Million Song Database Analysis

**By the *Dancing Dads* aka: John Butcher, Austin Poulton, Ian McNair and Barrington Trim**

Submitted for: Big Data Processing (ECS765P) Coursework Assignment 2

18 December 2015

# **1. Problem Statement**

Perform detailed analysis on the Million Song Data using Spark and Scala to understand

* The nature of the data
* Key correlations
* Key insights

The analysis was performed using the following techniques

* 1. Spearman correlation key fields
* 2. Song Tempo and Song 'Hotness Heatmap
* 3. Linear Regression classifier
* Top 10 ‘hottest’ places on the planet based on song hotttnesss – plotted on map using R Studio
* Cluster artists by K means
* Anything else?

The Million Song Dataset can be downloaded at <http://labrosa.ee.columbia.edu/millionsong/>

# **2. Approach**

The following approach was used

* 1. Highlevel data analysis to understand data structure and to inform further analysis
* 2. Set-up development environment
* 3. Develop and unit test Spark code to analyse the data
* 4. Review results to feed into further analysis and development

# **3. Implementation Approach**

**3.1 High-Level Data Analysis**

From an initial inspection of the dataset comprised on Hadoop, the data comprised of 26 csv files containing a total of 1 million data records.

The dataset had minimal metadata, so this had to be determined before detailed analysis could commence. Meta data was constructed by comparing the data to various examples and looking at field position. **T**able 1 below documents the constructed metadata.

A high-level summary analysis was performed on the fields in order to understand data quality and completeness, key observations are listed below

* Each field per row was contained within ““ , e.g. *"1","Progressively Funky","401198","0.0","*
* 1 million total records exist within the data
* 50% of Artist latitude and longitude were missing
* Anything else we should put here????

| **Field Pos** | **Field name** | **Type** | **Description** | **Example values from a Rick Astley song** |
| --- | --- | --- | --- | --- |
| **1** | analysis sample rate | float | sample rate of the audio used | 22050 |
| **2** | artist 7digitalid | int | ID from 7digital.com or -1 | 3204 |
| **3** | artist familiarity | float | algorithmic estimation | 0.754917645 |
| **4** | artist hotttnesss | float | algorithmic estimation | 0.507464243 |
| **5** | artist id | string | Echo Nest ID | ARWPYQI1187FB4D55A |
| **6** | artist latitude | float | latitude | 53.45644 |
| **7** | artist location | string | location name | Newton-le-Willows, Merseyside, England |
| **8** | artist longitude | float | longitude | -2.63265 |
| **9** | artist mbid | string | ID from musicbrainz.org | db92a151-1ac2-438b-bc43-b82e149ddd50 |
| **10** | artist name | string | artist name | Rick Astley |
| **11** | artist playmeid | int | ID from playme.com, or -1 | 1338 |
| **12** | audio md5 | string | audio hash code | bf53f8113508a466cd2d3fda18b06368 |
| **13** | danceability | float | algorithmic estimation | 0 |
| **14** | duration | float | in seconds | 211.69587 |
| **15** | end of fade in | float | seconds at the beginning of the song | 0.139 |
| **16** | energy | float | energy from listener point of view | 0 |
| **17** | key | int | key the song is in | 1 |
| **18** | key confidence | float | confidence measure | 0.324 |
| **19** | loudness | float | overall loudness in dB | -7.75 |
| **20** | mode | int | major or minor | 1 |
| **21** | mode confidence | float | confidence measure | 0.434 |
| **22** |  |  |  | 1 |
| **23** | release | string | album name | Big Tunes - Back 2 The 80s |
| **24** | release 7digitalid | int | ID from 7digital.com or -1 | 786795 |
| **25** | song hotttnesss | float | algorithmic estimation | 0.864248831 |
| **26** | song id | string | Echo Nest song ID | SOCWJDB12A58A776AF |
| **27** | start of fade out | float | time in sec | 198.536 |
| **28** | tempo | float | estimated tempo in BPM | 113.359 |
| **29** | time signature | int | estimate of number of beats per bar, e.g. 4 | 4 |
| **30** | time signature confidence | float | confidence measure | 0.634 |
| **31** | title | string | song title | Never Gonna Give You Up |
| **32** | track id | string | Echo Nest track ID | 8707738 |
| **33** | track 7digitalid | int | ID from 7digital.com or -1 | TRAXLZU12903D05F94 |
| **34** | year | int | song release year from MusicBrainz or 0 | 1987 |

**Table 1:** Million Song Data Constructed Metadata

The fields for were then analysed for format, quality and completeness, key observations are listed below

* 50% of Artist latitude and longitude were missing
* Anything else we should put here????

**3.2 Configure Development Environment**

To facilitate collaborative code development, the team installed the following components

* Scala IDE (including Eclipse)
* Maven
* SBT
* Github Enterprise

The following Gitub repository was to manage code development- <https://github.research.its.qmul.ac.uk/ec15541/million-song>

**3.3 Project Framework**

A Scala framework was created in Github to

* Discussion of Track encapsulation and companion object
* Discussion of Model trait and classification/regression models
* tracks, objects, how was the data cleaned\normalized in the framework etc
* Pattern for Spark driver objects
* Collaborative working
* Code and making into generic class
* Key design principles – e.g. raw tracks into object structure,
* Data cleansing strategy
* Unit Testing

**3.4 Write Code**

Team members created code classes and tested locally using local framework file and a unit test data sent. Once classed were complete they were then committed to the central project

# **4. Analysis of Results**

**4.1. Spearman Correlation**

Barry, to give some input on the code functionality and provide the results

A Spearman correlation was coded in Spark to understand the correlation between Song Tempo and Hotness, the output value is

Code module……

A little description……

**4. 2. Song Tempo and Song 'Hotness' Heat Map**

The table below is a binned scatterplot between song hotness (x, columns) and song tempo (y, rows)

Barry, to give some input on the code functionality and provide the results

|  | **Song Hotness** | | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Song Tempo** |  | **0.1** | **0.2** | **0.3** | **0.4** | **0.5** | **0.6** | **0.7** | **0.8** | **0.9** | **1** |
| **15** | 0 | 0 | 6 | 1 | 7 | 2 | 3 | 1 | 1 | 0 |
| **30** | 0 | 2 | 57 | 42 | 45 | 32 | 13 | 8 | 0 | 0 |
| **45** | 0 | 14 | 435 | 322 | 353 | 283 | 145 | 65 | 22 | 1 |
| **60** | 0 | 38 | 908 | 720 | 760 | 651 | 345 | 158 | 46 | 4 |
| **75** | 0 | 106 | 3332 | 2538 | 2939 | 2336 | 1505 | 605 | 176 | 33 |
| **90** | 0 | 341 | 9938 | 8512 | 9892 | 8658 | 5472 | 2464 | 741 | 131 |
| **105** | 0 | 487 | 16438 | 13889 | 16646 | 14444 | 9414 | 4052 | 1191 | 204 |
| **120** | 0 | 521 | 14612 | 12362 | 14908 | 12752 | 8185 | 3877 | 1119 | 203 |
| **135** | 0 | 750 | 18108 | 15856 | 18340 | 15441 | 10119 | 4764 | 1417 | 233 |
| **150** | 0 | 437 | 11887 | 10541 | 12440 | 10786 | 7195 | 3423 | 978 | 146 |
| **165** | 0 | 306 | 8468 | 7549 | 9180 | 8327 | 5537 | 2580 | 815 | 144 |
| **180** | 0 | 216 | 5228 | 4596 | 5909 | 5168 | 3332 | 1558 | 407 | 59 |
| **195** | 0 | 118 | 3226 | 2994 | 3609 | 3193 | 2154 | 887 | 200 | 31 |
| **210** | 0 | 58 | 1755 | 1596 | 2014 | 1861 | 1216 | 545 | 117 | 23 |
| **225** | 0 | 32 | 863 | 807 | 1103 | 1006 | 647 | 273 | 70 | 12 |
| **240** | 0 | 16 | 411 | 394 | 477 | 491 | 272 | 119 | 35 | 3 |
| **255** | 0 | 11 | 203 | 171 | 179 | 178 | 84 | 50 | 7 | 1 |
| **270** | 0 | 0 | 6 | 5 | 9 | 5 | 6 | 1 | 1 | 0 |
| **285** | 0 | 1 | 2 | 1 | 3 | 0 | 2 | 0 | 0 | 0 |
| **300** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **315** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Table 2:** Tempo vs Hotness Heatmap

**4.3. Linear Regression**

Ian, Austin, to give some input on the code functionality and provide the results

# **5. Summary**

What did we learn about?

* Data Analysis in general?
* Our approach
* The MSD dataset?
* Spark and Scala

Our key observations from undertaking this project are

If we were to continue the project, what would be our next steps?

# **6. References**

Apache Foundation (2015) *Spark Programming Guide*.[online] Available at: <http://spark.apache.org/docs/latest/programming-guide.html> [Accessed 5 December 2015].

ScalaNLP (2015) *Breeze Quick start* Numeric Processing for Scala [online] Avaialble at <https://github.com/scalanlp/breeze/wiki/Quickstart> [Accessed 10 December 2015].

Murphy, K (2012) *Machine Learning – A Probabilistic Perspective* MIT Press, Cambridge, Mass. USA