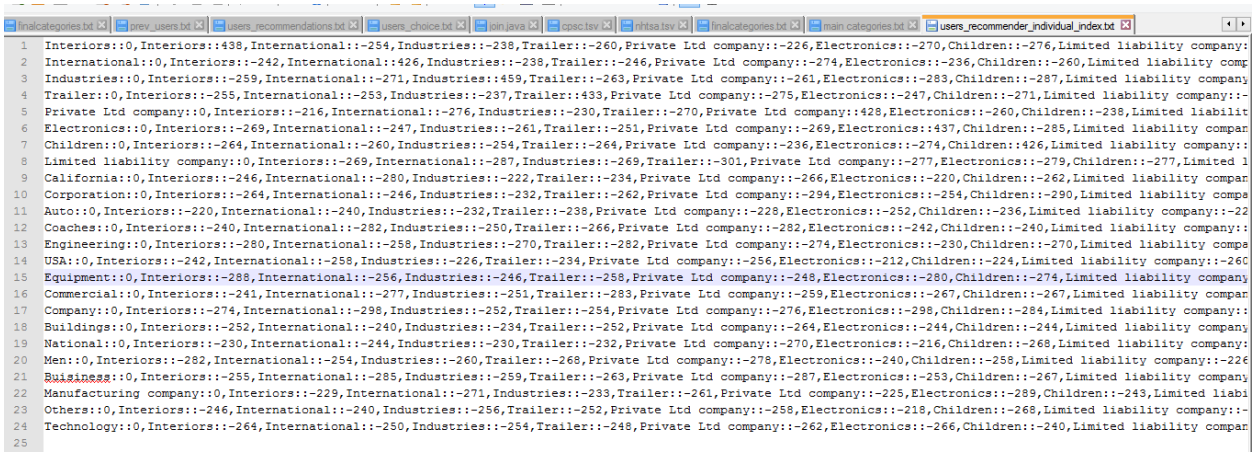
We had developed a program in java, to implement the above algorithm in providing recommendations to user.



My previous user database

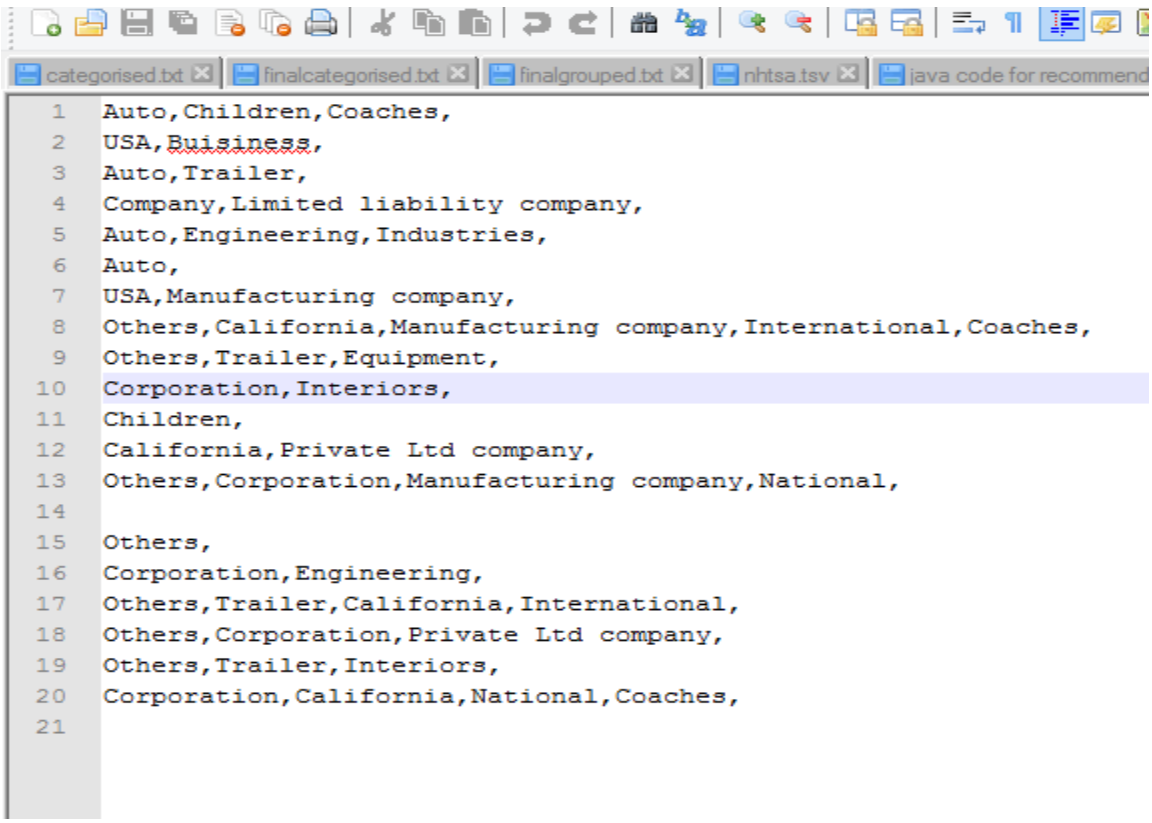


Recommendation Index table generated by our program for providing recommendations for any user subscribed with any combination of categories.



```
Interiors::0,Interiors::438,International::-254,Industries::-238,Trailer::-260,Private Ltd company::-226,Electronics::-270,Children::-276,Limited liability company:
International::0,Interiors::-242,International::426,Industries::-238,Trailer::-246,Private Ltd company::-274,Electronics::-236,Children::-260,Limited liability comp
Industries::0,Interiors::-259,International::-271,Industries::459,Trailer::-263,Private Ltd company::-261,Electronics::-283,Children::-287,Limited liability company
Trailer::0,Interiors::-255,International::-253,Industries::-237,Trailer::433,Private Ltd company::-275,Electronics::-247,Children::-271,Limited liability company::-
Private Ltd company::0,Interiors::-216,International::-276,Industries::-230,Trailer::-270,Private Ltd company::428,Electronics::-260,Children::-238,Limited liabilit
Electronics::0,Interiors::-269,International::-247,Industries::-261,Trailer::-251,Private Ltd company::-269,Electronics::437,Children::-285,Limited liability compan
Children::0,Interiors::-264,International::-260,Industries::-254,Trailer::-264,Private Ltd company::-236,Electronics::-274,Children::426,Limited liability company::
Limited liability company::0,Interiors::-269,International::-287,Industries::-269,Trailer::-301,Private Ltd company::-277,Electronics::-279,Children::-277,Limited l
California::0,Interiors::-246,International::-280,Industries::-222,Trailer::-234,Private Ltd company::-266,Electronics::-220,Children::-262,Limited liability compan
Corporation::0,Interiors::-264,International::-246,Industries::-232,Trailer::-262,Private Ltd company::-294,Electronics::-254,Children::-290,Limited liability compa
Auto::0,Interiors::-220,International::-240,Industries::-232,Trailer::-238,Private Ltd company::-228,Electronics::-252,Children::-236,Limited liability company::-22
Coaches::0,Interiors::-240,International::-282,Industries::-250,Trailer::-266,Private Ltd company::-282,Electronics::-242,Children::-240,Limited liability company::
Engineering::0,Interiors::-280,International::-258,Industries::-270,Trailer::-282,Private Ltd company::-274,Electronics::-230,Children::-270,Limited liability compa
USA::0,Interiors::-242,International::-258,Industries::-226,Trailer::-234,Private Ltd company::-256,Electronics::-212,Children::-224,Limited liability company::-260
Equipment::0,Interiors::-288,International::-256,Industries::-246,Trailer::-258,Private Ltd company::-248,Electronics::-280,Children::-274,Limited liability company
Commercial::0,Interiors::-241,International::-277,Industries::-251,Trailer::-283,Private Ltd company::-259,Electronics::-267,Children::-267,Limited liability compan
Company::0,Interiors::-274,International::-298,Industries::-252,Trailer::-254,Private Ltd company::-276,Electronics::-298,Children::-284,Limited liability company::
Buildings::0,Interiors::-252,International::-240,Industries::-234,Trailer::-252,Private Ltd company::-264,Electronics::-244,Children::-244,Limited liability company
National::0,Interiors::-230,International::-244,Industries::-230,Trailer::-232,Private Ltd company::-270,Electronics::-216,Children::-268,Limited liability company:
Men::0,Interiors::-282,International::-254,Industries::-260,Trailer::-268,Private Ltd company::-278,Electronics::-240,Children::-258,Limited liability company::-226
Buisiness::0,Interiors::-255,International::-285,Industries::-259,Trailer::-263,Private Ltd company::-287,Electronics::-253,Children::-267,Limited liability company
Manufacturing company::0,Interiors::-229,International::-271,Industries::-233,Trailer::-261,Private Ltd company::-225,Electronics::-289,Children::-243,Limited liabi
Others::0,Interiors::-246,International::-240,Industries::-256,Trailer::-252,Private Ltd company::-258,Electronics::-218,Children::-268,Limited liability company::-
Technology::0,Interiors::-264,International::-250,Industries::-254,Trailer::-248,Private Ltd company::-262,Electronics::-266,Children::-240,Limited liability compan
```

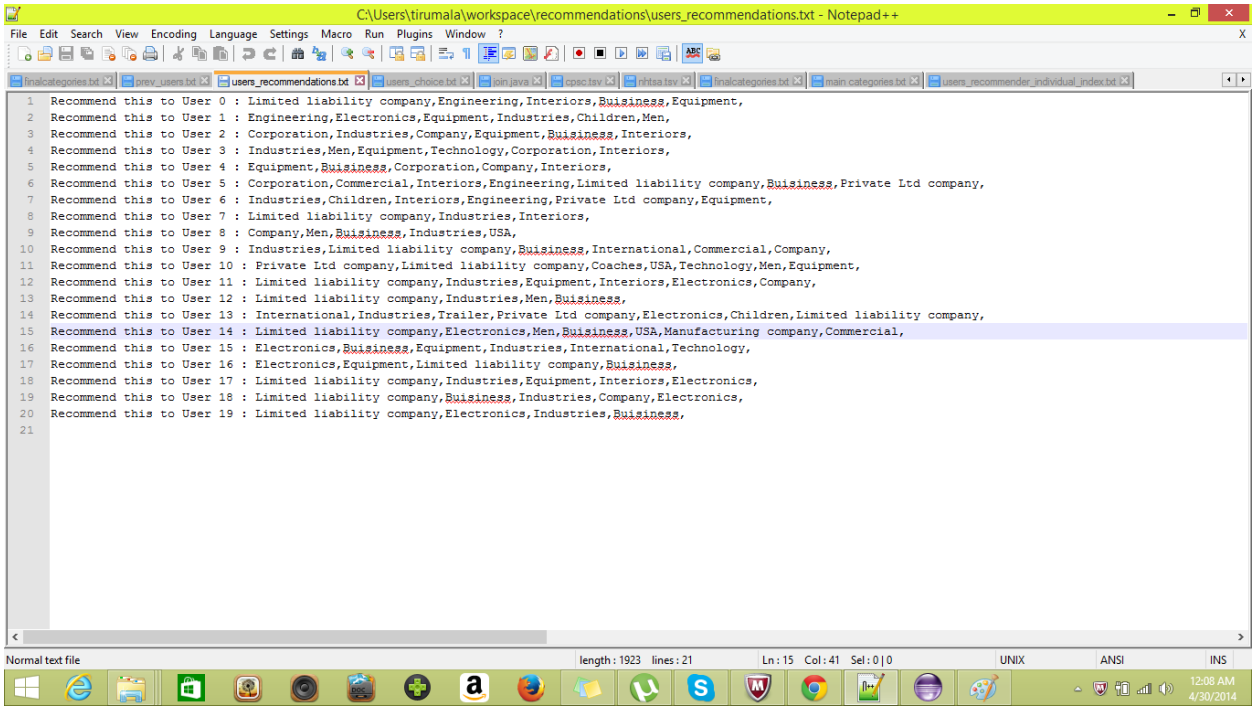
My present user's subscriptions:(to whom we need to provide recommendations now)



```
1  Auto,Children,Coaches,
2  USA,Buisiness,
3  Auto,Trailer,
4  Company,Limited liability company,
5  Auto,Engineering,Industries,
6  Auto,
7  USA,Manufacturing company,
8  Others,California,Manufacturing company,International,Coaches,
9  Others,Trailer,Equipment,
10 Corporation,Interiors,
11 Children,
12 California,Private Ltd company,
13 Others,Corporation,Manufacturing company,National,
14
15 Others,
16 Corporation,Engineering,
17 Others,Trailer,California,International,
18 Others,Corporation,Private Ltd company,
19 Others,Trailer,Interiors,
20 Corporation,California,National,Coaches,
21
```

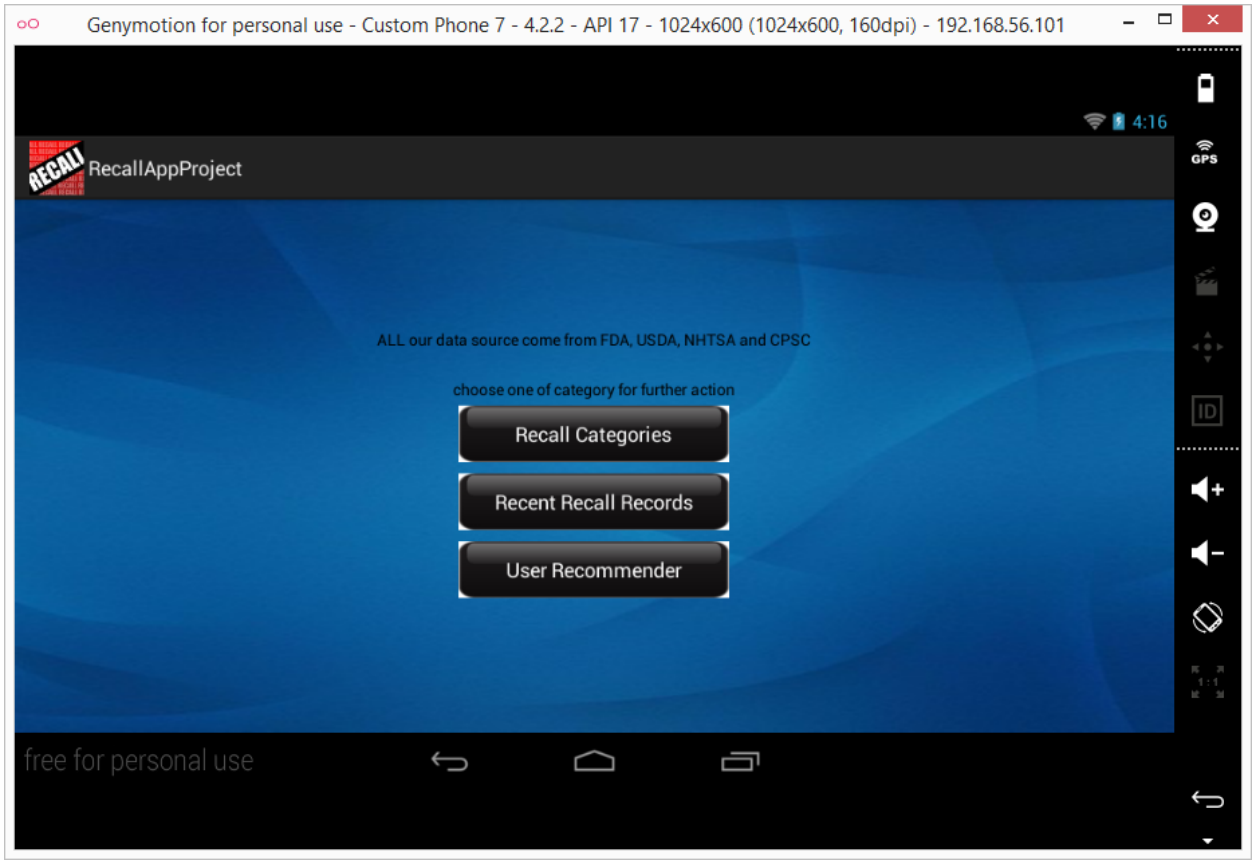
Recommendations provided by my algorithm for above users

This is just a file of reference to show recommendations list provided for user.

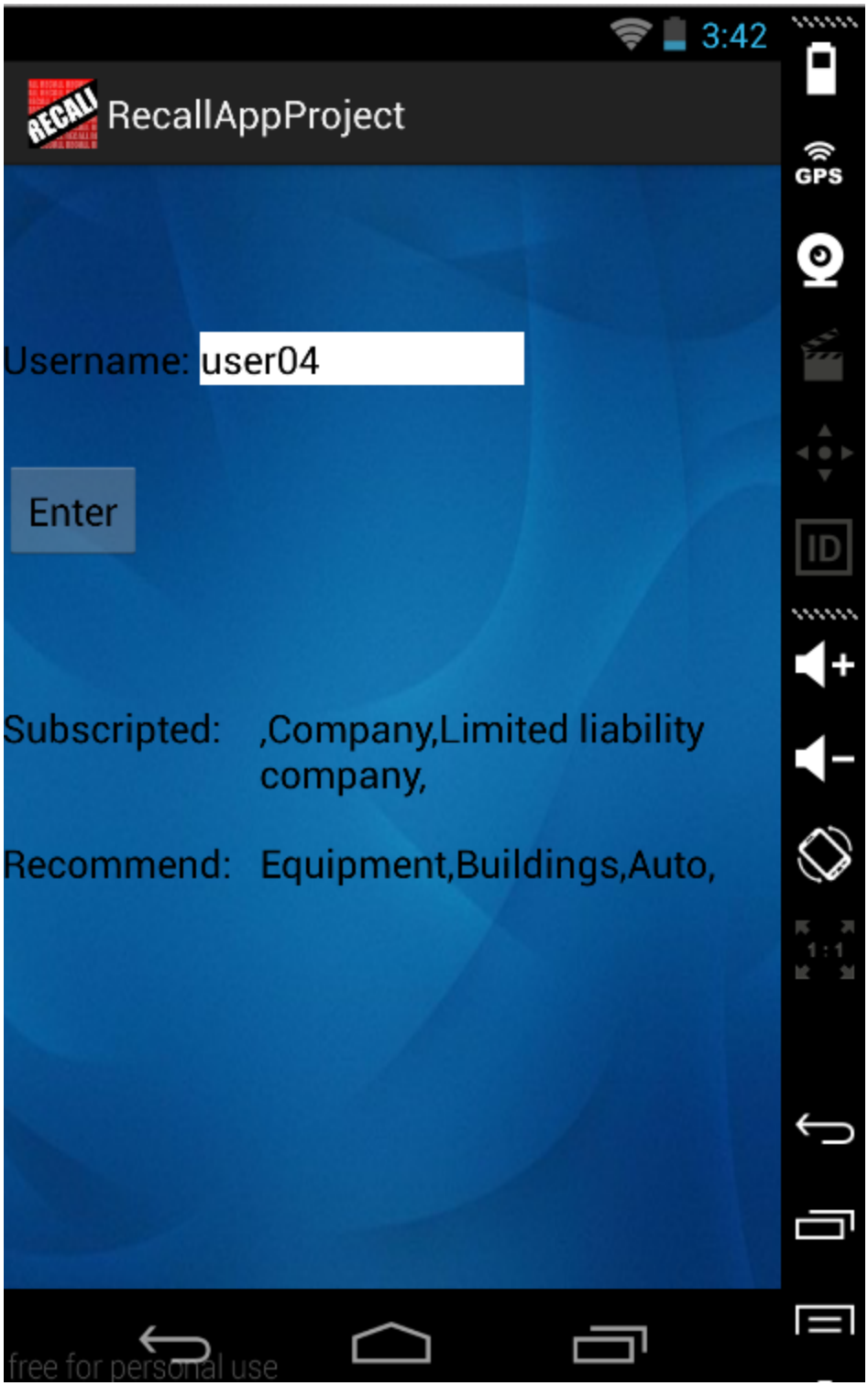


Implementation of Android Client

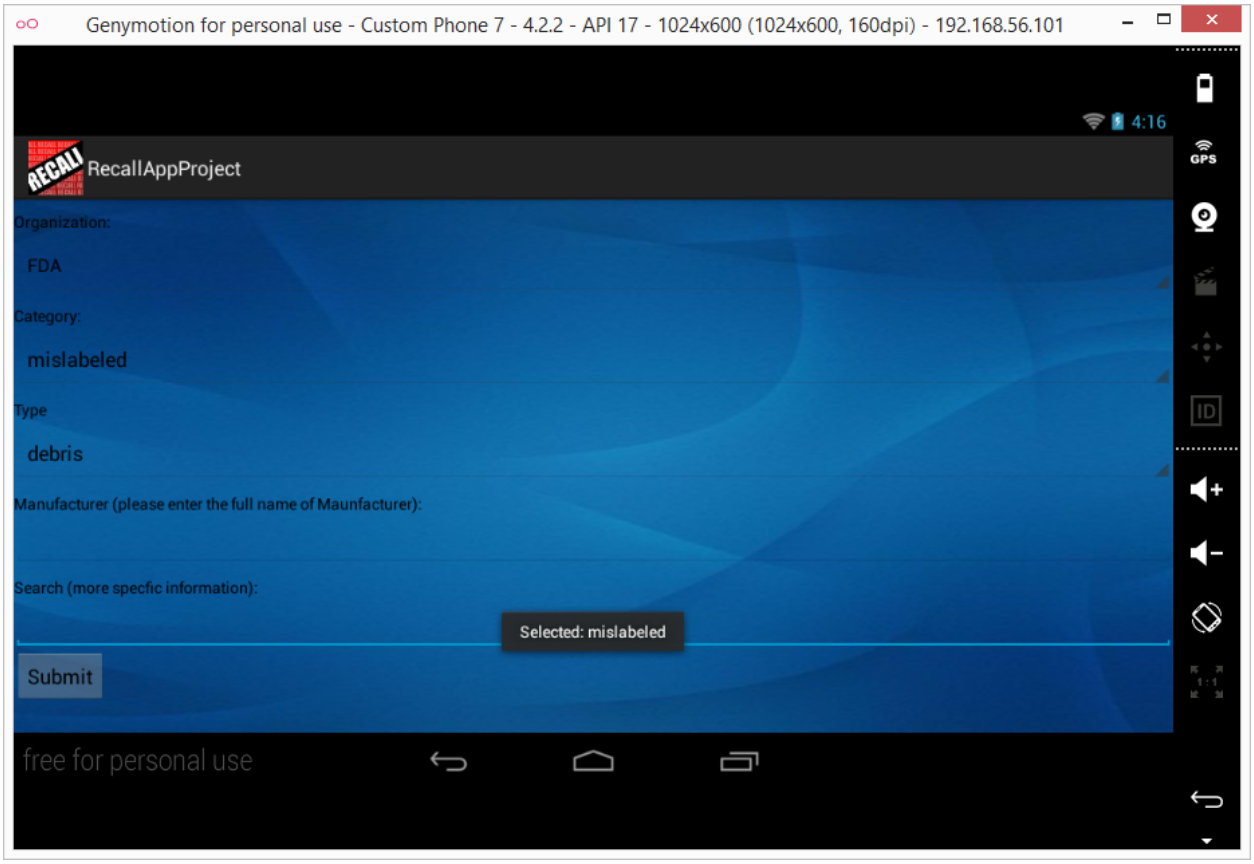
this is the main screen for our app now.



After we done the classification for data from 4 different organization and recommendation for user. I added a new view for the app, user need to enter their username, base on their subscription, out recommendation algorithm made 3 possible recommend categories for user.


3:42
RecallAppProject
Username: user04
Enter
Subscripted: ,Company,Limited liability company,
Recommend: Equipment,Buildings,Auto,
GPS
ID
1:1
free for personal use

I restructure our app UI. Now the user will choose the organization first , then for each organization, there are some classified categories. And also for each category, there are some different classified subcategories. Because there are too many classified subcategories for NHTSA recalls. So I made a editText view to let user type any manufacturer they wanna see. Also I made another editText view for user to search any additional information.



Appendix

Hadoop and Mahout linux shell commands for classification

This listing uses classes found at fredang/mahout-naive-bayes-example project on Github

1. **Convert training set to Hadoop sequence file format**
   ```
   java -cp target/tnb.jar com.chimpler.example.bayes.TweetTSVToSeq
   data/fda-training.tsv fda-seq
   ```
2. **Upload sequence file to HDFS**
   ```
   hadoop fs -put fda-seq fda-seq
   ```
3. **Transform training sets to vectors with Mahout (using term frequency by document frequency weights)**
   ```
   mahout seq2sparse -i fda-seq -o fda-vectors
   ```
4. **Split data into training set and testing set**
   ```
   mahout split -i fda-vectors/tfidf-vectors --trainingOutput
   train-fda-vectors --testOutput test-fda-vectors
   --randomSelectionPct 40 --overwrite --sequenceFiles -xm
   sequential
   ```
5. **Use training set to train classifier**
   ```
   mahout trainnb -i train-fda-vectors -el -li labelindex -o model
   -ow -c
   ```
6. **Test the classifier on the training set**
   ```
   mahout testnb -i train-fda-vectors -m model -l labelindex -ow -o
   fda-testing -c
   ```
7. **Test the classifier on the testing set**
   ```
   mahout testnb -i test-fda-vectors -m model -l labelindex -ow -o
   fda-testing -c
   ```
8. **To use classifier and classify new documents, copy several files from HDFS**
   ```
   hadoop fs -get labelindex labelindex
   hadoop fs -get model model
   hadoop fs -get fda-vectors/dictionary.file-0 dictionary.file-0
   hadoop fs -getmerge fda-vectors/df-count df-count
   ```
9. **Classify the new file (optional section in bold red writes to text file)**
   ```
   java -cp target/tnb.jar com.chimpler.example.bayes.Classifier
   model labelindex dictionary.file-0 df-count data/fda.tsv >
   fda-classified-result.txt
   ```