lstm_eval_parsed

January 26, 2021

1 LSTM model

1.1 Data

Here I am doing my first attempt to use Spotify's low-level descriptors. Rathar than having a single descriptor for each track, I'm working with several windows of audio description throughout each track.

- **Track lengths:** All tracks in the dataset have 5 non-overlapping time windows. In the original API-call, data comes with different lengths of time frame (e.g. track 1 has 10 observations and track 2 has 5). In order to provide regular sized inputs to the LSTM architecture, I divided each track length in 5 quantile bins and calculated the mean for quantile within within each bin.
- **Feature engineering:** I also had the idea of calculating cummmulative track properties. The cumulative feature would work as a forced carrier of information from previous tracks. For Instance, if overall valence of track_1 is 10, and overall valence of track_2 is 5, the cummulative valence would be 15. I did that for all variables in our data set. For the first track, cumulative value is simply the mean value of the feature within the track.
- **Data transformation:** Finally, all data was normalized with min-max transform, which I hear works better than z-scores for deep learning models by preserving the shapes of the distributions (?).

Below I'm showing two tracks of one album (5 rows for each).

[229]:

	track	loudness_overall	loudness_continuous	loudness_overall_cum
0	1	1.87827	1.88457	1.01777
1	1	1.87827	1.88814	1.01777
2	1	1.87827	1.88782	1.01777
3	1	1.87827	1.90497	1.01777
4	1	1.87827	1.85939	1.01777
5	2	1.88995	1.88711	1.06886
6	2	1.88995	1.89586	1.06886
7	2	1.88995	1.89616	1.06886
8	2	1.88995	1.89637	1.06886
9	2	1.88995	1.90940	1.06886

*Note: overal loudness is the grand average of the feature "loudness_continuous", which has 5 distinct values for each track. Here I'm asking if the overall loudness of track $i + 1$ is higher or lower than the overall loudness of track i.

1.2 Model architecture

The idea is to feed the LSTM with track i and ask it to predict wether valence, energy, loudness, tempo go up or down in relation to the current feature value.

- **Features:** all descriptive features from Spotify's web API plus the cumulative features described before.
- **Input:** 5 time steps of each feature of track i.
- **Output:** Categorical transformation of valence, energy, loudness and tempo from track $i+1$ (i.e. "greater" or "lower")

1.2.1 Input:

[232]:

	track	loudness_overall	loudness_continuous	loudness_overall_cum
0	1	1.87827	1.88457	1.01777
1	1	1.87827	1.88814	1.01777
2	1	1.87827	1.88782	1.01777
3	1	1.87827	1.90497	1.01777
4	1	1.87827	1.85939	1.01777
5	2	1.88995	1.88711	1.06886
6	2	1.88995	1.89586	1.06886
7	2	1.88995	1.89616	1.06886
8	2	1.88995	1.89637	1.06886
9	2	1.88995	1.90940	1.06886

1.2.2 Output:

[233]:

	track	loudness_output
0	2	Up

The output indicates that the overall loudness of track 2 went up in relation to loudness of track 1.

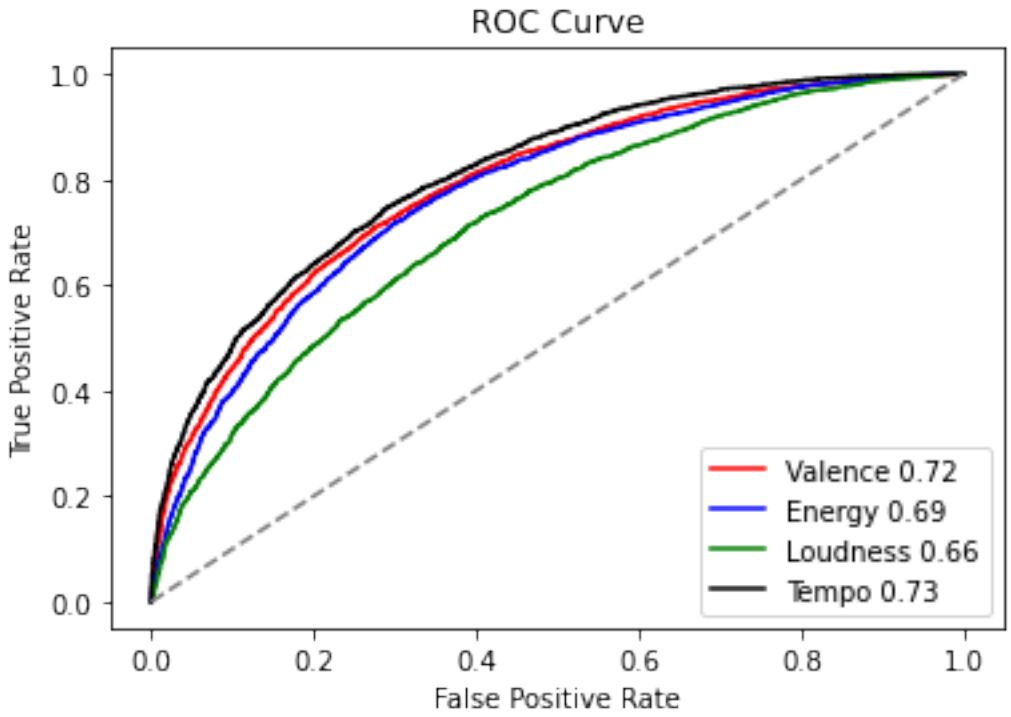
2 Model evaluation

Data was trained on 80% of the dataset (without duplicate albums), and tested on the remaining 20%. Accuracy ranges from 69% to 72%, against a baseline of 50%.

The complete code that I wrote (in python) to train and test the model is here.

2.1 Accuracy by feature

2.2 ROC curve and AUC for each feature

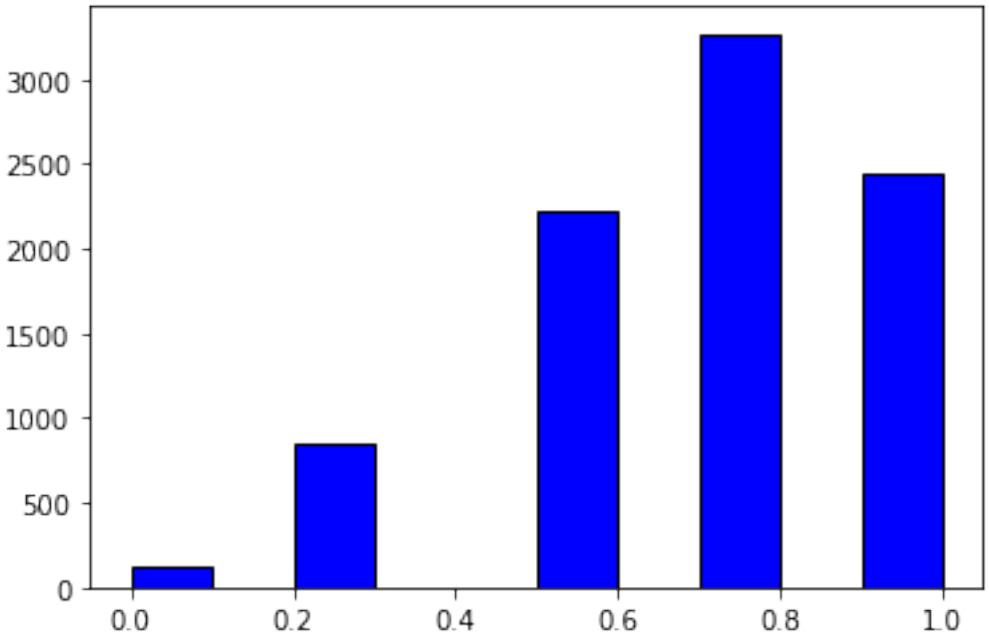


2.3 Percentage of agreement

The metrics above relate to single feature. Below im evaluating the overall percentage of agreement (from real to predicted outputs) together with all variables. I basically ran the percentage of agreement between two arrays: 1) the real categorical output for valence, energy, loudness and tempo, and 2) the predicted categorical output for the same variables.

Results are plotted below.

[212]:
```
(array([ 123.,    0.,  843.,    0.,    0., 2219.,    0., 3269.,    0.,
        2437.]),
 array([0. , 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1. ]),
 <BarContainer object of 10 artists>)
```



Overall agreement: 0.6983466426723653