Music for Running

Nathan Luskey, Reagan Matthews, Dylan Reese, David Wen

CS4641 Summer 2020

Contents

- Motivation
- Background
- Concept
- Data Source and Improvement
- Approaches
- Data Analysis & Feature Engineering
- Results & Discussion

Background

- Music with ideal tempo enhances workout performance
- Existing approaches use convolutional neural networks
- Millions Song Data Set

Concept

- Music tailored to taste for workouts
- Playlist based on BPM of songs
- Mix of low and high BPM songs to correlate with walking and running
- Assessment of music taste using machine learning approaches

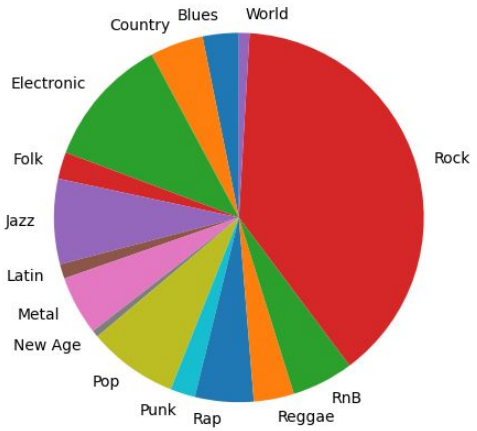
Data Source

- The data was obtained from Dolthub
  - Dolthub - Git for data
  - Uses MySQL to query datasets/databases
  - Our projects makes use of the million-songs dataset

- To obtain the data, we created MillionSongsAPI
  - Functionality build on Doltpy
    - Python API for Dolt
  - First, this clones the million-songs repository
  - Then, it allows clients to query this dataset by row
  - Converts data types to python types for easy use

Data Improvement

- Additional functionality could have benefitted this project
  - Unbalanced dataset genres
  - Most songs were rock genre
  - Need a function to parse database for a balanced dataset

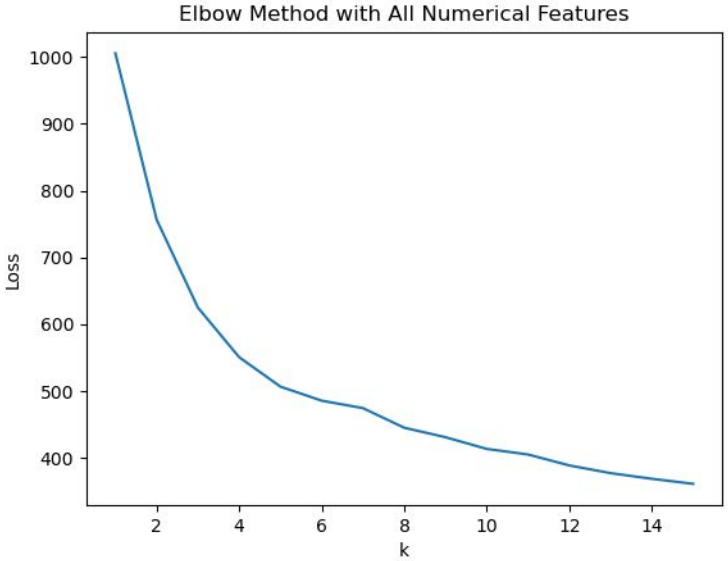


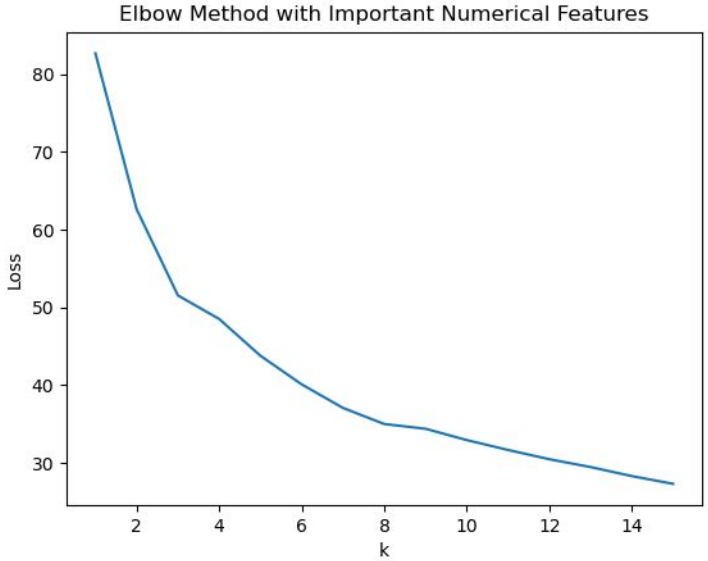
Potential Approaches

- Many approaches exist
- Different approaches may yield different results
- Two approaches: supervised and unsupervised
  - **Supervised:** Decision Tree
  - **Unsupervised:** Kmeans Clustering

Unsupervised Approach

- K-Means clustering with PCA
  - Similar songs not necessarily in the same genre





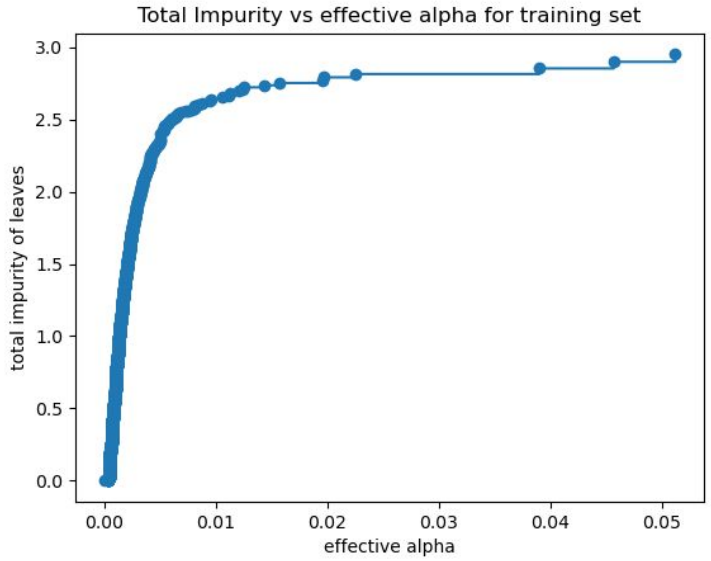
Georgia Tech
CREATING THE NEXT

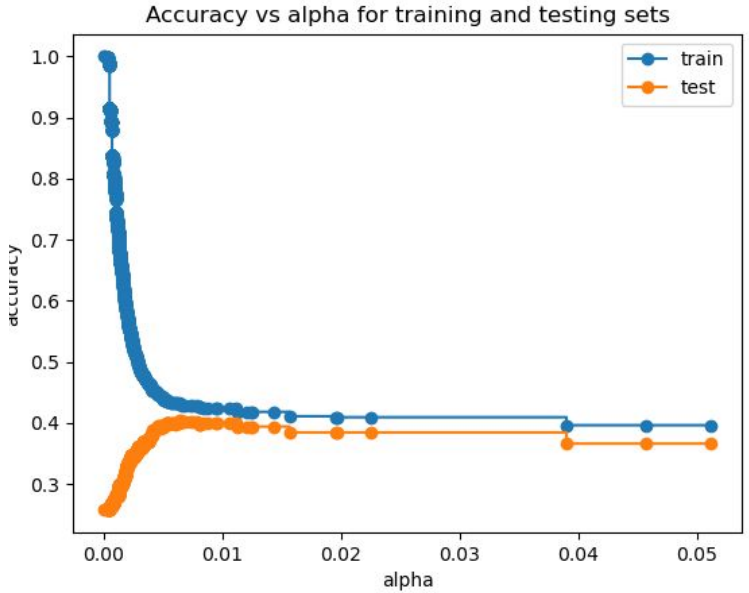
Results and Shortcomings

- Good
  - No bias from unevenly represented genres
  - Fewer clusters than genres
- Bad
  - Optimal cluster number is ambiguous
  - No heuristically obvious elbows

Supervised Approach

- Decision Tree with α Pruning
  - Penalize larger trees by removing the weakest link

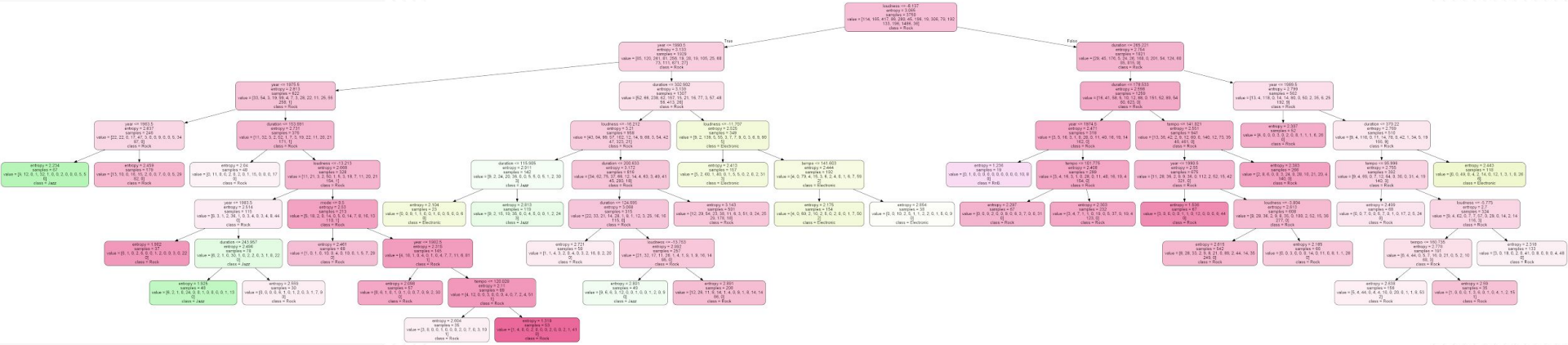




Georgia Tech
CREATING THE NEXT

Resulting Decision Tree

Color indicates purity & class of node

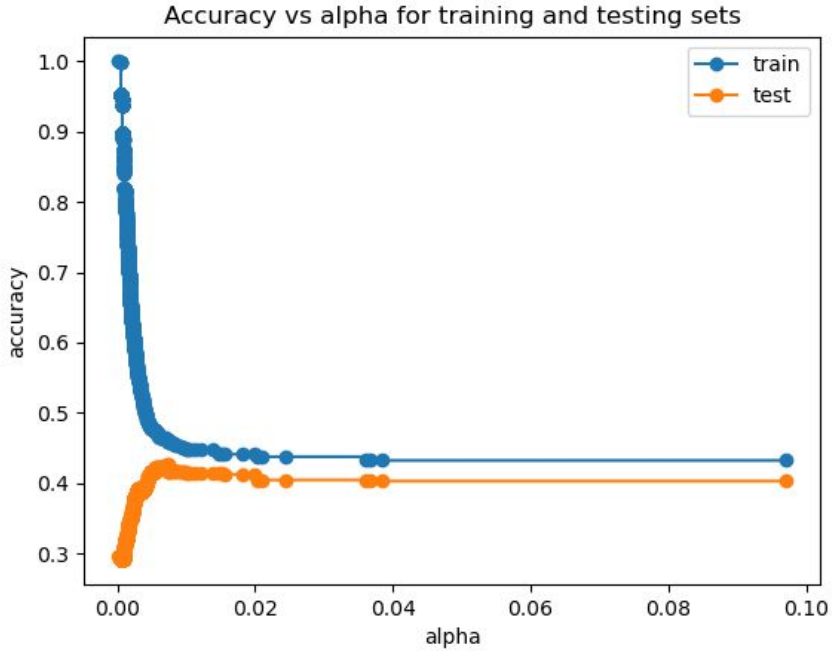


Results and Shortcomings

- Good:
  - Decision Tree has max depth of 9
- Bad:
  - Only ~40% Accuracy of Genre Estimation
  - *Large* bias due to plurality of genre labels being ‘Rock’

Supervised Learning Side Note

- PCA didn’t affect decision tree size or accuracy



Discussion

Approach	The Good	The Bad
Unsupervised	• Fast evaluation • Little bias • No genre constraint	• Disjoint song groupings • Ambiguous cluster count • Difficult visualization
Supervised	• Simple solution • Good visualization	• Inefficient implementation • Heavily biased

Future Work

- Obtain a more balanced dataset
  - Could improve decision tree accuracy
- Further unsupervised analysis
  - Parameter tweaking
- Other possible solutions
  - RL Approaches

Our Repository

https://github.com/nilnate/CS4641_Project

Sources

Dolthub Million Song Database:
https://www.dolthub.com/repositories/Liquidata/million-songs

Scikit K Means:
https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

Scikit Decision Tree:
https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html#:~:text=As%20alpha%20increases%2C%20more%20of,total%20impurity%20of%20its%20leaves.&text=In%20the%20following%20plot%2C%20the,tree%20with%20only%20one%20node.&text=Next%2C%20we%20train%20a%20decision%20tree%20using%20the%20effective%20alphas.

Alpha Pruning:
https://medium.com/@sanchitamangale12/decision-tree-pruning-cost-complexity-method-194666a5dd2f