# Exploring a million songs

## Parallel Regression Trees in Ufora

Ronen Hilewicz, VP of Engineering
Ronen@Ufora.com

UFORA

# Agenda

- Ufora – language and distributed runtime
- Million Song Database
- The Problem - Year Prediction
- Prior Approaches
- Regression Trees Refresher
- Demo
- Performance and Scalability

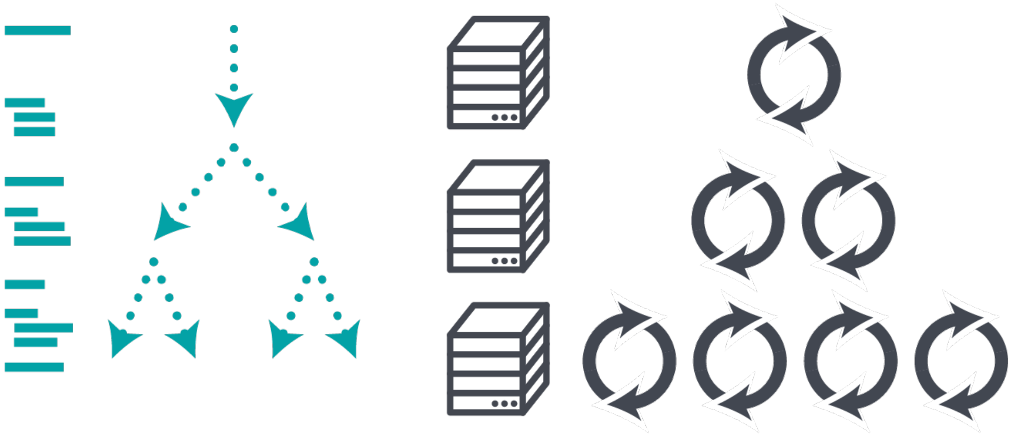
# What is UFORA?

**An *implicitly* parallel language and runtime for data science**

## IMPLICIT PARALLELISM

Code that would normally execute serially...

... in Ufora will **split** across multiple processes on multiple machines, ...



... use more data, and finish faster.

# Million Song Database

- A 280GB free collection of audio features and metadata for a million contemporary popular music tracks

- Does not contain any audio – only the derived features

- Provided by Columbia's LabROSA (Recognition and Organization of Speech and Audio)

- Created by The Echo Nest

# The Data

Static Metadata:

Title • Artist • Album • Genre • Year • Artist Location • etc.

Dynamic Metadata (from EchoNest API):

Song/Artist Hotttness • Familiarity • Similar Artists • Artist Keywords

Audio Features:

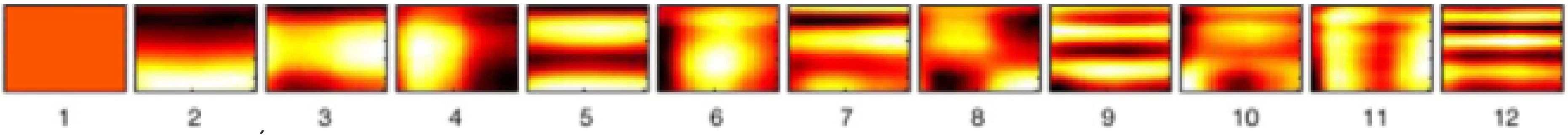
- Key • Mode • Time-Signature • Tempo
- Loudness • Energy
- sections ⌈ bars ⌈ beats ⌈ tatums ⌈ segments

# Demo

## Exploring the Dataset

# Year Prediction

- From audio features, try to predict the year when the song was released
- **Timbre**:



# Year Prediction - Prior Approaches

- **The Million Song Dataset**, T. Bertin-Mahieux, D. Ellis, B. Whitman and P. Lamere, ISMIR '11

- Methods:
  - Baseline: Uniform prediction
  - $k$ nearest neighbors ($k$-NN)
  - Vowpal Wabbit – linear regression using gradient descent

- Challenges:
  - Only half the dataset has year info

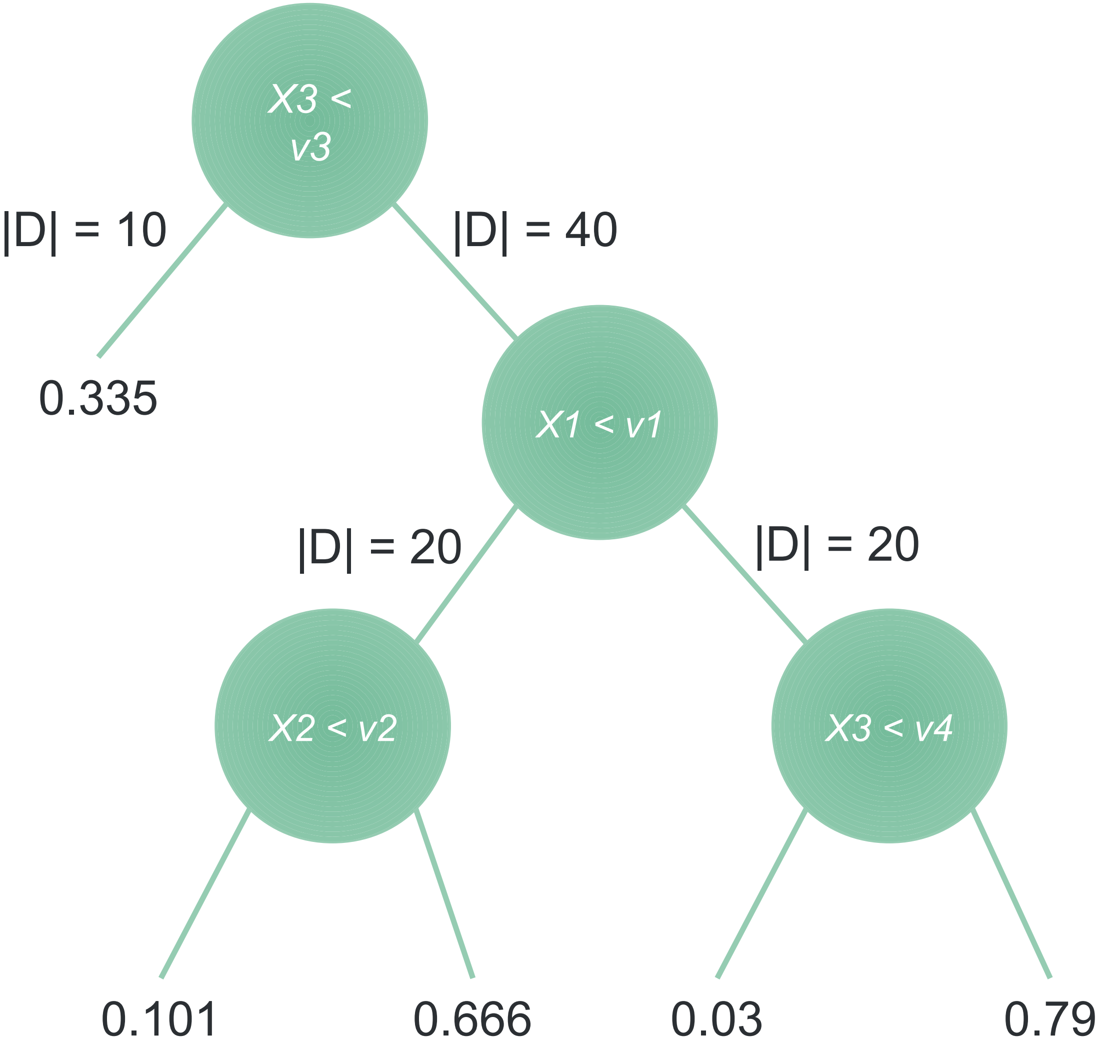
# Prior Results

| | | |
|---|---|---|
| Constant | 8.13 | 10.80 |
| 1-NN | 9.81 | 13.99 |
| 50-NN | 7.58 | 10.20 |
| VW | 6.14 | 8.76 |

# Our Approach

- Regression Trees
  - Highly parallelizable
  - Have been implemented at scale using MapReduce (PLANET)
  - Recently implemented a regression and classification tree library in Ufora

# Regression Trees Refresher



# Learning The Model

- Using a greedy top-down approach
- Partition $D*$ along the split predicate, and proceed recursively on the partitions to build child nodes.
- We select split predicates that minimize the *impurity* in $Y$ values of the training records that are passed to the node. In other words, we want to maximize

  $$|D| \times Var(YD) - (|DL| \times Var(YDL) + |DR| \times Var(YDR))$$

  Where $YDL$ and $YDR$ are the $Y$ values in the partitions $DL$ and $DR$ of $D$

# Demo

## Building the Model

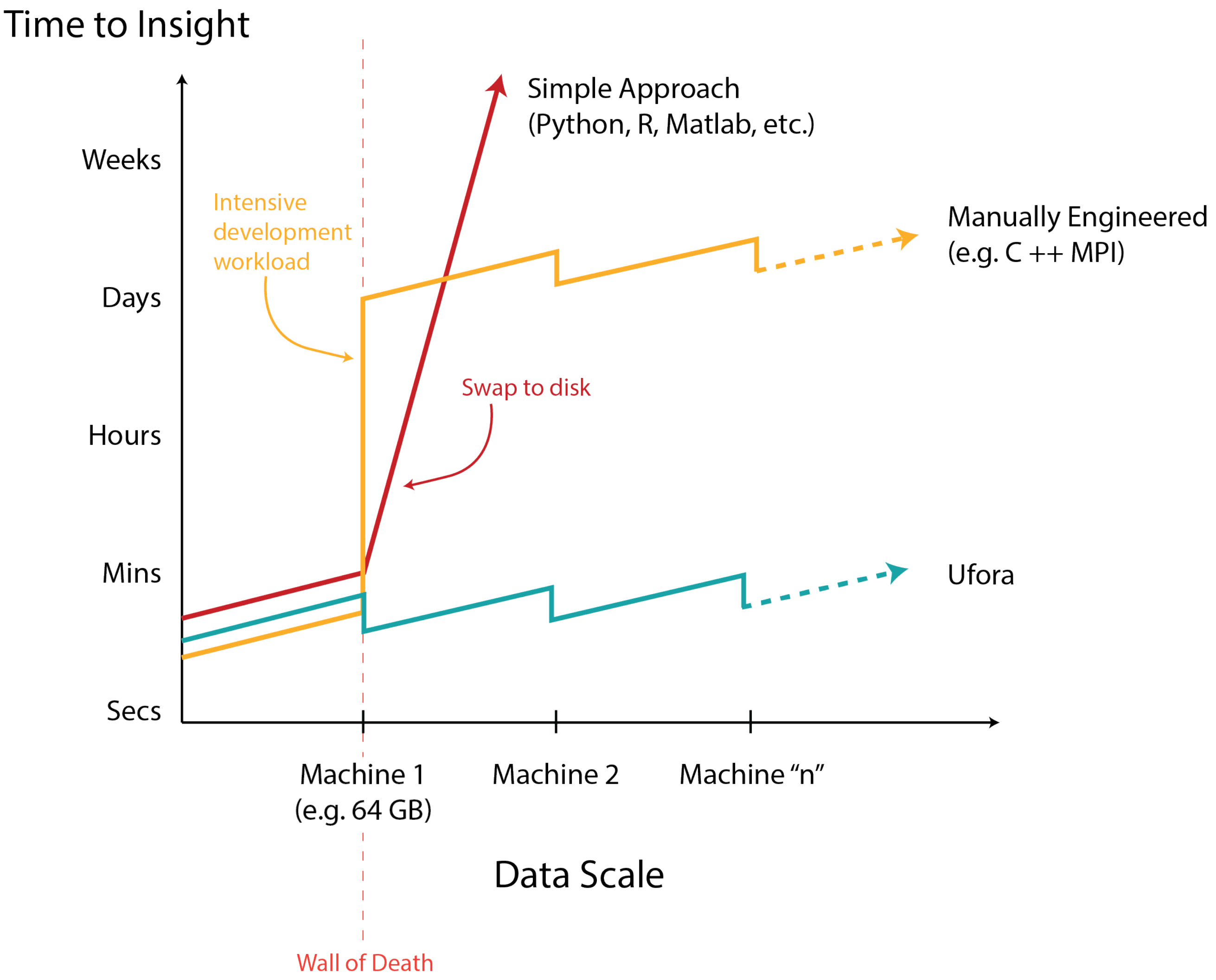
# Results

| | | |
|---|---|---|
| Constant | 8.13 | 10.80 |
| 1-NN | 9.81 | 13.99 |
| 50-NN | 7.58 | 10.20 |
| Tree Model | 6.87 | 9.66 |
| VW | 6.14 | 8.76 |

# Performance (in seconds)

| Depth | SciKit | Ufora |
|---|---|---|
| 1 | 7.02 | 1.4 |
| 5 | 35.14 | 6.43 |
| 8 | 56.21 | 10.7 |
| 10 | 85.16 | 17.18 |

# Comparison of Approaches



# Resources

http://labrosa.ee.columbia.edu/millionsong/

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

- Vowpal Wabbit - https://github.com/JohnLangford/vowpal_wabbit/wiki
- SciKit Learn - http://scikit-learn.org/

Biswanath Panda, Joshua S. Herbach, Sugato Basu, Roberto J. Bayardo

PLANET:  Massively Parallel Learning of Tree Ensembles with MapReduce

*Proceedings of the 35th International Conference on Very Large Data Bases (VLDB*

# We’re Hiring!

- Lead Front End Engineer – Javascript, node.js, Angular.js
- Senior C++ Engineer – C++, boost, Python

info@ufora.com

ronen@ufora.com