CDC2023 – Music Taste Classification

Jay Sakarvadia & Jack Pamukci

What Nostalgic Sitcom Character Shares your Music Taste?

Anyone a fan of these shows?

1. How I Met Your Mother
2. Friends
3. The Office

What's My Own Music Taste? 😱

THE SPOTIFY API 😒

- Machine & Deep Learning Training 🚫
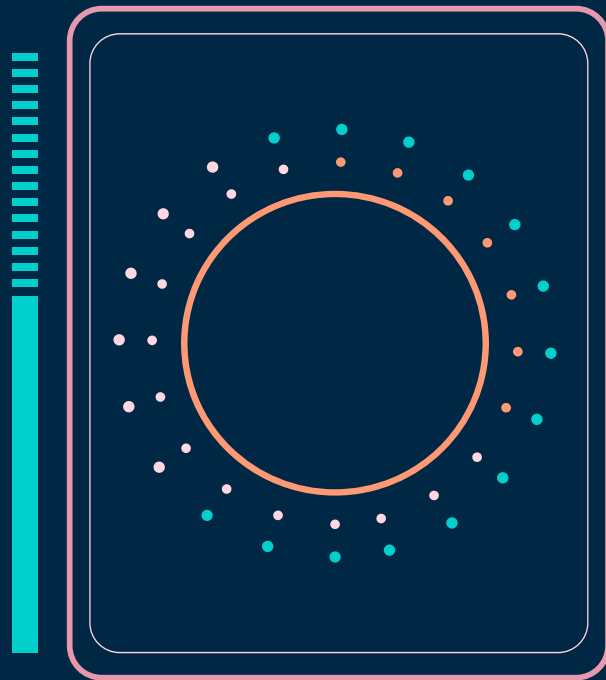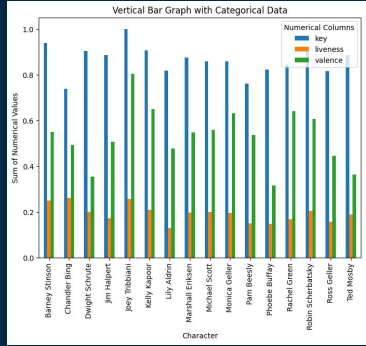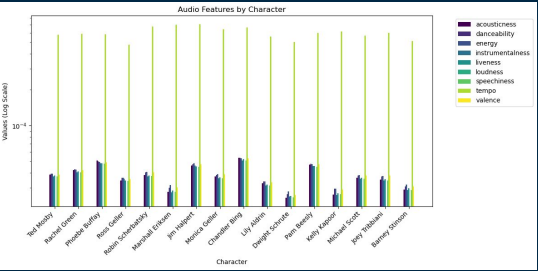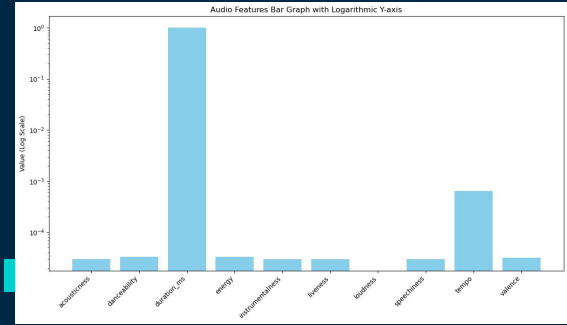- Rate Limits
- Tokens, Authentication, etc.

BUT...

THE SPOTIFY API 🥰

- Ability to build custom datasets
- Granular data e.g. audio features

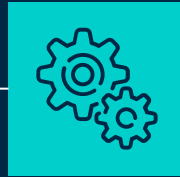**Ability to build our own narrative; control over entire data pipeline**

General Data Processing (Pre-Analysis)

- Connect to Spotify API (created subsequent methods for API connection)
  - **get_user()** - retrieves audio feature data given users top 100 long-term songs
  - **get_playlist()** - retrieves audio feature data given selected playlist

- Preprocess data: **get_feature_list(data)**
  - Dropping categorical variables (one-hot encoding proved to be inefficient),
  - max-min normalization for all remaining numeric features

- **consolidate_features()** - Consolidate both user and playlist audio features data into a singular mean feature vector

OUR ANALYSIS APPROACH

01
Cosine Similarity

02
Euclidean Distance

UNDERSTANDING THE PROBLEM

Cosine Similarity

Cosine similarity is a metric used to measure how similar two vectors are by calculating the cosine of the angle between them. A value of 1 indicates total similarity, 0 indicates orthogonality (no similarity), and -1 indicates total dissimilarity. It's often used in text analysis to determine the similarity between documents or sentences.

Euclidean Distance

Euclidean Distance is a measure of the straight-line distance between two points in Euclidean space. It's commonly used to gauge the similarity between two data points, with smaller distances indicating higher similarity.

Cosine Similarity
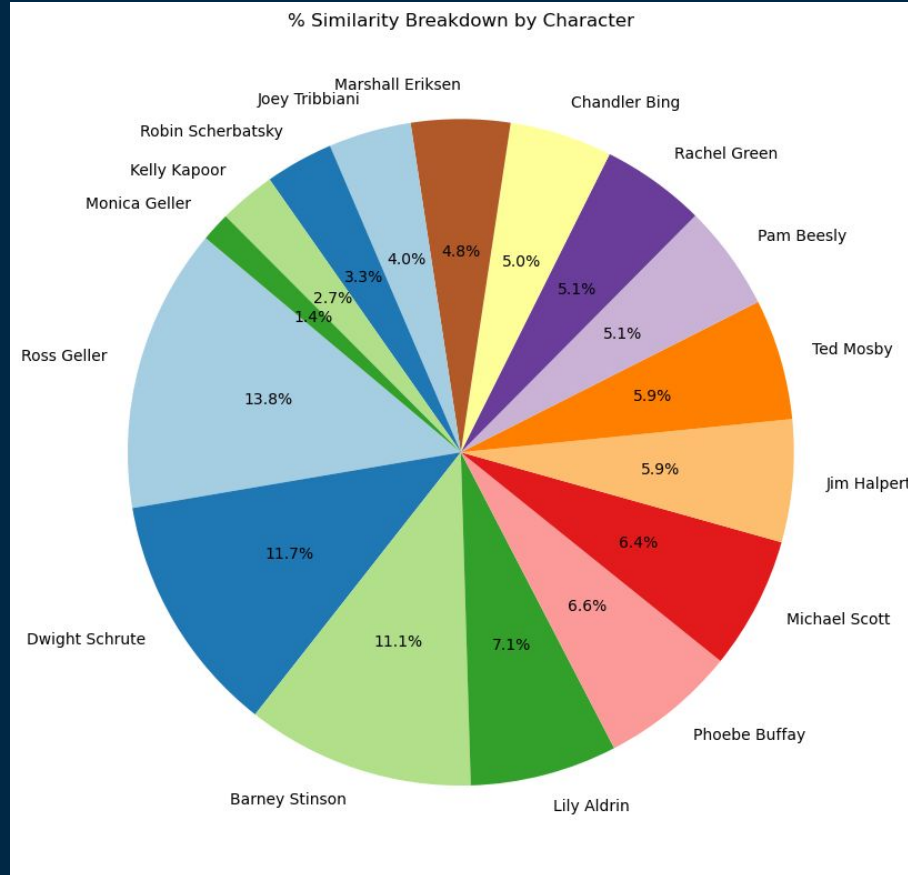
Here we go.

01

Cosine Similarity

- **cosine_calc()** - calculates cosine similarities
  - outputs similarity measurement in domain (-1,1)
  - otherwise interpreted as angle between two vectors (user vector and playlist vector) in space

$$\cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum\limits_{i=1}^{n} A_i B_i}{\sqrt{\sum\limits_{i=1}^{n} A_i^2} \sqrt{\sum\limits_{i=1}^{n} B_i^2}}$$

- then converting similarity output degrees using equation below in order to eventually be able to understand user-character % Similarity Breakdown

$$\text{angle (in degrees)} = \arccos(\text{cosine similarity}) \times \frac{180}{\pi}$$

- % Similarity Breakdown = cosine_similarity/sum(cosine_similarities)

% Similarity Breakdown by Character

- Marshall Eriksen — 4.8%
- Joey Tribbiani — 4.0%
- Robin Scherbatsky — 3.3%
- Kelly Kapoor — 2.7%
- Monica Geller — 1.4%
- Chandler Bing — 5.0%
- Rachel Green — 5.1%
- Pam Beesly — 5.1%
- Ted Mosby — 5.9%
- Jim Halpert — 5.9%
- Michael Scott — 6.4%
- Phoebe Buffay — 6.6%
- Lily Aldrin — 7.1%
- Barney Stinson — 11.1%
- Dwight Schrute — 11.7%
- Ross Geller — 13.8%

Euclidean Distance

Round 2.

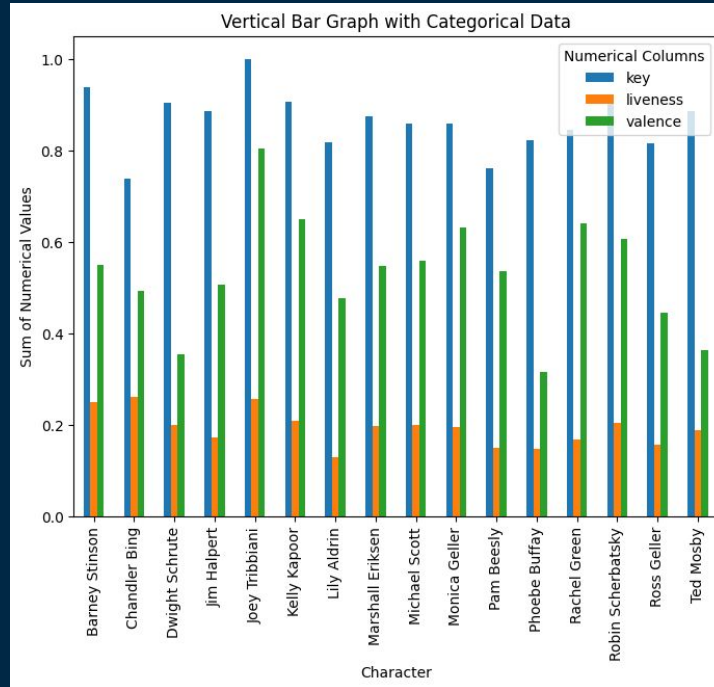**01**

Euclidean Distance

- Regression analysis (repeated n # of audio features) for audio feature selection; take 3 most correlative features
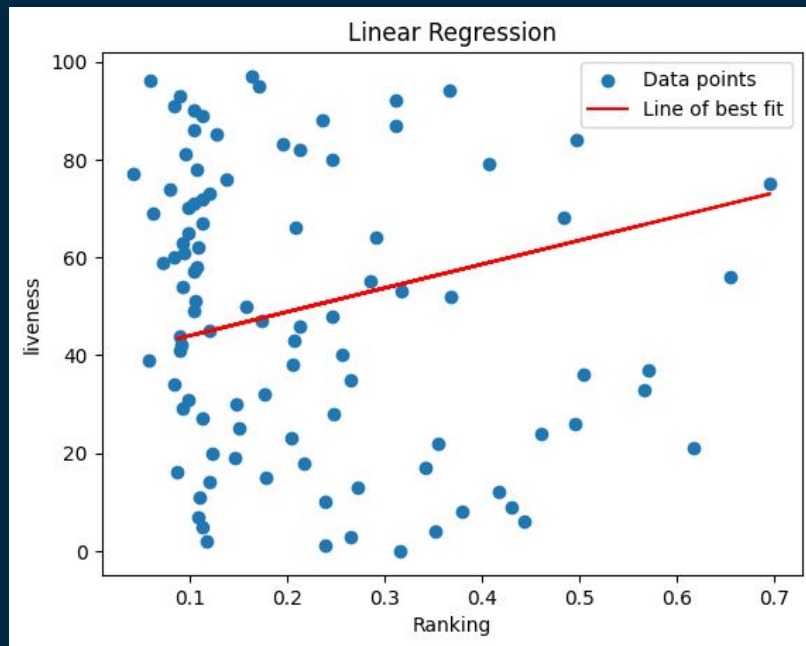
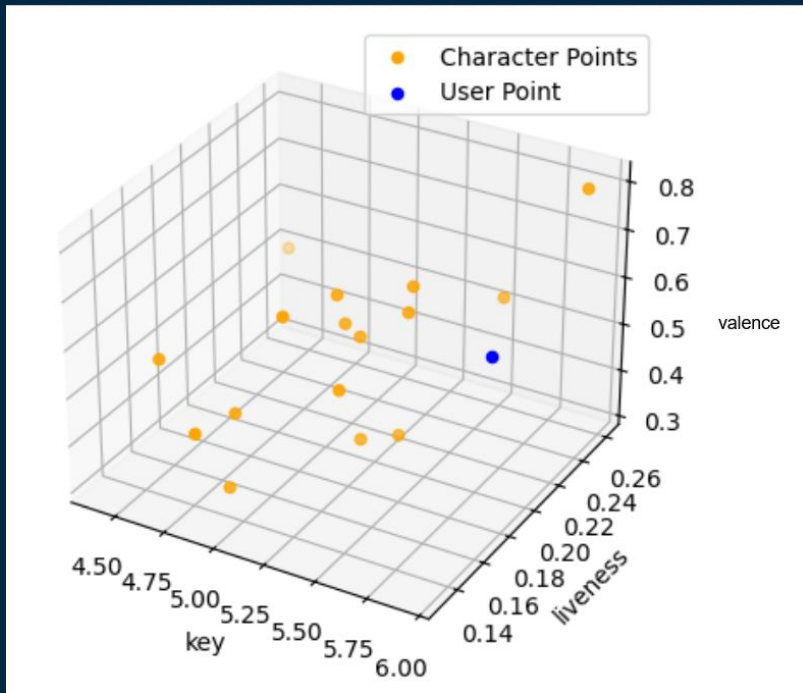- Above represents strong correlation between ranking of song and the feature

Euclidean Distance

- Consolidated user audio correlative feature data into a single mean feature vector
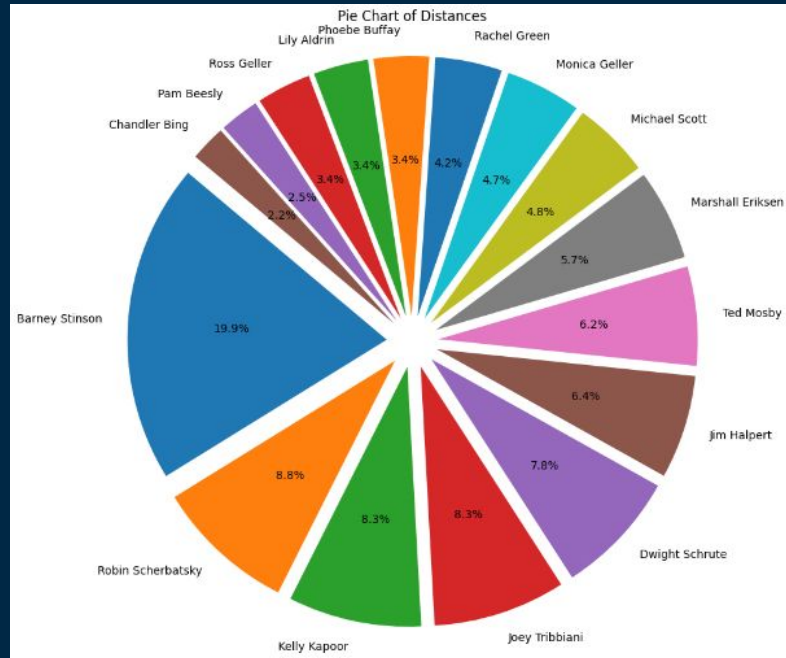
Euclidean Distance

- Collected distance between user vector and playlist vector across 3D linear space using distance formula

Euclidean Distance

- Converted distance to % using inverse normalization formula:
  - (1 / actual_distance) * 100

Pie Chart of Distances

OUR WORK IN A DIFFERENT LIGHT

COMPARATIVE ANALYSIS

FEATURE SELECTION
- the bain of high-dimensionality

ACCESS TO ML
- not everyone has access to the latest and greatest; equity

QUESTIONS?
(BE NICE, OR DON'T)

THANKS