

Lyrics classification

The document at hand is the written documentation part for the Lyrics classification task received within the Information Retrieval and Text Mining course.

Before all, a small exploratory data analysis has been conducted. The first thing that became apparent was that, the target labels are somewhat unbalanced. Thus, the following metrics have been considered throughout the whole experimentation process: accuracy, f1-score, precision, recall and Matthew's correlation coefficient. By checking various aspects of the train dataset I tried to come up with various features:

- mean_verse_length - the average length of a verse, defined by the \n newline character
- unique_verses_count - how many unique verses are in the lyrics
- repeated_verses_num - how many times the repeated verses have been repeated
- filtered_verse_counts_mean_repetition_count - if multiple unique repetitions exist, the average of their count
- contains_chorus - whether a reference to a chorus is present
- contains_I - how many mentions are of the self
- double_quotes - whether there exist a reference, that is highlighted by a double quote
- contains_question - whether the lyrics, contain a question
- contains_exclamation - whether the lyrics, contain a question
- contains_profanity_1 - whether two common swear words are included in the verses
- contains_profanity_2 - whether another common swear word is included in the verses
- profanity_count - count of the above mentioned profanities

And lastly, I was experimenting w/ various topic words that could emphasise one genre or another. The following five have been included

- random_words_1 - baby; a common reference in various pop songs
- random_words_2 - love|desire
- random_words_3 - happy
- random_words_4 - river|mountain|flower
- random_words_5 - hell

With regards to the modelling part, it should be noted, that each cross-validation implied stratification as well.

Next I gathered a couple of models and vectorisers with various parameters:

```
LogisticRegression(),
KNeighborsClassifier(),
RandomForestClassifier(n_jobs=-1),
XGBClassifier(tree_method = gpu_hist),
LGBMClassifier(),
SVC(decision_function_shape = ovo, class_weight=balanced)
MultinomialNB()
```

```
vectorizer_1 = CountVectorizer(ngram_range=(1,1))
vectorizer_2 = CountVectorizer(ngram_range=(1,2))
vectorizer_3 = CountVectorizer(ngram_range=(1,3))
```

```
vectorizer_4 = TfidfVectorizer(ngram_range=(1,1))
vectorizer_5 = TfidfVectorizer(ngram_range=(1,2))
vectorizer_6 = TfidfVectorizer(ngram_range=(1,3))
```

And, tried to find which of these combinations performs best/make sense, given the metrics mentioned earlier. Next, vectorisers 3 and 6 have been removed as they performed quite bad, and I tried to reduce the size of the vocabulary via the parameters of the vectorisers. Samples w/ less than 5 occurrences and samples that appear in more than 75% of the entries have thus been removed.

In a subsequent step, I've added all the previously mentioned features, and experimented with the best models from an earlier step: LightGBM, Logistic regression, XGBoost, SVC and MultinomialNB. Given, that the features have been created in a rather naive way, I wanted to perform a simple feature selection on them. The approach consisted of taking a LightGBM model and one-by-one adding a feature and checking whether it influenced the outcome in a positive way. If so, the feature was kept. In the contrary case, it was dropped. To somewhat diminish the effect of randomness, I've performed the same iteration with the features reversed, and removed at last, only those features that have been excluded by both iterations.

The following features have been kept:

- mean_verse_length,
- unique_verses_count,
- repeated_verses_num,
- filtered_verse_counts_mean_repetition_count,
- contains_chorus,
- contains_l,
- double_quotes,
- contains_question,
- contains_exclamation,
- contains_profanity_2,
- profanity_count,
- random_words_1,
- random_words_4,
- random_words_5,

At last, hyper parameter searches have been conducted for some combinations of model, vectoriser and features via the Bayesian optimisation library Optuna. Below, are the results of these attempts, first cross-validated on the train set, than evaluated on the test set:

LightGBM, count vectoriser w/ limited vocabulary and the selected features

- cross validated on the train set using three folds

	acc	f1	precision	recall	mcc
fold 1	0.439151	0.420392	0.471176	0.439151	0.361562
fold 2	0.425539	0.407937	0.459130	0.425539	0.345406
fold 3	0.422784	0.408719	0.461228	0.422784	0.342722

- test set

Metrics:

Accuracy	0.4417139256458727
F-Score	0.42647891464019655
Precision	0.47193768428700705
Recall	0.4417139256458727
Matthew's cc	0.36493470205418543

	precision	recall	f1-score	support
Country	0.58	0.49	0.53	810
Electronic	0.31	0.15	0.21	660
Folk	0.50	0.20	0.29	495
Hip-Hop	0.83	0.81	0.82	960
Indie	0.32	0.06	0.10	510
Jazz	0.47	0.34	0.40	660
Metal	0.65	0.57	0.60	810
Pop	0.33	0.43	0.37	1110
R&B	0.48	0.13	0.21	510
Rock	0.30	0.61	0.40	1410
...				
accuracy			0.44	7935
macro avg	0.48	0.38	0.39	7935
weighted avg	0.47	0.44	0.43	7935

XGB, count vectoriser w/ limited vocabulary and the selected features

- cross validated on the train set using three folds

	acc	f1	precision	recall	mcc
fold 1	0.400907	0.366084	0.444758	0.400907	0.319322
fold 2	0.397991	0.361801	0.454577	0.397991	0.315838
fold 3	0.402366	0.372051	0.451313	0.402366	0.321355

- test set

Metrics:

Accuracy	0.39798361688720857
F-Score	0.3653362327263961
Precision	0.43722293346846725
Recall	0.39798361688720857
Matthew's cc	0.31581716180178876

	precision	recall	f1-score	support
Country	0.50	0.41	0.45	810
Electronic	0.41	0.07	0.13	660
Folk	0.62	0.10	0.18	495
Hip-Hop	0.78	0.81	0.79	960
Indie	0.00	0.00	0.00	510
Jazz	0.43	0.17	0.24	660
Metal	0.55	0.51	0.53	810
Pop	0.32	0.40	0.36	1110
R&B	0.56	0.11	0.19	510
Rock	0.26	0.66	0.37	1410
accuracy			0.40	7935
macro avg	0.44	0.32	0.32	7935
weighted avg	0.44	0.40	0.37	7935

MultinomialNB, same setting as above.

	acc	f1	precision	recall	mcc
fold 1	0.415978	0.401978	0.401012	0.415978	0.339913
fold 2	0.419543	0.407519	0.410115	0.419543	0.343835
fold 3	0.418247	0.406186	0.409162	0.418247	0.341748

Metrics:

Accuracy	0.43289224952741023
F-Score	0.42122872078919676
Precision	0.4216986057622007
Recall	0.43289224952741023
Matthew's cc	0.35958950942814444

	precision	recall	f1-score	support
Country	0.46	0.58	0.52	810
Electronic	0.29	0.13	0.18	660
Folk	0.38	0.36	0.37	495
Hip-Hop	0.76	0.77	0.76	960
Indie	0.24	0.21	0.22	510
Jazz	0.46	0.37	0.41	660
Metal	0.54	0.70	0.61	810
Pop	0.33	0.41	0.37	1110
R&B	0.29	0.22	0.25	510
Rock	0.35	0.34	0.35	1410
accuracy			0.43	7935
macro avg	0.41	0.41	0.40	7935
weighted avg	0.42	0.43	0.42	7935