[CS156] Final Project Report

December 18, 2020

Spotify Skip Prediction Challenge
Report

Taha Bouhoun

1 Introduction

In early 2019, Spotify shared exciting statistics about their platform. Out of 35+ million songs on the service, Spotify users created over 2+ billion playlists (Oskar Stål, 2019). I thought of the analogy that our music taste is like our DNA, very diverse across 7 billion people, yet the building blocks (nucleotides/songs) are the same. As a result, inferring a user's music taste is challenging, mostly since Spotify's business model relies on its ability to recommend new songs.

Like all entertainment services, Spotify is batteling for its user's attention, making it necessary to recommend songs that are less likely to be skipped. This project explores a portion of the 130 million streamings that Spotify shared as part of the Skip Prediction Challenge. I aim to experiment with some machine learning models and report their performance.

2 Dataset Descriptions

The data contains a history of 130 million streamings of roughly 4 million unique songs. Spotify didn't share details on how many users are represented in the dataset. However, they included the following features: - Characteristics of the user: Details on the user's activity in the platform (e.g., subscription, the position of the song in a streaming session, etc.) - Song features: ranging from duration and popularity estimate in the US to audio breakdown of the track (e.g., tempo, acoustics, instrumentals, etc.)

The output of interest is a binary indicator of whether the user skipped the song or not.

3 Assumptions

A crucial step in modeling is to lay out all the assumptions and limitations in order to properly interpret the result. Some assumptions are due to the data collection process and others are part of the modeling process:

- The users are homogenous, i.e., the mechanism that leads a user to skip a song is static across the population regardless of their music taste.
- Songs are broken down into audio features hencec the lyrics are not interpreted as natural language text. This limitation is important to consider since lyrical meaning can be a strong predictor of song skipping.

4 Modeling

4.1 Classification based on Audio Features

The first modeling step is to only include songs' audio features are predictors of whether the song would be skipped or not. The idea is that we want to test the predictive accuracy of a model that assumes homogeneity across all Spotify users, in other words, if a song is skipped then it's mainly due to its audio features.

4.1.1 XGBoost: Discussing the Results

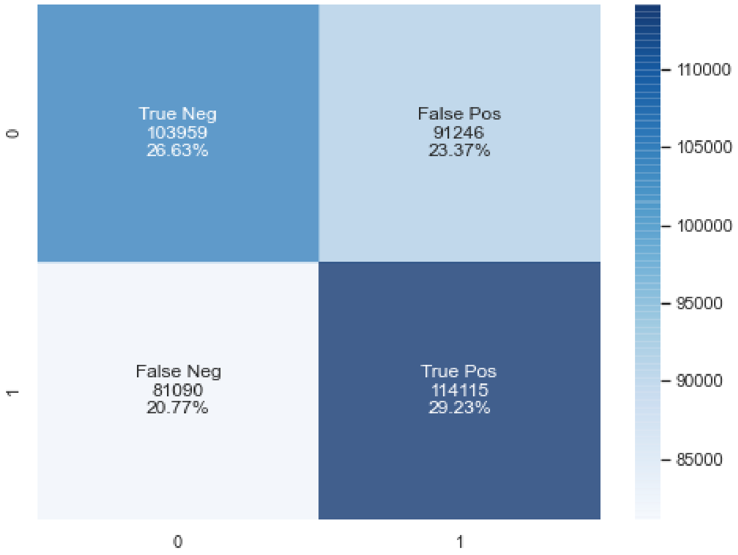
The classification output suggests that the target variable is imbalanced. Thus the model favors making a 'Negative' classification, representing roughly 65% of the target variable. We ended up with a confusion matrix that barely captures any true positives. It seems that the machine is not learning much.

4.2 Classification using Audio and User features

In this part, I dealt with the imbalance in the output variable by randomly sampling without replacement from the majority class. The goal is to have roughly the same number of the dependent variable in both train and test sets. Furthermore, I added user features to the prediction input, such as premium subscription and shuffle mode. Perhaps some users' behavior would add some predictive power to whether they would skip a song.

4.2.1 LightGBM: Discussing the Results

Even after balancing the output variable in the dataset and applying feature engineering on song recency, the classification's output still hovers around 56%. Upon using Bayesian optimization for hyper-parameter tuning, the accuracy improved by 3% to reach 58,76%. These results don't reflect a successful model despite not over-fitting. Many reasons can be attributed to these poor results. Still, I can test if Spotify listeners' diversity makes it hard to infer whether a song would be skipped. In the upcoming section, I experiment with my own Spotify streaming history.



iter	target	baggin...	colsam...	learni...	max_depth	min_ch...	n_esti...	reg_alpha
1	0.5801	0.8749	0.9556	0.7323	19.97	4.744	23.94	0.8662
2	0.572	0.9202	0.7373	0.02156	29.25	20.98	53.64	0.1834
3	0.5847	0.8608	0.5723	0.4325	12.28	15.68	80.11	0.3664
4	0.5876	0.8912	0.8067	0.2005	17.86	15.22	155.8	0.1705
5	0.5788	0.813	0.954	0.9657	25.21	8.311	174.2	0.4402
6	0.5786	0.8871	0.1662	0.4989	17.36	15.12	157.1	0.8782
7	0.5802	0.9002	0.8857	0.8992	6.991	20.99	164.4	0.1152
8	0.5864	0.8273	0.7063	0.5117	5.656	2.333	243.1	0.7896
9	0.5772	0.9866	0.9057	0.9924	17.14	15.02	156.0	0.9237
10	0.578	0.8523	0.838	0.9795	8.781	21.24	166.4	0.9066

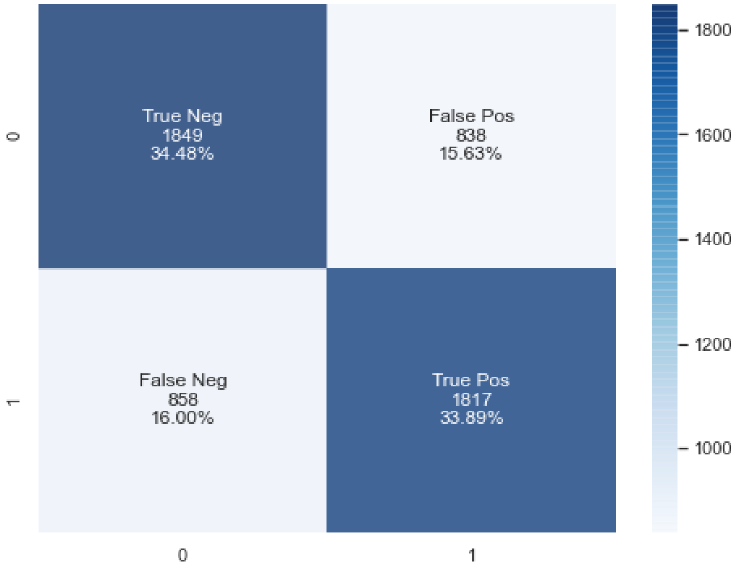
4.3 Application on my Spotify Data

Narrowing the analysis to one user might help us disentangle better predictive power from classification models. In this section, I used my entire streaming history on Spotify in 2018 to predict whether I would skip a song:

- I requested my data (it took Spotify 24 hours to email it), then I used Spotify API to retrieve the track IDs and their audio
- I compute the duration gap between the length of the song and how long I played it.
- I use LightGBM binary classification to infer my song-skipping habits based solely on audio features.

4.3.1 Discussing the Results

The model performs better with personalized data with an accuracy of 74.17% (28th iteration of Bayesian Optimization). The assumption that Spotify users are homogeneous is a strong one, and the performance can be improved if we gather more user level details.



iter	target	baggin...	colsam...	learni...	max_depth	min_ch...	n_esti...	reg_alpha
1	0.7289	0.8749	0.9556	0.7323	19.97	4.744	23.94	0.8662
2	0.6944	0.9202	0.7373	0.02156	29.25	20.98	53.64	0.1834
3	0.7356	0.8608	0.5723	0.4325	12.28	15.68	80.11	0.3664
4	0.7372	0.8912	0.8067	0.2005	17.86	15.22	155.8	0.1705
5	0.7285	0.813	0.954	0.9657	25.21	8.311	174.2	0.4402
6	0.7379	0.9144	0.8335	0.5383	26.8	20.24	161.2	0.007412
7	0.7367	0.9239	0.2602	0.5006	28.29	23.33	133.9	0.1984
8	0.7364	0.8663	0.8002	0.121	25.65	21.26	161.7	0.7487
9	0.7288	0.8735	0.2552	1.0	5.0	8.948	107.9	0.0
10	0.7389	1.0	0.1	0.6267	30.0	1.0	141.8	0.0
11	0.7396	0.8454	0.3199	0.3481	6.432	13.87	249.9	0.5242

iter	target	baggin...	colsam...	learni...	max_depth	min_ch...	n_esti...	reg_alpha
12	0.7314	0.8993	0.9595	0.8483	29.93	1.76	237.8	0.1786
13	0.7284	1.0	0.1	1.0	5.0	25.0	225.7	0.0
14	0.7266	1.0	0.1	1.0	30.0	25.0	102.4	0.2
15	0.7288	0.8269	0.8713	0.9709	26.76	24.25	248.3	0.831
16	0.7293	0.82	0.482	1.0	5.0	1.0	139.2	0.0
17	0.729	0.9387	0.8074	0.7696	6.789	1.506	230.6	0.9314
18	0.6972	0.8562	0.7732	0.7428	6.364	24.79	10.32	0.8157
19	0.7318	0.8	1.0	1.0	5.0	1.0	81.67	1.0
20	0.7357	1.0	1.0	0.6155	30.0	25.0	205.0	0.0
21	0.6887	1.0	0.1	0.001	30.0	1.0	207.0	0.0
22	0.6567	0.8	1.0	0.001	30.0	1.0	119.5	1.0
23	0.6882	1.0	0.1	0.001	30.0	2.758	155.8	1.0
24	0.737	0.8494	0.9136	0.116	23.91	23.67	151.4	0.01489
25	0.7268	0.9446	0.3283	1.0	22.31	14.8	141.7	0.0
26	0.7373	0.8621	0.5178	0.5964	15.47	10.34	243.2	0.4136
27	0.6667	1.0	0.9308	0.001	9.934	22.29	242.0	0.6607
28	**0.7417**	**0.8318**	**0.6385**	**0.2465**	**11.57**	**8.097**	**249.9**	**0.9841**
29	0.739	0.9623	0.9428	0.5138	5.164	5.273	243.8	0.439
30	0.7251	0.8	0.1	1.0	15.67	1.0	243.0	0.0

5 Conclusion

The Spotify competition was an interesting challenge to explore. As of today, more than 30K looked up the dataset, and roughly 700 submissions have been attempted. Neural Network might be a viable alternative, but the dataset's size requires a huge computational cost. Yet, I've tried to build an RNN and work with a subset of the data.

Overall, recommendation engines require both personalized learning about the user and general learning about the songs. In this project, I experimented with machine learning classification using only audio features, audio and user features, and my personal listening history. A further investigation might include the causal relationships between the covariates because perhaps understanding the mechanism by which the data is generated is more informative than curve-fitting.

6 Appendix

- Dataset Citation: Brost, Brian and Mehrotra, Rishabh and Jehan, Tristan (2019). The Music Streaming Sessions Dataset. Proceedings of the 2019 Web Conference. ACM. Retrieved from: https://www.aicrowd.com/challenges/spotify-sequential-skip-prediction-challenge
- HC/LO Applications:
  - **#HC-testability**: Drawing from the conclusion that the user's diversity worsen the predictive power of the model, I tested this claim on my own Spotify streaming and had better performance.
  - **#HC-sampling**: To solve the problem of imbalanced output data, I took a random sample of the majority class that has the same size as the minority class. This technique os known as "Under-sampling" and is appropriate for large imbalanced datasets.
  - **#HC-gapanalysis**: Throughout the report, I developed hypothesis about the poor performance of the model then explicitly stated the changes that I employed (feature engineering, balancing data, tuning hyper-parameters, etc) to bridge the gap between merely random classification (50%) to a somewhat decent classifier (74.2%)

7 References

- Oskar Stål (2019). Music Recommendations at Spotify. Nordic Data Science and Machine Learning Summit. Retrieved from: https://youtu.be/2VvM98flwq0
- Brian Brost, Rishabh Mehrotra, and Tristan Jehan. 2019. The Music Streaming Sessions Dataset. In Proceedings of the 2019 World Wide Web Conference (WWW '19), May 13–17, 2019, San Francisco, CA, USA. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3308558.3313641