

# models\_2\_parsed

December 19, 2020

## 1 Track orders - regression approach

### 1.1 Structure

Predictors:

- Valence of track  $i$  (continuous)
- Energy of track  $i$  (continuous)
- Loudness of track  $i$  (continuous)
- Tempo of track  $i$  (continuous)
- Number of track  $i$  (from 1 to 16)
- Position  $i$  (sections from 1 to 3)
- Interactions between valence, energy, loudness and tempo

Dependent variable:

- Valence/energy/loudness/tempo of track  $i + 1$  (continuous)

### 1.2 Models

I attempted to predict the value of track in position  $i + 1$  given the features of track in position  $i$ . In this first attempt, I used 1) splines regression; 2) random forrest and 3) LSTM. All models were trained in 75% of the albums, and tested on both train and test splits.

### 1.3 Model diagnostics

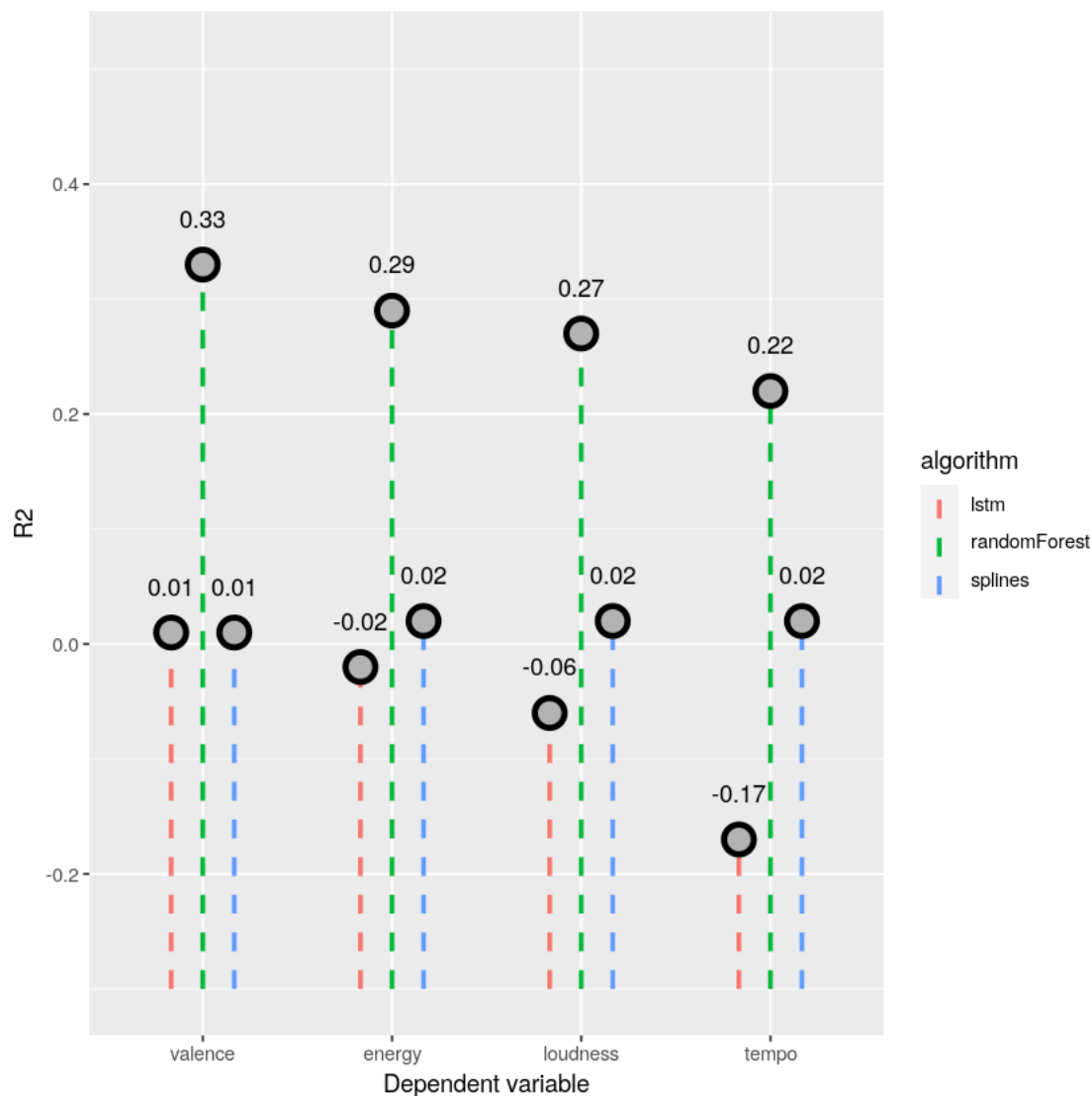
Detailed diagnostics and parameters of each model are omitted for brevity. Here, for each regression attempt, I report: 1)  $R^2$  (from training dataset) and RMSE (from test dataset). I also provide some visualizations of true vs predicted data (by album).

#### 1.3.1 $R^2$

The Random Forest algorithm is, by far, the best fit to our data, with 20 to 30 percent of variance explained. This is consistent with the previous (classification) model, which was able to guess, with ~32% accuracy, if a track pertained to the 1st, 2nd, 3d or 4th section of the album.

Splines regression was very close to baseline (3 knots and 4 degree polynomials). LSTM performed badly, which might be due to poor choices of parameters -on my end. The negative  $R^2$  values

indicate that our model fit actually increases error in comparison with the zero rule (predicting the mean).

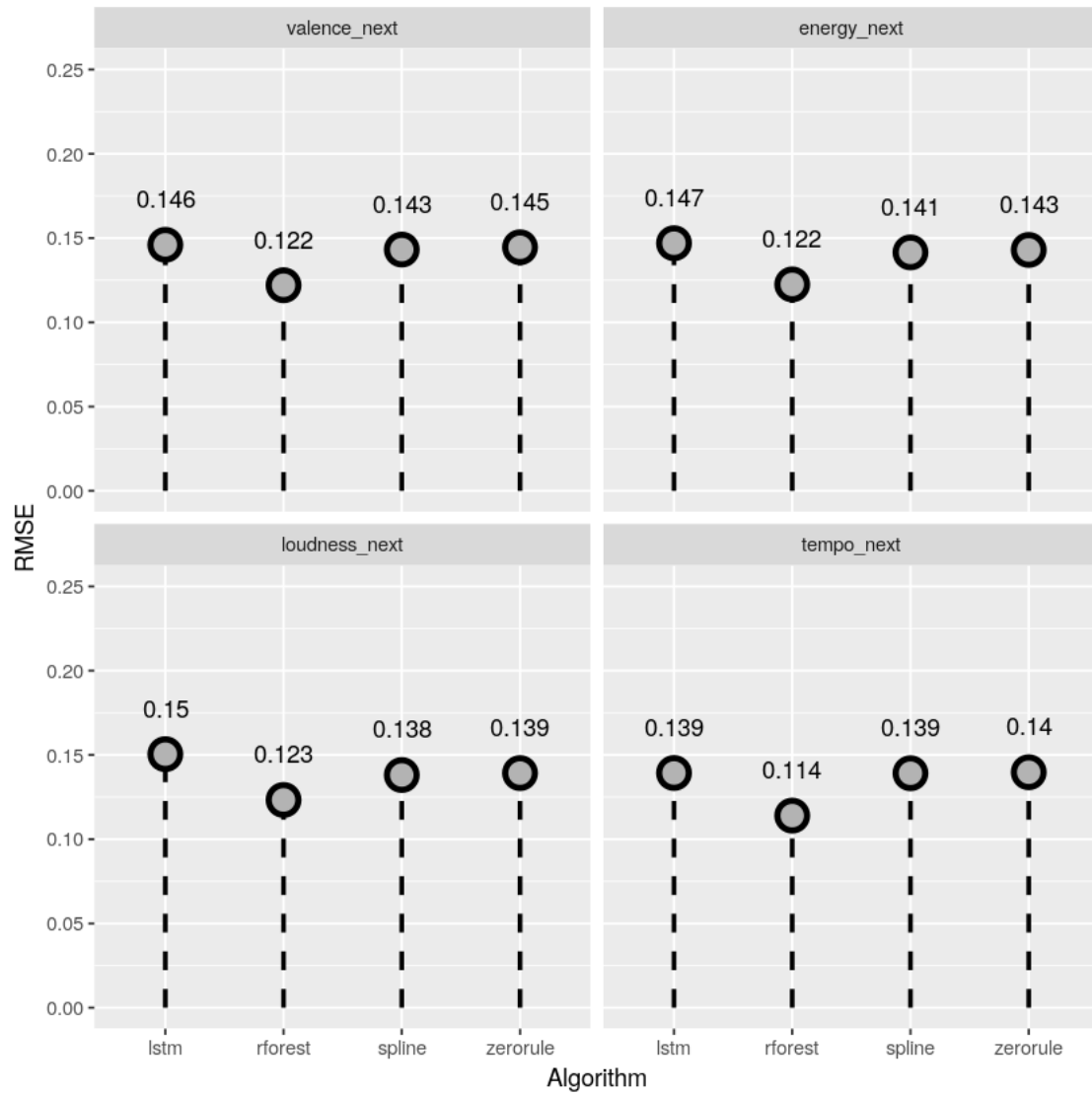


### 1.3.2 RMSE

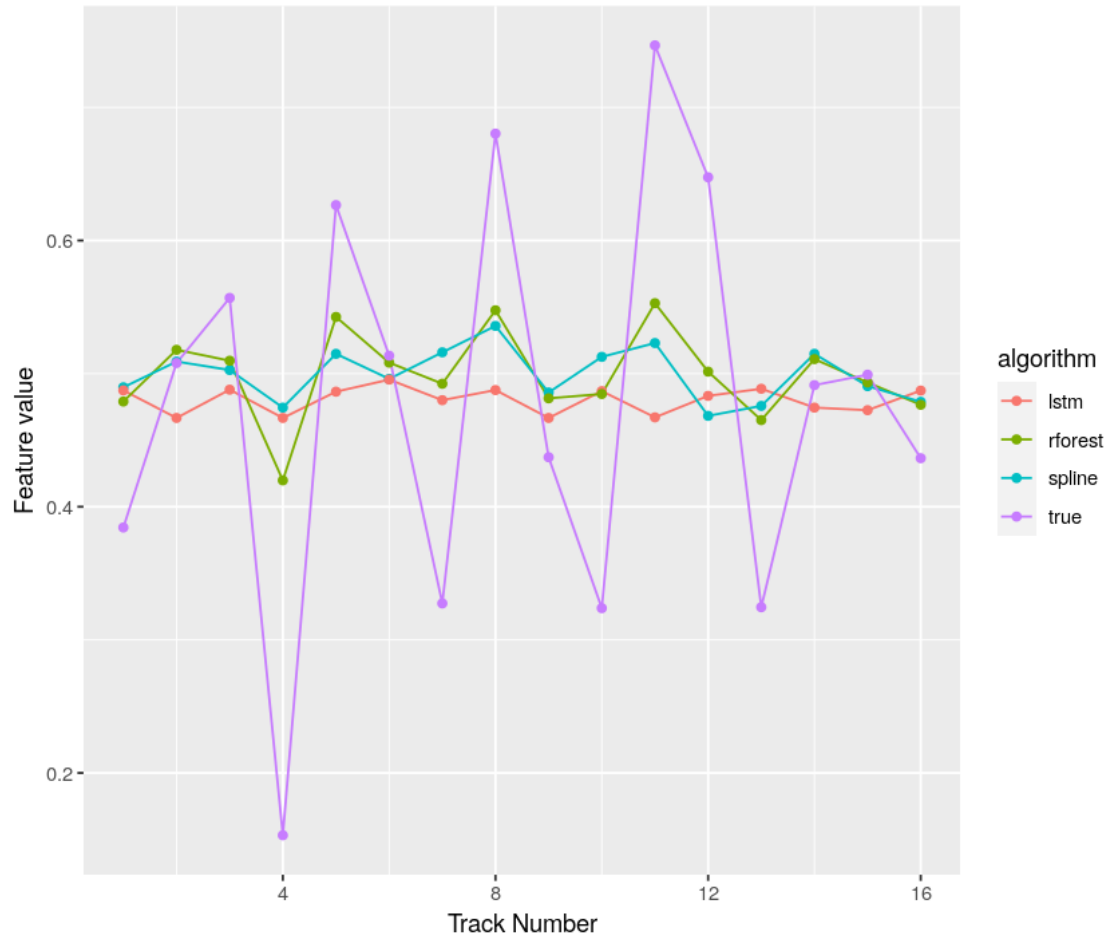
Here I provide a baseline RMSE by predicting the feature in position  $i + 1$  as the mean of that feature's distribution.

This baseline shows that, with the exception of the Random Forest algorithm, all models perform at baseline.

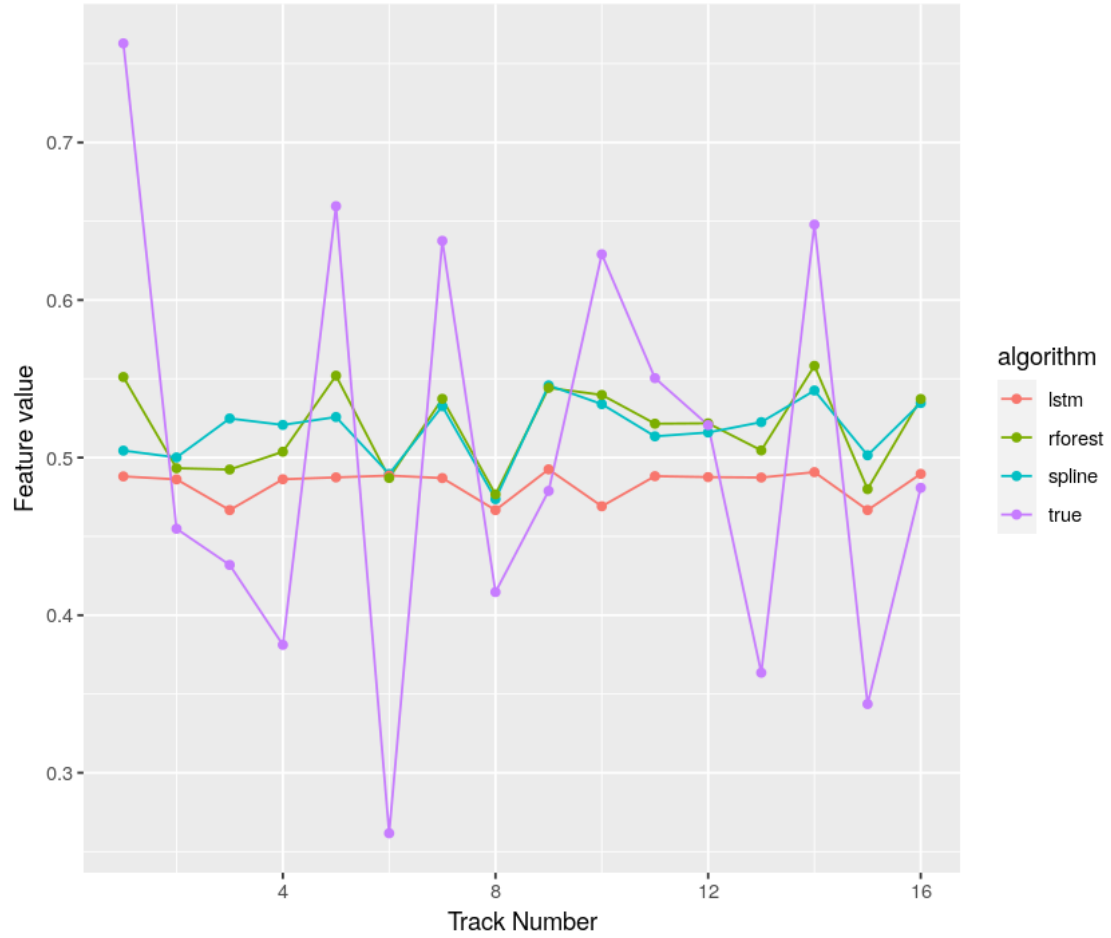
The next figure shows some predictions on randomly picked albums. Here it is clear that the Random Forest is really getting somewhere. It makes some right decisions towards increases/decreases of the next feature, but its magnitude is just very far from the actual data.



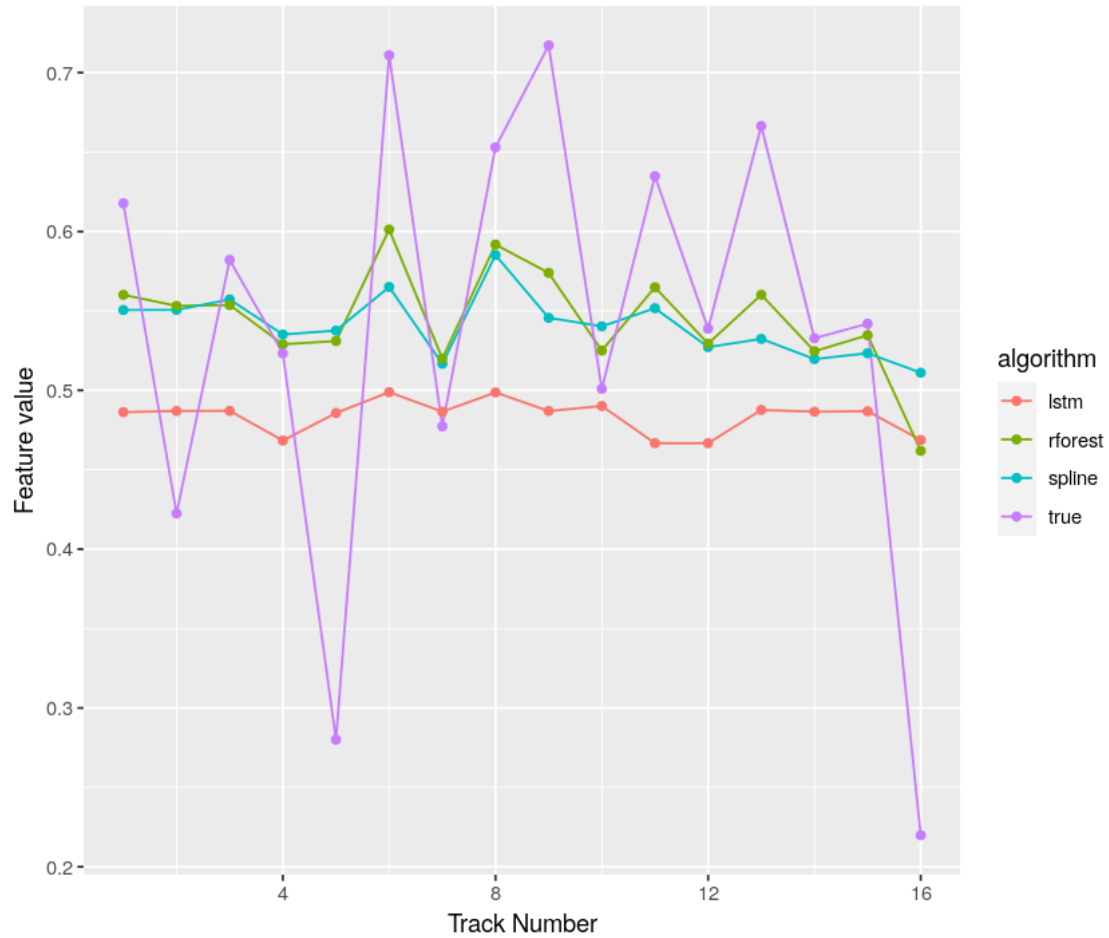
valence\_next  
RMSE spline: 0.144040583506345  
RMSE lstm: 0.149755001413725  
RMSE rForest: 0.123189432942204  
RMSE zerorule: 0.151935527288465

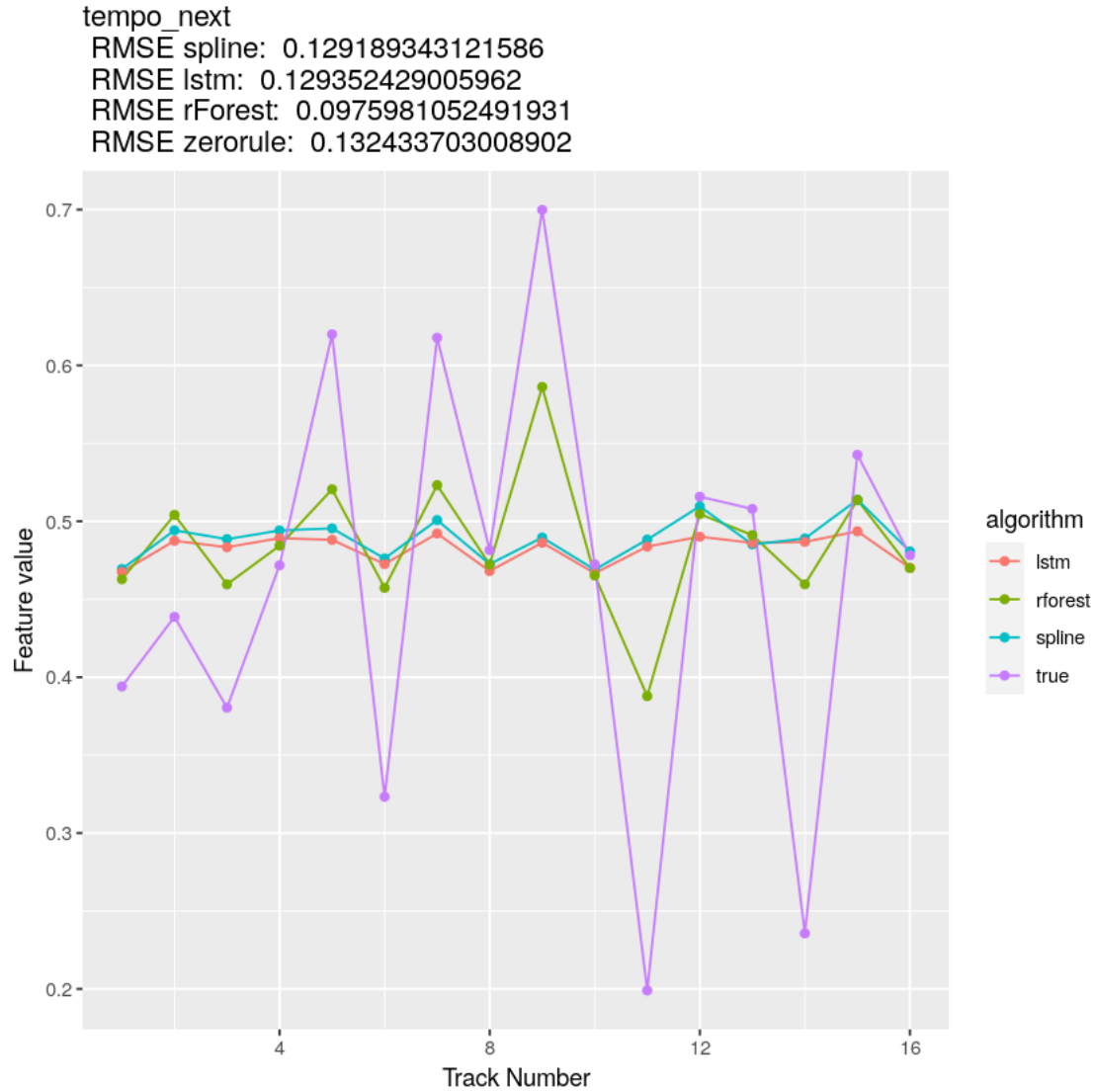


energy\_next  
RMSE spline: 0.127573250006259  
RMSE lstm: 0.13256354648546  
RMSE rForest: 0.112932576751307  
RMSE zerorule: 0.134062601013065



loudness\_next  
RMSE spline: 0.126685990797984  
RMSE lstm: 0.143691238082227  
RMSE rForest: 0.111228850562864  
RMSE zerorule: 0.135982916261812





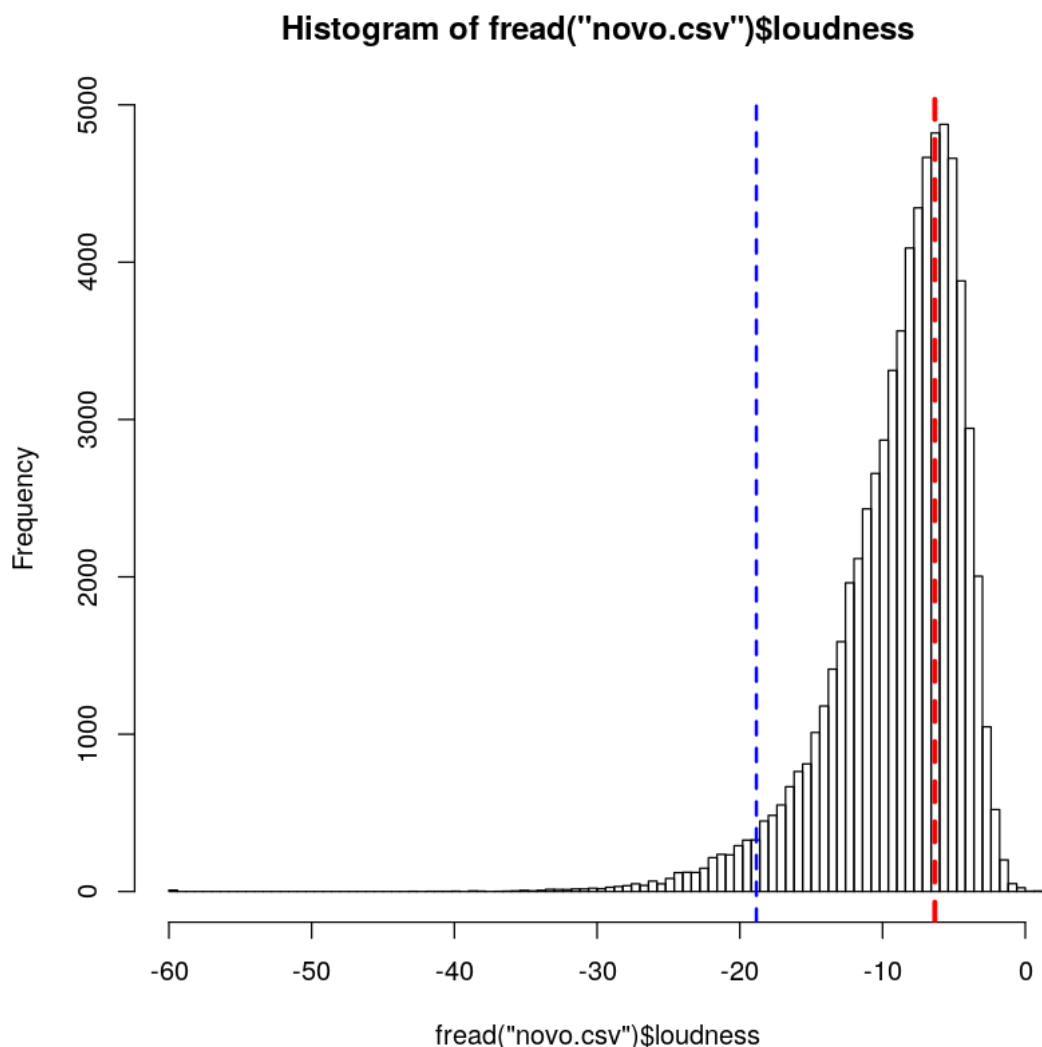
## 1.4 Accuracy versus RMSE

If we look at the predictions made by the random forest, we see that most of the predicted values are done in the correct direction. For instance, the last graph shows that, if the true valence of track in position  $i + 1$  is higher than that of the track in position  $i$ , the random forest algorithm also predicts feature in position  $i + 1$  to be higher than in position  $i$ .

Perhaps I can compute an accuracy score, which says if the model guessed the next track in the right direction or not. With this evaluation strategy, the random forest model would become very accurate, since most of its mistakes appear to be related to the magnitude of the next feature.

## 1.5 Possible issue

When we built these regressions models, we were trying to predict the feature of track  $i + 1$  given the feature(s) of track  $i$ . However, If we look at the pre-normalized distribution of say, loudness (below), we could hypothesize that, given an observed track (e.g. blue line), the loudness of the next track would, most likely, be higher than the current (e.g. red line). After z-score transformation, the distribution gets less skewed, and more like a gaussian curve. Still, this problem remains.



My point: If we use the current feature to predict the next feature, we might bias our models into going back to some central tendency measure, since values at the extreme of the distribution will most likely be followed by more probable values (a kind of regression towards the mean effect). So I'm wondering if any of these regression results are informative to the questions that we're asking.

If instead of asking "what is the feature of the next track?", we asked "where did they put this track



in the album?” we could avoid this distributional bias. For instance, the musician could choose to always open an album with a track at the extreme right of the loudness distribution, and to close the album with a track at the extreme left.

In fact, if we try to categorize the position of a track, we’re not explicitly looking for sequential effects. We are, however, trying to fit tracks into categories that correspond to a certain position in time (i.e. 1st, 2nd, 3d and 4th parts of the album).

## 1.6 Proposed next steps

I am also planning on:

- Tweaking the LSTM parameters to see if I can increase its accuracy;
- Training/testing the same models on the edges of the albums (i.e. 1st and last parts);
- (dissimilarity matrices) Perhaps we can find a way to include some of the dissimilarity matrix information in our regression models;
- (dissimilarity matrices - further ahear/brainstorming), we could think about treating each matrix as an image - again -, and training a deep learning model do generate new images (dissimilarity matrices) from a given seed (e.g. a list of album features).
- Try to improve the classification models.