1. I used the dataset directly from S3 and was able to successfully read all values from single as well as multiple files.

2. I used ALS model to extract features. For that I converted the user_id and song_id to numbers using StringIndexer.

3. I calculated the frequency of each song listened by a user and passed this frequency value as an argument to .setRatingCol in ALS.

4. After getting the output from ALS, the features were an array so I converted them to Vector using a UDF called "*toVector*".

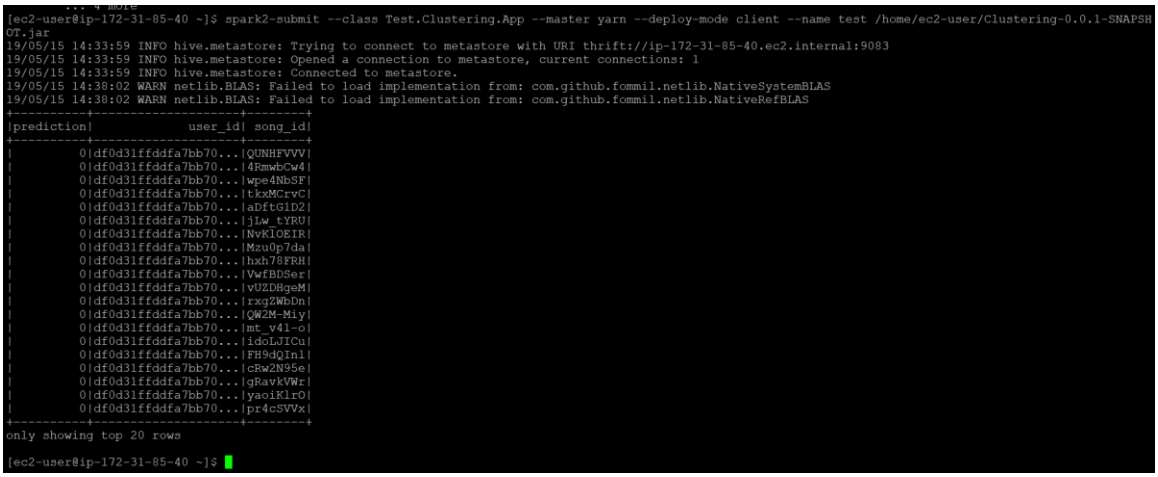
5.This feature vector was then passed as an input to K-means model.

6. From this model, I obtained predictions. I randomly chose k=10 for my model hence I got 10 user clusters who like similar songs.

7. Using **IndexToString** I converted the numeric user_id back to their String equivalents and store the final result in **finalPredictedData dataset.**

8. Next I joined this dataset with the original dataset which had user_id and song_id to fetch song_id listened by each user_id in a cluster and named this dataset **joinedUserWithMetaData**

9. The below screenshot shows the result of above step.



10. After this, I wanted to join the **joinedUserWithMetaData** dataset with the actual metadata having corresponding artist_id with each song but I repeatedly got exceptions and hence ultimately commented that part from my code.

11. My approach was to use the above joined dataset having user_id, song_id, artist_id, prediction columns, partition by prediction and save each prediction file as a csv.

12. From each csv dataset, I would have tried to find out the artist_id having max count and make that cluster correspond to that artist.

13. I could not reach upto the part of calculating CTR and notification clicks, etc. hence also not providing any details about my further approach.

Command used to run jar file:

spark2-submit --class Test.Clustering.App --master yarn --deploy-mode client --name test /home/ec2-user/Clustering-0.0.1-SNAPSHOT.jar