

Laxman Dutt Degala | CS560 | [Date]

Mashup Application using Bigdata analytics & Machine Learning

KDM - Midterm

# Plan:

Plan is to build a mash-up application using the following features:

1. Cloud delivered big data analytic services
2. Change the algorithm if required for the data picked.
3. Develop an application to bring useful information/services from the existing data.

Datasets considered:

1. Dataset is taken from **UCI – Machine Learning Repository** - <http://archive.ics.uci.edu/ml/datasets/Car+Evaluation> .
2. Dataset is related to – Cars evaluation

Theme of the Application:

To start with, I have considered only USA for my project. So my application will be taking the inputs from the users about the cost, maintenance, number of persons, number of doors, Boot luggage space and recommends user to whether go with the car or not.

Basically, User will be given either of three recommendations:

1. NO – Don’t go with the car. Plan for other!
2. TRY – It’s Kind of OK. Can be considered as a option!
3. YES – Too good for its features. Go with it!! Good Buy!!

Steps followed:

Work flow chart:

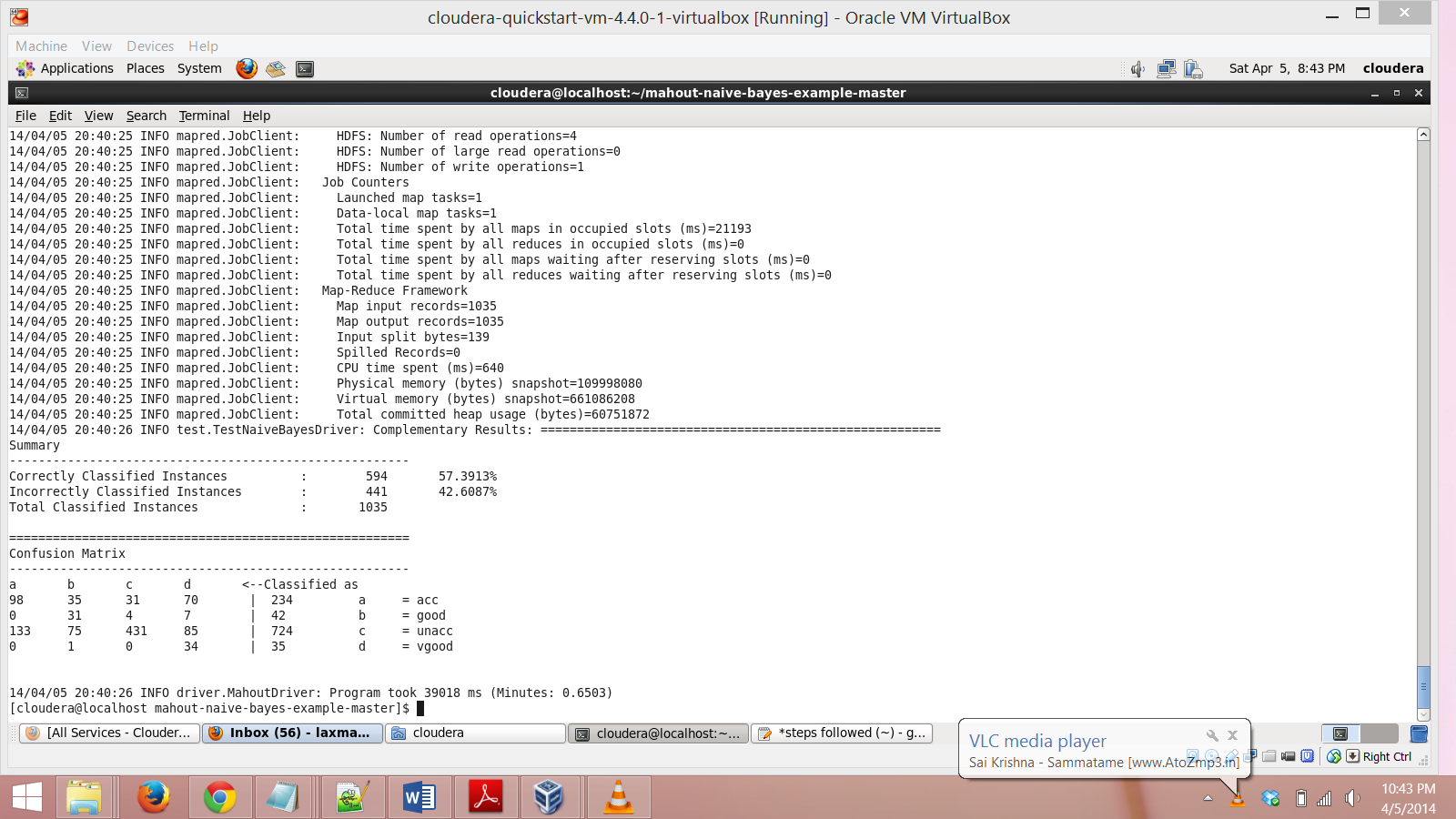
Take the data -> Consider the input and target fields -> Consider the algorithm which best suits for the data -> Run Map Reduce -> Take the refined trained and tested data and move it to SOLR -> Create a web or mobile application -> Create a RESTFUL web service to retrieve the data from SOLR in .JSON format -> Manipulate the data and pop the useful information back on the web application.

1. Considered the data and tried figuring out the way to use the data effectively.
2. Initially, I have gone with 4 classifiers and done the machine learning with the existing data.
3. Have done a couple of iterations to increase the percentage matching of the classifier with training and testing data.

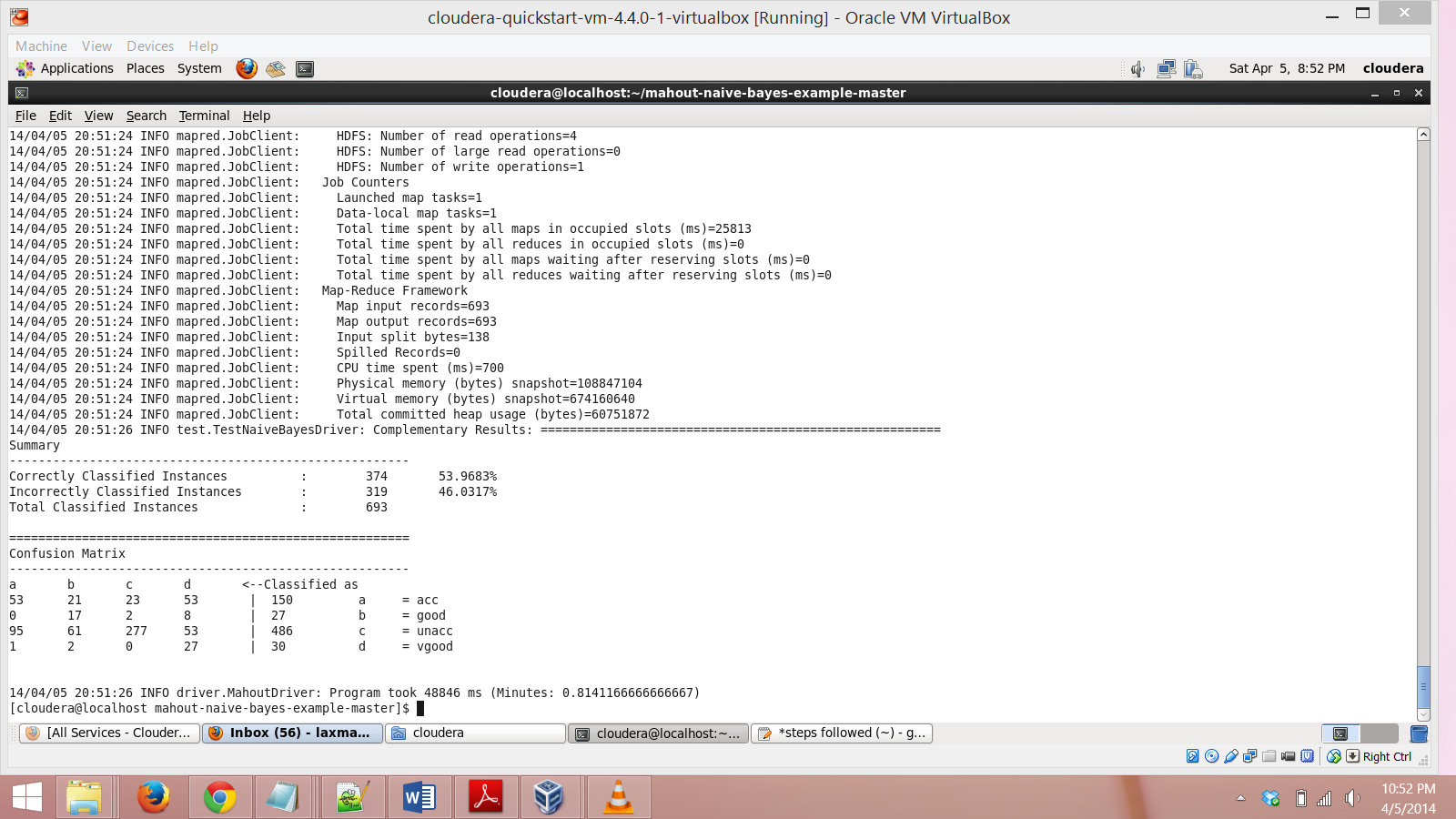
Iteration 1:

Considered 4 classifiers – Not acceptable, Acceptable, Good, Vgood. But the confusion matrix was not giving the expected results.

Correctly classified instances were only 57% for training and 53% for testing and also giving 4 options for the users whether or not to select a car is looked not right.

Classifier with Train data: 

Classifier with Test data:

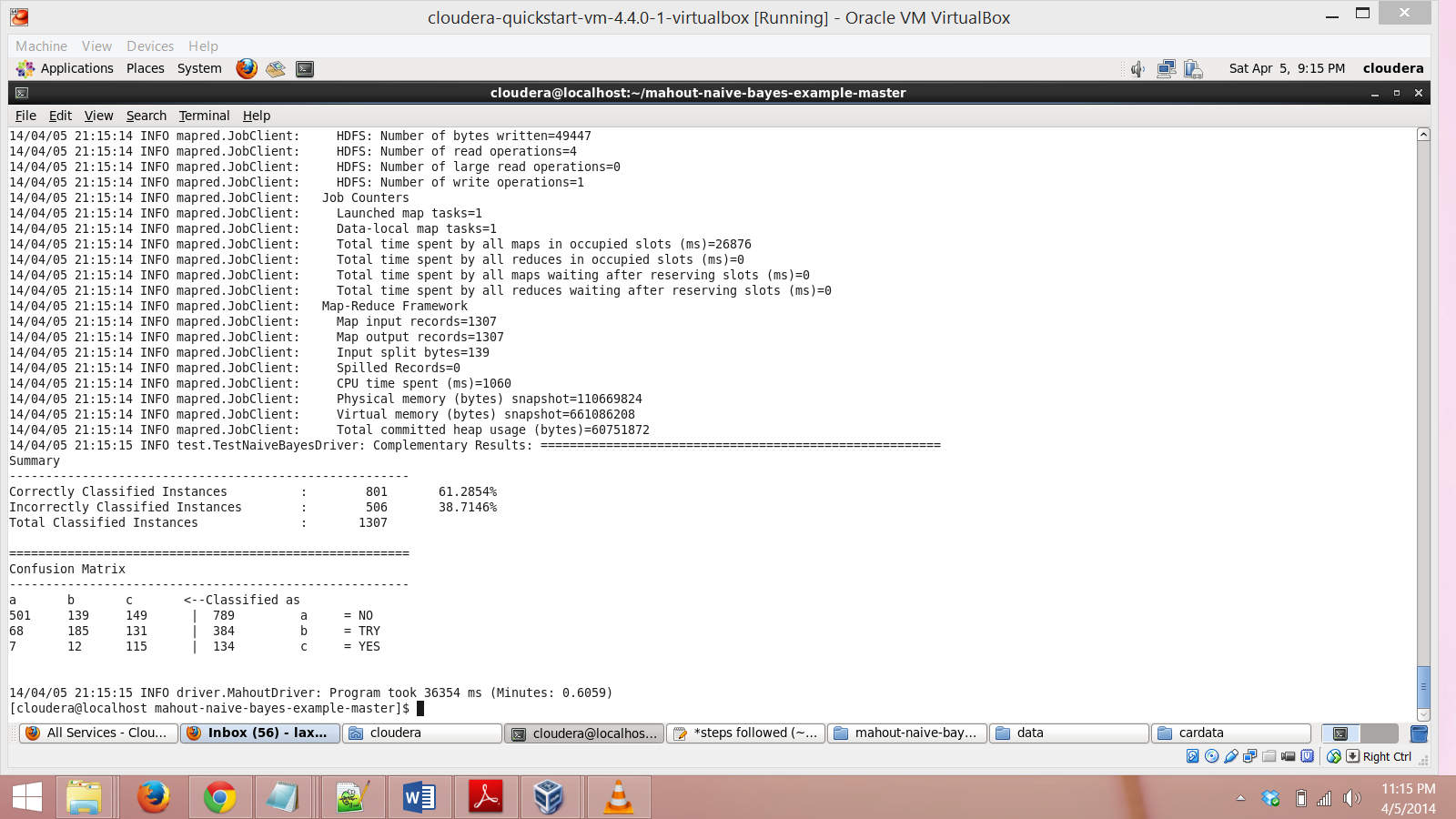


Iteration 2:

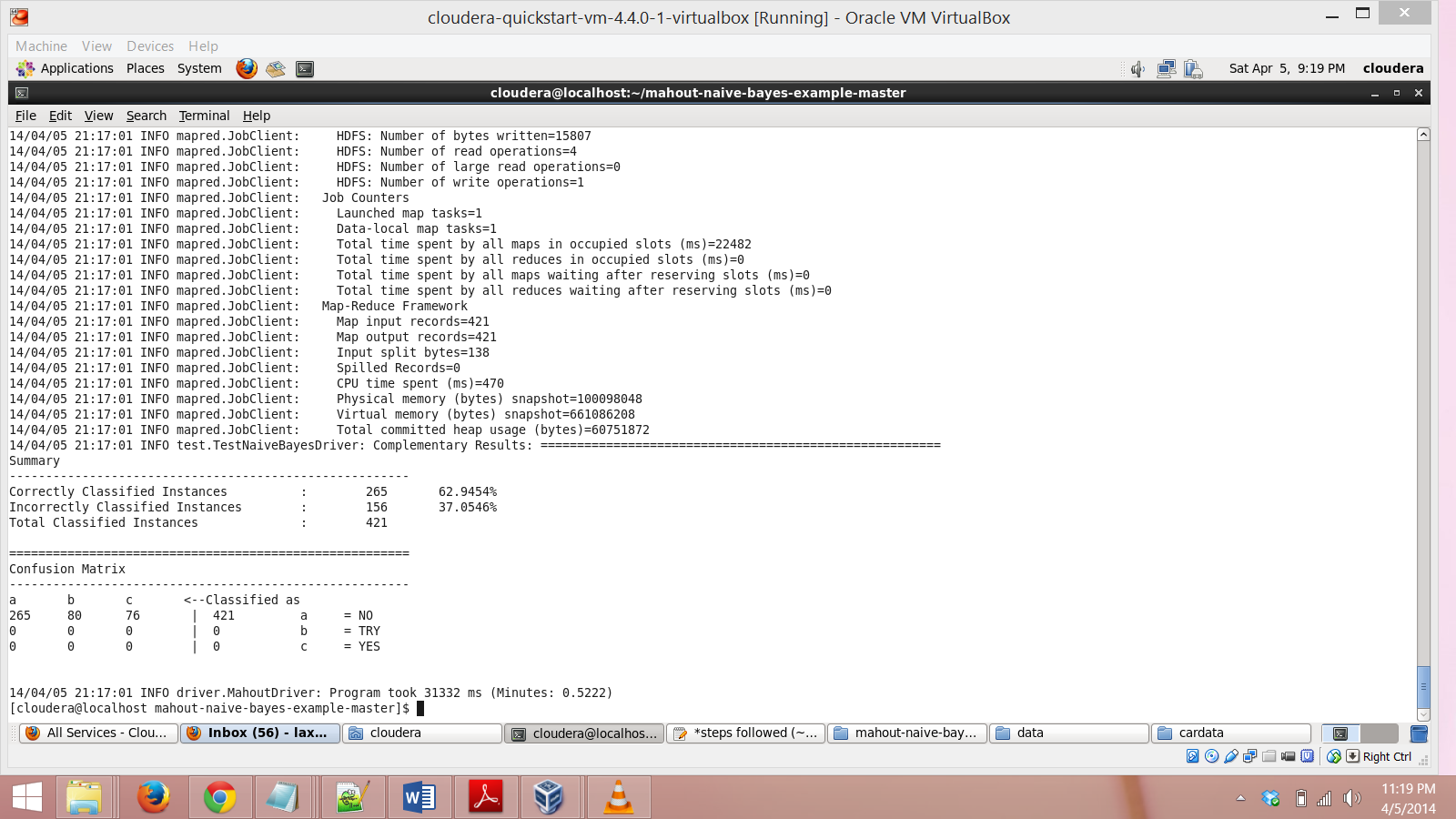
Now, there are only 3 classifiers, No, Try and Yes.

No – Not to consider, Try – Can consider, Yes – Best Buy.

Train data with the new classifiers:



Test data with the new classifiers:



Map Reduce:

As the data received from the Car evaluation site has the map reduce algorithms run at the time of machine learning and the data is structured as well. Thus, I didn’t require a map reduce job separately for bringing in the useful information.

Convert the .txt output file from the machine learning algorithm to .Json format in order to move the complete data to SOLR

Have written a java code to convert the .txt to .json format. As I had 8 columns which are to be converted to json format, conversion took good amount of time.

Please find attached the source code details in turnitin.

Upload to SOLR:

Faced many issues and learnt many interesting things related to it, while uploading the data to SOLR.

I had 8 attributes/fields that are to be pushed to SOLR, then found the method to upload different types of data to SOLR with the help of google.

We can upload any no. of attributes to SOLR for a record and also we can specify user defined attribute names and not just “ID” or “TITLE”.

Its just that we will have to specify the datatype of the attribute along with the attribute name.

For eg: In this case, I will have to upload the attributes cost, maintenance, doors, persons to SOLR. So in this case, we can create attributes for each one of these like stated below:

Cost – “cost\_s” – s indicates string. Possible values for the field/attribute is vhigh, high, medium, low.

Car doors – “door\_i” – I indicates integer. Doors could have values 2, 4, 6 etc which are integers.

Also,

SOLR in the cloudera image has bad health from 3 days and the SOLR provided to the class (134.193.136.127:8983) is unable to upload the data when pushed using the CURL command.

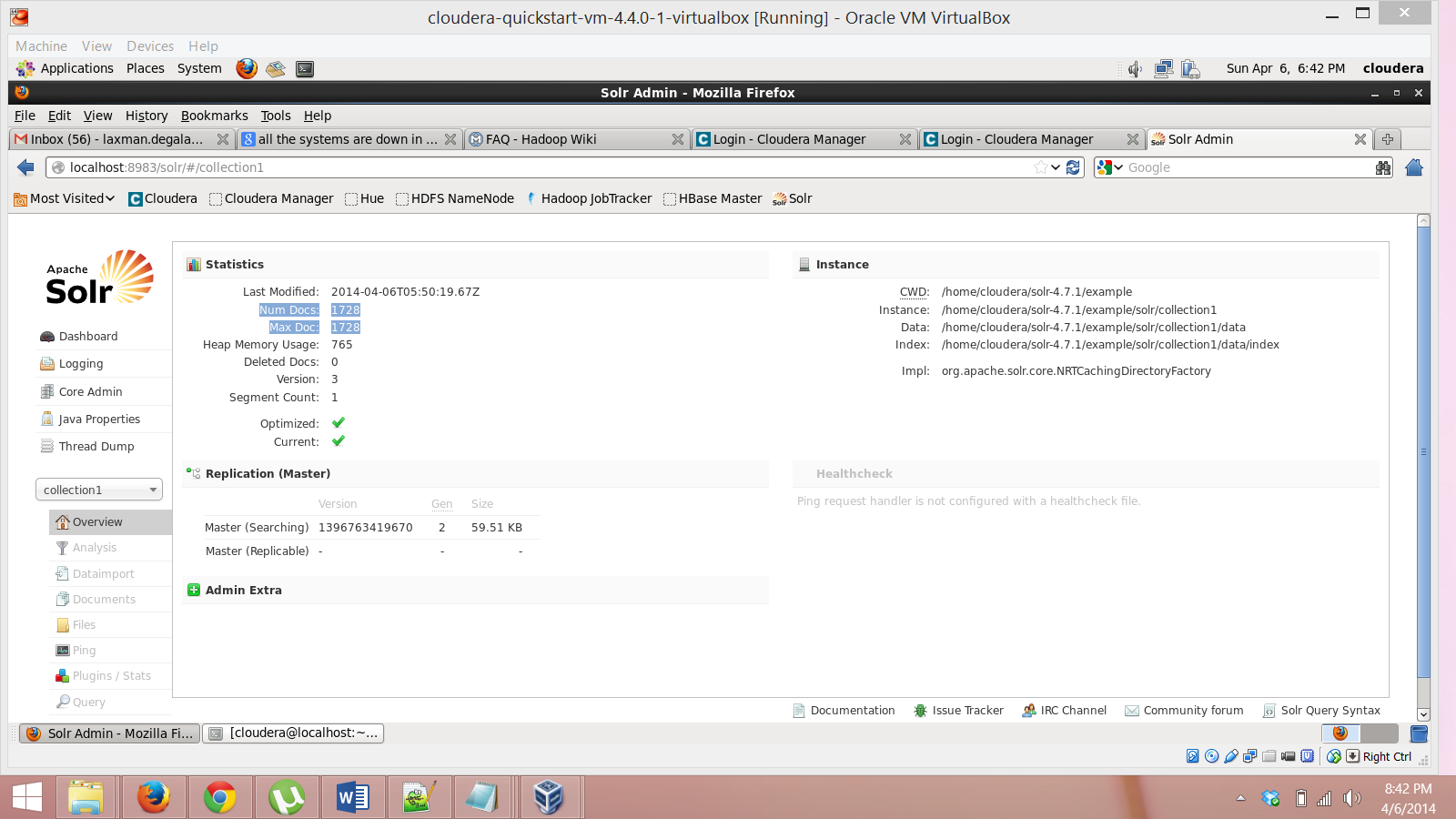
Thus, I have stopped the SOLR that’s in the cloudera image and downloaded a new version Lucene/SOLR4.7.1 on top of cloudera image. It helped me upload the data in my local machine.

Commands and method to store the data in SOLR are little different from what we learnt earlier but it’s very easy to learn.

Please find the link below which I used to upload the data to SOLR:

<http://www.solrtutorial.com/solr-in-5-minutes.html>

All 1728 records have been pushed to SOLR using the PUSH command.

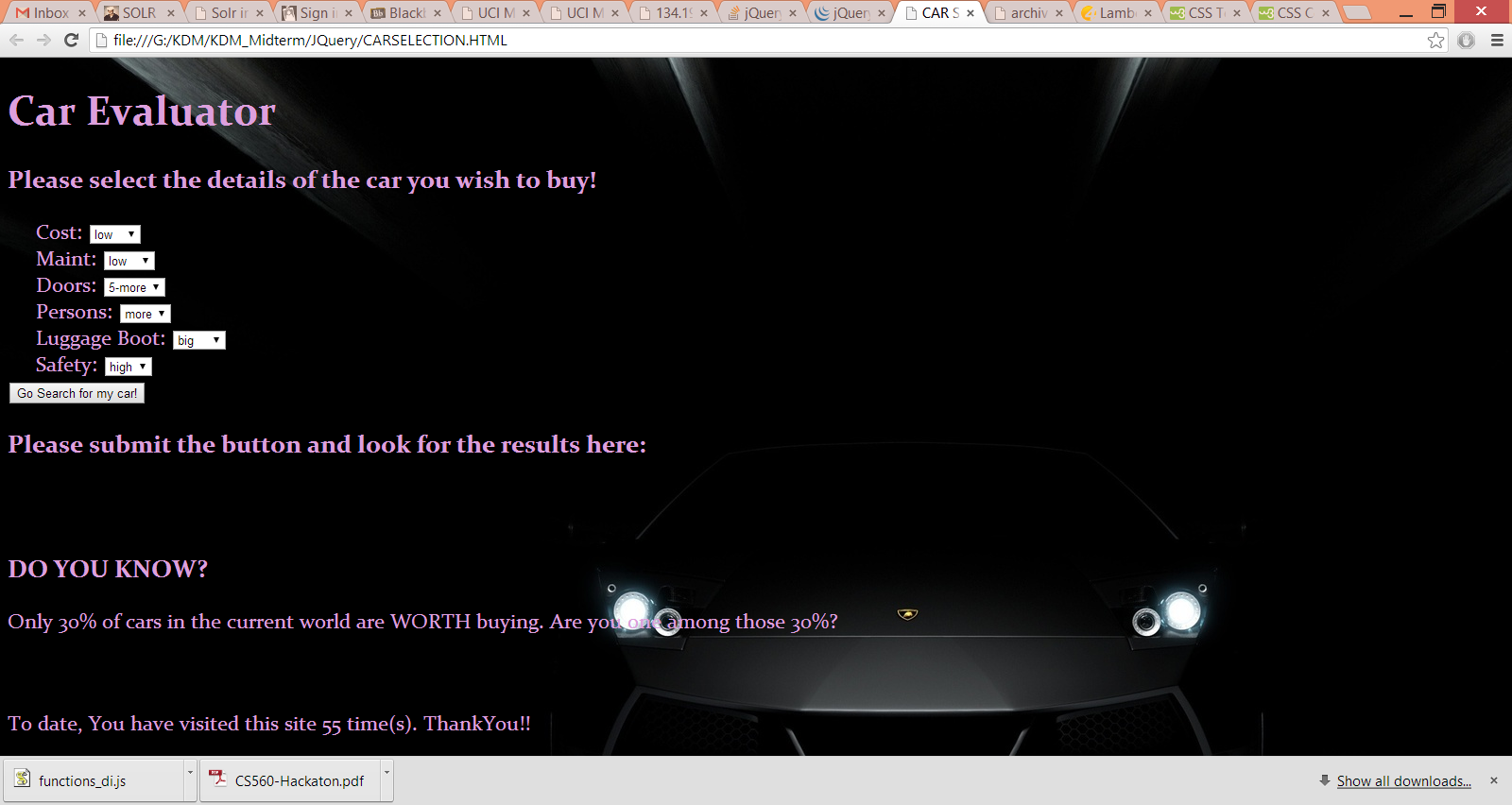


Web Application:

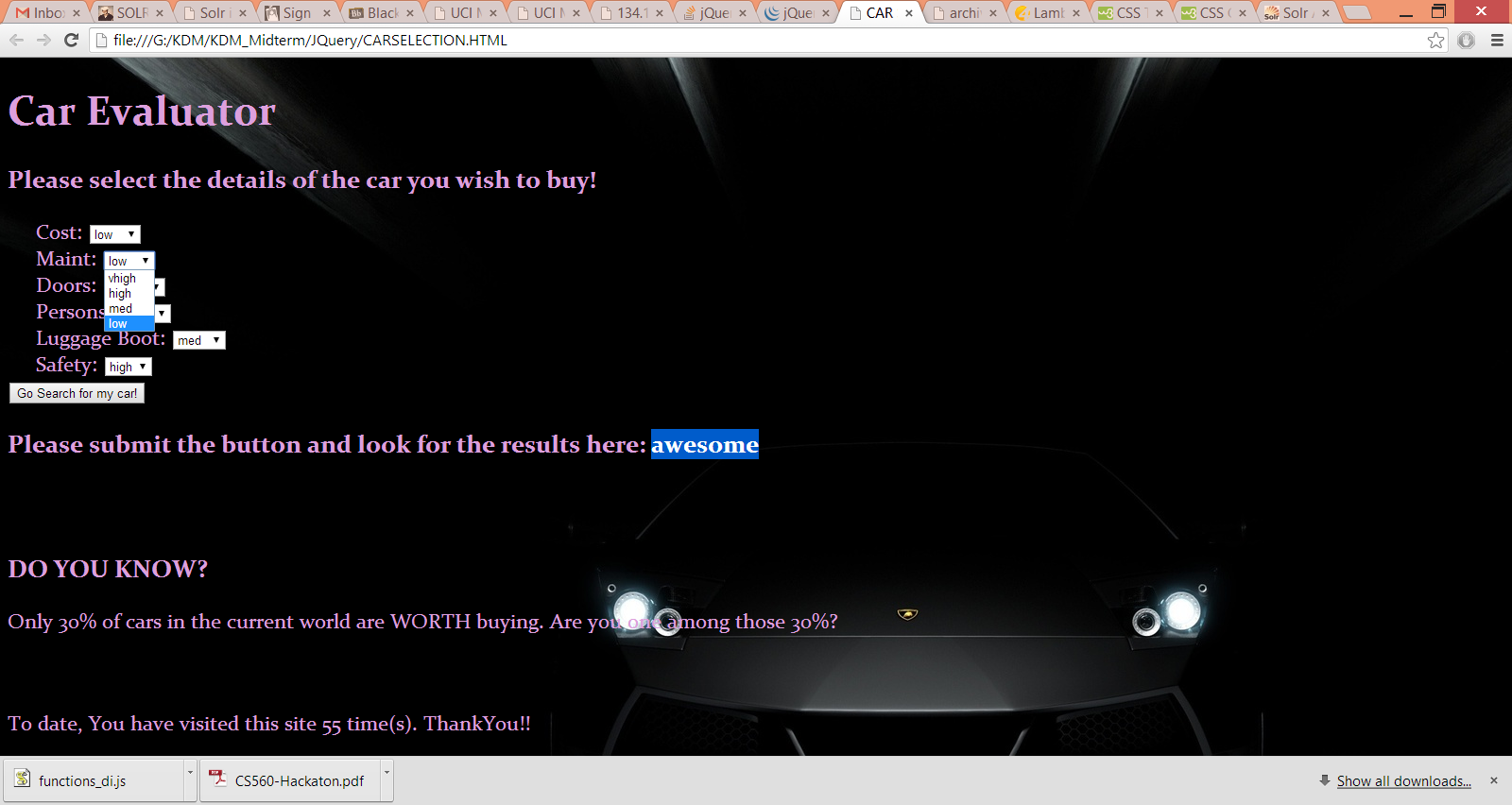
As stated earlier, my web application would take the inputs from users about and gives them an idea of whether to take it or not.

1. **No** – Lets the users know that its not a good buy!
2. **Try** – Lets the user know that its OK, but can Try!
3. **Awesome** – Great Buy!

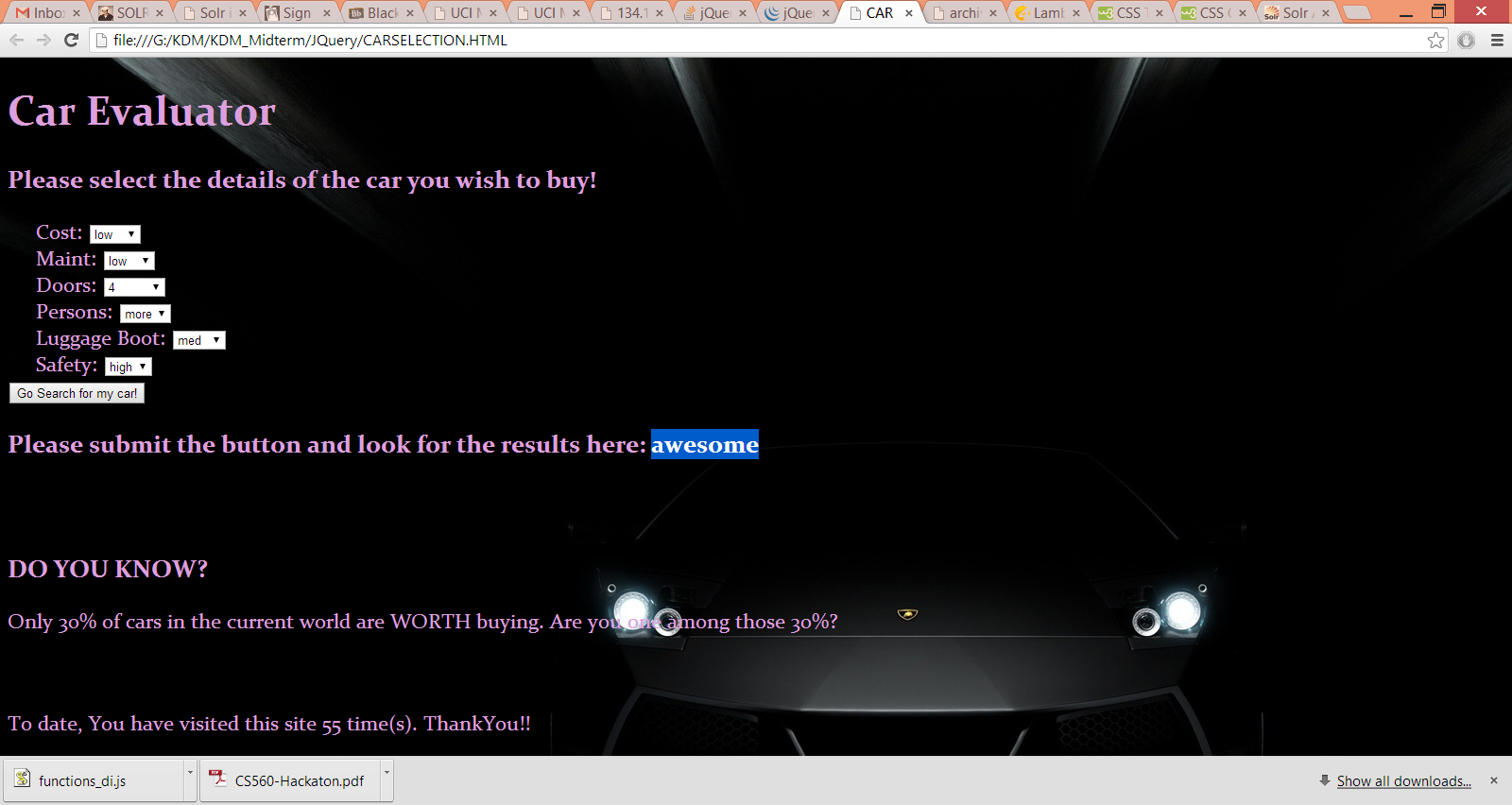
Please have a look at the UI below:



Dropdown box to select the option:



Result says, whether to go for it or not. In the below case it says “Awesome Buy”



Useful Information, which can be popped on the web application:

Considering all the cars from USA, there are 60% cars which are categorized as BAD, 20% cars which are considered as OK, 10% cars which are considered as VERY GOOD buy.

Things learnt from the exam:

1. Sequential flow of what to be done when a data set is provided.
2. Decision making on what can be retrieved from a given chunk of data
3. Map reduce code changes based on data (Have tried but couldn’t come with correct results, will look into this)
4. How to work on SOLR.
5. How to retrieve data from SOLR using a RESTFUL service. (Complex task)

Tools or services used:

1. HDFS/Mahout/Lucene/SOLR4.7.1
2. Naïve bayes algorithm
3. Java – for map reduce and Code to convert the file from .txt to .json
4. HTML5, Javascript and Jquery for Web application
5. RESTFul service using Ajax.

Github link:

https://github.com/LaxmanDuttDegala/KDM\_Midterm.git

Also provided the web application link for testing!

Reference:

* Map Reduce used for the word count
* <http://www.solrtutorial.com/solr-in-5-minutes.html>
* <http://skipperkongen.dk/2011/01/11/solr-with-jsonp-with-jquery/>
* <http://stackoverflow.com/>

Thanks for the opportunity!

Laxman Dutt Degala