Predicting the Popularity of Spotify Songs

Team: Aashvi Manakiwala, Akash Jain, Arjun Agarwal, Ashwin Balaji, . **Project Mentor TA:** Chandler Cheung

1) Abstract

For this project, we study the problem of predicting the popularity of a song, which is important for guiding up and coming artists in their journey to fame. In addition to predicting the popularity of artists, we want to add an element of interpretability to our models to highlight important metric-based and lyrical features of a song that makes it popular. Our main contribution is creating a dataset of song related features such as danceability, valence, and energy as well as lyrics based features such as sentiment. In the first part of the project we explore the impact of various song related features on popularity prediction and successfully improve the MSE through feature engineering. In the second part of the project, we merge a lyrical dataset and extract features related to the sentiment of the songs. We explored the impact of the sentiment feature on XGBoost, neural network, and logistic regression performance. Our XGBoost was modeled in the following manner, $\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)$, where fk(xi) is the prediction in the kth decision tree for the ith song. We also did an analysis on whether similar song lyrics yield similar popularity scores through a KNN on similarity scores. Our KNN model followed the following format, $\hat{y}_{new} = \frac{1}{K} \sum_{j=1}^{K} y_{N_j}$, where yNj is the popularity score of the jth nearest neighbor and K represents the number of nearest neighbors. For our results, we noted that our genre one-hot encoding seemed to decrease MSE marginally, while dimensionality reduction with PCA seemed to decrease MSE slightly more. We found an ideal number of components in the PCA to be 20 with respect to balancing the number of components as well as variance. We mainly noticed a strong correlation between the sentiment conveyed in song lyrics and the accuracy of the XGBoost model used in assessing the relationship between features and popularity, noted by the significant decrease in MSE after additional sentiment analysis was performed.

2) Introduction

Motivation: Being an artist has never been tougher than it is today. The talent of an artist is just one of many factors that will determine if his/her songs will reach a mainstream audience; as such, artists are constantly experimenting with not only the musicality of their songs but also their style of promotion and presence on social media. By conducting this analysis, we hope to give artists more insight into how best they can maximize their chances of success.

Problem Setup: We propose to predict the popularity of a song (measured by Spotify's internal popularity metric that ranks songs from 0-100) as a function of song metrics (ex: danceability, energy, etc), and embedded lyrics. We also intend to analyze the impact of certain features on popularity. Generally, our inputs and outputs are described as such:

Datasets:

- **Spotify Songs** - The Kaggle dataset that we are using has 114K rows and 21 columns: 20 of which will be inputs into our regression model and one (the popularity column) which will be the output vector we are trying to predict. We'll explore various regression models, tune hyperparameters, and utilize broad feature engineering to maximize the predictive accuracy of our models.
- **Genius Lyrics:** This is a Kaggle dataset containing lyrical information for 5 million songs. It draws from the popular website, Genius Lyrics. In addition to lyrics, this dataset contains information about song title, artist, language, genre, and year released. Though these metrics are already present in the above dataset, the lyrical information provides a rich foundation for sentiment analysis and similarity score calculation, which we will further discuss in the report.

Intermediary features and processing:

- **Categorical Encoding:** This allows some of the categorical columns in our dataset, such as genre and explicit, to be incorporated into the modeling.
- **Artist count feature:** This is a feature that we engineered, drawn from our prior understanding that songs with multiple featured artists tend to gain more public attention from combined fanbases.
- **Lyrical sentiment:** Lyrics are a great teller of a song's sentiment, which may be connected to popularity.
- **PCA:** PCA is a preprocessing technique that we used to reduce dimensionality from the categorical encoding of genres.

Outputs:

- **Popularity Prediction:** Using this data and regression-based models, we intend to predict the popularity of a song. The resulting performance metric is mean squared error, as this conveys the distance between predictions and actual values.
- **Classification**: Another approach we took to the analysis is classifying which range of popularity a song will fall under (low, medium, high).

- **Interpretability:** To enhance interpretability, we chose transparent models such as XGBoost and included feature importances. We also performed a KNN algorithm on only song lyrics to evaluate the predictive power of similar song lyrics to popularity.

3) Background

Overall past work on song popularity prediction Has used datasets limited to just song lyrics or song metrics such as duration, tempo, danceability etc. We aim to enhance this analysis by using both song metrics and lyrical data.

Angela Xue, Nick Dupoux "Predicting A Song's Commercial Success Based on Lyrics and Other Metrics" (https://cs229.stanford.edu/proj2014/Angela%20Xue,%20Nick%20Dupoux,%20Predicting%20the%20Commercial%20Success%20of%20Songs%20Based%20on%20Lyrics%20and%20Other%20Metrics.pdf)

This project done at Stanford predicts a song's' "hotness" using genre, duration, artist familiarity, artist hottnesss, year, tempo, danceability, energy, loudness, key, time signature. The project employs KNN, Linear regression and L1 penalized regression in the prediction of hotness which is a value from 0 to 1. The overall results showed that hotness prediction was somewhat effective from the given dataset but there was much room for improvement by using a more predictive and diverse set of features. In the future work section of this project they mentioned that more interesting predictions could have been made using sentiment scores of song lyrics. They pointed out that song genres turned out to be the most predictive feature in their data set and that lyric information could correspond to this and enhance this information. This has inspired our part 2 contribution, where we incorporate sentiment scores into the analysis.

Mohamed Nasreldin, Stephen Ma, Eric Dailey, Phuc Dang, "Song Popularity Predictor", 2018. (https://towardsdatascience.com/song-popularity-predictor-1ef69735e380)

This project is focused on predicting song success on the Billboard Top 100 using a multifaceted approach incorporating audio analysis, artist-related, and song-related features. Leveraging the XGBoost algorithm, the previous work achieved a notable accuracy of 0.63, benchmarked against alternative models such as logistic regression, random forests, and KNNs. This team encountered challenges stemming from a skewed dataset containing only 1,200 non-hit songs. Therefore, our goal is to further refine predictive capabilities by training the model on a more comprehensive and diverse range of song data. By doing so, we aim to enhance the accuracy and robustness of our predictive model, ultimately advancing the understanding of factors contributing to song success on the Billboard charts and Spotify.

4) Summary of Our Contributions

We propose to use our regression analysis to give more insight into what factors of a song matter (and do not matter) in evaluating the possibility of success and popularity. Through both contributions below, we'll look at numerical metrics (danceability, energy, acousticness) as well as lyrical embeddings and sentiment analysis scores.

5) Detailed Description of Contributions

5.1 Methods

Part 1: Predicting Popularity using Spotify Songs Dataset

Our goal is to effectively predict the popularity of a song. We are using a Kaggle dataset containing songs from Spotify and various metrics such as popularity (target), danceability, energy, etc. In addition to predicting the popularity of a song, we want to explore what makes a song popular and what features are most important in this decision. To incorporate this research aspect, we are also going to incorporate some EDA and feature importance sections in this contribution. Our feature selection methods will be referencing [Romero, 2024], who applied SVD to reduce dimensionality and explored the correlation between various features and popularity.

We will distribute this part in the following sections:

Data cleaning

- Approach: We will explore the null values in the dataset and evaluate whether to drop the column or drop the rows.
- Result: There were very few null values in the dataset, so we decided to drop the null rows, still maintaining a dataset size of almost 114,000 rows.

EDA

- Approach: We will explore the distributions of features in the dataset with respect to high and low popularity (determined by a threshold we will decide).
- Result: We visualized the distribution of popularity scores. It seems to have a gaussian distribution with the exception of a spike in popularity scores of 0.

Data Preprocessing

- Approach: If there are categorical features in the dataset, we will experiment with encoding them to incorporate them in the model. This is a tradeoff between more information and increased dimensions that could make the dataset sparse. We will also create a training/testing split for the model.
- Result: We created binary encoding for the explicit column and a one hot encoding for the genres column. This introduced 109 columns in the dataset. We have created a 80/20% training/testing split for the model.

Baseline model

- Approach:
  - We are using XGBoost as a baseline model as this is commonly used in the industry and is known to have good empirical results. Then, we will compare the mean squared error of the encoded and non-encoded data to see whether encoding offers better performance.
  - Feature importance - we will list the feature importance from both training sets
- Results
  - XGBoost had a lower test MSE on the training set with encodings, despite the increased dimensions. The MSE on the training set with encodings was 323.8 and the MSE without encodings was 367.5
  - We also plotted the percentage of predictions that fell between a varying margin of error from the correct popularity score. Popularity scores range from 0 to 100. The model achieves a 90% accuracy for a margin of error of around 30, which is quite large. In this plot, the green line represents the one-hot encoded training set and blue represents the original.
  - Feature importance: The most important features in the model without encodings were

    explicit: 0.11594360321760178
    acousticness: 0.08888020366430283
    time_signature: 0.08230465650558472
    instrumentalness: 0.08042598515748978
    danceability: 0.07778016477823257

    The most important features including encodings were related to specific genres. This is because feature importance is calculated based on the information gained from each feature, so it makes sense for certain genres to be predictive of popularity. This was also discussed in one of our prior works.

Hyperparameter Tuning

- Approach: We will use some version of grid search/cross validation to find the optimal combination XGBoost parameters for better MSE.
- Result: Currently, we are experimenting with ranges of values for max tree depth, subsample ratio, and gamma (how conservative leaf node splits are). Due to computational limitations, we were unable to compute more combinations of parameters. However the optimal combination we found was {'gamma': 2, 'max_depth': 5, 'subsample': 0.6}. This produced a MSE of 303.

Feature Engineering

- Approach: To augment the dataset with more predictive features, we will use our background knowledge of the music industry to engineer features from the existing dataset.
- Results: We have added features for the number of artists in a song and the total number of songs created by the artist. The former indicates a positive correlation between artists and fanbase size. The latter indicates a positive correlation between total songs and how established the artist is. These additional features have reduced the MSE to 289.8

Principal Component Analysis

- Approach: PCA is useful in reducing the dimensionality and multicollinearity in a dataset. This may be a useful technique if we choose to encode the genre column.

- Results: We inspected the relationship between the number of principal components and MSE. We found 25 principal components to be optimal

XGBoost after feature engineering and PCA

- Approach: This will quantify the improvement in performance from these steps
- Result: The resulting MSE from 25 components was 318. However, we decided to use all components because this was able to run relatively fast and produce an even higher MSE of 309.

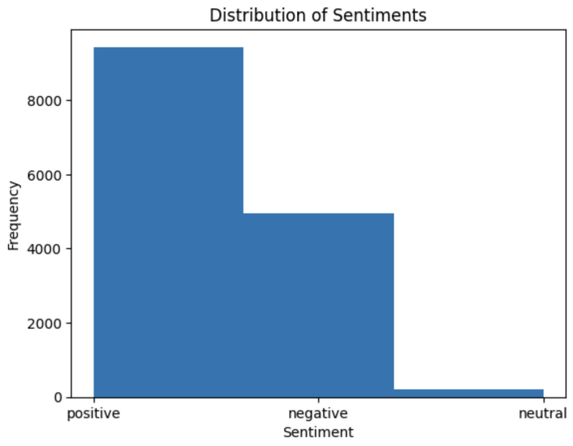
Part 2: Enhancing popularity prediction using song lyrics

Merging Genius Lyrics Dataset with Spotify Songs Dataset

The lyrics data set contained 5 million rows and was 3 GB large. Therefore we outsourced the preprocessing for this on Kaggle, which will be discussed later. In order to combine the two dataframes we had to explode the Spotify Song data to produce multiple rows for songs with multiple artists. Upon doing so we merged the two data frames using title and artist. This was a unique combination that we thought was sufficient to match the data given that there were no common IDs between them.

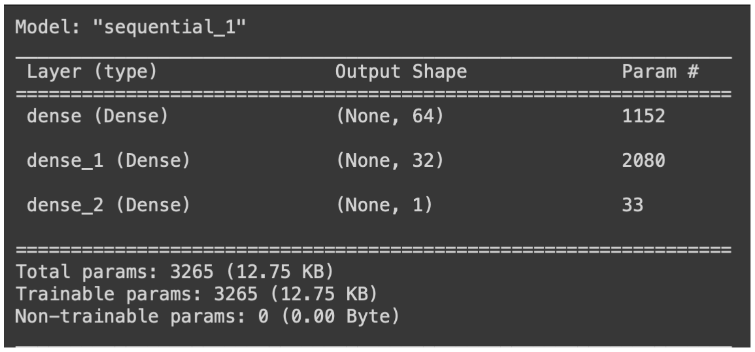
Feature Engineering: Sentiment

Our first step, motivated by the prior work, was to create a sentiment column using song lyrics. We produced this sentiment score using TextBlob which is a library that streamlines sentiment analysis and other NLP related tasks. For each row in our dataframe, we were able to produce a polarity score using this library, and then categorized the polarity into positive, negative and neutral labels.



Most of the sentiment labels were positive and negative with very few neutral. Since this is just one feature in our dataset we decided that it was not vital for us to balance the classes. After producing the scores we ran the dataset on XGBoost and saw a huge improvement in MSE to 189.

Lastly we ran the data on a neural network of the following architecture in hopes of diversifying our models.



```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 64)                1152

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 1)                 33

=================================================================
Total params: 3265 (12.75 KB)
Trainable params: 3265 (12.75 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

We ran this model on 50 epochs and used a learning rate of 0.001. The resulting MSE was higher indicating that this network was not structured well enough to properly capture the patterns in the data. Perhaps in the future some further tuning could make this a more potent model.

Logistic Regression: Classifying popularity into high, medium and low

We thought it would be interesting to see the logistic regression popularity classification abilities using our data set. We experimented with two target variables: the first being hard-coded popularity ranges and the second being popular revenge is based on popularity percentiles. We tested the classification power with and without the sentiment column. Surprisingly, we saw a higher accuracy without the sentiment column.

Lyric Similarity Analysis

Finally we were interested in the predictive power of lyrics on popularity. We created a K Nearest Neighbor model to inspect whether songs with similar lyrics would have similar popularity scores. The first step was to create embeddings for our songs. This was done using a library called Sentence Transformers, from which we loaded in a model called MiniLM. This is a lightweight model that was suitable for our computational restrictions. After making the embeddings we created a custom distance function for similarity scores of two lyric embeddings. Finally, we ran the KNN on lyrical embeddings as the X variable and popularity as the target variable, with our custom distance function. Our resulting MSE was 747 indicating that lyrical similarity alone is not a good predictor of popularity. One caveat is that we only use 1,000 randomly sampled rows in this analysis to speed up the algorithm. This section builds upon this notebook, which takes in new lyrics and identifies the most similar song in the dataset.

5.2 Experiments and Results

Regression

The output prediction is a popularity score

Model	Inputs	MSE	Takeaways
XGBoost	Numerical Spotify data	367	Baseline model chosen for empirical success with regression
XGBoost	Numerical Spotify data, genre and explicit encodings	323	Genre had an impact on prediction
XGBoost	Numerical Spotify data, genre and explicit encodings, PCA	318	PCA helped reduce dimensions from genre encoding
XGBoost	Numerical Spotify data, genre encodings, hyperparameter tuning	303	Hyperparameter tuning chose more optimal parameters
XGBoost	Numerical Spotify data, genre and explicit, tuning, encodings, artist count, song count	289	Artist count is the number of artists involved in each song and song count is the number of songs produced by each artist - had an impact on MSE
XGBoost	Numerical Spotify data, genre encodings, artist count, song count, sentiment	183	Lyrical sentiment had a large impact on MSE.
Neural Network	Numerical Spotify data, genre encodings, artist count, sentiment	333	The neural network was not as effective in prediction
KNN	Embedded lyrics, using similarity score as distance function	747	Lyrics alone are not predictive of popularity.

Classification

The output prediction is a

Model	Inputs	Outputs	Accuracy	Takeaways
Logistic Regression	Numerical Spotify data, genre encodings, artist count	bucketed popularity score with automated buckets	82.7%	
Logistic Regression	Numerical Spotify data, genre encodings, artist count, sentiment	bucketed popularity score with automated buckets	54.1%	Sentiment scores actually reduced the accuracy
Logistic Regression	Numerical Spotify data, genre encodings, artist count	Hardcoded bucket ranges	82.7%	Hardcoded bucket ranges did not have an impact on the accuracy score
Logistic Regression	Numerical Spotify data, genre encodings, artist count, sentiment	Hardcoded bucket ranges	54.1%	Hardcoded bucket ranges did not have an impact on the accuracy score

6) Compute/Other Resources Used

In order to preprocess the lyrical data we needed to use Kaggle's compute resources. This is because the dataset was 3 GB and had around 5 million rows, making it too large to load into Google Colab. On Kaggle, we viewed the original lyrics dataset, dropped null values, and merged this data with the Spotify song data from part 1. On this kaggle notebook we had to preprocess the Spotify song data set in order to merge. This involved exploding the dataframe based on artists so that a song with multiple artists would have multiple rows. Since the two data sets were from different sources Spotify and Genius lyrics, the unique key we used to merge was the combination of song title and artist. Upon doing so we ended up with around 30k rows, which was much more manageable to work with on Google Colab.

7) Conclusions

The main boosts in model performance came from feature engineering artist count and song count, as well as adding a sentiment feature. This aligns with the understanding that more experienced artists are more likely to have popular songs. Additionally, songs with multiple artists will have popular songs due to the combined fan bases. The sentiment column provided an improvement in MSE as well. Some future steps would be evaluating the distribution of sentiment within popular and unpopular songs. Lastly, another takeaway is that similarity in song lyrics do not offer much predictive power for popularity.

Some interesting areas for further analysis would be to cluster songs using the lyrical similarity scores and see whether the clusters can provide further predictive power. Additionally, more work can be done with the use of KNNs to evaluate whether similarity of lyrics and other features combined can predict popularity. For instance, a distance metric taking into account embeddings similarity combined with genre and tempo could provide insights. Finally, our lyric embeddings are a good starting point for other NLP related feature engineering, such as identifying the presence of specific emotions (sorrow, happiness, etc).

In a world where Spotify and other music streaming platforms are so commonly used, we hope that our predictive modeling harnesses this data to empower artists and music producers to tailor their creations to align with audience preferences. These models can potentially democratize the music industry by providing opportunities for emerging artists in niche genres. Findings from our models should be taken with a grain of salt, however. We do not want to prescribe a template for popular songs but rather enhance artists' understanding of the streaming space.

(Exempted from page limit) Other Prior Work / References (apart from Sec 3) that are cited in the text:

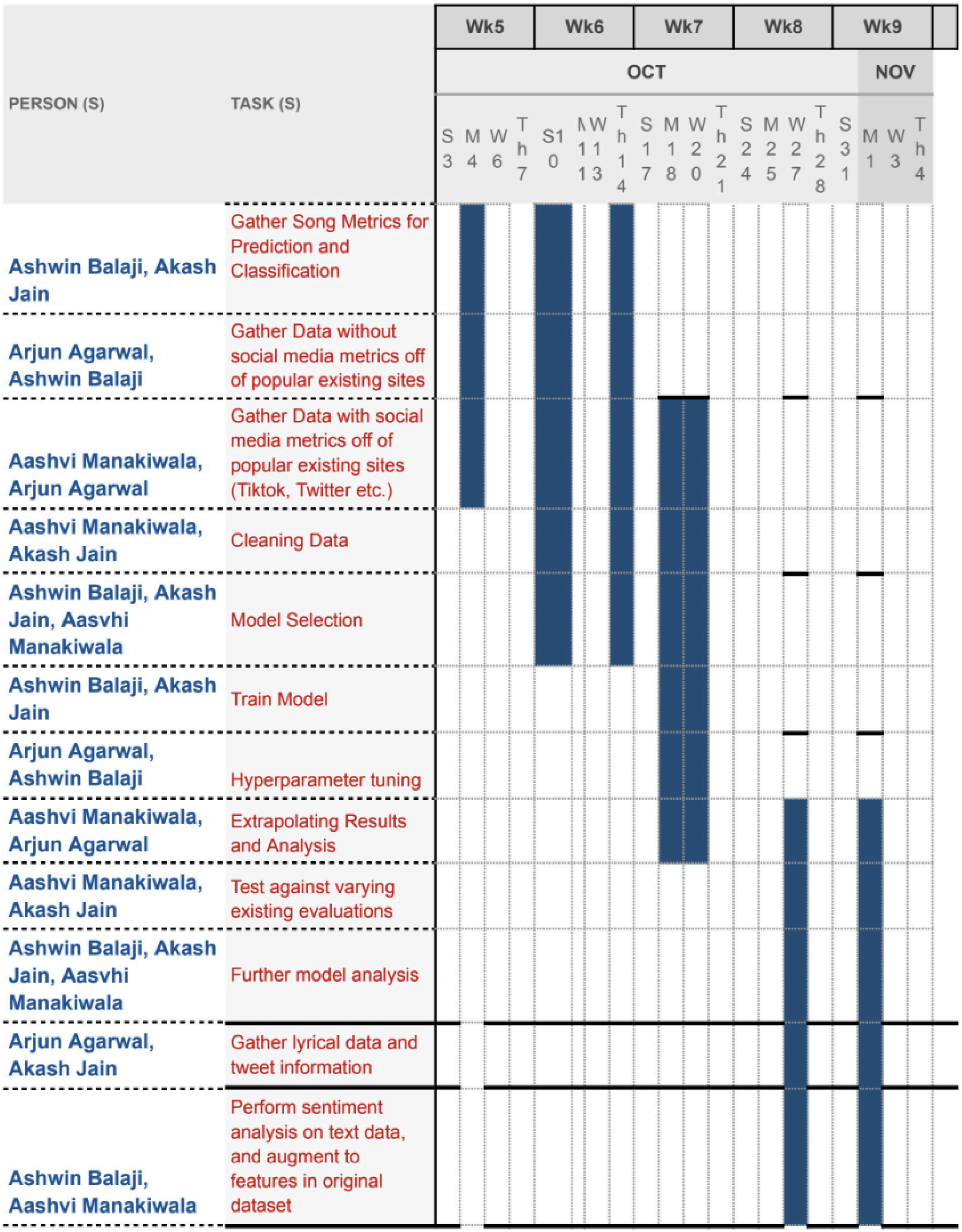
1. Angela Xue, Nick Dupoux “Predicting A Song’s Commercial Success Based on Lyrics and Other Metrics” (https://cs229.stanford.edu/proj2014/Angela%20Xue,%20Nick%20Dupoux,%20Predicting%20the%20Commercial%20Success%20of%20Songs%20Based%20on%20Lyrics%20and%20Other%20Metrics.pdf)
2. Mohamed Nasreldin, Stephen Ma, Eric Dailey, Phuc Dang, “Song Popularity Predictor”, 2018. (https://towardsdatascience.com/song-popularity-predictor-1ef69735e380)
3. Romero, “What Produces a Musical Hit, Data Selection” (https://www.kaggle.com/code/victormarmolromero/what-produces-a-musical-hit-data-selection)
4. Holmes, “MDM Song Lyrics”, 2024 (https://www.kaggle.com/code/benholmes0/mdm-song-lyrics)

Broader Dissemination Information:

Your report title and the list of team members will be published on the class website. Would you also like your pdf report to be published?

[✓ YES] / NO

(Exempted from page limit) **Work Report: This may look like your GANTT chart from the midway report, with more completed steps now. Okay to modify.** (Mark completed steps in green, as shown here. For convenience, you may split into two charts, one till Nov 8, and another for after Nov 8, placed one below the other.)



PERSON (S)	TASK (S)	Wk5				Wk6				Wk7				Wk8				Wk9			
		OCT																	NOV		
		S 3	M 4	W 6	Th 7	S 10	M 11	W 13	Th 14	S 17	M 18	W 20	Th 21	S 24	M 25	W 27	Th 28	S 31	M 1	W 3	Th 4
Ashwin Balaji, Akash Jain	Gather Song Metrics for Prediction and Classification		■			■			■												
Arjun Agarwal, Ashwin Balaji	Gather Data without social media metrics off of popular existing sites		■			■			■												
Aashvi Manakiwala, Arjun Agarwal	Gather Data with social media metrics off of popular existing sites (Tiktok, Twitter etc.)		■			■			■		■	■									
Aashvi Manakiwala, Akash Jain	Cleaning Data					■			■		■	■									
Ashwin Balaji, Akash Jain, Aasvhi Manakiwala	Model Selection					■			■		■	■									
Ashwin Balaji, Akash Jain	Train Model										■	■									
Arjun Agarwal, Ashwin Balaji	Hyperparameter tuning										■	■									
Aashvi Manakiwala, Arjun Agarwal	Extrapolating Results and Analysis										■	■				■			■		
Aashvi Manakiwala, Akash Jain	Test against varying existing evaluations															■			■		
Ashwin Balaji, Akash Jain, Aasvhi Manakiwala	Further model analysis															■			■		
Arjun Agarwal, Akash Jain	Gather lyrical data and tweet information															■			■		
Ashwin Balaji, Aashvi Manakiwala	Perform sentiment analysis on text data, and augment to features in original dataset															■			■		

PERSON (S)	TASK (S)	Wk10				Wk11				Wk12				Wk13				Wk14			
		OCT																	NOV		
		S 3	M 4	W 6	Th 7	S 10	M 11	W 13	Th 14	S 17	M 18	W 20	Th 21	S 24	M 25	W 27	Th 28	S 31	M 1	W 3	Th 4
Aashvi	Merging Lyrics with existing dataset		🟩	🟩																	
Ashwin, Aashvi	Sentiment Analysis		🟩	🟩																	
Ashwin, Aashvi	Neural Network Analysis		🟩	🟩		🟩	🟩														
Arjun, Aakash	Neural Network Further analysis						🟩					🟩	🟩								
Arjun, Aakash	Logistic Regression								🟩	🟩											
Arjun, Aakash	Logistic Regression Analysis								🟩	🟩											
...	Task 7																				
...	Task 8																				
...	Task 9																				
...	Task 10																				

(Exempted from page limit) Attach your midway report here, as a series of screenshots from Gradescope, starting with a screenshot of your main evaluation tab, and then screenshots of each page, including pdf comments. This is similar to how you were required to attach screenshots of the proposal in your midway report.

(Exempted from page limit) Supplementary Materials if any (but not guaranteed to be considered during evaluation):

Added above