Music Data Analysis

In order to proceed we need to setup the data first. Since this is not production mode and we are working in pseudo-distributed mode, we will directly use the provided datasets.

Below shows the data received from web server, which is in XML format. We have stored the file in below location- **/home/acadgild/project/web**

```
[acadgild@localhost project]$ ls -l
total 12
drwxrwxr-x. 2 acadgild acadgild 4096 Jan 15 01:14 mob
drwxrwxr--. 2 acadgild acadgild 4096 Jan 15 00:48 scripts
drwxrwxr-x. 2 acadgild acadgild 4096 Jan 15 01:14 web
[acadgild@localhost project]$ cd web
[acadgild@localhost web]$ ls -l
total 8
-rw-rw-r--. 1 acadgild acadgild 6716 Jan 15 01:17 file.xml
[acadgild@localhost web]$ cat file.xml
<records>
<record>
<user_id>U106</user_id>
<song_id>S205</song_id>
<artist_id>A300</artist_id>
<timestamp>2016-05-10 12:24:22</timestamp>
<start_ts>2016-05-10 12:24:22</start_ts>
<end_ts>2017-05-09 08:09:22</end_ts>
<geo_cd>AP</geo_cd>
<station_id>ST407</station_id>
<song_end_type>2</song_end_type>
<like>1</like>
<dislike>1</dislike>
</record>
<record>
<user_id>U114</user_id>
<song_id>S209</song_id>
<artist_id>A303</artist_id>
<timestamp>2016-06-09 22:12:36</timestamp>
<start_ts>2016-05-10 12:24:22</start_ts>
<end_ts>2017-05-09 08:09:22</end_ts>
<geo_cd>U</geo_cd>
```

Below shows the data received from mobile server, which is in text format. We have stored the file in below location- **/home/acadgild/project/mob**

```
[acadgild@localhost mob]$ pwd
/home/acadgild/project/mob
[acadgild@localhost mob]$ cat file.txt
U114,S207,A303,1465130523,1465230523,1475130523,A,ST415,3,1,0
U107,S202,A303,1495130523,1465230523,1465230523,U,ST415,0,1,1
U100,S204,A302,1495130523,1475130523,1465130523,AU,ST408,2,1,1
U104,S202,A303,1465230523,1475130523,1465130523,A,ST409,2,0,1
U102,S207,A301,1465230523,1485130523,1465230523,AU,ST403,3,1,1
,S203,A302,1495130523,1475130523,1465230523,E,ST400,0,0,1
U106,S202,A302,1465230523,1465130523,1465130523,AU,ST408,0,1,1
U105,S207,A300,1465230523,1485130523,1465130523,U,ST400,2,0,1
U108,S205,A304,1465130523,1465130523,1475130523,,ST410,2,1,0
U105,S203,,1475130523,1465230523,1465130523,AU,ST408,2,0,1
U110,S203,A300,1465230523,1465130523,1485130523,A,ST415,0,1,1
U113,S200,A303,1465230523,1475130523,1465130523,E,ST413,3,1,1
U119,S208,A302,1495130523,1465230523,1465230523,U,ST415,3,0,0
U118,S208,A303,1475130523,1465130523,1465230523,E,ST415,3,0,0
U107,S210,A302,1475130523,1485130523,1485130523,AP,ST404,2,1,0
U118,S202,A300,1495130523,1465230523,1465230523,AP,ST410,1,0,0
U111,S206,A305,1465130523,1465130523,1485130523,AU,ST415,0,1,1
U116,S208,A303,1465230523,1485130523,1475130523,A,ST413,1,0,1
U101,S202,A300,1465230523,1465130523,1475130523,U,ST401,0,0,1
U120,S206,A303,1495130523,1485130523,1465130523,AU,ST414,0,0,0
[acadgild@localhost mob]$
```

Now we are using below script to start the services-

```
#!/bin/bash

if [ -f "/home/acadgild/project/logs/current-batch.txt" ]
then
 echo "Batch File Found!"
else
 echo -n "1" > "/home/acadgild/project/logs/current-batch.txt"
fi

chmod 775 /home/acadgild/project/logs/current-batch.txt
batchid=`cat /home/acadgild/project/logs/current-batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_$batchid
```

```
echo "Starting daemons" >> $LOGFILE

start-all.sh
start-hbase.sh
mr-jobhistory-daemon.sh start historyserver
```

Same can be seen in below screenshot-

```
[acadgild@localhost scripts]$ pwd
/home/acadgild/project/scripts
[acadgild@localhost scripts]$ cat start-daemons.sh
#!/bin/bash

if [ -f "/home/acadgild/project/logs/current-batch.txt" ]
then
 echo "Batch File Found!"
else
 echo -n "1" > "/home/acadgild/project/logs/current-batch.txt"
fi

chmod 775 /home/acadgild/project/logs/current-batch.txt
batchid=`cat /home/acadgild/project/logs/current-batch.txt`
LOGFILE=/home/acadgild/project/logs/log_batch_$batchid

echo "Starting daemons" >> $LOGFILE

start-all.sh
start-hbase.sh
mr-jobhistory-daemon.sh start historyserver
```

```
[acadgild@localhost project]$ sh /home/acadgild/project/scripts/start-daemons.sh
This script is Deprecated. Instead use start-dfs.sh and start-yarn.sh
18/01/15 01:42:09 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Starting namenodes on [localhost]
localhost: starting namenode, logging to /usr/local/hadoop-2.6.0/logs/hadoop-acadgild-namenode-localhost.localdomain.out
localhost: starting datanode, logging to /usr/local/hadoop-2.6.0/logs/hadoop-acadgild-datanode-localhost.localdomain.out
Starting secondary namenodes [0.0.0.0]
0.0.0.0: starting secondarynamenode, logging to /usr/local/hadoop-2.6.0/logs/hadoop-acadgild-secondarynamenode-localhost.localdomain.out
18/01/15 01:42:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
starting yarn daemons
starting resourcemanager, logging to /usr/local/hadoop-2.6.0/logs/yarn-acadgild-resourcemanager-localhost.localdomain.out
localhost: starting nodemanager, logging to /usr/local/hadoop-2.6.0/logs/yarn-acadgild-nodemanager-localhost.localdomain.out
starting master, logging to /usr/local/hbase/logs/hbase-acadgild-master-localhost.localdomain.out
starting historyserver, logging to /usr/local/hadoop-2.6.0/logs/mapred-acadgild-historyserver-localhost.localdomain.out
[acadgild@localhost project]$
```

We can see that the all services have been started-

```
[acadgild@localhost project]$ jps
3872 ResourceManager
3713 SecondaryNameNode
4515 JobHistoryServer
4820 Jps
3512 DataNode
3979 NodeManager
4412 HMaster
3407 NameNode
[acadgild@localhost project]$
```

We are also maintaining logs at location- **/home/acadgild/project/logs** for each batch run-

```
[acadgild@localhost project]$ cd logs/
[acadgild@localhost logs]$ ls
current-batch.txt  log_batch_1
[acadgild@localhost logs]$ cat current-batch.txt
1[acadgild@localhost logs]$ cat log_batch_1
Starting daemons
[acadgild@localhost logs]$
```

We have kept the lookup files at below location- **/home/acadgild/project/lookupfiles**, which will be used to create the lookup tables in hbase-

```
[acadgild@localhost project]$ cd lookupfiles
[acadgild@localhost lookupfiles]$ ls -l
total 16
-rw-rw-r--. 1 acadgild acadgild 100 Jan 15 01:57 song-artist.txt
-rw-rw-r--. 1 acadgild acadgild 125 Jan 15 01:57 stn-geocd.txt
-rw-rw-r--. 1 acadgild acadgild 240 Jan 15 01:57 user-artist.txt
-rw-rw-r--. 1 acadgild acadgild 405 Jan 15 01:57 user-subscn.txt
[acadgild@localhost lookupfiles]$
```

Below is the script used to create the lookup tables in hbase from above mentioned files-

```
#!/bin/bash

batchid=`cat /home/acadgild/project/logs/current-batch.txt`
```

```
LOGFILE=/home/acadgild/project/logs/log_batch_$batchid

echo "Creating LookUp Tables" >> $LOGFILE

echo "create 'station-geo-map', 'geo'" | hbase shell
echo "create 'subscribed-users', 'subscn'" | hbase shell
echo "create 'song-artist-map', 'artist'" | hbase shell

echo "Populating LookUp Tables" >> $LOGFILE

file="/home/acadgild/project/lookupfiles/stn-geocd.txt"
while IFS= read -r line
do
 stnid=`echo $line | cut -d',' -f1`
 geocd=`echo $line | cut -d',' -f2`
 echo "put 'station-geo-map', '$stnid', 'geo:geo_cd', '$geocd'" | hbase shell
done <"$file"

file="/home/acadgild/project/lookupfiles/song-artist.txt"
while IFS= read -r line
do
 songid=`echo $line | cut -d',' -f1`
 artistid=`echo $line | cut -d',' -f2`
 echo "put 'song-artist-map', '$songid', 'artist:artistid', '$artistid'" | hbase shell
done <"$file"

file="/home/acadgild/project/lookupfiles/user-subscn.txt"
while IFS= read -r line
do
 userid=`echo $line | cut -d',' -f1`
 startdt=`echo $line | cut -d',' -f2`
 enddt=`echo $line | cut -d',' -f3`
 echo "put 'subscribed-users', '$userid', 'subscn:startdt', '$startdt'" | hbase shell
 echo "put 'subscribed-users', '$userid', 'subscn:enddt', '$enddt'" | hbase shell
done <"$file"

hive -f /home/acadgild/project/scripts/user-artist.hql
```

So basically we are creating 3 tables in hbase namely- **station-geo-map, subscribed-users, song-artist-map** and populating those using corresponding lookup files-

```
[acadgild@localhost project]$ cat /home/acadgild/project/scripts/populate-lookup.sh
#!/bin/bash

batchid=`cat /home/acadgild/project/logs/current-batch.txt`

LOGFILE=/home/acadgild/project/logs/log_batch_$batchid

echo "Creating LookUp Tables" >> $LOGFILE

echo "create 'station-geo-map', 'geo'" | hbase shell
echo "create 'subscribed-users', 'subscn'" | hbase shell
echo "create 'song-artist-map', 'artist'" | hbase shell


echo "Populating LookUp Tables" >> $LOGFILE

file="/home/acadgild/project/lookupfiles/stn-geocd.txt"
while IFS= read -r line
do
 stnid=`echo $line | cut -d',' -f1`
 geocd=`echo $line | cut -d',' -f2`
 echo "put 'station-geo-map', '$stnid', 'geo:geo_cd', '$geocd'" | hbase shell
done <"$file"


file="/home/acadgild/project/lookupfiles/song-artist.txt"
while IFS= read -r line
do
 songid=`echo $line | cut -d',' -f1`
 artistid=`echo $line | cut -d',' -f2`
 echo "put 'song-artist-map', '$songid', 'artist:artistid', '$artistid'" | hbase shell
done <"$file"


file="/home/acadgild/project/lookupfiles/user-subscn.txt"
while IFS= read -r line
do
 userid=`echo $line | cut -d',' -f1`
```

```
 startdt=`echo $line | cut -d',' -f2`
 enddt=`echo $line | cut -d',' -f3`
 echo "put 'subscribed-users', '$userid', 'subscn:startdt', '$startdt'" | hbase shell
 echo "put 'subscribed-users', '$userid', 'subscn:enddt', '$enddt'" | hbase shell
done <"$file"

hive -f /home/acadgild/project/scripts/user-artist.hql
```

We are also running below mentioned **user-artist.hql** to create a database in hive named **project** and table named **users_artists-**

```
[acadgild@localhost project]$ cat /home/acadgild/project/scripts/user-artist.hql
CREATE DATABASE IF NOT EXISTS project;

USE project;

CREATE TABLE users_artists
(
user_id STRING,
artists_array ARRAY<STRING>
)
ROW FORMAT DELIMITED
FIELDS TERMINATED BY ','
COLLECTION ITEMS TERMINATED BY '&';

LOAD DATA LOCAL INPATH '/home/acadgild/project/lookupfiles/user-artist.txt'
OVERWRITE INTO TABLE users_artists;
```

```
[acadgild@localhost project]$ sh /home/acadgild/project/scripts/populate-lookup.sh
2018-01-15 04:08:35,641 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015

create 'station-geo-map', 'geo'
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerB
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2018-01-15 04:08:38,651 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
0 row(s) in 2.4200 seconds

Hbase::Table - station-geo-map
2018-01-15 04:08:45,694 INFO  [main] Configuration.deprecation: hadoop.native.lib is deprecated. Instead, use io.native.lib.available
HBase Shell; enter 'help<RETURN>' for list of supported commands.
Type "exit<RETURN>" to leave the HBase Shell
Version 0.98.14-hadoop2, r4e4aabb93b52f1b0fef6b66edd06ec8923014dec, Tue Aug 25 22:35:44 PDT 2015

create 'subscribed-users', 'subscn'
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hbase/lib/slf4j-log4j12-1.6.4.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/StaticLoggerB
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
2018-01-15 04:08:48,659 WARN  [main] util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-jav
0 row(s) in 2.2170 seconds
```

So below are the lookup tables which have been created in hbase-

```
hbase(main):025:0> list
TABLE
song-artist-map
station-geo-map
subscribed-users
3 row(s) in 0.0100 seconds

=> ["song-artist-map", "station-geo-map", "subscribed-users"]
hbase(main):026:0>
```

Below are the contents of lookup table song-artist-map-

```
hbase(main):026:0> scan 'song-artist-map'
ROW                                   COLUMN+CELL
 S200                                 column=artist:artistid, timestamp=1515969693975, value=A300
 S201                                 column=artist:artistid, timestamp=1515969703495, value=A301
 S202                                 column=artist:artistid, timestamp=1515969712798, value=A302
 S203                                 column=artist:artistid, timestamp=1515969722426, value=A303
 S204                                 column=artist:artistid, timestamp=1515969731956, value=A304
 S205                                 column=artist:artistid, timestamp=1515969741407, value=A301
 S206                                 column=artist:artistid, timestamp=1515969750828, value=A302
 S207                                 column=artist:artistid, timestamp=1515969760354, value=A303
 S208                                 column=artist:artistid, timestamp=1515969769801, value=A304
 S209                                 column=artist:artistid, timestamp=1515969779610, value=A305
10 row(s) in 0.1010 seconds
```

Below are the contents of lookup table station-geo-map-

```
hbase(main):027:0> scan 'station-geo-map'
ROW                                   COLUMN+CELL
 ST400                                column=geo:geo_cd, timestamp=1515969551105, value=A
 ST401                                column=geo:geo_cd, timestamp=1515969560367, value=AU
 ST402                                column=geo:geo_cd, timestamp=1515969569704, value=AP
 ST403                                column=geo:geo_cd, timestamp=1515969579263, value=J
 ST404                                column=geo:geo_cd, timestamp=1515969588939, value=E
 ST405                                column=geo:geo_cd, timestamp=1515969598285, value=A
 ST406                                column=geo:geo_cd, timestamp=1515969607892, value=AU
 ST407                                column=geo:geo_cd, timestamp=1515969617463, value=AP
 ST408                                column=geo:geo_cd, timestamp=1515969626817, value=E
 ST409                                column=geo:geo_cd, timestamp=1515969636256, value=E
 ST410                                column=geo:geo_cd, timestamp=1515969645533, value=A
 ST411                                column=geo:geo_cd, timestamp=1515969655851, value=A
 ST412                                column=geo:geo_cd, timestamp=1515969665699, value=AP
 ST413                                column=geo:geo_cd, timestamp=1515969675168, value=J
 ST414                                column=geo:geo_cd, timestamp=1515969684470, value=E
15 row(s) in 0.0600 seconds
```

Below are the contents of lookup table subscribed-users-

```
hbase(main):028:0> scan 'subscribed-users'
ROW                                   COLUMN+CELL
 U100                                 column=subscn:enddt, timestamp=1515969798883, value=1465130523
 U100                                 column=subscn:startdt, timestamp=1515969789338, value=1465230523
 U101                                 column=subscn:enddt, timestamp=1515969818276, value=1475130523
 U101                                 column=subscn:startdt, timestamp=1515969808641, value=1465230523
 U102                                 column=subscn:enddt, timestamp=1515969837175, value=1475130523
 U102                                 column=subscn:startdt, timestamp=1515969827640, value=1465230523
 U103                                 column=subscn:enddt, timestamp=1515969856415, value=1475130523
 U103                                 column=subscn:startdt, timestamp=1515969846941, value=1465230523
 U104                                 column=subscn:enddt, timestamp=1515969875370, value=1475130523
 U104                                 column=subscn:startdt, timestamp=1515969865787, value=1465230523
 U105                                 column=subscn:enddt, timestamp=1515969894637, value=1475130523
 U105                                 column=subscn:startdt, timestamp=1515969885024, value=1465230523
 U106                                 column=subscn:enddt, timestamp=1515969913525, value=1485130523
 U106                                 column=subscn:startdt, timestamp=1515969903978, value=1465230523
 U107                                 column=subscn:enddt, timestamp=1515969932904, value=1455130523
 U107                                 column=subscn:startdt, timestamp=1515969923265, value=1465230523
 U108                                 column=subscn:enddt, timestamp=1515969952254, value=1465230623
 U108                                 column=subscn:startdt, timestamp=1515969942619, value=1465230523
 U109                                 column=subscn:enddt, timestamp=1515969971096, value=1475130523
 U109                                 column=subscn:startdt, timestamp=1515969961605, value=1465230523
 U110                                 column=subscn:enddt, timestamp=1515969990258, value=1475130523
 U110                                 column=subscn:startdt, timestamp=1515969980685, value=1465230523
 U111                                 column=subscn:enddt, timestamp=1515970009170, value=1475130523
 U111                                 column=subscn:startdt, timestamp=1515969999794, value=1465230523
 U112                                 column=subscn:enddt, timestamp=1515970028607, value=1475130523
 U112                                 column=subscn:startdt, timestamp=1515970018586, value=1465230523
 U113                                 column=subscn:enddt, timestamp=1515970048653, value=1485130523
 U113                                 column=subscn:startdt, timestamp=1515970038636, value=1465230523
 U114                                 column=subscn:enddt, timestamp=1515970068388, value=1468130523
 U114                                 column=subscn:startdt, timestamp=1515970058378, value=1465230523
15 row(s) in 0.1000 seconds
```

We can also see that a database named **“project”** has been created in hive-

```
hive> show databases;
OK
b1
custom
default
demo
employees
olympics
project
rakesh
Time taken: 0.022 seconds, Fetched: 8 row(s)
hive>
```

Also a table named users_artists has been created in hive-

```
hive> use project;
OK
Time taken: 0.025 seconds
hive> show tables;
OK
users_artists
Time taken: 0.037 seconds, Fetched: 1 row(s)
hive> select * from users_artists;
OK
U100    ["A300","A301","A302"]
U101    ["A301","A302"]
U102    ["A302"]
U103    ["A303","A301","A302"]
U104    ["A304","A301"]
U105    ["A305","A301","A302"]
U106    ["A301","A302"]
U107    ["A302"]
U108    ["A300","A303","A304"]
U109    ["A301","A303"]
U110    ["A302","A301"]
U111    ["A303","A301"]
U112    ["A304","A301"]
U113    ["A305","A302"]
U114    ["A300","A301","A302"]
Time taken: 0.484 seconds, Fetched: 15 row(s)
hive>
```

Now we need to format the data received from web server since it is in XML format. So we are running script **dataformatting.sh** which contains a pig script- **dataformatting.pig** which is formatting XML data from input XML file and storing it into HDFS in below location-

/user/acadgild/project/batch${batchid}/formattedweb/

```
[acadgild@localhost project]$
[acadgild@localhost project]$ sh /home/acadgild/project/scripts/dataformatting.sh
18/01/15 16:48:31 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/01/15 16:48:32 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /user/acadgild/project/batch1/web
18/01/15 16:48:33 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/01/15 16:48:34 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /user/acadgild/project/batch1/formattedweb
18/01/15 16:48:36 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/01/15 16:48:37 INFO fs.TrashPolicyDefault: Namenode trash configuration: Deletion interval = 0 minutes, Emptier interval = 0 minutes.
Deleted /user/acadgild/project/batch1/mob
18/01/15 16:48:38 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/01/15 16:48:41 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/01/15 16:48:43 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
18/01/15 16:48:46 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
```

Same can be seen in below screenshot-

```
[acadgild@localhost project]$ hadoop fs -ls /user/acadgild/project/batch1/formattedweb
18/01/15 16:58:19 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
Found 2 items
-rw-r--r--   1 acadgild supergroup          0 2018-01-15 16:57 /user/acadgild/project/batch1/formattedweb/_SUCCESS
-rw-r--r--   1 acadgild supergroup       1236 2018-01-15 16:57 /user/acadgild/project/batch1/formattedweb/part-m-00000
[acadgild@localhost project]$
```

```
[acadgild@localhost project]$ hadoop fs -cat /user/acadgild/project/batch1/formattedweb/part-m-00000
18/01/15 16:59:25 WARN util.NativeCodeLoader: Unable to load native-hadoop library for your platform..
U106,S205,A300,1462863262,1462863262,1494297562,AP,ST407,2,1,1
U114,S209,A303,1465490556,1462863262,1494297562,U,ST411,2,1,0
U113,S203,A304,1465490556,1465490556,1462863262,U,ST405,0,0,1
U108,S200,A302,1468094889,1462863262,1468094889,U,ST414,0,0,1
U102,S203,A305,1465490556,1465490556,1494297562,U,ST404,2,0,0
,S208,A300,1465490556,1494297562,1465490556,U,ST411,1,0,1
U115,S200,A300,1465490556,1494297562,1465490556,AU,ST404,3,0,0
U111,S204,A300,1465490556,1465490556,1468094889,U,ST410,3,1,1
U120,S201,A300,1494297562,1465490556,1468094889,,ST410,3,0,1
U113,S203,,1465490556,1465490556,1465490556,A,ST402,1,1,0
U109,S203,A304,1462863262,1494297562,1468094889,E,ST405,1,1,1
U110,S202,A303,1494297562,1494297562,1468094889,AU,ST402,2,1,0
U100,S200,A301,1494297562,1494297562,1494297562,AP,ST410,3,1,1
U101,S208,A300,1462863262,1468094889,1462863262,E,ST408,0,1,1
U106,S206,A300,1494297562,1465490556,1462863262,A,ST405,3,1,0
U107,S202,A304,1494297562,1468094889,1462863262,U,ST409,0,0,0
U103,S204,A300,1468094889,1494297562,1465490556,AU,ST411,2,1,0
U103,S202,A300,1465490556,1465490556,1465490556,A,ST415,2,1,1
U113,S203,A303,1462863262,1468094889,1494297562,U,ST408,2,0,0
U113,S204,A301,1494297562,1494297562,1465490556,E,ST415,3,0,1
[acadgild@localhost project]$
```

The script dataformatting.sh also contains a HQL file - **formatted_hive_load.hql** which is placing the data from above HDFS location to a new hive table **formatted_input**-

```
hive> select * from formatted_input;
OK
U114    S207    A303    1465130523      1465230523      1475130523      A       ST415   3       1       0       1
U107    S202    A303    1495130523      1465230523      1465230523      U       ST415   0       1       1       1
U100    S204    A302    1495130523      1475130523      1465130523      AU      ST408   2       1       1       1
U104    S202    A303    1465230523      1475130523      1465130523      A       ST409   2       0       1       1
U102    S207    A301    1465230523      1485130523      1465230523      AU      ST403   3       1       1       1
        S203    A302    1495130523      1475130523      1465230523      E       ST400   0       0       1       1
U106    S202    A302    1465230523      1465130523      1465130523      AU      ST408   0       1       1       1
U105    S207    A300    1465230523      1485130523      1465130523      U       ST400   2       0       1       1
U108    S205    A304    1465130523      1465130523      1475130523              ST410   2       1       0       1
U105    S203            1475130523      1465230523      1465130523      AU      ST408   2       0       1       1
U110    S203    A300    1465230523      1465130523      1485130523      A       ST415   0       1       1       1
U113    S200    A303    1465230523      1475130523      1465130523      E       ST413   3       1       1       1
U119    S208    A302    1495130523      1465230523      1465230523      U       ST415   3       0       0       1
U118    S208    A303    1475130523      1465130523      1465230523      E       ST415   3       0       0       1
U107    S210    A302    1475130523      1485130523      1485130523      AP      ST404   2       1       0       1
U118    S202    A300    1495130523      1465230523      1465230523      AP      ST410   1       0       0       1
U111    S206    A305    1465130523      1465130523      1485130523      AU      ST415   0       1       1       1
U116    S208    A303    1465230523      1485130523      1475130523      A       ST413   1       0       1       1
U101    S202    A300    1465230523      1465130523      1475130523      U       ST401   0       0       1       1
U120    S206    A303    1495130523      1485130523      1465130523      AU      ST414   0       0       0       1
U106    S205    A300    1462863262      1462863262      1494297562      AP      ST407   2       1       1       1
U114    S209    A303    1465490556      1462863262      1494297562      U       ST411   2       1       0       1
U113    S203    A304    1465490556      1465490556      1462863262      U       ST405   0       0       1       1
U108    S200    A302    1468094889      1462863262      1468094889      U       ST414   0       0       1       1
U102    S203    A305    1465490556      1465490556      1494297562      U       ST404   2       0       0       1
        S208    A300    1465490556      1494297562      1465490556      U       ST411   1       0       1       1
U115    S200    A300    1465490556      1494297562      1465490556      AU      ST404   3       0       0       1
U111    S204    A300    1465490556      1465490556      1468094889      U       ST410   3       1       1       1
U120    S201    A300    1494297562      1465490556      1468094889              ST410   3       0       1       1
U113    S203            1465490556      1465490556      1465490556      A       ST402   1       1       0       1
U109    S203    A304    1462863262      1494297562      1468094889      E       ST405   1       1       1       1
U110    S202    A303    1494297562      1494297562      1468094889      AU      ST402   2       1       0       1
U100    S200    A301    1494297562      1494297562      1494297562      AP      ST410   3       1       1       1
U101    S208    A300    1462863262      1468094889      1462863262      E       ST408   0       1       1       1
U106    S206    A300    1494297562      1465490556      1462863262      A       ST405   3       1       0       1
U107    S202    A304    1494297562      1468094889      1462863262      U       ST409   0       0       0       1
```

Now we have to create the look up tables in hive which are currently placed in hbase. The script **create_hive_hbase_lookup.sh** is doing the same-

```
[acadgild@localhost project]$ hive -f /home/acadgild/project/scripts/create_hive_hbase_lookup.hql

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-0.14.0.jar!/hiv
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/hive-jdbc-0.14.0-standalone.jar!/org/slf4j/im
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
OK
Time taken: 1.088 seconds
OK
Time taken: 1.864 seconds
OK
Time taken: 0.219 seconds
OK
Time taken: 0.234 seconds
[acadgild@localhost project]$
```

We can see the look up tables from hbase have been created in hive. These tables are external tables-

```
hive> USE project;
OK
Time taken: 0.029 seconds
hive> show tables;
OK
formatted_input
song_artist_map
station_geo_map
subscribed_users
users_artists
Time taken: 0.026 seconds, Fetched: 5 row(s)
hive>
```

Below shows the content of **station_geo_map** table-

```
hive> select * from station_geo_map;
OK
ST400   A
ST401   AU
ST402   AP
ST403   J
ST404   E
ST405   A
ST406   AU
ST407   AP
ST408   E
ST409   E
ST410   A
ST411   A
ST412   AP
ST413   J
ST414   E
Time taken: 0.192 seconds, Fetched: 15 row(s)
hive>
```

Below shows the content of **song_artist_map** table-

```
hive> select * from song_artist_map;
OK
S200    A300
S201    A301
S202    A302
S203    A303
S204    A304
S205    A301
S206    A302
S207    A303
S208    A304
S209    A305
Time taken: 0.158 seconds, Fetched: 10 row(s)
hive>
```

Below shows the content of **subscribed_users** table-

```
hive> select * from subscribed_users;
OK
U100    1465230523      1465130523
U101    1465230523      1475130523
U102    1465230523      1475130523
U103    1465230523      1475130523
U104    1465230523      1475130523
U105    1465230523      1475130523
U106    1465230523      1485130523
U107    1465230523      1455130523
U108    1465230523      1465230623
U109    1465230523      1475130523
U110    1465230523      1475130523
U111    1465230523      1475130523
U112    1465230523      1475130523
U113    1465230523      1485130523
U114    1465230523      1468130523
Time taken: 0.116 seconds, Fetched: 15 row(s)
```

Now since before analyzing we have to massage our data and enrich it in terms of removing invalid records and bad records. We are running the script- **data_enrichment.sh** to do the same

```
[acadgild@localhost project]$ sh /home/acadgild/project/scripts/data_enrichment.sh

Logging initialized using configuration in jar:file:/usr/local/hive/lib/hive-common-0.14.0.jar!/hive-log4j.properties
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/local/hive/lib/hive-jdbc-0.14.0-standalone.jar!/org/slf4j/impl/StaticLoggerBinder.clas
SLF4J: Found binding in [jar:file:/usr/local/hadoop-2.6.0/share/hadoop/common/lib/slf4j-log4j12-1.7.5.jar!/org/slf4j/impl/Sta
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.slf4j.impl.Log4jLoggerFactory]
OK
Time taken: 1.001 seconds
OK
Time taken: 1.032 seconds
Query ID = acadgild_20180115173636_6bbe4616-0445-4252-8ce7-3f71ce7739e3
Total jobs = 2
Launching Job 1 out of 2
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapreduce.job.reduces=<number>
Starting Job = job_1516007939759_0010, Tracking URL = http://localhost:8088/proxy/application_1516007939759_0010/
Kill Command = /home/acadgild/hadoop-2.6.0/bin/hadoop job  -kill job_1516007939759_0010
Hadoop job information for Stage-1: number of mappers: 3; number of reducers: 1
2018-01-15 17:37:12,514 Stage-1 map = 0%,  reduce = 0%
```

After running above script we can see that **enriched_data** table has been created in hive-

```
hive> show tables;
OK
enriched_data
formatted_input
song_artist_map
station_geo_map
subscribed_users
users_artists
Time taken: 0.05 seconds, Fetched: 6 row(s)
hive>
```

Below shows the sample content of **enriched_data** table-

```
hive (project)> select * from enriched_data;
OK
enriched_data.user_id   enriched_data.song_id   enriched_data.artist_id enriched_data.timestamp enriched_data.start_ts  enriched_data.end_ts    enriched_data.geo_cde
nriched_data.station_id enriched_data.song_end_type     enriched_data.like      enriched_data.dislike   enriched_data.batchid   enriched_data.status
U113    S200    A300    1465230523      1475130523      1465130523      J       ST413   3       1       1       1       fail
U100    S200    A300    1494297562      1494297562      1494297562      A       ST410   3       1       1       1       fail
U120    S201    A301    1494297562      1465490556      1468094889      A       ST410   3       0       1       1       fail
U107    S202    A302    1495130523      1465230523      1465230523      NULL    ST415   0       1       1       1       fail
U103    S202    A302    1465490556      1465490556      1465490556      NULL    ST415   2       1       1       1       fail
U106    S202    A302    1465230523      1465130523      1465130523      E       ST408   0       1       1       1       fail
U109    S203    A303    1462863262      1494297562      1468094889      A       ST405   1       1       1       1       fail
        S203    A303    1495130523      1475130523      1465230523      A       ST400   0       0       1       1       fail
U110    S203    A303    1465230523      1465130523      1485130523      NULL    ST415   0       1       1       1       fail
U111    S204    A304    1465490556      1465490556      1468094889      A       ST410   3       1       1       1       fail
U113    S204    A304    1494297562      1494297562      1465490556      NULL    ST415   3       0       1       1       fail
U100    S204    A304    1495130523      1475130523      1465130523      E       ST408   2       1       1       1       fail
U106    S205    A301    1462863262      1462863262      1494297562      AP      ST407   2       1       1       1       fail
U108    S205    A301    1465130523      1465130523      1475130523      A       ST410   2       1       0       1       fail
U111    S206    A302    1465130523      1465130523      1485130523      NULL    ST415   0       1       1       1       fail
U114    S207    A303    1465130523      1465230523      1475130523      NULL    ST415   3       1       0       1       fail
U102    S207    A303    1465230523      1485130523      1465230523      J       ST403   3       1       1       1       fail
        S208    A304    1465490556      1494297562      1465490556      A       ST411   1       0       1       1       fail
U118    S208    A304    1475130523      1465130523      1465230523      NULL    ST415   3       0       0       1       fail
U119    S208    A304    1495130523      1465230523      1465230523      NULL    ST415   3       0       0       1       fail
U101    S208    A304    1462863262      1468094889      1462863262      E       ST408   0       1       1       1       fail
U107    S210    NULL    1475130523      1485130523      1485130523      E       ST404   2       1       0       1       fail
U115    S200    A300    1465490556      1494297562      1465490556      E       ST404   3       0       0       1       pass
U108    S200    A300    1468094889      1462863262      1468094889      E       ST414   0       0       1       1       pass
U107    S202    A302    1494297562      1468094889      1462863262      E       ST409   0       0       0       1       pass
U101    S202    A302    1465230523      1465130523      1475130523      AU      ST401   0       0       1       1       pass
U110    S202    A302    1494297562      1494297562      1468094889      AP      ST402   2       1       0       1       pass
U118    S202    A302    1495130523      1465230523      1465230523      A       ST410   1       0       0       1       pass
U104    S202    A302    1465230523      1475130523      1465130523      E       ST409   2       0       1       1       pass
U102    S203    A303    1465490556      1465490556      1494297562      E       ST404   2       0       0       1       pass
U113    S203    A303    1465490556      1465490556      1462863262      A       ST405   0       0       1       1       pass
U113    S203    A303    1462863262      1468094889      1494297562      E       ST408   2       0       0       1       pass
U105    S203    A303    1475130523      1465230523      1465130523      E       ST408   2       0       1       1       pass
U113    S203    A303    1465490556      1465490556      1465490556      AP      ST402   1       1       0       1       pass
```

Now we have to do data analysis. We are using Spark to analyze the data.

1. **Determine top 10 station_id(s) where maximum number of songs were played, which were liked by unique users.**

Below is the command we are using to achieve this-

```
val TopTenStation = spark.sql("SELECT station_id,COUNT(DISTINCT song_id) AS
total_distinct_songs_played,COUNT(DISTINCT user_id) AS distinct_user_count FROM project.enriched_data
WHERE status='pass' AND batchid=1 AND like=1 GROUP BY station_id ORDER BY total_distinct_songs_played
DESC LIMIT 10")
```

Below screenshot shows the result for same-

```
scala> val TopTenStation = spark.sql("SELECT station_id,COUNT(DISTINCT song_id) AS total_distinct_songs_played,COUNT(DISTINCT user_id) AS distinct_user_count FROM pr
oject.enriched_data WHERE status='pass' AND batchid=1 AND like=1 GROUP BY station_id ORDER BY total_distinct_songs_played DESC LIMIT 10")
TopTenStation: org.apache.spark.sql.DataFrame = [station_id: string, total_distinct_songs_played: bigint ... 1 more field]

scala> TopTenStation.show
+----------+--------------------------+------------------+
|station_id|total_distinct_songs_played|distinct_user_count|
+----------+--------------------------+------------------+
|     ST402|                         2|                 2|
|     ST411|                         2|                 2|
|     ST405|                         1|                 1|
+----------+--------------------------+------------------+
```