# Win or Lose? Big Data Recommendation!

Team *1* - Kezhi Li, Yinchen Ni, Shaoze Yang, Jiache Zhang

## Data Preparation

We read *.h5* files and store important data in *.avro* files, and then use **python** to do exploratory data analysis and data pre-processing.

```
Track.h5
├─ analysis
│  ├─ bars confidence
│  └─ ...
├─ metadata
│  ├─ artist terms
│  └─ ...
└─ musicbrainz
   ├─ artist mbtags
   └─ ...
```

Table: Statistic of Raw Data

| | tempo | hotness | year | time_signature | ... |
|---|---|---|---|---|---|
| **count** | 1000000 | 581965 | 1000000 | 1000000 | |
| **mean** | 123.889 | 0.356 | 1030 | 3.59 | |
| **std** | 35.056 | 0.234 | 999 | 1.22 | |
| **min** | 0.000 | 0.000 | 0 | 0 | |
| **25%** | 97.995 | 0.215 | 0 | 3 | |
| **50%** | 122.086 | 0.378 | 1969 | 4 | |
| **75%** | 144.089 | 0.532 | 2002 | 4 | |
| **max** | 302.300 | 1.000 | 2011 | 7 | |

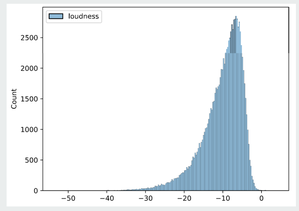



Figure: Distribution of Raw Data

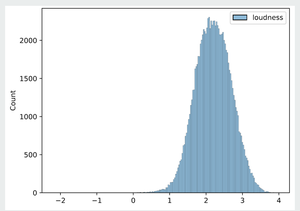


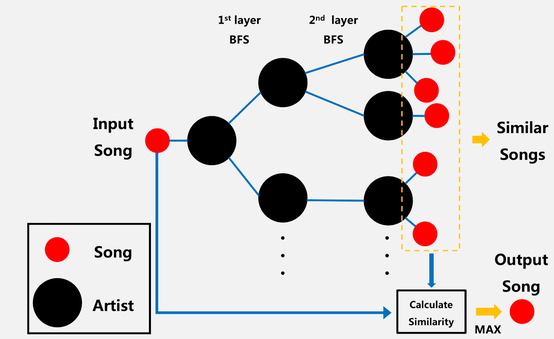

Figure: Distribution of Adjusted Data

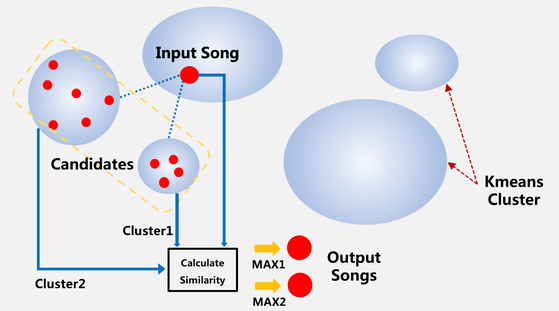
## Basic Database Query

To test on the avro dataset, we use **drill** to query one of the dataset and retrieve the following information:

- The dataset covered songs from 1922 to 2011, namely the age of the songs vary from 12 years to 101 years.
- *Jingle Bell Rock* is the song the hottest song that is the shortest and shows highest energy with lowest tempo.
- First Time In A Long Time: The Reprise Recordings is the album with most tracks. Indeed, it has 4CDs and 80+ tracks.
- The band *Mystic Revelation of Rastafari* has recorded *Grounation* which has highest duration.

## Big Data Recommendation




**Two strategies**

- Similar Artist Based **BFS**
- **KMeans** Diverse Recommendation

**Examples (Old Man Mose):**

- Story of Two(L2), Kentish (Cosine)
- Professor Ironside($C_1$), Por Telefono($C_2$)

## Similarity Metrics

$Song_A = [a_1, \dots, a_n]$ and $Song_B = [b_1, \dots, b_n]$

1. $L_1$ **Norm** $d_{L_1} = \sum_{i=1}^{n} |a_i - b_i|$
2. **Cosine Similarity**

$$cos\theta = \frac{\sum_{i=1}^{n}(a_i \times b_i)}{\sqrt{\sum_{i=1}^{n} a_i^2} \times \sqrt{\sum_{i=1}^{n} b_i^2}}$$

## MapReduce v.s. Pyspark

Pyspark has **8** times speed up!

Performance summary:

- **MapReduce**: 258.674s
- **Pyspark**: 32.016s

## Year Prediction

**No PCA**

```
1 False   79282
2 True    23275
```

**Applying PCA**

```
1 False   71696
2 True    30861
```

**PCA** raises prediction accuracy from **22.69%** to **30.09%**