🎼

Project Introduction

Tasks	Understand project steps
Day	Monday

Your Project for the Week

Goals:

- Scrape artists' lyrics from a website and save them locally.
- Build a model that can predict the artist given some song lyrics.
- Make it into a python program that a user can interact with.

Data:

- You get the data yourself, by scraping e.g. lyrics.com.
- Big part of the week — you will spend a lot (most) of your time on this.

Model:

- A lot of the models that you already know work for this use case:
    - Logistic regression
    - Decision trees
    - Random forest
    - 🆕 Naive Bayes
- Feature engineering is an important step that allows us to run these models we already know on unstructured data like text (but don't worry — feature engineering is in a way much simpler than in previous weeks!).

Steps:

- Download HTML pages:
  - think about two artists you like — should sing in the same language, not too similar
  - download their song lists page from lyrics.com and save it to a file
- Get a list of song urls:
  - examine the song lists page in a text editor
  - find where the links to individual songs are
  - use *regular expressions* (or *BeautifulSoup*) to automate extracting song links from the song lists page
- Extract lyrics from song urls:
  - loop through the list of song links you extracted
  - download each song to a file locally
    - tip: one folder per artist, one file per song
    - tip: track your progress (with print statements or `tqdm`)
  - extract lyrics from html file
- Get data into tabular form:
  - your `x` will be a list of strings, each string representing one song
  - your `y` will be a list of artists, labels
- Feature engineer your data:
  - convert lyrics to numbers/features by vectorizing them
- Train a classification algorithm:
  - LogReg, trees, forests, or Naive Bayes
- Balance out your dataset
- Write a command-line interface

```
X = []
y = []
for artist in artists:
  for song in songs:
    X.append[song.text]
    y.append[artist]
```