# Machine Learning Term Project Proposal

**Project Title:** Predicting YouTube Music Video Views Using Song Attributes and Statistics From Spotify

**Group Members:** Ava Vellozzi, Aaryan Anand, Devika Kumar, Sofia Lynch, Abbhinav Sriram
Team Leader: Ava

**Dataset:**

Description of dataset: The dataset includes information on songs from Spotify and their corresponding YouTube music videos. It contains 26 features that describe various aspects of the song's musical characteristics, streaming performance, and video popularity. The key attributes are song metadata (i.e. Track, Artist, Album), Spotify metrics (i.e. Danceability, Energy, Tempo), and YouTube engagement metrics (i.e. Views, Likes, Comments). The dataset has over 20000 values.

Source: https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube. The song, artist info, and data taken from Spotify in 2023.

Type of data: The dataset has numerical, categorical, and boolean values for the features.

Why it's relevant: Understanding what makes a music video/song go viral on YouTube is valuable for artists, producers, and marketers. By analyzing song attributes and statistics (e.g. speechiness, energy) from Spotify we can uncover trends that influence viewership. Predicting YouTube views based on these factors can help music industry professionals optimize their content, marketing strategies, and production decisions to reach a larger audience and increase their chances of success.

**Initial Plan:**

Problem we are addressing: Our goal is to create a predictive ML model that can estimate the number of views on a Youtube music video based on song attributes and statistics from Spotify. This could be useful for artists/music producers to determine what traits make a song go viral.

Methods and techniques we intend to use: Since we are trying to predict the number of views and likes for a video, this is a regression task. We would use models such as linear regression, polynomial regression, and DecisionTreeRegressor. To further improve accuracy of these models, we can use bagging techniques such as random forests, and boosting methods as well. We will assess model performance using evaluation metrics such as Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and R-squared. These metrics will help us understand the accuracy of our predictions and identify any

potential overfitting or underfitting issues. Additionally, we will do hyperparameter tuning using GridSearchCV to optimize model performance.

Preliminary steps:

- Load the dataset, visualize relation between features and target (views) using heatmaps, scatterplots
- Clean the dataset by handling outliers or any missing values, incorrectly labeled points
- Evaluate features that may not contribute significantly to predicting views and consider dropping them to reduce noise (feature selection techniques!)
- Adjust for categorical variables (such as artist name) by using LabelEncoder
- Split the data into training and testing sets
- Execute models, adjusting hyperparameters as necessary
- Record performance using metrics such as MAE, RMSE, and R-squared

---

**THINGS SAAD SUGGESTED:**

- Split the work into 2 tasks
    - (1) Regression Task
        - Output the number of views based on features like loudness, liveness, tempo, etc
            - Explore all ML models including linear regression, linear regression with polynomial features, ensemble technique, KNN
                - Gradient Boosting,
                - *Visualization Analysis
    - (2) Classification Task
        - SVM, Decision Tree classifier
        - Have the stream distribution and group the streams by magnitude. Have 5 classes/groups:
            - Group 1: 0 - 1 mil streams
            - Group 2: 1 - 10 mil
            - Group 3: 10 - 100 mil
            - Group 4: 100 mil - 1 bil
            - Group 5: 1 bil+
        - So based on these groups transform into classification task and then implement the ML models
- For the Features
    - Need to drop the 1st column (index/id column) in order to avoid overfitting

- Need to handle textual features (i.e. url spotify) and categorial features (i.e. album & single).
- Numerical features also need to be scaled

---

**SPLIT:**

**Regression:**

**Classification:**