CS8803 BDS / CS4365

Homework Assignment 2

(Programming Category)

Student Name: Enmao Diao

Problem 2. Hand-on Experimentation with Classification

Your task for this assignment is to explore run and evaluate the C4.5 decision tree classifier or Naïve Bayesian classifier using Mahout (http://mahout.apache.org/users/classification/) or R or Weka. You can implement your own classifier using any programming language that you are familiar with to compare with those from Mahour, R or Weka.

1. The program should be executable with at least 3 parameters: the name of the training dataset file, the name of the test dataset file and the name of the output file.
2. The program should output a file that contains the class labels for all the records in the test dataset and the classification accuracy computed as the percentage of correctly classified records in the test dataset.

3. Evaluate your implementation using the provided mushroom dataset (mushroom.training, mushroom.test). You will use the training set to build your classifier and the test set to evaluate its accuracy. The provided two datasets were created using the original mushroom dataset from UCI repository but with one attribute with missing values removed. The training dataset contains 7423 records and the test dataset 701 records. The first attribute is the class of each record and the rest 21 attributes are categorical attributes.

4. Write a brief report that include the following:
- Present and discuss the results of your experiments or your implementation on the provided dataset and other datasets (if any).
- Discuss, if any, the experiences and lessons you have learned from the implementation and experimentation.

5. Deliverable.
- One tar or zip file that contains your source files, the executable, a readme file explaining how to compile/run your program, the output file for the test dataset screen shots of your execution process.

- Runtime statistics in excel plots or tabular format.
- Report in pdf/word/ppt.

Install

1. Install Mahout on Cloudera with Hadoop already built in.
   Download from the official website and extract the files
2. Set environmental variables in ~/.bashrc

```
nano ~/.bashrc
```

Paste following to ~/.bashrc

```
export MAHOUT_HOME=the path you install mahout

export MAHOUT_LOCAL=true #for running standalone

export HADOOP_HOME=/usr/lib/hadoop
```

Run

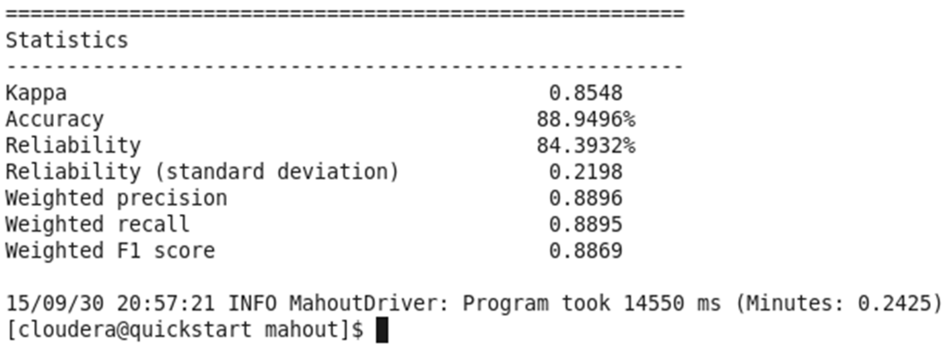
At first I try 20news example that is given from Mahout official website. The sources code is in classify-20newsgroups.sh

```
cd $MAHOUT_HOME

mvn -DskipTests clean install #Only run once

./examples/bin/classify-20newsgroups.sh
```

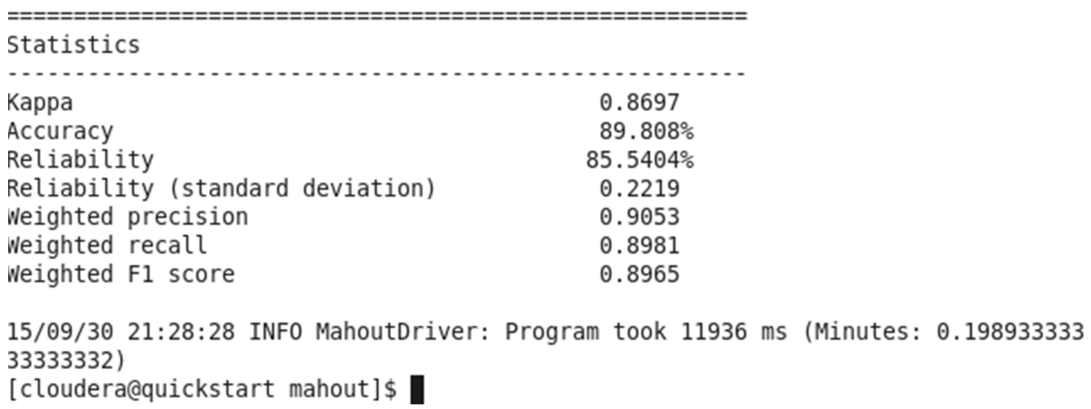
Make sure you have cd $MAHOUT_HOME otherwise it wont work



```
=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.8548
Accuracy                                   88.9496%
Reliability                                84.3932%
Reliability (standard deviation)            0.2198
Weighted precision                          0.8896
Weighted recall                             0.8895
Weighted F1 score                           0.8869

15/09/30 20:57:21 INFO MahoutDriver: Program took 14550 ms (Minutes: 0.2425)
[cloudera@quickstart mahout]$ 
```

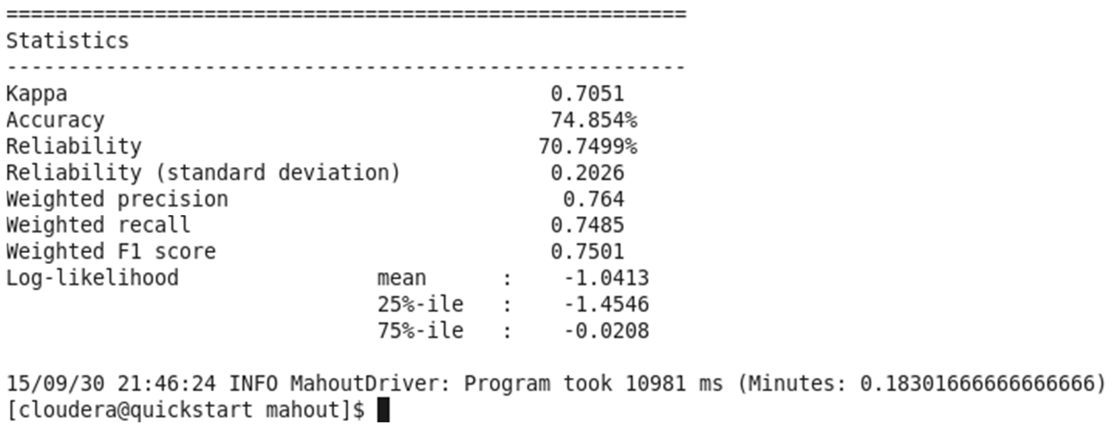
Fig. 1. Complement Naive Bayes on 20-news group example



```
=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.8697
Accuracy                                    89.808%
Reliability                                85.5404%
Reliability (standard deviation)            0.2219
Weighted precision                          0.9053
Weighted recall                             0.8981
Weighted F1 score                           0.8965

15/09/30 21:28:28 INFO MahoutDriver: Program took 11936 ms (Minutes: 0.198933333
33333332)
[cloudera@quickstart mahout]$ 
```

Fig. 2. Naive Bayes on 20-news group example



```
=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.7051
Accuracy                                    74.854%
Reliability                                70.7499%
Reliability (standard deviation)            0.2026
Weighted precision                           0.764
Weighted recall                             0.7485
Weighted F1 score                           0.7501
Log-likelihood              mean      :     -1.0413
                            25%-ile   :     -1.4546
                            75%-ile   :     -0.0208

15/09/30 21:46:24 INFO MahoutDriver: Program took 10981 ms (Minutes: 0.18301666666666666)
[cloudera@quickstart mahout]$ 
```

Fig. 3. Logistic Regression - trained via SGD on 20-news group example

After I finish step I thought I am close the answer. Actually, I am too naïve.

First analyses the classify-20newsgroups.sh source code.

Following steps are needed to finish Naive Bayes task:

1. mahout seqdirectory

   This command transcribes input data into sequence files.

2. mahout seq2sparse

   This command transcribes sequence files into vector files.

3. mahout trainnb

   Train the classifier.

4. mahout testnb

   Test the classifer.

The details of each command can be found here

https://mahout.apache.org/users/classification/bayesian.html

Experience

At first I try to modify classify-20newsgroups.sh in order to fit in this project. In classify-20newsgroups.sh, the dataset is sequenced and vectorized together and later the training and test vector files are split by using mahout split.
In my case, I just have to sequence and vectorize twice for my training and test dataset mushroom. training and mushroom.test.

Remember this

1. Make sure you put the .sh file inside Mahout directory otherwise when error occurs, the terminal just shut down without giving any error message.
2. Make sure you run your .sh from MAHOUT_HOME for example using ./examples/bin/mushroom.sh
3. In classify-20newsgroups.sh, it assumes you run from MAHOUT_HOME and you put it in ./examples file therefore it has code like

```
cd $START_PATH

cd ../..
```

You can change it into

```
cd $MAHOUT_HOME
```

4. Make sure set-dfs-commands.sh is also in the same directory as your .sh file because you will need it

```
# Set commands for dfs

source ${START_PATH}/set-dfs-commands.sh
```

At this point I can see mushroom-sequence and mushroom-vectors folder in /tmp/mahout-work-${USER}. However, it always fails in training step. The code I have at this step is mushroom_old.sh

The problem is that the input data file is stored like a tabular format. Each row is a record which should be sequenced separately. In 20-news group example, each file is an email information and each file is a record. If I still use seqdirectory I will need more than 7000 files for each record from each row.

I get this answer from
http://stackoverflow.com/questions/11994930/converting-csv-to-sequencefile

Therefore I have to write code to generate sequence files. I need to use mahout as library which requires me to use maven in Eclipse.
Instruction on install maven in Eclipse
https://www.youtube.com/watch?v=63k560Livmg
I add following dependency to pom.xml

```
<dependency>
            <groupId>org.apache.mahout</groupId>
            <artifactId>mahout-core</artifactId>
            <version>0.9</version>
      </dependency>
      <dependency>
            <groupId>org.slf4j</groupId>
            <artifactId>slf4j-simple</artifactId>
            <version>1.7.12</version>
</dependency>
```

I write a file to generate source code NaiveBayes.java
It will read data file by assuming the delimiter is tab (\t) and then set key as /key/row#
Key is the first column of data and row# is the number of rows (records) that this key current belongs to

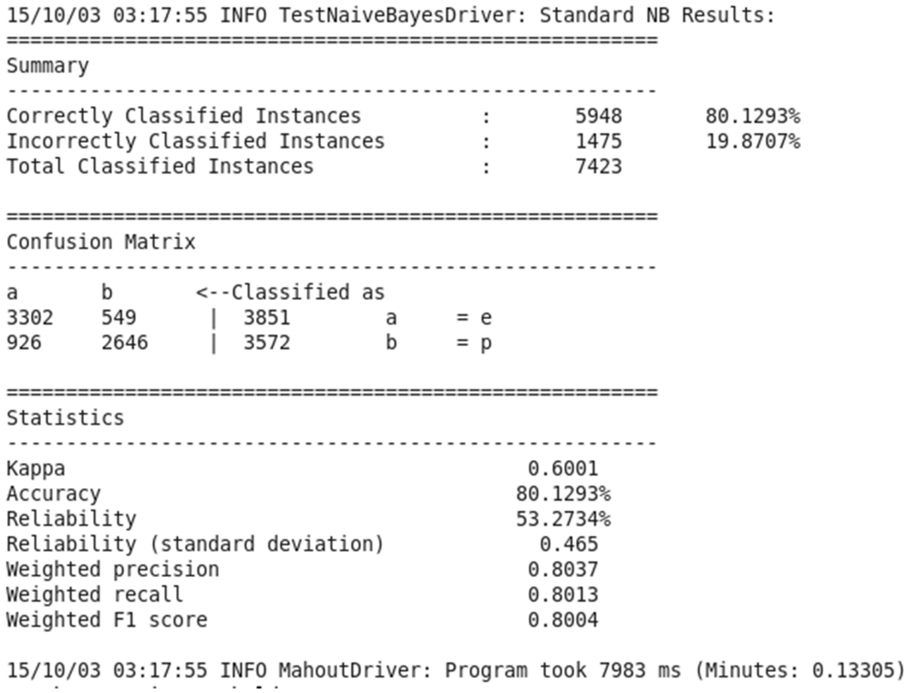
Remember this

1. The Class of key and value is org.apache.hadoop.io.Text. This is required by seq2sparse and most online tutorials confuse me by using org.apache.mahout.math.VectorWritable. If you need to use VectorWritable the value of data should be double instead of String.
Some tutorials on this
http://stackoverflow.com/questions/11645294/how-can-i-use-mahouts-sequencefile-api-code/11645430#11645430
http://stackoverflow.com/questions/13663567/mahout-csv-to-vector-and-running-the-program

2. Another thing is extremely tricky and I spent about two nights on it. When I start to use mahout Naïve Bayes, I always wonder how the program knows which data or column is the category lable. Later I found the key, value pair and I thought I figured it out. Actually, I start to write key as just the String value in the first column like “e” or “p”. The program fails no matter what I try. The most annoying part is that the error message basically gives you nothing more than “I fail at training classifer”. The tricky part is that the format of key has to be like directory which seqdirectory kind of makes sense. By like directory I mean like /key/row# instead of just key. Later I try to use /key and it turns out that it does not work.

3. The format of value seems not matter that much. I try delimiter data with “,” and “ ”. The result shows no difference. Interestingly, if I put no delimiter between each data column, it fails.
4. When making sequence files like this, I need to manually put my data into mushroom-seq file and run .sh file. I just don’t want to bother with more code to do this.

Then what left is straightforward, I just run mushroom.sh to get the result. I almost scream when it finally works. Complement Naive Bayes gives the same output as Naïve Bayes.



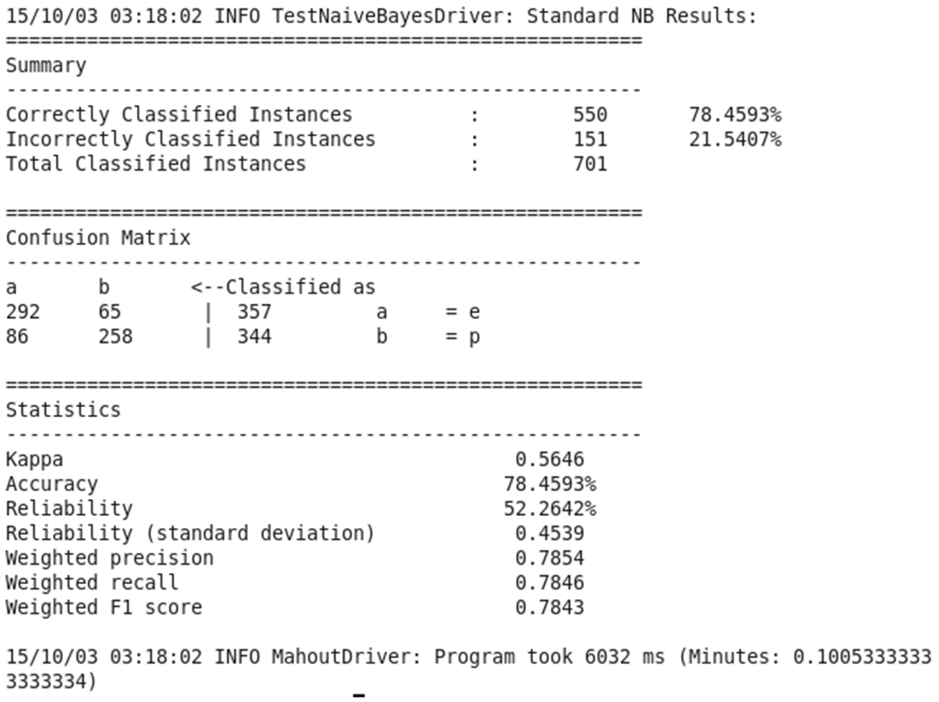
```
15/10/03 03:17:55 INFO TestNaiveBayesDriver: Standard NB Results:
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :       5948       80.1293%
Incorrectly Classified Instances        :       1475       19.8707%
Total Classified Instances              :       7423

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       <--Classified as
3302    549      |  3851        a     = e
926     2646     |  3572        b     = p

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.6001
Accuracy                                   80.1293%
Reliability                                53.2734%
Reliability (standard deviation)             0.465
Weighted precision                          0.8037
Weighted recall                             0.8013
Weighted F1 score                           0.8004

15/10/03 03:17:55 INFO MahoutDriver: Program took 7983 ms (Minutes: 0.13305)
```

Fig. 4. (Complement) Naive Bayes on mushroom dataset when self-test with the training vector files.



```
15/10/03 03:18:02 INFO TestNaiveBayesDriver: Standard NB Results:
=======================================================
Summary
-------------------------------------------------------
Correctly Classified Instances          :        550       78.4593%
Incorrectly Classified Instances        :        151       21.5407%
Total Classified Instances              :        701

=======================================================
Confusion Matrix
-------------------------------------------------------
a       b       <--Classified as
292     65       |  357         a     = e
86      258      |  344         b     = p

=======================================================
Statistics
-------------------------------------------------------
Kappa                                       0.5646
Accuracy                                   78.4593%
Reliability                                52.2642%
Reliability (standard deviation)            0.4539
Weighted precision                          0.7854
Weighted recall                             0.7846
Weighted F1 score                           0.7843

15/10/03 03:18:02 INFO MahoutDriver: Program took 6032 ms (Minutes: 0.1005333333
3333334)
```

Fig. 5. (Complement) Naive Bayes on mushroom dataset when test with the test vector files.

Table 1. Statistics of Naive Bayes on mushroom dataset

Statistics	Self-test with training vector files	Test with test vector files
Kappa	0.6001	0.5646
Correctly Classified	80.13%	78.46%
Incorrectly Classified	19.87%	21.54%
Reliability	53.27%	52.26%
Reliability(standard deviation)	0.465	0.4539
Weighted precision	0.8037	0.7854
Weighted recall	0.8013	0.7846
Weighted-F1 score	0.8004	0.7843

Thoughts

I spent 4 nights on this, seriously, from 10pm to 4am.

1. Documentation is really really insufficient for a beginner like me. When I try to browse the internet for answer, I cannot see any official documentation that describes how to tackle with different errors. The only instruction is the 20-news group example which in reality is not enough representative, because most of time we will have dataset with each row as a record instead of each file as a record. When I first time learn Java, I remember each time I have trouble I can go to the Javadoc website to get some hints. For mahout, the only Javadoc documentation I saw is made by cloudera and the format or maybe even the content is outdated. Honestly, I doubt if I download the source code and Javadoc each of them then upload to the website, I could make several hundred bucks each month.

2. Maybe because mahout covers so many algorithms that it could not give enough instructions for each algorithm. However, I cannot understand why it does not even give the most important ones like the format of sequence file keys and values. These things are very basic but I think they are very important for beginners who want to modify based on their example. Althoug I use seqdumper to see the content of 20-news group example sequence file and I see the format is like /foldername/count, how could I know it is strictly required to have that kind of format?

3. Maybe because mahout is an open source project which by nature restricts its development. Will it be possible to have an extremely user friendly website where user can just drag their datasets, choose the tasks and algorithms they want to perform, and then download the output statistics files?