

Deep Learning for Music Generation Music Classification

Programa de Pós-Graduação em Informática (PPGI)
UNIRIO

Marcos de Pinho Cotrim
Jean-Pierre Briot
02/07/2019



Agenda

- Objetivo
- Metodologia
- Preparação dos dados (representação)
- Arquitetura
- Conclusões
- Trabalhos futuros

Objetivo

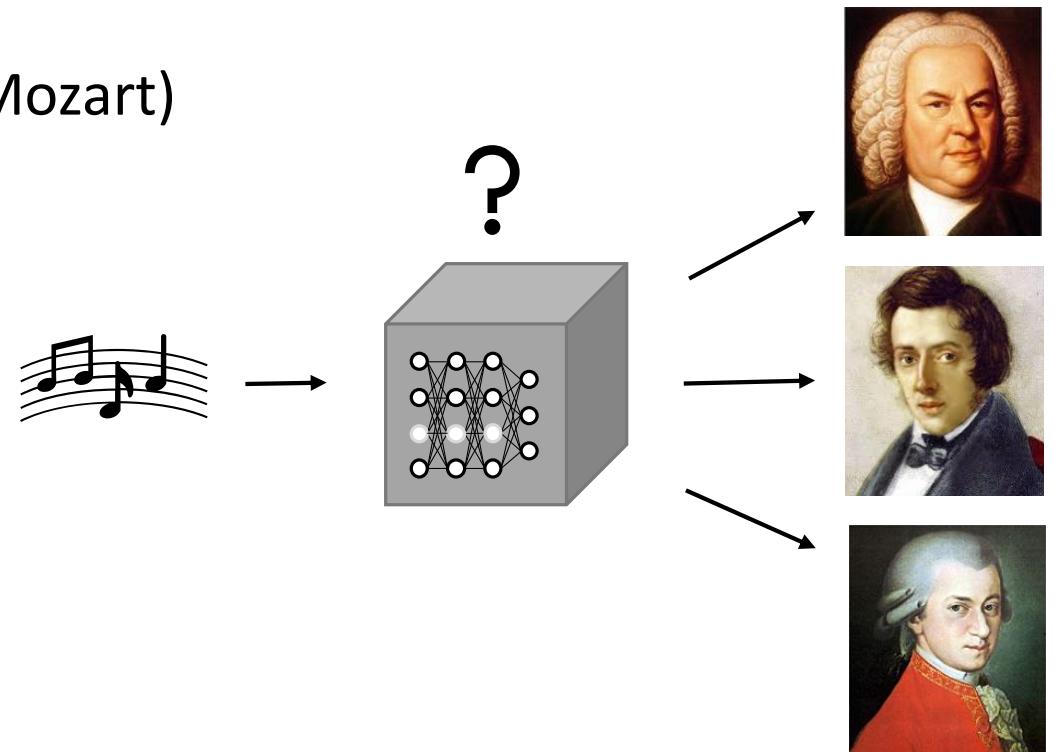
Classificação de melodias

Três compositores possíveis (Bach, Chopin e Mozart)

Utilizando redes neurais artificiais (ANN)

Input: melodias (Music21, XML, MIDI)

Output: compositor mais provável



Aprendizado supervisionado

Classificação é uma tarefa de aprendizado supervisionado

Modelo aprende a partir de exemplos de input/output

Exemplos de algoritmos de classificação

Logistic Regression

Naive Bayes

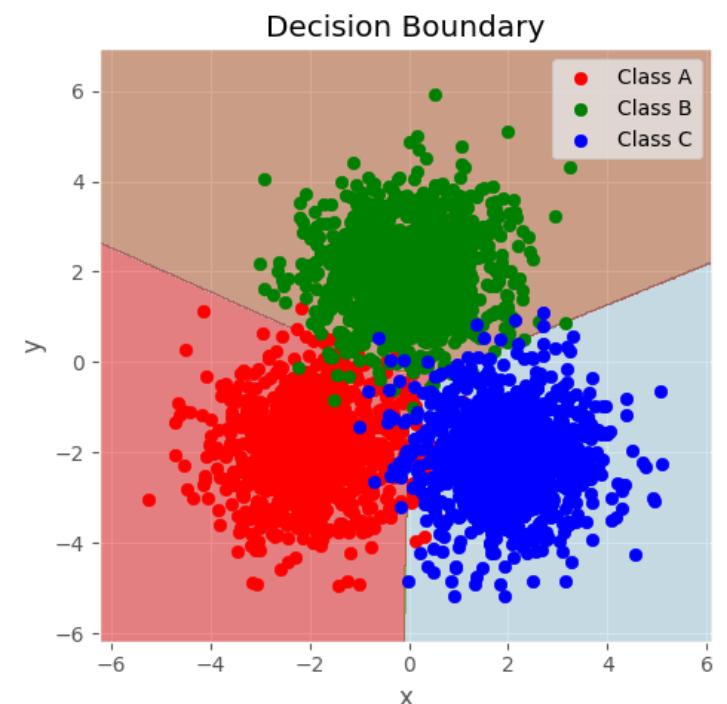
Support Vector Machines (SVM)

Decision Trees

Random Forest

Nearest Neighbors (KNN)

Artificial Neural Networks (ANN)



Metodologia

Aquisição dos dados (music + labels)

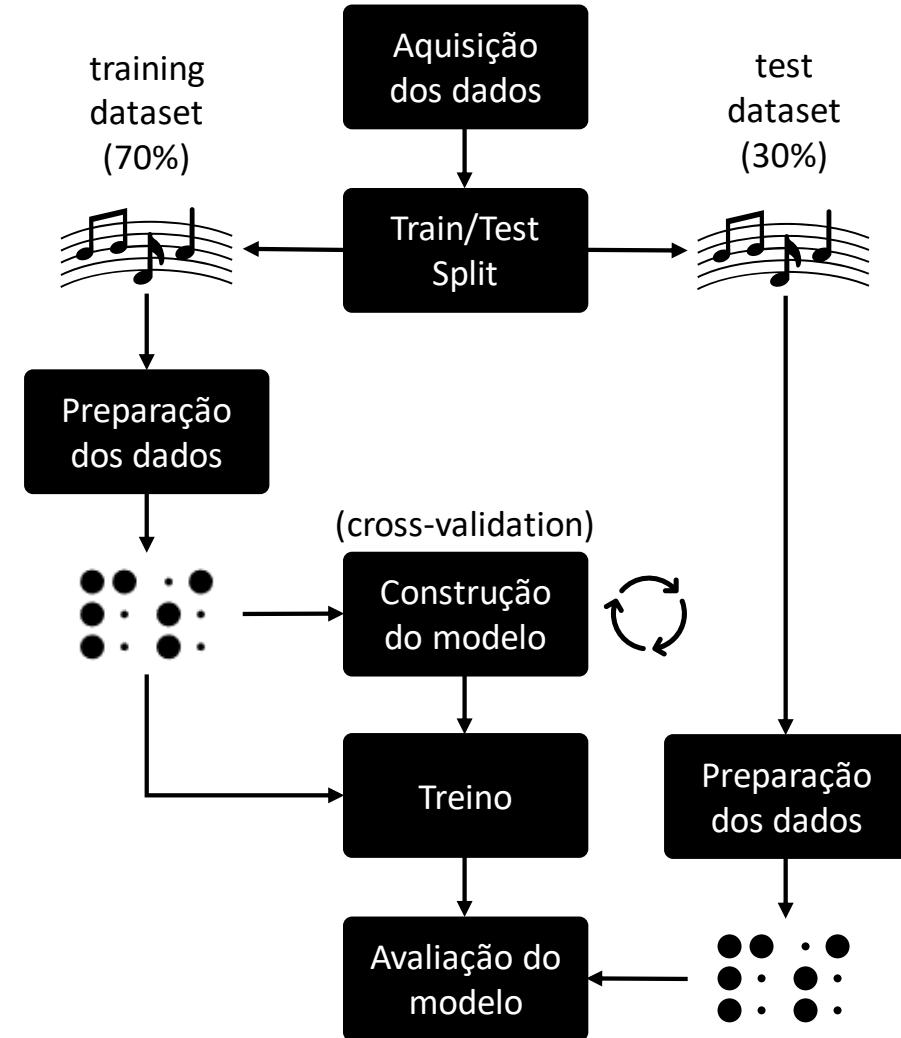
Train/Test split (training set + test set)

Preparação dos dados (representation)

Construção do modelo

Treino do modelo (training set)

Avaliação final do modelo (test set)

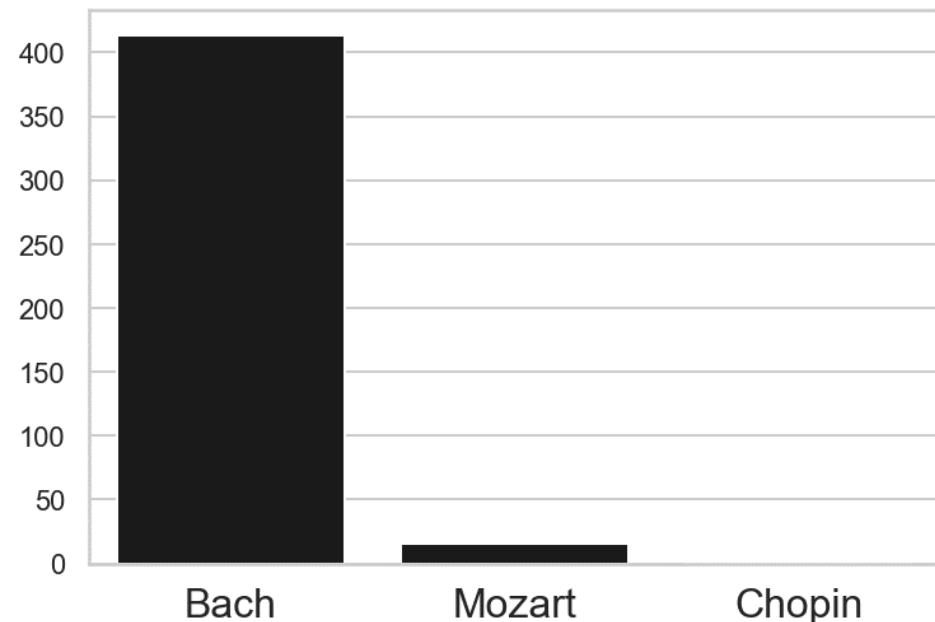




Aquisição dos dados

Corpus do Music21 contém poucas músicas de Mozart e Chopin

Necessidade de aquisição de músicas de outras fontes



Compositor	Music21
Bach	413
Mozart	16
Chopin	1

Preparação dos dados (representação)

Problema

Como transformar uma música em um vetor de [0..1]?

Como transformar diferentes músicas em vetores de [0..1] de mesma dimensão?

Representação escolhida

Many-hot encoding (acordes)

Piano roll (simplicidade)

Fatias de tempo

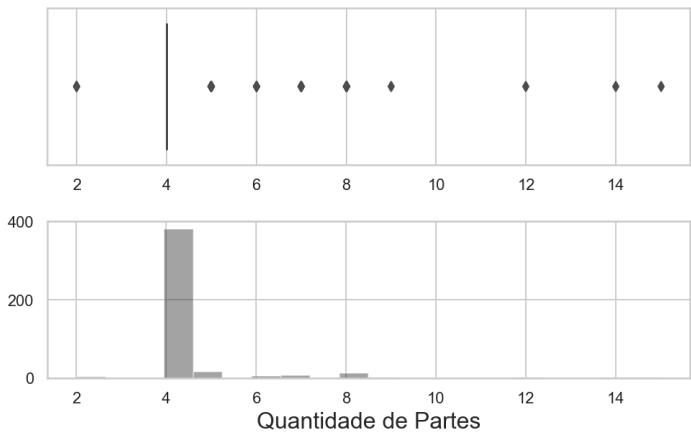
Duração fixa (nota de menor duração)

Tamanho fixo (intervalo entre maior e menor nota)

Quantidade fixa de fatias de tempo (duração total fixa)

Quantidade fixa de partes por música (4 partes)

Quantidade de partes em cada música do corpus do Music21



Preparação dos dados (representação)



1. A partir de todas as músicas do training set, obter:
 1. Quantidade de partes mais frequente (p)
 2. Média de duração das músicas (t_T)
 3. Nota mais baixa (n_B)
 4. Nota mais alta (n_A)
 5. Nota de menor duração (d_M)
2. Definir quantidade de partes por música (fixa) $\rightarrow p$
3. Definir quantidade de fatias de tempo por parte (fixa) $\rightarrow f = t_T / d_M$
4. Definir tamanho do vetor de representação de cada fatia de tempo $\rightarrow m = n_A - n_B$
5. Definir tamanho do vetor de representação de cada música $= V_{(m.f.p)}$
6. Codificar (encode) cada fatia de tempo de cada parte de cada música

Quartet No.2 in D Major, K.155

W.A. Mozart

Violin I
Violin II
Viola
Cello

Vln. I
Vln. II
Vla.
Vc.

Vln. I
Vln. II
Vla.
Vc.

4 partes

W.A. Mozart

4r

Violin I

Violin II

Viola

Cello

Vln. I

Vln. II

Vla.

Vc.

Vln. I

Vln. II

Vla.

Vc.

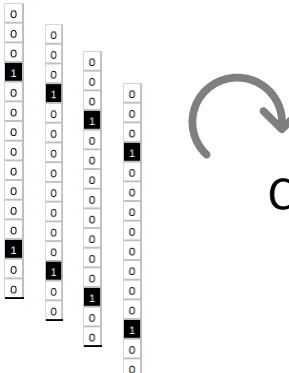
Trecho de 1 parte



many-hot encoding

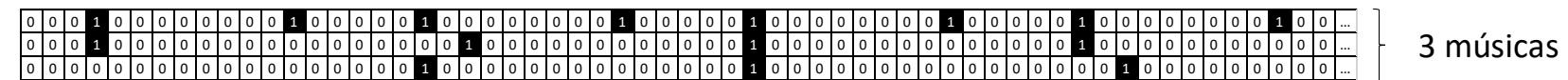
1/32

tempo



Cada música representada como um vetor de tamanho n (fixo), em que:

n = nº de partes x nº fatias de tempo x tamanho da fatia



Dataset representado como uma matriz ($m \times n$) em que:

m = número de músicas

n = tamanho do vetor de representação de cada música (fixo)

Feedforward Architecture

Input layer

Nº de neurons = tamanho do vetor de representação de cada música

Hidden layers

Nº de neurons

Activation function (sigmoid, tanh, relu)

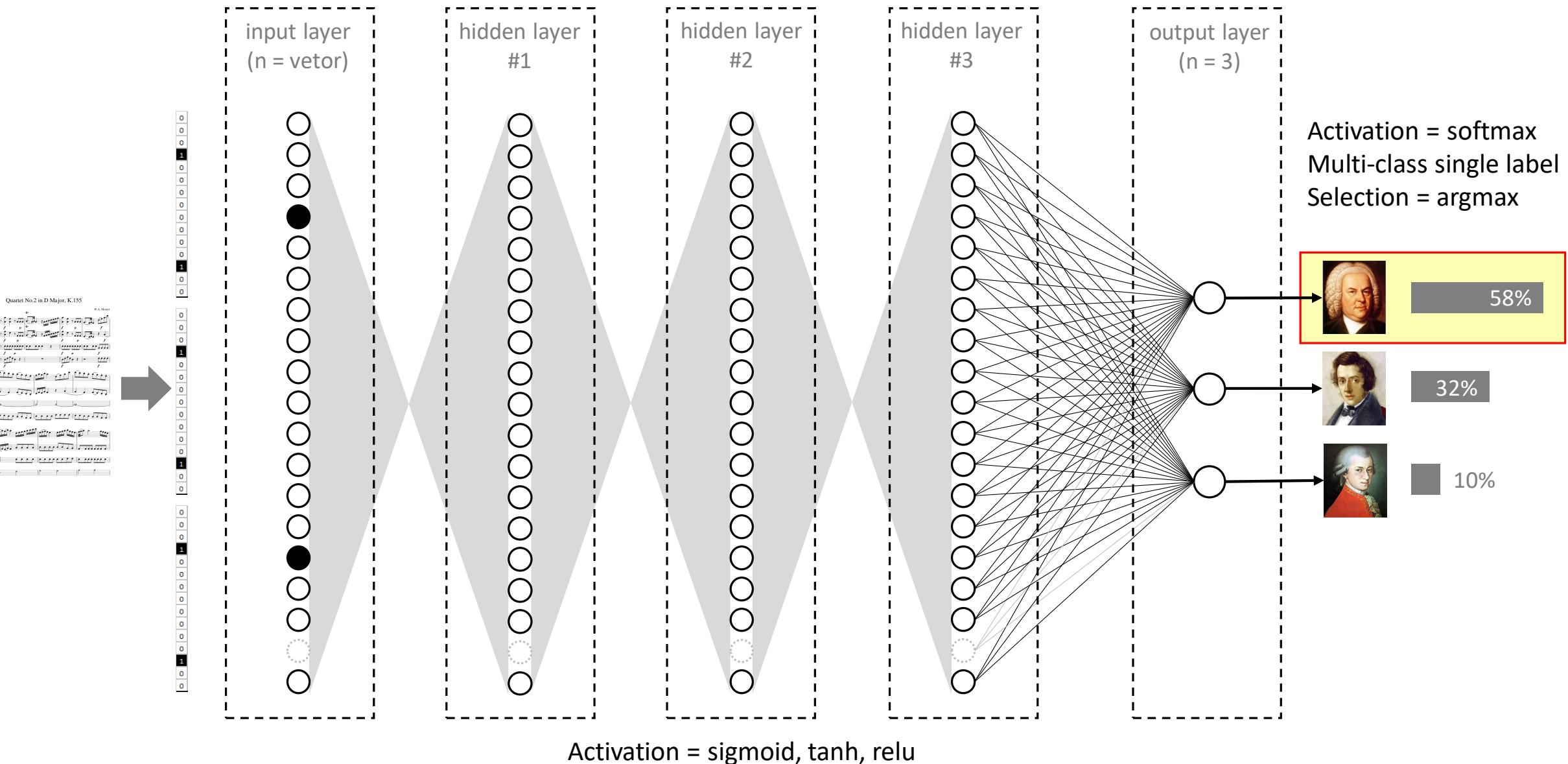
Output layer

3 neurons = Bach, Chopin e Mozart (multi-class single label)

Activation function = softmax

Interpretation = argmax

Loss function = categorical cross-entropy



Construção do modelo

Cross-validation para *tuning* do modelo em diferentes configurações de:

- Número de neurônios em cada camada oculta

- Quantidade de camadas ocultas

- Função de ativação nas camadas ocultas (sigmoid, tanh, relu)

Análise das curvas de aprendizado

- Quantidade de épocas

Selecionar modelo com boa acurácia mas, principalmente, estável

- Capacidade de generalização (*bias-variance tradeoff*)

Avaliação do modelo final

Treino do modelo final

Training dataset completo

Predição (avaliação)

Test dataset (unseen data)

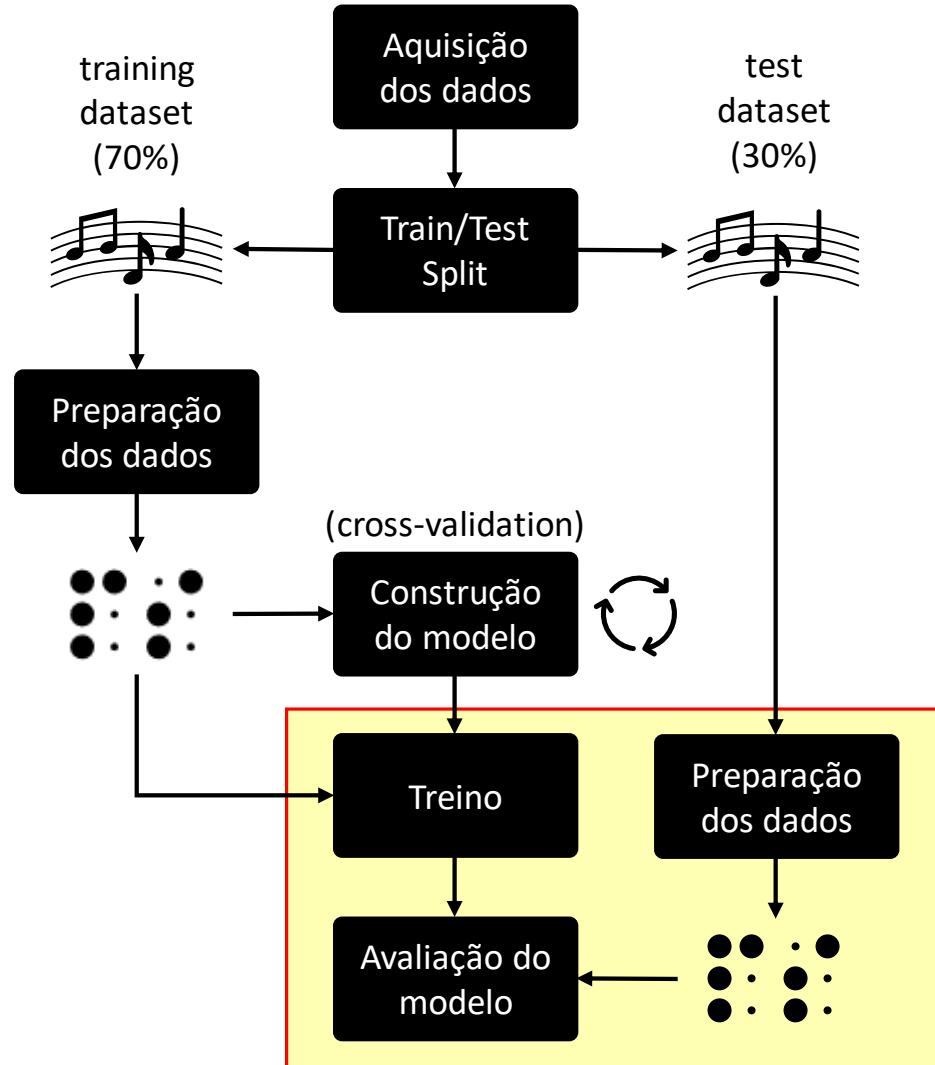
Métricas

Matriz de confusão

Acurácia

F1-Score

Log-Loss



Conclusões

Poucos exemplos de música de Chopin e Mozart no corpus do Music21 e a necessidade de alterações extensas nos códigos de exemplo disponibilizados dificultaram a implementação do modelo e a análise dos resultados.

Músicas possuem particularidades e são diferentes em estruturas e características. A representação de um conjunto variado de músicas em uma estrutura de codificação padronizada e de tamanho fixo, como a utilizada no projeto, limita a quantidade de informação utilizada para o aprendizado.

Algoritmos mais simples podem ser mais indicados para problemas de classificação no lugar de ANN, mas a representação de cada música precisa ser realizada com número menor de variáveis.

Propostas de trabalhos futuros

Incorporação de mais exemplos de músicas no dataset (Chopin e Mozart)

Implementação do modelo proposto (Python, Music21, Keras, TensorFlow)

Avaliação de variações na forma de representação das músicas

Hold/rest, quantidades de partes, quantidades de fatias de tempo

Avaliação de variáveis para redução da dimensionalidade do *dataset*

Frequência de notas por música

Atributos das músicas (tempo, duração, partes/vozes, etc.)

Avaliação de outros algoritmos de classificação (mais simples)

Referências

- Material de aula - <http://www-desir.lip6.fr/~briot/cours/unirio2/>
- Music21 User's Guide -
<http://web.mit.edu/music21/doc/usersGuide/index.html>
- <https://towardsdatascience.com/exploring-activation-functions-for-neural-networks-73498da59b02>
- <https://scikit-learn.org/stable/>
- <https://keras.io/>
- <https://medium.com/usf-msds/choosing-the-right-metric-for-evaluating-machine-learning-models-part-2-86d5649a5428>