Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

# CSU44061 Machine Learning

*Week 5 Assignment - Group Project Proposal*

*Spotify Modality Predictor*

| Group 72 Members | |
|---|---|
| **Student Name** | **Student ID** |
| Prathamesh Sai Sankar | 19314123 |
| Eligijus Skersonas | 19335661 |

School of Computer Science and Statistics,
Trinity College Dublin

# 1 Motivation

Our proposed application project called Spotify Modality Predictor will be based around predicting the modality of songs on Spotify [1]. Given a song on Spotify, our machine learning models will predict whether a song is written in a major key ($\hat{y} = +1$) or minor key ($\hat{y} = -1$). We chose this project because we enjoy listening to music in our free time and we find it interesting that songs contain unique characteristics that differentiate them from others. A logical project that came into mind was using these unique audio features to train machine learning models. We finalized on predicting modality for two reasons. Firstly, there were no previous projects done on this from our Google searches [2], therefore it was unique. Secondly, our proposed model could be used to benefit Spotify, Spotify Artists and also Spotify Users. Currently, Spotify Artists have to manually input their song's metadata when uploading their music. By predicting the modality of their song, it would save the artist some work as well as speed up Spotify's song uploading process. Also, the modality can be used to recommend songs to Spotify users (songs with a modality of 'major' tend to be happy while songs with a modality of 'minor' tend to be sad). The modality of a song is only available using the Spotify API for developers. Users of the Spotify app do not have any way of knowing if a song has a modality of major or minor. Hence, by using the song's audio features to predict the modality, users of the Spotify app can predict if a song will be to their preferred liking.

# 2 Dataset

We want our machine learning models to be trained and tested with data that contains various types of music to get the best predictions later on. We decided that the best way to do this is going to be using the Spotify API [3] to search for songs and scrape data. This gives us the opportunity to work with new data (there are over 80 million songs on Spotify [4]) as well as get experience with the messiness of handling such data. We will use the Spotify APIs "search" functionality [5] to search Spotify's public library of songs for the first 50 search results (i.e. songs) of every available searchable genre on Spotify (114 of them), resulting in 5700 songs. We will then use Spotify's "audio features" functionality [6] to get the audio features of each song we get from the search results (including the mode). For feature selection, we plan on using every combination of the 12 features returned by the API to identify important/unimportant features, as well as plotting each feature against the target value to visualize the dependence on each feature.

# 3 Method

We plan on using a kNN classifier as well as a kernelized SVM (support vector machine) as they both use distance based algorithms which suits the fact that songs with similar modality would have similar feature values hence using the distances between data points to determine their similarity is a good way to predict. We will use a kernelized SVM rather than a regular SVM to ensure we use two significantly different models to compare them both. We plan on choosing a range of values and cross validating the chosen values for each hyper-parameter to get the best value for the hyper-parameter. We will split the data into training and testing data using a 90/10 split. Then we will use the chosen hyper-parameters to train and test the models with the data.

## 4 Intended experiments

We will compare our two models with each other and also with a baseline predictor which predicts randomly to ensure they are better than a dummy model. Comparisons include calculating confusion matrices for each model to compare the recall, precision and accuracy of each model. Also, we'll plot the ROC curves and use the AUC values to further analyse the performance of our models. These evaluations will help us verify that our models actually make sensible predictions.

## 5 References

[1]: Spotify
[2]: No results on Google for this project description being done before
[3]: Spotify API
[4]: Spotify's total number of tracks amounts to over 80 million
[5]: Spotify APIs "search" functionality
[6]: Spotify APIs "audio features" functionality