

MSDS610 Week 7 Mahout Assignment - Nathan Worsham

Playing with Mahout's Spark Shell

I had trouble navigating the Mahout website for at first I could not find the "Quick Start" but in the "Playing with Mahout's Spark Shell" section it gave instructions for installing both Spark and Mahout. The installation in this section for Mahout is different than previous weeks as it says to clone a github repo. Later I did find the quick start and the normal place to download Mahout, but the installation from git isn't that much different. This meant first using `yum install git` to get the package to use git.

```
[hadoop@week1 ~]$ git clone https://github.com/apache/mahout mahout
-bash: git: command not found
[hadoop@week1 ~]$ yum install git
Loaded plugins: fastestmirror
You need to be root to perform this command.
[hadoop@week1 ~]$ exit
logout
[root@week1 ~]# yum install git
Loaded plugins: fastestmirror
base
extras
updates
(1/2): extras/7/x86_64/primary_db
(2/2): updates/7/x86_64/primary_db
Determining fastest mirrors
 * base: ftp.osuosl.org
 * extras: mirrors.gigenet.com
 * updates: mirror.lax.hugeserver.com
Resolving Dependencies
--> Running transaction check
--> Package git.x86_64 0:1.8.3.1-6.el7 will be installed
--> Processing Dependency: perl-Git > 1.8.3.1-6.el7 for package: git-1.8.3.1-6.el7.x86_64
--> Processing Dependency: rsync for package: git-1.8.3.1-6.el7.x86_64
```

The instructions say to create a directory for Mahout, change to that directory, then clone from git. But the git command it gives creates a folder anyway, so I elected to just run their command from the home directory of my hadoop user.

```
[hadoop@week1 ~]$ git clone https://github.com/apache/mahout mahout
Cloning into 'mahout':
remote: Counting objects: 93288, done.
remote: Total 93288 (delta 8), reused 0 (delta 0), pack-reused 93288
Receiving objects: 100% (93288/93288), 46.72 MiB | 769.08 KiB/s, done.
Resolving deltas: 100% (52531/52531), done.
[hadoop@week1 ~]$ ls
apache-hive-1.2.1-bin  hadoopdata  hive          ml-108k      pig-0.15.0
derby.log             hbase       mahout        ml-100k.zip  pig_1455010061684.log
hadoop               hbase-1.1.2  metastore_db  pig         pig_test
```

Now that I had the files from git I was ready to build Mahout with the `mvn` command which to no surprise was not installed. So again I switched to root and installed it—I ended up needing to use `yum provides mvn` to find that it was part of the "maven" package.

```
[hadoop@week1 mahout]$ mvn -DskipTests clean install
-bash: mvn: command not found
[hadoop@week1 mahout]$ exit
logout
[root@week1 ~]# yum install mvn
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.osuosl.org
 * extras: mirrors.gigenet.com
 * updates: mirror.lax.hugeserver.com
No package mvn available.
Error: Nothing to do
[root@week1 ~]# yum provides mvn
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.osuosl.org
 * extras: mirrors.gigenet.com
 * updates: mirror.lax.hugeserver.com
extras/7/x86_64/filelists_db
updates/7/x86_64/filelists_db
maven-3.0.5-16.el7.noarch : Java project management and project comprehension tool
Repo:
: base
Matched from:
Filename: /usr/bin/mvn

[root@week1 ~]# yum install maven
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
```

Trying to build using the maven package, I received a build error. After looking at a stackoverflow.com thread (2015), I realized that the maven I installed was an older version and that Mahout has a requirement for a newer version. I went ahead and tried to update through yum but there were no updates for maven through the regular yum repos.

```
[WARNING] Rule 1: org.apache.maven.plugins.enforcer.RequireMavenVersion failed with message:
[ERROR] Detected Maven version 3.0.0 is not in the allowed range [3.3.0,3.5.0]
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] Mahout Build Tools ..... FAILURE [19.770s]
[INFO] Apache Mahout ..... SKIPPED
[INFO] Mahout Math ..... SKIPPED
[INFO] Mahout HDFS ..... SKIPPED
[INFO] Mahout Map-Reduce ..... SKIPPED
[INFO] Mahout Integration ..... SKIPPED
[INFO] Mahout Examples ..... SKIPPED
[INFO] Mahout Math Scala bindings ..... SKIPPED
[INFO] Mahout H2O backend ..... SKIPPED
[INFO] Mahout Spark bindings ..... SKIPPED
[INFO] Mahout Spark bindings shell ..... SKIPPED
[INFO] Mahout Release Package ..... SKIPPED
[INFO] -----
[INFO] BUILD FAILURE
[INFO] -----
[INFO] Total time: 20.428s
[INFO] Finished at: Mon Feb 22 11:47:20 MST 2016
[INFO] Final Memory: 120M/32M
[INFO] -----
[ERROR] Failed to execute goal org.apache.maven.plugins:maven-enforcer-plugin:1.4:enforce (enfo
ic messages explaining why the rule failed. -> [Help 1]
[ERROR]
[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.
[ERROR] Re-run Maven using the -X switch to enable full debug logging.
[ERROR]
[ERROR] For more information about the errors and possible solutions, please read the following
[ERROR] [Help 1] http://wiki.apache.org/confluence/display/Maven/RubyExecutionException
hadoop@week1 ~$ mvn
```

```
[root@week1 ~]# mvn -version
Apache Maven 3.0.5 (Red Hat 3.0.5-16)
Maven home: /usr/share/maven
Java version: 1.8.0_65, vendor: Oracle Corporation
Java home: /opt/jdk1.8.0_65/jre
Default locale: en_US, platform encoding: UTF-8
OS name: "linux", version: "3.10.0-327.4.4.el7.x86_64", arch: "amd64", family: "unix"
[root@week1 ~]# yum update maven
Loaded plugins: fastestmirror
Loading mirror speeds from cached hostfile
 * base: ftp.osuosl.org
 * extras: mirrors.gigenet.com
 * updates: mirror.lax.hugeserver.com
No packages marked for update
```

So following another site (gluster.org, 2013) instructions for installing the latest maven through yum, I downloaded a new repo and then used `yum install apache-maven`. But again I received some errors, this time because it was conflicting with some the dependencies that were installed for the previous maven. I went ahead and ran `yum autoremove` to get rid of the old dependencies and then I was able to install `apache-maven`. The article then said to update symlinks for `mvn` but did not say how or where to do it. I went ahead and just created a single symbolic link to `/usr/local/bin/mvn` since I knew that was in the `PATH`. Now I was able to run the `mvn` command, a command that took a long time to complete, downloaded quite a bit of content and expressed several warning messages, but ultimately did say the build was a success.

```
Dependencies Resolved


=====
Package Arch
=====
Installing:
apache-maven noarch

Transaction Summary
=====
Install 1 Package

Total download size: 7.5 M
Installed size: 9.0 M
Is this ok [y/d/N]: y
Downloading packages:
apache-maven-3.3.3-4.el7.noarch.rpm
Running transaction check
Running transaction test
Transaction check error:
file /usr/share/java/plexus/plexus-cipher.jar from install of apache-ma
file /usr/share/java/plexus/plexus-sec-dispatcher.jar from install of ap
file /usr/share/java/maven-wagon/file.jar from install of apache-maven-0
file /usr/share/java/maven-wagon/http-shared.jar from install of apache-
file /usr/share/java/maven-wagon/provider-api.jar from install of apache-
```

```
[INFO] -----
[INFO] Reactor Summary:
[INFO]
[INFO] Mahout Build Tools ..... SUCCESS [ 36.541 s]
[INFO] Apache Mahout ..... SUCCESS [ 0.105 s]
[INFO] Mahout Math ..... SUCCESS [ 35.012 s]
[INFO] Mahout HDFS ..... SUCCESS [01:11 min]
[INFO] Mahout Map-Reduce ..... SUCCESS [ 25.001 s]
[INFO] Mahout Integration ..... SUCCESS [ 56.037 s]
[INFO] Mahout Examples ..... SUCCESS [ 23:247 s]
[INFO] Mahout Math Scala bindings ..... SUCCESS [01:12 min]
[INFO] Mahout H2O backend ..... SUCCESS [ 50.114 s]
[INFO] Mahout Spark bindings ..... SUCCESS [01:50 min]
[INFO] Mahout Spark bindings shell ..... SUCCESS [ 46.175 s]
[INFO] Mahout Release Package ..... SUCCESS [ 2.033 s]
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 08:49 min
[INFO] Finished at: 2016-02-22T12:13:34-07:00
[INFO] Final Memory: 75M/270M
[INFO] -----
[hadoop@week1 mahout]$
```

Next it had me start Spark and confirm by going to a local webpage—which is something I did not see last week. After trying for a bit to get to port 8080, I realized the local firewall was in the way and I elected to simply turn it off with `service firewalld stop`. After that I was to copy the URL in order to place it into an environment variable which I did.


Spark Master at spark://week1:7077

URL: spark://week1:7077
REST URL: spark://week1:6066 (cluster mode)
Alive Workers: 1
Cores in use: 1 Total, 0 Used
Memory in use: 1024.0 MB Total, 0.0 B Used
Applications: 0 Running, 0 Completed
Drivers: 0 Running, 0 Completed
Status: ALIVE

Workers	
Worker Id	Address
worker-20160222221518-10.0.2.15-54972	10.0.2.15:54972

Running Applications

Application ID	Name	Cores	Memory per Node
----------------	------	-------	-----------------

```
[hadoop@week1 ~]$ vi ~/.bashrc
[hadoop@week1 ~]$ tail -5 ~/.bashrc
export SPARK_LOCAL_IP=10.0.2.15

export MAHOUT_HOME=/home/hadoop/mahout
export SPARK_HOME=/home/hadoop/spark
export MASTER=spark://week1:7077
[hadoop@week1 ~]$ . ~/.bashrc
```

Now I was ready to try to get to the mahout `spark-shell`. The first time I tried it I received an error about trying to connect to localhost on port 9000, which made me realize I had better start up HDFS and yarn (as the instructions did not indicate but was maybe implied). After starting those services and trying

again I received a much better result because I got back the `mahout>` prompt. However I did receive a couple of warnings about plugins already being registered and failing to get a database default.

```
16/02/23 03:11:25 WARN ObjectStore: Version information not found in metastore: hive.metastore.schem
16/02/23 03:11:26 WARN ObjectStore: Failed to get database default, returning NoSuchObjectException
16/02/23 03:11:30 WARN General: Plugin (Bundle) "org.datanucleus" is already registered. Ensure you
op/spark-1.6.0-bin-hadoop2.6/lib/datanucleus-core-3.2.10.jar" is already registered, and you are try
ore-3.2.10.jar."
16/02/23 03:11:30 WARN General: Plugin (Bundle) "org.datanucleus.store.rdbms" is already registered.
e:/home/hadoop/spark-1.6.0-bin-hadoop2.6/lib/datanucleus-rdbms-3.2.9.jar" is already registered, and
atanucleus-rdbms-3.2.9.jar."
16/02/23 03:11:30 WARN General: Plugin (Bundle) "org.datanucleus.api.jdo" is already registered. Ens
use/hadoop/spark-1.6.0-bin-hadoop2.6/lib/datanucleus-api-jdo-3.2.6.jar" is already registered, and y
anucleus-api-jdo-3.2.6.jar."
SQL context available as "val sqlContext".
mahout>
```

The next step in the exercise was to setup a matrix that represented ingredients (protein, fat, carbohydrate and sugars in milligrams) of various cereals, along with a column of combined user ratings of the cereals. The purpose of the exercise is to fit a linear model which infers the customer rating from the ingredients using a linear regression algorithm. I created the matrix and pulled out the X and Y values into variables.

```
mahout> val drnData = drnParallelize(dense(
| (2, 2, 10.5, 10, 29.509541), // Apple Cinnamon Cheerios
| (1, 2, 12, 12, 18.042051), // Cap'n Crunch
| (1, 1, 12, 13, 22.736446), // Cocoa Puffs
| (2, 1, 11, 13, 32.207502), // Frost Loops
| (1, 2, 12, 11, 21.071292), // Honey Graham Ohs
| (2, 1, 16, 8, 36.107559), // Wheaties Honey Gold
| (6, 2, 17, 1, 50.764999), // Cheerios
| (3, 2, 13, 7, 40.400200), // Clusters
| (3, 3, 13, 4, 45.011716), // Great Grains Pecan
| numPartitions = 2);
drnData: org.apache.mahout.math.drm.CheckpointedDrm[Int] = org.apache.mahout.sparkbindings.drm.CheckpointedDrmSpark@9facde
mahout> drnData
res0: org.apache.mahout.math.drm.CheckpointedDrm[Int] = org.apache.mahout.sparkbindings.drm.CheckpointedDrmSpark@9facde
mahout> print (drnData)
org.apache.mahout.sparkbindings.drm.CheckpointedDrmSpark@9facde:mahout>
mahout> val drnX = drnData.collect(0 until 4)
drnX: org.apache.mahout.math.drm.DrmLike[Int] = OpMapBlock(org.apache.mahout.sparkbindings.drm.CheckpointedDrmSpark@9facde,<function>,4,-1,true)
mahout> val y = drnData.collect(0, 4)
y: org.apache.mahout.math.Vector = (0:29.509541,1:10.042051,2:22.736446,3:32.207502,4:21.071292,5:36.107559,6:50.764999,7:40.400200,8:45.011716)
mahout>
```

Using Ordinary Least Squares, I then created variables containing matrix multiplication so that the two variables could then be multiplied to find "beta". What is interesting is after making the variable the equation for each value, I then have to run a "collect" command in order to fetch the values into memory so that the solve function can be run on them.

```
mahout> val drnXtX = drnX.t %>% drnX
drnXtX: org.apache.mahout.math.drm.DrmLike[Int] = OpAB(OpAt(OpMapBlock(org.apache.mahout.sparkbindings.drm.CheckpointedDrmSpark@9facde,<function>,4,-1,true)),OpMapBlock(org.apache.mahout.sparkbindings.drm.CheckpointedDrmSpark@9facde,<function>,4,-1,true))
mahout> val drnXty = drnX.t %>% y
drnXty: org.apache.mahout.math.drm.DrmLike[Int] = OpAX(OpAt(OpMapBlock(org.apache.mahout.sparkbindings.drm.CheckpointedDrmSpark@9facde,<function>,4,-1,true)),OpMapBlock(org.apache.mahout.sparkbindings.drm.CheckpointedDrmSpark@9facde,<function>,4,-1,true))
mahout> val XtX = drnXtX.collect
XtX: org.apache.mahout.math.Matrix =
{
0 => (0:69.0,1:40.0,2:291.0,3:137.0)
1 => (0:40.0,1:32.0,2:207.0,3:120.0)
2 => (0:291.0,1:207.0,2:1546.25,3:968.0)
3 => (0:137.0,1:120.0,2:968.0,3:633.0)
}
mahout> val Xty = drnXty.collect(0)
Xty: org.apache.mahout.math.Vector = (0:021.6057190800001,1:549.744517,2:3970.7015094999995,3:2272.779989)
mahout> val beta = solve(XtX, Xty)
beta: org.apache.mahout.math.Vector = (0:5.247349465370446,1:2.750794979407531,2:1.1527013010791554,3:0.10312017617400000)
```

Now I was to check how well the model fits by first multiplying the ingredient features by the estimated beta (yFitted), and then by looking at the difference between actual y and fitted y using something called L2-norm. I receive an answer of about 14.2. Looking for information on L2-norm brought me to a blog post (Rorasa, 2015) that essentially stated that it is also known as Euclidean distance.

```
mahout> val yFitted = (drnX %>% beta).collect(0)
yFitted: org.apache.mahout.math.Vector = (0:29.131693510783975,1:25.019756349376444,2:23.172001947004997,3:27.26665011304207,4:25.716636173200357,5:32.514955735899626,6:56.68688824372747,7:36.95163570033205,8:39.393069750271316)
mahout> (y - yFitted).norm(2)
res2: Double = 14.200396725608045
```

The next part in the exercise was to refactor everything that was just done into two functions and then also add a constant bias term to the model. The result was a goodness of fit of nearly half of what it previously was at approximately 7.6.

```

mahout> def ols(drx: DrMLike[Int], y: Vector) =
  | solve(drx.t %*% drx, drx.t %*% y)(::, 0)
ols: (drx: org.apache.mahout.math.dr.DrMLike[Int], y: org.apache.mahout.math.Vector)org.apache.mahout.math.Vector
mahout> def goodnessOfFit(drx: DrMLike[Int], beta: Vector, y: Vector) = {
  | val fitted = (drx %*% beta).collect(::, 0)
  | (y - fitted).norm(2)
  | }
goodnessOfFit: (drx: org.apache.mahout.math.dr.DrMLike[Int], beta: org.apache.mahout.math.Vector, y: org.apache.mahout.math.Vector)Double
mahout> val drxWithBiasColumn = drx.cbind(1)
drxWithBiasColumn: org.apache.mahout.math.dr.DrMLike[Int] = OpCbindScalar(OpMapBlock(org.apache.mahout.sparkbindings.dr.CheckpointedDrSpark@6ffade, function: 4, 1, true), 1, 0, false)
mahout> val betaWithBiasTerm = ols(drxWithBiasColumn, y)
betaWithBiasTerm: org.apache.mahout.math.Vector = (0,-1.3362653883272289,1,-13.15770132067483,2,-4.152654199020216,3,-5.6799080
94232256,4,163.1793268704127)
mahout> goodnessOfFit(drxWithBiasColumn, betaWithBiasTerm, y)
res3: Double = 7.623288714561956

```

Last the exercise was to implement a caching functionality to cache the value of the bias column into memory. This was for improved performance as it resulted in the same outcome.

```

mahout> val cachedDrx = drxWithBiasColumn.checkpoint()
cachedDrx: org.apache.mahout.math.dr.CheckpointedDrMLike[Int] = org.apache.mahout.sparkbindings.dr.CheckpointedDrSpark@63af522b
mahout> val betaWithBiasTerm = ols(cachedDrx, y)
betaWithBiasTerm: org.apache.mahout.math.Vector = (0,-1.3362653883272289,1,-13.15770132067483,2,-4.152654199020216,3,-5.6799080
94232256,4,163.1793268704127)
mahout> val goodness = goodnessOfFit(cachedDrx, betaWithBiasTerm, y)
goodness: Double = 7.623288714561956
mahout> cachedDrx.uncache()
res4: cachedDrx.type = org.apache.mahout.sparkbindings.dr.CheckpointedDrSpark@63af522b
mahout> goodness
res5: Double = 7.623288714561956

```

Twenty Newsgroups Classification Example

After completing the Spark shell exercise, I was really just interested in poking around the other exercises to see if anything looked interesting. I ended up trying the newsgroups classification exercise, which takes a dataset collection of about 20,000 newsgroup documents and classifies them into 20 different newsgroups. The exercise starts off with just having you run one of the examples contained within the Mahout folder. I went ahead and tried running the `./examples/bin/classify-20newsgroups.sh` script but I ended up receiving an error about the "temp/weights" folder not being publicly accessible. I took that to mean that it was not a world writable folder, so I tried `chmod 777` the rights of the folder and ran the example again but I received the same error.

```

at org.apache.mahout.driver.MahoutDriver.main(MahoutDriver.java:190)
Caused by: java.util.concurrent.ExecutionException: java.io.IOException: Resource file:/home/hadoop/mahout_git/temp/weights is
not publicly accessible and as such cannot be part of the public cache.
at java.util.concurrent.FutureTask.report(FutureTask.java:122)
at java.util.concurrent.FutureTask.get(FutureTask.java:192)
at org.apache.hadoop.mapred.LocalDistributedCacheManager.setup(LocalDistributedCacheManager.java:145)
... 21 more
Caused by: java.io.IOException: Resource file:/home/hadoop/mahout_git/temp/weights is not publicly accessible and as such cannot
be part of the public cache.
at org.apache.hadoop.yarn.util.FSDownload.copy(FSDownload.java:257)
at org.apache.hadoop.yarn.util.FSDownload.access$000(FSDownload.java:60)
at org.apache.hadoop.yarn.util.FSDownload$2.run(FSDownload.java:355)
at org.apache.hadoop.yarn.util.FSDownload$2.run(FSDownload.java:355)
at java.security.AccessController.doPrivileged(Native Method)
at java.security.auth.Subject.doAs(Subject.java:422)
at org.apache.hadoop.security.UserGroupInformation.doAs(UserGroupInformation.java:1556)
at org.apache.hadoop.yarn.util.FSDownload.call(FSDownload.java:352)
at org.apache.hadoop.yarn.util.FSDownload.call(FSDownload.java:59)
at java.util.concurrent.FutureTask.run(FutureTask.java:266)
at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1142)
at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:617)
at java.lang.Thread.run(Thread.java:745)
[[hadoop@week1 mahout]$ ls
bin      conf      doap Mahout.rdf  hdfs      math      mr      README.md  src
buildtools derby.log  examples  integration  math-scala NOTICE.txt spark      target
CHANGELOG distribution h2o      LICENSE.txt  metastore_db pom.xml   spark-shell temp
[[hadoop@week1 mahout]$ cd temp/
[[hadoop@week1 temp]$ ls -l
total 0
drwxr-xr-x. 2 hadoop hadoop 84 Feb 23 14:12 summedObservations
drwxr-xr-x. 2 hadoop hadoop 84 Feb 23 14:12 weights
[[hadoop@week1 temp]$ chmod 777 weights/
[[hadoop@week1 temp]$ cd ../
[[hadoop@week1 mahout]$ ./examples/bin/classify-20newsgroups.sh
Discovered Hadoop v2.
Setting dfs command to /home/hadoop/hadoop/bin/hdfs dfs -dfs rm to /home/hadoop/hadoop/bin/hdfs dfs -rm -r -skipTrash.

```

Even after opening the rights on the folder, when I ran the example again the folder's rights went back to the way they were (775), so the script must have a bug I'm guessing. At this point I decided to walk through the example as they gave step-by-step instructions for how the example script works. I first checked to see if I could find the file the script already downloaded but could not find it as it took awhile to download. I then set the environment variables accordingly and made the working directory.

```
[hadoop@week1 mahout]$ find . -name "*28news-hydate*"
[hadoop@week1 mahout]$ vi ~/.bashrc
[hadoop@week1 mahout]$ tail ~/.bashrc

export SPARK_DIST_CLASSPATH=$HADOOP_HOME
export SPARK_LOCAL_IP=10.0.2.15

export MAHOUT_HOME=/home/hadoop/mahout
export SPARK_HOME=/home/hadoop/spark
export MASTER=spark://week1:7077
export MAHOUT_LOCAL=true
export WORK_DIR=/tap/mahout-work-$(USER)
export PATH=$PATH:$MAHOUT_HOME/bin
[hadoop@week1 mahout]$ mkdir -p ${WORK_DIR}
mkdir: missing operand
Try 'mkdir --help' for more information.
[hadoop@week1 mahout]$ . ~/.bashrc
[hadoop@week1 mahout]$ mkdir -p ${WORK_DIR}
```

Next I re-downloaded the data set, unpacked it, and then tried to make another required directory that apparently already existed.

```
[hadoop@week1 mahout]$ curl http://people.csail.mit.edu/rremie/20NewsGroups/20news-bydate.tar.gz -o $(WORK_DIR)/20news-bydate.tar.gz
% Total    % Received % Xferd  Average Speed   Time    Time     Current
                                 Dload  Upload   Total    Spent    Left     Speed
100 13.7M  100 13.7M    0     0  67563      0  0:03:34  0:03:34  --:--:-- 129K
[hadoop@week1 mahout]$ mkdir -p $(WORK_DIR)/20news-bydate
[hadoop@week1 mahout]$ cd $(WORK_DIR)/20news-bydate && tar xzf ../20news-bydate.tar.gz && cd .. && cd ..
[hadoop@week1 tnp1]$ mkdir $(WORK_DIR)/20news-all
mkdir: cannot create directory '/tmp/mahout-work-hadoop/20news-all': File exists
```

The instructions had an optional section about trying to "put" a folder and contents into the HDFS, but it seems the command is deprecated and it complained of "No such file or directory". Given that it was optional, I decided to move on without it.

```
hadoop@week1 tmp$ cp -R $(WORK_DIR)/20news-bydate/* $(WORK_DIR)/20news-all
hadoop@week1 tmp$ hadoop dfs -put $(WORK_DIR)/20news-all $(WORK_DIR)/20news-all
DEPRECATED: Use of this script to execute hdfs command is deprecated.
Instead use the hdfs command for it.

put: '/tmp/mahout-work-hadoop/20news-all': No such file or directory
```

I then successfully ran the commands to convert the dataset into a sequence file and then a sequence file with term frequencies for each document.

```

16/02/23 14:32:17 INFO Job: map 100% reduce 0%
16/02/23 14:32:17 INFO Job: Job job_local1617541775_0001 completed successfully
16/02/23 14:32:17 INFO Job: Counters: 18
  File System Counters
    FILE: Number of bytes read=95444499
    FILE: Number of bytes written=79622251
    FILE: Number of read operations=0
    FILE: Number of large read operations=0
    FILE: Number of write operations=0
  Map-Reduce Framework
    Map input records=18846
    Map output records=18846
    Input split bytes=1494488
    Spilled Records=0
    Failed Shuffles=0
    Merged Map outputs=0
    GC time elapsed (ms)=234
    CPU time spent (ms)=0
    Physical memory (bytes) snapshot=0
    Virtual memory (bytes) snapshot=0
    Total committed heap usage (bytes)=49455104
  File Input Format Counters
    Bytes Read=0
  File Output Format Counters
    Bytes Written=19352679
16/02/23 14:32:17 INFO MahoutDriver: Program took 67881 ms (Minutes: 1.13135)

```

```

16/02/23 14:54:07 INFO Job: Job Job_Linux132752949_0000 completed successfully
16/02/23 14:54:07 INFO Job: Counters: 33
File System Counters
  FILE: Number of bytes read=2137567366
  FILE: Number of bytes written=2050102867
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
Map-Reduce Framework
  Map input records=10846
  Map output records=10846
  Map output bytes=20362595
  Map output materialized bytes=20437750
  Input split bytes=139
  Combine input records=0
  Combine output records=0
  Reduce input groups=10846
  Reduce shuffle bytes=20437750
  Reduce input records=10846
  Reduce output records=10846
  Spilled Records=37692
  Shuffled Maps=1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=127
  CPU time spent (ms)=0
  Physical memory (bytes) snapshot=0
  Virtual memory (bytes) snapshot=0
  Total committed heap usage (bytes)=196451288
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=28913427
File Output Format Counters
  Bytes Written=28913427
16/02/23 14:54:07 INFO HadoopUtil: Deleting /tmp/mahout-work-hadoop/20news-vectors/partial-vectors-0
16/02/23 14:54:07 INFO HadoopDriver: Program took 62078 ms (Minutes: 1.0345)
[hadoop@weel1 tmp]$

```

Split the dataset into training and testing sets.


```
[hadoop@week1 tap]$ mahout split -i $(WORK_DIR)/20news-vectors/termf-vectors --trainingOutput $(WORK_DIR)/20news-train-vectors)
--testOutput $(WORK_DIR)/20news-test-vectors --randomSelectionPct 40 --overwrite --sequenceFiles -xm sequential
MAHOUT_LOCAL is set, so we don't add HADOOP_CONF_DIR to classpath.
MAHOUT_LOCAL is set, running locally
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/mahout_git/examples/target/mahout-examples-0.11.2-SNAPSHOT-job.jar/org/slf4j/in
jl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/mahout_git/examples/target/dependency/slf4j-log4j12-1.7.12.jar/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
16/02/23 14:35:54 WARN MahoutDriver: No split.props found on classpath, will use command-line arguments only
16/02/23 14:35:54 INFO AbstractJob: Command line arguments: [--endPhase=12147403647] --input[/tap/mahout-work-hadoop/20news-v
ectors/termf-vectors] --method[sequential] --overwrite=null --randomSelectionPct[40] --sequenceFiles=null --startPhase[
0] --tempDir=/temp --testOutput[/tap/mahout-work-hadoop/20news-test-vectors] --trainingOutput[/tap/mahout-work-hadoop/20n
ews-train-vectors]
16/02/23 14:35:54 INFO HadoopUtil: Deleting /tap/mahout-work-hadoop/20news-train-vectors
16/02/23 14:35:54 INFO HadoopUtil: Deleting /tap/mahout-work-hadoop/20news-test-vectors
16/02/23 14:35:57 INFO SplitInput: part-r-00000 has 162419 lines
16/02/23 14:35:57 INFO SplitInput: part-r-00000 test split size is 64968 based on random selection percentage 40
16/02/23 14:35:57 INFO ZlibFactory: Successfully loaded & initialized native-zlib library
16/02/23 14:35:57 INFO CodecPool: Got brand-new compressor [deflate]
16/02/23 14:35:57 INFO CodecPool: Got brand-new compressor [deflate]
16/02/23 14:36:04 INFO SplitInput: file: part-r-00000, input: 162419 train: 11322, test: 7524 starting at 0
16/02/23 14:36:04 INFO MahoutDriver: Program took 10078 ms (Minutes: 0.16796666666666665)
[hadoop@week1 tap]$
```

When I tried to run the command to train the classifier I received an error complaining about the `-e1` option, so not really knowing what to do, I decided to simply remove the option which luckily still worked.

```
MAHOUT_LOCAL is set, so we don't add HADOOP_CONF_DIR to classpath.
MAHOUT_LOCAL is set, running locally
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/home/hadoop/mahout_git/examples/target/mahout-examples-0.11.2-SNAPSHOT-job.jar/org/slf4j/in
jl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/home/hadoop/mahout_git/examples/target/dependency/slf4j-log4j12-1.7.12.jar/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
16/02/23 14:36:20 WARN MahoutDriver: No train.props found on classpath, will use command-line arguments only
16/02/23 14:36:20 WARN MahoutDriver: No train.props found on classpath, will use command-line arguments only
Unexpected -e1 while processing Job-Specific Options:
  -input <input> --output <output> --label <label> --trainComplementary
  --labelIndex <labelIndex> --overwrite --help --tempDir <tempDir> --startPhase
  --endPhase <endPhase>
Job-Specific Options:
  -input <input> Path to job input directory.
  -output <output> The directory path where job output
  --label <label> Labeling parameter.
  --trainComplementary <tc> train complementary.
  --labelIndex <li> labelIndex The path to store the label index in
  --overwrite <ow> to overwrite the output directory
  --help <h> Print out help.
  --tempDir <tempDir> Intermediate output directory
  --startPhase <startPhase> First phase to run
  --endPhase <endPhase> Last phase to run
16/02/23 14:36:20 WARN MahoutDriver: Program took 10078 ms (Minutes: 0.16796666666666665)
```

```
16/02/23 14:38:45 INFO Job: Counters: 33
File System Counters
  FILE: Number of bytes read=422901078
  FILE: Number of bytes written=369750038
  FILE: Number of read operations=0
  FILE: Number of large read operations=0
  FILE: Number of write operations=0
Map-Reduce Framework
  Map input records=20
  Map output records=1
  Map output bytes=167
  Map output materialized bytes=105
  Input split bytes=111
  Combine input records=1
  Combine output records=1
  Reduce input groups=1
  Reduce shuffle bytes=105
  Reduce input records=1
  Reduce output records=1
  Spilled Records=2
  Shuffled Maps=1
  Failed Shuffles=0
  Merged Map outputs=1
  GC time elapsed (ms)=44
  CPU time spent (ms)=0
  Physical memory (bytes) snapshot=0
  Virtual memory (bytes) snapshot=0
  Total committed heap usage (bytes)=338029312
Shuffle Errors
  BAD_ID=0
  CONNECTION=0
  IO_ERROR=0
  WRONG_LENGTH=0
  WRONG_MAP=0
  WRONG_REDUCE=0
File Input Format Counters
  Bytes Read=2784969
File Output Format Counters
  Bytes Written=277
16/02/23 14:38:46 INFO MahoutDriver: Program took 13241 ms (Minutes: 0.22068333333333334)
```

Finally I was able to test the classifier and received the output that the example script did not supply.

```
16/02/23 14:40:36 INFO TestNaiveBayesDriver: Complementary Results:
=====
Summary
-----
Correctly Classified Instances      :    6686      88.8623%
Incorrectly Classified Instances   :     838      11.1377%
Total Classified Instances         :    7524

=====
Confusion Matrix
-----
a      b      c      d      e      f      g      h      i      j      k      l      m      n      o      p      q      r      s      t      <--Classified as
294    1      0      1      1      0      0      0      0      0      0      1      0      2      5      0      0      1      11
2      320    2      12      6      14      7      4      0      1      2      10      2      1      6      1      4      1      3      1
4      17      235    52      5      13      11      3      3      3      7      10      3      7      3      1      2      4      3
1      9      4      331    8      5      17      3      1      2      4      2      12      6      5      4      1      1      3
0      1      3      3      335    3      3      3      1      2      1      2      4      2      1      1      0      1      2      2
1      11    1      5      3      343    1      3      3      3      0      4      1      2      3      0      1      2      0      1
0      3      1      23      7      1      303    13      2      3      3      3      14      3      1      0      4      1      5      3
0      1      1      3      1      0      0      4      3      3      0      1      3      0      0      2      1      3      0      0
0      1      0      0      0      0      1      4      397    0      1      0      0      0      0      0      2      0      0      0
0      0      0      0      1      0      0      1      1      390    1      1      3      2      0      1      0      0      0      1
0      1      0      0      0      0      1      1      0      4      372    0      0      0      0      0      0      0      0      0
1      0      0      0      0      0      1      0      0      0      0      397    2      0      0      0      0      0      3      1
0      2      0      9      6      1      10      5      5      3      1      3      329    1      5      2      1      1      2      3
2      1      2      0      0      1      1      2      0      1      1      1      4      367    3      2      3      0      3      2
1      2      0      0      0      0      0      1      0      0      0      1      3      2      399    0      3      3      3      3
3      0      0      0      0      0      0      2      0      2      0      2      0      1      1      371    0      1      2      3
0      0      0      0      0      0      0      0      0      1      0      4      0      2      1      0      358    2      10      0
3      0      0      0      0      0      0      0      0      1      0      0      0      0      0      3      0      358    0      0
2      0      0      0      0      1      0      1      0      1      3      1      0      0      0      0      12      6      261    2
33      2      0      0      0      0      0      0      0      0      0      0      0      1      4      23      2      0      5      179

Statistics
-----
Kappa      0.8533
Accuracy   88.8623%
Reliability 84.4187%
Reliability (standard deviation) 0.2176
Weighted precision 0.8891
Weighted recall 0.8886
Weighted F1 score 0.8863
16/02/23 14:40:36 INFO MahoutDriver: Program took 10642 ms (Minutes: 0.17736666666666667)
```

References

stackoverflow.com, 2015. Retrieved from <http://stackoverflow.com/questions/31968515/detected-maven-version-3-0-5-is-not-in-the-allowed-range-3-2>

rorasa.wordpress.com, 2015. Retrieved from <https://rorasa.wordpress.com/2012/05/13/l0-norm-l1-norm-l2-norm-l-infinity-norm/>

mahout.apache.org, 2016. Retrieved from <http://mahout.apache.org/users/sparkbindings/play-with-shell.html> and <http://mahout.apache.org/users/classification/twenty-newsgroups.html>