Spotify Skip Prediction

Kregg Jackson

Business Context

- Spotify is an online music streaming service and one of their core challenges is to recommend the right music to each user
- This project is designed to predict whether or not a user will skip the next recommended song
- If executed properly the system will be able to lower the current skip rate

Spotify Music Streaming Sessions Data

- The dataset originally consists of two data frames the session data and track features data
- The session data frame had about 130 million listening sessions with the user interactions on Spotify
- I had to scale the dataset down to just one listening session data frame that consisted of almost 2.9 million sessions

Spotify Music Streaming Sessions Data

The session data frame includes some key columns such as:

- ‘not_skipped’
- ‘hist_user_behavior_start’
- ‘no_pause_before_play’

The track features data set has track info like:

- ‘key’
- ‘release_year’
- ‘danceability’

Dependent Variable

- This dataset has three different skip variations:
  - ‘skip_1’
  - ‘skip_2’
  - ‘skip_3’
- One of the biggest challenges for classification models is an imbalance of classes
- `skip_2` is very close to being a 50-50 split on true or false data points

How is success determined?

- I will be using precision, recall, and the f1-score metrics in unison
- Precision is highly sensitive to false positives
- Recall is very sensitive to false negatives
- f1-score is a mean of recall and precision

Process Steps

My next steps were to:

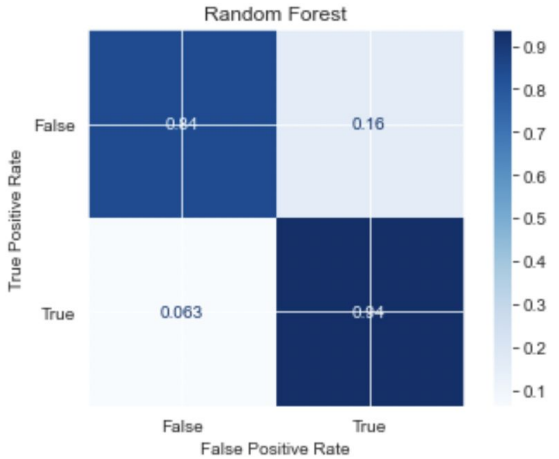
- Build and tune the binary classifiers
- Find the best parameters for the precision, recall, and f1-score

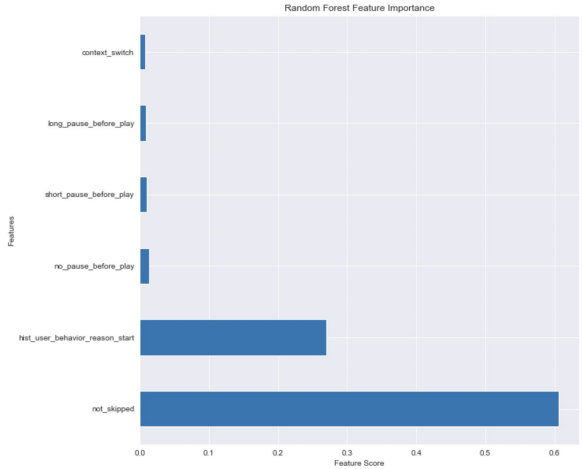
The best performing classifiers were:

- Random Forest
- Gradient Boosting

Random Forest Results

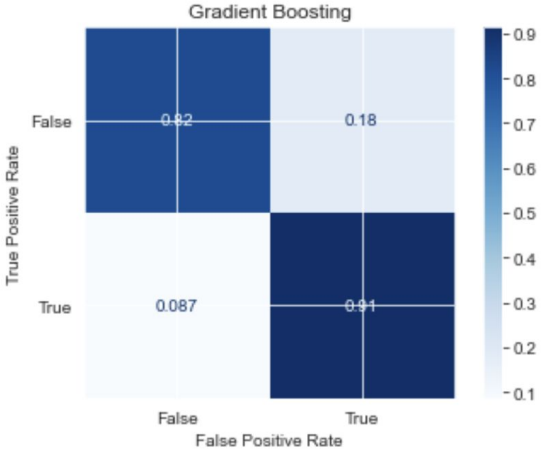
- I got the best results when predicting no skip with the grid searched random forest classifier
- This model had the highest precision, recall, and f1-scores out of the whole project
- The features with the highest feature importance score for the random forest classifier are :
    - ‘not_skipped’
    - ‘hist_user_behavior_start’
    - ‘no_pause_before_play’

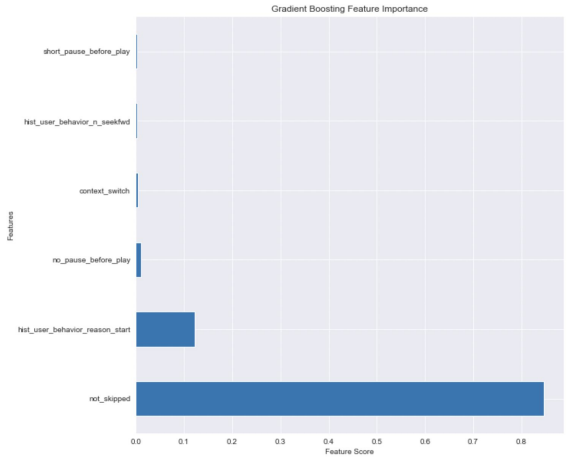




Gradient Boosting Results

- The Gradient Boosting classifier had the second highest precision, recall, and f1-scores out of the whole project
- The classifier is fit well to the data and I got very similar performance on the training set and the test set
- The features with the highest feature importance score for the gradient boosting classifier are also:
    - ‘not_skipped’
    - ‘hist_user_behavior_start’
    - ‘no_pause_before_play’





Conclusive Evaluation

- Random forest was the best performing classifier and the most important features are ‘not_skipped’, ‘hist_user_behavior_start’, and ‘no_pause_before_play’
- The classifier is saying the most important features deal with user behavior much more than the track features
- I recommend Spotify evaluate when a user is skipping a lot and either play songs already in the user's library or play something very different than the recently skipped tracks

Future Improvements

- Host the project online instead of Jupyter Notebook in order to be able to utilize the entire dataset
- Run a broader grid search on the random forest to see if there are more optimal hyperparameters
- Utilize a sklearn scaler like MinMaxScaler() to optimize the scale and distribution of the data

Thank You

Email: kreggthegoat@gmail.com

Github: @kreggthegoat