PREDICTION OF MUSIC POPULARITY USING HIERARCHICAL REGRESSION

C. A. Rasmussen 144466, F. Z. Lehmann 154109, C. Foss s154312

Technical University of Denmark
DTU Management

(Link to GitHub)

ABSTRACT

This paper presents a comparative analysis between a baseline linear regression model and a hierarchical regression model, for predicting the *popularity* of music tracks from Spotify, using the Pystan framework. The Spotipy package was used to extract audio features of Spotify tracks from different genres, including ”pop”, ”metal”, ”classical”, and ”rap”. The hierarchical regression model allowed grouping the data by levels (genres), and assigning parameters accordingly. The implementation of this hierarchy yielded a better model performance in terms of accuracy compared to the baseline linear regression model.

***Index Terms*—** Spotify, Music, Python, STAN, Machine Learning, Hierarchical regression, Popularity, Genre, Audio features, Distributions, Performance

1. INTRODUCTION

The data has been manually extracted using the Spotipy Python library [1], inspired by the Kaggle data set for the science project ”Top Spotify Tracks of 2017” [2]. The data set consists of random tracks from four different genres. Each song has a set of audio features [3], and an associated popularity value in the interval from 0 to 100, where 0 is considered *not* popular and 100 is *very* popular. More specifically, the data set contains the following:

- 13,200 rows corresponding to the number of songs
- One feature representing the genre of the track, where the genre is $I \in \{1, 2, 3, 4\}$
- One feature representing the popularity of the track, $\mathbb{N} \in [0; 100]$
- 13 audio feature variables (see Jupyter Notebook): *danceability*, *energy*, *tempo*, *key*, etc.

This paper seeks to investigate if the genre and the audio features can be used to estimate the popularity of a single track. Furthermore, it is of interest to analyse if the estimation of popularity can be improved by proposing a hierarchical regression model that can differentiate between the music genres. The frequency of the four music genres in the data set can be seen in Figure 1 below:

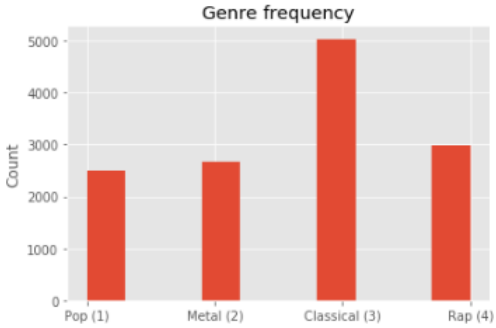

Fig. 1: *Distribution of music genres in the data set*

2. MOTIVATION

Data science and machine learning has been transforming many industries lately, and the music industry is no exception. To further address the importance of data science within the music industry, the company behind Spotify organized what is known as the ”RecSys Challenge 2018” [4]. In this challenge, participants were given a data set consisting of one million user-created playlists, and the objective is to make an automatic playlist continuation where, given playlist features, participants would generate recommended tracks to a given playlist.

However, it could be of interest to the music industry to see which audio features or genres that contribute the most to the *popularity*, as popular tracks tend to generate more revenue, and vice versa for unpopular songs. As previously mentioned, all tracks in this data set each belong to one of four genres, and thus naturally, their audio features are not independent. For that reason, *hierarchical regression* seems appropriate for dealing with the research question in this paper, that is:

Research question

How can the popularity of a Spotify track be predicted with hierarchical regression using the observed audio features, with genre as the level, and how does this compare with a simpler baseline model?

3. METHOD

The approach to the research question is described in this section. First of all, the distributions in the data sets were investigated to check if there were any immediate relations. This was not entirely clear from the data set alone. However, the group was able to make some preliminary assumptions based on the descriptions of the audio features from the Spotipy documentation, which will become evident later in the results section.

Since it is of interest to compare a Hierarchical Regression Model (HRM) to a baseline Linear Regression model (LRM), both have to be investigated individually. Thus, for each of the models, a Probabilistic Graphical Model (PGM) is constructed with an associated joint distribution expression. From that, two generative stories are presented, and these are then implemented in STAN from a generative story. Hence, this is carried out as seen below (for both models):

$$PGM \rightarrow Joint\ distribution \rightarrow Generative\ story$$

The models are trained using 80 % of the data set and the remaining 20% of the data set is used for testing. The two models are then compared in terms of the mean absolute error (MAE) and root mean squared error (RMSE), respectively.

4. RESULTS

Below is given the PGM, joint distribution and generative story for both the LRM and HRM, respectively. The two models are build upon a regression framework. In both cases, a multiple linear relationship is assumed between the audio features and popularity, because two or more explanatory variables have a linear relationship with the target variable, popularity.

The hierarchical regression model [5] is a modification, where the data set is grouped into distinct music genres and hence, each genre gets its own set of parameters. Notice also the hierarchical prior in the joint distribution for the HRM. Notice in the joint distributions, for convenience, that *all* audio feature variables for track n are denoted $Audiofeatures_n$. The PGM of the LRM and HRM are seen below in figures 2 and 3 with their corresponding joint distribution and generative story.

4.1. Linear Regression Model

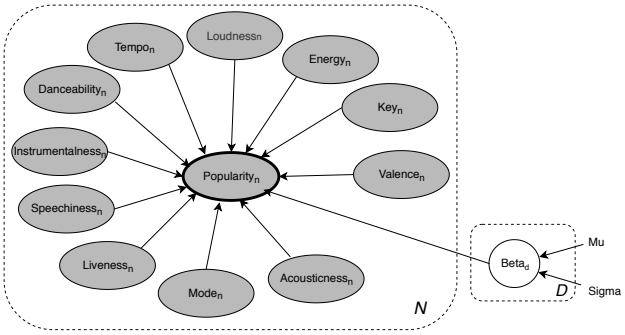

Fig. 2: *PGM of LRM*

$$p(\textbf{popularity}, \beta_d, ..., \beta_D, \mu, \sigma | \textbf{Audio features}) = \prod_{d=1}^{D} p(\beta_d | \mu, \sigma) \cdot \prod_{n=1}^{N} p(Popularity_n | Audiofeatures_n, \beta_d, ..., \beta_D)$$

Generative story: LRM

1. Draw an intercept parameter $Intercept \sim N(50, 100)$
2. Draw a variance parameter $\sigma \sim Cauchy(50, 100)$
3. For each of the audio features $D \in \{1, ..., D\}$
   (a) Draw a beta parameter $\beta_d \sim N(50, 100)$
4. For each of the Spotify tracks $N \in \{1, ..., N\}$
   (a) Draw target popularity $y_n \sim N(Intercept + \beta_d[l_n]' \cdot X'_n, \sigma)$

4.2. Hierarchical Regression model

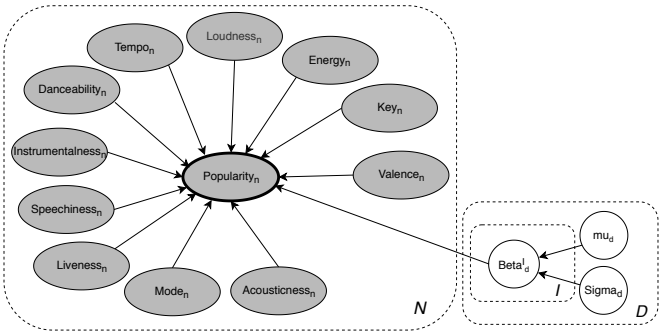

Fig. 3: *PGM of HRM*

$$p(\textbf{popularity}, \beta_d^i, ..., \beta_D^I, \mu_d, ..., \mu_D, \sigma_d, ..., \sigma_D | \textbf{Audio features}) = \Big(\prod_{d=1}^{D} p(\mu_d) \cdot p(\sigma_d) \cdot \prod_{i=1}^{I} p(\beta_d^i | \mu_d, \sigma_d) \Big) \cdot \prod_{n=1}^{N} p(Popularity_n | Audiofeatures_n, \beta_d^i, ..., \beta_D^I)$$

Generative story: HRM

1. For each of the audio features $d \in \{1, ..., D\}$
   (a) Draw a global mean parameter $\mu_{prior,d} \sim N(50, 100)$
   (b) Draw a global variance parameter $\sigma_{prior,d} \sim Cauchy(50, 100)$
   (c) For each of the four music genres $i \in \{1, ..., I\}$
      i. Draw a beta parameter $\beta_{d,i} \sim N(\mu_{prior,d}, \sigma_{prior,d})$
      ii. Draw a variance parameter $\sigma_i \sim Cauchy(50, 100)$
      iii. Draw a intercept parameter $Intercept_i \sim N(50, 100)$
2. For each of the Spotify tracks $N \in \{1, ..., N\}$
   (a) Draw the target popularity $y_n \sim N(Intercept_i + \beta_{d,i}[:, l_n]' \cdot X'_n, \sigma_i[l_n])$

The results from the two models are seen in Table 1 in terms mean absolute error (MAE) and root mean squared error (RMSE).

Table 1: *Accuracy measures for model comparison*

	MAE	**RMSE**
LRM	19.918	23.703
HRM	12.169	15.490

Furthermore, a visualization of the distribution of the true popularity values (blue) compared with the predictions (red) can be seen in Figure 4 and 5.

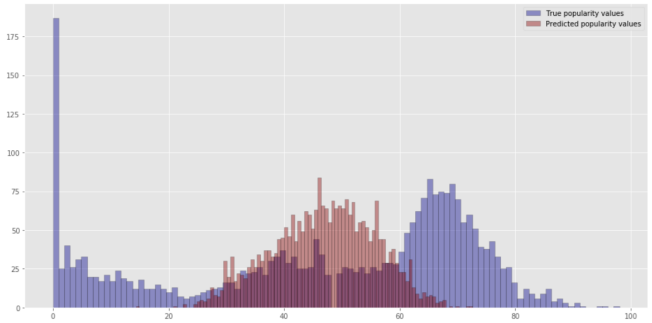

Fig. 4: *Comparing Linear Regression Model with true values.*

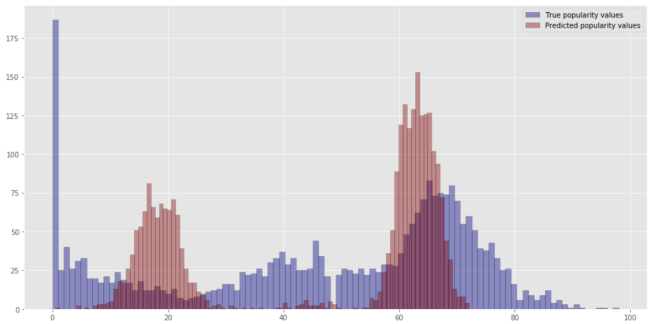

Fig. 5: *Comparing Hierarchical Regression Model with true values.*

The Linear Regression Model has a Gaussian distribution which is clustered at the middle of popularity values. The hierarchical model's predictions are clustered at both the lower and higher end of popularity values.[**?**]

5. DISCUSSION

First of all, the structure of the PGMs should be addressed. The first PGM, regarding the simple linear model, is based on a common structure for such a model. Regarding the hierarchical model, on the other hand, required a greater understanding of what should be levelled upon - genre or the audio features. Since it is a matter of perspective whether it is the genre which determines the audio features for a track, or vice versa, this paper chose to look upon it as audio features are defining the genres.

Secondly, when computing the model in STAN for both of the models, a Gaussian distribution was chosen to predict the popularity. Using this distribution is just one approach to perform a hierarchical regression, other approaches might be more appropriate. Since the Popularity in a sense is a discrete variable taking on values in the interval $\{1, 2, , .., 58, .., 100\}$, a Hierarchical Logistic Regression would have been better suited, since it uses a categorical distribution, but due to implementation issues, only the models using hierarchical regression is presented in this paper.

Thirdly, the two-three clusters in the popularity distribution from the data should also be addressed. The data was extracted supposedly at random, but it is hard to say how the Spotipy package actually does this, or if there simply exist very few songs with a popularity of 30-60. However, the latter explanation is highly unlikely. Moreover, considering the accuracy measures chosen for model comparison, the MAE is a metric for the average absolute error in popularity prediction, while the RMSE is essentially providing the same information as MAE however, it penalizes the model more in terms of larger errors. As Seen in Figure 4, the baseline LRM mainly captures the middle popularity values although, the true distribution of popularity values tends to be rather different in this case. On the other hand, the HRM's prediction distribution in Figure 5 is more dense in the two peaks. As a result, this better captures the true distribution of popularity values. Moreover, to optimize the model accuracy of the HRM, the two peaks should be spread out more evenly. Perhaps assumptions about the hyper-parameters could be adjusted differently in order to obtain a better performance.

Lastly, as seen in Figure 4 and 5, there is a relatively large amount of observations having a popularity value of "0". These are mainly observed from the classic genre, as the popularity value from Spotipy is based on how many current playbacks it has, and the release date of the track. It was decided not to exclude these observations, so the genre was preserved in the analysis.

6. CONCLUSION

In conclusion, using a simple Linear Regression Model (LRM) to predict the popularity of a track resulted in a lower performance compared to using a Hierarchical Regression Model. By simply adding a genre level and estimating the hyper priors for each audio feature, the accuracy is increased, seeing as the RMSE was reduced from 23.703 to 15.490. The reason for this is mainly due to the hierarchical model being able to capture more of the true distribution, despite only having two clusters, whereas the Gaussian distribution model was between them. Although the performance was increased by using a Hierarchical Regression Model, a Gaussian distribution was only assumed about the popularity distribution.

7. FUTURE WORKS

The models presented in this article are using a simpler likelihood function to predict popularity for each track, compared to how the data for Popularity may actually be distributed. For future works, a Beta-distribution can be investigated. For example, two shape parameters for explaining the distribution in popularity values can be a way of modelling the problem, by parameterizing the the linear function of $\mathbf{X} \cdot \beta$, and this may lead to a better model accuracy [6] [7]. With that being said, further exploration of different distributions and hyper-priors can be of interest to improve model accuracy further.

8. REFERENCES

[1] Spotipy, "Spotipy documentation," https://spotipy.readthedocs.io/en/latest/, API, data extraction.

[2] Nadine Taner, "Top spotify tracks of 2017," https://www.kaggle.com/nadintamer/top-tracks-of-2017/version/1, 2017, Kaggle.

[3] Spotify, "Audio features," https://developer.spotify.com/documentation/web-api/reference/tracks/get-several-audio-features/, features, data.

[4] Spotify, "Recsys challenge 2018," https://recsys-challenge.spotify.com/, 2018, Challenge, Data Science.

[5] Filipe Rodrigues and Francisco Pereira, "Classification models," Classification models from 42186 model-based machine learning, F19, DTU.

[6] Michael Clark, "Github - rstanbetaregression.r," https://github.com/m-clark/Miscellaneous-R-Code/blob/master/ModelFitting/Bayesian/rstanBetaRegression.R, 2016.

[7] Dalton Hance, "Beta regression in stan," https://github.com/daltonhance/stan_beta_reg, 2016.

9. CONTRIBUTION STATEMENT

The contributions to the report can be seen in Table 2.

Table 2: *Contributions of each of the group members*

Name	StudentID	Contribution
Christian Rasmussen	s144466	Abstract, 2, 4, 5
Frederikke Lehmann	s154109	5, 6, 7
Clara Foss	s154312	Abstract, 1, 2, 3, 6, 9