Song genre classification using Spotify and Genius data

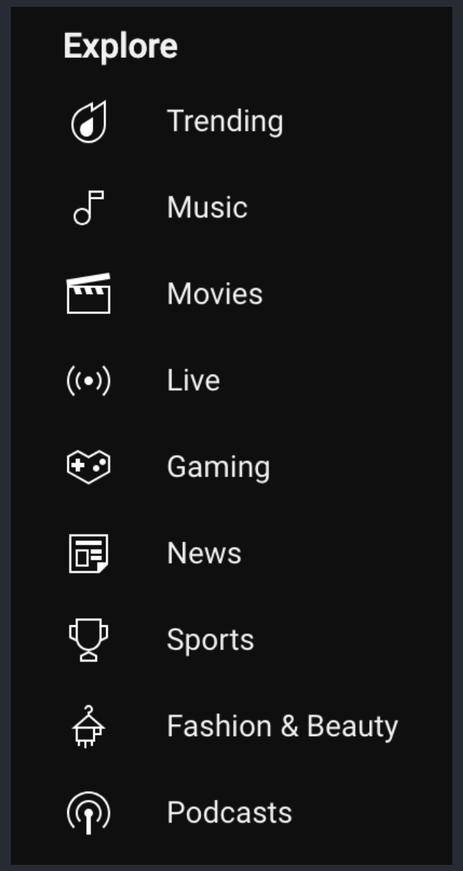
Marco Ballarini
Laura Legrottaglie
Pietro Renna

Introduction

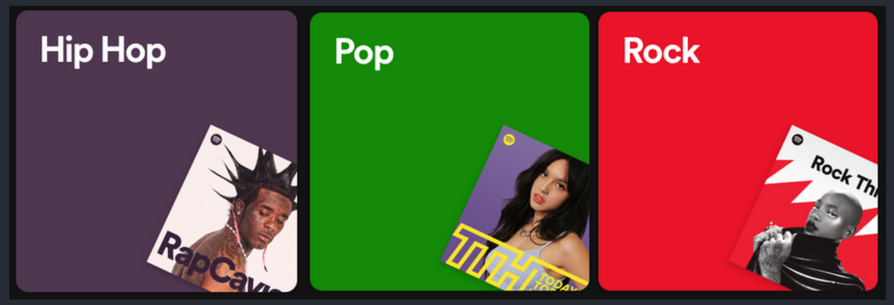
- Nowadays, the world is full of big data
- Streaming services add contents every month
- Contents have to be labeled in order to provide properly summarized informations, depending on the kind of service

Introduction

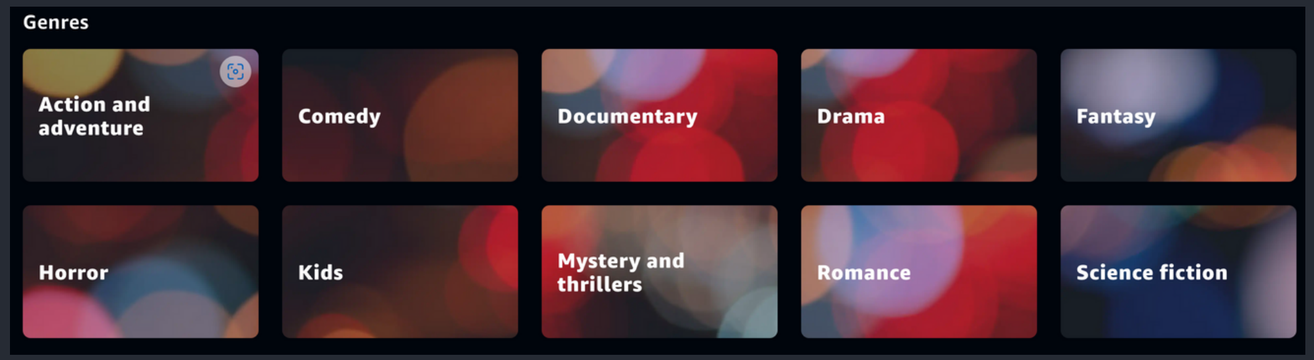
Youtube



Spotify



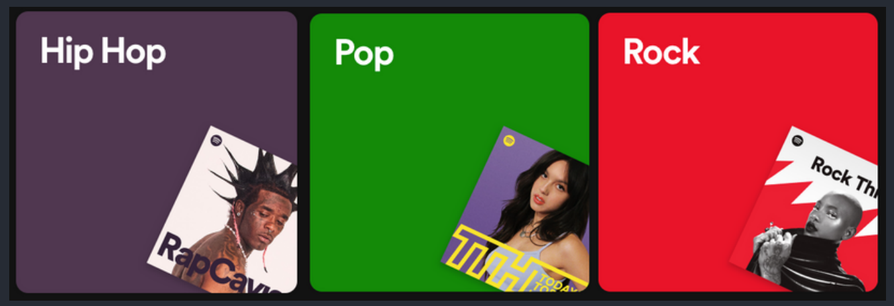
Prime Video



Introduction

Case of study for this work

Spotify



Introduction

How is labeling performed?

Manually (human)

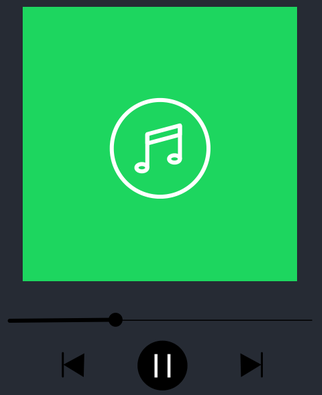
- 👍 Precise
- 👎 Takes lot of time

Automatically (classification model)

- 👍 Reduces the amount of work
- 👍 Helpful in a big data context
- 👎 May give the wrong label ⟶ find a good Model

Introduction

- (Multi-class) Classification of songs



Model

Pop

Rap

Rock

Note: for simplicity matter, the analysis considers only Pop, Rap and Rock as genres

Tasks

Interpretability

Are lyrics terms and audio features significant to determine the song genre?

Use models to check if there are any significant coefficients related to text terms

Are lyrics terms coherent with the genre for which they are significant?

Check whether the selected coefficients (and the related terms) are coherent with the genre

Prediction

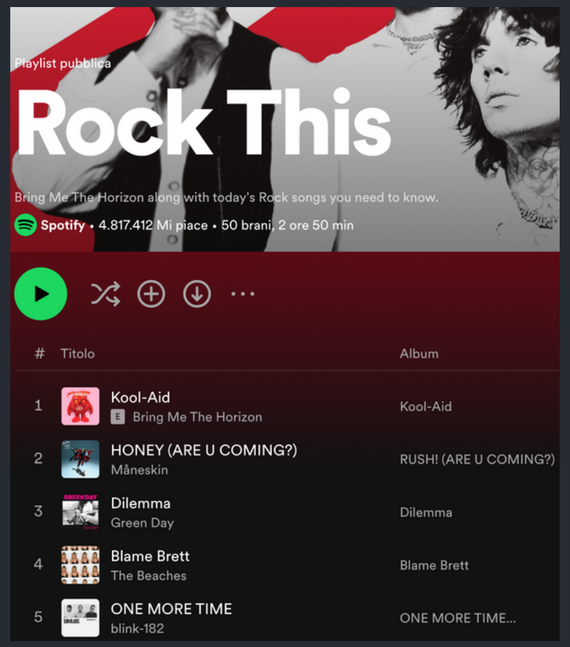
How do models behave in terms of predictions?

Compare different models using several metrics

General Overview

- Data Scraping
- Audio features
- Text Mining
- Models
- Conclusions

Data Scraping



Playlists

Spotify API

Audio Feature

Artist Name
Song Name

Dataset

Lyrics

Genius API

Audio features


Acousticness


Danceability


Duration (ms)


Energy


Instrumentalness

key


Liveness


Loudness

Mode


Speechiness


Tempo

Valence


Genre (Response variable)

Text Mining

Set of operations performed on textual data to make them cleaned and structured

1) TEXT CLEANING

- Transformation to lower case
- Removal of:
  - Artifacts related to the specific context (“Chorus”, “Lyrics”, “Contributors”, “Verse”)
  - Special characters
  - Punctuation
  - Numbers
  - Stopwords (using *stopwords-iso*)
- Lemmatization and Stemming

Text Mining

TEXT CLEANING: Example

- Transformation to lower case
- Removal of:
  1. Artifacts related to the specific context (“Chorus”, “Lyrics”, “Contributors”, “Verse”)
  2. Special characters
  3. Punctuation
  4. Numbers

"ContributorsTranslationsÐ ÑƒÑ□Ñ□ÐºÐ¸Ð¹TÃ¼rkÃ§eEspaÃ±ol PortuguÃªsItalianoHebrewFranÃ§ais Smells Like Teen Spirit Lyrics Load up on guns, bring your friends It's fun to lose and to pretend She's over-bored and self-assured Oh no, I know a dirty word..."

"smell like teen spirit load up on guns bring your friends it s fun to lose and to pretend she s overbored and selfassured oh no i know a dirty word..."

Lyrics from "Smells like Teen Spirit" by Nirvana

Text Mining

TEXT CLEANING

Lyrics from "Smells like Teen Spirit" by Nirvana

"smell like teen spirit
load up on guns bring your friends
it s fun to lose and to pretend
she s overbored and selfassured
oh no i know a dirty word..."

1 →

"smell teen spirit
load guns bring friends fun lose
pretend
overbored selfassured
dirty word..."

2 →

"smell teen spirit load gun bring friend fun lose pretend overbored selfassured dirty word..."

↓ 3

"smell teen spirit load gun bring friend fun lose pretend overbor selfassur dirti word..."

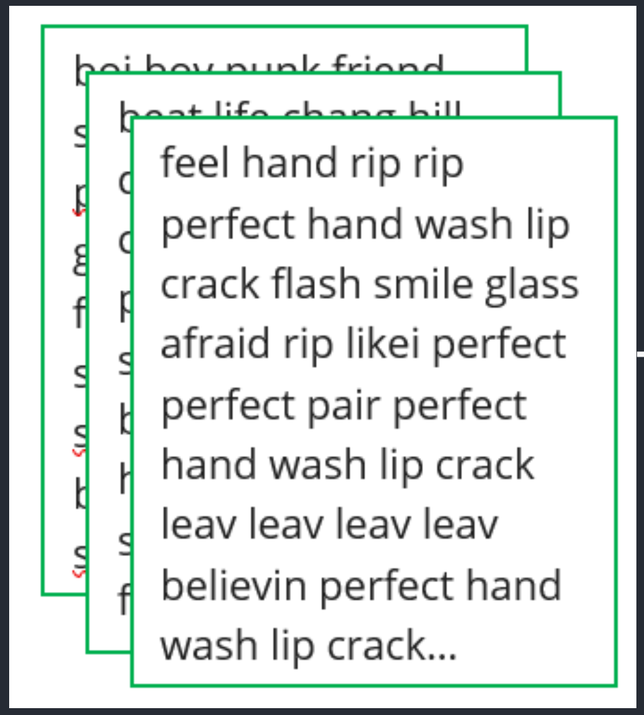
1. Removal of stopwords (using *stopwords-iso*)
2. Lemmatization (word -> word lemma or root)
3. Stemming (removal of suffixes or prefixes)

Text Mining

DATA WORKING

- Final dataset:
  - 2285 lyrics and 13554 predictors (high dimensional problem)
  - balanced across all genres
- Train and test split:
  - 80% training, 20% test
- Document Term Matrix on unigrams of the training set and the test set

Preprocessed lyrics



Document Term Matrix

	feel	hand	rip	...
S1	1	3	3	
S2	0	1	0	
S3	3	0	1	
...				

Text Mining

TF-IDF (Term Frequency-Inverse Document Frequency)

- Use TF-IDF instead of Term Frequency:
  - reduce the impact of common words
  - identify distinctive words between lyrics
  - balance term frequencies independently from lyric's size
- TF-IDF on the test set
  - Based on Train set vocabulary (to avoid data leakage)

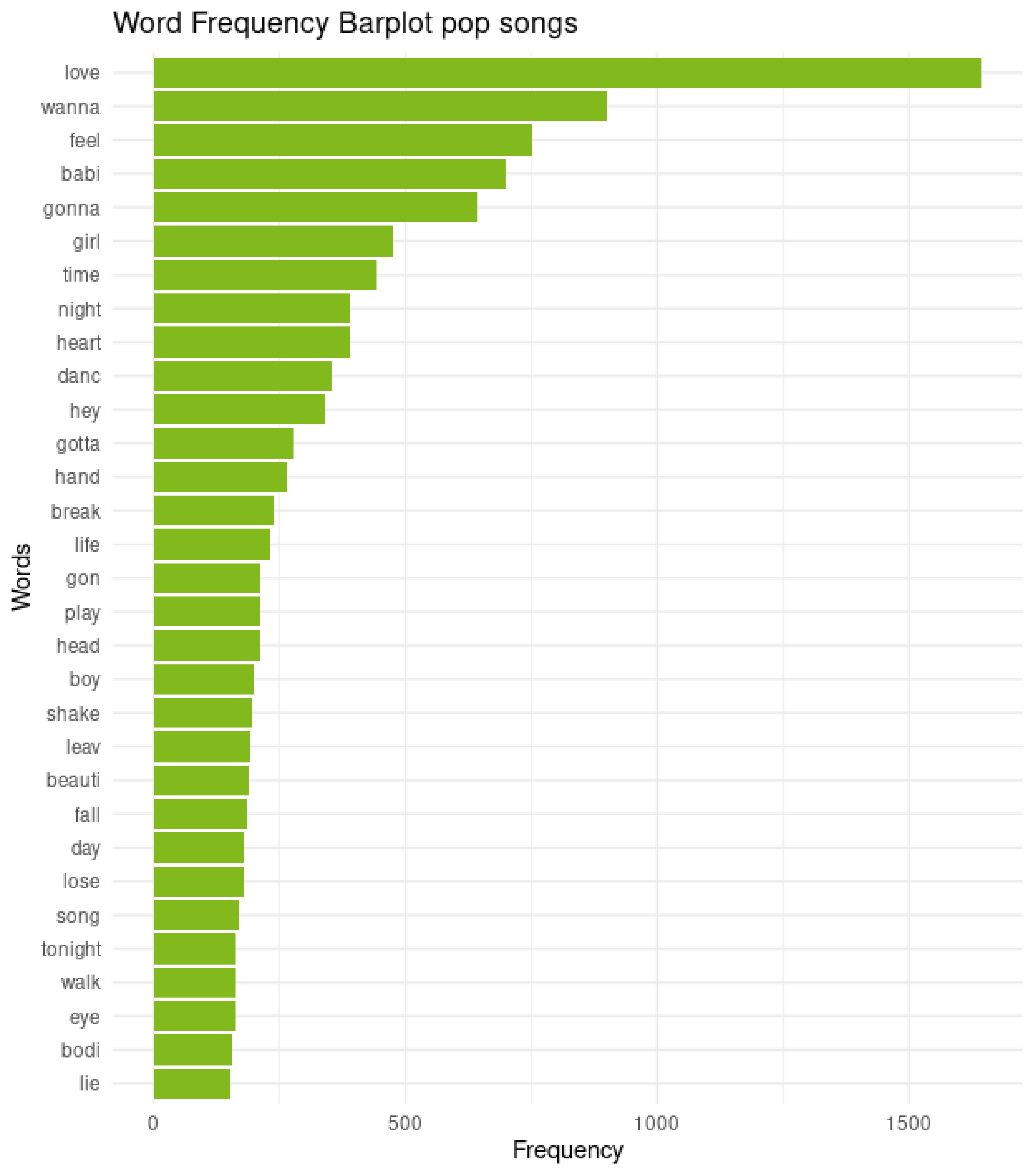
$$TF(t,d) = \frac{number\ of\ times\ t\ appears\ in\ d}{total\ number\ of\ terms\ in\ d}$$

$$IDF(t) = \log \frac{total\ number\ of\ documents}{number\ of\ documents\ with\ term\ t\ in\ it}$$

$$TF - IDF(t,d) = TF(t,d) * IDF(t)$$

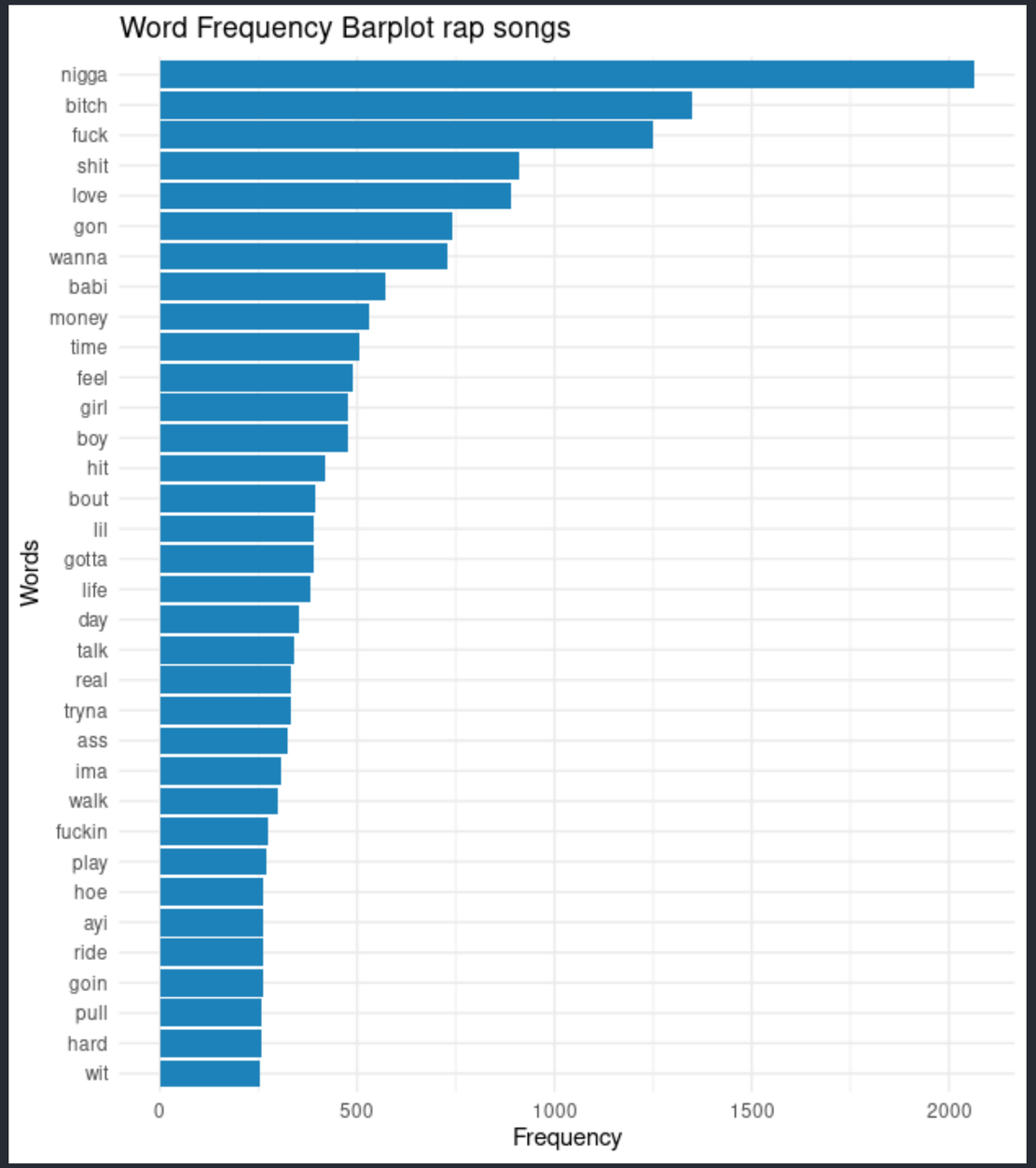
Text Mining

POP ANALYSIS



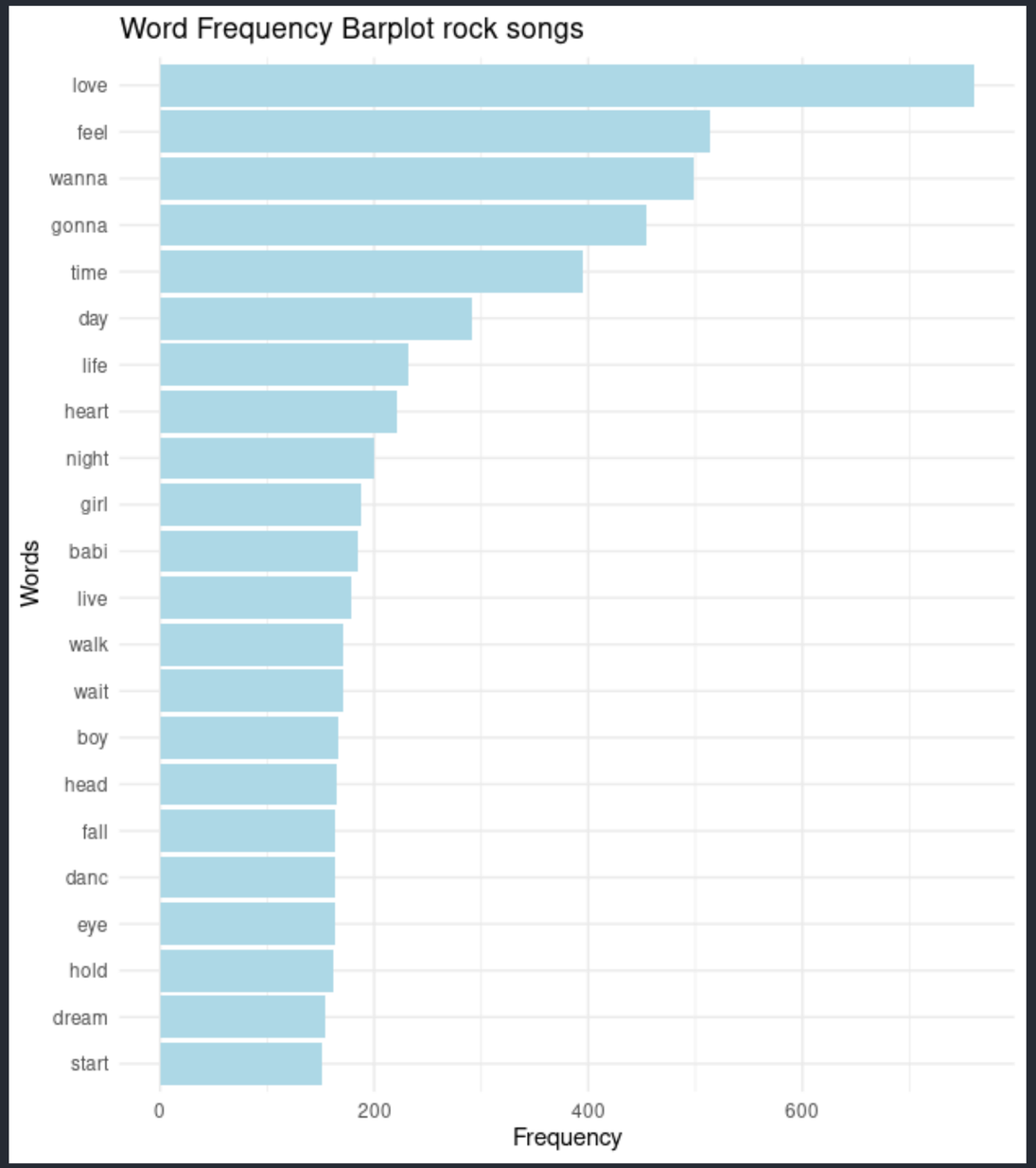
Text Mining

RAP ANALYSIS



Text Mining

ROCK ANALYSIS



Models - Shrinkage Methods

Multinomial Logistic Regression with Lasso constraint:

- 👍 Variable selection
- 👍 Interpretation of coefficients estimation
- 👎 Struggle to deal with highly correlated variables

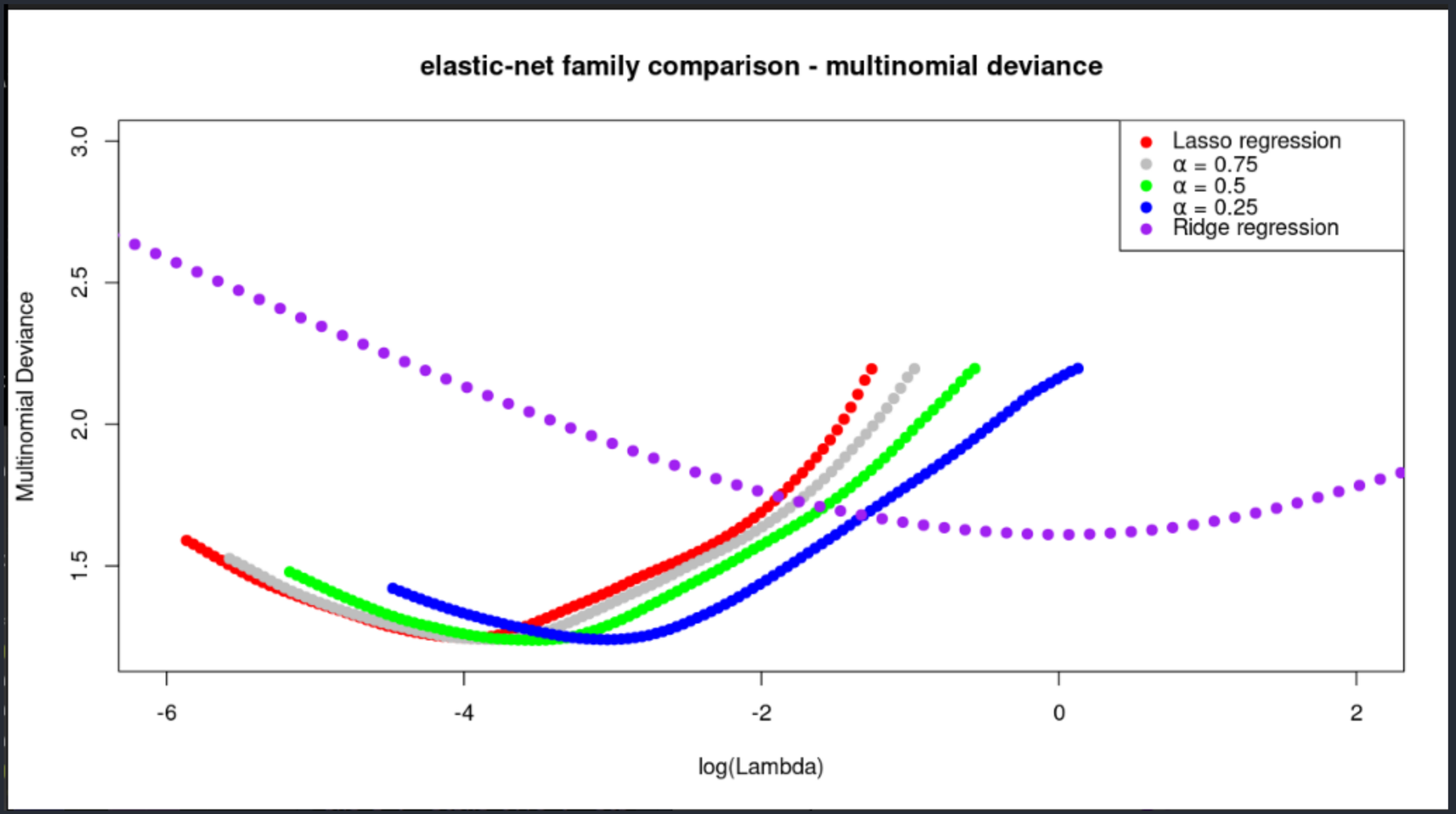
Multinomial Logistic Regression with Ridge constraint:

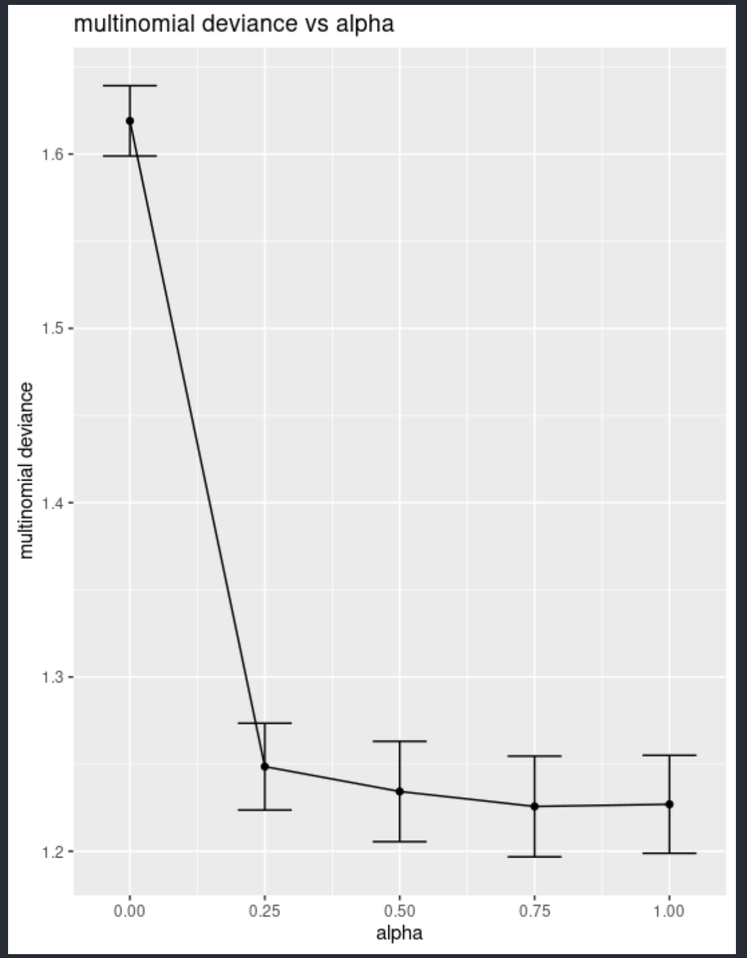
- 👍 Deal with highly correlated variables
- 👎 No feature selection

Multinomial Logistic Regression with Elasticnet constraint (α = 0.25,0.5, 0.75):

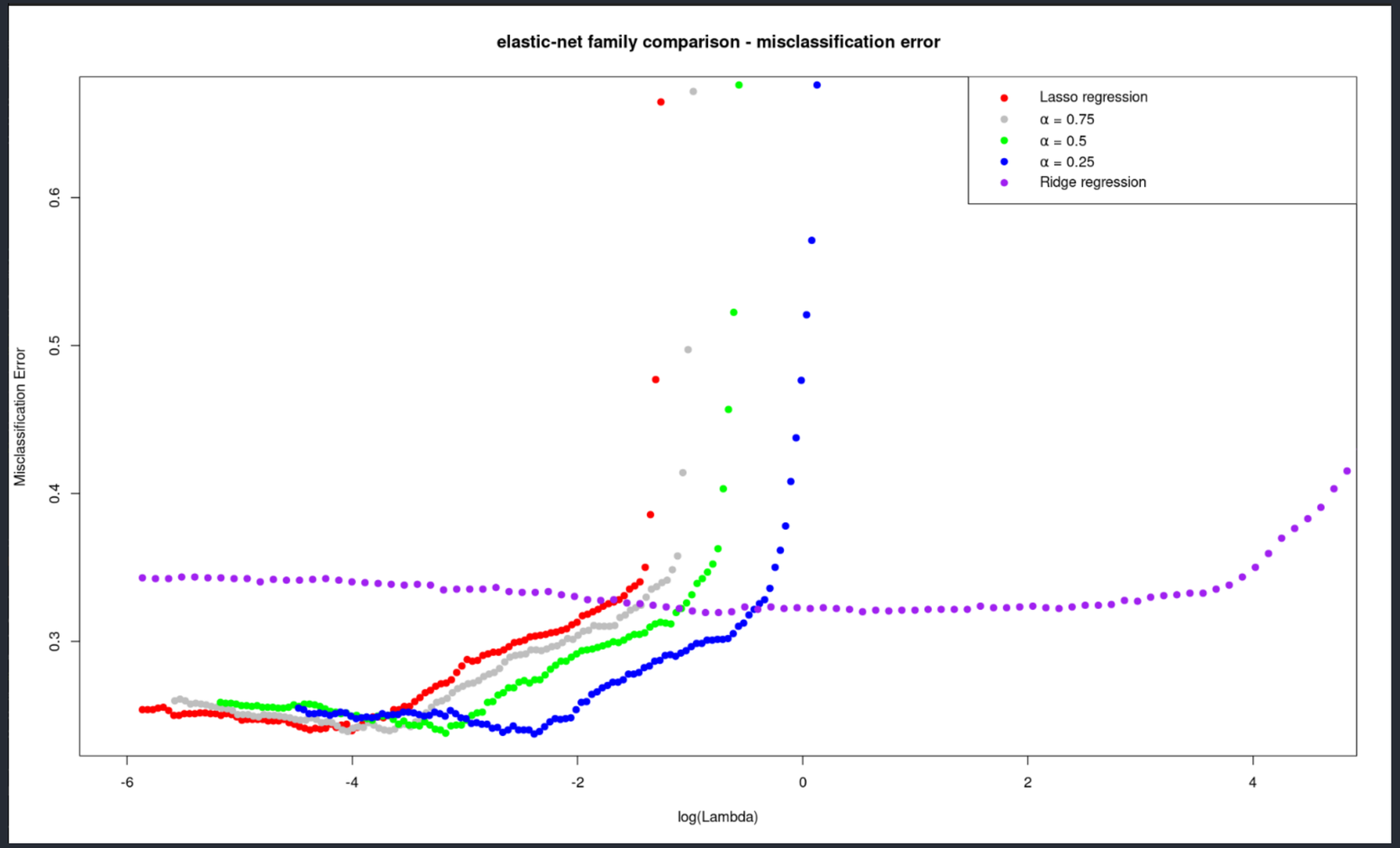
- 👍 Deal with highly correlated variables
- 👍 Variable selection

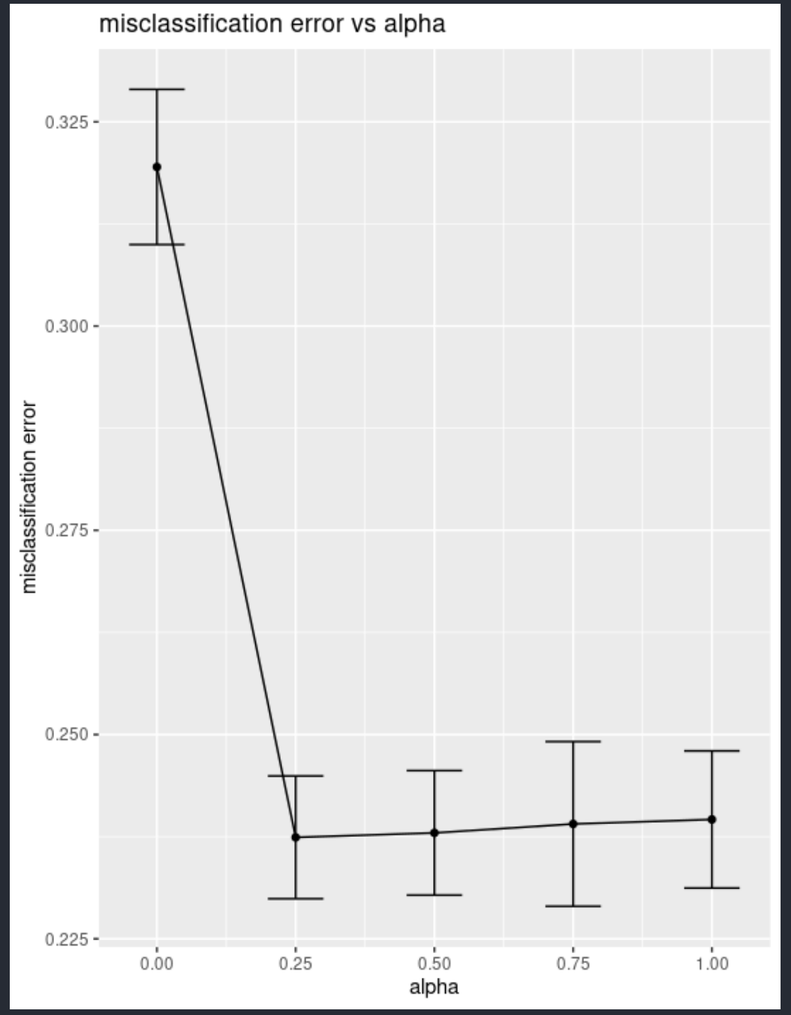
Models - Shrinkage Methods





Models - Shrinkage Methods



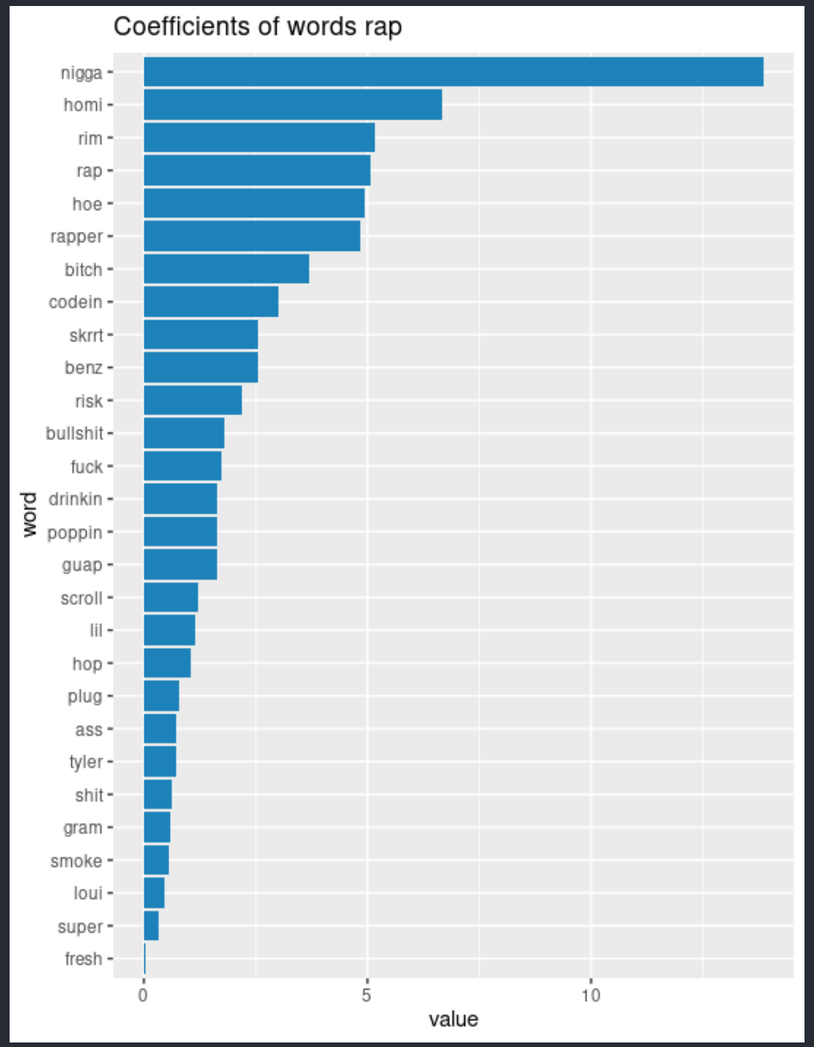


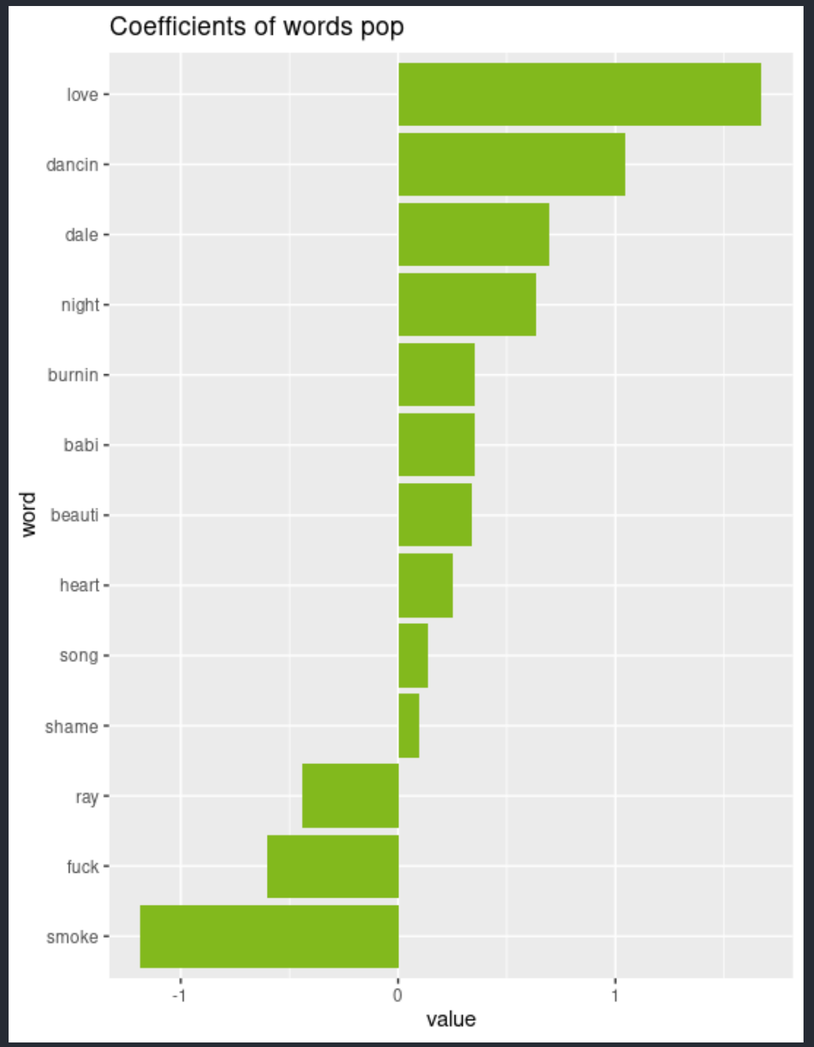
Models - Shrinkage Methods

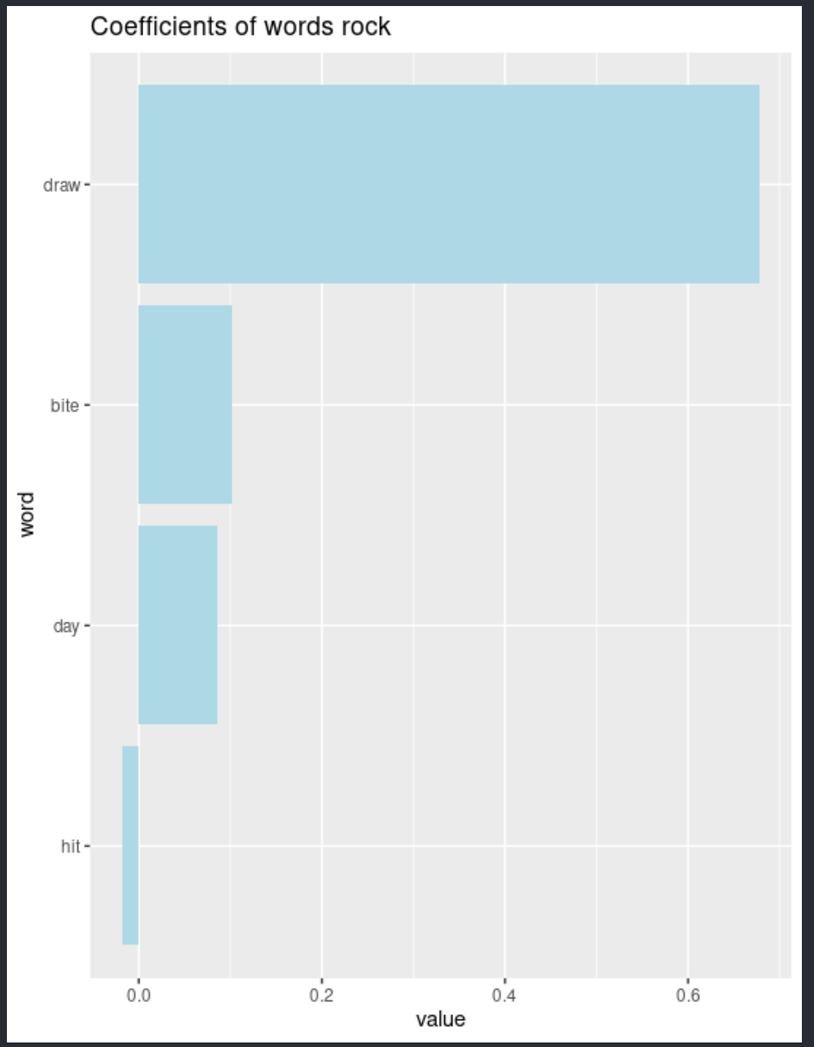
Selected model: Lasso Multinomial Logistic Regression with λ_{1se}
The choice of this model have two main reasons:

- It select less parameter than the one with λ_{min} , so it is more interpretable and less complex.
- Testing both of them against the test set it results that the λ_{1se} model presets slightly better results.

Models - Shrinkage Methods







Models - Shrinkage Methods

Audio features coefficients

VARIABLE	VALUE
Danceability	0.0671
Speechiness	0.195
Valence	-0.407

Rap

VARIABLE	VALUE
Loudness	0.0671
Acousticness	0.195
Duration	-0.000000654
Key D	-0.0353

Pop

VARIABLE	VALUE
Danceability	-7.39
Energy	1.63
Key E	0.0362
Loudness	-0.0540
Mode Major	0.118
Acousticness	-0.181
Instrumentalnes	0.0741
Valence	0.525
Duration	0.000000524

Rock

Models - Graphical Lasso

In order to push a little further the interpretation of variables selected by lasso.

It could be interesting to search if there is a dependency structure between them. Maybe it’s possible to see that the presence of some words is conditionally independent from the presence of other words, or viceversa.

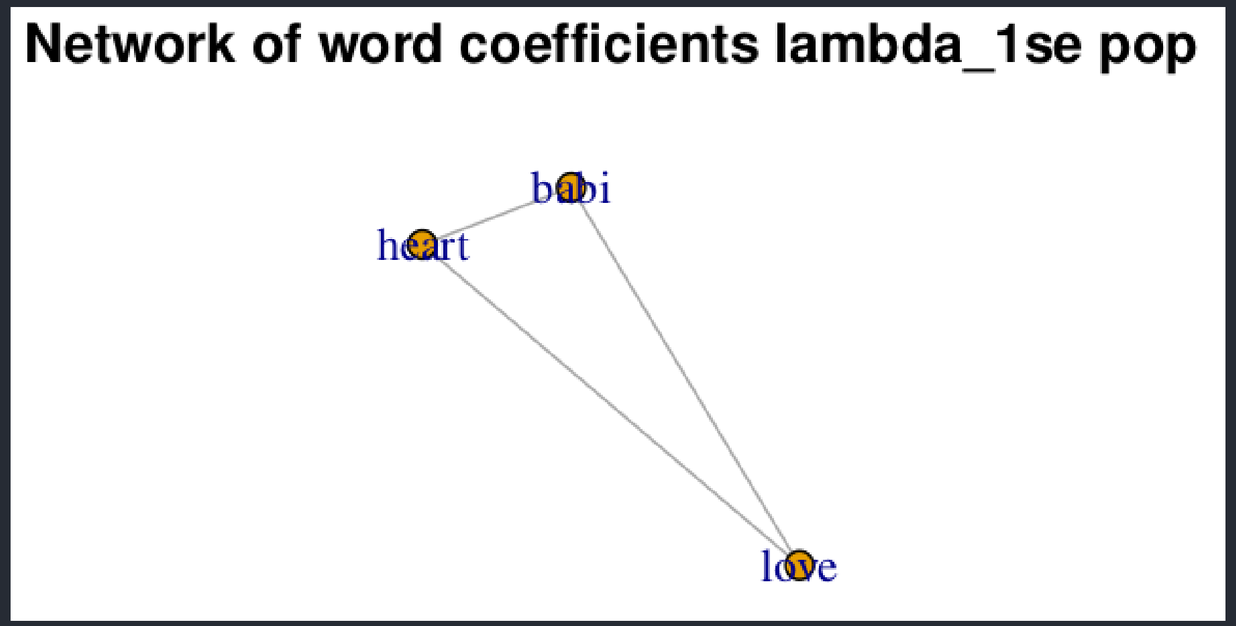
To do so a Graphical Lasso on TF-IDF matrix of words selected by Lasso is performed.

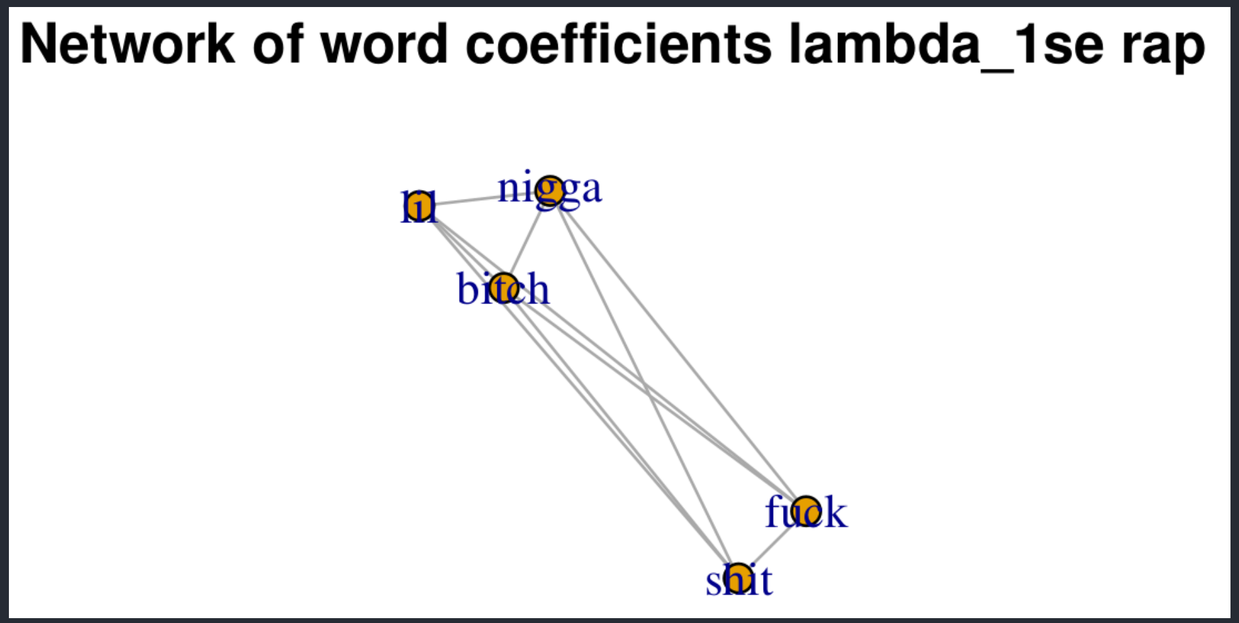
Models - Graphical Lasso

Then the resulting estimated covariance matrix is used to build a network graph: words are represented as vertex, if the estimated covariance between two words is higher than zero, there will be an edge connecting those words, otherwise no edges

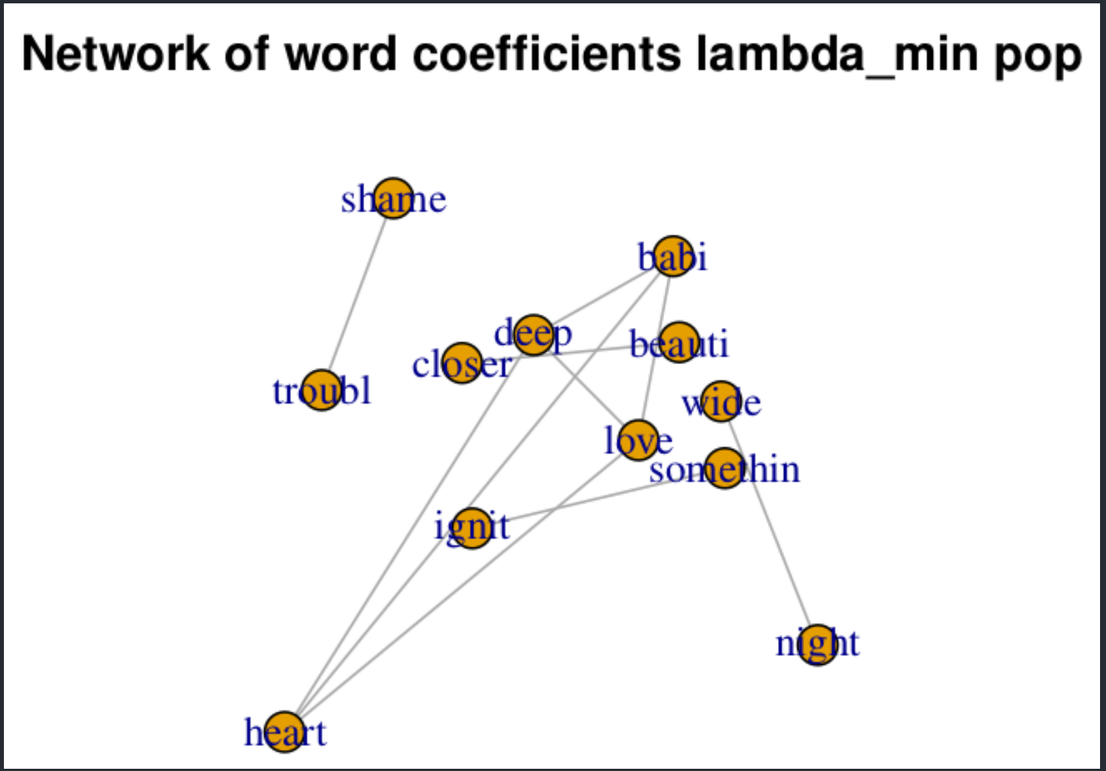
In following plots only words with at least one edge (not including self-loop).

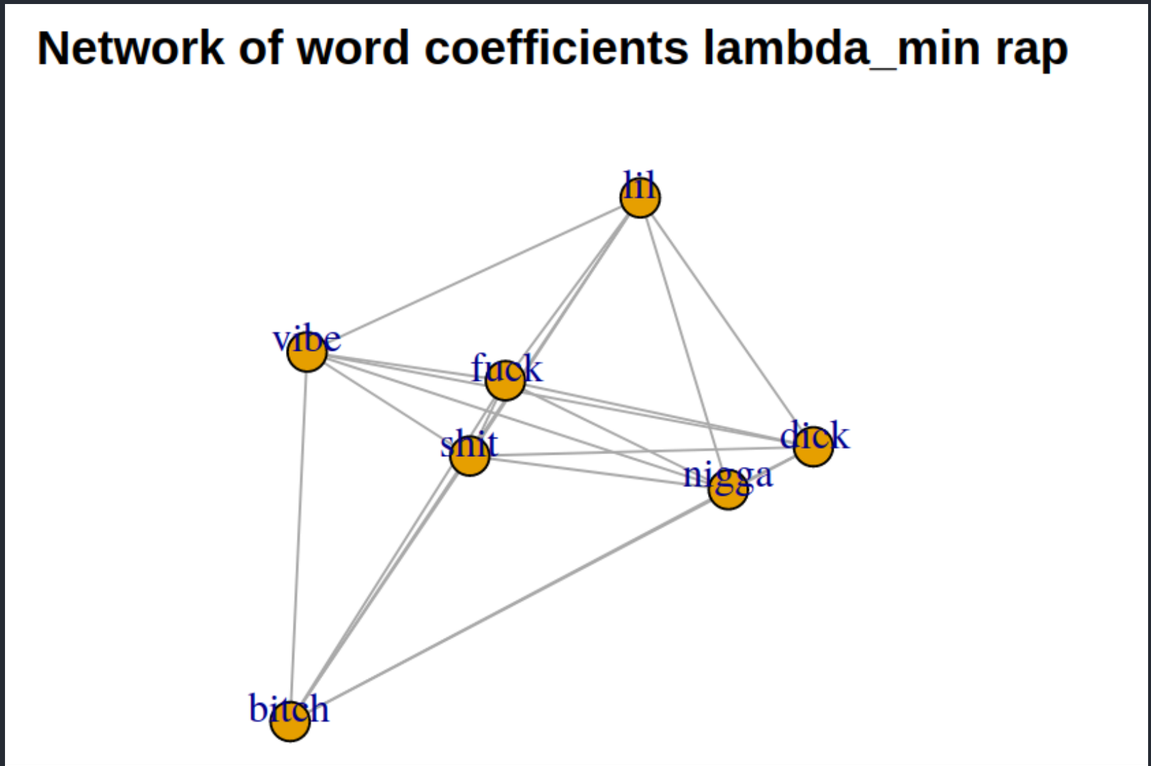
Models - Graphical Lasso





Models - Graphical Lasso





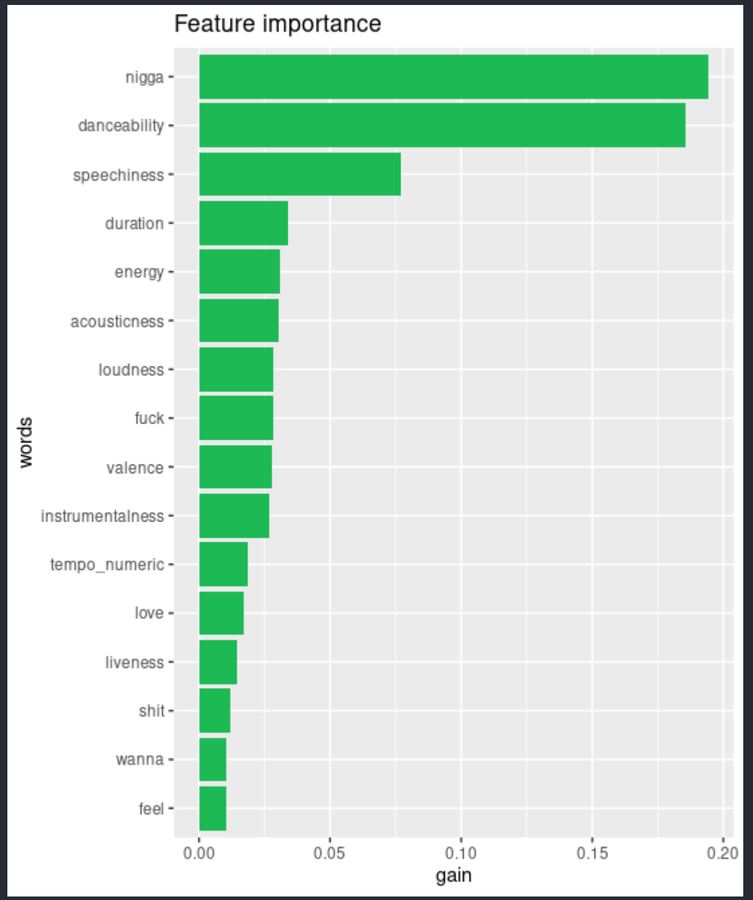
Models - XGBoost

- Boosting approach based on classification trees
- Can be used for both interpretability and prediction tasks
- Tuned using 10-fold Cross Validation on:
  - *max-depth (12)*
  - *learning rate (0.2)*

Models - XGBoost

Interpretability

- *Feature importance:* based on how many times (occurence) a predictor is used to perform splitting in the trees
  - It gives a global measure of significance (i.e., not related to a specific genre)
- Audio features are significant
- Significant terms that appear also in Lasso:
  - “Love”
  - “Nigga”
  - “Fuck”



Models - Neural Network

Word Embedding

TF-IDF matrices are inefficient as input of NN due to:

- Sparsity
- Curse of dimensionality
- Inability to capture semantic meaning of words

Solution: mapping words into a vector space

"love" ⟶ [-0.895 0.411 ... -0.212]

"dancin" ⟶ [-0.129 0.166 ... -0.321]

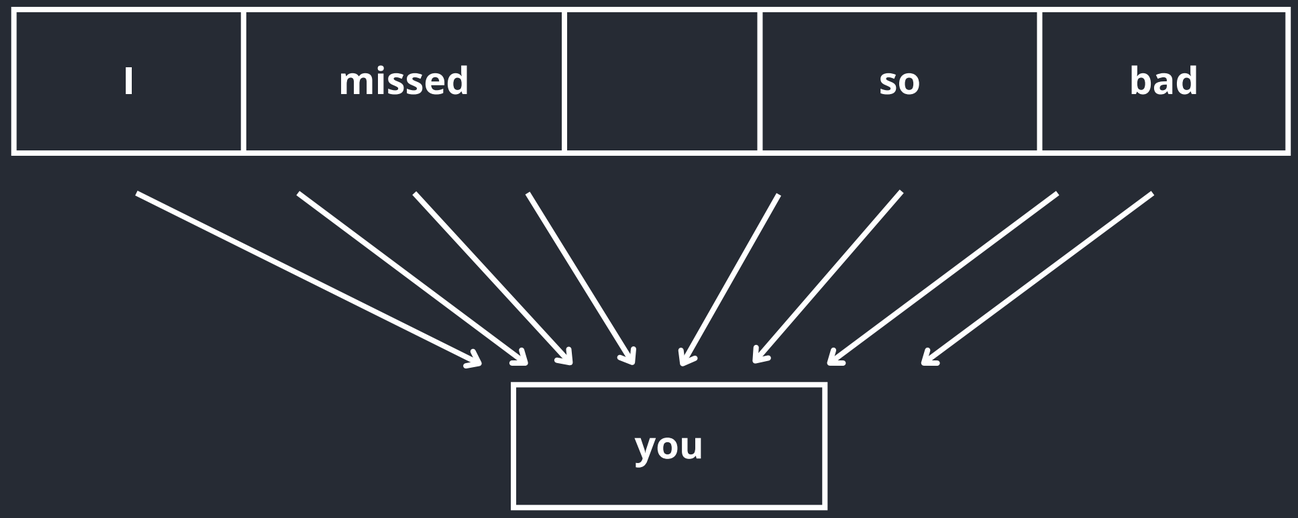
"nigga" ⟶ [0.129 0.382 ... -1.114]

Word2Vec

- Word2Vec is a two-layer neural network that generates word embedding given a text corpus.
- **Objective:** semantic similar words (similar context) will have similar vector representation
- Chosen architecture:

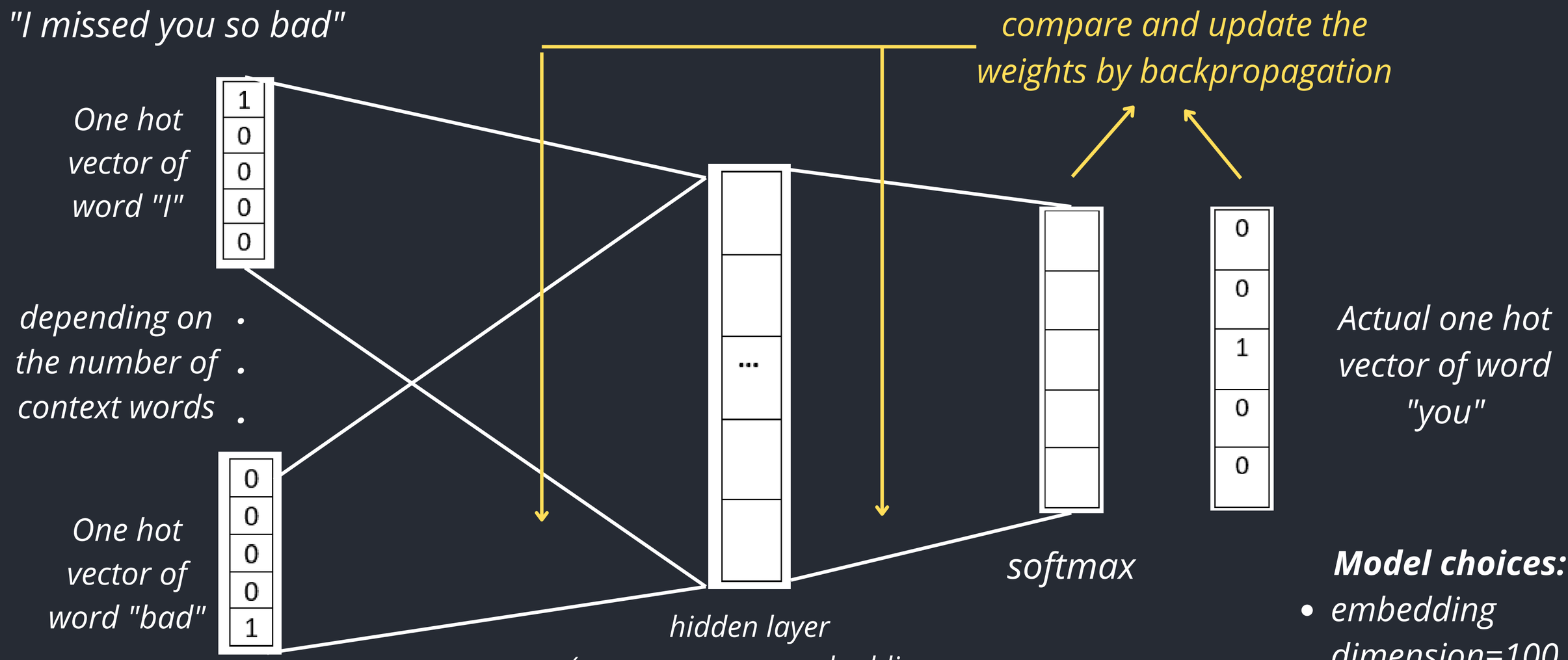
Continuous Bag of Words (CBOW)

Predicting the target word from the context



From *"Call Me Maybe"* by Carly Rae Jepson

Simplified representation of CBOW

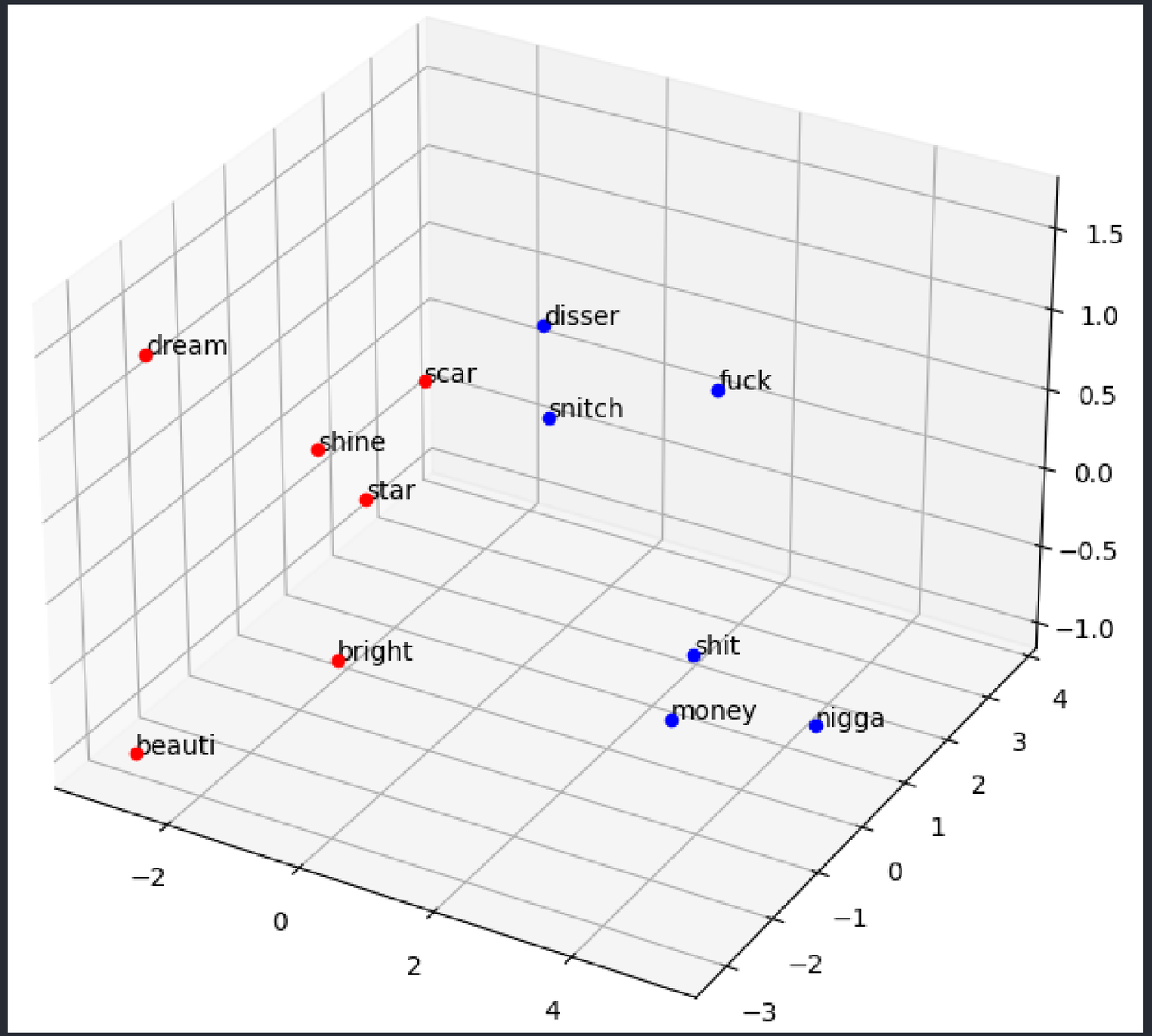


Model choices:

- *embedding dimension=100*
- *window context=5*

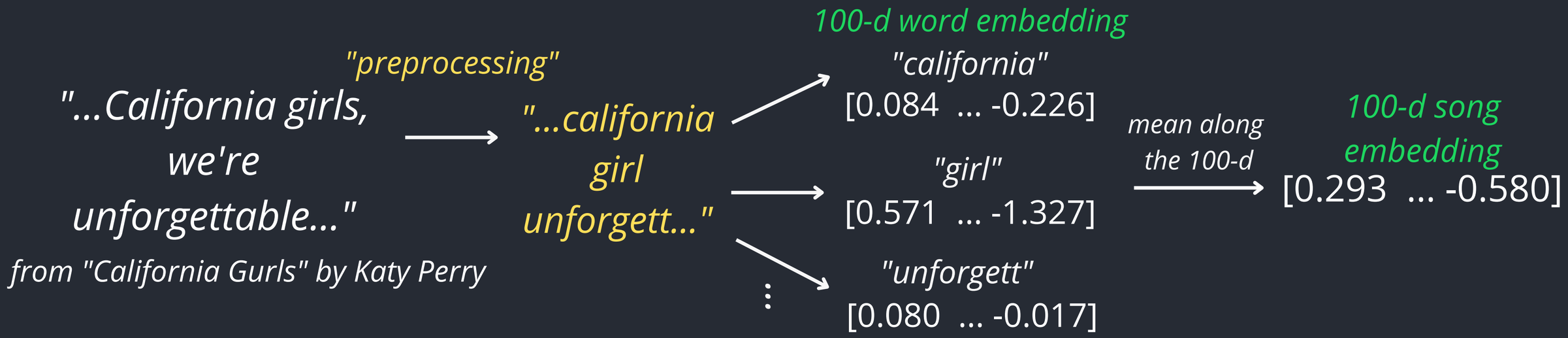
Word embedding for words "beauti" and "nigga"

Starting from a 100-word embedding representation, using PCA, the number of dimensions is reduced to 3 only for visualization purposes. 5 most similar words to "beauti" and "nigga" are plotted



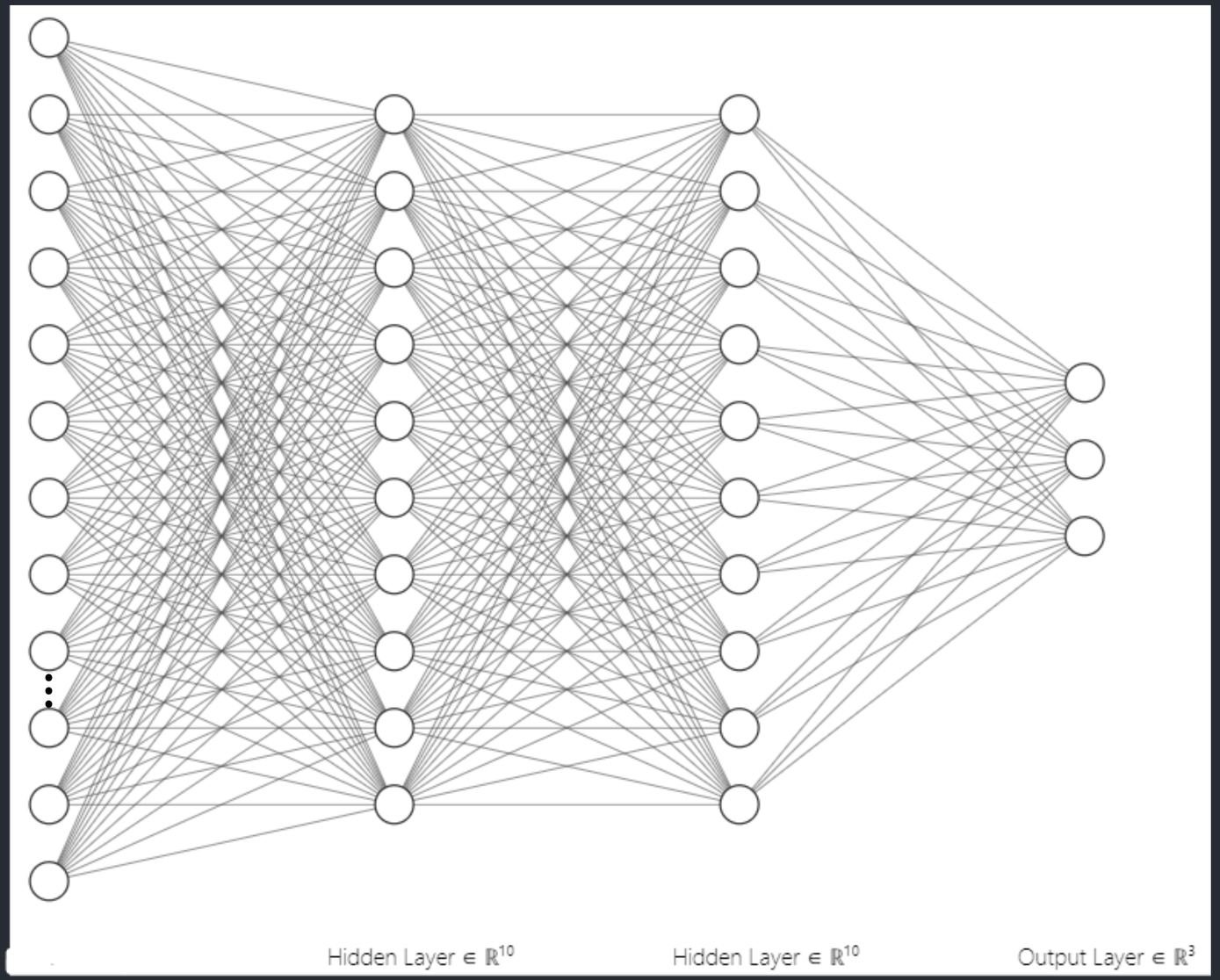
Highlights: Neural Network

- **Step 1:** each word is converted into a 100-dimensional vector
- **Step 2:** to handle with different length songs, each song in the corpus is converted into a 100-dimensional vector (mean of the vectors' words in it)



- **Step 3:** the final input of the NN is the combination of numerical features and lyrics vector
- **Step 4:** cross validation

Highlights: Neural Network



Best model: 2-hidden layers with 10 neurons each, learning-rate=0.001 and ADAM as optimizer

Advantages: good performances

Disadvantages: lack of interpretability

Conclusions
Interpretability

- Text predictors:
  - many significant and reasonable words in pop and rap
  - few and not reasonable words in rock
- Audio features:
  - few significant predictors for pop and rap
  - many significant predictors for rock

Conclusions
Predictions

Model	Accuracy
Lasso (λ_{1se})	0.709
Neural Network	0.779
XGBoost	0.781

Model	Sensitivity	Specificity	Precision	F1 Score
Lasso (λ_{1se})	0.7754	0.7116	0.5377	0.6350
Neural Network	0.7391	0.8150	0.6335	0.6823
XGBoost	0.8116	0.7837	0.6188	0.7022

Metrics for class Pop

Model	Sensitivity	Specificity	Precision	F1 Score
Lasso (λ_{1se})	0.8529	0.9252	0.8286	0.8406
Neural Network	0.8015	0.9533	0.8790	0.8385
XGBoost	0.6397	1.0000	1.0000	0.7803

Metrics for class Rap

Model	Sensitivity	Specificity	Precision	F1 Score
Lasso (λ_{1se})	0.5519	0.9380	0.8559	0.6711
Neural Network	0.7923	0.9015	0.8430	0.8169
XGBoost	0.8634	0.8869	0.8360	0.8495

Metrics for class Rock