

# Emotion Patterns in Music Playlists

Sara Giammusso<sup>12</sup>    Mario Guerriero<sup>12</sup>

<sup>1</sup>MSc student in Data Science Department, EURECOM, Télécom ParisTech, France

<sup>2</sup>MSc student in Department of Control and Computer Engineering, Politecnico di Torino, Italy

Sixth Project meeting

# Table of Contents

- 1 Introduction
- 2 Last Week Summary
- 3 Feature Selection
- 4 Playlists Classification
- 5 Conclusion

# Table of Contents

- 1 Introduction
- 2 Last Week Summary
- 3 Feature Selection
- 4 Playlists Classification
- 5 Conclusion

# Introduction

## Previously On Sara&Mario Project...

We worked on feature engineering to see how we could improve our classifiers

Next steps:

- Make sure the POS tagger we are using is good
- Make some experiments with other datasets
- Keep working on feature engineering
- Start looking at playlists

# Table of Contents

- 1 Introduction
- 2 Last Week Summary**
- 3 Feature Selection
- 4 Playlists Classification
- 5 Conclusion

# POS Tagger Analysis

- We are using spaCy's built-in POS tagger
- There is no authoritative paper regarding its internal details
- Our experiments suggest that it is good for our purpose <sup>1</sup>
  - Each line is treated as a separate sentence
  - The tagger is smart enough to recognize less common words and abbreviations (very common in songs)

---

<sup>1</sup>Please refer to the following Jupyter Notebook to know more about what we did on this topic: [https://github.com/sgiammy/emotion-patterns-in-music-playlists/blob/master/src/POS\\_tagger\\_verification.ipynb](https://github.com/sgiammy/emotion-patterns-in-music-playlists/blob/master/src/POS_tagger_verification.ipynb)

# More on Feature Engineering

- We had built a big list of features in the past few weeks
- We concluded that we do not need all of them (of course)
- We got the best performances just by using a subset of 10 of them <sup>2</sup>
  - WORD\_COUNT, ECHOISMS, SELFISH\_DEGREE, DUPLICATE\_LINES, IS\_TITLE\_IN\_LYRICS, VERB\_PRESENT, VERB\_PAST, VERB\_FUTURE, ADJ\_FREQUENCIES and PUNCT\_FREQUENCIES

---

<sup>2</sup>For more information, please refer to

[https://github.com/sgiammy/emotion-patterns-in-music-playlists/blob/master/src/Advanced\\_Feature\\_Engineering.ipynb](https://github.com/sgiammy/emotion-patterns-in-music-playlists/blob/master/src/Advanced_Feature_Engineering.ipynb)

# MoodyLyrics + EmoInt

- We are very limited by MoodyLyrics<sup>3</sup>
- EmoInt is made of tweets classified according to four emotions: anger, fear, joy and sadness
  - We have mapped anger to angry, joy to happy, sadness to sad and discarded fear
- We were able to increase our dataset by 2254 items
- No good results at the end

---

<sup>3</sup><https://github.com/sgiammy/emotion-patterns-in-music-playlists/blob/master/src/EmoInt.ipynb>



# Table of Contents

- 1 Introduction
- 2 Last Week Summary
- 3 Feature Selection**
- 4 Playlists Classification
- 5 Conclusion

# Two More Features

We wanted to expand our model with two features

- Sentiment/Polarity
- Subjectivity

To achieve this goal we used TextBlob<sup>4</sup>

- Built on top of NLTK and Pattern
- Makes use of Pattern's analyzer to generate polarity and subjectivity

At the end we will have 12 features (311 in reality, because the lyrics content vector is expanded)

---

<sup>4</sup><http://textblob.readthedocs.io/en/dev/index.html>

# Obtained Results: ANN

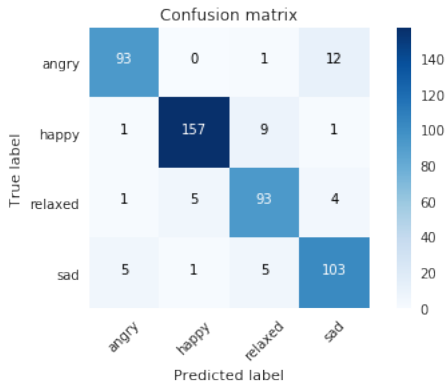


Figure 1: 90.84% accuracy on MoodyLyrics

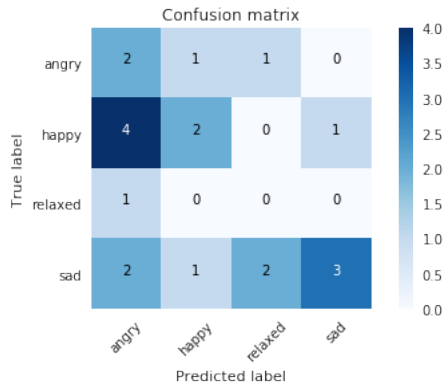


Figure 2: 50% accuracy on extra test

# Obtained Results: Logistic Regression

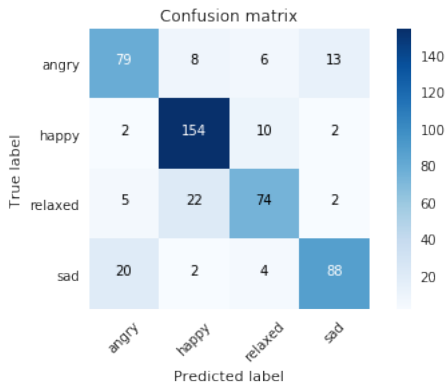


Figure 3: 80% accuracy on MoodyLyrics



Figure 4: 55% accuracy on extra test

# Table of Contents


- 1 Introduction
- 2 Last Week Summary
- 3 Feature Selection
- 4 Playlists Classification**
- 5 Conclusion

# Which classification model?

We considered different approaches

- Give each song in the playlist an emotion label and choose using a majority rule
- Consider the text of all the songs in a playlist as belonging to a single song and classify it
- Compute an "emotion vector" for each song and average all of those vectors over the playlist<sup>5</sup>
  - This is the one which makes more sense to us
  - It produces 4 playlist features (one per emotion) you may want to use in your recommendation system

---

<sup>5</sup>Remember the sliders approach? We got inspired from this 

# Our results

- Playlists classified using our best model (neural network) all appears to be very similar to each other
- Logistic Regression provided more heterogeneous classifications
- We can't validate what we did :-)
- Could you please test the feature we extracted into your recommendation system?

# Table of Contents

- 1 Introduction
- 2 Last Week Summary
- 3 Feature Selection
- 4 Playlists Classification
- 5 Conclusion**



# What's next?

- Can we start writing our report? (if yes, any hint?)
- ...