Understanding What Makes A Song Danceable: Insights From Statistical and Deep Learning Approaches?

Kevin Gabalebe

tg370@kent.ac.uk

Table of Contents

BACKGROUND AND MOTIVATION

Music is one of the most universally consumed forms of art. Whether through radio, party playlists, or personal headphones, music accompanies people in almost every environment. Music also spans across multiple genres and sub-genres, so with millions of songs out there, there will always be a form that anyone can find easy to engage with.

Commercially speaking, we have seen that danceability of a song is of key interest, this is a measure developed by algorithms (e.g Spotify) to quantify how suitable a track is for dancing based on its musical features such as rhythm stability, beat strength and regularity. Songs that are "danceable" tend to perform well commercially, whether they are used in viral social media trends or played in every party/club. They therefore receive a lot of publicity and replayability, so for streaming platforms, accurately predicting danceability can help improve their recommendation algorithms, increase user engagement, and assist artists/producers in replicating the music. Psychological research further suggests that high-tempo and beat clarity are more likely to induce movement and bodily synchronisation (Sensorimotor Coupling in Music and the Psychology of the Groove, Petr Janata, Stefan T. Tomic and Jason M. Haberman,2012), suggesting tempo as one of the key predictors in danceability.

The dataset used in this project was downloaded from Kaggle. The author derived this data from Spotify's publicly available data (Spotify Web API) and extracted all songs from 471 playlists with all the songs' audio features.

For this project, I attempt to investigate which audio features are most associated with a songs danceability. By converting danceability into a binary classification outcome, several machine learning models were evaluated, i.e. Logistic Regression (GLM), Support Vector Machines (SVM), Single-Layer Neural Network (NNET), and H2O's Deep Learning Framework. By comparing the linear, non-linear and deep learning approaches, the project aims to explore how different modelling techniques can capture the underlying structure of the audio features and determine which are most effective to classify a danceable song.

Ultimately, this project provides insight into:

1. How accurately can audio features predict whether a song is danceable using statistical and machine learning models?
2. Which music attributes are most strongly associated with danceability?
3. How well can modern Machine Learning models distinguish between danceable and non-danceable tracks in comparison to classical statistical approaches?

These findings illustrate how statistical learning methods can be applied to real-world music data, and how such models can complement the algorithms used by major streaming services.

DATA DESCRIPTION & PREPARATION

The dataset contains 32833 songs (rows) with 23 columns. Each observation represents a single song, containing information such as track name, artist, popularity, genre, release date, playlist extracted from and engineered audio features.

Before modelling, data cleaning was performed. Began by removing all duplicate songs (6603 songs), identified by using the track name and the track artist, to avoid biased representation of repeated entries. All rows with missing attributes where then removed to ensure data was clean. In addition, the track release date was converted into a numerical release year, enabling it to be used as a predictor. The outliers were retained as they represent genuine variation in the music world.

After cleaning the data, we outlined the Spotify audio features to be used as predictors across all machine learning models:

- Track Popularity- A numerical measure of listener engagement.
- Energy- Represents a perceptual measure of intensity and activity.
- Key- The estimated overall key of the track. Integers map to pitches using standard Pitch Class Notation.
- Loudness- The average loudness of the song in decibels (dB).
- Mode- Indicates the modality (major or minor) of a song.
- Speechiness- Detects the presence of spoken words in the song.
- Acousticness- A confidence measure from 0.0 to 1.0 of whether the track is acoustic.
- Instrumentalness- Presents whether a track contains no vocals.
- Liveness- Detects the presence of an audience in the recording.
- Valence- Measures the musical positivity conveyed by a track.
- Tempo- The overall estimated tempo of a track in beats per minute (BPM).
- Duration- The duration of the song in milliseconds.

The target variable danceability median was then used as a threshold to make a binary classification task. Songs with a danceability equal to or above the median were labelled as 1 (danceable), and those below were labelled 0 (not danceable). This ensures classes were balanced.

EXPLORATORY DATA ANALYSIS

Exploratory data analysis was then carried out to understand the structure of the Spotify dataset and to identify the patterns of the dataset. By looking at the distribution of the danceability, a skewed, unimodal shape was seen centred around 0.70. The distribution shows how broad the danceability scores are. The median was drawn at danceability=0.67, showing the threshold to be used to create the binary classification outcome.

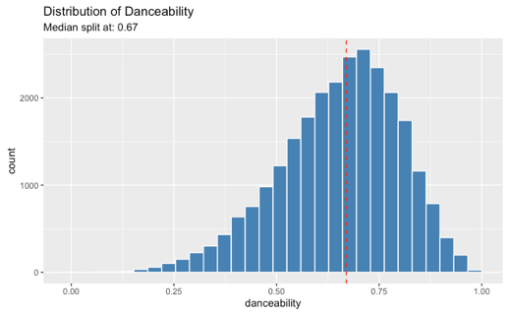

Fig 3.1: Distribution of Danceability

Predictor Distributions by Danceability Class

The boxplots then show that most predictors do not show strong separation between danceable (1) and non-danceable songs (0), other than Valence, which clearly increases for danceable songs. For other features, including energy, tempo, release year, we can see that the interquartile ranges and medians overlap, but can also be noticed that the upper quartiles of the non-danceable songs are slightly higher which could be due to the presence of the outliers.

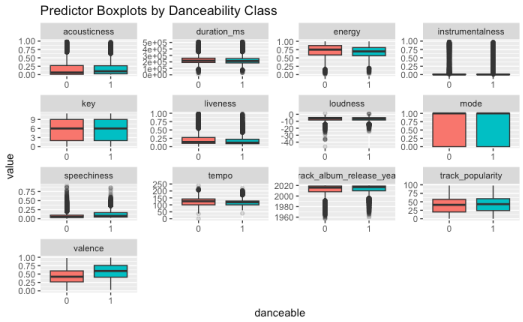

Fig 3.2: Predictor Boxplot by Class

But overall, this suggests that individual predictors offer limited ability to choose between the 2 classes on their own, and that classification is more likely to arise from interactions between features.

Correlation Matrix

To assess relationships among predictors, a correlation heatmap was drawn. We can see a strong positive correlation between energy and loudness as expected, and that happier songs tend to be more energetic. Some features, such as speechiness, showed weaker correlations with others, showing that they may contribute independently during classification. These show that a non-linear model may classify better as the predictors do not move together in linear patterns.

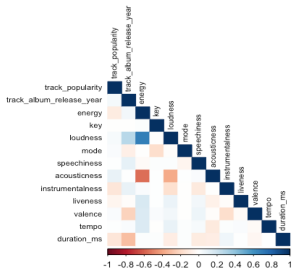

Fig 3.3: Correlation Matrix

METHODOLOGY

The project aims to classify songs as danceable or non-danceable using Spotify audio features. The modelling consists of data preprocessing, model training, parameter tuning and model evaluation.

Data Preparation

In addition to the data preparation done earlier, the data was then split into 70% training set and a 30% test set. All predictors were then centred and scaled using the mean and standard deviation of the training data, this mean and standard deviation was applied to both sets to ensure that models operate on similar scales. For Deep learning the train and test models were converted to H2O frames before modelling.

Model Training and Parameter Tuning

1.Logistic Regression (GLM)

A logistic regression model with a binomial link function was fitted to provide a linear baseline that is easy to interpret. The GLM estimates the log-odds of danceability as a linear combination of predictors assuming additive and linear relationships. Although EDA suggested weak separations for most features on their own (without interaction), this still provides insight into the predictors, i.e. Odds Ratios, the coefficient sign, p-values, these can all show how the predictors affect how danceable a song is.

This model does not have any parameters to tune.

2.Support Vector Machine (SVM, Radial)

An SVM model with a radial basis function (RBF) was then used to learn the non-linear boundaries. The RBF kernel allows the classifier to separate the classes using flexible curved boundaries in a high-dimensional space. By separating points based on similarity, it can automatically capture any interactions between predictors, which as seen by EDA, may be useful as the individual predictors show weak separation in a 1-dimensional space. It has 2 parameters, Cost (Controls how strict the SVM penalises misclassified points) and Gamma (Controls how far the influence of each training point goes, i.e. the curvature of the boundary), which control the smoothness of the boundary.

The Cost and Gamma were tuned by using the function tune.svm(), over 3 cost values (0.1,1,10) and 3 gamma values (0.01,0.05,1) using 5-fold cross validation. This trains all possible cost-gamma combinations on 4 different sections of the training data, and tests on 1 section of the test data, and searches for the combination with the lowest misclassification error.

3.Neural Networks (NNET)

A single hidden-layer neural network model was fit which uses a logistic sigmoid activation function, which introduces non-linearity by transforming the weighted sum of predictors (input) into 0 and 1's (output). Allows the model to learn the non-linearity brought by the need for interactions of the predictors. It has 2 parameters, Decay (Penalises large weights to prevent overfitting) and Size (Number of neurons in hidden layers- controls the complexity of the neural network), which help balance the model flexibility while preventing overfitting.

The Size and Decay were tuned by looping across all Sizes (3,5,7) and Decays (0,0.01,0.001) and finding the misclassification error of all neural network size and decay combinations and taking the combination with the lowest error.

4.Deep Learning (H2O)

A multi-layer neural network was then trained using H2O's deep learning framework to investigate whether hierarchical feature extraction could improve classification. This was included to test whether more layers could capture the non-linear interactions more effectively than a simpler architecture. A dropout was included to reduce overfitting by randomly deactivating neurons during training, ensuring the model does not rely heavily on specific pathways. The model learns by going through the h2o training data 50 times (epochs=50), and updates weights based on the loss computed from batches of data. The sizes of the layers represent the number of neurons in each layer.

The number of layers and the sizes of layers were manually tuned. I tested the model with layers (200,200), (50,50,50) and (128,64,32) and picked the one with the highest AUC. I also tested the model without a dropout and with a dropout.

Model Evaluation

All models were evaluated on the test data set. For each classifier, the predicted probabilities for danceable=1 were used to calculate:

- Accuracy
- Sensitivity (True Positive Rate)
- Specificity (True Negative Rate)
- Receiver Operating Characteristic (ROC) curves
- Area Under the Curve (AUC)

AUC was then used as the main performance measure as it evaluates the model's ability to discriminate across all possible thresholds, providing a more reliable assessment. The confusion matrices were used to summarise each classification performance, and the ROC curves were compared to assess the ability to separate classes across the models. Variable Importance was also calculated for Deep Learning

(using the H2O importance scores), Logistic Regression (using the absolute z-scores and calculating the odds ratios), SVM and Neural Networks (by randomly shuffling each predictor values and seeing the drop in AUC of the model).

MODEL RESULTS AND INTERPRETATION

The models performed rather equally with exception to the Logistic Regression which showed noticeably lower Accuracy, Sensitivity and AUC compared with the other 3 models. This aligns with the EDA findings, showing weak separation of the weak linear separation of the predictors.

Logistic Regression (GLM)

	Model <chr>	Accuracy <dbl>	Sensitivity <dbl>	Specificity <dbl>	AUC <dbl>
Accuracy	GLM	0.6863244	0.6861999	0.6864493	0.7512694

Fig 5.1: Logistic Regression Performance

GLM produced the weakest performance among the models, with lower AUC as seen in Fig. 5.1, indicating that it struggled to distinguish danceable songs from non-danceable. From the table we can also see that the GLM truly predicted 68.6% of the danceable songs as danceable (as seen by its Sensitivity) and 68.6% of the non-danceable songs were correctly predicted as non-danceable (as seen by the Specificity). The GLM also gave us underlying information about the variable importance of the data.

	OddsRatio	OddsRatio0.1	CI_lower	CI_upper	CI0.1_lower	CI0.1_upper
(Intercept)	1.0060940	1.0006077	0.9743913	1.0388303	0.9974091	1.0038168
track_popularity	1.0548375	1.0053529	1.0205830	1.0902584	1.0020395	1.0086789
track_album_release_year	1.4877242	1.0405243	1.4299550	1.5483677	1.0364115	1.0446899
energy	0.5765759	0.9464238	0.5457704	0.6089497	0.9412413	0.9516081
key	0.9850908	0.9984990	0.9536424	1.0175677	0.9952646	1.0017430
loudness	1.1944060	1.0179236	1.1356473	1.2564572	1.0128015	1.0230922
mode	0.9442698	0.9942821	0.9140147	0.9755213	0.9910494	0.9975247
speechiness	1.3110290	1.0274513	1.2675386	1.3563227	1.0239910	1.0309469
acousticness	0.7354217	0.9697363	0.7069781	0.7648953	0.9659187	0.9735543
instrumentalness	1.2455682	1.0222021	1.2021324	1.2907644	1.0185802	1.0258520
liveness	0.8295416	0.9814853	0.8017946	0.8580658	0.9781519	0.9848091
valence	2.3362190	1.0885575	2.2504179	2.4260358	1.0844919	1.0926718
tempo	0.7453053	0.9710317	0.7211434	0.7701957	0.9678369	0.9742269
duration_ms	1.0502228	1.0049123	1.0140277	1.0877321	1.0013940	1.0084449

Fig 5.2: Logistic Regression Odds Ratios and Intervals

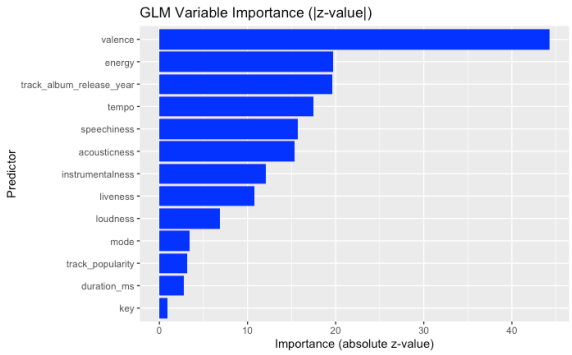

Fig 5.3: Logistic Regression Variable Importance Graph

From the Fig 5.2 we can see the odds ratios of each variable, with an increase in valence by 0.1 increasing the odds of a song being danceable by 1.0886, which is an additional metric to be used as most of the predictors have values between 0 and 1. It can be seen that Valence, Release Year, Popularity, Speechiness, Instrumentalness and Duration all have a positive increase in the odds ratio, meaning increasing these values lead to an increase in the chances of the song being danceable, and all the other predictors having a negative effect. The strength of the effects differs as seen by the Variable Importance graph based on the Absolute z-value shown on Fig 5.3, with predictors like Valence

having the largest impact on the song being danceable, followed by energy, release year and tempo. Showing that more faster and energetic songs are more danceable, and additionally that the more recent the song is, the higher the chances of it being danceable. With predictors like key, duration, track popularity and mode being on the other end in terms of strength in predicting how danceable a song is. Can also be noted that only the Key can have either a negative or positive effect on a song with a Confidence Interval spanning from 0.9536 to 1.01757.

Support Vector Machine (SVM, Radial)

	Model <chr>	Accuracy <dbl>	Sensitivity <dbl>	Specificity <dbl>	AUC <dbl>
Accuracy	SVM (Radial)	0.7451703	0.7516489	0.7386653	0.8251358

Fig 5.4: SVM Performance

With a tuned SVM that produces the best gamma and cost combination as 0.05,1 respectively, we can see the increase in all values as seen in Fig. 4.4. We can see that the AUC went up to 82.51% and the sensitivity and specificity went up to 75.16% and 73.87% respectively. Showing that a non-linear decision boundary is more appropriate for this dataset.

By randomly shuffling each predictor and destroying its relationship with the target and looking at the drop in AUC, we can see that the results have a similar pattern to that of GLM. Showing that the Valence, Energy and Tempo had the largest drop in AUC when these predictors are permuted, indicating that these are the most influential for distinguishing danceable songs from non-danceable. This shows that even with the non-linear boundaries in SVM, these predictors are still the strongest predictors.

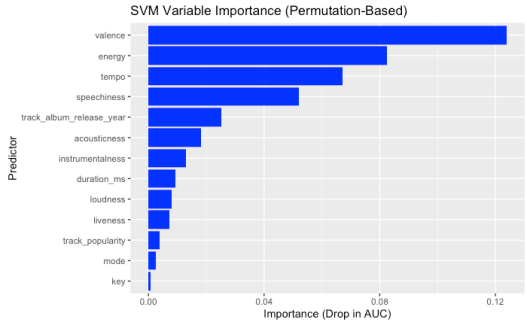

Fig 5.5: SVM Variable Importance Graph

Neural Networks (NNET)

	Model <chr>	Accuracy <dbl>	Sensitivity <dbl>	Specificity <dbl>	AUC <dbl>
Accuracy	Neural Network	0.7475852	0.7592593	0.7358635	0.8256424

Fig 5.5: Neural Network Performance

Neural Networks had similar results to those of SVM. With a size of 7 and a decay of 0, being the combination with the lowest error. We see that the AUC = 82.56% was very similar to that of the SVM, and the similarities spread to the sensitivity and specificity which were 75.92% and 73.59% respectively.

By performing the same permutation-based method as in SVM, we see that the pattern of results is similar. But this time the energy, tempo and speechiness proved more important with a larger drop in AUC, showing that the neural network adds more emphasis to the interactions involving these predictors compared to SVM. Showing its flexibility as it identifies a broader pattern in comparison to GLM.

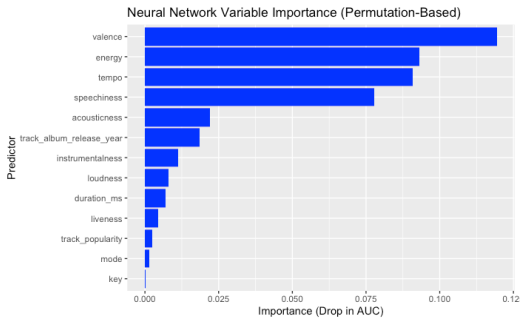

Fig 5.6: Neural Network Variable Importance Graph

Deep Learning (H2O)

	Model <chr>	Accuracy <dbl>	Sensitivity <dbl>	Specificity <dbl>	AUC <dbl>
Accuracy	Deep Learning (H2O)	0.7295374	0.8305429	0.6281202	0.8171486

Fig 5.7: Deep Learning Performance

Using a deeper model with 3 layers of sizes (128,64,32) seemed to have provided a small drop in AUC to 81.17%, but a major increase in sensitivity, going up to 83.05%. Meaning it predicts 83% of the true danceable songs as danceable, but the specificity of 62.81% may mean that it struggled to identify non-danceable songs and tended to over-predict the positive class. This shows that a deeper network leads to better results in terms of finding the danceable songs, but in terms of generally distinguishing between a danceable song and a non-danceable song it had the same performance as the 2 other non-linear models. I also noticed that without a dropout, these values all go down and the same occurred with the other layer/size options, highlighting the role in preventing overfitting and stabilising learning. Overall, this shows that deeper architectures provide limited additional benefit over the simpler architectures.

Using the H2O relative importance plot, we see that deep learning identifies tempo as the most influential predictor, followed by speechiness. The top 6 is still similar, with a change in the order, meaning that more layers helped this model learn a different structure in the data in comparison to the other models.

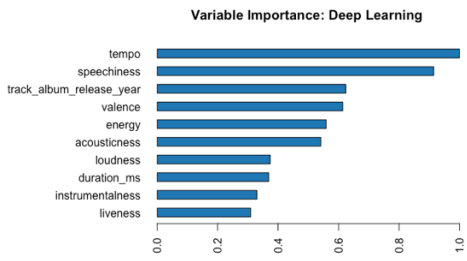

Fig 5.8: Deep Learning Variable Importance Graph

Model Comparison

	Model <chr>	Accuracy <dbl>	Sensitivity <dbl>	Specificity <dbl>	AUC <dbl>
Accuracy	GLM	0.6863244	0.6861999	0.6864493	0.7512694
Accuracy1	SVM (Radial)	0.7451703	0.7516489	0.7386653	0.8251358
Accuracy2	Neural Network	0.7475852	0.7592593	0.7358635	0.8256424
Accuracy3	Deep Learning (H2O)	0.7295374	0.8305429	0.6281202	0.8171486

Fig 5.9: Model Performance Comparison

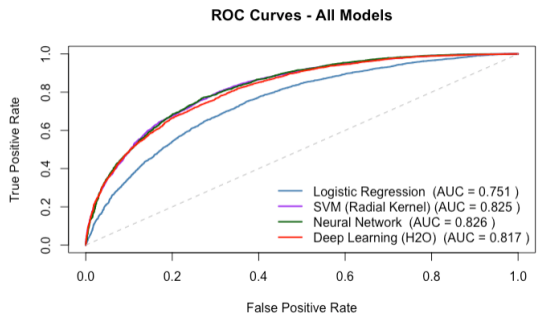

Fig 5.10: ROC Curves Comparison

The four models were chosen to represent the ability to classify danceable songs as we increase the flexibility of the models. With the GLM being the baseline linear model, the SVM then tests for the non-linear boundaries, followed by the neural network capturing the non-linear interactions automatically based on patterns, and the deeper learning model tests if a deeper model would improve the predictability. Comparing these models, therefore, answers the research questions based on which variables are most important, and the ability of machine learning models to predict over classical statistical testing techniques.

After testing all models, we analysed the ROC Curve. By comparing each model's False Positive Rates with its True Positive Rates, we get a graph of that structure in Fig. 5.10, with better performing models being further from the grey-dashed line. We can notice the underperformer was the GLM, as shown by its ROC Curve being the closest to the diagonal reference line, which is due to this GLM models' inability to handle the interactions. The 3 other models that capture the non-linearity then all had similar performances, which is shown by the ROC curves basically overlapping and the similar AUCs, with the single layer Neural Network having the highest ability to distinguish between danceable songs and non-danceable (highest AUC). The Deep Learning model, however, had the best ability to identify danceable songs of all models though there was a sacrifice in the specificity and accuracy.

All 4 models additionally showed us that the top 4 music attributes for predicting danceability are tempo, speechiness, valence, acousticness, release year and energy. Valence was the most important predictor in all models except Deep Learning, which instead had tempo as the most important predictor. But this shows that these 4 predictors are the most important in classifying whether a song is danceable.

Due to the models used, the numbers may not be perfectly reproduced as above, but general results will follow. I also noticed that the tuning of the SVM took the longest run-time, which is why a 5-fold cross-validation was used instead of the normal the 10-fold. Due to the processing time, NNET is the best, having the highest AUC, Specificity and Accuracy with the second least processing power.

CONCLUSION

This project sets out to predict whether a song is danceable using a range of audio features extracted from Spotify. The exploratory analysis revealed that most predictors did not have the ability to separate between classes on their own, with only valence showing a clear shift between danceable and non-danceable songs. This structure was shown in the modelling results, with the classical linear model displaying the worst ability to distinguish.

Although psychological research accredited tempo as the best predictors of danceability, from the EDA to the first 3 models, we saw that Valence instead proved to be the most important predictor, with tempo and energy following it. It was only in the Deep learning model where we got to see tempo being more prevalent using a deeper model.

In terms of the models, we did see that the modern models were significantly better at identifying the underlying relationships of the dataset. With Neural Network being the best in terms of the AUC and Specificity, while the multilayer deep learning network having the best Sensitivity. But there was not that much to separate these modern models, showing that increasing the model depth did not significantly improve overall discrimination given the dataset.

In summary, the analysis shows that danceability can be predicted moderately well from Spotify's audio features with non-linear models. Valence, energy, tempo and speechiness emerged as the most influential techniques, indicating that both musical positivity and faster upbeat music play important roles in danceability. However, the overall discriminative power cannot capture all the track structure suitable for dancing, showing that the need for additional predictors like actual lyrics positivity conveyed, the beat clarity (as mentioned in the research paper in the Background), and other features may enable models to learn stronger performance and improve the ability to predict.

APPENDIX

The codes, plots and more details can be accessed on my github link :

https://github.com/KevinGabalebe/MAST-6100-Final-Project