Popular Song Predictor

By: Johnny Koponen

Motivation / Importance

- Last year when my spotify wrapped released I had hundreds of plays on the same songs I had the previous year. (2021 and 2020 were relatively similar)
- As an avid listener during the year, I always try and find new music to listen to.
- Seeing that both my 2020 and 2021 spotify wraps were very similar, I wanted to branch out to new music.
- Therefore, using my machine learning skills I thought it would be fascinating to be able to predict whether a new releasing song would be popular or not.

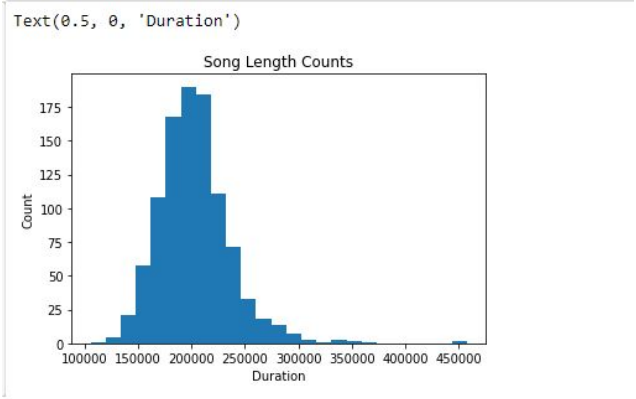
Approach

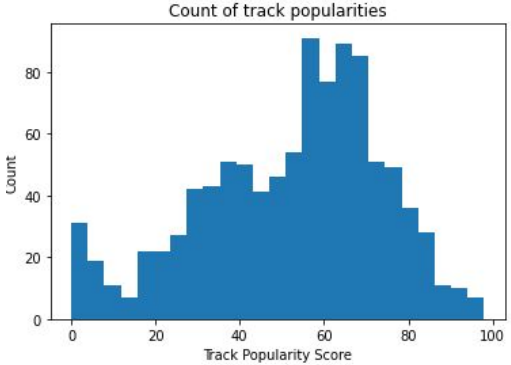
- Initially was planning on creating my own “score” column for the data by assigning values to categorical data.
- With this score, I was going to predict whether a song would be popular or not.
- Was going to utilize a binary classification.
- After receiving comments from Professor Iyer, I changed my approach.
- Got a new dataset (recommended by him) that had a track_popularity score.
- Decided I would now try and predict that track_popularity score with my algorithms.

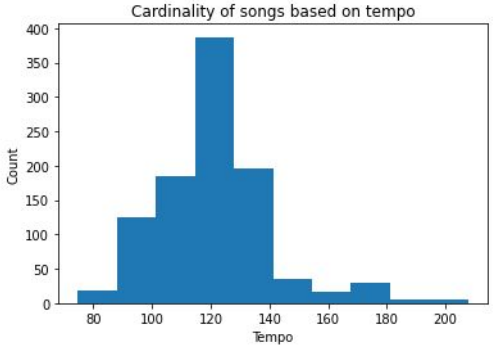
Approach

During my Data Exploration phase, I noticed:

- Most of the songs in the dataset are 3 minutes.
- Track_popularity values (0-100) were skewed to the left (50-60).
- Most apparent tone throughout the dataset was a C.
- Most songs were 120bpm in the dataset (relatively upbeat).
- Some columns were unnecessary, so I also removed those columns.
    - Song name, artist, track_album_id, etc







Results and Findings

- After filtering and cleaning my data, my models outputted very low accuracy %'s and the r^2 values were very irregular.
- I decided to add a new column to my dataset.
    - This new column was named ['Target']
        - If a track_popularity is >=50, the target row will equal 1
        - Otherwise, the row will = 0.
            - I now decided that my model will predict the 'Target' value rather than the track_popularity score.

Results and Findings (First Attempt)

- When looking at the data presented to the right, it is apparent that the models are not responding well to predicting the track_popularity score.

```
Linear Regression: 7.20%
Mean Absolute Error =  20.180292886898823
Mean Squared Error =  582.0521935359664
RMSE =  24.125757885214018
r^2 =  0.07199004365983208

Logistic Regression: 7.97%
Mean Absolute Error =  41.322113920194944
Mean Squared Error =  2339.5834602497716
RMSE =  48.36924084839219
r^2 =  -2.730175350101844

K-Nearest Neighbors: 10.95%
Mean Absolute Error =  28.227078891257996
Mean Squared Error =  1291.743679561377
RMSE =  35.94083582168585
r^2 =  -1.0595249171556769

Decision Tree: 21.50%
Mean Absolute Error =  20.81449893390192
Mean Squared Error =  855.4876637222053
RMSE =  29.24872071941276
r^2 =  -0.36396886443713594

Support Vector Machine (Linear Kernel): 8.03%
Mean Absolute Error =  40.605848309473046
Mean Squared Error =  2282.6859579652755
RMSE =  47.77746286655744
r^2 =  -2.6394593469713845

Support Vector Machine (RBF Kernel): 8.53%
Mean Absolute Error =  41.86765153822723
Mean Squared Error =  2387.4287237282974
RMSE =  48.86132134652416
r^2 =  -2.806458682361199

Neural Network: 9.38%
Mean Absolute Error =  33.05741699664941
Mean Squared Error =  1687.878921717941
RMSE =  41.08380364228635
r^2 =  -1.691113377539684

Random Forest: 25.18%
Mean Absolute Error =  24.238044471519952
Mean Squared Error =  1124.1524520255864
RMSE =  33.52838278273479
r^2 =  -0.7923215125887055

Gradient Boosting: 15.82%
Mean Absolute Error =  30.78632348461773
Mean Squared Error =  1540.7381967712458
RMSE =  39.252238111619135
r^2 =  -1.4565157602639651
```

Results and Findings (Second Attempt)

- After adding the 'Target' column to my dataset, I retrained my models and ran my program.
- The new accuracy scores were significantly higher ~60-65% accuracy, and the r^2 values were no longer as irregular as before.

```
Linear Regression: 4.80%
Mean Absolute Error =  0.47055578762812056
Mean Squared Error =  0.23502630833951574
RMSE =  0.48479511996256286
r^2 =  0.047998019946730364

Logistic Regression: 59.76%
Mean Absolute Error =  0.4023758757234237
Mean Squared Error =  0.4023758757234237
RMSE =  0.6343310458454826
r^2 =  -0.6298712817332806

K-Nearest Neighbors: 57.86%
Mean Absolute Error =  0.42141334145598536
Mean Squared Error =  0.42141334145598536
RMSE =  0.6491635706476337
r^2 =  -0.7069847980908355

Decision Tree: 57.49%
Mean Absolute Error =  0.4250685348766372
Mean Squared Error =  0.4250685348766372
RMSE =  0.6519728022522391
r^2 =  -0.7217905932314861

Support Vector Machine (Linear Kernel): 59.70%
Mean Absolute Error =  0.40298507462686567
Mean Squared Error =  0.40298507462686567
RMSE =  0.6348110542727384
r^2 =  -0.6323389142567224
```

```
Support Vector Machine (RBF Kernel): 60.89%
Mean Absolute Error =  0.3911056960097472
Mean Squared Error =  0.3911056960097472
RMSE =  0.6253844385733844
r^2 =  -0.584220080049608

Neural Network: 59.47%
Mean Absolute Error =  0.4052695705147731
Mean Squared Error =  0.4052695705147731
RMSE =  0.6366078624355601
r^2 =  -0.6415925362196289

Random Forest: 63.40%
Mean Absolute Error =  0.3659762412427658
Mean Squared Error =  0.3659762412427658
RMSE =  0.604959702164339
r^2 =  -0.48243023845763555

Gradient Boosting: 61.21%
Mean Absolute Error =  0.3879074017666768
Mean Squared Error =  0.3879074017666768
RMSE =  0.6228221269083789
r^2 =  -0.5712650093015388
```

Lessons Learned

- Throughout the project, I learned a lot more about the analytical part of reviewing data.
- Some of my data did not have values and were needed to be given values.
- Also, some of my data that I had had no correlation to whether a song was deemed popular or not.
    - Therefore, some of it had to have been removed.

Conclusion

- Being able to create a program to predict song popularity was very enjoyable.
- The fact that the project correlated to something I was already very passionate about made it very rewarding to see my machine learning program work.
- The project gave me much more experience with pandas.

References

A few of the resources I used throughout the project.

https://towardsdatascience.com/song-popularity-predictor-1ef69735e380

https://medium.com/m2mtechconnect/predicting-spotify-song-popularity-with-machine-learning-7a51d985359b

https://www.youtube.com/watch?v=Yg3RGlucdQU&t=717s

Thank you

Thank you for the opportunity to gain an introduction into Machine Learning!

Johnny Koponen