Report

Shreysa Sharma, Jashangeet Singh

10/25/2017

Code Repository: https://github.ccs.neu.edu/pdpmr-f17/a6-jashangeet-shreysa.git

Specifications of host execution Environment

Attribute	Value
Java Version	1.8.0_102
Java(TM) SE Runtime Environment	(build 1.8.0_102-b14)
Java HotSpot(TM) 64-Bit Server VM	(build 25.102-b14, mixed mode)
Model Identifier	MacBookPro11,2
Processor Name	Intel Core i7
Processor Speed	2.2 GHz
Number of Processors	1
Total Number of Cores	4
L2 Cache (per Core)	256 KB
L3 Cache	6 MB
Memory	16 GB

Summary of the design of evaluated program

The implementation involves reading the data and putting it in SparkContext. We have created a class CleanUp which does the clean up validuty checks on the provided data and returns the desired column value. We have written a simple CheckValidity method in CleanUp class where we have put all the validity checks as per our assumptions below. The SongAnalysis object is the entry point to the program that taken in song_info.csv and artist_terms.csv locations as arguments and calls the methods in the CleanUp class to get the desired output.

Data Analysis

Assumptions

For loudness of song, the closer the loudness score to 0 in negative, the louder the song. If value is 0, then it is an invalid entry. We have taken the Song Id, Song Name and Loudness score combination to represent the loudness score. This combination is unique. So if there is a song with same id and same name but different loudness score or any of the other combinations it would appear as a distinct entry in the list.

For longness of song, the duration should be greater than 0, the larger the value, longer is the song. We have taken the Song Id, Song Name and duration combination to represent the longest songs. This combination is unique.

For fastness of song, the larger the value of tempo, the faster is the song. We have taken the Song Id, Song Name and tempo score combination to represent the how fast the song is. This combination is unique.

For artist familiarity, the higher the value from 0, the more is the score. If value is 0 or less than 0, record is invalid We have taken artist name, artist id and artist familiarity score combination to represent the artist familiarity. This combination is unique. So if there is an artist with same id but different name and different familiarity score or any of the other combinations it would appear as a distinct entry in the list.

For song hottness and artist hottness the values should be between 0 and 1. Any value less than 0 or greater than 1 is invalid. We have taken song id or artist id, song name or artist name and song hotness or artist hotness score combination to represent the song hotness and artist hotness respectively. This combination is unique. So if there is an artist or a song with same id but different name or different hotness score or any of the other combinations it would appear as a distinct entry in the list.

For the bigger dataset :

Number of distinct songs: 999056

Number of distinct artists: 44745

Number of distinct albums: 149275

Top 5 loudest songs:

(Assumption: the Song Id, Song Name and Loudness Score combination is unique)

Song Id	Song Name	Loudness Score
SOAKZAH12AB0187EA3	The Spectre's Sinister Commandment)	-0.003
SOSKGCK12AC90715B1	A Body So Warm	-0.009
SOBNOEN12AB018AE49	Mechanoid 002	-0.01
SOLPFVW12A8C137059	Penvag	-0.021
SOMGDNA12AC3DF6A17	Face the Nation	-0.025

Top 5 longest songs:

(Assumption: the Song Id, Song Name and duration combination is unique)

Song Id	Song Name	Duration
SOOUBST12AC90977B6	Grounation)	3034.90567
SOXUCQN12A6D4FC451	Raag - Shuddha KalyaN	3033.5995
SOOMVZJ12AB01878EB	Discussion 2	3033.44281
SOTNVEE12A8C13F470	Chapitre Un (a): Toutes Les Histoires	3032.76363
SOGFXNB12A8C137BE5	Der Geist des Llano Estacado Ein Spion	3032.58077

Top 5 fastest songs:

(Assumption: the Song Id, Song Name and Tempo Score combination is unique)

Song Id	Song Name	Tempo Score
SOVVTEZ12AB0184AAB	Beep Beep	302.3
SOMSJWX12AB017DB99	Late Nite Lounge: WVIP	296.469
SOUTBKH12A8C136286	A Place Called Hope	285.157
SOEVQJB12AC960DA2C	Bellas Lullaby - Perrier Citron	284.208
SOTUXOB12AB0188C3A	Troubled Times	282.573

Top 5 most familiar artists:

(Assumption: the Artist Name, Artist Id and Familiarity Score combination is unique)

Artist Name	Artist Id	Familiarity Score
Akon	ARCGJ6U1187FB4D01F	1.0
Akon_ San Quinn_ JT the Bigga Figga	ARCGJ6U1187FB4D01F	1.0
Akon / Eminem	ARCGJ6U1187FB4D01F	1.0
Akon / Styles P	ARCGJ6U1187FB4D01F	1.0
Akon / Wyclef Jean	ARCGJ6U1187FB4D01F	1.0

Top 5 Hot artists:

(Assumption: the Artist Name, Artist Id and Hotness Score combination is unique)

Artist Name	Artist Id	Hotness Score
Daft Punk	ARF8HTQ1187B9AE693	0.997066533839045
Daft Punk	ARF8HTQ1187B9AE693	0.997004803235357
Black Eyed Peas	ARTDQRC1187FB4EFD4	0.982623202516712
Kanye West	ARRH63Y1187FB47783	0.972399563931911
Kanye West / Jamie Foxx	ARRH63Y1187FB47783	0.972399563931911

Top 5 hottest songs:

(Assumption: the Song Id, Song Name and Hotness Score combination is unique)

Song Id	Song Name	Hotness Score
SOXQYSC12A6310E9	Bitter Sweet Symphony	1.0
SOHGFPP12A8AE4624D	White Flag	1.0
SONQEYS12AF72AABC9	Mr. Jones	1.0
SOULTKQ12AB018A183	Nothin' On You [feat. Bruno Mars]	1.0
SOAAXAK12A8C13C030	Immigrant Song	1.0

Top 5 Prolific artists:

(Assumption: the Artist Id, Artist Name and Hotness Score combination is unique)

Artist Id	Artist Name	Number of songs by the artist
AR6681Y1187FB39B02	Ike & Tina Turner	208
ARXPPEY1187FB51DF4	Michael Jackson	204
ARH861H1187B9B799E	Johnny Cash	201
AR8L6W21187B9AD317	Diana Ross & The Supremes	196
ARLHO5Z1187FB4C861	Beastie Boys	194

Top 5 most popular keys (must have confidence > 0.7):

Key	Ocurrances
7	30420
0	28333
2	25845

Key	Ocurrances
9	21283
4	15214

Top 5 most common words in song titles (excluding articles, prepositions, conjunctions):

Word	Ocurrances
LOVE	28666
LIVE	17297
NO	15461
YOUR	14206
DE	13586

Top 5 hottest genres (mean artists hotness in artist_term)

Genres	Mean Hotness
merseybeat	0.840462688027116
piano rock	0.6257724090398099
breakbeat hardcore	0.601436482837353
fearless records	0.585867596137709
maine	0.585867596137709

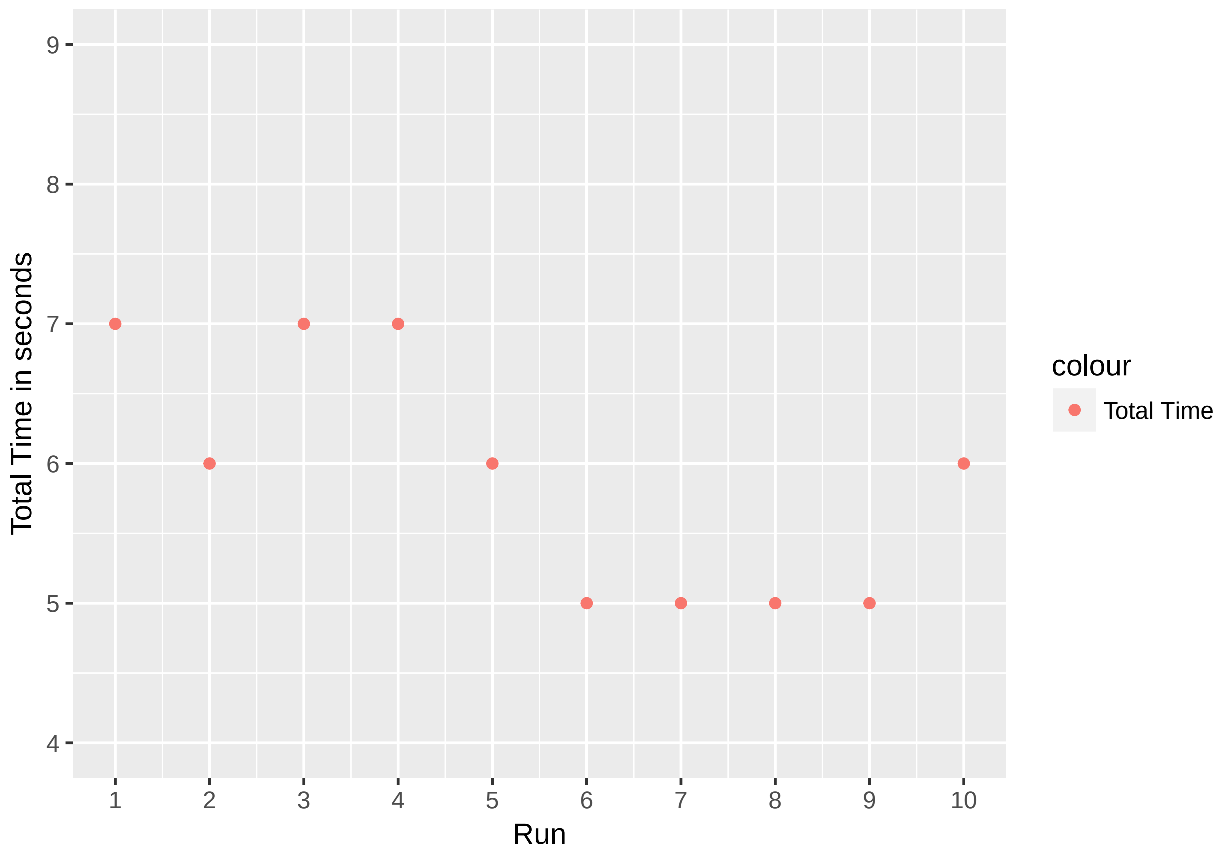
Performance Analysis

Table 1: Time taken for 10 runs on small dataset

Run	total_time	total_avg_time
1	7	5.9
2	6	5.9
3	7	5.9
4	7	5.9
5	6	5.9
6	5	5.9
7	5	5.9
8	5	5.9
9	5	5.9
10	6	5.9

The above table represents the total time taken and total average time of 10 runs of Scala job in seconds. Each run takes about 5.9 seconds to finish. These values are for the subset on the assignment page on the local machine.

Plot 1: Total time in seconds vs Run for the subset



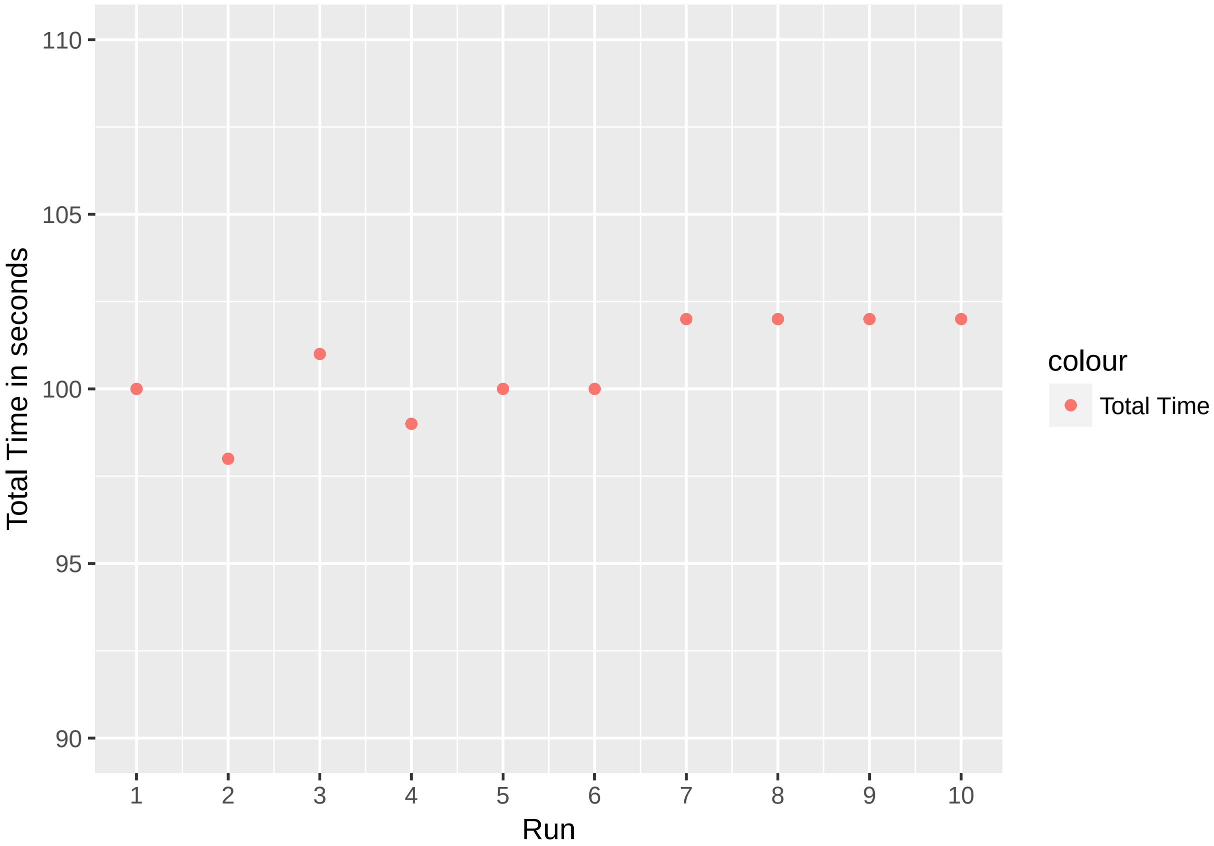
Plot 1 represents the graph for the data shown in table 1. This represents the total time taken by each run in seconds on the small dataset. It can be seen from the graph that the values range between 5 to 7 seconds averaging out to 5.9 seconds for 10 runs.

Table 2: Time taken for 10 runs on complete dataset

Run	total_time	total_avg_time
1	100	100.6
2	98	100.6
3	101	100.6
4	99	100.6
5	100	100.6
6	100	100.6
7	102	100.6
8	102	100.6
9	102	100.6
10	102	100.6

Table 2 represents the total time taken and total average time of 10 runs of Scala job in seconds for the complete dataset. Each run takes about 100.6 seconds to finish. It is noticed that all the requirements but “top 5 hottest genres (mean artists hotness in artist_term)” finish in about 70 seconds, however the query mentioned involves a join and hence takes the majority of the time.

Plot 2: Total time in seconds vs Run for the big dataset



Plot 2 represents the graph for the data shown in table 2. This represnts the total time taken by each run in seconds on the complete dataset. It can be seen from the graph that the values range between 98 to 102 seconds averaging out to 100.6 seconds for 10 runs.

Conclusion

The program analysis some features of the Million song dataset. It was observed that if scala is picking up features from the same file, it runs fairly fast however while joining data points from 2 seperate files makes the program run very very slow. All the requirements but "top 5 hottest genres (mean artists hotness in artist_term)" finish in about 70 seconds but the the one mentioned takes about 14 minutes alone.