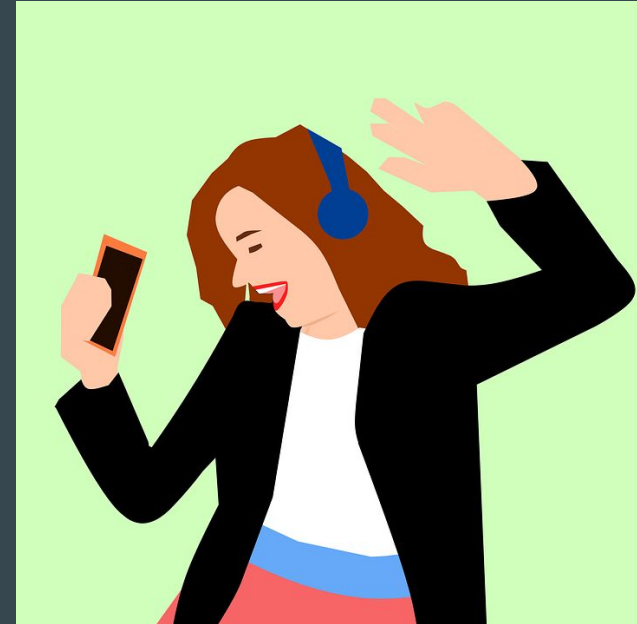
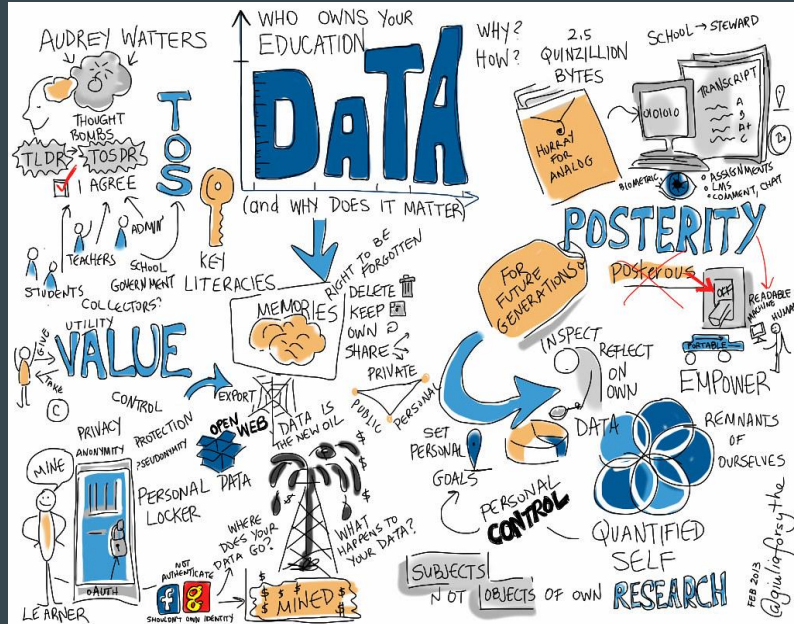


# Spotify Music Recommendation Engine

By Rane Najera

Data Science Capstone #3



# What are Recommendation Engines?

- Recommendation Engines are a Machine Learning algorithm that tries to find patterns in your data
- By learning the patterns about you, Recommendation Engines are able to recommend something to you based on your own personal data



# Recommendation Engines are everywhere!

Who uses Recommendation Engines?

- Google
- Amazon
- Facebook
- Spotify / Apple Music
- Netflix
- Youtube
- Any targeted ad on the internet (“cookies”)



What can Recommendation Engines recommend?

- A funny cat video
- The perfect pair of socks
- Someone you may know on social media
- The newest movie / video game
- Your future favorite song you have not heard yet

# Building my own Recommendation Engine

The Problem:

With the value of recommendation engines almost limitless, I set out to learn more about my the patterns in my own data

The Goal:

To build my own music recommendation engine, and discover some cool, new songs



# Data

- The Datasets for this project was all taken from my own personal Spotify account, by using the Spotify API, Spotipy



Datasets broken down into 3 categories:

- “Ratings”
  - Top Artist Dataset
- “Train/Test”
  - Saved Song Dataset
- “Validation”
  - Spotify’s Recommended Songs Dataset

# EDA

- My Top 10 Artists:

- 1) Young Dolph
- 2) Bo Burnham
- 3) Laurent Korcia
- 4) Black Eyed Peas
- 5) The Beatles
- 6) Oasis
- 7) Drake
- 8) Key Glock
- 9) Gucci Mane
- 10) Kanye West



- My Top 10 Songs:

- 1) Les Valseuses
- 2) Yebba's Heartbreak
- 3) Came For The Low
- 4) READY TO GO
- 5) Don't Wanna Know
- 6) SUVs (Black on Black)
- 7) Who Want Smoke??
- 8) JU\$T
- 9) Garden Party
- 10) Still Trappin'

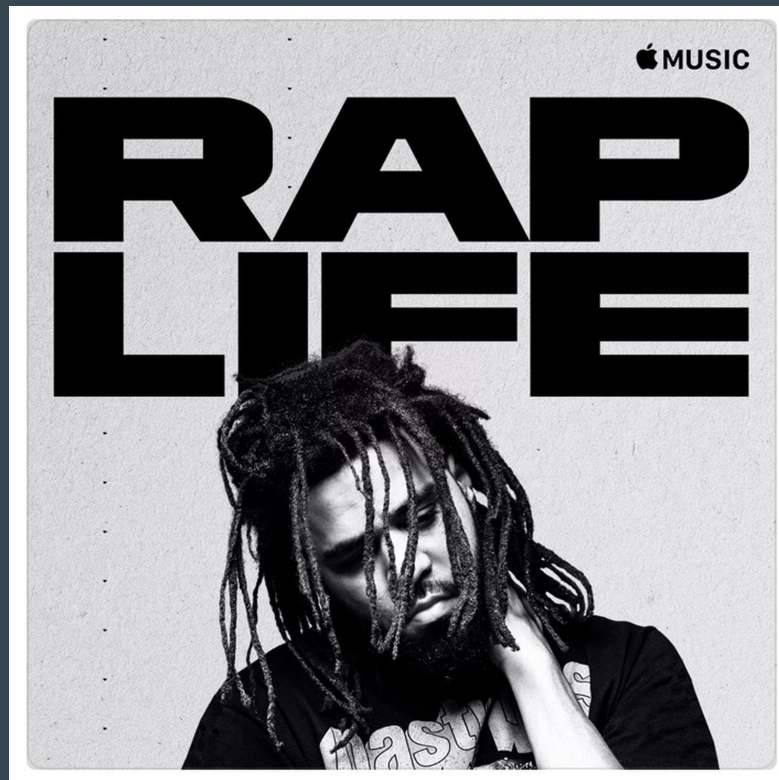
# Model Metrics

- Classification Problem
- Used the “top artist” dataset to rate the “training/test” dataset. “1” rating if the song was by one of my top artists, “0” if not
- Applied rating to 399 songs in a dataset of 1133
- Trained data with song content variables, such as:
  - Tempo
  - Loudness
  - Danceability
  - Acousticness
  - Key
  - Liveness
  - Valence
  - Speechiness
  - Energy
  - Instrumentalness
  - Mode
  - Genre

# Model Metrics

- My Top 10 features of importance:

1. rap: 0.0733806
2. speechiness: 0.05040762
3. energy: 0.0453377
4. danceability: 0.04341639
5. instrumentality: 0.04269710
6. loudness: 0.0409288
7. acousticness: 0.0350782
8. detroit hip hop: 0.03430248
9. tempo: 0.0318333
10. valence: 0.0313792





# Modeling

- Fit 3 Different types of Classification Models and compared accuracy to determine best model:

- Logistic Regression
- kNN
- Random Forest Classifier

Accuracy Results:

- Logistic Regression: 0.665
- kNN: 0.656
- Random Forest Classifier: 0.947

# Recommendation Engine Outcomes

- By using my trained models with my top artist spotify data, I was able to successfully predict 116 songs, both old and new, that I would enjoy with a probability of over 0.75



(Me after listening the the recommended songs)



# Recommendations for how to use

- Model can be trained specific to anyone, by using other Spotify users data
- Datasets can be “rated” using other methods, such as: favorite songs
- Simply enter in personal data gathered from the Spotipy API, and my model can recommend songs specific to you!



# Conclusions

- Song Recommendations were lit!



- Because I used “top artist” data, and my top artists were predominately rap/hip hop genre, and the recommendations with the highest probability were all of a similar genre/style
- Model could be more accurate if given a more complex rating system, such as rating every song specifically, etc

# HAPPY LISTENING!

An abstract graphic featuring various musical symbols in shades of blue and white. It includes a large treble clef on the left, several musical notes of different sizes, a heart symbol, and a sharp sign (#). The background is a solid dark blue.

**Rane Najera**

**Email : [ranenajera@gmail.com](mailto:ranenajera@gmail.com)**

**Github: <https://github.com/Ranealdinho/Springboard>**

**Linkedin: <https://www.linkedin.com/in/rane-najera/>**