・制作期間：2022/9~

〓〓〓〓〓〓〓〓〓〓〓〓〓〓〓〓

：概要：更新2022/9/18

学園祭までに音楽分析ツールを作ります。

雰囲気の分析までをどうにか形にしたい。

〓〓〓〓〓〓〓〓〓〓〓〓〓〓〓〓

◆作りたいツールのコアテーマ：

┣音楽を「分析」し、音楽趣向を「解析」

┗今回は「分析」することを到達目標とする。

◆音楽分析の内容：

┣BPM

┣曲のキー

┣曲の雰囲気(要素：軽快0、清澄1、激動2、情熱3、哀愁4、情緒5、安静6、恐怖7)

┗→やることはBPM、キー、雰囲気、結果画面で絞る。雰囲気を優先して作成しようと思う。最悪キーはやらない。

◆使えそうな手段：

┣キー解析：教師あり学習 分類

┣雰囲気解析：教師あり学習 重回帰分析

┗librosa(BPMの関数はすでにある)

◆データ収集方法：

┣歌詞は無視して判断すること

┗訓練データ80曲、テストデータ20曲

├自前：有名サブカル：有名アイドルグループ：有名商標：洋楽=1:1:1:1:1

└学習で使う秒数は1曲の8分の1で、これで1曲と換算。曲の中でどの部分を学習で使うかはランダム。

◆曲のキー分析方案：

┣周波数成分をオクターブごとに分割、成分の多い7音から分析。

┗メジャーかマイナーかは最後の音階で決定？

◆曲の雰囲気の分類方案：

┣BPMやキーは現時点で検討材料にはしないことにする。

┣判断材料：各音響特徴量と正解データ間の関連性学習

┣正解データを手作業で与えるときにcsvファイルに保存する。

├最大値は100として連続値で正解データを作る。

└出力値は加工しない。結果画面では適当に10段階にしておく。

┣推論段階：音楽データを長さで8等分し、それぞれ独立して8回分析することで、楽曲内での変化を追う。

┣雰囲気要素ごとに別の学習プログラムとして設計する。全部で8個のプログラムを学習させる。

┣キー解析を先に作った場合：キーになった7つの音階以外にあたる周波数成分は重みを軽くし、オクターブごとに切り貼りした7つのデータをそれぞれ解析学習することによって簡易的なものにできないか？(短時間フーリエ解析使用)

┗分析結果の表示方法：分割処理したデータを集めて時間ごとに複数個のレーダーチャート。

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

◆進捗

・BPM算出→動作確認済み

┗精度：2分の1、2倍、±2

・楽曲データの自動トリミングと出力→動作確認済み

・現在の状態：データ収集作業

┗悩み：偏りのないデータ収集

・次の状態：csv手作業入力

・感想：sklearnの関数便利すぎてあたまわるくなりそう。1行で各種アルゴリズムが書けるんですかぁ…。

〓〓〓〓〓〓〓〓〓〓〓〓〓〓〓〓

・参考⑥：[librosaで楽曲のテンポ（BPM）を推定してみる - プログラム日記φ(．．) (hateblo.jp)](https://memomemokun.hateblo.jp/entry/2019/09/09/162341)

┗BPM算出

├コピペしてきて、floatをintに変えて、複数ファイルのテンポを配列で出力した。np.appendとfor使いました。

├精度：2分の1、2倍、±2

└「雰囲気」を決めるときの要素の1つにしたいからBPMは出そうと思ってたんだけど、そもそも要素として寄与するんでしょうか？

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

・参考⑦：[曲調の種類や例（音楽の種類にも通じるジャンル的曲調の分類・情緒的曲調など） | うちやま作曲教室 (sakkyoku.info)](https://sakkyoku.info/tips/tune-style-01/)

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

・参考⑧：[楽曲のムードを深層学習で分類するレシピ (axross-recipe.com)](https://axross-recipe.com/recipes/86)

┣結構やりたいことに近いことが書かれてある記事を見つけた。ログインしないと見れない。

┣雰囲気分析の方法は、ここに書かれてあることと、アノ本に書かれてあることを組み合わせれば実現することができそう。

┣5割くらいしかわからないから自分でも頑張るけどまた助けてください。

┗・参考⑨：[MFCC（メル周波数ケプストラム係数）入門 - Qiita](https://qiita.com/tmtakashi_dist/items/eecb705ea48260db0b62)

├⑧の1次元分類モデルの章でfor文を使いmfcc(音響特徴量の一種)の要素を増やしていますが、⑨により、mfccの代わりにログメルスペクトログラムを分析要素にする試みを考えています。

└⑨内の「メルスペクトラム(log-mel spectrum)」はおそらく「ログメルスペクトログラム（Log-melspectrogram）」と同義かと思われます。

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

・参考⑩：[［機械学習 × 音楽］MIRとよく使われる特徴量 (acceluniverse.com)](https://www.acceluniverse.com/blog/developers/2019/11/-cd-mir-mirmusic-information-retrieval-ismirmir201920.html)

・参考⑪：[音声分類の前処理はメルスペクトログラムでOK？　従来のMFCCより高精度な画像分類ベースの論文の紹介 - Qiita](https://qiita.com/koshian2/items/ca99b4a489d164e9cec6)

・参考⑫：[MFCCと音色 – blog｜メディア情報研究室｜村上真研究室｜東洋大学総合情報学部 (makotomurakami.com)](http://makotomurakami.com/blog/2020/06/12/5756/)

・参考⑬：[LibROSA で MFCC（メル周波数ケプストラム係数）を算出して楽器の音色を分析 - Wizard Notes (wizard-notes.com)](https://www.wizard-notes.com/entry/music-analysis/insts-timbre-with-mfcc#%E6%99%82%E9%96%93%E5%B9%B3%E5%9D%87%E3%81%97%E3%81%9FMFCC%E4%BF%82%E6%95%B0%E3%81%AE%E6%AF%94%E8%BC%83)

┗⑩、⑪、⑫、⑬は⑧、⑨の周辺資料。

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

・参考⑭：[asj.pdf (kthrlab.jp)](http://www2.kthrlab.jp/members/kawamura/kawamura_site/asj.pdf)

┗先行研究の論文。

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

・参考⑮：[scikit-learn で線形回帰 (単回帰分析・重回帰分析) – Python でデータサイエンス (plavox.info)](https://pythondatascience.plavox.info/scikit-learn/%E7%B7%9A%E5%BD%A2%E5%9B%9E%E5%B8%B0)

┗⑧は分類問題なので、補強として重回帰分析用

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

・参考⑯：[【Python/pydub】mp3、wavファイルから部分抽出（切り分け、分割） (joho.info)](https://algorithm.joho.info/programming/python/pydub-split/)

┗音声トリミングのコード

＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝＝

・参考⑰：[いまさら聞けない？scikit-learnのキホン | DevelopersIO (classmethod.jp)](https://dev.classmethod.jp/articles/introduction-scikit-learn/)

┗重回帰分析

・参考⑱：[Scikit-learnでよく使う分類モデルまとめ | βshort Lab (betashort-lab.com)](https://betashort-lab.com/%E3%83%87%E3%83%BC%E3%82%BF%E3%82%B5%E3%82%A4%E3%82%A8%E3%83%B3%E3%82%B9/%E6%A9%9F%E6%A2%B0%E5%AD%A6%E7%BF%92/scikit-learn%E3%81%A7%E3%82%88%E3%81%8F%E4%BD%BF%E3%81%86%E5%88%86%E9%A1%9E%E3%83%A2%E3%83%87%E3%83%AB%E3%81%BE%E3%81%A8%E3%82%81/)

┗sklearnまとめ

・参考⑲：[誰でも出来る！！scikit-learn(sklearn)で重回帰分析しちゃう - Qiita](https://qiita.com/yossyyossy/items/ac6961ca5a0e765f56be)

┗重回帰分析