Using Linear Regression To Predict Outcome Of A Blackbox Machine Learning Algorithm

Pritish Sharma
Dublin City University
pritish.sharma3@mail.dcu.ie
19210582

Saumitra Das
Dublin City University
saumitra.das2@mail.dcu.ie
19211286

Mitul Verma
Dublin City University
mitul.verma3@mail.dcu.ie
19210961

***Abstract*— The objective of this research is to demonstrate that the behavior of the complex machine learning implementations could be predicted using simpler machine learning algorithms. For the purpose of this research, we would be predicting the valence of a song generated using Spotify's algorithm. We applied multivariate linear regression on 15 features and iteratively improved the model after reviewing the results to generate our final model that predicted values generated by another machine learning algorithm with a good level of confidence.**

Keywords—Linear Regression, Music, Spotify

I. Introduction

Machine learning algorithms are present everywhere in the field of commerce. Search results on search engines, product recommendations on e-commerce websites, media discoverability on popular social media platforms all use machine learning algorithms. These commercial implementations of ML are a black box to users and to those whose businesses rely on such online platforms. These algorithms are being tuned continuously and tweaked to improve performance and yield better results. From a business perspective, developing insights into such algorithms is vital to stay competitive and make informed business decisions.

In this paper, we'd be using linear regression to predict the outcome of one such algorithm whose internal working is unknown to us. We'd be predicting Spotify's valence value of a song. The valence is defined as the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry). The value of valence is calculated using various features of the track, and we wanted to identify what features were primarily having an impact on valence calculation.

The primary purpose of the paper is to demonstrate that machine learning could be used to model the behaviour of other machine learning algorithms whose internal functioning is unknown to us.

II. Related Work

The inspiration for our work primarily came from the article where the author attempted to reverse engineer an offering of a commercial data analytics firm[1]. Author used the KMeans algorithm to identify significant clusters in the hourly power load data of 15000 buildings in the US to develop insights in power usage. Author was able to draw notable conclusions from his model which could help utilities companies to keep consumption as stable as possible. The book, Spotify Teardown: Inside the Black Box of Streaming Music, provides in great detail information on the business as well as the technical side of Spotify[2]. The book describes the importance of Spotify's technologies and user base for music artists and labels. Spotify plays a significant role in popularity of a song among the masses. Therefore, it is imperative for music creators to take into consideration Spotify's music discovery and playlist generating system. Machine learning has been used to predict, label and classify various properties of music such as genre, popularity and playlist using Spotify data and API[3][4][5].

III. Data Mining - Methodology and Techniques

In the following subsections, we would describe in detail and chronological order the techniques used to explore and examine our research question.

A. Methodology

To ensure the extensive investigation of the problem, assessment of the solutions and implementation of further enhancements, we conceived a methodology to which we adhered throughout this exercise. Our methodology is based on the KDD process. We started by researching music discovery and analysis algorithms and studying the basics of music itself, followed by the implementation of a linear regression model on pre-processed data. Upon the review of the first model, we further enhanced the model iteratively. After each review, we examined and processed the data to improve our solution. We implemented "evidence-backed actions" policy, i.e. all our actions and operations on data were backed by the evidence that emerged in the last review. This ensured the credibility and reproducibility of the research.

B. Dataset

The dataset, acquired using official Spotify APIs, was sourced from Kaggle[6]. The dataset had 232,724 records with 18 columns. The below are few of the columns from the dataset.

- acousticness: A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
- danceability: Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
- energy: Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity.
- loudness: The overall loudness of a track in decibels (dB).

- valence: The value we are trying to predict. A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

C. Preprocessing

The dataset had no missing values, but there were duplicate records that were removed in the preprocessing. The duplicate records were identified using *artist_name* and *track_name*, i.e. the records with the same artist name and track name. The number of records was reduced to 175,245.

For model consideration, *track_name* and *track_id* were not considered while predicting valence. The feature, *track_id*, is merely an identifier, while *track_name* could not be fitted in a linear regression model. However, future work could take the impact of the name of tracks on valence by using NLP techniques.

D. Assumptions

The observations (or records) in the dataset are independent, thus satisfying the requirement of independence for linear regression. Assumptions that we have made for our model:

- Linearity: There is existence of linear relationship between the *valence* and features of a track.
- Homoscedasticity: The variance in residual is same for any value of *valence*
- Normality: For any fixed value of any feature, *valence* is normally distributed.
- Independence: This assumption is satisfied in the data. Each observation is independent.

E. First Model And Evaluation

After having done the basic preprocessing by removing duplicates and two unrequired features, we implemented the first multivariate regression model on the data to predict the value of valence.

We intended to include every feature in our first model to develop an insight into how much of variance in valence could be described by all features before eliminating them. The first challenge was to accommodate categorical features in the data. The features that did not have a continuous numerical value are - *artist_name*, *genre*, *key*, *mode*, and *time-signature*. To overcome this problem, we created dummy variables(or One Hot Encoding) for each of the features mentioned except *artist_name*. This ballooned our list of features to 51. We then implemented the multivariate linear regression on the data and generated the below results.

TABLE I.

R-Squared	0.48
Total Features	51

Fig. 1. Result of First Model

In the above multivariate linear regression model, we incorporated all the features, including the categorical ones through dummy variables. The r-squared value of 0.48 indicates the existence of a linear relationship between the features and the valence. However, with the current set of features, the linear relationship is weak.

The primary issue with the model is how the categorical data was incorporated. Take genre; for instance, there are 26 unique values in the genre. Using one hot encoding on the genre, our features increased by 25. Similarly, by employing dummy variable technique on other features such as key, mode, and time-signature, the total features increased to 51. This introduced complexity to our model, and we suspected degraded performance. We concluded that the categorical features must be dealt with a better method to optimize the model. Furthermore, we excluded the *artist_name* as it couldn't be accommodated even by using one-hot encoding because of the large number of unique values. However, based on our preliminary analysis, we were confident that the artist's name does affect the valence of the song.

With all features included except one, there also existed strong multicollinearity in our model.

F. Better Approach Towards Categorical Data

The features with categorical data are - *artist_name*, *genre*, *key*, *mode*, and *time-signature*. These features cannot be eliminated for the sake of simplifying the model because even from an intuitive perspective, one could conclude that these features indeed have an effect on the valence (describing the musical positiveness conveyed by a track) of the song. For example, for the genre “children’s music”, we expect the valence to be higher. Similarly, key, mode and instrumentalness could not be excluded from the model without substantial evidence to do so.

Therefore, to reduce the complexity of the model, we devised a method wherein we gave a rank, to be used as a weight in our model, to each category present in the categorical features. The ranks were calculated by taking the average valence for each category. For instance, the average value of valence was highest for “Reggae” in the *genre*. Hence it was given the rank of 26. Similarly, the lowest value of average valence in the *key* feature was D# and therefore was given rank 1. Using this technique, we were able to include *artist_name* feature too in our model.

$$\text{Rank, R of a value} \propto \frac{\sum valence}{N} \quad (1)$$

where N = Total observations for that value

After assigning the ranks to each unique data value in the five attributes, we then scaled them using Max Absolute Scaler as the value of valence lies between 0 to 1. This brought the values between 0 to 1 without changing the sparsity of the data.

G. Second Iteration And Evaluation

After implementing the above method, we reduced the features to 15. We were also able to include *artist_name* in our model. We then applied the multivariate linear regression model to the 15 features and got the below results.

TABLE II.

R-Squared	0.63
Number of Features	15

Fig. 2. Result after second iteration

The model has improved R-Squared value, as well as we had a significant reduction in features. It gives credibility to our approach towards categorical data. However, there still existed the possibility of strong multicollinearity in our system.

H. Feature Selection

In the previous model, although we had improved performance, we still hadn't assessed the possibility of multicollinearity and had not examined the outliers. For feature selection, we first implemented the Pearson correlation matrix to identify which features had the least impact on the valence. Once such features were identified, we examined the data in those features by plotting the gaussian distribution and boxplots. The features that had a large number of outliers, an extremely skewed distribution and a negligible impact on valence were flagged for further review. Below are the examples of some of the plots of features that were flagged.

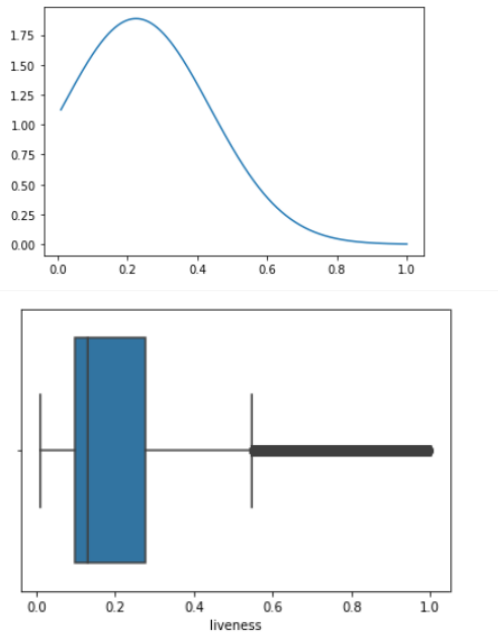

Fig. 3. Distribution and boxplot of *liveness* feature showing presence of large number of outliers

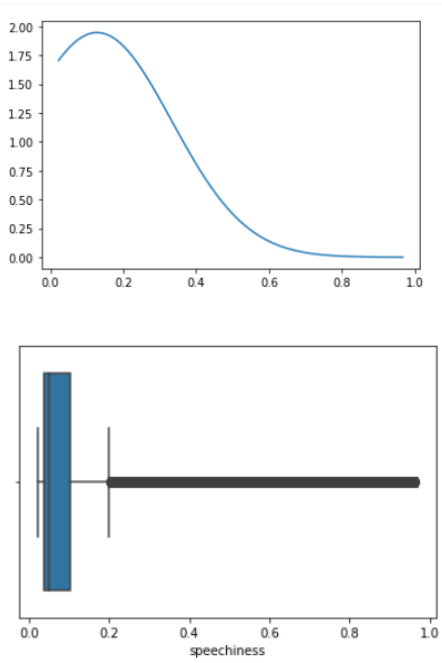

Fig. 4. Distribution and boxplot of *speechiness* feature showing presence of large number of outliers

Additionally, we also calculated VIF for each of the features to determine the level of multicollinearity.

TABLE III.

Feature	VIF
popularity	6.32
acousticness	5.44
danceability	17.40
duration_ms	4.28
energy	18.49
instrumentalness	2.01
liveness	3.37
loudness	10.39
speechiness	2.63
tempo	14.37
genre_rank	8.60
mode_rank	12.22
key_rank	4.94
time_key_rank	23.66
artist_rank	20.17
valence	10.03

Fig. 5. VIF values of all the features

Once we had determined the correlated features, we applied bidirectional elimination on the remaining features and implemented numerous combinations of features for the final iteration. Below are the VIF values for the final selected features.

TABLE IV.

Feature	VIF
danceability	1.726672
energy	2.378563
genre_rank	2.122755
artist_rank	3.389094
artist_rank	2.563769
valence	2.304496

Fig. 6. VIF values of final selected features

I. Third Iteration and Evaluation

The model improved significantly after the feature selection procedure. Below is the result of multivariate linear regression applied after feature selection.

TABLE V.

Dep. Variable:	valence	**R-squared:**	0.61
Model:	OLS	**Adj. R-squared:**	0.61
Method:	Least Squares	**F-statistic:**	5.48E+04
Date:	Thu, 23 Apr 2020	**Prob (F-statistic):**	0
Time:	00:26:59	**Log-Likelihood:**	64460
No. Observations:	175245	**AIC:**	-1.29E+05
Df Residuals:	175239	**BIC:**	-1.29E+05
Df Model:	5		
SSE	5.48	**Mean Absolute Error**	0.14
Mean Squared Error	0.034	**Root Mean Squared Error**	0.19

Fig. 7. Results of final iteration

Furthermore, the model also satisfied the condition of normal distribution of residuals.

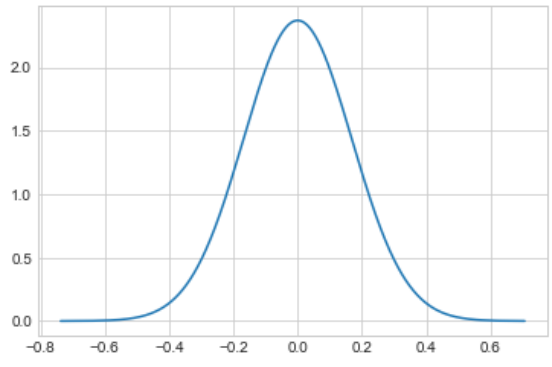

Fig. 8. *Distribution plot of residuals*

We also observed homoscedasticity in the residuals but with bias[7]. Missing features could explain the bias. It is possible that Spotify may have used additional features which we are unaware of.

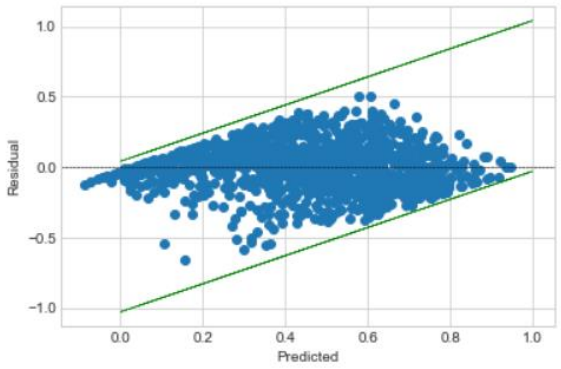

Fig.9. Homoscedasticity with bias in Residuals

The value of r-squared shows that there is linear relationship between valence and features. We also verified the normal distribution of *valence* against the final selected features. Our model satisfied all assumptions required for the linear regression.

IV. Conclusion and Future Work

We started by first implementing the model on every feature and iteratively optimized it by reviewing data after each iteration. The iterate-review-improve model in combination with "evidence-backed actions" policy, i.e. all our actions and operations on data were backed by the evidence emerged in each iteration, allowed us to predict values generated by another machine learning algorithm with a good level of confidence.

This project work is a proof of concept that the outcome of commercial implementation of machine learning algorithms could be predicted by using machine learning itself. Spotify's algorithm was a blackbox for us, but we were able to identify the impact of various attributes of a track on the outcome (valence) by creating a simple yet powerful multivariate linear regression model.

However, there are potential problems when using machine learning algorithms to predict outcome of other algorithms. in our case, we had the data and knew the varaibles that impacted the value we trying to predict. But in most cases , where we want to predict outcomes of more complicated products such as search engines, we wont really have the data.

It would be incredibly hard to scrap data and identify the variables or the combination of variables that affect outcome.

Furthermore, one slight change to the algorithms we are trying to mimic, could break our model completely and may again require work from the ground up.

To overcome these problems, we could use ML itself to scrap and identify the data intelligently and use domain expertise to make better models.

By using Neural networks and deep learning, as well as traditional supervised and unsupervised techniques, the models predicting the outcomes of other blackbox models could be optimized further.

We strongly believe the reverse-engineering or unravelling commercial implementation of machine learning algorithms would be a massive field in future, helping organizations, governments and individuals to compete and expand their businesses.

REFERENCES

[1] G. Mauro, "I reverse-engineered a $500M Artificial Intelligence company in one week. Here's the full story.", *Medium*, 2020. [Online]. Available: https://medium.com/startup-grind/i-reverse-engineered-a-500m-artificial-intelligence-company-in-one-week-heres-the-full-story-d067cef99e1c. [Accessed: 24- Apr- 2020].

[2] M. Eriksson, R. Fleischer and A. Johansson, *Spotify Teardown*. .

[3] K. Luo, "Machine Learning Approach for Genre Prediction on Spotify Top Ranking Songs", 2020. Available: https://cdr.lib.unc.edu/concern/masters_papers/ns064961b. [Accessed 24 April 2020].

[4] R. Nijkamp, "Prediction of product success: explainingsong popularity by audio features from Spotify data", 2020. Available: http://essay.utwente.nl/75422/1/NIJKAMP_BA_IBA.pdf. [Accessed 24 April 2020].

[5] E. Georgieva, M. Suta and N. Burton, "HITPREDICT: PREDICTING HIT SONGS USING SPOTIFY DATA", 2020. Available: https://pdfs.semanticscholar.org/5f40/e603ac969eb476f1af21b1efdd70153c24c0.pdf. [Accessed 24 April 2020].

[6] Z. Hamidani, "Spotify Tracks DB", *Kaggle.com*, 2020. [Online]. Available: https://www.kaggle.com/zaheenhamidani/ultimate-spotify-tracks-db. [Accessed: 24- Apr- 2020].

[7] "Linear Regression", *Condor.depaul.edu*, 2020. [Online]. Available: https://condor.depaul.edu/sjost/it223/documents/regress.htm. [Accessed: 24- Apr- 2020].