model_parsed

December 2, 2020

1 Model 1

First I'm attempting to model Track Number Tn as a function of features f and weights β,

$$Tn(\beta, f) = \beta_0 + \beta_1 f_1 + ... + \beta_p f_p + \epsilon \tag{1}$$

where p is the number of features from our dataset. The intuition behind this model is that there could be a tendency of increase/decrease in track number related to increases/decreases in f.

For instance, based on these graphs, I would hypothesize that valence, energy and loudness get lower towards the end of the album. If this is true, β should be negative (as feature magnitude increases, track number decreases).

I chose to explain Tn as a function of f instead of trying to explain f as a function of Tn, because if I did it the second way, I would have to fit one model to each one of our dependent variable (f), or compute some sort of composite score of features (which is a possibility for the future).

1.1 Result - Model 1

```
Call:
lm(formula = track_number ~ valence * energy * loudness * tempo,
    data = dt)

Residuals:
    Min      1Q  Median      3Q     Max 
-7.7229 -3.1608 -0.2466  2.7683  9.2236 

Coefficients:
                                Estimate Std. Error t value Pr(>|t|)    
(Intercept)                    6.4356498  0.0201779 318.945  < 2e-16 ***
valence                       -0.1115567  0.0218890  -5.096 3.47e-07 ***
energy                        -0.1454631  0.0253258  -5.744 9.33e-09 ***
loudness                      -0.1744116  0.0251046  -6.947 3.77e-12 ***
tempo                         -0.0146344  0.0211298  -0.693 0.488568    
valence:energy                 0.0902890  0.0238297   3.789 0.000152 ***
valence:loudness              -0.0636685  0.0252488  -2.522 0.011684 *  
```

```
energy:loudness                 0.0710992  0.0183980   3.864 0.000111 ***
valence:tempo                  -0.0378966  0.0225093  -1.684 0.092266 .
energy:tempo                    0.0423689  0.0254210   1.667 0.095584 .
loudness:tempo                 -0.0224496  0.0258820  -0.867 0.385738
valence:energy:loudness        -0.0108280  0.0148856  -0.727 0.466975
valence:energy:tempo            0.0001661  0.0229706   0.007 0.994229
valence:loudness:tempo          0.0340216  0.0252154   1.349 0.177268
energy:loudness:tempo          -0.0174704  0.0178783  -0.977 0.328482
valence:energy:loudness:tempo -0.0147285  0.0136430  -1.080 0.280340
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.635 on 43557 degrees of freedom
Multiple R-squared:  0.01155,       Adjusted R-squared:  0.01121
F-statistic: 33.93 on 15 and 43557 DF,  p-value: < 2.2e-16
```

1.2 Discussion - Model 1

Results show an overall significant model, with significant βs for valence, energy and tempo. More importantly, β for these variables was negativelly correlated with Tn, which means that: as valence, energy and loudness increase, Tn decreases. In other words: albums tend to begin with more energy, loudness and higher valence.

1.2.1 Problems

R square shows a small effect size, explaining around 1% of the variance. Also, here I'm predicting continuous values for an rank-ordered variable (Tn). Model 3 presents an alternative to the second problem.

2 Model 2

Instead of predicting Tn by f, I'm predicting track position (1st, 2nd, 3d and 4th) based on averaged features for each position. Each album has 4 rows - one for each position - and 4 columns - one for each averaged feature.

2.1 Result - Model 2

```
Call:
lm(formula = position ~ valence * energy * loudness * tempo,
    data = dt)

Residuals:
```

```
    Min      1Q  Median      3Q     Max
-2.2830 -0.9532  0.0347  0.9765  2.6542

Coefficients:
                              Estimate Std. Error t value Pr(>|t|)
(Intercept)                   2.509563   0.010270 244.347  < 2e-16 ***
valence                      -0.114886   0.020183  -5.692 1.28e-08 ***
energy                       -0.169959   0.022596  -7.522 5.71e-14 ***
loudness                     -0.191242   0.021928  -8.721  < 2e-16 ***
tempo                        -0.006069   0.019258  -0.315   0.7527
valence:energy               -0.007481   0.035399  -0.211   0.8326
valence:loudness             -0.040723   0.037343  -1.090   0.2755
energy:loudness              -0.019458   0.028697  -0.678   0.4978
valence:tempo                -0.022490   0.033452  -0.672   0.5014
energy:tempo                  0.033016   0.037552   0.879   0.3793
loudness:tempo               -0.008280   0.036903  -0.224   0.8225
valence:energy:loudness      -0.028984   0.036509  -0.794   0.4273
valence:energy:tempo         -0.079617   0.047893  -1.662   0.0965 .
valence:loudness:tempo        0.113875   0.052367   2.175   0.0297 *
energy:loudness:tempo        -0.028120   0.043247  -0.650   0.5156
valence:energy:loudness:tempo -0.112389   0.045591  -2.465   0.0137 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.098 on 15076 degrees of freedom
Multiple R-squared:  0.03668,	Adjusted R-squared:  0.03572
F-statistic: 38.27 on 15 and 15076 DF,  p-value: < 2.2e-16
```

2.2 Discussion - Model 2

R squared increased to 0.036, which could be expected, since I drastically reduced the variance by avearging each feature by album section. Still, the model shows that Tn gets lower with increases in valence, energy and loudness.

3 Model 3

Here I'm using a multinomial logistic regression to predict whether a track is in position 1, 2, 3 or 4. Predictor varibles are normalized and features are averaged within positions.

I fit the model to 75% of the data, and evaluated its performance on the remaining 25%. Data was perfectly balanced with 25% of trials on each category.

3.1 Evaluation strategies

1) Accuracy - Performance was evaluated using a cross-validation strategy: I fitted the model on 10 different training sets, and evaluated it on 10 different test sets.
2) Sanity check - In order to evaluate if our results could be due to something different from track ordering factors, I repeated the same model on a randomized version of the original dataset. First I shuffled track orders within each album, and then I computed the average features within each album position, for each album. Should our model have any value, fitting it in a randomized dataset would decrease its accuracy. Results are also cross validated.

3.2 Results 3

3.2.1 Accuracy

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4
         1 4664 3604 3309 2702
         2 1144 1368 1290 1167
         3 1210 1439 1480 1268
         4 2452 3059 3391 4333

Overall Statistics

               Accuracy : 0.3127
                 95% CI : (0.308, 0.3174)
    No Information Rate : 0.25
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.0836

 Mcnemar's Test P-Value : < 2.2e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.4925  0.14446  0.15628   0.4576
Specificity            0.6616  0.87325  0.86213   0.6867
Pos Pred Value         0.3266  0.27531  0.27423   0.3274
Neg Pred Value         0.7964  0.75382  0.75403   0.7916
Prevalence             0.2500  0.25000  0.25000   0.2500
Detection Rate         0.1231  0.03611  0.03907   0.1144
Detection Prevalence   0.3770  0.13118  0.14248   0.3494
Balanced Accuracy      0.5770  0.50885  0.50920   0.5721
[1] Number of observations per category:
```

```
   1    2    3    4 
9470 9470 9470 9470 
```

3.2.2 Sanity check

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4
         1 2035 2071 2043 1914
         2 2493 2438 2538 2431
         3 3487 3464 3463 3772
         4 1425 1467 1396 1323

Overall Statistics
                                          
               Accuracy : 0.2452          
                 95% CI : (0.2409, 0.2496)
    No Information Rate : 0.25            
    P-Value [Acc > NIR] : 0.9846          
                                          
                  Kappa : -0.0064         
                                          
 Mcnemar's Test P-Value : <2e-16          

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity           0.21557  0.25826  0.36684  0.14015
Specificity           0.78715  0.73651  0.62136  0.84859
Pos Pred Value        0.25239  0.24626  0.24411  0.23579
Neg Pred Value        0.75065  0.74867  0.74646  0.74752
Prevalence            0.25000  0.25000  0.25000  0.25000
Detection Rate        0.05389  0.06457  0.09171  0.03504
Detection Prevalence  0.21353  0.26218  0.37569  0.14860
Balanced Accuracy     0.50136  0.49739  0.49410  0.49437

[1] Number of observations per category:

   1    2    3    4 
9440 9440 9440 9440 
```

3.3 Discussion - Model 3

With an overall accuracy of 31%, multinomial logistic regression barely surpasses chance level (25%).

However, the **error** was not well distributed across categories, and there was a higher accuracy for tracks in the **1st and 4th positions** of the album.

Sensitivity (proportion of true positives) was higher for positions 1 and 4. Specificity (proportion of true positives), on the other hand, was lower for these categories.

With the sanity check (shuffled dataset), we acheived an accuracy of 0.247 throughout all categories, and a uniform distribution throughout the confusion matrix.

4 Model 4

Here I'm attempting to predict the same thing as models 2 and 3, but with a random forest model. Evaluation was also done in terms of accuracy and sanity check.

4.1 Results 4

4.1.1 Accuracy

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4
         1 3987 2194 2097 1743
         2 1832 3351 1942 1734
         3 1807 1871 3333 1934
         4 1814 2024 2068 4029

Overall Statistics

               Accuracy : 0.3893
                 95% CI : (0.3844, 0.3942)
    No Information Rate : 0.25
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.1857

 Mcnemar's Test P-Value : 6.147e-16

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.4224  0.35498  0.35307   0.4268
```

```
Specificity           0.7869  0.80551  0.80184   0.7915
Pos Pred Value        0.3979  0.37826  0.37261   0.4055
Neg Pred Value        0.8034  0.78932  0.78806   0.8055
Prevalence            0.2500  0.25000  0.25000   0.2500
Detection Rate        0.1056  0.08874  0.08827   0.1067
Detection Prevalence  0.2654  0.23461  0.23689   0.2631
Balanced Accuracy     0.6046  0.58024  0.57745   0.6091
```

```
[1] Number of observations per category:


   1    2    3    4 
9440 9440 9440 9440 
```

4.1.2 Sanity check

```
Confusion Matrix and Statistics

          Reference
Prediction    1    2    3    4
         1 2610 2258 2269 2136
         2 2266 2568 2277 2151
         3 2284 2272 2531 2184
         4 2280 2342 2363 2969

Overall Statistics
                                          
               Accuracy : 0.2828          
                 95% CI : (0.2782, 0.2874)
    No Information Rate : 0.25            
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.0437          
                                          
 Mcnemar's Test P-Value : 0.002849        

Statistics by Class:

                     Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity           0.27648  0.27203  0.26811  0.31451
Specificity           0.76472  0.76363  0.76201  0.75335
Pos Pred Value        0.28146  0.27726  0.27300  0.29827
Neg Pred Value        0.76024  0.75886  0.75749  0.76728
Prevalence            0.25000  0.25000  0.25000  0.25000
Detection Rate        0.06912  0.06801  0.06703  0.07863
Detection Prevalence  0.24558  0.24529  0.24552  0.26361
Balanced Accuracy     0.52060  0.51783  0.51506  0.53393
```

```
[1] Number of observations per category:

   1    2    3    4
9440 9440 9440 9440
```

4.2 Discussion - Model 4

Accuracy increases to 38% on 10 runs of crossvalidation. Correct predictions are still concentrated on the 1st and 4th sections (balenced accuracy is promosing for these categories).

Again, shuffled album orders dropped the accuracy to baseline.

5 Overall discussion

It seems like models 3 and 4 are more accurate for sections 1 and 4.

These results made me think back about the dissimilarity matrix, which shows that Track 1 has a generally higher dissimilarity rating.

Perhaps there is a higher level of information within the first and last sections of the album. Even though I didn't find any study about this "opening/closing effect" , it seems like there is some mentions to it in blog posts that teach you how to sequence album and playlist tracks:

5.1 Anecdotes

- 'Lead with an impactful track. It's vital for inviting your listeners in for the long haul. The lead track is your album's first impression, so make it count.' Blog post
- 'Finish strong - The last track on the album should stay with the listener – if you've front loaded tracks 1 to 3 then save the next "best" to end with.' Blog post
- 'you need to let listeners know straightaway why you are worth listening to. Don't start with a skit. Don't start with that cool, arty track you love but no one else seems to because you want to "set the mood." Hit 'em with your best shot, right off the bat.' Blog post
- 'Start with a hook. Whatever your theme, genre, or taste, one thing about playlists is universal: it's got to start with a great song. Lead off with a song that'll hook everyone who'll listen, or will kick off your personal favorites playlist with a bang.' Blog post
- 'Skip to the end of each song, so that you get the last 15 to 20 seconds of it and figure out how that transition sounds. Even if you don't know what the tempo is precisely, you can still tell the difference between an upbeat song and a slow song' Blog post