Scott Axcell (827239918)
Assignment 3: Writing Component
04-17-2019

Q1.

It is entirely possible to leave the million song dataset in its current form, where the data is split into analysis data and metadata data, and complete the set of tasks in HW3-PC using a single MapReduce job. To accomplish this two mappers and a single reducer must be implemented. The following implementation can be used to accomplish this single MapReduce job.

Create a mapper for the metadata data where the key and the value output is a string. The string key is the song ID and the value is a concatenation of various metadata fields. For example, the artist name, the song title, year, etc.

Create a second mapper for the analysis data where the key and the value output is a string. The value will, again, be a concatenation of various analysis fields. For instance, duration, time signature, song hotttnesss, etc.

Create the single reducer that expects string inputs for the key and value. The reduce method is responsible for iterating over all the values, which will either be a concatenated string of fields of the analysis data or a concatenated string of fields of the metadata data. These concatenated strings will be split on some denominator and the field information will be stored in the appropriate data structures for computation later. The setup method is where these “global” data structures will be initialized. The cleanup method is responsible for computing answers to the ten questions and printing the answers to disk. Printing the output for each task to a separate file is accomplished by using the MultipleOutputs.write method. This method allows the reducer to write out a key value pair to a specified file.

This implementation is made possible due to the fact that setup method and cleanup method are only run once.

Q2.

The back bone of the new streaming service will be comprised of servers located around the globe. These servers will leverage the thread pool and micro batching architecture from HW2-PC in order to load balance active network connections and client requests. Clients will generally connected to the closest server over TCP. If a server is having issues keeping up with client demand it may instruct the client to connect to a more balance server elsewhere. The servers will provide two major capabilities to the client for request. These include streaming songs and the other the ability to query song information. Therefore, the original HW2-PC server task architecture will need to be extended to handle client requests of these two flavors. For instance, I can envision tasks of the following nature; account registration, account login, find artist, find artists related to an artist, start streaming a song, pause streaming a song, select a position in the song, and so on. The servers would be connected with one another in a network overlay similar to the network overlay implemented in HW1-PC. This overlay will be created using Dijkstra’s shortest path algorithm and dynamically updated as the weights on the links between servers changes. An extension that may prove to be beneficial would be to add a

server load component to the weighting of a link. This would allow the overlay to exclude servers connected to many clients and/or handling a high number of requests.

The metadata for all the available songs will be stored on a Hadoop distributed file system (HDFS). The servers, mentioned previously, will act as name nodes for HDFS and resource managers for YARN as was implemented in HW3-PC. This means song metadata requests can be implemented using MapReduce jobs. Seeing as job reduction was so important in HW3-PC, it would be interested for the server to attempt to combine client requests into single jobs during the execution of micro batched tasks. I'm not sure if this is done in the real world, but it would be an interesting avenue to explore.

The songs themselves, let's assume they're of the .mp3 format, will also be stored on a HDFS. The main reason for this is to leverage the built-in deduplication feature that HDFS provides. As well as the fact that data queries are best completed in a streaming manner. This aligns with the feature of streaming a song as the general case will be to start at the beginning of a song and stream it to the end of the song. Of course, there will be requests to move to different locations in the song. I imagine this would involve creating an entirely new job and seeking through the song to find the appropriate location.

Q3.

The service would be implemented in the same manner that Netflix tackles this issue. Each shared account will be able to create profiles or personas for each family member using the shared account. This enables the song recommendation service to be catered to each individual on the shared account. When using the streaming service the customer would choose the persona they wish to listen to music with.

The measure for satisfaction would include the following data points in order to understand if the song recommendation service is meeting the needs of a customer.

- The number of times the customer chooses to listen to a recommended song.
- The number of times the customer chooses to repeat a recommended song.
- The number of times the customer listens to a recommended song in its entirety. From start of the song to the end of the song.

The data points would be we weighted accordingly. The number of times the customer chooses to listen to a recommended song is the most important data point because this gives us a good indication of whether or not the customer is utilizing the song recommendation service. Followed up second, by the number of times the customer chooses to repeat a recommended song. This gives us an idea of whether the recommend song suited their taste. The idea being, the higher the number of repetitions of a song, the higher the amount of satisfaction.

The measure of satisfaction algorithm might look something like the following.
Measure of satisfaction =

(0.9) * The number of times the customer chooses to listen to a recommended song +
(0.8) * The number of times the customer chooses to repeat a recommended song +
(0.5) * The number of times the customer listens to a recommended song in its entirety

Q4.

We would begin by doing an analysis of the BillBoard Hot 100 songs, the top 100 songs, on a per country basis. This analysis would be used to define the characteristics of popular songs in each country. To decide what characteristics are important the songs will compared against each other (within a country's top 100 songs), in order to find the common traits amongst the top 100 songs. A machine learning algorithm could be implemented to complete this work.

With these characteristics defined, another machine learning (MI) algorithm could be implemented to suggest the micro adjustments needed to make the local artist's songs popular in some country. The MI algorithm would take as inputs the local artist's song and the common traits of the top 100 songs for a given country. The MI algorithm would do an analysis of the local artist's song, then make suggestions based on a comparison of the local artist's song traits and the top 100 songs common traits. The local artist can head back to the studio and make the necessary micro adjustments to their songs as was recommended.

One method that may be used to battle concept drifts would be to complete a historical analysis of the top 100 songs for each country. I would be willing to bet that the time when a song becomes commercially successful is different in each country and that there is a pattern. For instance, songs that become commercially successful in country A at time X, tend become commercially successful in country B at time X + Y. Therefore, this would allow the local artist to theoretically predict the next big hit in country B based on the current analysis of country A.

Q5.

Let's assume we can analyze the following datasets about people; age, location, musical taste, historical musical taste, activities participated in, schooling history, job history, social media posts, etc. Let's assume this information was gathered from social media, video and music services, consensus data, population data, etc. With this wide variety of information we can paint a clear picture of what makes this person unique. We can then use of this person's uniqueness, let's call it a person's picture, and associate it with their historical and current music tastes. With this information we can attempt to predict the musical taste of our target individual. The type of music (music taste) a person enjoys can be defined by finding patterns in the metadata of the songs they listen to.

One avenue is to find all other people who have at some point in their music listening past listened to the same music as the target currently does. We could choose the most popular next phase of musical taste from this group and recommend songs in this group to our target. The assumption being, that we are all different, but we are all also the same, we're humans after all.

The other avenue is to attempt to group the target with other people with the same or very similar picture. Looking at the group of people's current musical taste may be able to provide us with different musical taste than the target's current taste. From these other musical

tastes, the most popular musical taste may be chosen as a prediction for the target. Or one of these musical tastes could be chosen at random. The assumption here is that since this group's picture is very similar, it is highly likely that they may enjoy the same music.

These two avenues could also be combined together to try to identify the next phase in the target's musical journey.