Song Genre Classifying and Clustering

Andres Espinosa, Leonardo Aramayo, Oronde Gibson

Introduction

- **Initiative:** Generative audio models have not yet reached the point that eclipse visual models like ChatGPT and Dall-E. How can we bring AI audio to the same level of viability that LLMs and visual AI models are at today?
- **Objective:** Our project intends to create a machine learning model that could classify songs based on genre

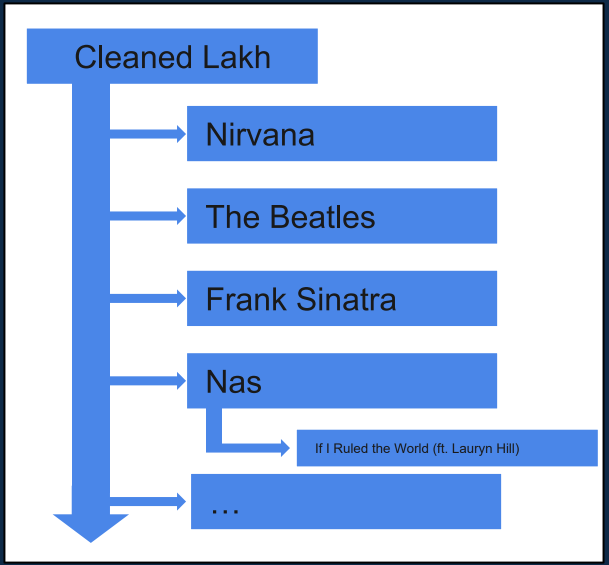
Data Set

- Our group utilized the Cleaned Lakh Midi Data Set from Kaggle which contained 17,000 MIDI files
- MIDI files are audio files that describe what notes are played, when they are played, and how loud each note should be
- This data set was combined with a separate data set of popular songs/artists and their respective genres
- We used this to a create new data set with the Lakh data and their genres

Create Dataset using Lakh

Columns of dataset created from scratch

- **artist_name:** *name of artist*
- **title:** *name of song*
- **file_path:** *path of MIDI file location in system*
- **song_artist:** *format "artist_name - title"*



Note: MIDI files are not directly added to dataset since the pandas dataframe so we added the file paths instead

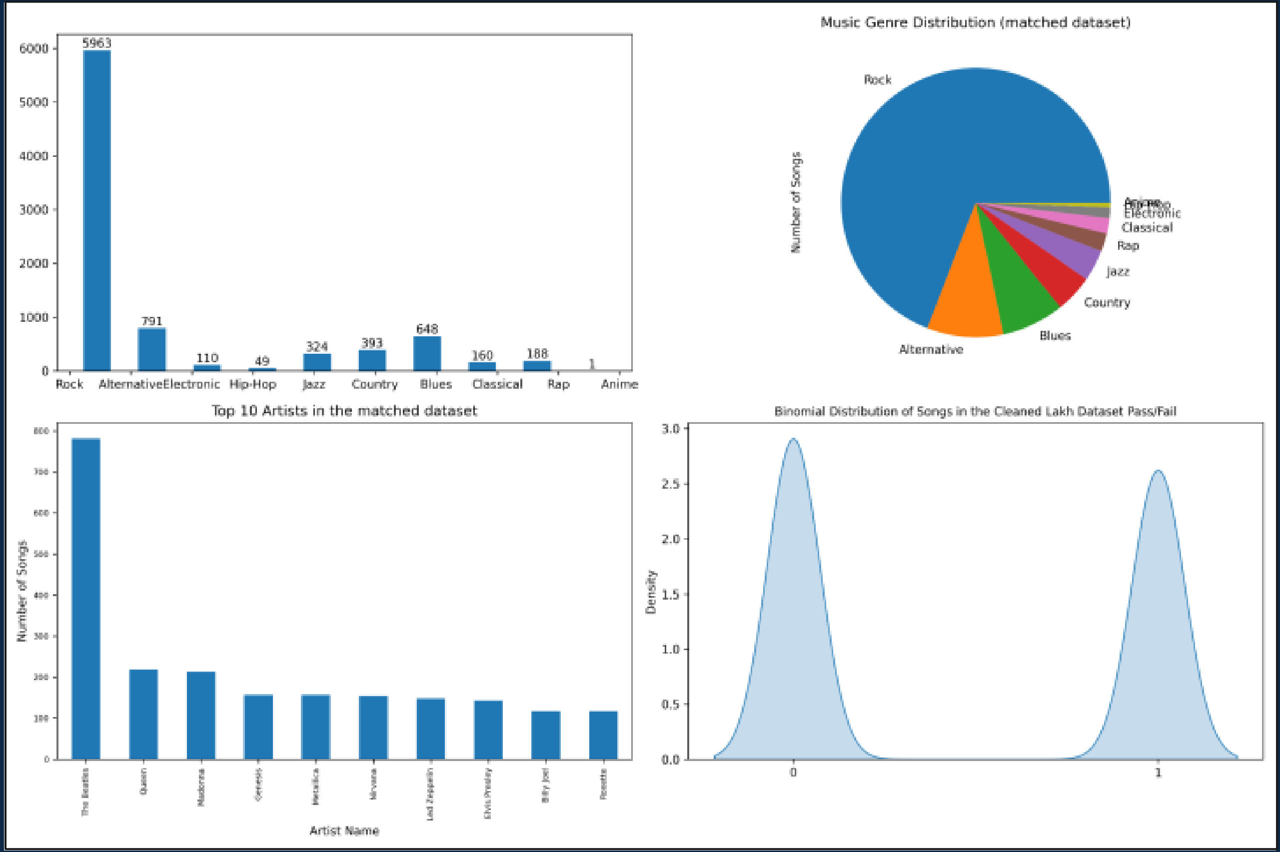
Combining with Genre Dataset

- Matching *music_genre.csv* with the songs in our dataset

	artist_name	title	file_path	songartist	num_genres	first_genre	second_genre
0	10,000 Maniacs	A Campfire Song	Cleaned Lakh\10,000_Maniacs\A_Campfire_Song.mid	10,000 Maniacs - A Campfire Song	1	Rock	Rock
1	10Cc	Dreadlock Holiday.1	Cleaned Lakh\10cc\Dreadlock_Holiday.1.mid	10Cc - Dreadlock Holiday.1	1	Rock	Rock
2	10Cc	Dreadlock Holiday.2	Cleaned Lakh\10cc\Dreadlock_Holiday.2.mid	10Cc - Dreadlock Holiday.2	1	Rock	Rock
3	10Cc	Dreadlock Holiday.3	Cleaned Lakh\10cc\Dreadlock_Holiday.3.mid	10Cc - Dreadlock Holiday.3	1	Rock	Rock
4	10Cc	Dreadlock Holiday.4	Cleaned Lakh\10cc\Dreadlock_Holiday.4.mid	10Cc - Dreadlock Holiday.4	1	Rock	Rock

52% of songs in dataset matched with music_genre.csv

Results of Matching Dataset



Using Full Dataset

- We don’t want to use only 52% of the data from Cleaned Lakh, so we will use the entire dataset and clean the information

	artist_name	title	file_path	songartist	num_genres	first_genre	second_genre
17226	Zucchero	Un Piccolo Aiuto Feat. Gerard Depardieu	Cleaned Lakh\Zucchero\Un_piccolo_aiuto_feat._G...	Zucchero - Un Piccolo Aiuto Feat. Gerard Depar...	1.0	Random	Random
17227	Zucchero	Voodoo Voodoo.1	Cleaned Lakh\Zucchero\Voodoo_voodoo.1.mid	Zucchero - Voodoo Voodoo.1	1.0	Random	Random
17228	Zucchero	Voodoo Voodoo.2	Cleaned Lakh\Zucchero\Voodoo_voodoo.2.mid	Zucchero - Voodoo Voodoo.2	1.0	Random	Random
17229	Zucchero	Voodoo Voodoo	Cleaned Lakh\Zucchero\Voodoo_voodoo.mid	Zucchero - Voodoo Voodoo	1.0	Random	Random
17230	Zucchero	You Make Me Feel Loved	Cleaned Lakh\Zucchero\You_Make_Me_Feel_Loved.mid	Zucchero - You Make Me Feel Loved	1.0	Random	Random

Building our Features

analyze_songs function

- Function that uses the music_processors file
- Takes the dataframe with .mid files and adds the extracted features into new columns

music_processors.py file

- A file that contains all of the code necessary for analysis
- The .py file contains a few functions and mappings
  - Functions analyze a song and return some information
  - Mappings of genres and artists

The music_processors.py file makes use of the music21 Python library for midi file processing

Some Extracted Features

Key: The key of the song extracted and returned as a case-sensitive string. (ex. The keys of “f minor” and “Eb Major” → “f” and “E-”)

Ambitus: $Highest\ Note\ Pitch - Lowest\ Note\ Pitch$

Tonality Percentage: $\dfrac{\#\, of\, Notes \in Key \mid Note \in Song|}{\#\, of\, Notes \in Song}$

Crowdedness: $\dfrac{\#\, of\, Notes\ Played}{Length\, of\, Song}$

Enriching the Dataset P.2

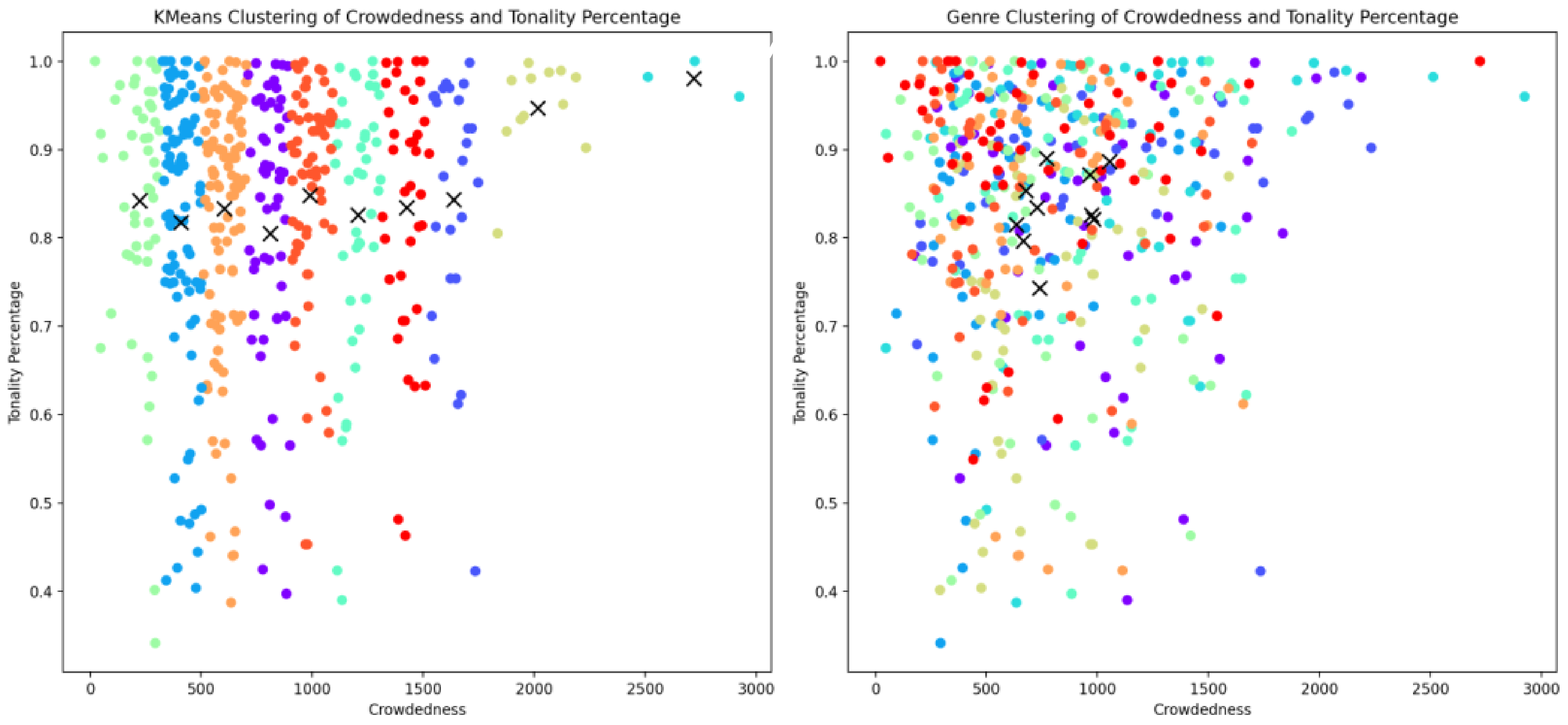
Creating a Dataset from Scratch

artist_name	title	file_path	songartist	num_genres	first_genre	second_genre	Key	Mode	Time Signature	...	Tempo Changes	Ambitus	Instrument Names	Has Drums	Pitch Names	Most Played Note	Amount of Notes	Highest Note Repetition	Crowdedness	Tonality Percentage
Michael Jackson	Liberian Girl	Cleaned Lakh\Michael_Jackson\Liberian_Girl.mid	Michael Jackson - Liberian Girl	1.0	Random	Random	g	minor	4/4	...	0.0	62.0	Piano, Guitar, Vocal, Bass, Drums	True	G, G, F, G, G, D, C, C, C, D, D, F, G, G, F, G...	B-	1943.0	0.229542	526.655263	0.981987
Michael Jackson	Off The Wall	Cleaned Lakh\Michael_Jackson\Off_the_Wall.mid	Michael Jackson - Off The Wall	1.0	Random	Random	e-	minor	4/4	...	0.0	69.0	Piano, Guitar, Vocal, Bass, Drums	True	C#, E-, C#, E-, C#, E-, C#, E-, F#, C#, E-, C#...	E-	6185.0	0.188682	1112.545732	0.423767
Michael Jackson	Remember The Time	Cleaned Lakh\Michael_Jackson\Remember_the_Time...	Michael Jackson - Remember The Time	1.0	Random	Random	b-	minor	4/4	...	0.0	29.0	Bass, Electric Bass	True	E-, E-, C#, C, G, C, C, E-, F, C, F, E-, C#, C...	B-	969.0	0.253870	338.486301	0.750258
Michael Jackson	Smooth Criminal	Cleaned Lakh\Michael_Jackson\Smooth_Criminal.mid	Michael Jackson - Smooth Criminal	1.0	Random	Random	a	minor	4/4	...	0.0	68.0	Vibraphone	True	A, A, A, A, G, A, B, B, A, A, B, C, C, B, C, G...	A	3398.0	0.269865	975.663366	0.978517
Michael Jackson	Shes Out Of My Life	Cleaned Lakh\Michael_Jackson\Shes_Out_Of_My_Li...	Michael Jackson - Shes Out Of My Life	1.0	Random	Random	A	Major	4/4	...	6.0	57.0	Piano, Guitar, Vocal, Bass, Drums	True	B, E, B, A, E, B, A, E, E, G, G#, B-, B, G#, A...	E	1587.0	0.211720	421.794020	0.882168

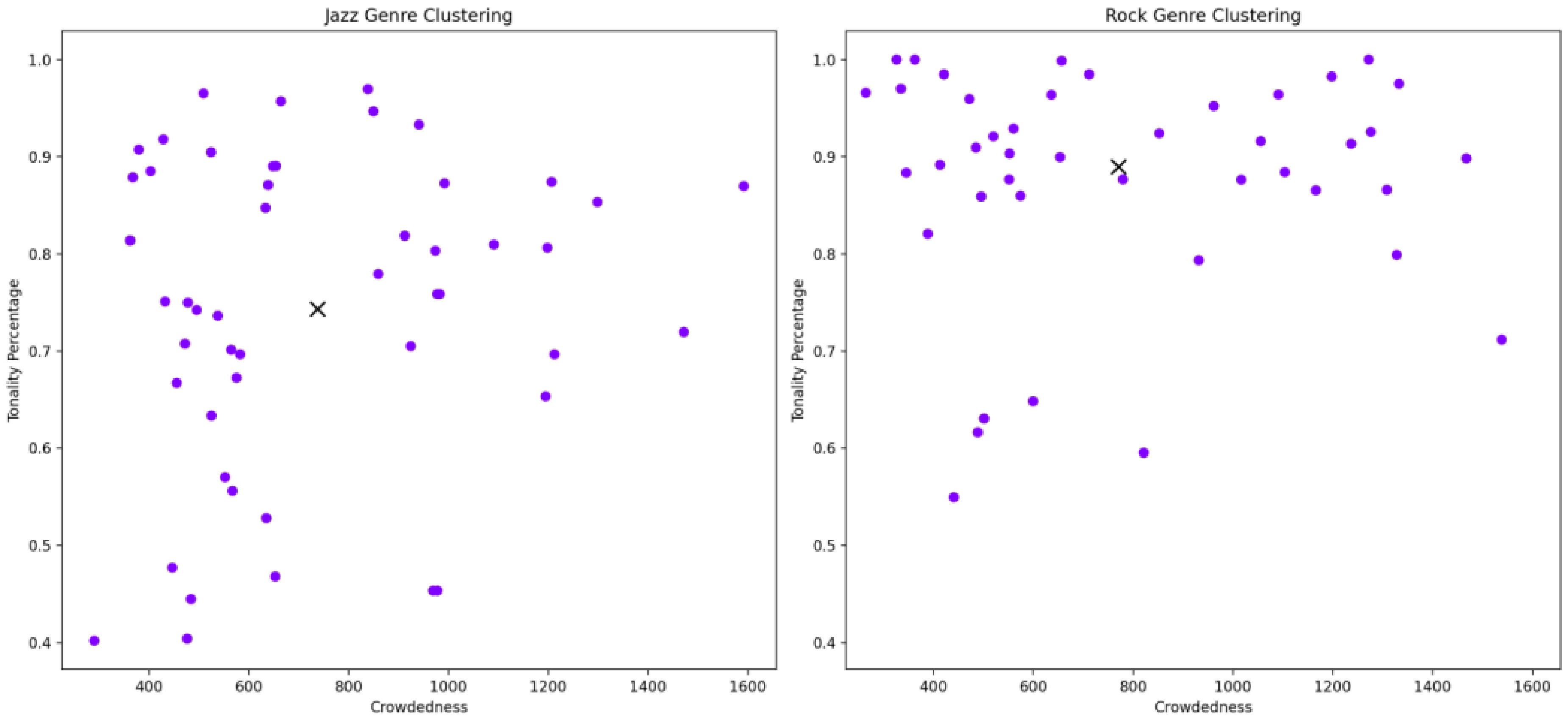
Analysis of the full dataset (17,000) songs, would have taken around 70 hours to complete. So we decided to select a sample of 500 songs

Visualizations P.2

K-Means of Crowdedness vs Tonality



Jazz/Rock Crowdedness vs Tonality %



Selecting the Right Model

Performing five-fold Cross Evaluation on the training split data with three multi-classification models

Random Forest

- Cross Eval Score
  - **0.39**

Logistic Regression

- Cross Eval Score
  - 0.33

SVM

- Cross Eval Score
  - 0.35

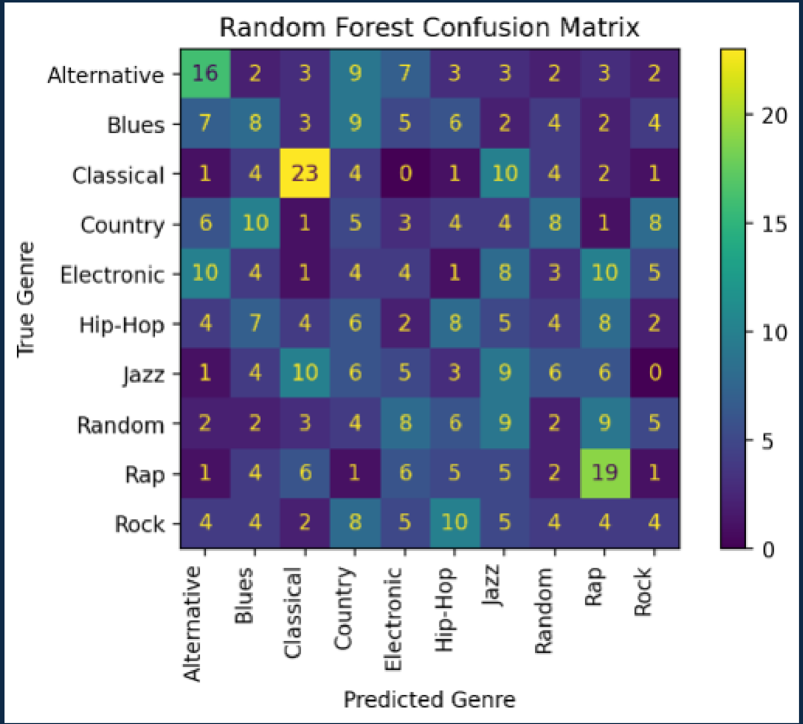
Random Forest Wins!

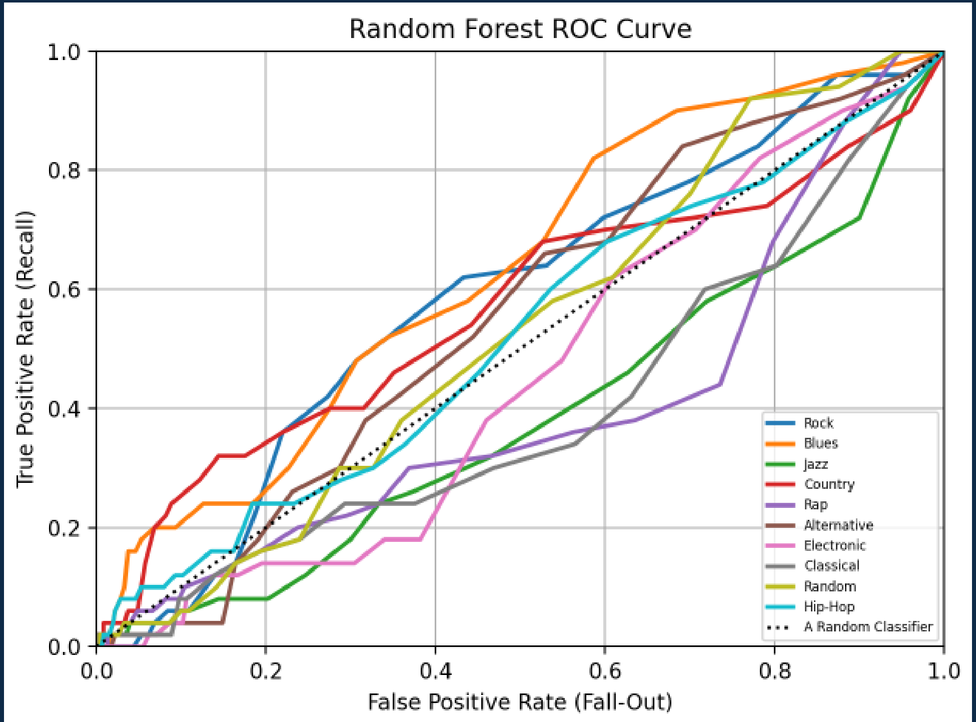
Random Forest Classifier

- Performed a Randomized Search to get a local optimal solution for the max_features, and n_estimators hyperparameters
  - max_features = 2
  - n_estimators = 75
- Added a ClusterSimilarity feature to our model
  - Output the distance to the cluster center of the genre

Visualizing Model Accuracy

Calculated metrics: Precision = 0.196, Recall = 0.196, F1 = 0.189





Discussion and Evaluation

	mean							
	Amount of Notes	BPM	Crowdedness	Duration	Has Drums	Highest Note Repetition	Tempo Changes	Tonality Percentage
first_genre								
Alternative	3696.98	115.5718	971.564031	4.022311	0.78	0.262842	4.40	0.826252
Blues	3654.48	118.9404	966.478791	4.402281	0.76	0.267992	1.86	0.871437
Classical	2697.66	104.4696	633.924945	5.980914	0.12	0.187758	15.12	0.815164
Country	3732.24	119.6374	1054.584142	3.708471	0.66	0.265680	1.48	0.886378
Electronic	4673.96	115.4820	982.834540	6.425394	0.74	0.303277	5.42	0.821038
Hip-Hop	3099.98	98.2842	665.774669	4.684735	0.68	0.240776	1.12	0.796769
Jazz	2694.40	115.0204	738.208390	3.945990	0.70	0.197807	1.92	0.743439
Random	3516.46	102.2400	728.425193	4.951828	0.60	0.241277	1.18	0.834544
Rap	2545.58	106.5320	677.613299	3.855838	0.64	0.251144	2.20	0.853752
Rock	3416.52	121.1862	770.567555	6.663548	0.72	0.284194	5.64	0.890032

Pivot table of the numerical categories of the dataset
Features of note: Jazz Tonality %, Classical Drums proportion, Classical Tempo Changes

Conclusion

- We were able to successfully clean our data set, but only certain genres were able to be correctly classified due to the broad umbrella of many of these genres
- We did run into difficulties when attempting to process the MIDI files and experienced extremely long run times with certain functions, which did affect our prediction modeling
- We did discover how certain genres, specifically rap and classical, are much more easily classified than others due to distinct differences
- However, we hope that his model could be further developed to distinguish between more similar genres, and possibly even generate songs