# Increment 2

Group 6: LeeStudents
James Clark
Tirumala Konireddy
Jagadish Tirumalasetty
Wang Zhang
Dr. Yugyung Lee
CS 560: KDM
April 1, 2014

# Data

It turns out there's an ethical approach to consider when crawling sites. Two things to immediately consider: 1) robots.txt usually found at domain.com/robots.txt and 2) the robots meta tag such as <meta name="robots" content="index, follow" />. In the case of this site, it is acceptable to crawl for more data. I will conduct such a crawl at a metered pace, something like one request per 15 seconds. Results should be stored to prevent the need for repeated crawling of the same pages. MongoDB is probably ideal for this. More info is available at <a href="http://www.metatags.info/meta\_name\_robots">http://www.metatags.info/meta\_name\_robots</a>
<a href="http://www.robotstxt.org/robotstxt.html">http://www.robotstxt.org/robotstxt.html</a>

# Discovery

• For different algorithms, data format is different<sup>1</sup>, therefore we need a data format processor.

Suggested workflow:

Store data set in DB such as HBase or MongoDB.

Get data from DB

Convert data to format

**Process with Mahout** 

• For the classifier to work properly, this set (Naive Bayes classifier) must have at least 50 tweets messages in each category.<sup>2</sup>

The CPSC has an API:

http://www.cpsc.gov/en/Recalls/CPSC-Recalls-Application-Program-Interface-API-Information/Using this API, I found 5000+ links to recall data in date range 1972-01-01 to 2015-01-01

USA.gov has an API:

http://search.digitalgov.gov/developer/recalls.html

# **Algorithms**

The CPSC provides an ASMX web service endpoint, and they serve data from 1973 to current. A one-time process is needed to gather all recall data. After that, a periodic check is needed to look for the latest updates.

Connect to CPSC web service, <a href="http://www.cpsc.gov/cgibin/CPSCUpcWS/CPSCUpcSvc.asmx">http://www.cpsc.gov/cgibin/CPSCUpcWS/CPSCUpcSvc.asmx</a> Invoke getRecallByDate method foreach result in list do

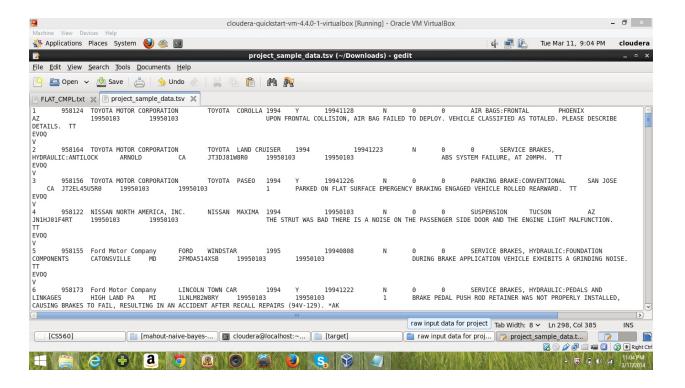
<sup>&</sup>lt;sup>1</sup> Feichen, Tutorial 7, Page 18

<sup>&</sup>lt;sup>2</sup> Feichen, Tutorial 7, Page 19

start json object store result attributes to json object visit result url parse web page at url store fields from html into json object return json to caller

#### Analytical Tasks

 Collection of data: We get data from http://www-odi.nhtsa.dot.gov/downloads/flatfiles.cfm about all the product recalls that had done in past till now. Now, we are manually monitor and download the updated recall data flat .txt file.



#### 2. Classification of data:

We are using naive bayes algorithm to classify user comments to a recall category. We used mahout tool for naive bayes classification. The following steps are followed to classify data using mahout.

(i) Converting data into sequential files:

Mahout works directly only on sequential files, So to convert this data which had a tab separator between every two columns in a line, we had developed a java program "CarsTSVToSeq".

#### (ii) Upload data:

After converting data into sequential files, we have to upload the sequential file to hadoop on which mahout directly works on.

### (iii) Training and Testing:

We had developed a Classifier, trainer java files using bayes classes in java to do train the classifier with the given seq files.

We had trained the classifier with varying number of input data records.

We got 90.67% accuracy in testing the trained data entries when we trained 10040 lines of entries, whereas we got only 77.4%, 55.08% accuracy when we trained with 100,000 and 967,487 data records.

# (iv) Classification:

We had used 217 data entries for classification and we found good results by cross-verification.



#### (v) Uploading results to Solr:

Because of some issues to work with Glassfish server, we are unable to upload this results to Solr using web restful service.

## **Implementation**

Implementation of data model and algorithms (Machine Learning)

We are using flat data tsv files which contains details of about complaint id, manufacturer, model, recalled component description, number of deaths involved in fail of such component, customer description of that component failure etc., in 47 columns using a tab separator between every two columns.

We are using naive bayes algorithm which uses bayes probability by assuming every occurrence of event as independent for training a classifier to classify user comments to a recalled component category. For implementing this naive bayes algorithm we used hadoop files system, mahout tool for classifying data. We had used Solr server to store the classified results so that they will be available for accessing by end user. We are developing a web service to convert the classified results into json format and upload them to Solr server using Glassfish server.

### **Application Interface**

Most Likely finish the Application interface, furthermore we will change xml to json or HTML format. Here is some screenshots for our app.

