Classification Writeup, Miriam A Feldman

I. Describing the training set

Words per song by genre

First, I totaled the number of words in each song. The results reveal that hip hop and rap have much higher mean number of words per song than pop and rock, with rap lyrics having both the highest mean and median words per song. These are in line with expectations, as "hip hop lyrics are typically rapped, and thus have more words." (Thompson, 2021)

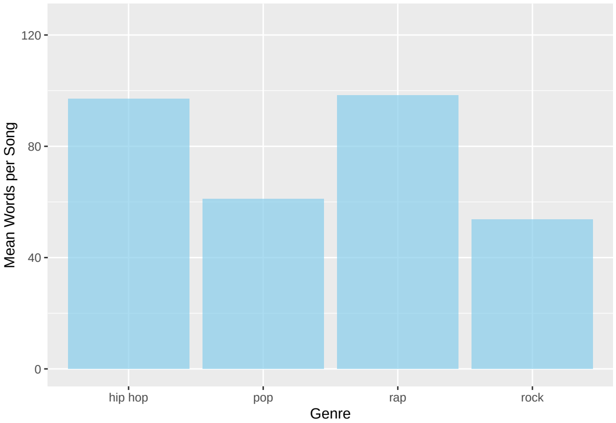

Figure 1: Mean Words Per Song by Genre

Profanity by genre

In line with Jewalikar and Fragapane (2015), I use the list of profane words compiled by Lewis (2014), available at https://gist.github.com/ryanlewis/a37739d710ccdb4b406d to determine the profanity of words used in song lyrics. Profanity levels vary by genre, with a stark divide between hip hop/rap and pop/rock. Hip hop and rap have the highest average of profane words per song (3.48 and 3.72, respectively), which differs greatly from pop and rock, both of which use less than one profane word per song on average. Hip hop is the genre with the highest proportion of songs featuring some profanity in the training data (79%), with rap a close second (74%). Lower proportions of pop and rock songs feature any profanity (27% and 21%, respectively).

Table 1: Profanity by Genre

genre	Mean Profane Words per Song	Proportion of Songs Featuring Profanity
hip hop	3.48	0.79
pop	0.68	0.27
rap	3.72	0.74
rock	0.46	0.21

Word clouds by genre

The word clouds can give visual intuition for the difference in vocabulary and themes across genres. Hip hop and rap feature many words themed around money and women (e.g. "money", "girl", "b*tches", "h*es"), as well as the profanity discussed above, while pop's core vernacular reflects uplifting words such as "tonight," "better," "heart," and "good."

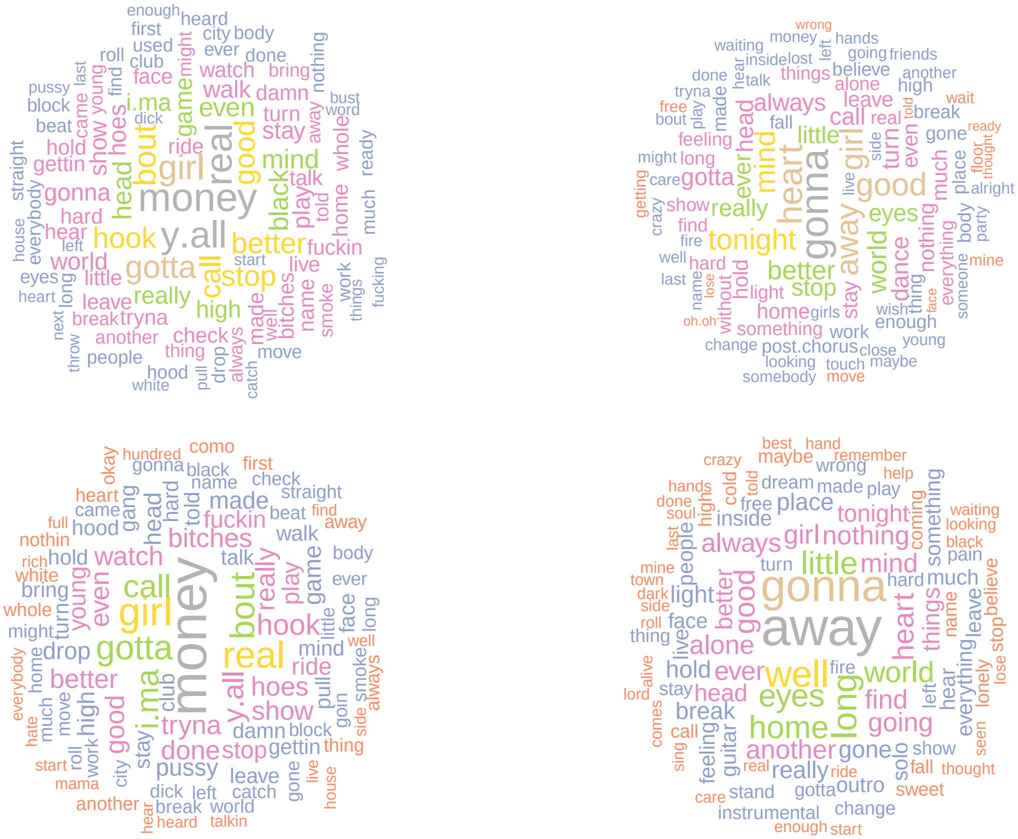

Figure 2: Most Frequent Words by Genre. Clockwise from top left: Hip hop, pop, rock, rap

Audio features by genre

Audio features reveal differences between genres which are otherwise similar when just lyric-based features are considered. For instance, hip hop and rap might be distinguished on the basis of their average tempo (114.17 and 123.08, respectively). Similarly, rock and pop might be differentiated on the basis of their length: pop songs are the shortest on average among the genres we investigate, with the only median length under 3.5 minutes. I expected speechiness to sharply distinguish rap from hip hop (with rap lyrics more often spoken than sung), but their summary statistics did not reflect this, so the feature does not seem as useful as some of the other audio features provided.

Table 2: Audio Features by Genre

genre	Median Tempo	Median Length (min)	Median Speechiness
hip hop	100.00	3.90	0.24
pop	121.99	3.47	0.05
rap	126.03	3.70	0.19
rock	122.98	3.98	0.04

II . Feature Construction and Transformation

Following Model 1, four features resulting from this preliminary analysis of the training data were constructed and added to the data set: `totalwords`, the word count of the song; `profanewords`, the total number of profane words in the song, `profane`, a binary variable (1 if any profanity, 0 if clean); and `profanity_rate`, profane words as proportion of total words in the song. Each of these appear to be useful for distinguishing rap and hip hop from pop and rock, and are included on this basis.

Until Model 4B, audio features were maintained in their original form. A final stage of feature engineering examines their distributions to assess where transformations might be useful. This observation found some strongly right-skewed features. After checking whether they feature many true-zero values, as log(0) is undefined, and finding that although values were very small they were non-zero, `audio_speechiness` `audio_acousticness` and `audio_liveness` were log-transformed. An illustrative example of the impact on `audio_liveness` is shown below.

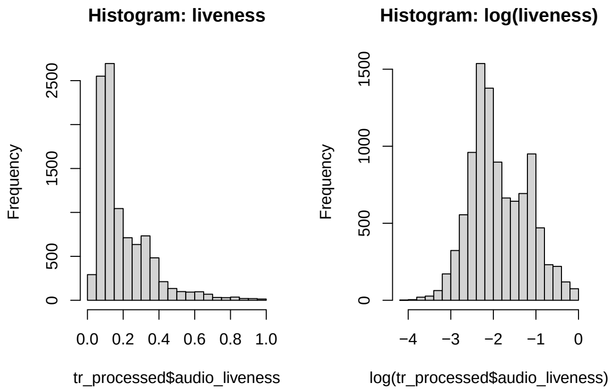

Figure 3: Log-Transformation of audio liveness

III. Algorithms

Summary of Models Submitted to Kaggle

Model	Method	PCA	Embedding	log Transformations	Tuning	Test F1
1	Logistic	All Features	BoW	No	No	0.51971
2	Logistic	Lyrics	BoW	No	No	0.64732
2A	Lasso	Lyrics	BoW	No	Yes	0.64619
3	Logistic	Lyrics	TF-IDF	No	No	0.64816
4	SVM	Lyrics	TF-IDF	No	No	0.50563
4A	SVM	Lyrics	TF-IDF	No	Yes	0.66647
4B	SVM	Lyrics	TF-IDF	Yes	No	0.66957
5	LightGBM	None	TF-IDF	Yes	No	0.67774
5A	LightGBM	Lyrics	TF-IDF	Yes	Yes	0.67549
5B	LightGBM	None	TF-IDF	Yes	Yes	0.68422

Dimension Reduction: PCA

Here, dimensionality reduction is performed, particularly because of the high dimensionality in lyric features (there are over 1200 features per observation, just with lyric words alone!). While other dimensionality reduction techniques such as t-SNE might be preferred for visualisation, in this case dimensionality reduction is being used as part of the data processing pipeline for the classification algorithms below, so PCA is selected. To select the number of principal components, I checked the scree plots of proportion of variance explained for each component for a visually apparent "elbow" point, which resulted in selecting three principal components.

Following the decreased performance of Model 1 relative to the audio-feature-only benchmark provided, the PCA approach was adjusted to only reduce the dimensionality of the lyric-based features (as those are the features which make the data extremely high-dimensional). Following Model 2, PCA was repeated with the TF-IDF calculations; in this case the scree plot had an elbow at four features, so four principal components were maintained.

Alternative Word Embedding: TF-IDF

The Bag of Words approach is criticised for genre classification by lyrics. (Kumar et al., 2018). However, the data is structured in this fashion rather than given in full text. More semantics-driven analysis does not appear possible. Therefore, as one alternative approach, in line with Mayer et al. (2008) (Thompson, 2021) and Boonyanit and Dahl, I constructed term frequency–inverse document frequency (TF-IDF) embeddings for each word. The corpus, in this case (used to calculate inverse document frequency), is the collection of training data. This approach aims to "extract words which are more frequent in certain genres" as well as down-weight terms which appear frequently across the corpus such as "the" or "and." (Thompson, 2021)

Model 1: Logistic Regression, PCA on all features

Submitted 10 May, Test F1: 0.51971

The same basic multinomial logistic model as the benchmark was fit after PCA was performed on the entire data set. This approach decreased performance, reducing F1 on the test set from the benchmark of 0.57971 to **0.51971**, the validation set F1 by genre is reported below:

hip hop	pop	rap	rock
0.5617716	0.4037075	0.5363735	0.6336000

Model 2: Logistic Regression, Additional Features and PCA on lyrics only

Submitted 10 May, Test F1: 0.64732

Model 2, still a logistic regression, includes all audio features, my constructed features for profanity and word count, and the first three principal components from PCA on the lyric features only. It achieves improved fit over Model 1 and the Benchmark, with a test set F1 of **0.64732**—the validation set F1 by genre is additionally reported below:

hip hop	pop	rap	rock
0.5828571	0.6030881	0.5655471	0.7520436

Model 2A: Lasso Regression, 10-Fold CV

Submitted 13 May, Test F1: 0.64619

I performed 10-fold cross-validation on a lasso model based on the data used in Model 2, with hyperparameter tuning on the value of λ. I selected the model with λ 1 standard error away from the λ yielding the best training data error rate, with the goal of reducing overfitting to the training set. This approach resulted in a CV error rate of 1.695 for the 1se value of λ , and a test set F1 of **0.64619**, very similar to Model 2.

Table 6: Lambdas and CV Errors

Lambda	Value	CV.Error
Minimum	0.0004	1.6861
1 Standard Error	0.0037	1.6959

Model 3: Logistic Regression Based on TF-IDF

Submitted 13 May, Test F1: 0.64816

For Model 3, I attempted the same algorithm as Model 2, but with an alternative word embedding, motivated by Mayer et al. (2008) and Boonyanit and Dahl (n.d.). PCA was performed in this case on the TF-IDF weighted lyric attributes. Here, due to the scree plot, I kept four principal components. This resulted in a slight performance improvement on the test set to Test Set F1: **0.64816**, but the validation set performance was broadly similar (with marginally decreased performance in hip hop):

hip hop	pop	rap	rock
0.5627803	0.5984848	0.5644302	0.7547170

Model 4: SVM

Submitted 17 May, Test F1: 0.50563

In line with Kumar et al. (2018) and Rajanna et al (2015), I used an SVM as my next model attempt. However, without tuning, this model heavily overfit to the training data, yielding an extremely high training set F1 in the 0.90-1.00 range for all genres (see table below), but a drastically worse test set performance, with test set F1 of only **0.50563**.

hip hop	pop	rap	rock
0.9249047	0.9682075	0.9175299	0.9903561

Model 4A: Tuned SVM

Submitted 17 May, Test F1: 0.66647

Following the initial SVM attempt, I tuned the hyperparameters: cost (the regularization parameter which defines the cost of a violation to the margin) and gamma (a hyperparameter for curvature). By default, `tune.svm()` performs 10-fold cross validation, which I anticipated would help to reduce the overfitting observed in the initial SVM by holding out validation sets with which to assess performance. After searching a grid of cost values smaller than my initial cost of 1: cost = $2^{-5}, 2^{-4}, ...2^{-1}$ and gamma values $\gamma = 0.005, 0.01, 0.015, 0.02, 0.025, 0.03$, the best model was selected with cost 0.5 and gamma 0.03. The training set F1 statistics were much more in line with prior models. Tuning significantly improved the performance on the test set, leading to an F1 of **0.66647**.

hip hop	pop	rap	rock
0.6176808	0.6737708	0.6279119	0.8034713

The cost, gamma, and CV error for the five top-performing models are reported below:

Table 10: Tuning Model 5A

gamma	cost	error	dispersion
0.025	0.50	0.340	0.012
0.030	0.50	0.340	0.012
0.020	0.50	0.340	0.012
0.030	0.25	0.341	0.012
0.025	0.25	0.342	0.012

Model 4B: SVM After Log-Transformations

Submitted 20 May, Test F1: 0.66957

Following the log-transformations of `audio_speechiness`, `audio_acousticness` and `audio_liveness`, the SVM was run again with the cost (0.5) and gamma (0.03) values from Model 5A, and Test Set F1 improved to **0.66957**.

Model 5: LightGBM

Submitted 20 May, Test F1: 0.67774

Moving forward from the improved performance of the SVM and the final stages of feature engineering, I subsequently constructed a LightGBM model on a data set with the audio features (with the log-transformations above) and the raw TF-IDF scores for lyrics (without PCA). After preparing the data in a lightgbm dataset, and separating a validation set, the model was fit using outcome `multiclass`, as our genre classification task features more than two outcomes. While LightGBM itself was not featured in the literature I reviewed on genre classification, Boonyanit and Dahl note the success other authors have found (notably, Kumar et al) with XGBoost, another popular boosting method. To mitigate the risk of overfitting, I employed early stopping. Its validation set F1 is reported below:

hip hop	pop	rap	rock
0.5593420	0.6995349	0.6212425	0.8011152

Model 5A: Tuned LightGBM with PCA

Submitted 21 May, Test F1: 0.67549

Models 5A and 5B were tuned using random hyperparameter search using 200 draws (to save time over performing a comprehensive grid search). Maximum depth values were all kept below 10 to reduce overfitting to the training data, and `multi_logloss` was calculated based on 5-fold cross-validation. In both cases, a `lgb.train` model was subsequently fit with the best-performing parameters, with early stopping based on the validation set.

This model was trained on the same data as Model 4B, and and tuned with random hyperparameter search. The hyperparameters and 5-fold CV loss scores for the 5 best-performing models are reported below. The subsequent Test Set F1 of **0.67549** in Model 5A was slightly worse than in Model 5.

Table 12: Tuning Model 6A

learning_rate	num_leaves	max_depth	feature_fraction	bagging_fraction	is_unbalance	score
0.072	70	5	0.718	0.505	FALSE	0.771
0.054	107	5	0.649	0.595	FALSE	0.775
0.043	36	8	0.844	0.735	TRUE	0.777
0.082	152	4	0.781	0.248	TRUE	0.777
0.174	49	3	0.610	0.442	TRUE	0.777

hip hop	pop	rap	rock
0.5625745	0.7044146	0.6253687	0.8076225

Model 5B: Tuned LightGBM without PCA

Submitted 21 May, Test F1: 0.68422

Model 5B was trained on the same data as Model 5, and tuned with random hyperparameter search. The hyperparameters and 5-fold CV loss scores for the 5 best-performing models are reported below. Using the best-performing model hyperparameters, the resulting LightGBM model attained a Test Set F1 of **0.68422** and the validation set F1 reported below, improving over Model 5 in all genres except pop.

Table 14: Tuning Model 6B

learning_rate	num_leaves	max_depth	feature_fraction	bagging_fraction	is_unbalance	score
0.076	57	7	0.651	0.886	FALSE	0.750
0.104	17	7	0.279	0.350	FALSE	0.751
0.084	20	7	0.247	0.176	FALSE	0.751
0.094	161	7	0.374	0.302	FALSE	0.752
0.103	62	5	0.649	0.526	FALSE	0.753

hip hop	pop	rap	rock
0.5761511	0.6910798	0.6317907	0.8098720

IV: Conclusion

Generally, more advanced methods like LightGBM tended to outperform simpler methods like Multinomial Logistic Regresion on the public test set for this task. However, significant gains to performance were realised in the process of feature construction (such as the profanity features), alternative word embeddings, feature engineering/transformations, and more generally focusing on the quality and form of the data. Improvements were also made through hyperparameter tuning—whether through grid or random search—refining those models which already exhibited promising performance. Also unsurprisingly given the initial exploration of the training data, the validation set and CV errors presented for each classifier reveal that the classifiers performed best at distinguishing pop and rock, and struggled more with performance on the hip hop and rap categories (likely because they are difficult to differentiate from one another).

Bibliography

Bandhi, N. (n.d.). Music Genre Classification of Lyrics using LSTM. Retrieved May 17, 2022, from https://nbandhi.medium.com/music-genre-classification-of-lyrics-using-lstm-f5c762a1b3d

Boonyanit, A., & Dahl, A. (n.d.). Music Genre Classification using Song Lyrics. https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1214/reports/final_reports/report003.pdf

Jewalikar, V. and Fragapane, F. (2015), 'Hip hop lyrics are the most profane'. https://lab.musixmatch.com/profanity_genres/

Kumar, A., Rajpal, A., & Rathore, D. (2018). Genre Classification using Word Embeddings and Deep Learning. 2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2142–2146. https://doi.org/10.1109/ICACCI.2018.8554816

Rajanna, A. R., Aryafar, K., Shokoufandeh, A., & Ptucha, R. (2015). Deep Neural Networks: A Case Study for Music Genre Classification. 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), 655–660. https://doi.org/10.1109/ICMLA.2015.160

Thompson, C. (2021). Lyric-Based Classification of Music Genres Using Hand-Crafted Features. Reinvention: An International Journal of Undergraduate Research, 14(2), Article 2. https://doi.org/10.31273/reinvention.v14i2.705