Predicting a song's genre using the Spotify Top Hits from 2000-2019 Dataset

Supervised Learning

Artificial Inteligence

Group
09_1B

Problem Specification

The problem can be defined as a Multilabel Classification Problem - each song can be attributed with more than one genre. Using Tom Mitchell's machine learning formalism:

Task ***(T)*** Classify a song's genre using listeneable and computed parameters

Experience ***(E)*** The parametrized Spotify Top Hits from 2000 to 2019 (Spotify API fetched data for a broader experience, if needed)

Performance ***(P)*** Classification accuracy, which is the number of correctly labeled genres, nuanced by the number of incorrectly labeled genres

Data Preprocessing

Looking at the training dataset, some assumptions can be made as to what can impact or not the predictive model.

artist	song	duration_ms	explicit	year	popularity	danceability	liveness	valence	tempo	genre
Taylor Swift	Bad Blood	211933	FALSE	2014	54	0.646	0.201	0.287	170.216	pop
Taylor Swift	Bad Blood	200106	FALSE	2015	70	0.654	0.139	0.221	170.16	pop
Cardi B	Bodak Yellow	223962	TRUE	2017	59	0.929	0.346	0.458	125.022	hip hop, pop
Cardi B	Bodak Yellow	223712	TRUE	2018	72	0.926	0.231	0.485	125.022	hip hop, pop

Duplicates

Remove *Artist/Song* **parameter**

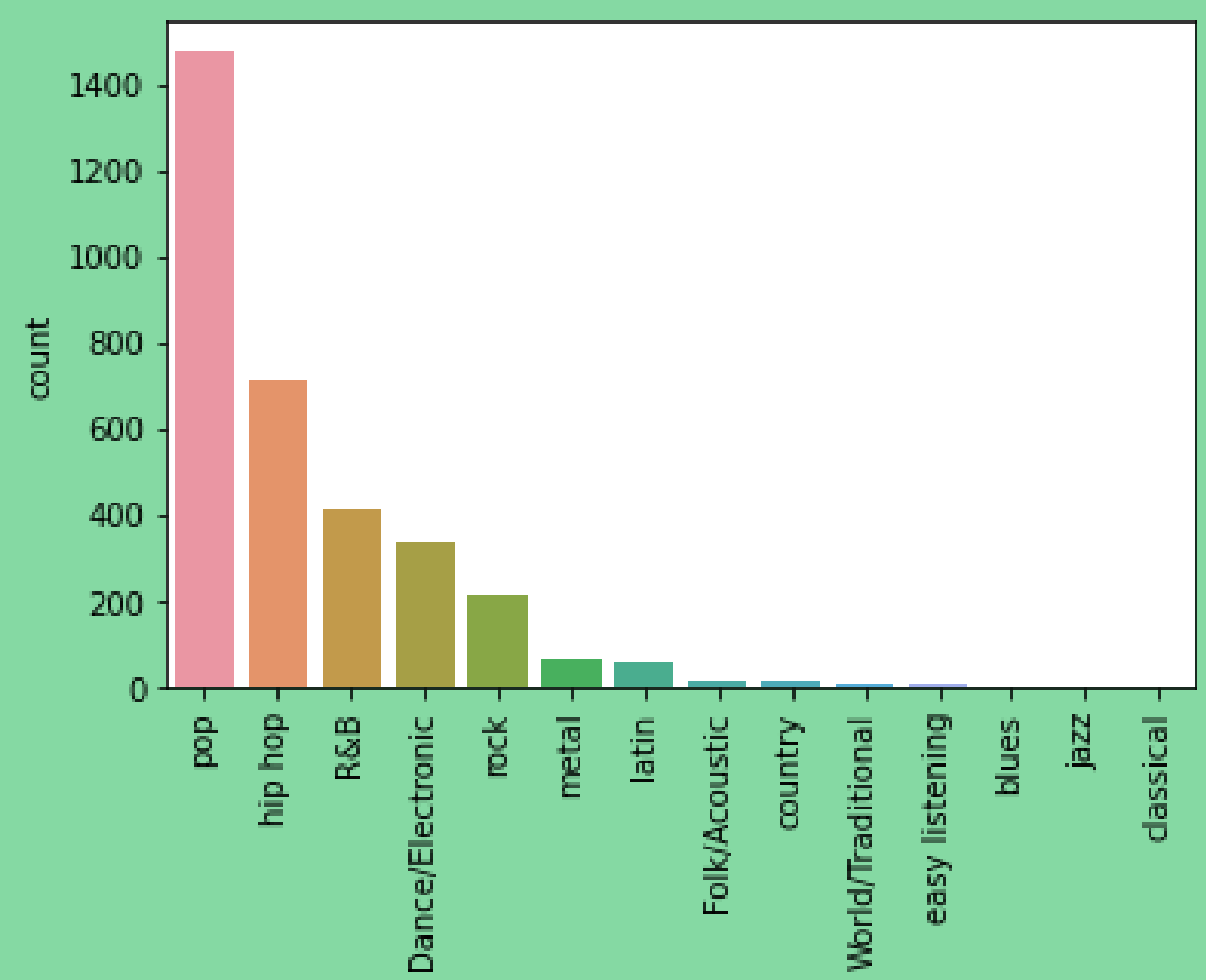
Remove *Empty set "genre"*

Remove *popularity* **parameter**

Only consider genres with
more than 50 songs (7 genres)

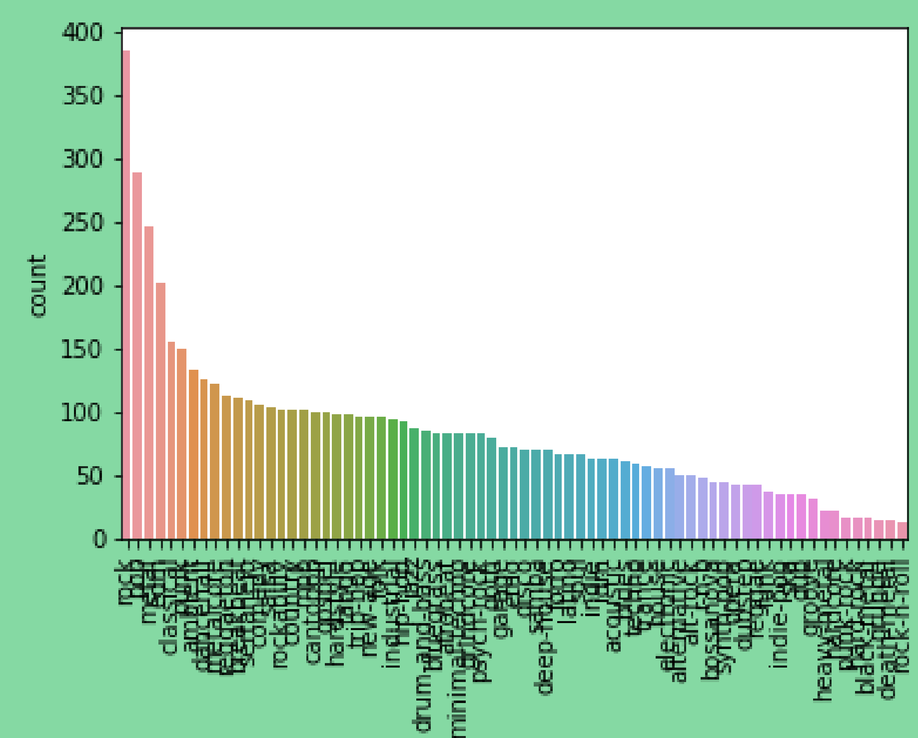
Fetch data from Spotify API
to balance the dataset

Boolean parameter
to **Binary Integer** *conversion*

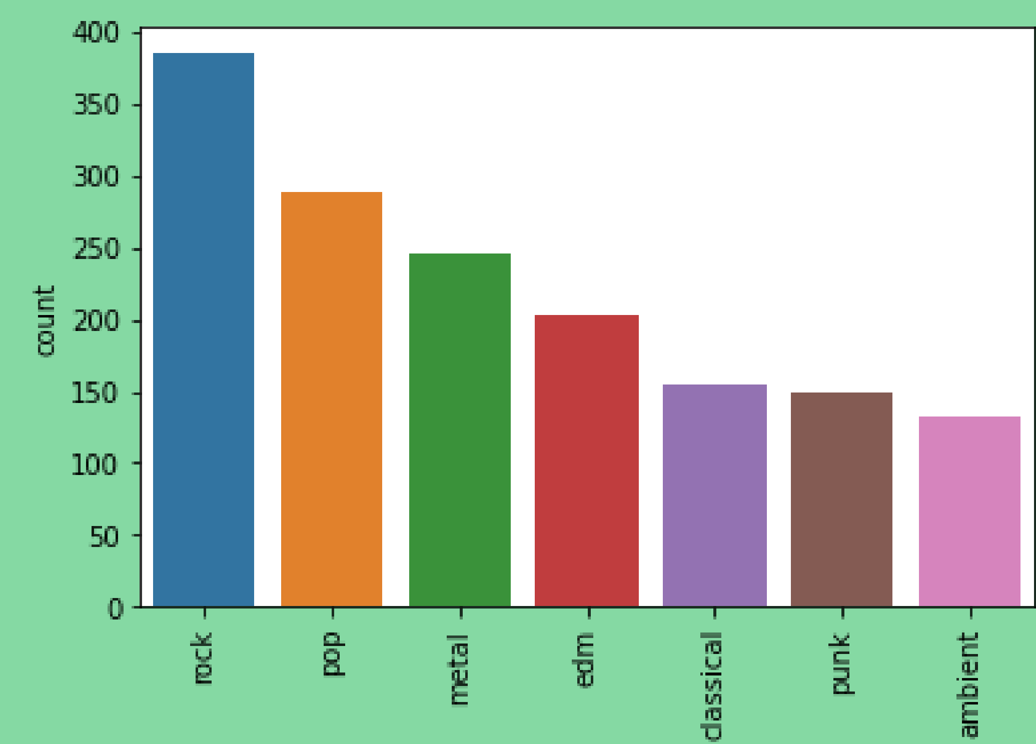


Data Preprocessing

We noticed two main things about the dataset: it was small and it was extremely **skewed to one genre** - *pop*. The disparity was so big that it was **not possible to undersample** without losing most data **nor oversample** without repeating much of it. As such, we used the Spotify Web API to fetch data and generate a new dataset:



genres with more than
130 songs (7 genres)



While still not completely uniform, it was a great improvement as most genres had a reasonable sample of songs for the prediction model.

Tools and Algorithms used

Since this is a Multilabel Classification problem, it is necessary to encode the array of genres into multiple columns. We opted to create a column for each genre where the value represents whether the song has that genre.

Used estimators with built-in support for multilabelling

These models can be used for multilabelling directly:

DecisionTreeClassifier *RandomForestClassifier*
MLPClassifier

Estimators wrapped in MultiOutputClassifier

These models were wrapped using scikit-learn's MultiOutputClassifier, which creates a duplicate of the model for each label to predict:

LogisticRegression *GradientBoostingClassifier*
SVC *GaussianNB* *AdaBoostClassifier*

genre
pop
rock, pop
pop, country
rock, metal
pop

→

pop	metal	rock	country
1	0	0	0
1	0	1	0
1	0	0	1
0	1	1	0
1	0	0	0

Tools and Algorithms used

Train/Test split

Datasets → Training test (80%)

Datasets → Testing set (20%)

The training set was used for **fitting** and **hyper-parameter tuning**.
The testing set was used for **validating** the results.

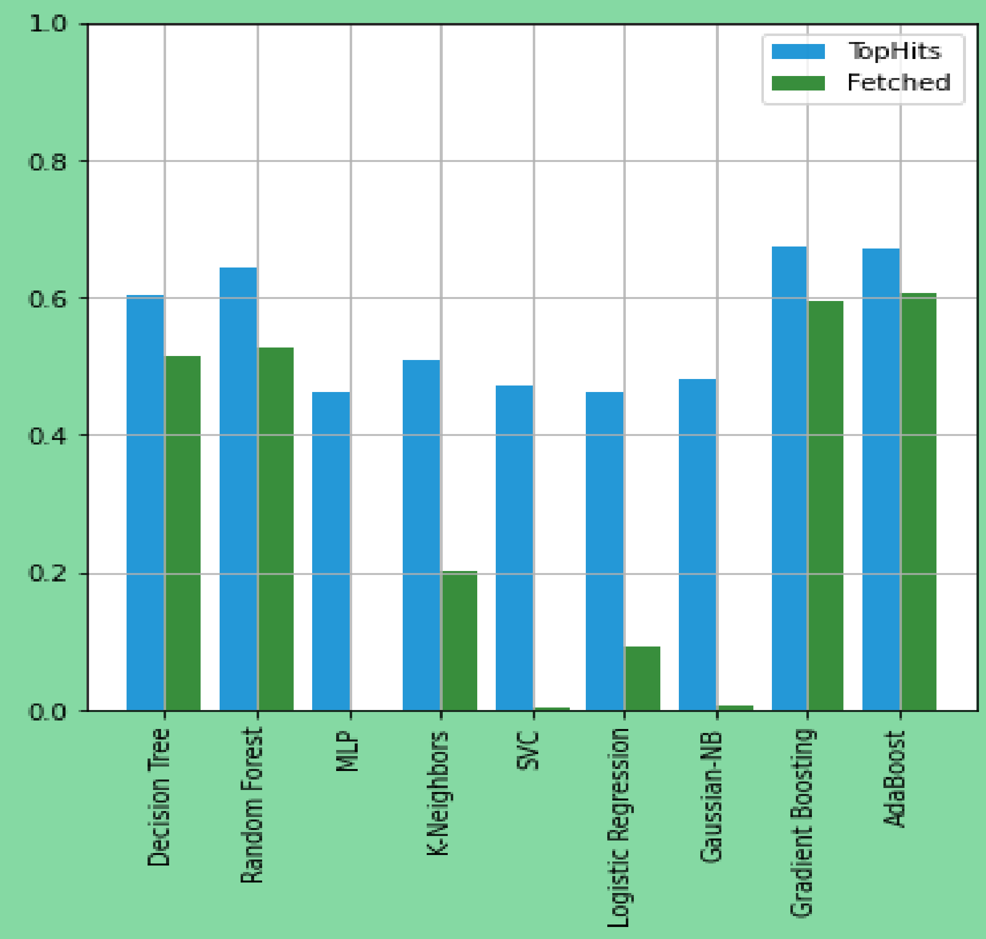
Hyper-parameter tuning

We tried different hyper-parameters by wrapping the estimators in a *RandomizedSearchCV* with **5-fold cross validation** and using **accuracy as the scoring metric**.

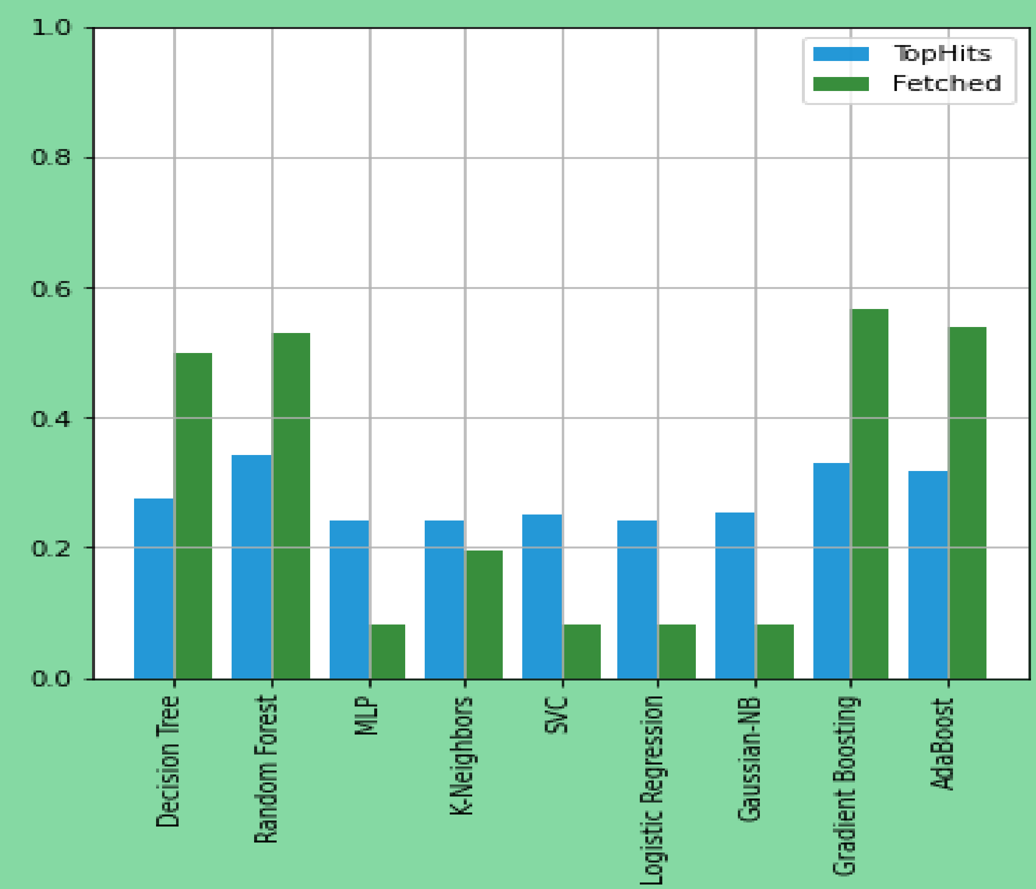
We chose the accuracy metric since it was the one producing the best results in the initial experiments with the models. However, we also considered the possibility of using f1-measure or precision.

Results

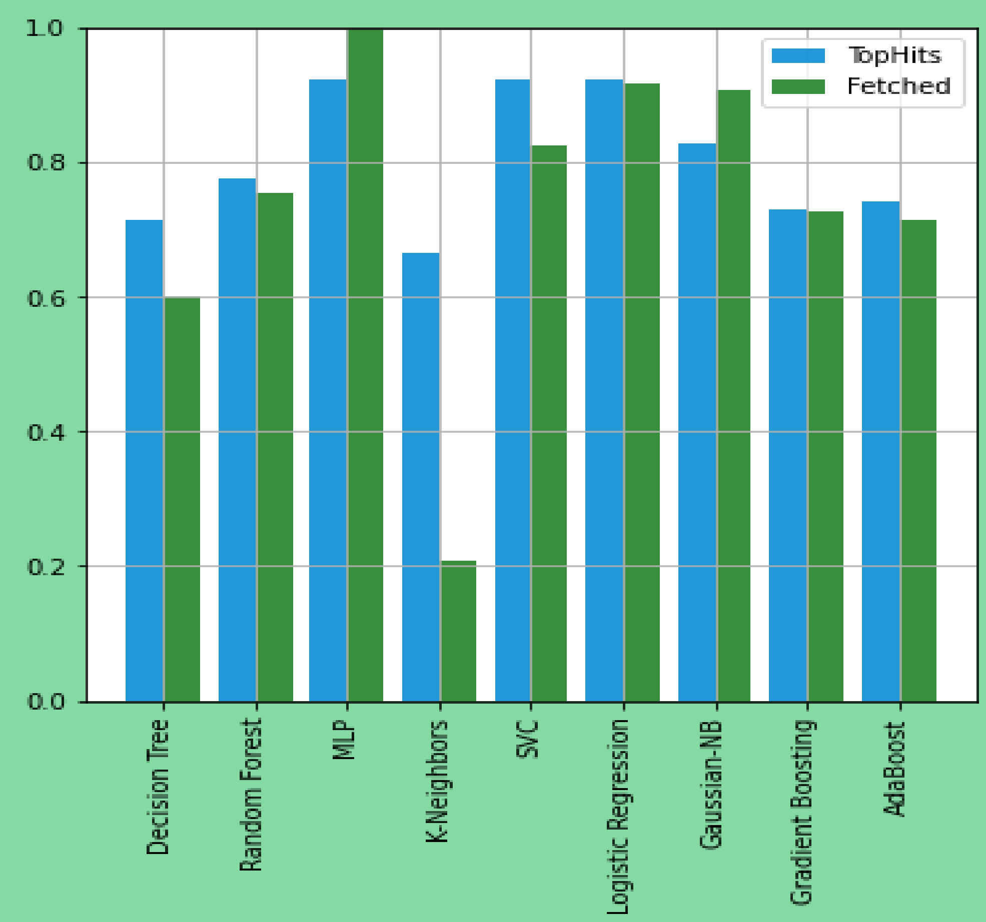
RECALL



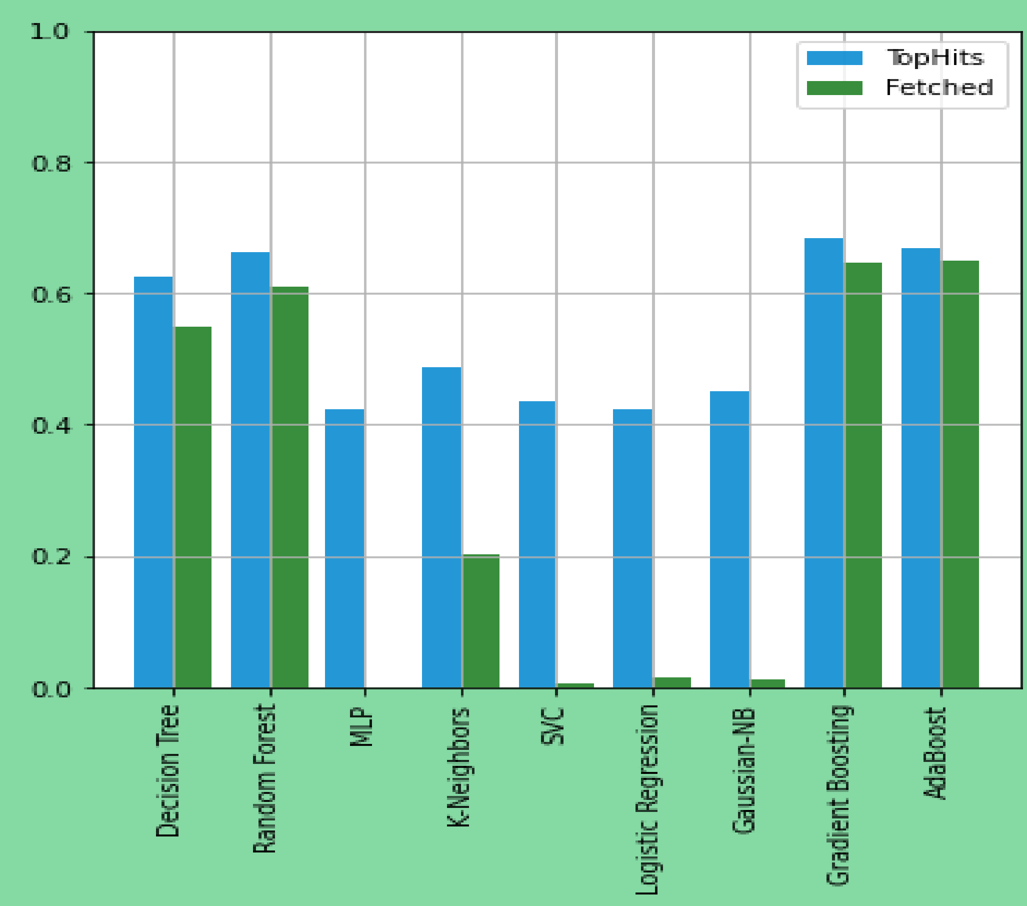
ACCURACY



PRECISION

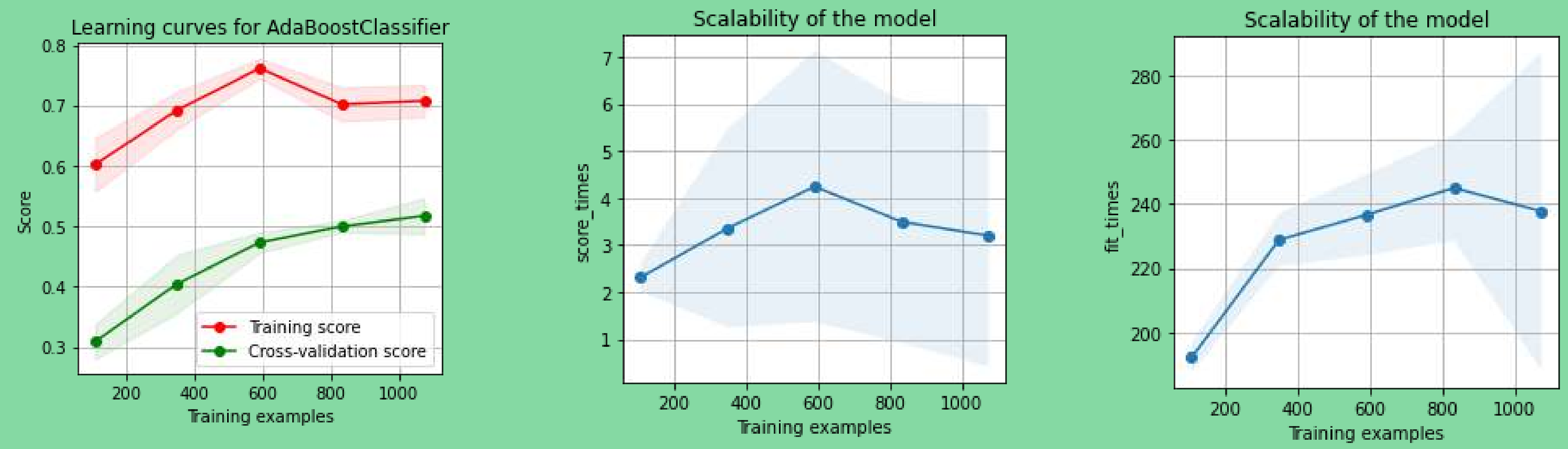


F1-SCORE

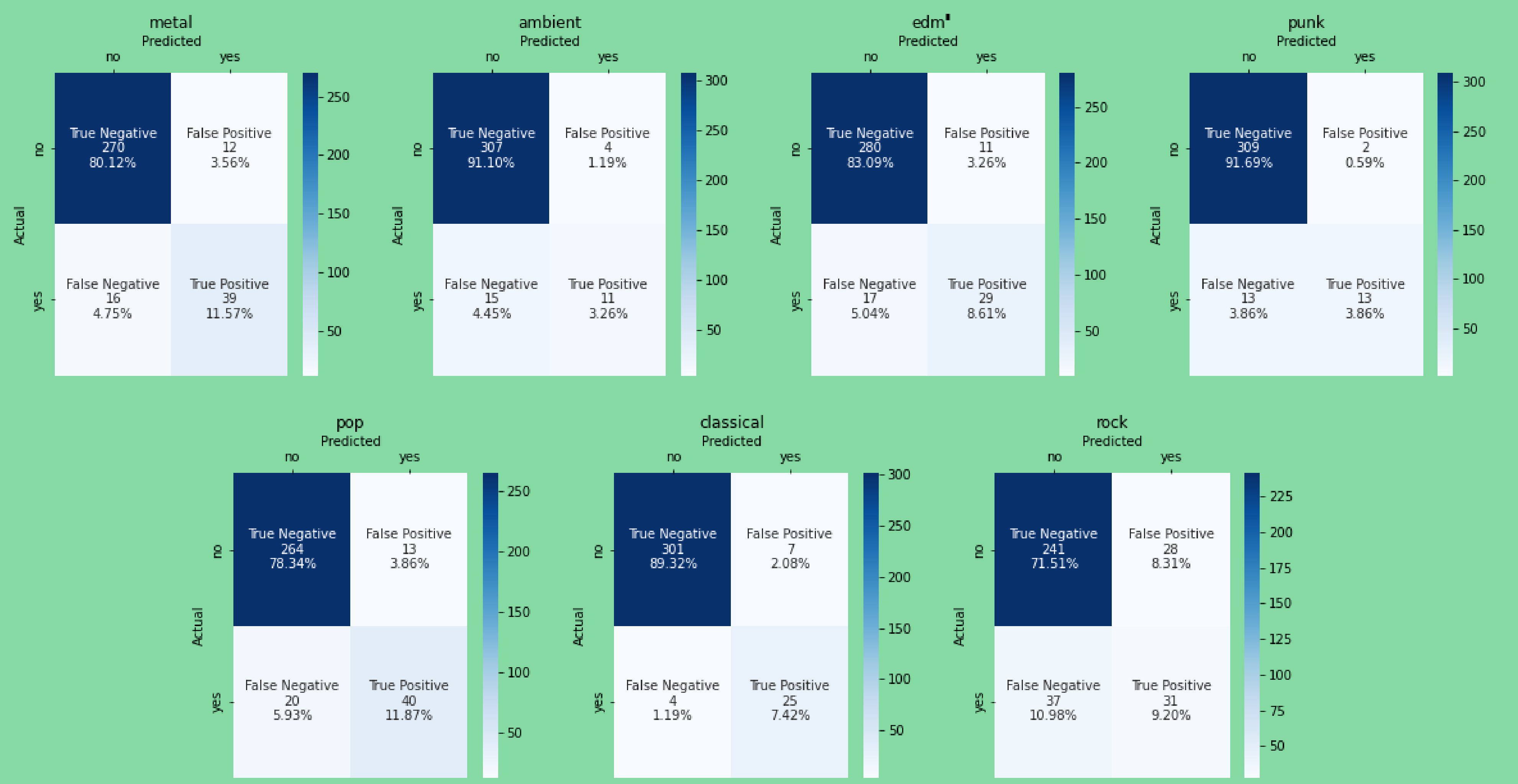


Results

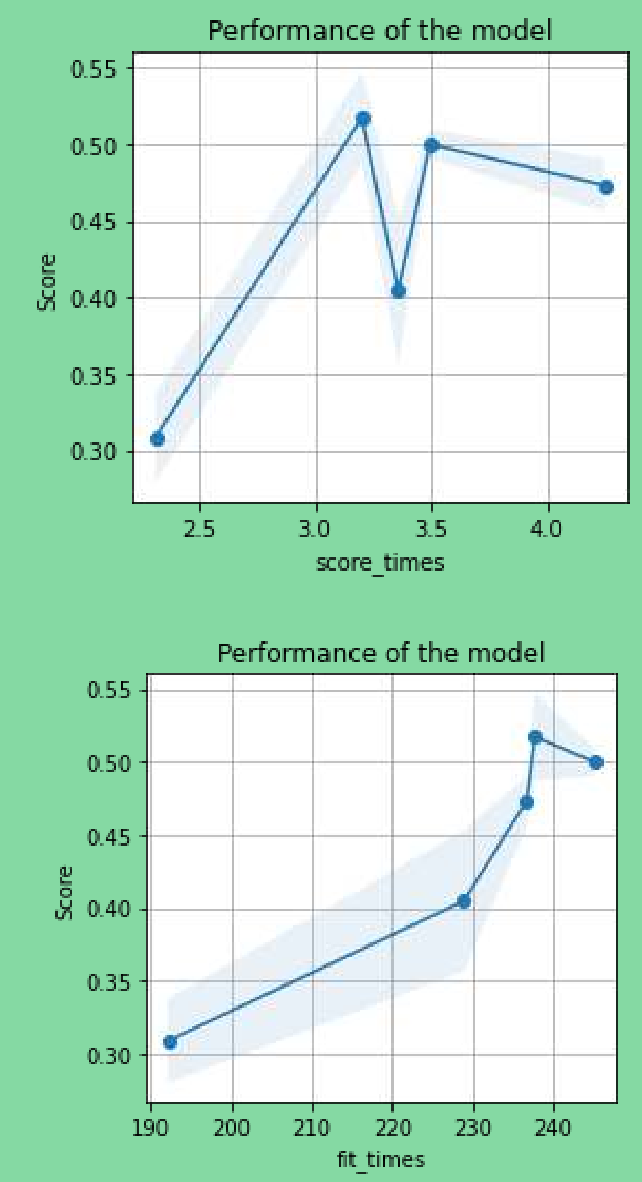
LEARNING CURVE AND SCALABILITY



CONFUSION MATRICES



Ada Boost Results



PERFORMANCE

Conclusions

Small and imbalanced dataset is likely the cause of the low performance in some of the models.

- Original **dataset is biased** towards popular songs in general (formulaic, low variety) and certain genres, namely, pop and rock.
- Hard to overcome: original dataset is **too small and imbalanced** for overfitting/underfitting.
- Multilabeling is **not supported on *imbalance-learn***.
- The nature of the problem leads to imbalance, even on the new dataset: it's a multilabel problem where **most songs only have one classification**. This implies that, for each column, the count of "No" is much bigger than "Yes".
- Multilabel is much more **data-hungry**: N labels lead to 2^N possible classifications.

As usual, **data is king**. The best and only impactful solution was **expanding and strengthening the dataset**, which we did and **proved promising**, as showed. Limited query size on Spotify's end handicapped this measure, yet it still yielded better results.

References

Spotify API
https://spotipy.readthedocs.io/

Scikit multilabeling
https://scikit-learn.org/stable/modules/multiclass.html

Classifier chains for multi-label classification [Read, 2011]
https://link.springer.com/article/10.1007/s10994-011-5256-5

How to Define Your Machine Learning Problem [Brownlee, 2013]
https://machinelearningmastery.com/how-to-define-your-machine-learning-problem/

Code for plotting learning curves adapted from:
https://scikit-learn.org/stable/auto_examples/model_selection/plot_learning_curve.html

The following sources were used as a guideline for the hyperparameters to use:

https://ai.plainenglish.io/hyperparameter-tuning-of-decision-tree-classifier-using-gridsearchcv-2a6ebcaffeda
https://www.kaggle.com/code/sociopath00/random-forest-using-gridsearchcv/notebook
https://datascience.stackexchange.com/questions/19768/how-to-implement-pythons-mlpclassifier-with-gridsearchcv
https://www.ritchieng.com/machine-learning-efficiently-search-tuning-param/
https://www.vebuso.com/2020/03/svm-hyperparameter-tuning-using-gridsearchcv/
https://machinelearningmastery.com/hyperparameter-optimization-with-random-search-and-grid-search/
https://stackoverflow.com/questions/39828535/how-to-tune-gaussiannb