# Evan Kerivan

## Mini Project Two

# Contents

# Problem Statement

## Predicting Pop Music Secret Sauce

- Annual music sales revenue $28.6B + touring +promotional + merchandise revenue
- Record labels spend millions on A&R
- The ability to accurately predict the popularity of a track would increase ROI in the music industry

# Dataset:

# ‘114000 Spotify Songs”

## Spotify offers API, Kaggel user leveraged API to create dataset of 114,000 records

| | | |
|---|---|---|
| track_id | The unique Spotify ID for each track. | |
| artists | Names of the artists who performed the track | |
| album_name | The name of the album in which the track appears. | |
| track_name | The title of the track. | |
| popularity | A value between 0 and 100 indicating the track's popularity based on recent plays. | |
| duration_ms | The length of the track in milliseconds. | |
| explicit | Boolean indicating whether the track contains explicit content. | |
| danceability | Describes how suitable a track is for dancing (0.0 = least danceable, 1,0 - most danceable). | combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity |
| energy | Represents the intensity and activity of a track (0.0 = low energy,,1.0 = high energy) | Represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. |
| key | The musical key of the track mapped using standard Pitch Class notation. | |
| loudness | Overall loudness of the track in decibels (dB). | |
| mode | Indicates the modality (major or minor) of the track. | |
| speechiness | Detects the presence of spoken words in the track. | |
| acousticness | Confidence measure of whether the track is acoustic (0.0 = not acoustic,1.0 = highly acoustic). | |
| instrumentalness | Predicts whether a track contains vocals (0.0 = contains vocals, 1.0 = instrumental). | |
| liveness | Detects the presence of an audience in the recording (0.0 = studio recording, 1.0 = live performance). | |
| valence | Measures the musical positiveness conveyed by a track (0.0 = negative,,1.0 = positive). | Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angryThis is determined by a Spotify model |
| tempo | Estimated tempo of the track in beats per minute (BPM). | |
| time_signature | Estimated time signature of the track (3 to 7). | |

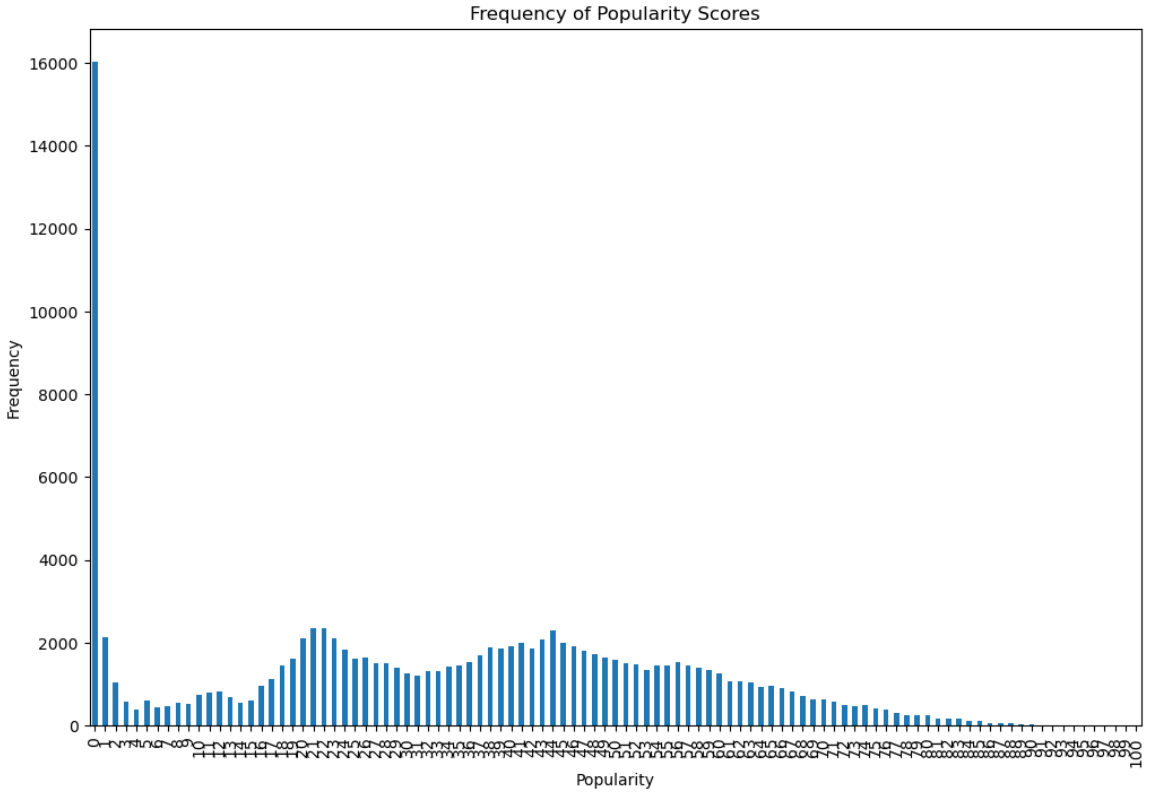
# Data Cleansing:

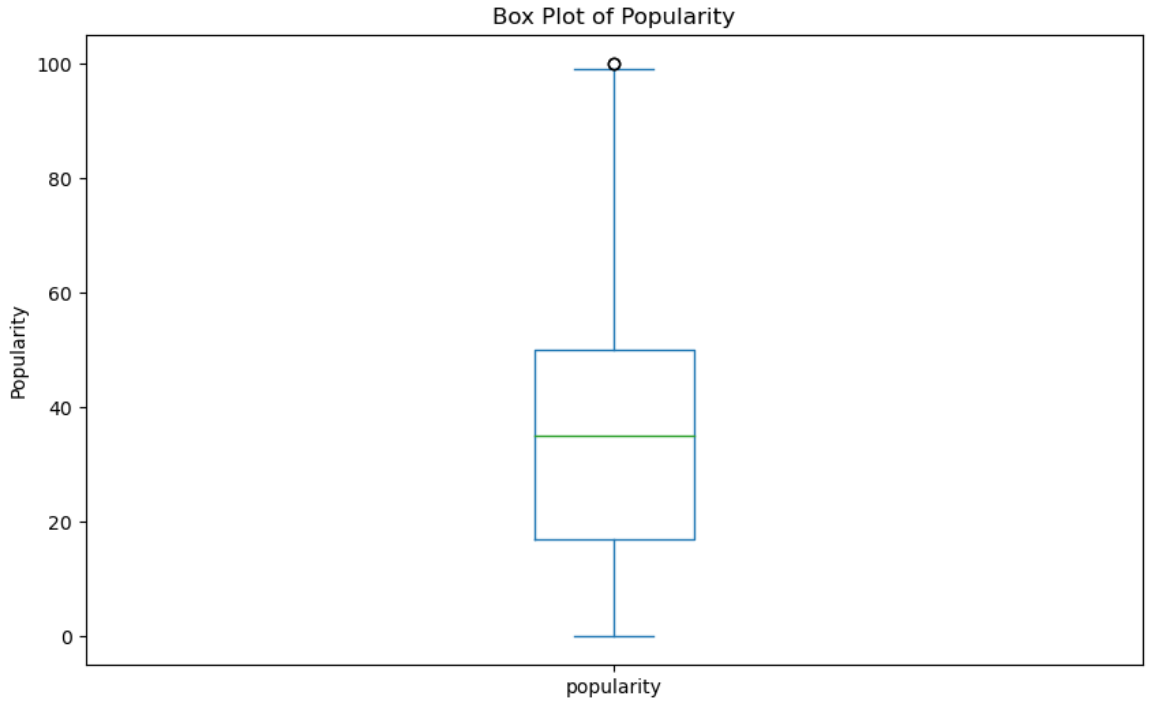
- Duplicate Track Ids dropped (25,000 dropped)
- Artist and Track name details dropped
- Otherwise data had no missing values
- Most features have values between 0-1
- No outliers
- Scaled other features
- Encoded genre feature

# EDA:

## Popularity

- Ranked with a score of 0-100
- Popularity is skewed with 16000 songs receiving a popularity score of zero

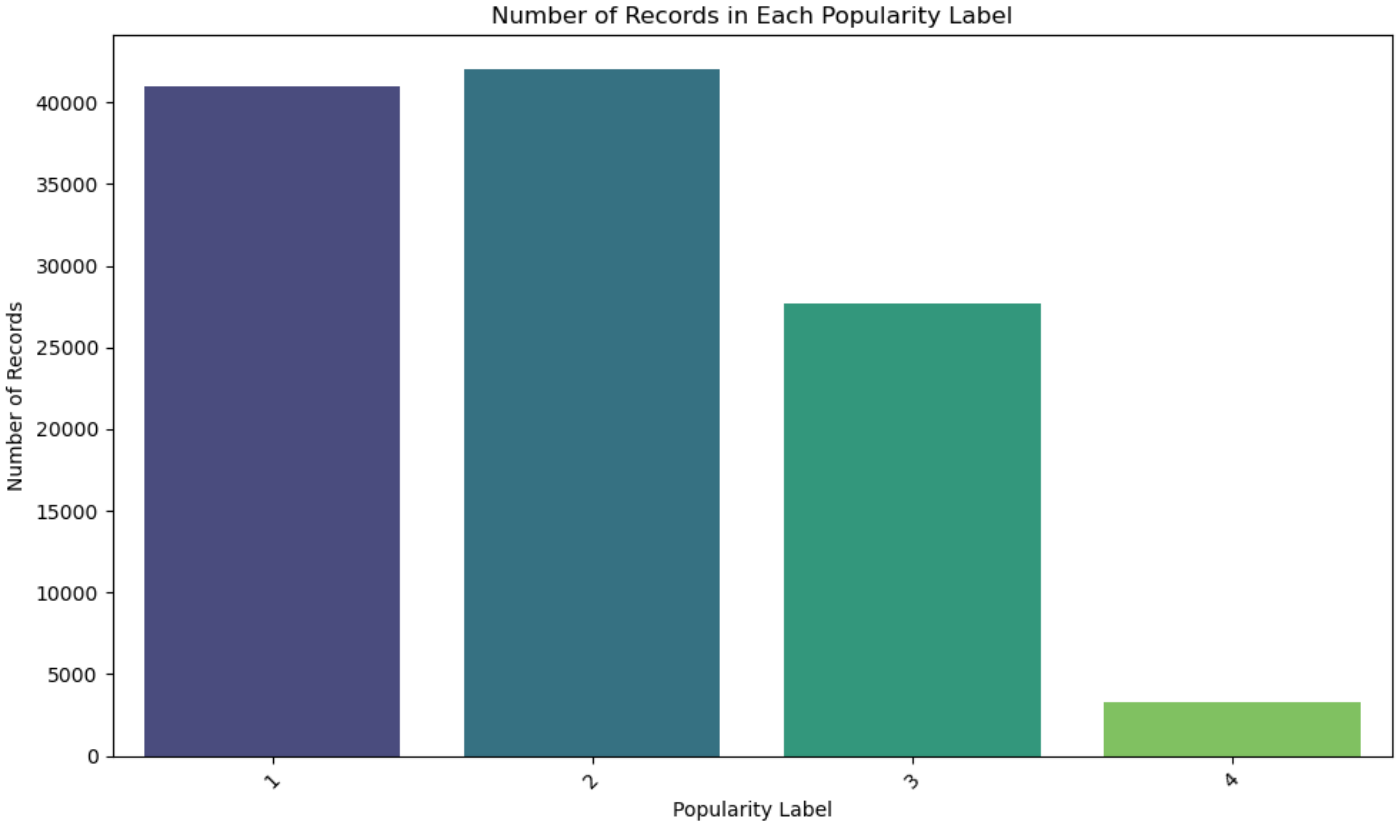


# EDA:

## Popularity

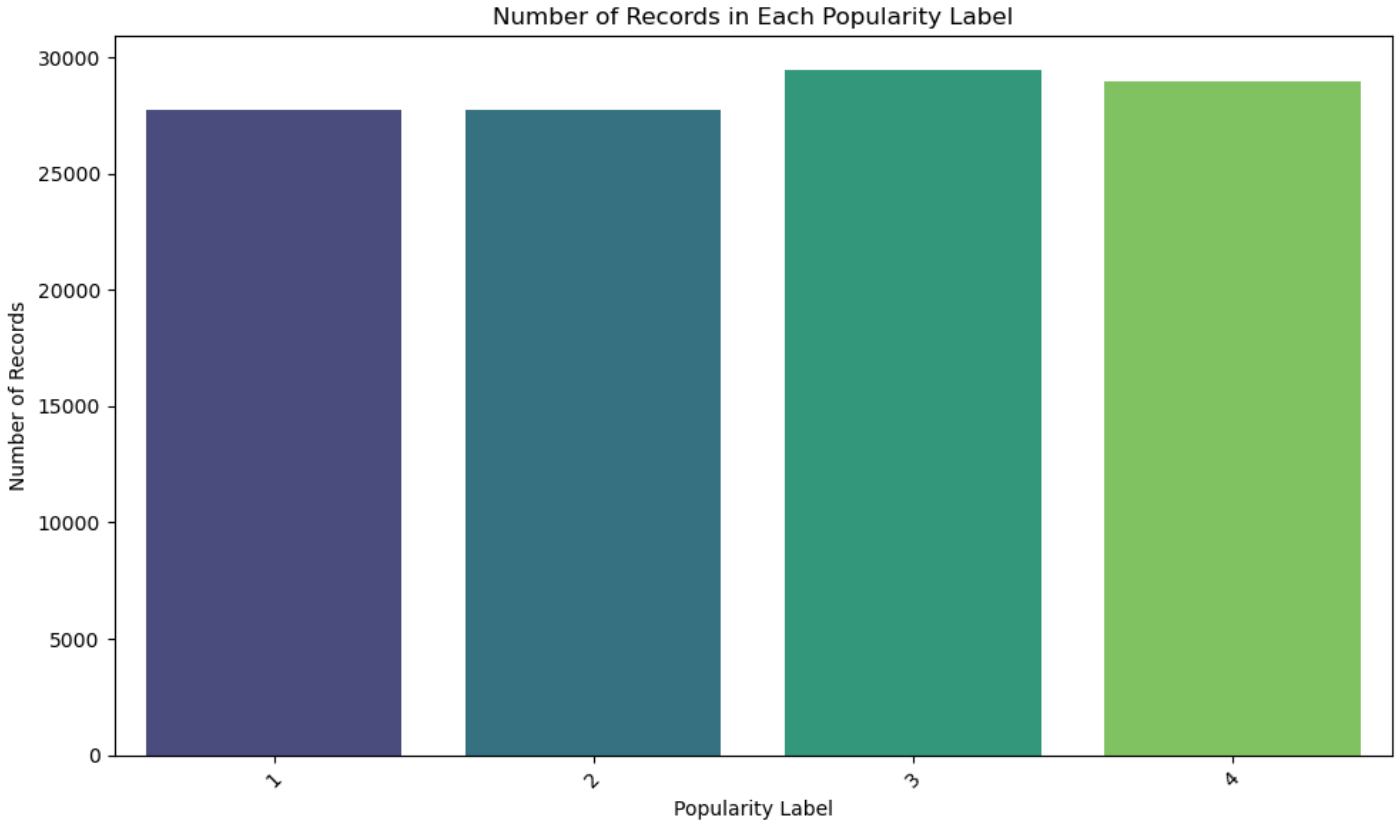
- When splitting the songs into 4 categories by score 0-24, 25-49,50-74,75,100 a vast majority fall in the lower 2 categories.
- This may make it difficult to predict a popular song because of the small sample size.



# EDA:

## Popularity

- Splitting into 4 categories that align with the quantiles is more sensible and balances the data set
- Being more popular than 75% of songs is a good measure of success.



# Models:

## Regression

- Logistic Regression
    - Training Accuracy= 99%
    - Testing Accuracy= 35%

- Random Forest Classifier
    - Training Accuracy 99%
    - Test Accuracy 61%

- Random Forest Classifier with randomised search
    - Training Accuracy 99%
    - Test Accuracy 64%

- XGBoost with randomised search
    - Training Accuracy 71%
    - Test Accuracy 65%

.

# Conclusion

## Spotify Popularity

- The model performs okay
- Care must be taken to not over fit
- With hyperprameter tuning the model performed reasonably well without overfitting
- For the model to answer the business question it is necessary to understand how Spotify assigns the scoring to each feature
- It may be possible to “reverse engineer” their models so a set of scores can be attributed to a new song.  Or the ability to run a song through Spotifiy’s model
- If Spotify’s model isn’t open source (doesn’t appear to be) it would require a model that can analyse the audio file likely using deep learning techniques
- This type of analysis is definitely happening in the music industry.