BOSTON UNIVERSITY BA476
SPRING 2024

Spotify Top Hits Popularity Predictor

Group 3: Cecily Wang, Emily Doherty, Axel Bautista-Tienda, Hanchao Tang, Sean Ryan

---

BOSTON UNIVERSITY

Statement of the Problem

Can we accurately predict the popularity of a song based on selected features?

Initially, the features that we thought would be the best indicators for how popular a song would be were: **danceability, mode, valence, instrumentalness, and tempo.**

Why is this important and relevant?

- help increase the generation of sales which play a high value when it comes to streaming platforms.
- help music artists to reach a wider audience.

Objective

- Develop a model that lowers MSE the most so that we can accurately predict popularity.

About our data

This dataset covers the top 2000 tracks on Spotify from 2000-2019. It is originally sourced with Spotify library for Python.

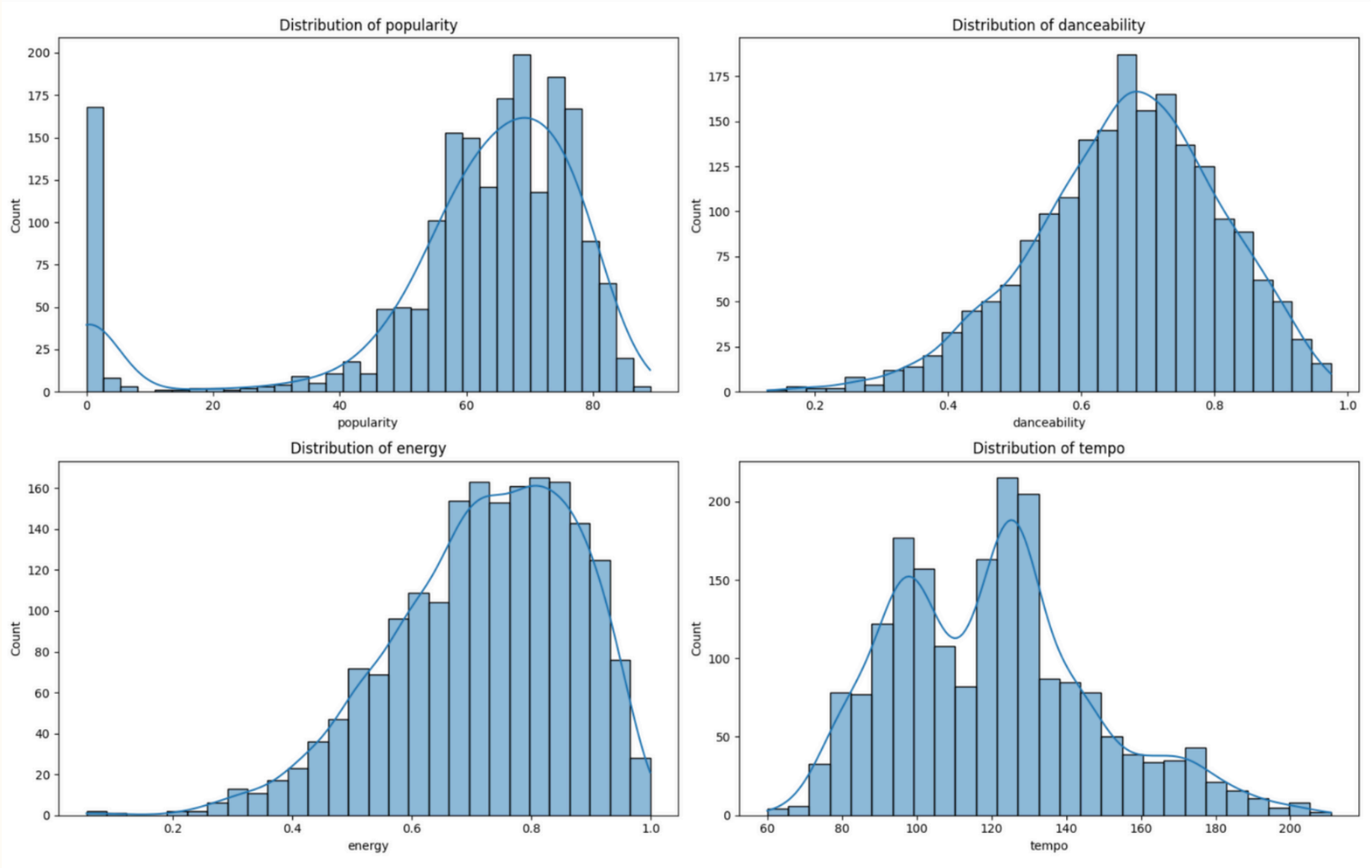
Contents of the Data

- Contains **categorical** and **numerical** data spanning artist names, track titles, release years, and various measures of track characteristics (like danceability, energy, and loudness).
- The entire dataset consists of **18 features and has over 2000 instances**.

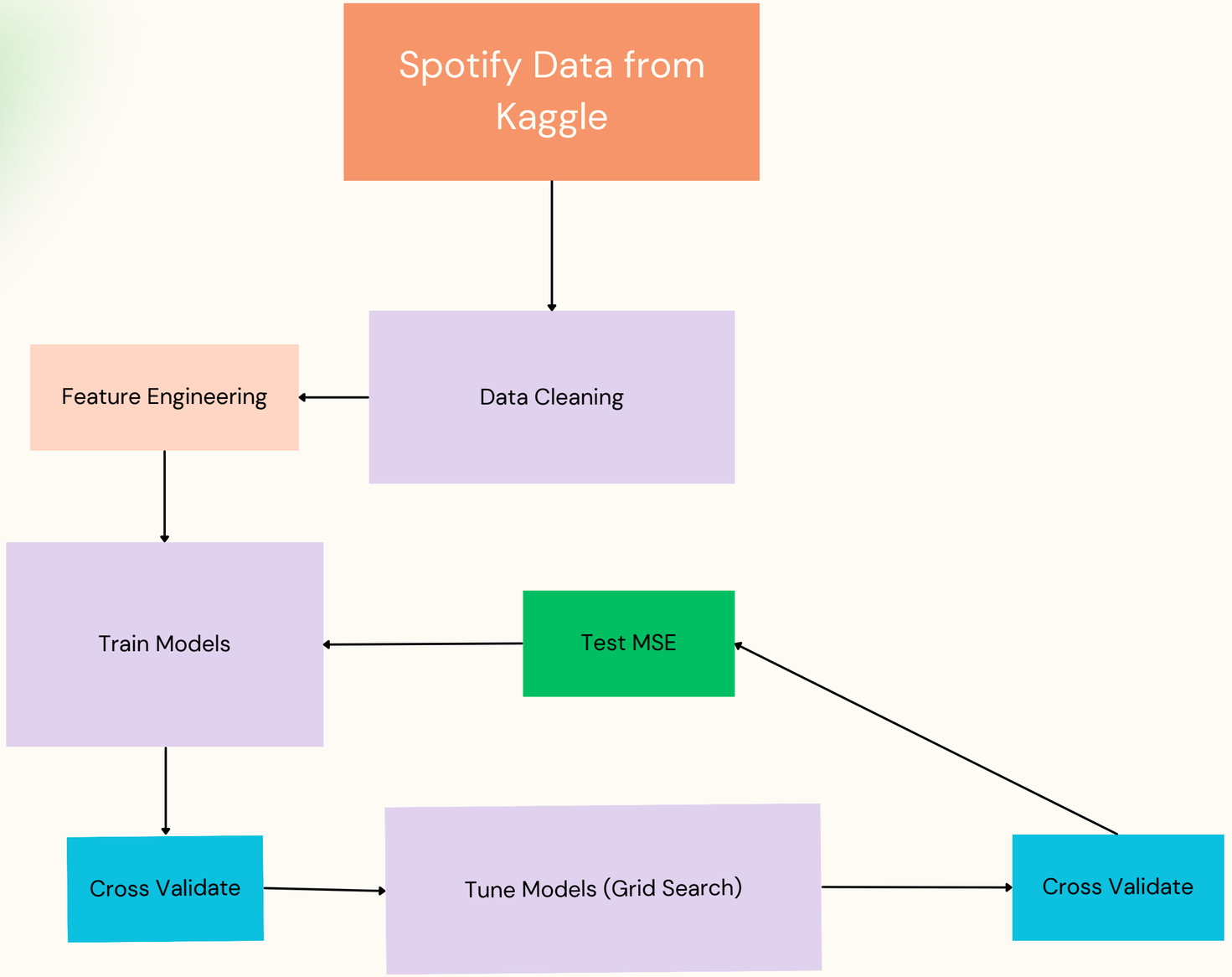
Understanding and Appropriateness of the Data

The purpose of this dataset is to analyze trends in music over the years 2000-2019 and understand **what influences track popularity.** This helps us train our ML algorithms to reach our problem goal.
This dataset is appropriate for the issue we want to solve because of how **comprehensive** it is in terms of the variety of features and the span of years which makes it suitable for the analysis of trends in musical attributes and popularity.

Basic Descriptive Visuals



Workflow


Spotify Data from Kaggle
Data Cleaning
Feature Engineering
Train Models
Test MSE
Cross Validate
Tune Models (Grid Search)
Cross Validate

Handling the Data

Data Cleaning and Feature Engineering

Missing Values	• For features where missing values were significant (e.g., genre), decisions needed to be made whether to fill these in based on external sources or to remove entries.
Normalization/Standardization	• Features like loudness and tempo have been standardized to ensure comparability across different recordings.
One hot Encoding Categorical Data	• Transforming categorical data such as artist names and genres into a usable format for our machine learning models.
Feature Creation	• 'artist_popularity' column reflects the average popularity of songs by the same artist. • ‘age_of_song’. • interaction term: tempo * valence.

Machine Learning Techniques

ML Techniques

- Linear Regression
  - Principal Component Analysis (PCA)
- Polynomial Regression
- Lasso and Ridge Regression
- Decision Trees and Random Forests
- Elastic Net
- Boosting and Bagging

Primary methods that we used to navigate and refine these models were:

- Cross Validation (k-fold, nested)
- Grid Search Cross Validation
- **Mean Squared Error and Adjusted R^2**. (Used to evaluate model performance)
- One hot encoding
- Standardization of data/features
- Feature engineering

Model Outcomes

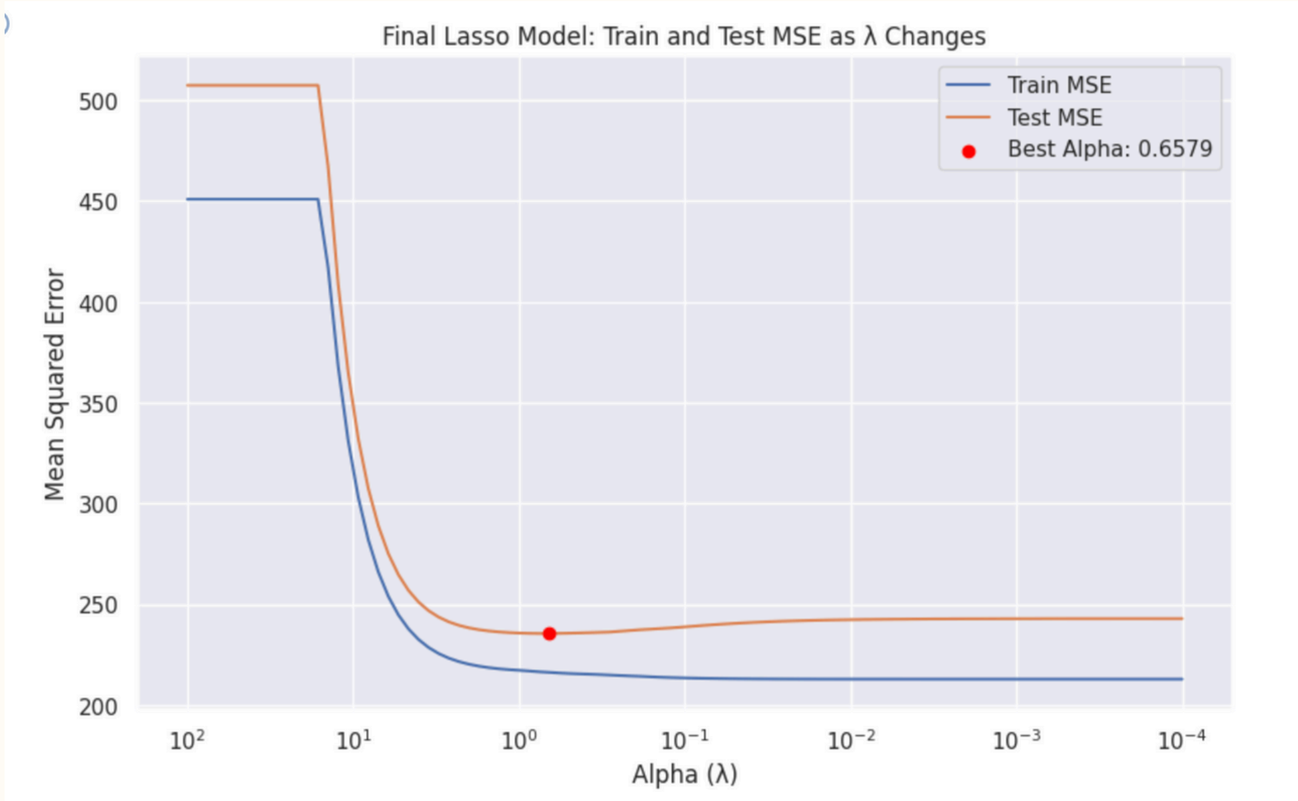
Model	MSE	Adjusted R^2	RMSE	Best alpha	Features chosen
BASELINE: Model: Linear Regression	511.00060515841136				'danceability', 'mode', 'valence', 'instrumentalness', 'tempo'
Linear Regression WITH Principal Component Analysis (data is standardized)	263.69549660799527	0.4751208622199029			68 PCA components
Ridge Regression Model (with one hot encoded, feature engineered columns, and ALL data columns)	242.3266399	0.5176550247245986	15.566844248640606		
Ridge Regression With Cross validation		0.5237442580550712			
Linear Regression; with our one hot encoded elements and feature engineered columns and ALL data columns	242.96681067162064	0.5163807811095031			
Ridge Regression;After parameter tuning, standardization, grid search cv. with our one hot encoded elements and feature engineered columns and ALL data columns	242.84053015839808			100	
Linear Regression(using lassocv to find best features and cv score and alpha). with our one hot encoded elements and feature engineered columns and ALL data columns	artist_popularity: 235.68767684343914 duration_ms: 504.6027514425772				LassoCV gave us features: 'Artist_popularity', 'duration_ms'
Lasso Regularization on all necessary features (onehot, feature eng cols)Data is standardized and cross validated,	**235.78458554155938**	**0.5306768164309301**	**0.1**	**1**	
ElasticNet w/ GridSearchCV	239.32686256709115	0.5166321392394344	15.583341431105143	0.1	one hot encoded features, the original float features, and the new features (artist_popularity, age_of_song)
Lasso w/ Polynomial Features, degree chosen using GridSearchCV	**238.95954816774116**	**0.5243571345738232**	**15.458316472622146**	**1**	**one hot encoded features, the original float features, and the new features**

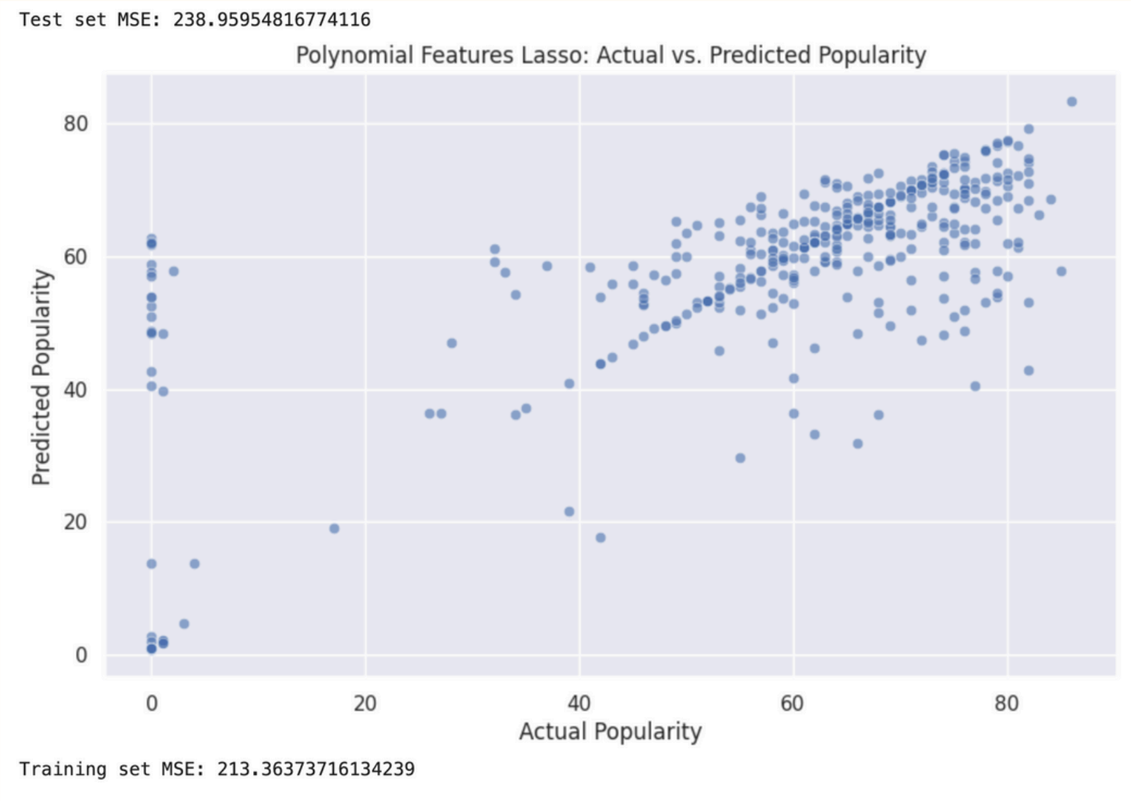
Model	MSE	Adjusted R^2	RMSE	Best alpha	Features chosen
Decision Tree With Pruning	0.43	-0.88			‘danceability', 'mode', 'valence', 'instrumentalness', 'tempo'
Decision Tree With Recursive Feature Elimination	**0.37**	**-0.63**			'energy', 'loudness', 'speechiness', 'acousticness', 'liveness'
Bagging w/ GridSearchCV	340.4853	0.25	19.3736		
Boosting w/ GridSearchCV	245.6471	0.49	15.9913		
Random Forest	15.4	0.99			"duration_ms", "danceability", "energy", "key", "speechiness", "acousticness", "instrumentalness", "liveness", "valence", "tempo", "artist_popularity", "age_of_song", "explicit_True"

The Final Lasso Regularization Model that gave us one of our best MSE at 236

The Final Polynomial Lasso also gave a competitive test MSE of 239.

- Feature selection: shrink less important feature's coefficients to zero,
- Prevent overfitting by reducing model complexity

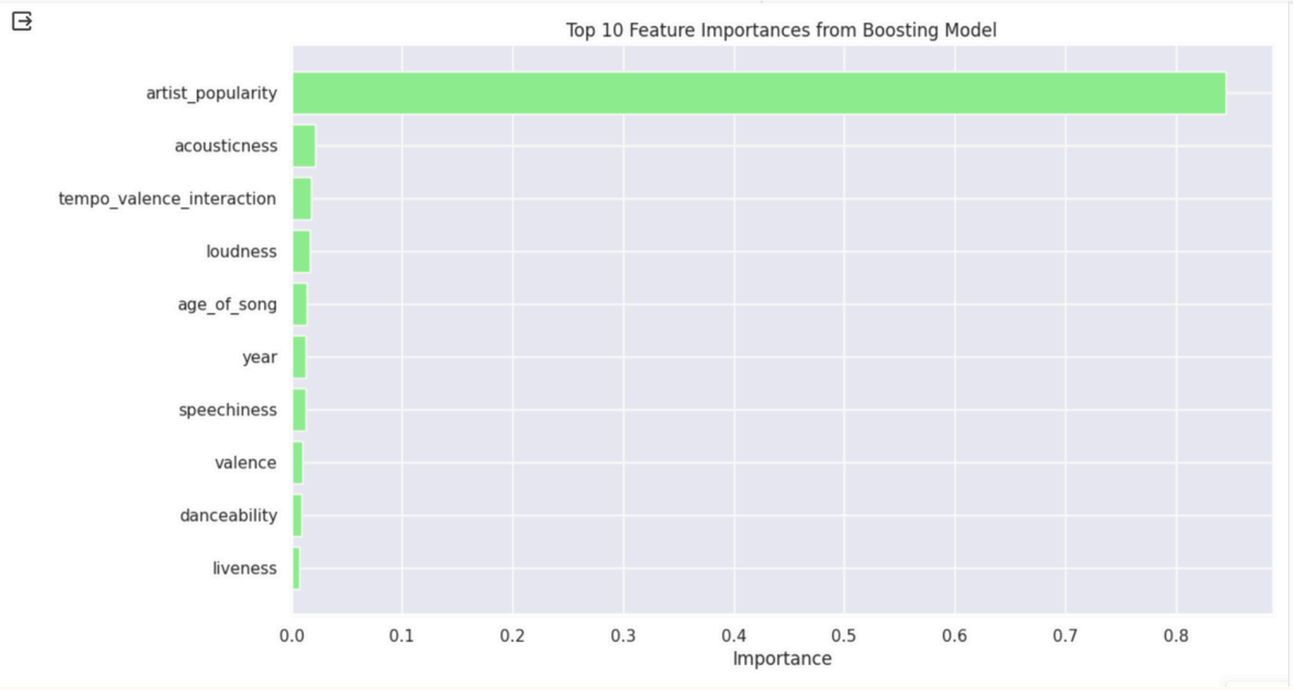


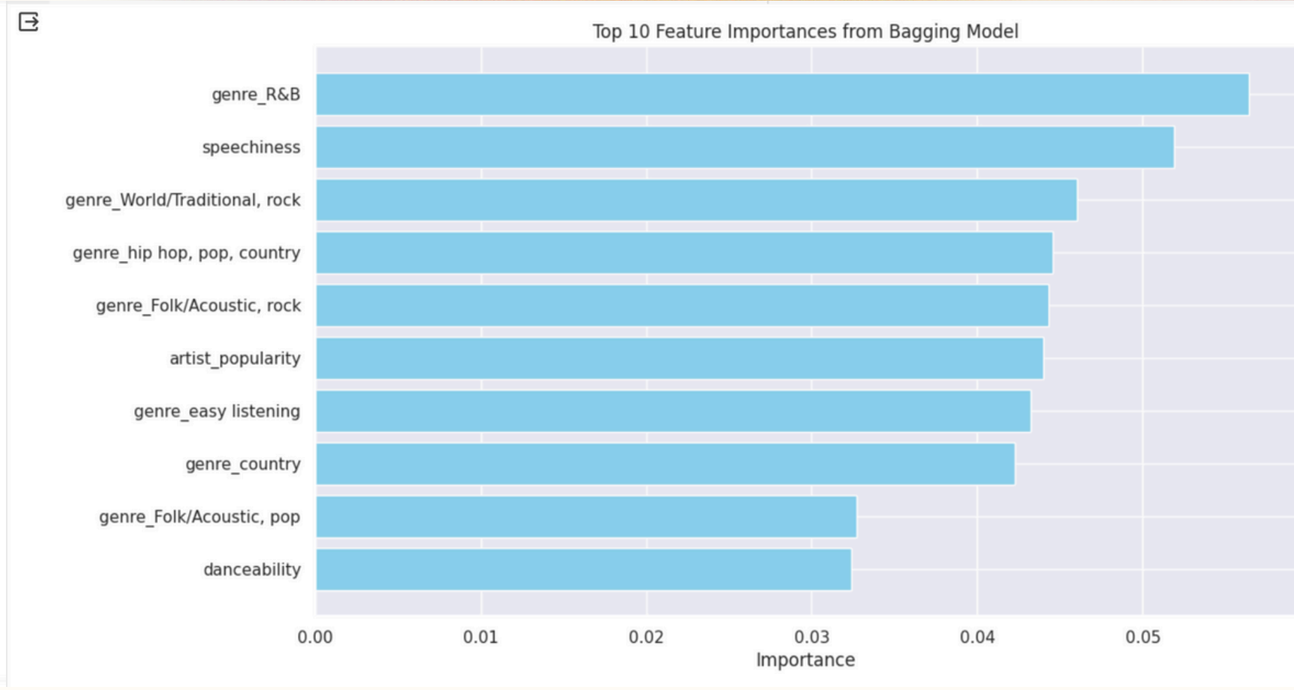


When we changed our approach to classification:

The Final Decision Tree With Recursive Feature Elimination Model gave us our lowest MSE at 0.37

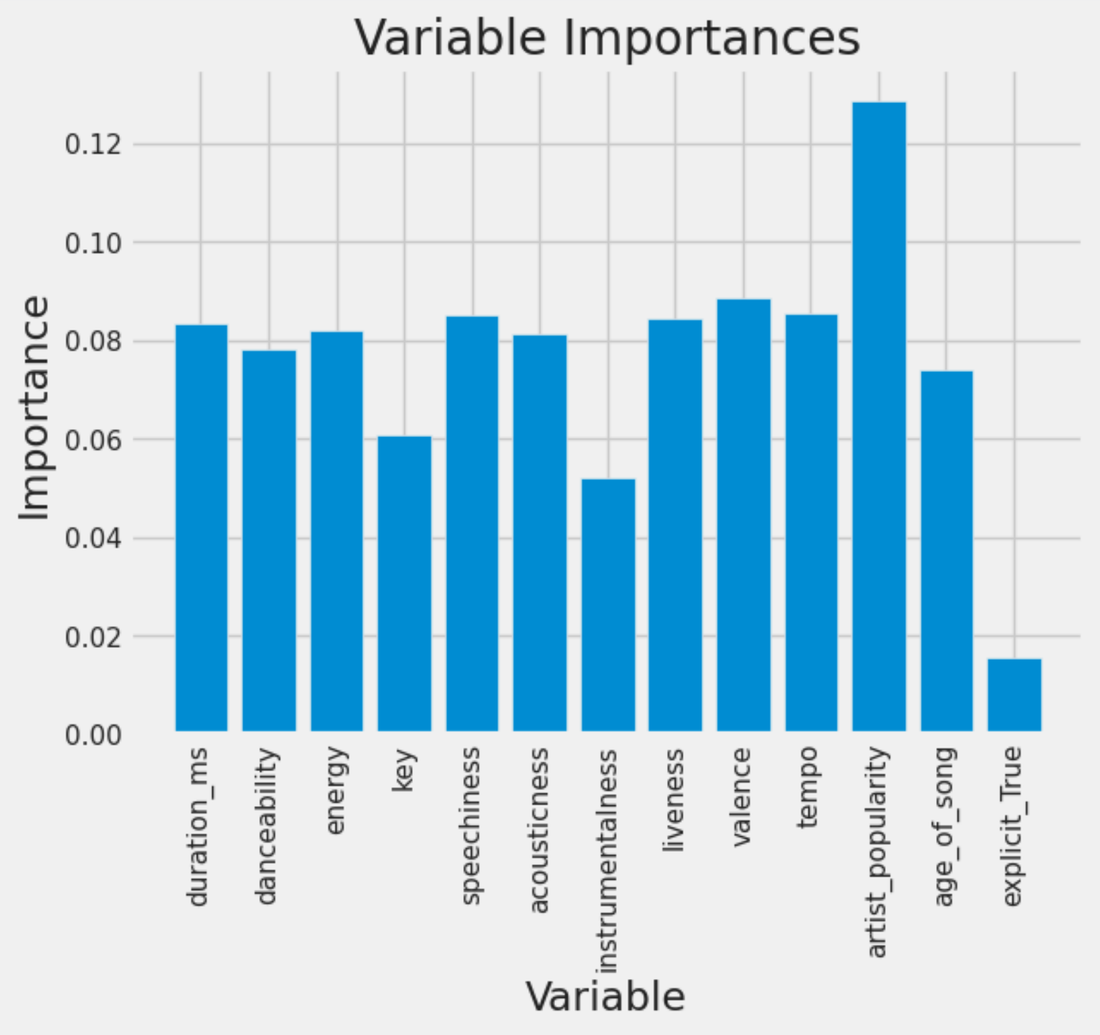
Bagging/Boosting Feature Importance





The boosting model prioritizes artist popularity and acoustic properties, while the bagging model emphasizes genres and vocal content in predicting song popularity.

Random Forest Feature Importance



Artist_Popularity (a measure of how many popular songs an artist has) had the strongest importance in predicting whether a song would have greater popularity of 0.13

Following variables were speechiness, valence, and tempo at 0.09

Challenges and Overcomings

Faced **high runtime** and so **we were limited in how much we could tune our models** using cross-validation and GridSearchCV based on hardware constraints and time

Another challenge - data was perhaps **not suitable for linear regression**.
Best models up to that point were linear regression, but still **not capturing a large proportion of the data**.

We performed **feature engineering on our response feature**
Found **greater success** treating this as a **classification** problem - response predictor of 1 or 0 depending on a threshold of popularity - rather than regression

Through this project, we learned that there is a great deal of **creativity** and **trial and error** that goes into **choosing and tuning** a great **ML model**

THANK YOU