ITCS-3162 Final Project

Group 9 Presentation

Andrew Morgan, Carson Webb, Jack Karegeannes, Toni Cerritos Asencio, Tyler Niemonen, Anish Singh

Introduction to the Problem

As a group, we set out to explore how different song features influence emotional responses in listeners. Our goal is to use a comprehensive dataset that captures key musical characteristics and emotional labels to uncover patterns and relationships.

Below are our main objectives for the project:

- Use machine learning to analyze how song features relate to emotional responses and popularity.
- Work with a Spotify dataset containing variables like energy, tempo, danceability, and main emotion.
- Build probabilistic and regression models, and apply clustering techniques to identify patterns.
- Help users select songs based on the emotions they evoke.

About the Data Set

Source:
"900K+ Spotify Songs with Lyrics, Emotions, & More" from Kaggle (Data can be found Here)

Purpose:
Created for NVIDIA Llama-Index contest (Abracadabra project – intelligent playlist creation)

File Details:

CSV file (~1.01GB) with ~500,000 unique songs

Features Included:
- 39 total columns

- **Key examples:**
  - Artist, Song Name, Lyrics, Genre, Emotion
  - Popularity, Energy, Danceability, Tempo, Loudness
  - Suitability tags (e.g., Good for Workout, Relaxation, Driving)
  - Similar artists and songs with similarity scores

Note: Different group members, altered their variables that fit specifically for what they were trying to accomplish. Working with over 1 GB of data requires a lot of processing power and memory, and was simply used to make the data more easy to work with.

Experiment 1 (Logistic Regression)

GOAL: Use logistic regression to see if there is any correlation between the emotion a song is given on our dataset and its numerical values given on the feature of positivity, popularity, and energy.

Dataset & Features Used:

- Dataset: Spotifyregression.csv
- Features: Popularity, Energy, Positiveness
- Target: Emotion

Preprocessing Steps:

- Lowercased all column names
- Removed unnecessary data

	emotion	popularity	energy	positiveness
0	sadness	40	83	87
1	sadness	42	85	87
2	joy	29	89	63
3	joy	24	84	97
4	joy	30	71	70
5	love	26	81	74
6	sadness	17	89	65
7	joy	27	88	95
8	surprise	33	72	58
9	sadness	21	68	67
10	sadness	34	76	88
11	surprise	23	93	40
12	joy	26	80	46
13	joy	20	83	63
14	anger	23	99	61
15	anger	18	92	77
16	fear	36	76	80
17	sadness	31	89	80
18	anger	0	40	52
19	anger	4	55	78
20	joy	4	63	92
21	sadness	4	54	96
22	anger	51	31	81
23	joy	31	56	73
24	joy	18	12	16

Experiment 1 (Logistic Regression)

Popularity as Predictor:

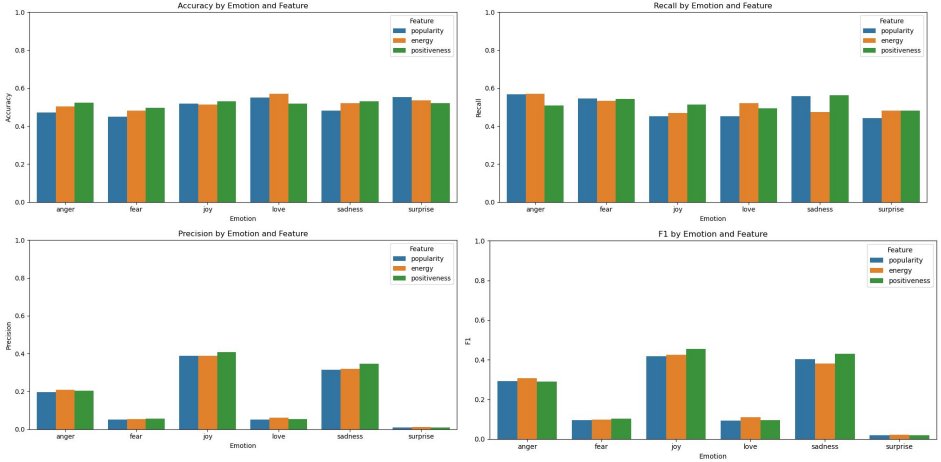
- Poor precision, especially for *joy*, *love*, and *surprise*
- Accuracy around **0.45 to 0.55**
- Weak emotional signal on its own

Energy as Predictor:

- Slight improvement over popularity
- *Fear*: recall = **0.681**
- *Joy*: better precision = **0.387**, F1-score = **0.424**
- Stronger connection to emotional intensity

Positiveness (Best Performer):

- Most consistent and reliable feature
- *Joy* precision: **0.544**, F1: **0.426**
- Stronger alignment with emotional tone
- Accuracy: **0.50–0.53**



Conclusion:

- **Positiveness** was the most meaningful predictor
- Additional features may enhance emotional prediction
- Future modeling should combine features (e.g., Random Forest) for better accuracy

Experiment 2 (Linear & Non-Linear Regression)

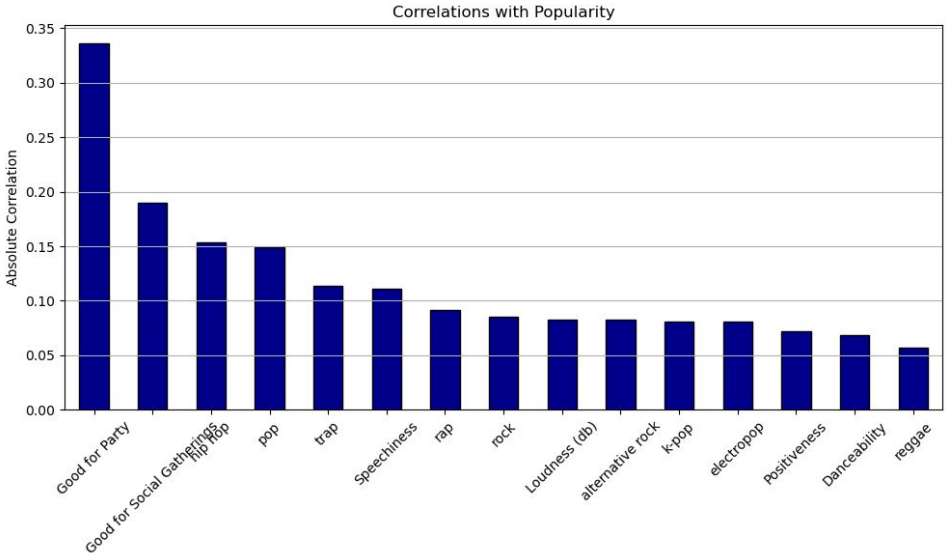
Analysis of feature Correlation with Popularity

Data Processing:

- Dropped unused features
- Used replacement, dummies, one-hot encoding
- Target Feature: Popularity

Correlation:

- Graph shows 15 features with the highest correlation to the target
- Overall low correlation with highest being Good for Party at 0.336



Experiment 2 (Linear & Non-Linear Regression)

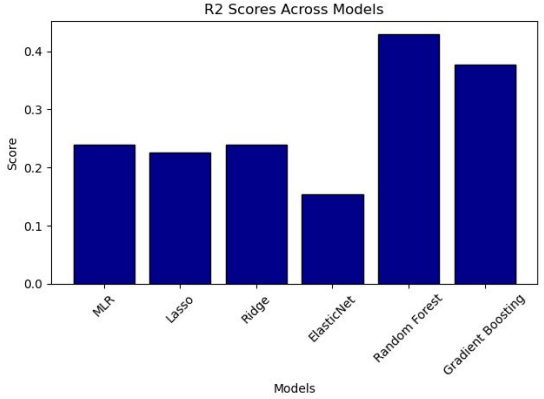
Linear Regression

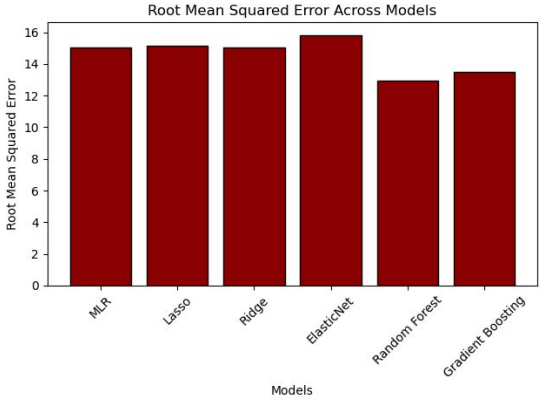
- Models Used: MLR, Lasso Regression, Ridge Regression, and ElesticNet
- Linear models performed poorly overall with R2 scores of 0.238, 0.225, 0.239, and 0.153 respectively

Non Linear Regression

- Models Used: Random Forest Regression and Gradient Boosting
- Performed Better than linear models with an R2 score of 0.429 and 0.377 respectively, but still poorly

Low R2 scores and high RMSE likely stems from overall low correlation between target and other features.

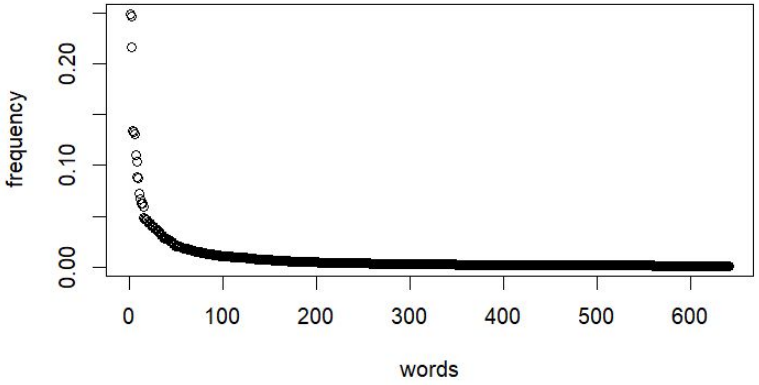




Experiment 3: Text Analysis

Analysis of the emotion categories

- Counted the frequency of words within each emotion category
  - Randomly sampled 1000 songs in each emotion
- Sorted them by frequency and created this chart
- Most common words were, unsurprisingly, very generic
  - I, the, you, and a, to, me, my, it, in
- Rapid dropoff in the frequency of words
- Some emotion categories were weird, like a “pink” emotion that consisted of a Pink Floyd song and a cover of that song



Experiment 3: Text Analysis

Relative word frequency

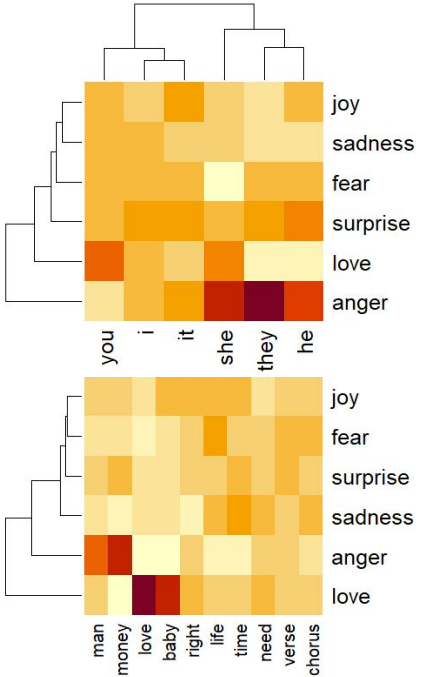
- Adjusted for overall frequency of the word

Pronouns

- You is more common in love
- I and it are fairly broad
- She is common in anger, uncommon in fear
- They and he is common in anger, uncommon in love

10 most common nouns (excluding swears)

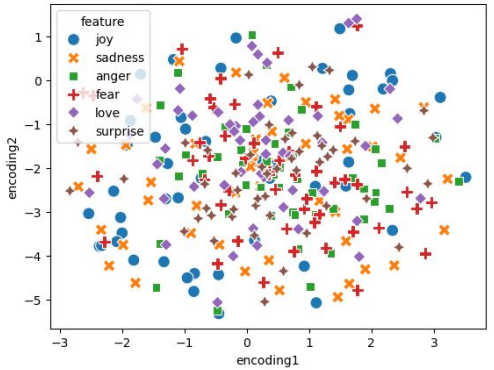
- Love and baby in love songs
- Man and money in angry songs
- In general, an individual word does not predict emotion



Experiment 4: Text Classification with Naive Bayes

- **Experiment overview**: compare document embedding methods and variants of the Naive Bayes classification model to classify song emotion from lyrics
- **Text embedding methods**
  - **TF-IDF (term-frequency inverse-document frequency)**: regularizes word frequency with how common that word is across all documents within a corpus
  - **Doc2vec**: uses a shallow neural network with a masked word-prediction objective. Text embeddings are iteratively updated such that the error of the network's prediction of masked words is minimized. Better preserves a word's context than TF-IDF.
  - Stopwords (junk/useless words) are removed before creating all embeddings.
  - **500 text features** used in all embeddings.

- **Attempted dimensionality reduction + visualization using T-SNE:**
  - **T-SNE (t-distributed stochastic neighbor embeddings):** reduces dimension of high-dimensional vector such that the KL-divergence (information loss) between the original and reduced distributions of data is minimized, preserving relations in the data

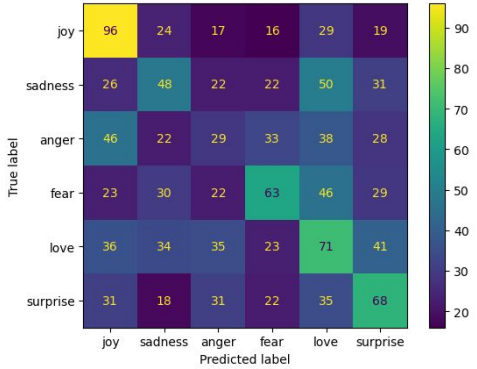


Disaster: Doc2vec text embedding visualization - similar for TF-IDF

- **Gaussian Naive Bayes**
  - Naive Bayes relies on Bayes' Theorem to calculate p(label | example).
  - Part of this involves calculating p(example_feature | label) or p(x_i | C_n) for each feature or i, and Gaussian Naive Bayes assumes that this probability forms a Gaussian (normal) distribution).
- **Multinomial Naive Bayes**
  - Multinomial Naive Bayes assumes that p(x_i | C_n) can be modeled as a multinomial probability distribution. In other words, it treats the p(feature | label) as the probability of that feature occurring in an example with that label.
- Note: I am working with a subset of our data, and am enforcing label class balance.

Gaussian NB + TF-IDF Embeddings

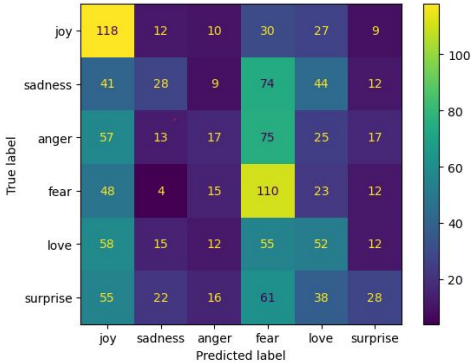
acc: 0.3
precision: 0.29
recall: 0.3
f1: 0.29
support: 1254



Gaussian NB + Doc2vec Embeddings

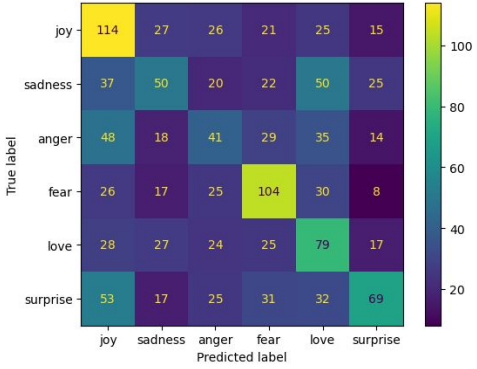
**Multinomial NB is incompatible with Doc2vec embeddings, because D2V embeddings use negative values!

acc: 0.28
precision: 0.28
recall: 0.28
f1: 0.25
support: 1254



Multinomial NB + TF-IDF Embeddings

acc: 0.36
precision: 0.36
recall: 0.36
f1: 0.36
support: 1254



Experiment 4 Conclusions

- **Lyrics are likely poor predictors of emotion** compared to song positiveness or popularity.
- There appear to be quality issues with our data. The dataset authors used a fine-tuned transformer to originally generate create the emotion labels – the transformer might not have done so well.
- **Doc2vec underperformed TF-IDF slightly**, to my surprise - it might be worth experimenting with my Doc2vec setup, maybe training on more epochs or raising/lowering the feature count?
- **Multinomial NB outperformed Gaussian NB**, indicating the text features don't clearly form a Gaussian distribution.

- **Experiment 3/4 Impact**
  - **Determining emotion from lyrics would be useful to streaming services to automatically categorize their catalog**
  - **Knowing all songs' emotion would be useful for building song recommender systems**
  - **However, in general, powerful recommender systems ("algorithms") can lead to social media addiction**

Overall Impact Section

- Companies will be able to use the data modeling that we identified to help develop playlists and algorithms to help individuals and users identify the correct songs when looking for a specific emotional response
- Understand that factors that correlate with specific songs that are tied to song lyrics
- Healthcare professionals can use these suggestions for emotion to provide them to their patients or wellness apps designed to help individuals with their mental state
- Companies utilizing marketing can create emotional ad campaigns using the patterns identified with what songs are tied to specific emotions

Links/References

- Dataset
- Sklearn, pandas, NLTK documentation
- https://www.ibm.com/think/topics/naive-bayes
  https://stackoverflow.com/questions/13208286/how-to-write-latex-in-ipython-notebook
  https://www.datacamp.com/blog/curse-of-dimensionality-machine-learning
  https://www.geeksforgeeks.org/removing-stop-words-nltk-python/ https://en.wikipedia.org/wiki/Tf%E2%80%93idf
  https://medium.com/wisio/a-gentle-introduction-to-doc2vec-db3e8c0cce5e
  https://www.statology.org/pandas-find-first-row-that-meets-criteria/
  https://stackoverflow.com/questions/38840319/put-a-2d-array-into-a-pandas-series
  https://www.geeksforgeeks.org/tokenize-text-using-nltk-python/
  https://stackoverflow.com/questions/15547409/how-to-get-rid-of-punctuation-using-nltk-tokenizer
  https://stackoverflow.com/questions/66280588/class-weight-balanced-equivalent-for-naive-bayes

Thank you!