

# A6-Code Review

Code Reviewer: Ankita , Jiangtao

Code Author : Shabbir, Sharad

Review:

Commit:

Reviewing - Commit number -9da21adc0733341c1051b139df898583190a2926 which was pushed on Saturday October 28 — changes made to the makefile. The commit before 2 PM on Friday had errors in the make file.

## MakeFile and Code Execution:

1. ReadMe doesn't have any instruction to modify the home paths for Hadoop and Spark.
2. Code runs as expected but make does not generate a report after running the program.

## Report:

1. No pdf or html version of report included.
2. When opening the existing Report in RStudio it doesn't Knit as 'microbenchmark' library not present, which the author didn't mention anywhere to install.
3. Cannot see and compare the results as `make run` in the report does not work. Hence a pdf version would have been really helpful.

## Code Implementation:

1. Use of helper functions to calculate the different tasks would be better as the main function becomes too long.
2. Authors have done good job when checking data conversion from `String` to `float`, `int` types, set invalid data to default. However, authors ignored empty value checking in the code, especially the ID and other String type fields. For example: while calculating the number of albums, there is no check done if the `artist_id` or `release` is empty then ignore it, which would influence the result by counting empty data.

```
timeBlock {  
    println("Ans:" + songs.map{x => (x.ARTIST_ID, x.RELEASE)}.distinct.count)  
}
```

3. Redundant code when using both `reverse` function and `negative symbol`, for example:

```
timeBlock {
    songs.top(topN)(Ordering[Double].reverse.on(x => -x.DURATION))
        .foreach{x => println(x.TRACK_ID, "\"" + x.TITLE + "\"" by [" + x.ARTIST_NAME +
"]", x.DURATION)}
}
```

Remove reverse and - as the following code can render the same result

```
timeBlock {
    songs.top(topN)(Ordering[Double].on(x => x.DURATION))
        .foreach{x => println(x.TRACK_ID, "\"" + x.TITLE + "\"" by [" + x.ARTIST_NAME +
"]", x.DURATION)}
}
```

4. Some code have the same functionality except the field name like DURATION , TEMPO , SONG\_HOTTNESSS , authors can write a high order function to decrease these lines of code:

```
println("\ntop 5 longest songs:")
timeBlock {
    songs.top(topN)(Ordering[Double].reverse.on(x => -x.DURATION))
        .foreach{x => println(x.TRACK_ID, "\"" + x.TITLE + "\"" by [" + x.ARTIST_NAME +
"]", x.DURATION)}
}

println("\ntop 5 fastest songs:")
timeBlock {
    songs.top(topN)(Ordering[Double].reverse.on(x => -x.TEMPO))
        .foreach{x => println(x.TRACK_ID, "\"" + x.TITLE + "\"" by [" + x.ARTIST_NAME +
"]", x.TEMPO)}
}

println("\ntop 5 hottest songs:")
timeBlock {
    songs.top(topN)(Ordering[Double].reverse.on(x => -x.SONG_HOTTNESSS))
        .foreach{x => println(x.TRACK_ID, "\"" + x.TITLE + "\"" by [" + x.ARTIST_NAME +
"]", x.SONG_HOTTNESSS)}
}
```

5. Although the code is simple, it would be more helpful if authors add some documentation about their code.