CS 181: Practical 3 – Recommending Music

Group 22: Treus Ex Machina (Anthony Soroka, Rohan Thavarajah, Jonne Saleva)

1. Introduction

The purpose of this practical was to identify and detect common patterns in how people listen to music, and predict the number of times a given user would listen to a given artist. These preferences were to be modeled using data from a streaming music service, detailing who listened to what and how many times, as well as user metadata, such as age and country (See Figure 1) In addition, each artist had a `MusicBrainz` identifier, using which metadata could also be leveraged as input features.

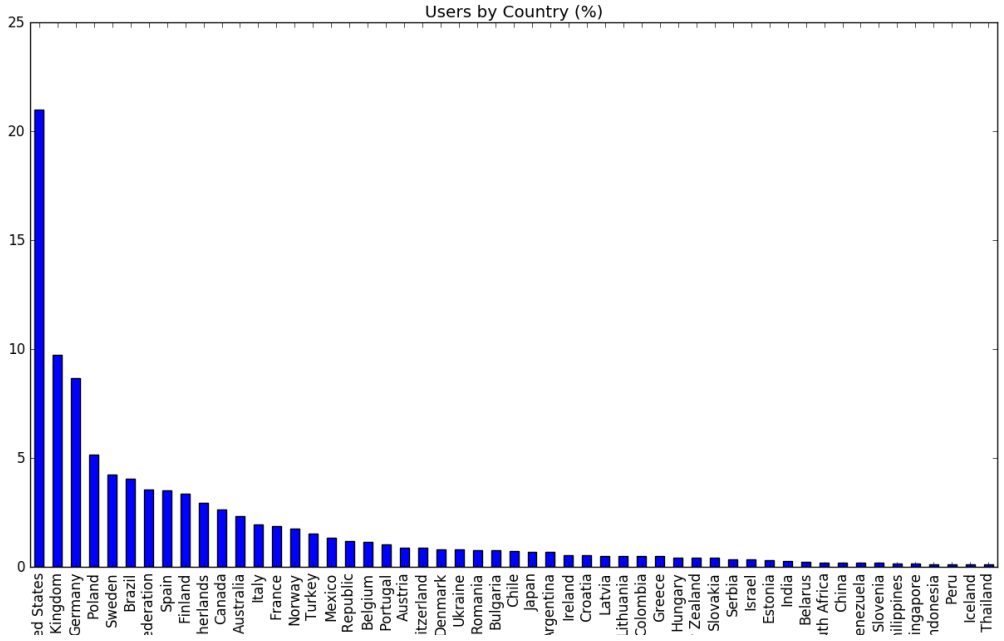

Figure 1: Percentage of users per country. Only countries with a share greater than 0.1% are shown.

The final prediction task was evaluated using MAE, or Mean Absolute Error. MAE is defined as

$$\text{MAE} = ||\hat{\mathbf{y}} - \mathbf{y}||_1 = \sum_i |\hat{y}_i - y_i|$$

After trying various models, we obtained a final MAE of 136.25039 on the Kaggle leaderboard.

2. Technical Approach

Model 1: GLM: Poisson Regression

Our first approach is a parametric generalized linear model. Plays of artist j by user i is a count variable measured over some exposure time so is particularly well suited to being modelled as a Poisson GLM. To this end, we model $plays_{ij}$ as a Poisson distributed random variable, parameterized by the linear predictor η_{ij}.

$$plays_{ij} \sim \text{Pois}\left(\exp(\eta_{ij})\right)$$

and

$$\eta_{ij} = \alpha_0 + \beta_0 user_0 + \cdots + \beta_{N-1} user_{N-1} + \gamma_0 artist_0 + \cdots + \gamma_{M-1} artist_{M-1}$$

where $user_0 = 1$ if $i = 0$ and is 0 otherwise and $artist_0$ is defined analagously. However we have 2000 artists and 233,286 users. Constructing this matrix of predictors is prohibitvely memory hungry. We shrink the space of predictors as follows. First we condense user info by modifying the response variable:

$$playsDemedianed_{ij} = plays_{ij} - userMedian_i$$

In addition we include two new variables, user age and gender. These variables have missing values. Gender is categorical and we adjust for missing values by including a "missing" category. Age is continuous and we adjust for missing values by imputing the mean and including the variable flag_age_miss which = 1 if age is missing.

We shrink the space of artists by first plotting the distribution of artist medians:

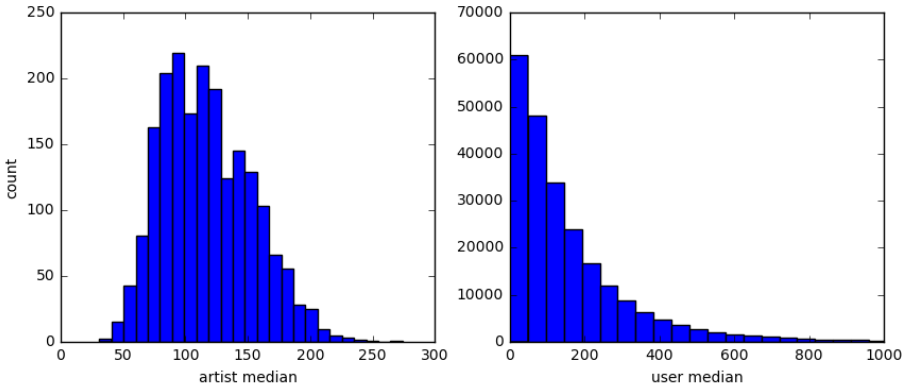

Figure 2: Histograms of median plays per artist and per user

We generate a new categorical variable which labels artists as "very low" to "very highly" addictive according to their median plays. In summary, we predict plays as the user median + some perturbation for artist and user characteristics. Note that because our response variable can now take negative values, we switch from Poisson GLM to the random forest regressor which has the added benefit of considering higher order interactions between our predictors.

However, we find our training MAE is 158 much worse than the baseline usermedian. We notice via visual inspection on our own validation set, that some of our worst errors are because we predict play counts way above or way below plays encountered in training for that user. We thus apply the following post-processing step:

1. We categorize predicted plays by quartile for each user.
2. For user-artist pairs whose predicted plays are between the 25th and 75th percentile, we set their predicted plays equal to the user's median. For these observations our predictions are identical to the usermedian baseline.
3. For those user-artist pairs with predicted plays below the 25th percentile however, we argue that we have identified an artist the user is likely to dislike. We predict plays are the lower quartile of that user's plays.
4. Similarly when predicted plays are above the 75th percentile we use the upper quartile.

Our reasoning is that although the values of our predictions are off in magnitude, they are correct ordinally. We get an in-training MAE of 130 (a vast improvement over the naked predictions). When applied to the test data however, this model disappointingly fails to beat the baseline.

Model 2: Modified GLM

This motivates model 2. We modify the response variable again this time generating a categorical variable that houses quartiles by user. Specifically:

$$playsCategorical_{ij} = \begin{cases} 0 & plays_{ij} \quad \text{is below the 25th percentile for user i} \\ 1 & plays_{ij} \quad \text{is between 25th and 75th percentile for user i} \\ 2 & plays_{ij} \quad \text{is above the 75th percentile for user i} \end{cases}$$

We apply multinomial logistic regression and the random forest classifier to predict the group for user-artist pairs. In post-processing we then apply the same mapping as described in model 1 (if predicted plays are very low, we predict the lower quartile, when they are very high we predict the upper quartile, and when they are somewhere in between, we agree with the baseline and predict the user median). The random forest classifier nets us an impressive in-training MAE of 90 but still fails to beat the baseline on kaggle.

Model 3: Clever re-weighted averages

For the third model, we abandanoned parametric approaches, and instead chose to focus on predicting some salient combination of user- and artist-specific means/medians as the play count for for a given user-artist pair. After lots of trial and error, a particular approach which, while seemingly somewhat arbitrary, delivered good results in practice was defined as follows

$$w \times \text{userMedian} + (1 - w) \times \text{userMedian} \times \text{ArtistFactor}$$

where

$$\text{ArtistFactor}(a) = \frac{1}{N} \sum_{u=1}^{N} \frac{\text{numberOfPlays}_{u,a}}{\text{userMean}_u}$$

We realize that the scaling is somewhat odd; we multiply by the median yet divide by the mean! However, in a competition with a leaderboard, what ultimately matters is the final score, so that is what we prioritized instead of trying to justify the choice to the last minutia.

To tune the hyperparameter w, we evaluate the MAE on various configurations of w. See Figure 3. We find the optimal weight to be $w = 0.25$. The red line in the figure indicates the user median baseline on Kaggle.

This model turns out to perform staggeringly well compared to other models we used, delivering an impressive leaderboard MAE of approximately 136.

Other models used

We also tried various matrix factorization approaches, such as non-negative matrix factorization and singular-value decomposition, but did not observe any gains in performance.

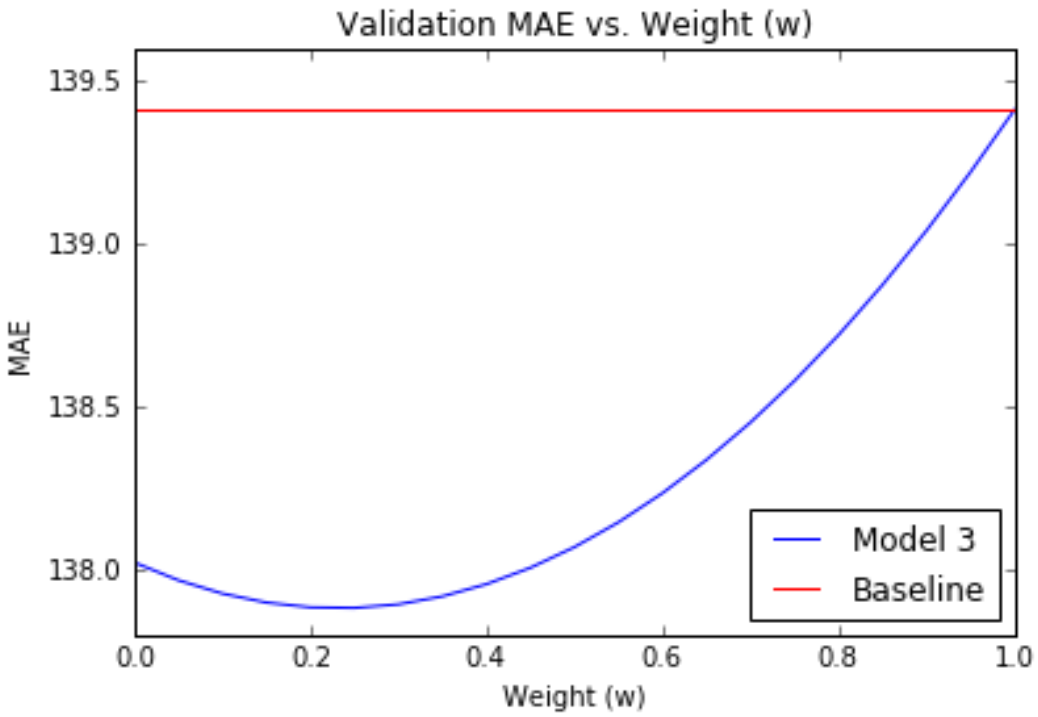

Figure 3: Model 3 Validation MAE as a function of its Weight (w)

3. Results and Discussion

Overall, we were surprised by the staggeringly good performance of the weighted means/medians model. In particular we would have thought that a matrix approach, or alternatively an established algorithm from `surprise` would have performed at least comparably to the mean approach, but instead the final result turned out rather different. See the final (rounded) performance metrics in the table below.

Model	Performance (MAE)
NNMF et al.	$>$ 250 in-training. Didn't try on Kaggle
Random Forest Regressor (based on GLM)	158 (in-training), bad on Kaggle
Modified RF Regressor [like/ambivalent/dislike]	130 (in-training), bad on Kaggle
Multinomial RF Classifier	90 (in-training), bad on Kaggle
Weighted Averages Approach	136 on Kaggle

With more time, we would like to explore ensemble methods. Specifically, given Weighted Average approach worked best, we would combine the studied methodology's (Random Forest Regressor, Multinomial RF Classifier etc.) to forecast predict plays. Similar to how we located the optimal weight (w), we might perform a grid search on this ensemble method to find the optimal combination of weights.

Lastly, we have a few higher-level takeaways from this assignment. First, we appreciate the value of forming simple initial baselines when doing machine learning / data science experiments. Had we stuck down the path of some of these state of the art approaches, we likely would not have beat the simple User Median approach. Additionally, we recognize the challenges of improving upon User-Item recommendation systems. In the famous Netflix Challenge, only after 3 years of research did teams manage to reduce RMSE by .10 (star ratings) from a Weighted Average approach. The fact that teams in this competition were only able to reduce MAE by approximately 1.5 plays versus the simple User Median approach again illustrates the innate challenges to these sort of problems.