"TuneTrends"

A Comprehensive Study on Regression, Classification, and Recommendations in Spotify Music

INTRODUCTION

Music listening is one of the biggest universal experiences that technology has revolutionized. There has been a transition from personal music devices towards online streaming of music. This transition has been guided by music recommending systems based on genres and titles and events. Metadata has been widely used as an endorsement of the music. This shift has been most significantly noted in last fifteen years and has paved way for research in areas of music emotion recognition. This is based on valence which Spotify describes as "the musical positiveness conveyed by a track."

Problem Statement:

The goal of this project is to implement:

1. Multi-linear regression to accurately predict the valence value.
2. Various classification algorithms to find which model achieves the highest accuracy in classifying valence-based classes.
3. Music recommender system based on the elements of the music provided in the dataset.

METHODOLOGY

A. Dataset

The Spotify dataset used for this analysis consists of 32,833 tracks with 23 columns, each containing track specific details / measures. They are namely, track_id, track_name, track_artist, track_popularity, track_album_id, track_album_name, track_album_release_date, playlist_name, playlist_id, playlist_genre, playlist_subgenre, danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo and duration_ms. Although, the column names are self-explanatory, description of each column along with their datatypes and statistical information is mentioned in the Appendix (Refer Fig. 1, Fig. 3 & Fig. 4).

B. Data Pre-processing

- **Handling missing data**: Out of 32833 tracks we found a total of 15 tracks that consisted of missing data. As this number was only 0.045% of the entire data, these records were removed from the dataset.
- **Handling Categorical data:** The data consists of significant number of categorical columns/values and encoding them in a conventional one hot encoding way resulted in 80000+ columns count. This makes it a fat and short data where number of features(M) >> number of examples(N). To overcome this, I came up with a solution of creating BERT embeddings after combining all the necessary columns. This ensures that no important information is lost while avoiding the curse of dimensionality. BERT embeddings of size 768 are created by combining track_name, track_artist, album_name, playlist_name, playlist_genre and playlist_subgenre into text before being given to a pretrained model (Refer appendix Fig. 5).
- **Feature Scaling**: Since we are provided with notable columns of numerical values describing each song, it is mandatory to perform feature scaling and bring all of them on the same scale for generation of better results. The scaling technique used here is the Minimum-Maximum Scaling.

Apart from the above given steps, it was required to handle the 'track_album_release_date' column particularly and extract only the 'year' part from the respective dates. Also, there were duplicate values for the 'track_name' column that was to be taken care of using the duplicate removal technique. The dataset is sorted based on 'track_popularity' in descending order and then kept only the first occurrence of each unique 'track_name', effectively removing duplicates and keeping only the rows with the highest 'track_popularity' for each unique 'track_name'. Also, columns like

'track_id', 'track_album_id' and 'playlist_id' should be dropped as they have no significance in the prediction / classification of valence.

C. Exploratory Data Analysis

D. Multi-linear Regression (valence prediction)

The dataset consists of a 'valence' column that describes the musical positiveness conveyed by a track. Tracks with high valence sound more positive (happy, cheerful, euphoric), while tracks with low valence sound more negative (sad, depressed, angry). The value in this column is numeric and continuous which is based on several track metrics so this can be used as a problem of multi-linear regression. For the given task we need to encode the categorical columns, but since doing this by getting the dummies for each will result in a fat data. So, BERT embedding is constructed for each of them. Also, method of Backward Elimination is used for feature selection.

E. Valence Classification

Since the valence column consist of continuous values, to convert this into a 'binary-classification problem', modification in the column was required. Considering 0.5 as a threshold, values greater than 0.5 will be termed as positive emotion and less than 0.5 as negative. Post this step, feature transformation was applied. For this project, I have chosen the two linear dimensionality reduction techniques:

- Principal Component Analysis (PCA) (unsupervised approach)
- Linear Discriminant Analysis (LDA) (supervised approach)

The resulted features were then fed to various classification algorithms to get the accuracies and F1 score. K-fold cross was implemented to get an average idea of the above metrics.

F. Recommender System

Machine learning solves many problems but making product recommendations is a widely known application of machine learning. There are two mainly types of recommendation systems –

1. **Collaborative filtering**: The collaborative filtering method is based on gathering and analyzing data on user's behavior. This includes the user's online activities and predicting what they will like based on the similarity with other users.
2. **Content based filtering**: Content-based filtering methods are based on the description of a product and a profile of the user's preferred choices. In this recommendation system, products are described using keywords, and a user profile is built to express the kind of item this user likes.

Considering the usage of both the techniques, content-based filtering suits our dataset and it contains description of a song in terms of metrics like danceability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence and tempo. Content-based filtering can be applied on music using different combination of features. For example, we can recommend songs based on artist name, popularity, playlist presence, genre, energy etc. Every feature will add in different result in the recommendation. In this study I aim to use variations of features to see how the recommender system works and give recommendations.

RESULTS

NOTE: All the result visualizations and timeline breakdown are reported in a separate Appendix document.

A. Exploratory Data Analysis (EDA)

To begin with, we find the correlation of each feature with another (Refer appendix Fig.6). We find that the features 'loudness' and 'energy' had the highest correlation of 0.68 followed by the correlation between 'valence' and 'danceability' that comes around 0.33. The highest negatively correlated features were 'acousticness' and 'energy' with a correlation of -0.55. Seeing the produced correlation values, it can be said that all the features are different from one another and there is no

possibility of linear combination. The same can be seen in the produced pair plot for the given dataset (Refer appendix Fig.7).

Next, we visualize the number of track albums that were released over the years (1957-2020). Over the years, we see an increasing trend in the number of albums release (Refer appendix Fig. 8) with the maximum number of albums being released in 2019.

Next, we visualize the change in the values of track metrics namely 'danceability', 'energy', 'loudness', 'valence', 'tempo', 'duration_ms', 'instrumentalness', 'duration_ms', 'speechiness' and 'liveness' over the years (Refer appendix Fig. 9) and distribution of data (using distplot and KDE line) (Refer appendix Fig. 10). The features 'speechiness', 'liveness', 'loudness', 'danceability' and 'energy' have observed a positive increase of the values in the songs over the years, while 'acousticness' and 'tempo' saw a decrease of their usage in tracks over the years.

Next, we visualize the top 10 Songs with highest popularity and top 10 Artist with highest number of releases (Refer appendix Fig. 11) and (Refer appendix Fig.12). The most popular track is the 'Track Monkey' followed by 'ROXANNE'. The track artist with highest album releases is 'Martin Garrix', followed by 'Hardwell'.

B. Multi-Linear Regression (valence prediction)

Multi-linear regression was implemented in three different ways, firstly considering only the numerical music features given in the dataset. No explicit feature selection was done for this as it is expected that the LinearRegression() class will handle the feature selection internally. For this given setup, the R2 score, MSE and RMSE are consolidated in Table I. In the second setup, I gave a try on doing the feature selection explicitly, using the 'backward elimination' technique. I implemented it using the 'ordinary least squares (OLS)' class and keeping significance level for P-value as 5% (0.05). Executing the above given setup gave an OLS summary in which the P values for all the chosen features were way below the confidence level (Refer appendix Fig. 13). Therefore, no explicit feature selection was required. Third setup involved the usage of original features along with the BERT embeddings created for the categorical data. For this given setup, the results are reported in Table I. We see a significant performance boost because of the inclusion of BERT embeddings. It is critical to mention R2 score here as even though the MSE and RMSE for the given models are less, the R2 score is not that satisfactory.

Regression Technique	No Feature selection			BERT Embeddings		
	R2 Score	MSE	RMSE	R2 Score	MSE	RMSE
Multiple Linear Regression	0.1978	0.0449	0.2120	**0.3452**	**0.0367**	**0.1915**

Table I: Multi-Linear Regression results

C. Valence Classification

The valence classification task started with modifying the target variable to make this problem fit for binary classification. The target class consists of 2 labels, '1' depicting positive emotion and '0' depicting negative emotion. The target feature is balanced consisting of 12134 examples for class '1' and 11315 examples for class '0' (Refer appendix Fig. 14). The classification is implemented in 3 different ways where the first one involves all the numerical music features along with the BERT embeddings for categorical data without any feature transformation. These features are fed into numerous classification algorithms and the accuracy and F1 score obtained from each of them is mentioned in Table II. In this approach, XGBoost resulted in the highest Accuracy and F1 score of 0.70 and 0.71 respectively. Next, two feature transformation techniques are used to draw comparison with the Accuracy and F1 score received in the first approach. To start with, PCA can be implemented using 7 principal components (Refer Fig. 15), as the given features were able to explain 95.82% variance in the data. In the case of LDA, it can only have the number of components (<= min (n_classes - 1, n_features), i.e. only 1 in our case. So, LDA is implemented using just 1 component. The results for each of the applied transformation technique is reported in Table II. For this problem, even though with 1 principal component, LDA was able to produce a feature that was able to provide classification results like that of no feature transformation. Accuracies received from

PCA transformation were also similar. It is commendable to see the power of both these approaches to give similar accuracies with such less features count as compared to the original. For LDA, XGBoost classifier gave the highest accuracy of 0.66 and for PCA, SVM with Gaussian kernel resulted the highest with 0.66.

Classification Algorithms / Feature Transformation Techniques	No feature transformation		PCA		LDA	
	Acc.	F1	Acc.	F1	Acc.	F1
Logistic Regression	0.66	0.69	0.64	0.67	0.66	0.69
K-Nearest Neighbor	0.62	0.63	0.61	0.62	0.60	0.62
Support Vector (Linear)	0.66	0.69	0.64	0.68	0.66	0.69
Support Vector (Poly)	0.69	0.70	0.64	0.68	0.59	0.70
Support Vector (Gaussian)	0.68	0.70	0.66	0.68	0.66	0.68
Naïve Bayes	0.62	0.68	0.62	0.68	0.66	0.70
Decision Tree	0.61	0.62	0.59	0.60	0.58	0.59
Random Forest	0.65	0.68	0.63	0.67	0.66	0.68
XGBoost	**0.70**	**0.71**	0.65	0.67	0.66	0.67

Table II: Valence Classification results

D. Recommender System

The recommendation system has been implemented using two different approaches first, recommending songs based on ‘track’s artist’, ‘album name’, ‘playlist name’, ‘playlist genre’ and ‘playlist sub-genre’ and the second one, based on track’s numerical properties 'acousticness', 'danceability', 'energy', 'instrumentalness', 'key', 'liveness', 'loudness', 'mode', 'speechiness', 'tempo', ‘valence’. The two approaches are chosen to draw a comparison of recommendations provided by them. Both the above approaches use cosine similarity, in the first one on the vectors representing each song and in the second, on the numerical track properties. The results are provided in the Fig. 1 & 2 below. Future aspect of this experiment is to validate the provided results through some benchmarks or baseline, currently this was out-of-scope for this project. Secondly, providing track artist along with the track name for more efficient predictions as there are chances that two different artists produce a song with exact same name.

track_artist	Songs Similar to Circles	track_album_release_year	track_popularity
Saba	Photosynthesis	2016	70
Jalen Santoy	Foreplay	2016	66
Berhana	Grey Luh	2016	65
Phony Ppl	Why iii Love The Moon.	2015	64
Caleb Belkin	I Fall in Love Too Easliy	2017	62
J. Cole	Let Nas Down	2013	56
FKJ	Waiting	2014	54
Chuuwee	Nothin' At All (feat. Skoolie 300)	2014	54
RJD2	See You Leave	2013	52

Fig. 1: Recommendation through text vectors

track_artist	Songs Similar to Circles	track_album_release_year	track_popularity
Basta	Вселенная	2013	36
MESSIAH!	ANGST II	2019	41
Khalid	My Bad	2019	71
Lord ADL	A Braba	2019	47
Barney Artist	I'm Going to Tell You	2016	42
Peter Tsotsi Juma	Kajo Golo Weka	2013	8
Jaheim	Put That Woman First	2002	57
Method Man	Bring The Pain	1994	0
Christon Gray	Gray's Conclusion	2011	11
Sick Jacken	The Sickside	2009	30

Fig. 2: Recommendation through music parameters

CONCLUSION

In this work, we explore multi-linear regression and classification to study valence of different tracks found in Spotify dataset on Kaggle. We explore the concepts of feature transformation, feature scaling, cross validation and various supervised machine learning algorithms. We achieve 0.0367 MSE in valence regression and 70% accuracy for classification. Furthmore, we delve into various techniques of music recommendation systems. As of now, we are unable to validate music recommendation results, but future works can delve into this limitation.

REFERENCES

[1] https://www.kaggle.com/datasets/joebeachcapital/30000-spotify-songs

[2] https://community.spotify.com/t5/Spotify-for-Developers/Valence-as-a-measure-of-happiness/td-p/4385221
[3] https://medium.com/mlearning-ai/what-are-the-types-of-recommendation-systems-3487cbafa7c9
[4] https://towardsdatascience.com/part-iii-building-a-song-recommendation-system-with-spotify-cf76b52705e7
[5] https://www.eliftech.com/insights/all-you-need-to-know-about-a-music-recommendation-system-with-a-step-by-step-guide-to-creating-it/
[6] https://medium.com/@briansrebrenik/introduction-to-music-recommendation-and-machine-learning-310c4841b01d
[7] https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421
[8] https://medium.com/mlearning-ai/what-are-the-types-of-recommendation-systems-3487cbafa7c9
[9] https://georgepaskalev.medium.com/how-to-build-a-content-based-song-recommender-4346edbfa5cf
[10] https://www.sqlservercentral.com/articles/dimensionality-reduction-techniques-pca-kernel-pca-and-lda-using-python
[11] https://medium.com/mlearning-ai/short-python-code-for-backward-elimination-with-detailed-explanation-52894a9a7880
[12] https://medium.com/@24littledino/xgboost-classification-in-python-f29cc2c50a9b
[13] https://machinelearningmastery.com/develop-first-xgboost-model-python-scikit-learn/
[14] https://scikit-learn.org/stable/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html