Predicting Decades of Songs with Lyrical Content

Springboard Capstone Project 1
Michael Buck

Motivation:

- Music is an important part of popular culture and one of the most universally adored artforms among humans.

- What trends can be discovered by analyzing music? Can analyzing music provide insights into popular culture?

The Problem:

- Can you predict what decade a song is from based purely on its lyrical content?

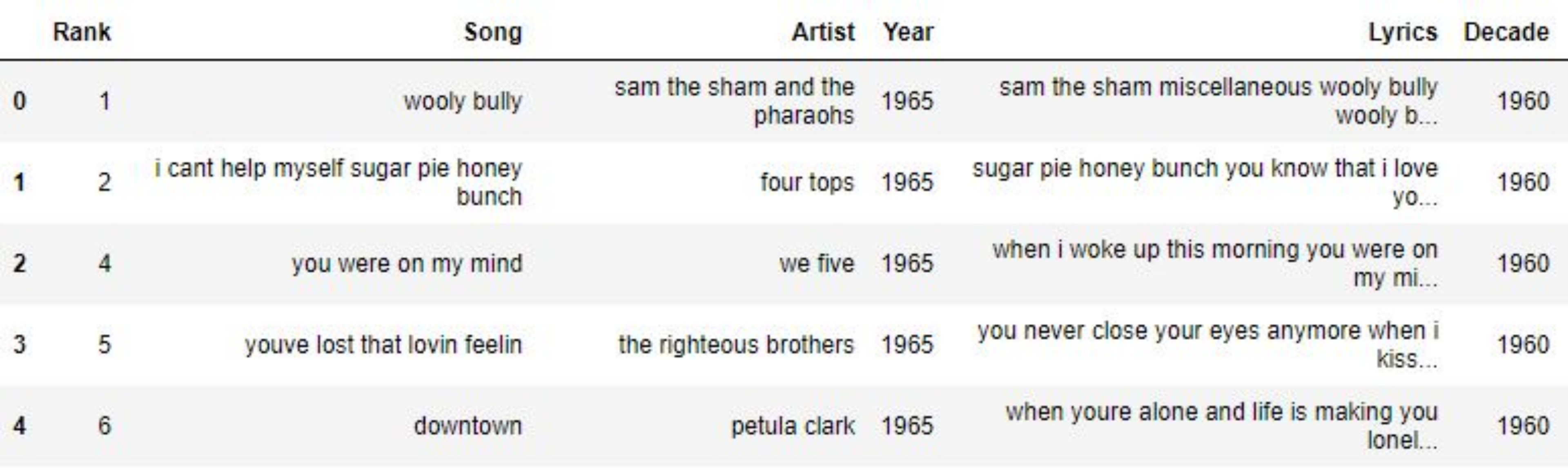
- How would you solve this problem?
  - Use a classification algorithm using the lyrical data of music from various decades to predict what decade a song is from

The data:

- The data for this project came from Kaggle user RakanNimer, and is the lyrical data of the Billboard Hot 100 songs from the years 1965-2015
    - https://www.kaggle.com/rakannimer/billboard-lyrics

- This data was chosen for this problem because it could show how language in popular music changes over time.

What does the data look like?



	Rank	Song	Artist	Year	Lyrics	Decade
0	1	wooly bully	sam the sham and the pharaohs	1965	sam the sham miscellaneous wooly bully wooly b...	1960
1	2	i cant help myself sugar pie honey bunch	four tops	1965	sugar pie honey bunch you know that i love yo...	1960
2	4	you were on my mind	we five	1965	when i woke up this morning you were on my mi...	1960
3	5	youve lost that lovin feelin	the righteous brothers	1965	you never close your eyes anymore when i kiss...	1960
4	6	downtown	petula clark	1965	when youre alone and life is making you lonel...	1960

Describing the data

- Most of the columns in the previous slide were self explanatory, however, ‘rank’ does mean the rank that a song achieved in the Billboard Hot 100

- Not pictured was a source column which was dropped because it only denoted the source of the lyrics, which was irrelevant to this project.

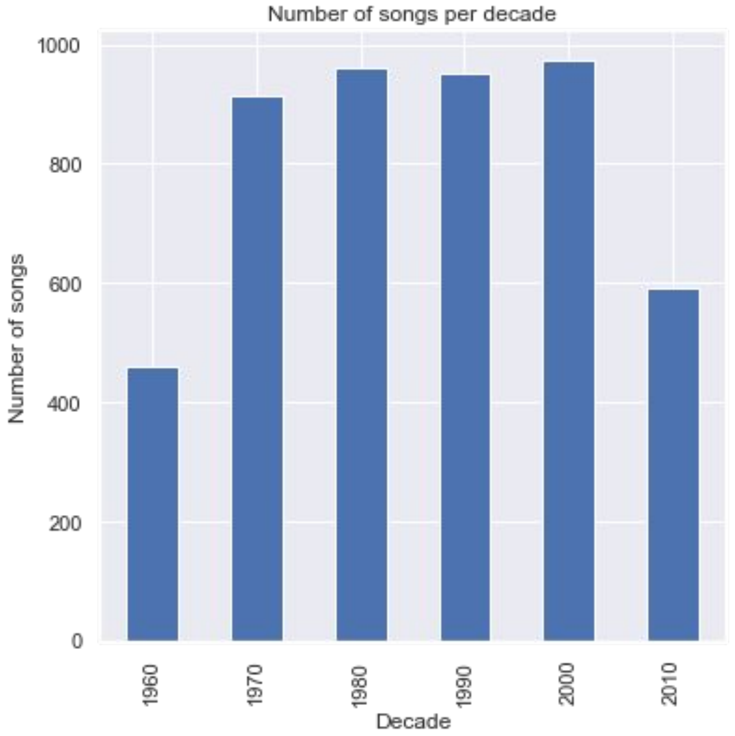
Cleaning the data:

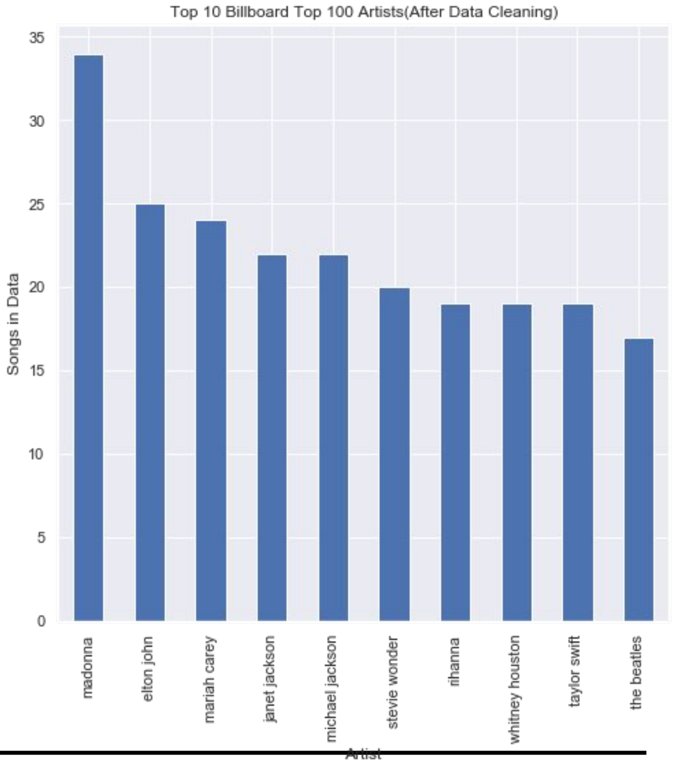
- Songs that were missing lyrics were dropped due to time constraints.
- Instrumental songs, while few and far between, were also dropped.
- Had to take care of some strange patterns in lyrics where artist names, genres, and album names appeared at the beginning of the song lyrics. These errors were cleaned as much as possible.

Preparing the data:

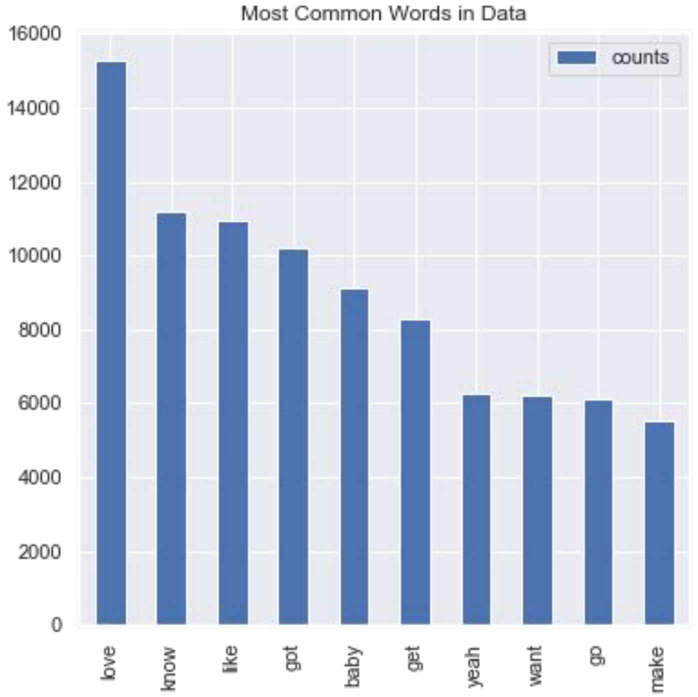
- Lyrics were tokenized, changed to lowercase, numbers were removed, and stop words were removed to ease the analysis process
- Tokenization:
    - “The quick 1 brown fox” = [‘the’, ‘quick’, ‘brown’, ‘fox’]
- Removing stop words:
    - Stop words: Very common words that carry little to no information
    - “The quick 1 brown fox” = [‘quick’, ‘brown’, ‘fox’]

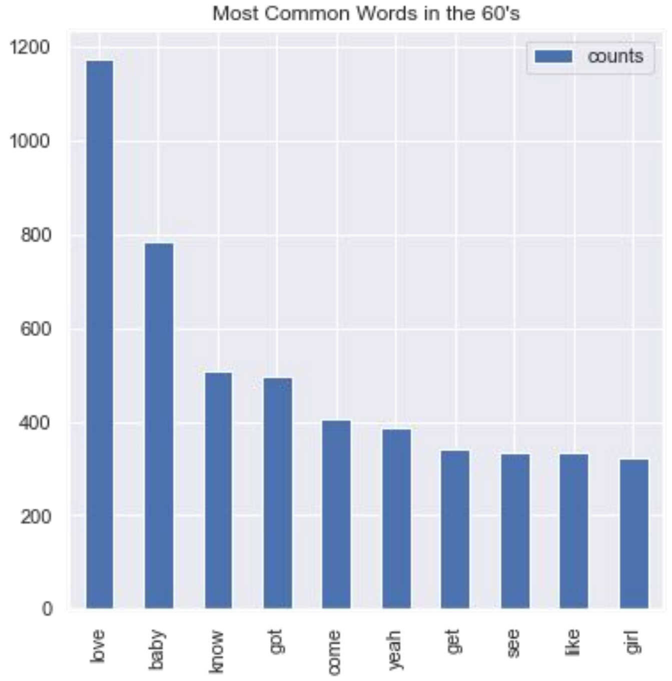
What’s left after cleaning?



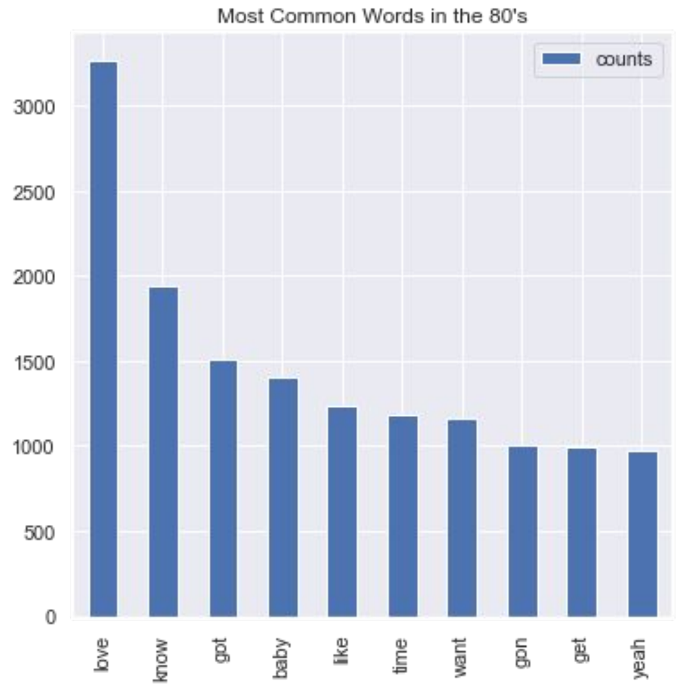


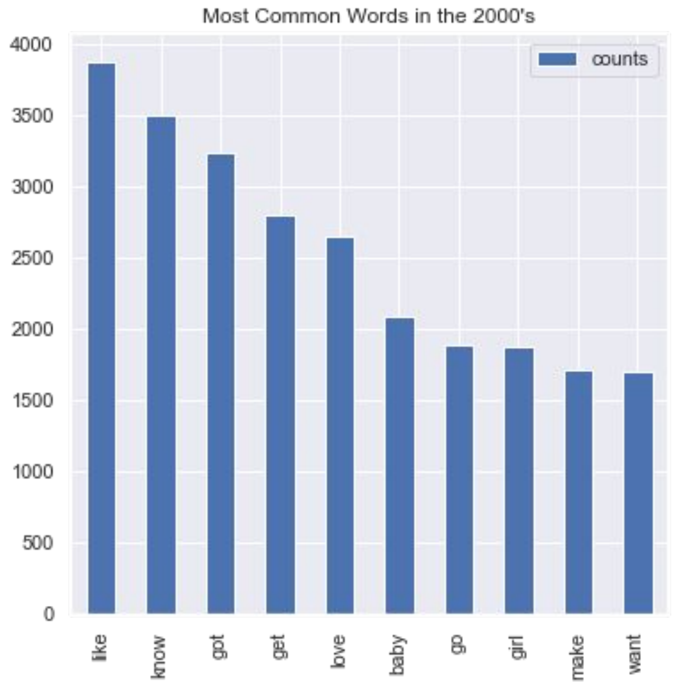
The big picture-word counts:





The big picture-word counts:



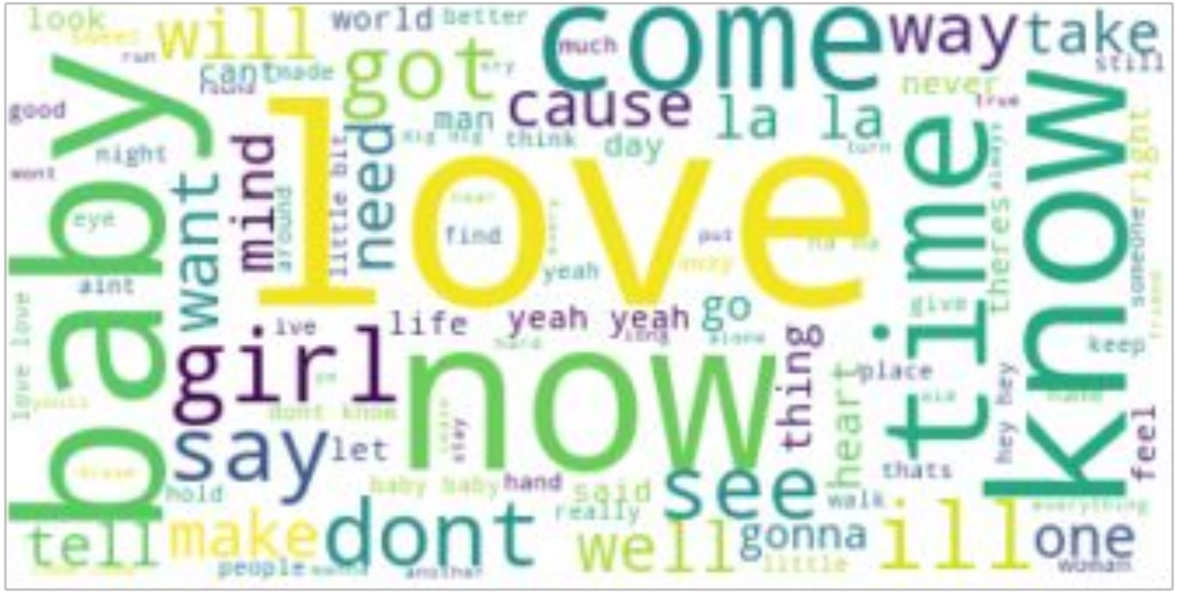


Word counts:

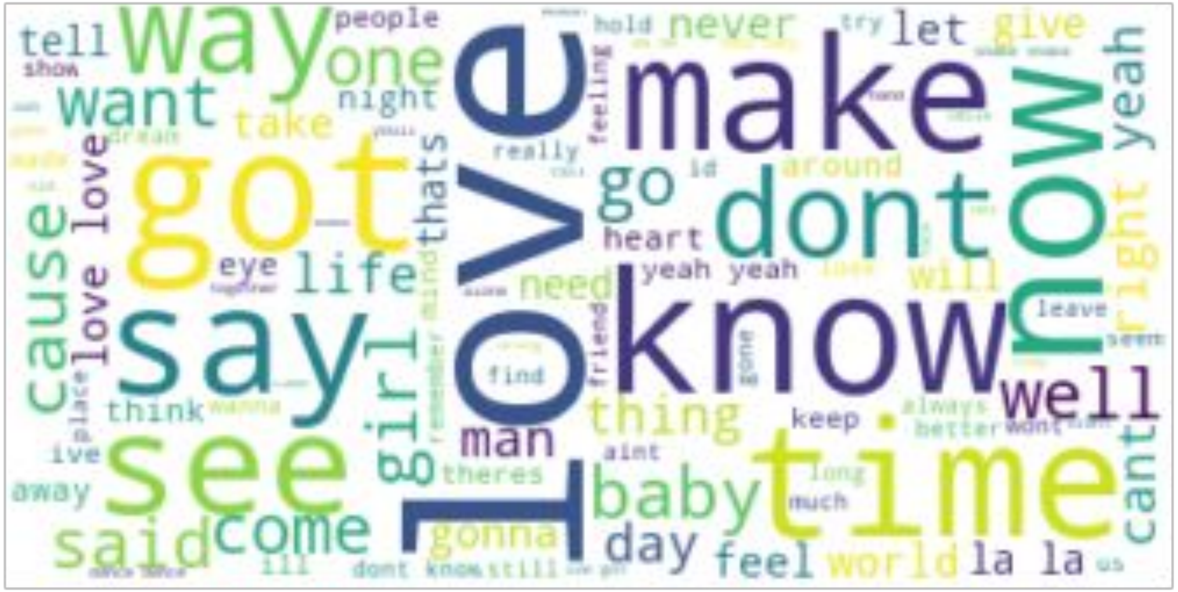
- Word counts show that most common words across decades do not vary wildly, with only slight variations in the most common words.

- Can word clouds provide a better insight into lyrical changes?

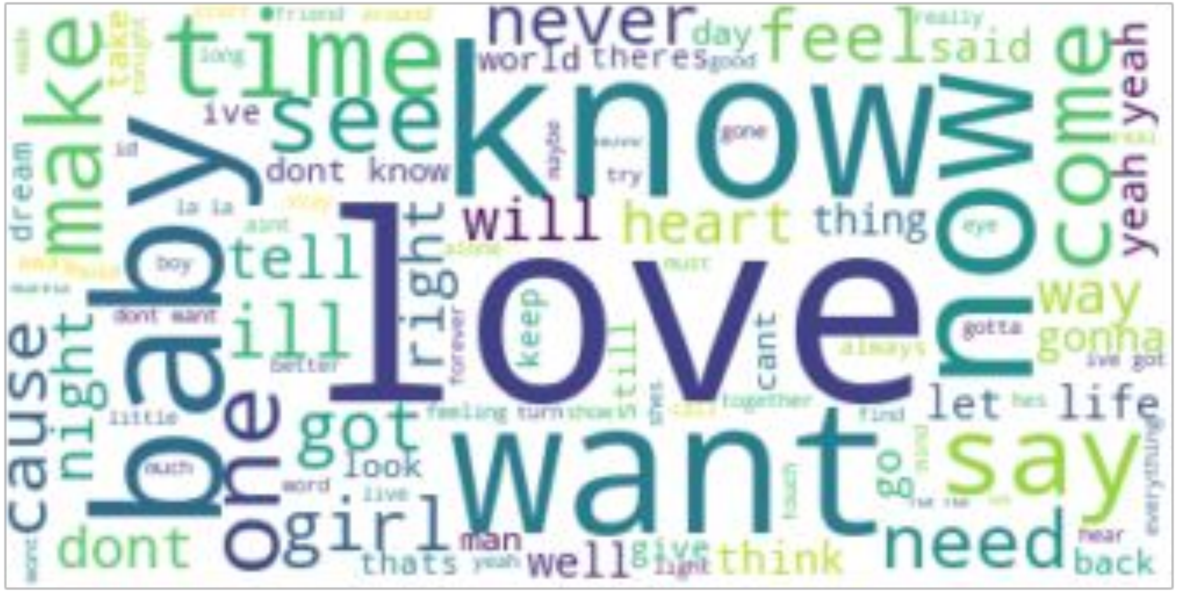
Word Cloud: 1960’s



Word Cloud: 1970's



Word cloud: 1980's

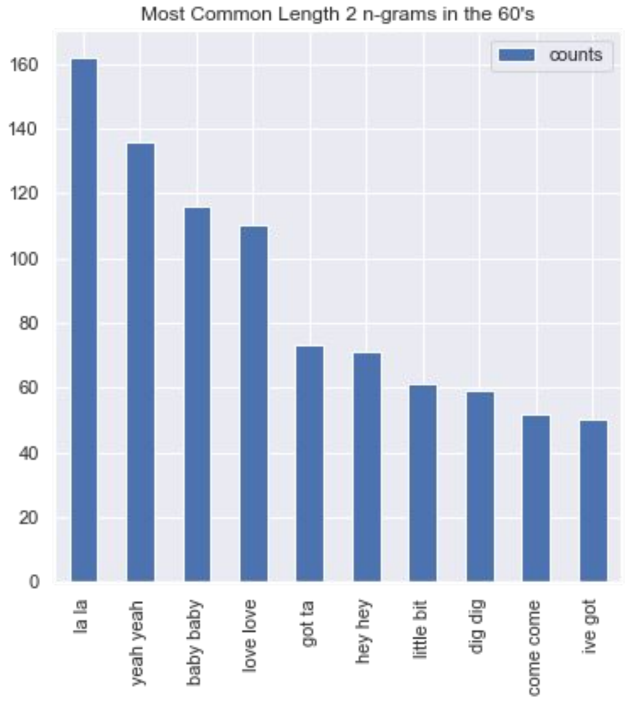


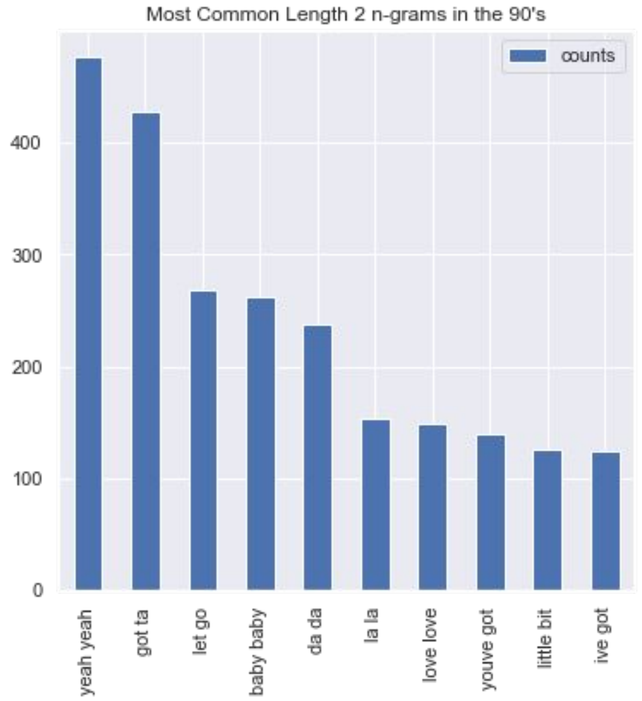
Word clouds:

- The word clouds to provide some insight into the variation of words across decades but are still quite similar.

- What insights can looking at the n-gram counts of each decade provide?

n-grams


Most Common Length 2 n-grams in the 60's
counts
0
20
40
60
80
100
120
140
160
la la
yeah yeah
baby baby
love love
got ta
hey hey
little bit
dig dig
come come
ive got

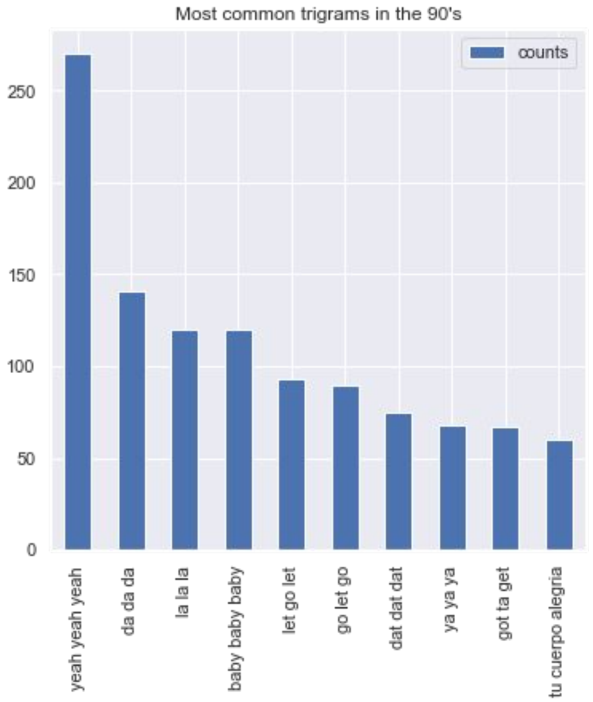

Most Common Length 2 n-grams in the 90's
counts
0
100
200
300
400
yeah yeah
got ta
let go
baby baby
da da
la la
love love
youve got
little bit
ive got

n-grams

- Many of the most common n-grams across decades are very similar, and did not provide much meaningful insight.

- Most n-grams are groups of words together that are nonsense, and these counts can be skewed in interesting ways

n-grams



- The most interesting set of n-grams were these n-grams from the 90’s, as a spanish phrase is the tenth most common trigram.

- ‘Tu cuerpo alegria’ only occurs in the Macarena, so it seems that phrase occurs more than 50 times in that song.

Moving to classification

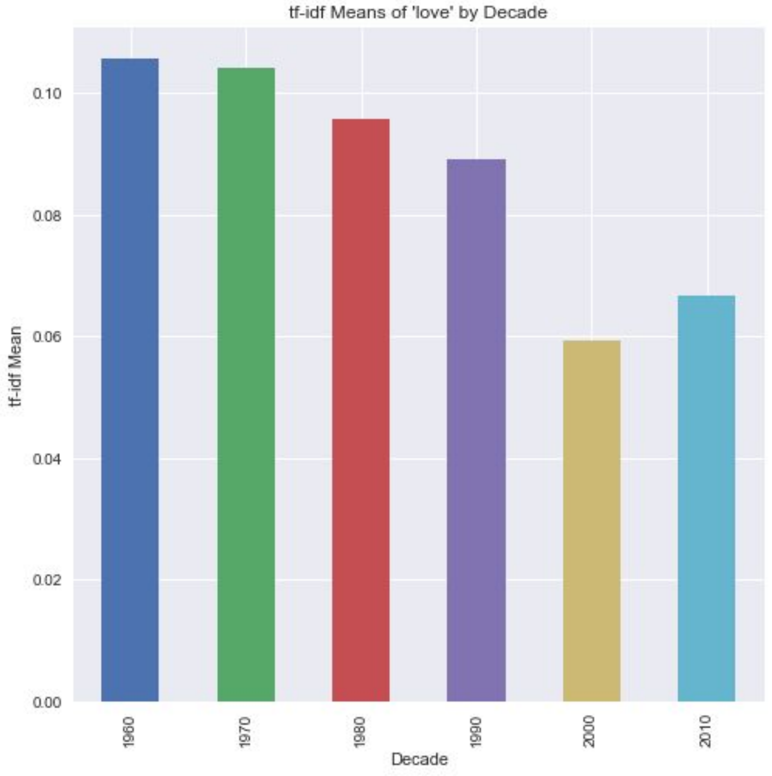
- Word counts will obviously not be good features for a classification model, so what can we use?

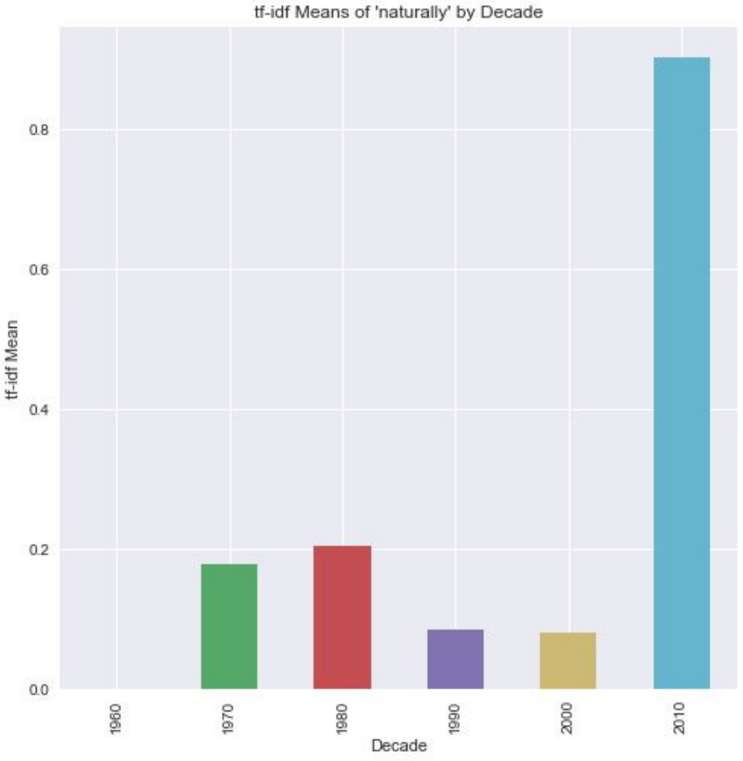
- Term frequency-inverse document frequency values can be used as they measure the importance of words to a document relative to other documents.
    - In our case, ‘documents’ are songs

Tf-idf values

- Tf-idf values were calculated using scikit-learn’s tfidf vectorizer.
- There were initially more than 40000 words after running the vectorizer, primarily due to typos. These values were cleaned, and the final number of words used as features was 3199

What does tf-idf data look like?





Looking at tf-idf data

- With ‘love’, we can see what appears to be a steady decline in loves importance across decades. This trend could mean good things for a classification model
- On the other hand, with ‘naturally’, a strange anomaly occurs. Only one song in the 2010’s used ‘naturally’, and was titled ‘Naturally’, so it is an outlier, and words with data like this are likely to cause noise for the model

Out of the box models

- Initial classification performance was checked using three out-of-the-box models:
  - A Support vector machine classifier
  - A random forest classifier
  - An XGBoost classifier

Accuracy of out-of-the-box models

Model	Training accuracy	Testing accuracy
Random Guessing	16.67%	16.67%
Support Vector Machine	20%	19%
Random Forest	98.5%	33%
XGBoost	69.7%	37.5%

Choosing a model

- The XGBoost classifier performed the best out of the box, so it was chosen as the model to use moving forward.
- The data was scaled and then had PCA performed on it in an effort to reduce the dimensionality of the features used in the model.
- After PCA, Parameter tuning for the model was performed using 5-fold cross validation

Parameter tuning

- The optimum number of primary components was found to be 1750, with a testing accuracy of 38.2%
- Extensive tuning was done across other parameters such as number of estimators, minimum child weight, maximum depth, etc.
- No parameter tuning actually increased the testing accuracy of the model(although it did increase training accuracy), so it seems this is the limit of the accuracy of the model with this data.

What have we learned

- Language in popular music does seem like it changes over time, however the changes appear to be small.
- The XGBoost model is still more than twice as accurate as random guessing in spite of the small changes in language.
- A neural network would have most likely performed better than an XGBoost classifier, however I did not know how to implement a neural network while completing this project.

Extending the problem

- Were I to do more work with this data, Id generate tf-idf values for n-grams and include them as features for the model.
- I would also use a neural network as a classifier and perhaps generate new features from the data as a neural network is a much more appropriate tool for this problem as I now know.