Relatório do Projeto de IA



Feito por: Miguel Salgado nº22002195 e Tiago Catano

Curso: Engenharia Informática

Ano: 2022/2023

Índice

[Introdução 3](#_Toc137555419)

[Bibliotecas 4](#_Toc137555420)

[DataSet 5](#_Toc137555421)

[Código 6](#_Toc137555422)

# Introdução

O objetivo deste projeto é o desenvolvimento e implementação de um sistema de classificação automatizado (sistema é completamente autónomo) de géneros musicais, fazendo uso de CNN (Convolution Neural Network). É proposto a utilização de técnicas de Deep Learning para a analise e extração de dados importantes de músicas tendo por fim categorizá-las por genro musical.

# Bibliotecas

* Pandas
  + Biblioteca, que serve para manipulação e análise de dados, este aplica uma estrutura de dados chamada DataFrame (estrutura de dados tabular composta por colunas e linhas).
* Librosa
  + Biblioteca utilizada para análise de áudio e processamento de sinais de áudio. Esta poder ler vários tipos de ficheiro áudio, WAV, MP3 e FLAC. Após uma leitura destes, Librosa fornece vários métodos de extração de dados, tais como espectrogramas, cronogramas e MFCCS (Mel frequency cepstral coefficients). O Librosa é frequentemente utilizado em conjunto com outras bibliotecas processamento de dados em Python, tais como NumPy, SciPy e Matplotlib.
* Keras
  + Utilizada para criação de redes neurais de maneira percetível. Este permite classificar imagens, o processar de linguagem, áudio e também permite fazer a previsão de tempo.
* Sklearn
  + Biblioteca usada para a mineração de dados e analise destes. Esta possui vários algoritmos, como por exemplo regressão logística, árvore de decisão, random forest e redes neurais.
* Seaborn
  + Biblioteca usada para visualização de dados em python, esta cria uma interface onde é mostrador vários tipos de gráficos estatísticos. Tipos de Gráficos:
    - Gráficos de distribuição;
    - Gráficos de dispersão;
    - Gráficos de barras.
* Matplotlib
  + Biblioteca que serve para visualização de dados em python, podendo também criar gráficos, mapas de calor, histogramas.

# DataSet

GTZAN Dataset - Music Genre Classification, apresenta informação sobre uma variedade de gêneros musicais, oferecendo uma visão geral de suas principais características e elementos musicais característicos. No entanto os dados que é considerado mais importante é as próprias músicas.

Este apresenta um total de 1000 músicas, sendo estas dividas em categorias, tais como Blues, Classical, country, disco, Rock e ETC… . O Dataset foi escolhido devido a múltipla utilização deste em outros projetos similares ao nosso, permitindo ter um dataset fidedigno, para adquisição de dados.

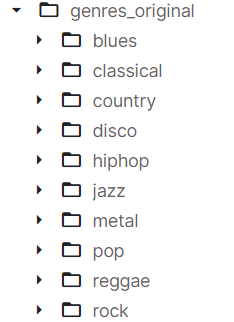
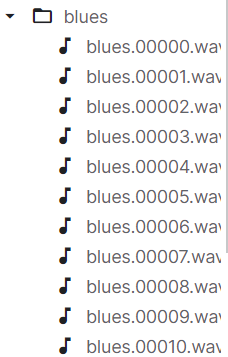


Figura 3 - Músicas

Figura 1 - Géneros De Música

# Código

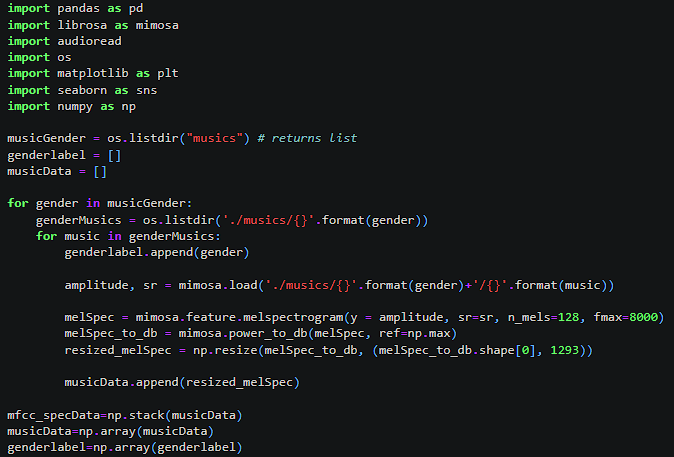


Figura 4 - Música e Pré Processar do Melspectogram

Nesta parte do código é feito o processamento das músicas adquiridas no dataset e a extração de recursos na espectrograma para cada música. Esta usa várias bibliotecas, tais como OS, Librosa e Numpy. No código é inicializado duas listas vazias que depois vão guardar os dados pretendidos. O código inicializa duas listas vazias, **genderlabel** para armazenar os gêneros das músicas e a **musicData** para armazenar os dados da espectrograma mel de cada música.

Após a inicialização das listas é executado a função **os.listdir()** para obter as musicas por cada género. Dentro do loop, o código repete sobre os arquivos de música em cada género, para cada música esta é dada a sua categoria/género.

Em seguida, o código executa a função **mimosa.load()** para obter a amplitude e a taxa de amostragem, para todas as musicas, é depois calculado o espetrograma para cada música usando o **mimosa.feature.melspectrogram()**. A **espectrograma** **mel** é uma representação visual do espectro de frequência da música ao longo do tempo.

O código após a adquisição do espectrograma mel é convertido para decibéis usando **mimosa.power\_to\_db()** , normalizando os valores do espectrograma. É usado o **np.resize()**, para o redimensionamento do espetrograma dando a ele uma forma fixa.

Por fim, a espectrograma mel redimensionado é adicionado à lista **musicData**, criando uma lista de espectrogramas mel correspondentes às músicas processadas.

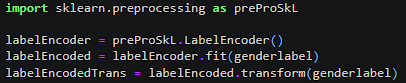


Figura5- Enconding das Categorias

No código a cima é feito a codificação dos géneros, usando como fonte de tal codificação o **sklearn.preprocessing.** Na primeira linha é demonstrado o responsável pelo o transformar dos textos em valores numéricos. Na segunda linha é apresentado **labelEncoder.fit(),** que vai ter como argumento a nossa lista de géneros, **genderlabel,** este método ira preparar os géneros para a codificação, na terceira linha onde ira acontecer a transformação dos géneros em valores numéricos que irão corresponder ao diferentes tipos de género utilizado.

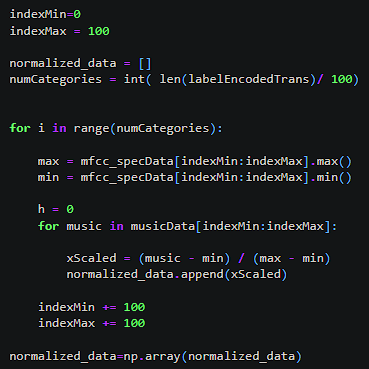


Figura 6 - Normalização de Valores

Na imagem de código demonstrada em cima é criado duas variáveis que são **indexMin** e **indexMax** que vão servir para gerir o número de faixas de músicas a serem normalizadas, juntamente com esta é criado uma lista chamada **normalized\_data,** que como o nome indica vai servir para guardar os dados normalizados. O **numCategories** irá calcular o tamanho dos rótulos codificados dividindo eles por 100, depois ira ser realizado um loop para ser feito a normalização por categoria da música. Esta normalização ira ser aplicada em cada faixa música e o seu respetivo espectrograma mel. Por fim este riam ser guardados na lista mencionada anteriormente, **normalized\_data,** que por ira se transformar numa matriz utilizando a biblioteca numpy. Anteriormente para normalização dos valores tinha sido usado outro método que implicava o uso da média para a normalização dos valores pretendidos, este normalizava todos os dados independentemente do seu género, já a normalização min max, verifica para cada genro o seu ponto máximo e mínimo e normaliza somente os dados desse género. Por fim provou-se que a normalização min max é mais adequado aos dados que procuramos.