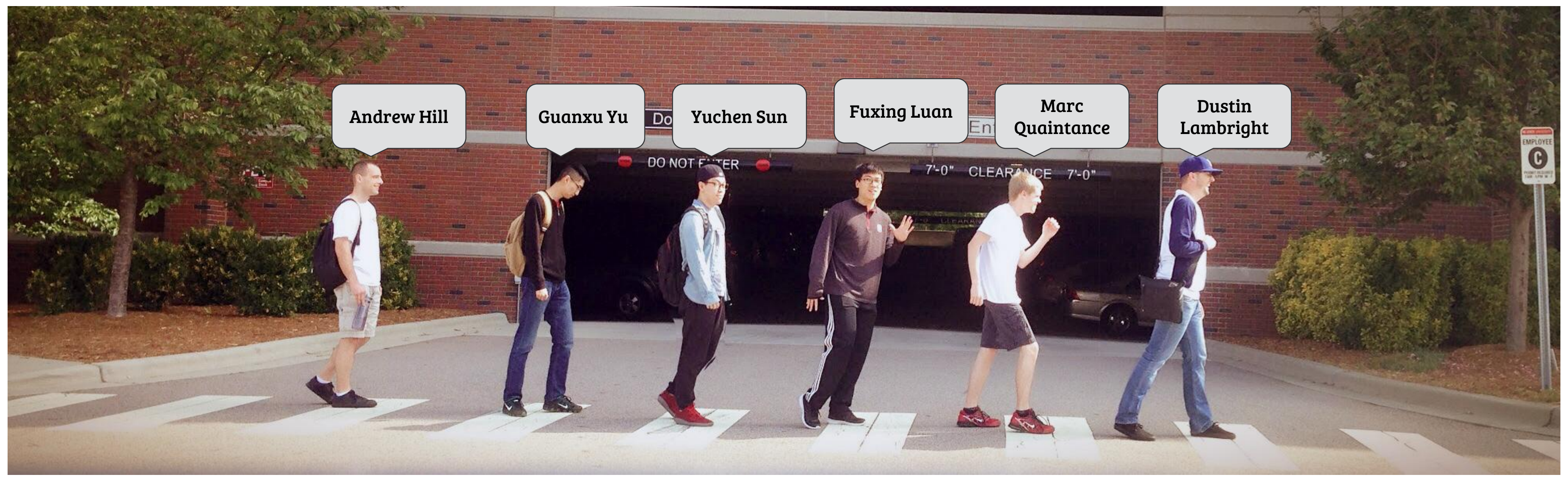
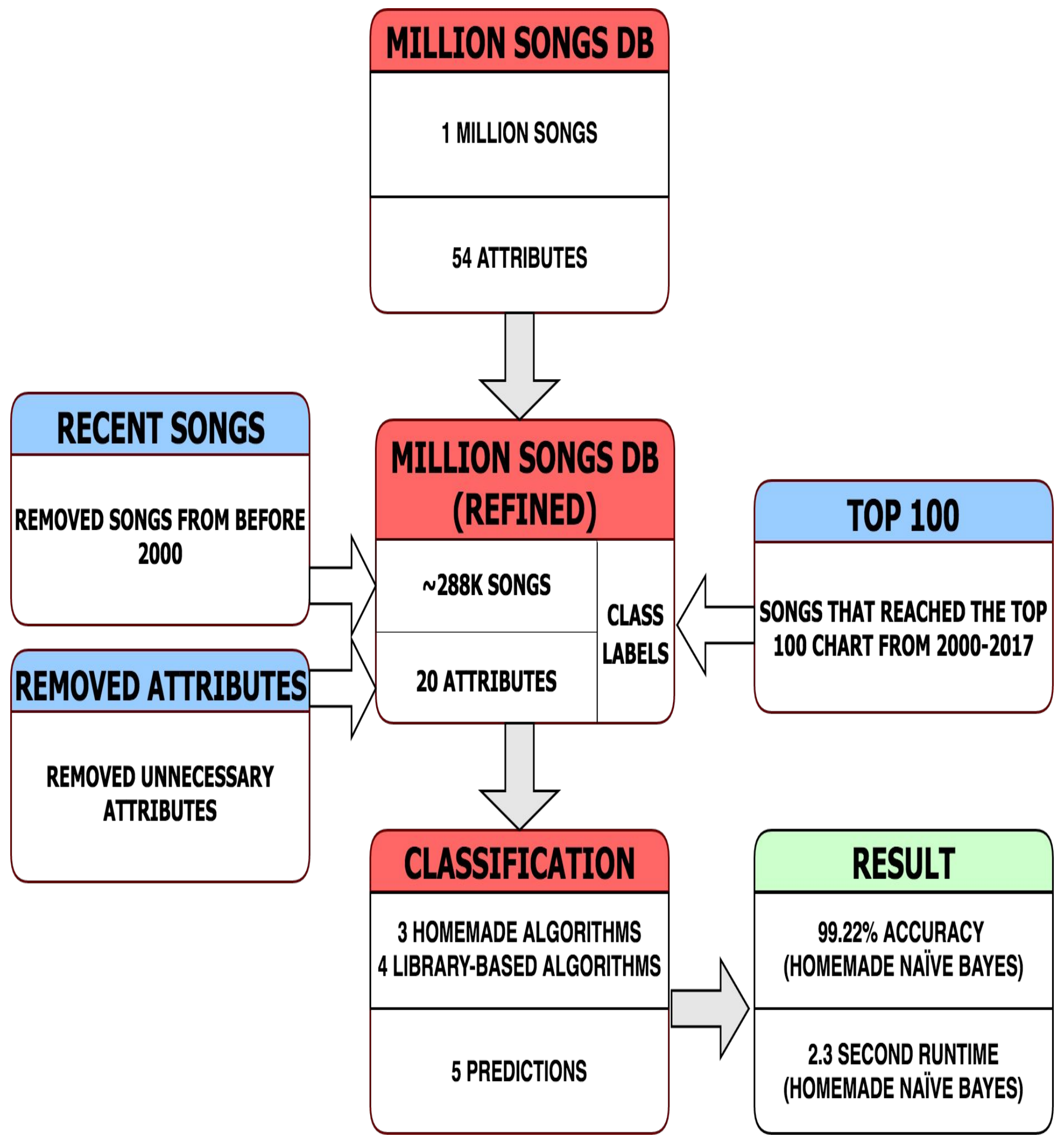
# Algo Rhythm

## A Successful Song?
## I'll tell you.



## 1. Introduction

Attempt to classify songs into those which are and are not Top 100 Hits. kNN, multi-layer perceptron, decision tree, naïve Bayes, and linear regression with LASSO were explored.



MILLION SONGS DB

1 MILLION SONGS

54 ATTRIBUTES

RECENT SONGS

REMOVED SONGS FROM BEFORE 2000

REMOVED ATTRIBUTES

REMOVED UNNECESSARY ATTRIBUTES

MILLION SONGS DB (REFINED)

~288K SONGS

20 ATTRIBUTES

CLASS LABELS

TOP 100

SONGS THAT REACHED THE TOP 100 CHART FROM 2000-2017

CLASSIFICATION

3 HOMEMADE ALGORITHMS
4 LIBRARY-BASED ALGORITHMS

5 PREDICTIONS

RESULT

99.22% ACCURACY (HOMEMADE NAÏVE BAYES)

2.3 SECOND RUNTIME (HOMEMADE NAÏVE BAYES)

## 2. Data description

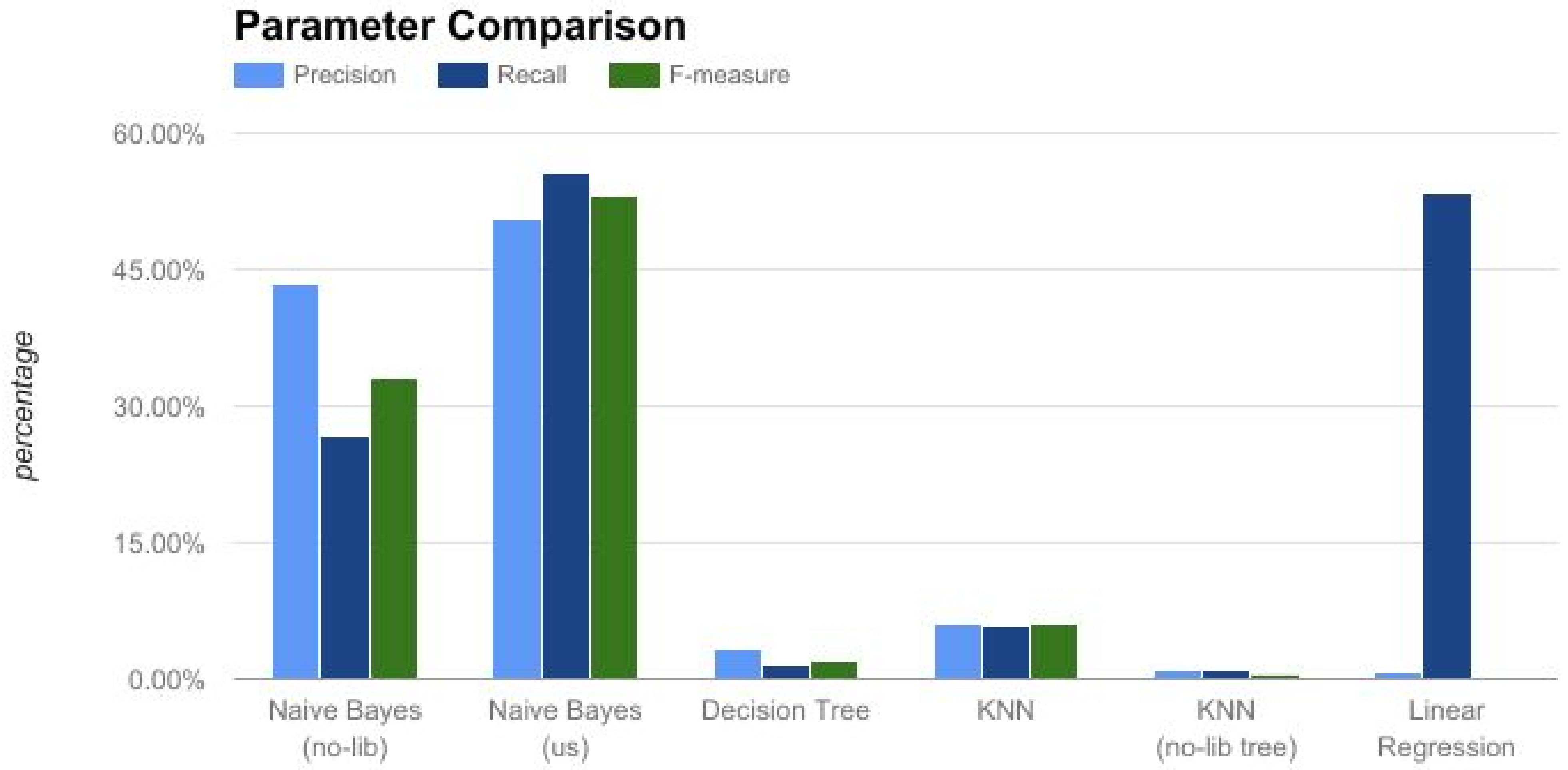
Using data from the Million Song Dataset [1] and a history of the Billboard Top 100 List [2].

Final data attributes considered:
track id, title, artist, release, year, key, key confidence, time signature, time signature confidence, mode, mode confidence, end of fade in, start of fade out, energy, duration, danceability, song hotness, tempo, loudness, top 100 label

## 3. Obstacles to Overcome

- Class imbalance is a major problem, with nearly all classifiers getting good accuracy on the whole data by always predicting "no"
- kNN is too slow on the dataset (runtime in weeks) without a tree structure to limit distance calculations.
- Isolation Forest works well on the initial sample, but runs out of memory on the full dataset.  SVM also does not scale well, but it did complete outlier detection.
- Naïve Bayes performance can be affected by the absence of certain attribute values from the training data and also affected by the splits of continuous values
- Decision Tree has an overfitting problem and it is hard to determine the proper depth to avoid this problem directly.

## 4. Results



nus = non-undersampling,  us = undersampling

In term of running time, basically we can put it into three categories. Our self-implemented Naïve Bayes and Decision Tree can be finished within seconds. But Self-implemented kNN might take a week or more, perhaps we did not avoid the redundant loop well. Others take a few minutes.

## 5. Parameter Choices

**Naïve Bayes**
- Manually decide the split point for continuous attributes

**kNN**
- k = 1 because the highest f-measure on test data is achieved when k = 1

**MLP**
- parameters

**Linear Regression**
- LASSO eps = 0.001 because didn't want parameters eliminated unless truly irrelevant

**Decision Tree**
- Split on best Gin and set the maximum depth 35 to avoid overfitting problem

**OneClassSVM**
- RBF kernel.  As discussed in class it usually works well in practice.

**Isolation Forest**
- Originally 100 base estimators (default)

## 6. Conclusions

Naïve Bayes (us) performed best on this task because it has built-in optimizations that we were unable to duplicate.

Naïve Bayes (us) was the best at finding the rare (positive) class.

Undersampling helped our algorithms predict both successful and unsuccessful

## 7. References

**[1] Million songs dataset:**
https://labrosa.ee.columbia.edu/millionsong
**[2] Billboard top 100 list**
http://www.billboard.com/charts/hot-100
**[3] Relevent paper**
http://ai2-s2-pdfs.s3.amazonaws.com/4193/5a4701ff429b71fb94f77840ffb1258ce894.pdf