Artist Recognition from Audio Features of Songs

An attempt to analyse the Million Song Dataset using MapReduce

The Team

Arulselvan Madhavan

M.S. in Computer Science
Northeastern University
Fall ‘14

Ameya Pandilwar

M.S. in Computer Science
Northeastern University
Fall ‘14

Yogendra Miraje

M.S. in Computer Science
Northeastern University
Fall ‘14

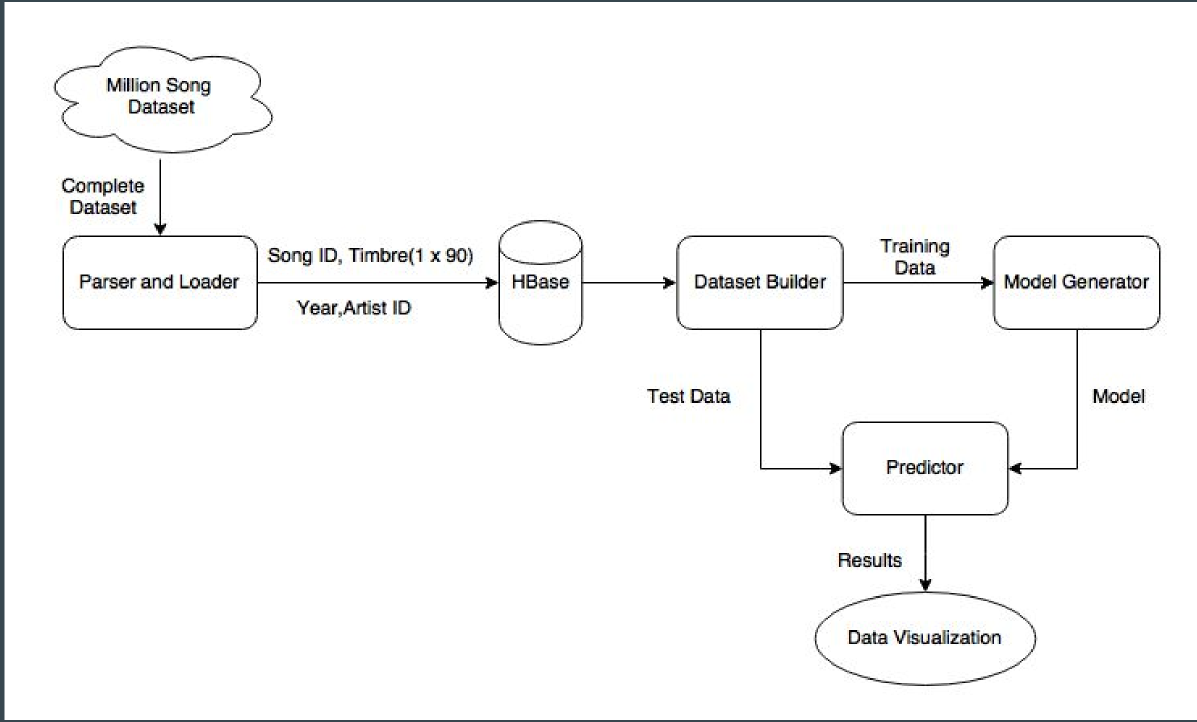
Introduction

- **Problem Statement:**
  To predict the artist of a particular song based on the audio features of the song.

- **Million Song Dataset:**
  - 273 GB of data
  - 1 million songs (from 1920 - 2010)
  - 44,745 unique artists
  - 42 attributes for each song
  - Key Fields: song id, timbre, year, artist id

https://aws.amazon.com/datasets/million-song-dataset/

Approach



Technology Stack

- Hadoop 2.6
- HBase 0.98.15
- Spark 1.5.0
- Scala 2.10
- Thrift 0.9.3
- Anaconda Python 2.7
- AWS EC2
- AWS EMR
- AWS S3

Implementation

Parsing & Loading Data	• Parse HDF5 Data ⇒ Extract Key Features • Load Data into HBase
Preparation of Train & Test Data	• Scan HBase ⇒ Filter Data • Store in LibSVM Format
Machine Learning	• Deciding ML algorithm • Spark's MLlib Logistic Regression
Analysis of Results	• Confusion matrix by year • Most and Least Predictable Artists
Insights & Trends	• Song Popularity & Artist Familiarity • Tempo & Song Popularity

Task 1:
Parsing and Loading Data

Parsing and Loading Data

- Data was in HDF5 format
- Python to parse HDF5 files and extract
  - ArtistID(String)
  - SongID(String)
  - Year(Int)
  - Timbre(2D array of Float Values)
- Key Feature: Timbre
  - Timbre represents the spectral surface of the song.
  - 12 Rows and hundreds of columns (one for each segment of the song)
  - Average and covariance of this 2D array to get 90 columns.

Parsing and Loading Data

- Mapper
  - Input : Absolute path to HDF5 files (1 File for each song. Total: 1,000,000)
  - Map Partitions - 8
    - Each Partition establishes and maintains a connection to HBase via Thrift Server
    - Each Partition is responsible for opening the HDF5 files and extracting SongId, Timbre, Year, Artist Id.
    - Insert the row into HBase via Thrift
  - HBase Row Identifier: SongId.
- Technologies Used: PySpark, Thrift , HBase (3 HMasters and 8 HRegionServers)

Task 2:
Preparing Test & Train Data

Preparing Test & Train Data - High Level Description

- Scan the entire HBase table
- Filter Records that did not have release year.
- Filter Artists who have at least 50 songs in the dataset.
- For each Artist who has at least 50 songs
  - Get the entire row from HBase
- Test Dataset : 5 Songs of each artist is randomly picked.
- Train Dataset: All the remaining songs of the artist is used for training.
- Train Data: 120,000+;Test Data: 9000+
- Output the data in LibSVM format.

Preparing Test & Train Data - Implementation

- Hadoop RDD of the HBase table
  - Input 1: HBase IP address and Port
  - Input 2: HBase Table Name
- Create an RDD with ArtistId,SongId,Year
- Mapper - For each Row, get the ArtistId, SongId and Year.
- Filter Records that don’t have a valid year
- Output: KeyValuePair RDD Format : (ArtistId,(SongId,Year))
- ReduceByKey - ArtistId
  - RDD Format : (ArtistId, [(SongId,Year),(SongId,Year),(SongId,Year).....])

Preparing Test & Train Data - Implementation

- Filter Artists who don't have at least 50 songs.
- Prepare Train RDD and Test RDD
  - TestRDD - Pick 5 songs from each Artist in the filtered RDD
  - TrainRDD - All remaining songs of the artists go into the TrainRDD
  - RDD Format - (ArtistId, SongId)
- Get all features from HBase for both TrainRDD and TestRDD
  - For each ArtistId, apply a map on all of his songs(SongId) to get all the features of a song.
  - RDD format: (ArtistId,Feature1,Feature2...,Feature90,Year)
  - Output the result in LibSVM format
  - Format: Label 1:Feature1 2:Feature2 ....91:Feature91

Task 3:
Predicting the Artists

Applying Machine Learning

- Deciding ML algorithm
  - Multi-Class ML algorithm
  - Considerations: KNN, PCA and Naive Bayes
  - Our Choice: Logistic Regression
- Training the model and Predicting
  - Apache Spark's MLlib library
  - Multinomial logistic regression algorithm
- Performance comparison on AWS (m3.xlarge)

1 Master, 2 Core	2 hour, 39 minutes
1 Master, 5 Core	1 hour, 17 minutes

Task 4:
Analysis of Results

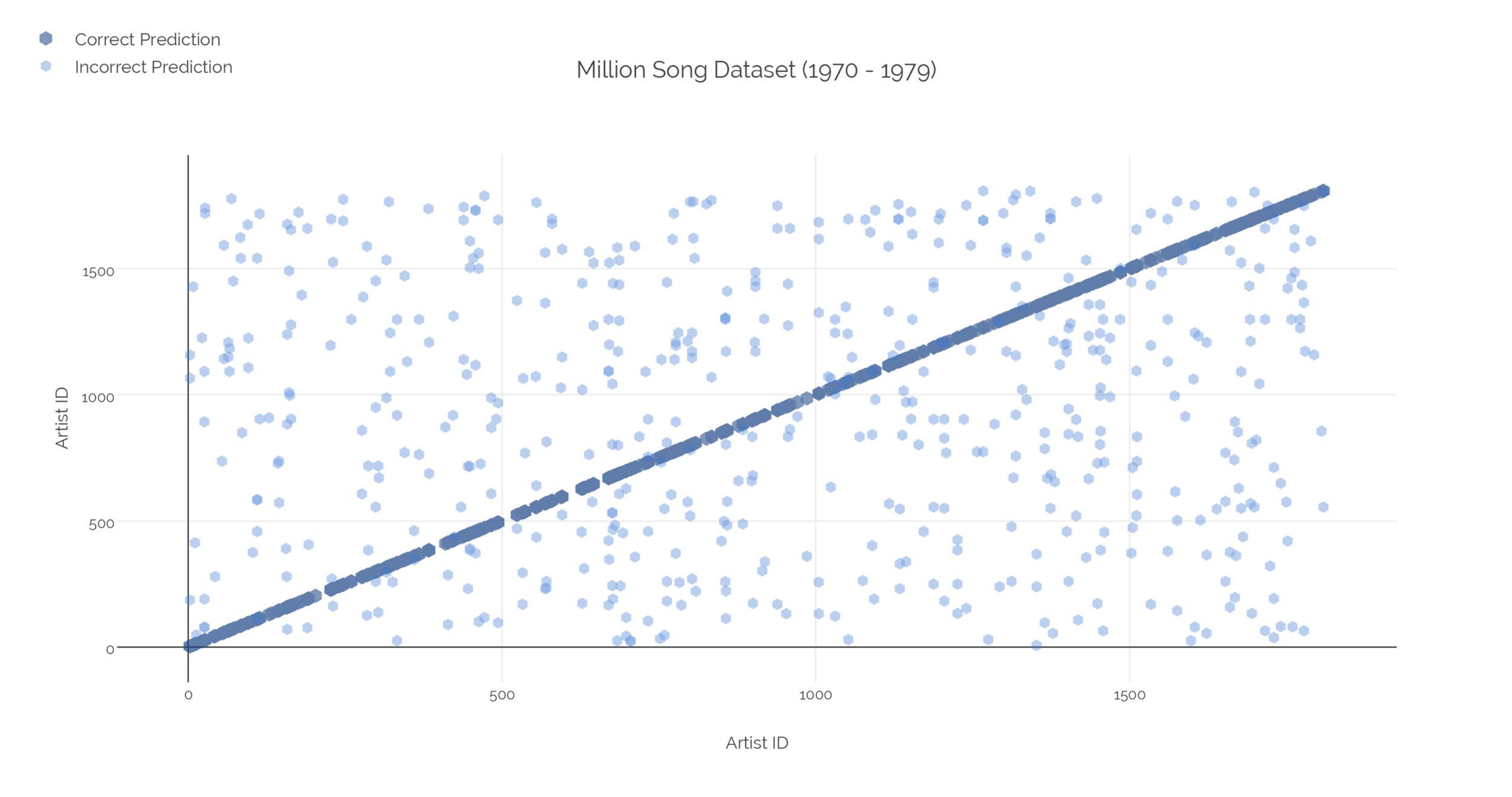
Breakdown of Results

The scattered plot of the actual vs predicted artists are categorized per decade.

The various ranges are:

- 1920 - 1929
- 1930 - 1939
- 1940 - 1949
- 1950 - 1959
- 1960 - 1969
- 1970 - 1979
- 1980 - 1989
- 1990 - 1999
- 2000 - 2009
- 2010

Year 1970 - 1979 (actual vs predicted)



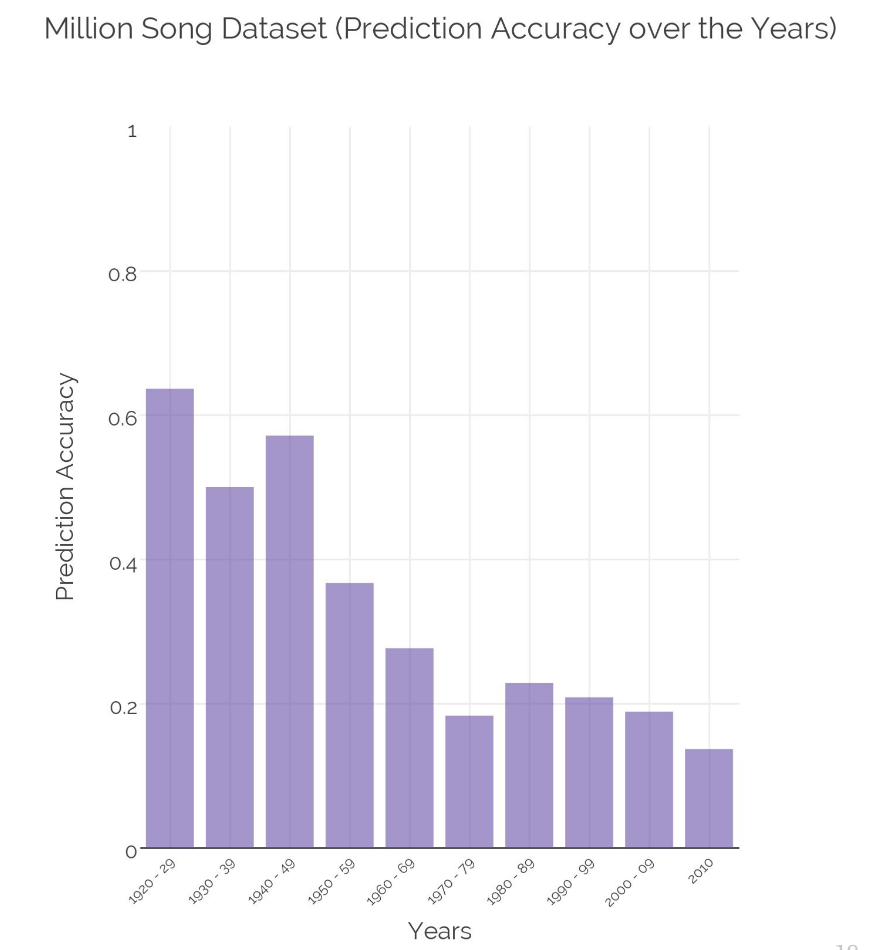
Prediction Accuracy

Findings

The prediction accuracy over the year varies due to the quantity and quality of data available throughout the years in the dataset.

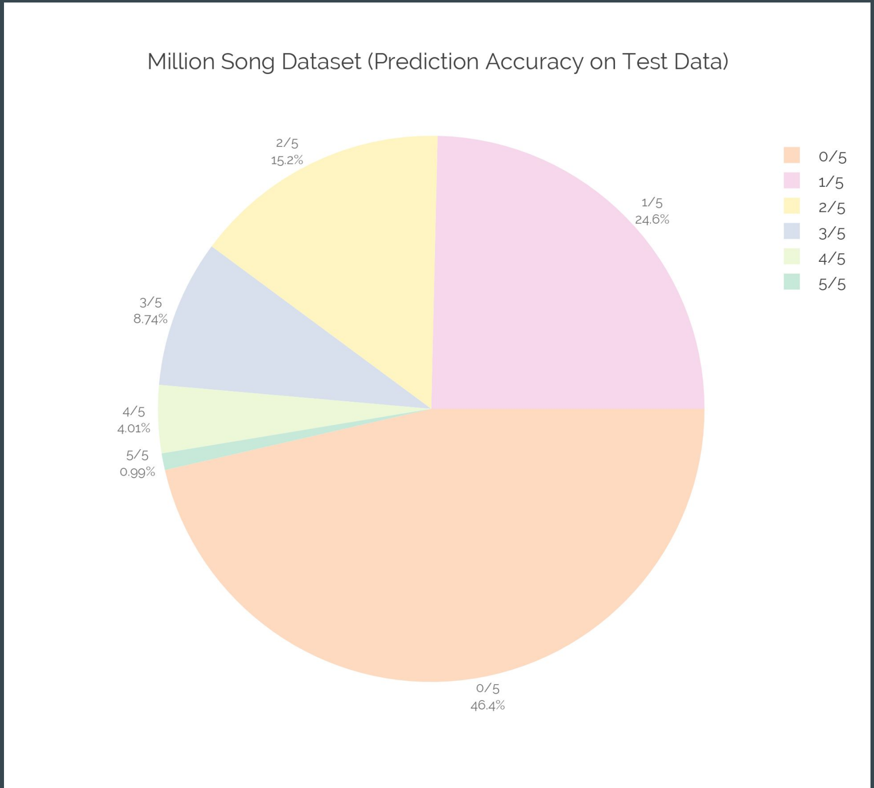
Statistics:

- Correct Predictions = 1861
- Overall Prediction Accuracy = **20.45%**



Predictability of Artists

- **Most Predictable Artists**
  - George Lopez
  - Yonder Mountain String Band
  - Down to the Bone
- **Least Predictable Artists**
  - Jimmy LaFave
  - The Turtles
  - Peter Dennis Blandford Townshend

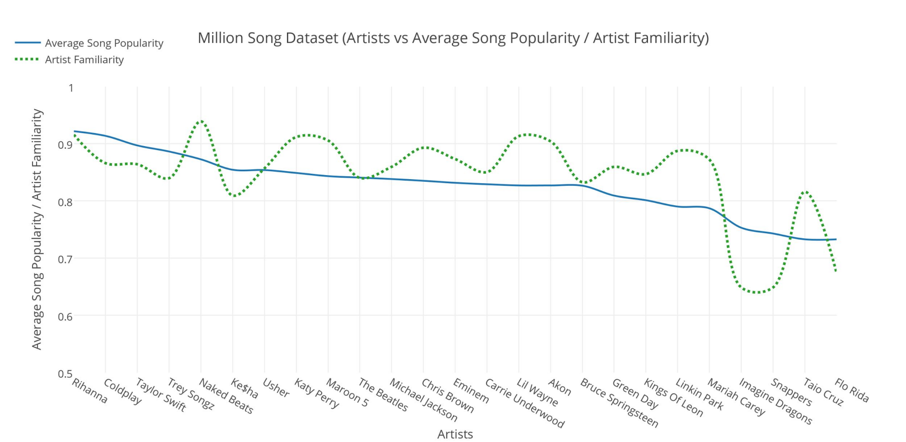


Insights

Trend 1 - Song Popularity & Artist Familiarity

The song popularity affects artist's familiarity.

Mapper : <offset of the line, text of the line>
Reducer : <Artist Name, <Song Popularity, Artist Familiarity>>

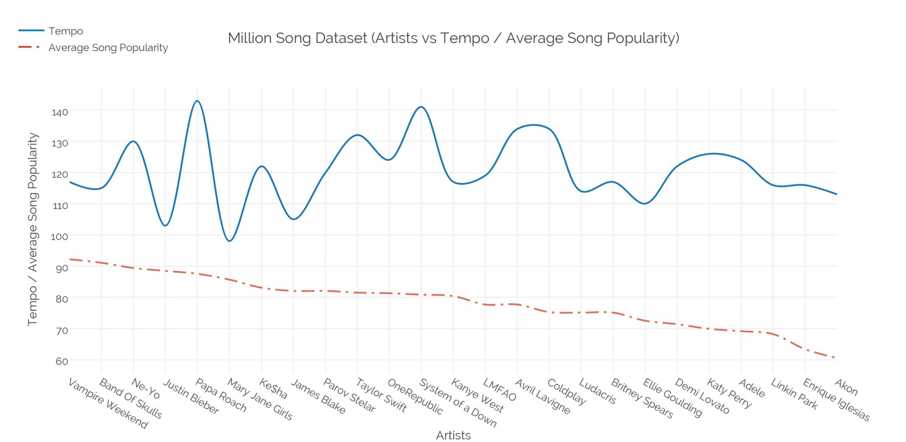


*** Top 25 artists displayed in the visualization*

Trend 2 - Tempo & Song Popularity

The tempo doesn't affect the song popularity.

Mapper : <offset of the line, text of the line>
Reducer : <Artist Name, <Tempo, Song Popularity>>



*** Top 25 artists displayed in the visualization*

Thank You!