Beatles Song Catalogue Data Challenge

PRODUCED @ ABBEY ROAD

Object 1 – Account for Duplicates and Anomalies

Checklist

- Account for duplicates in data files:
  - Example: "a-day-in-the-life" and "a-day-in-the-life-live-in-amsterdam"
  - This includes any weird or missing data
  - Remove anything that is deemed erroneous by eye or by logical method

Results

- A 20.07% reduction in the entries of data available.
- Used a few simple functions to check for entries where lyrics are blank, and where "instrumental" is not in the content of the file.
- Accounted for language disparities as there were some releases in German.

Objective 1 – Results Cont.

Checklist

- Run the **Cosine Similarity** of the lyrics for each song entry.
  - Only on the "old" lyrics set
- Set a threshold for acceptable similarity, in this case .7
- Create an omission list based on the above.

Results

Song_1	Song_2	Similarity
Revolution-1	Revolution	0.93
sgt-peppers-lonely-hearts-club-band_reprise	sgt-peppers-lonely-hearts-club-band	0.87

- Above entries omitted – essentially kept one instance (song_1) instead of both.
- Creation of a final "clean_lyrics" dataframe with data cleansing applied as per checklist.

Objective 2 – Answer 2 of 5 Available Questions of the Data (Approach)

Question 1

Question: ***Which song has the largest amount of repetition?***

- This question was analysed with two approaches. The first being a basic frequency count after looking at the "\n" delimited sentences in each set of lyrics.
  - Generated a count of how many lines repeated...
  - And what is the sum of those repetitions.
- The second approach involved looking at the n_gram (bigram) similarity of each sentence within the set of lyrics.
  - Returned a score based on the number of jaccard similarity points.

Question 2

Question: ***How many of the songs feature the song name (found in the file name) in the song lyrics?***

- This question was analysed by simply turning the song names into a format similar to how the lyric strings are formatted.
- The resulting strings are then checked to see if they appear in the lyrics string at any point.

Bonus

Question: ***Which songs are the most similar?***

- **Included as the results were available as per the previous step**
- *Except whereas before this data was used for cleaning purposes, with the omitted entries we can now look at similarity across songs.*

Objective 2 - Answers

Answer 1

Song	Repetition Score
All-together-now	788
--want-you-shes...	293
I-wanna-be-your-man	200

All-together-now is far and away the clear winner based on its score.

A look at its lyrics confirms this!

Answer 2

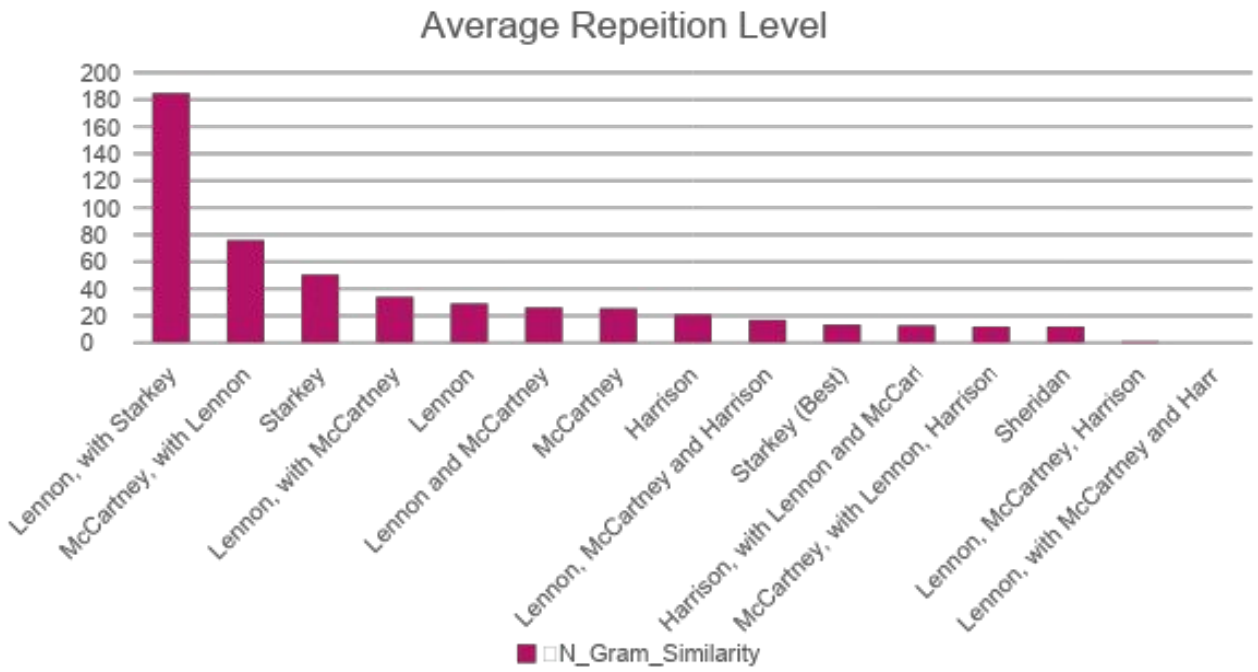
Appearance of Song Name in Lyrics	Count
Yes	91
No	125

Bonus Answer

Song1	Song2	Similarity
hey-jude	run-for-your-life	0.501634
love-me-do	p-s-l-love-you	0.496573
i-want-to-hold-your-hand	you-really-got-a-hold-on-me	0.496076
all-you-need-is-love	p-s-l-love-you	0.493689
i-will	p-s-l-love-you	0.493070

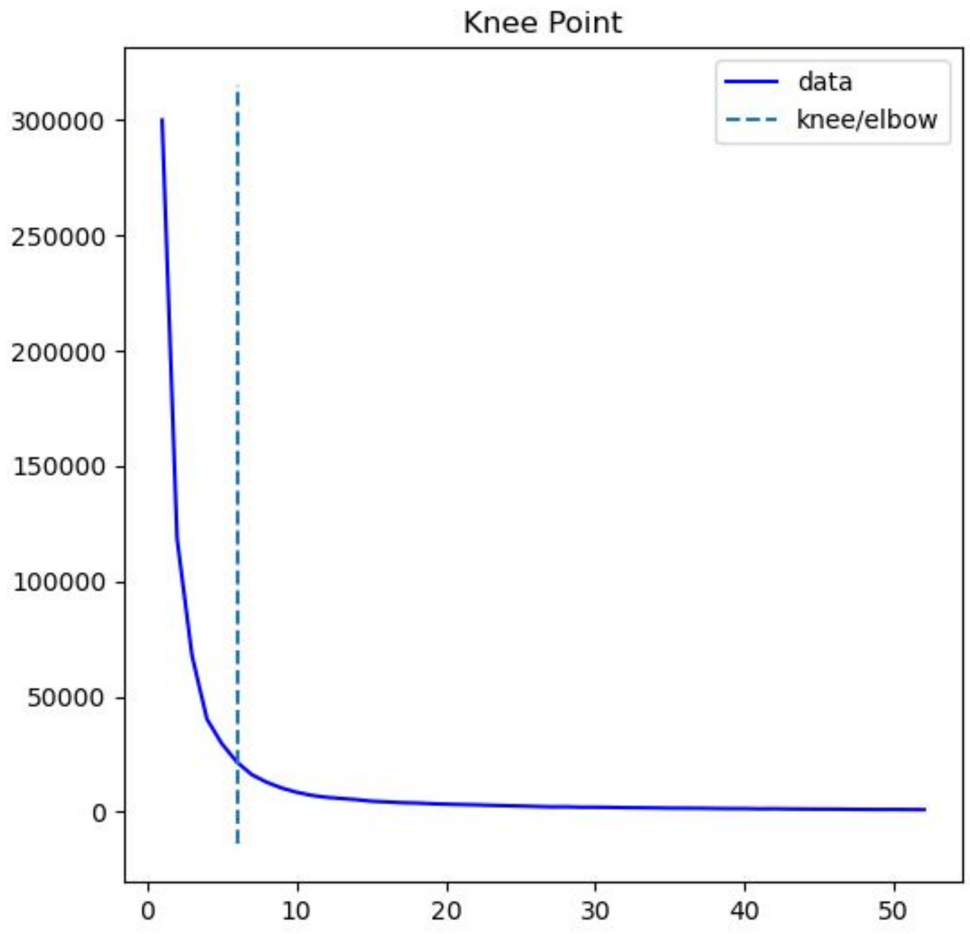
Objective 3 – Derive a single piece of insight from the data that is interesting

- In this instance, I used a third-party dataset to contextualise some of the results that were previously generated against aspects of the singers and songwriters. As well as the runtime of the songs in question. (source here)
- The first interesting bit of insight was the minor negative correlation seen between repetition (as previously calculated) and position in the billboard top 50.
  - With a score of –0.384 - we can see that the higher the level of repetition, the higher it was on the charts.
- The second concrete insight was the relationship between lead vocalist and repetitions
  - Seen to the right, we see that songs where John Lennon and Ringo Star were the lead vocals had far and away the highest level of repetition.



Objective 4&5: Clustering (Method and Value of "K")

- I chose to use K-Means Clustering for this analysis, and the value of K was determined by use of the elbow-plot method. This gave me a K value of 4 by determining (by eye) where the sum of squares slope changed trajectory most drastically.
- HOWEVER – I determined the value of 6 by use of the kneedle algorithm to compound this. The result was a value of 6.
- Further information on the algorithm and package can be seen HERE.



Objective 4&5: Clustering Results

- In order to try and see a logical distribution across clusters, I tried multiple methods before settling on two sets of input to test my data. The first was a mix of the tdifd-generated matrix along with the additional columns I had made during my previous analysis: **N_Gram_Similarity**, and **Compound Sentiment Score**.
- The second was simply on the lyrics alone, to leave out the columns I had engineered myself.
- These yielded two sets of cluster allocations (attached separate to this deck).
- The distributions for both are seen to the right.
- The new lyrics clustered to cluster "0" in each instance.

	count
Cluster Labels	
0	97
1	1
2	4
3	24
4	80
5	10

	count
Cluster Labels LYRICS_ONLY	
0	11
1	22
2	67
3	28
4	76
5	12

Objective 4&5: Cluster Meanings

- Taking the joined totality of words for lyrics in each cluster, I aimed to find common themes in each cluster.
- This was done by preprocessing the data in each cluster, and then looking at the key phrases in each.
- Then word frequencies were taken into account and used to order this data. The top X for each were extracted and used to determine commonalities in these clusters.
- For details sake, I've included the top 20 as a whole in the attached output to this deck.
- The meanings for this type of analysis I believe are best expressed by these common themes as dictated by word frequency.