Plan:

* Option 1: focus solely on billboard data. Use Genre, Artist, Lyrics, etc. to predict rank
* Option 2: focus solely on spotify data and create the dataset to compare hit songs v non-hit songs from the same album...predicting song likeability using 5-7 factors

Outline:

* Wednesday 7/15
  + Finish gathering data
  + Begin machine learning / analytics
* Saturday 7/18
  + Finish machine learning / analytics
* Monday 7/20
  + Hosting / displaying data
  + Javascript
* Wednesday 7/22
  + HTML / CSS / Bootstrap design
  + Finishing javascript touches

----------

Interesting bits - since there are 6, we could each take on 2 of these?

Outline

* The Question: Can we predict popularity based on song attributes?
* The Data:
  + Spotify API
    - Trey you can spend a few mins talking about this? Since you gathered our data instead of us using a kaggle set
  + Musical attributes
  + Artist attributes
  + A word on “popularity” being included
    - Relative popularity - songs by Drake right now will have a higher popularity than Sugar Ray, which couldve been a similar popularity 20+ years ago
* The Analysis:
  + A word on “class weights” imbalance and why that matters
    - We didn't have a even data set
    - 90% 0’s and 10% 1’s
    - Had to balance our data for training
    - Stratify
      * Incorporate Stratify into our code
      * Stratify essentially insures that if the testing data had 15% 1’s in it, that our test data would as well
* The Results:
  + A word on random states and variability in results
    - Random State = using the same data in training to predict the test results
    - 2 findings with Random states
      * Music has such a wide variation across all of our features (Genre, Time period, danceability, tempo, etc.) that different random states (slices of data) can greatly vary our model results
      * Different “data slices” is more sensitive to different results when there is limited data (our set is ~2500 songs)
  + Overall - mixed
  + A word on score vs precision vs recall and how it relates to our research question
    - Show Confusion Matrix
    - Precision
      * If there were 50 Hits within a data set of 500 songs, how many hits did it “find” or “select”
    - Recall
      * How many times did the model mistake a hit for an average song or vice versa
  + A word on the behavior of different models - Flatterer v Sour Puss
    - Logistic Regression
      * “Cost Saver / Pragmatic”
      * Selected ¾ of top hits but still missed 25%
      * Only picked an average song as a hit 70%
      * Pros
        + Would be very effective at choosing which songs to spend marketing on and which ones to pass
      * Cons
        + WOuld miss some hits that would make up for other average songs
    - Decision Tree / Random Forest
      * “Trash Man”
      * Basically says all your songs are Trash
        + Low positive recall - missed a lot of the hits
        + High negative recall - rarely mistook a average song for a top hit
    - SVM
      * “Yes Man”
      * High Positive Recall
        + It selected most of the actual top 10 songs in its test set
      * Low Negative Recall
        + 50/50  - basically said half the songs were “top 10 hits” even though they were average songs
  + Show some actual v predictions
* Conclusion:
  + It won’t help you identify songs that will make it to the Top 10 BEFORE they are produced
  + But once you produce a song and it’s on spotify for enough time, it may help you decide whether to invest in marketing to make it BIG
* Additional Fun with Songs:
  + Can ML models identify the artist?
  + Mainly, yes
  + A word on the false positives and how that might lead to interesting findings