



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA (ISEL)

DEPARTAMENTO DE ENGENHARIA ELETRÓNICA E DE  
TELECOMUNICAÇÕES E COMPUTADORES (DEETC)

---

LEIM

LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA  
UNIDADE CURRICULAR DE PROJETO

---

## Anotação de Eventos Sonoros em Vídeo

Carina Fernandes (45118)

---

*Orientadores*

*Professor Doutor*      Joel Paulo

*Professor Doutor*      Paulo Trigo

*Professor*                  Paulo Vieira

---

*Setembro, 2022*



# Resumo

O desempenho de um atleta, numa determinada modalidade desportiva, melhora quando é acompanhado de uma perspetiva externa no decurso da sua atividade desportiva. O atleta pode ser monitorizado por um treinador para atingir melhores resultados. Como complemento, é possível registar em vídeo e analisar posteriormente o seu desempenho. Neste sentido, é vantajoso o desenvolvimento de uma ferramenta capaz de realizar essa análise externa.

A ferramenta desenvolvida neste projeto permite extrair e reconhecer batidas de bola em raquete com base no vídeo da atividade desportiva do atleta (em treinos ou competições de padel /ténis).

Trata-se de um problema de classificação binário em que o sistema deve fazer a distinção entre batidas de bola e ruído. Estes eventos têm uma duração típica pré-determinada e a distinção entre eles foi feita com base nas características (*Onset Detect*, *Root Mean Square* e *Spectral Flux*) identificadas no áudio através de técnicas de aprendizagem automática.

O *dataset* utilizado no processo de aprendizagem foi construído de raiz e o desequilíbrio verificado neste conjunto de dados foi abordado através de um processo de sub-amostragem.

Entre os vários métodos de aprendizagem automática analisados, escolheu-se a rede neuronal do tipo perceptrão multi-camada (*Multi Layer Perceptron* - MLP) para realizar a discriminação entre os tipos de eventos mencionados.

No final, através de uma aplicação *web*, são visualizados os resultados repetitantes aos instantes em que ocorrem batidas de bola, dando uma perspetiva mais abrangente e objetiva do desempenho do atleta. A ferramenta identifica corretamente cerca de 85% dos eventos em análise.

Os ensaios são realizados no Laboratório de Áudio e Acústica do ISEL, LAA.

**Palavras-chave:** algoritmos de software de deteção do som, algoritmos de software de inteligência artificial.



# Abstract

In a given sport, the athlete's performance improves when he is supervised from an external perspective. In this scenario, the athlete can be supervised by a coach in order to achieve better results, or as a complement, it is possible to video record and analyze his performance afterwards. Thus, it is advantageous to develop a tool capable of performing this external analysis.

This is a binary classification problem in which the system must distinguish between ball hits and noise. These events have a predetermined typical duration and this distinction was made based on the features (*Onset Detect*, *Root Mean Square* and *Spectral Flux*) identified in the audio through machine learning techniques.

The *dataset* used in the learning process was built from scratch and the imbalance found in this dataset was addressed through an oversampling process.

Among the various machine learning methods analyzed, the multi-layer perceptron (MLP) was chosen to perform the discrimination between the mentioned event types.

At the end, through a *web* application, the results relative to the instants in which ball hits occur are visualized, giving a more comprehensive and objective perspective of the athlete's performance. The tool correctly identifies about 85% of the events under analysis, with some adjustments needed to improve the recognition process.

The tests are carried out at ISEL's Audio and Acoustics Laboratory, LAA.

**Keywords:** sound detection software algorithms, artificial intelligence software algorithms.



# Agradecimentos

Escrever aqui eventuais agradecimentos ...





*Eventual texto de dedicatória . . .*

*. . . mais texto,*

*. . . e o fim do texto.*



# Índice

Resumo	i
Abstract	iii
Agradecimentos	v
Índice	ix
Lista de Tabelas	xi
Lista de Figuras	xiii
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação . . . . .	1
1.2 Principais Contributos . . . . .	2
1.3 Validação e Testes . . . . .	2
1.4 Estrutura do Relatório . . . . .	2
<b>2 Fundamentos</b>	<b>5</b>
2.1 Conceitos de Processamento de Sinal . . . . .	5
2.1.1 Amostragem . . . . .	5
2.1.2 Transformada de Fourier . . . . .	6
2.1.3 Características extraídas do áudio . . . . .	6
2.2 Conceitos de Aprendizagem Automática . . . . .	7
2.2.1 <i>Dataset</i> . . . . .	8
2.2.2 Desiquilíbrio no <i>Dataset</i> . . . . .	8
2.2.3 Hiper-parâmetros . . . . .	8
2.2.4 Redes Neurais . . . . .	9

2.2.5	Validação Cruzada . . . . .	10
<b>3</b>	<b>Trabalho Relacionado</b>	<b>13</b>
<b>4</b>	<b>Modelo Proposto</b>	<b>15</b>
4.1	Descrição Geral do Sistema . . . . .	15
4.2	Requisitos . . . . .	15
4.3	Abordagem . . . . .	17
4.3.1	Construção do <i>Dataset</i> . . . . .	17
4.3.2	Construção do Classificador . . . . .	20
4.3.3	Desenvolvimento da Aplicação <i>web</i> . . . . .	20
<b>5</b>	<b>Implementação do Modelo</b>	<b>23</b>
5.1	Construção do <i>Dataset</i> . . . . .	23
5.2	Construção do Classificador . . . . .	24
5.3	Desenvolvimento da Aplicação <i>web</i> . . . . .	26
<b>6</b>	<b>Validação e Testes</b>	<b>27</b>
<b>7</b>	<b>Conclusões e Trabalho Futuro</b>	<b>29</b>
<b>A</b>	<b>Um Detalhe Adicional</b>	<b>31</b>
<b>B</b>	<b>Outro Detalhe Adicional</b>	<b>33</b>
	<b>Bibliografia</b>	<b>35</b>

# Lista de Tabelas

4.1	Requisitos funcionais do sistema em desenvolvimento para a construção do <i>dataset</i> . . . . .	16
4.2	Requisitos funcionais respeitantes ao processo de classificação.	16
4.3	Requisitos não funcionais do sistema. . . . .	16
4.4	Valores de duração (em segundos), amostras delizadas e N a considerar, para um valor de <i>samplingRate</i> igual a 44100Hz. .	18



# Lista de Figuras

2.1	Características (cima e meio) e áudio (baixo) correspondente. .	7
2.2	Representação típica de uma rede neuronal. . . . .	9
4.1	Sistema de anotação de eventos desejado. . . . .	15
4.2	Matriz de características. . . . .	18
4.3	Vetor de classes. . . . .	19
4.4	Matriz de características. . . . .	20
4.5	<i>Design</i> desejado da página <i>web</i> 2. . . . .	21
5.1	Processo detalhado de construção do <i>dataset</i> . . . . .	24
5.2	Resultados (independentes das classes) da aplicação de vários algoritmos aos dados. . . . .	25
5.3	Processo de construção do classificador. . . . .	26
B.1	Descrição pormenorizada do campo de padel. [Tennisnerd, 2015]	33
B.2	Resultados da aplicação de vários algoritmos aos dados para a classe dos positivos. . . . .	34
B.3	Resultados da aplicação de vários algoritmos aos dados para a classe dos negativos. . . . .	34





# Capítulo 1

## Introdução

A atividade desportiva de um atleta, numa determinada modalidade, deve ser monitorizada de uma perspetiva externa, no sentido de produzir melhores resultados. Para esse efeito, é necessário que o atleta seja supervisionado durante a sua atividade física, ou que o seu treino seja registado (em papel ou vídeo) e posteriormente analisado.

Neste trabalho, a anotação de eventos sonoros em vídeo refere-se à análise do áudio extraído de um vídeo previamente gravado, com o objetivo de identificar cada instante em que ocorre uma batida de bola (tipo ténis) em raquete. O processamento sobre o áudio será realizado em modo *offline*.

Entre os vários desportos praticados, este trabalho foca-se na análise da componente desportiva de padel ([Courel-Ibáñez et al., 2019]), no sentido de fornecer a atletas e/ou treinadores um complemento no decurso do treino. Uma figura referente ao campo onde este desporto é praticado e à raquete e bola utilizadas encontra-se disponível no Capítulo A, na qual se observam as dimensões do campo onde esta modalidade desportiva é realizada. O desporto de padel pode ser praticado em ambiente interior (*indoor*) ou exterior (*outdoor*). Este trabalho analisa apenas vídeos em ambiente *indoor*.

### 1.1 Motivação

A aplicação em desenvolvimento implica que se implemente um sistema onde a máquina é capaz de reconhecer batidas de bola. No entanto, a forma como uma máquina reconhece um som difere de como esse mesmo som é percebido pelo ouvido humano. O ouvido humano consegue fazer a distinção

entre sons, considerando as características dos sons emitidos: duração, intensidade, tom, entre outras [Council et al., 2004]. É necessário proporcionar à máquina a capacidade de reconhecer batidas de bola de forma semelhante à humana. Para esse efeito, será utilizada a aprendizagem automática.

A aprendizagem automática permite identificar e reconhecer padrões identificativos de batidas de bola no *dataset* fornecido.

A aplicação em desenvolvimento disponibiliza ainda uma interface, que permite que o utilizador participe no processo de anotação dos eventos (validando os resultados devolvidos pelo classificador), o que poderá contribuir numa perspetiva futura, para aumentar o *dataset* e melhorar deteção de batidas de bola.

## 1.2 Principais Contributos

A realização deste trabalho terá os seguintes contributos:

- Não existindo dados relativos a batidas de bola ou que estejam etiquetados nas condições desejadas, será construído um *dataset* de raiz. Os dados constituintes do *dataset* serão obtidos de vídeos amadores;
- Providenciar de um modelo que reconhece batidas de bola;
- Proporcionar de uma aplicação *web* que permite aumentar o *dataset*.

## 1.3 Validação e Testes

(A alterar depois de fazer o capítulo de validação e testes)

## 1.4 Estrutura do Relatório

O relatório está organizado da seguinte forma:

- O Capítulo 1 introduz a necessidade do uso da aprendizagem automática para reconhecer batidas de bola em vídeos de treinos de padel, enuncia os principais contributos da aplicação em desenvolvimento, e realiza uma apreciação global dos resultados obtidos.

- 
- O Capítulo 2 faz referência a trabalhos relacionados com o trabalho corrente. Estes trabalhos foram sendo analisados durante o desenvolvimento do trabalho e baseiam-se também no uso de técnicas de aprendizagem automática para reconhecer eventos em modalidades desportivas.
  - O Capítulo 3 introduz alguns conceitos nos quais o trabalho em desenvolvimento se baseia: conceitos sobre processamento de sinal e de aprendizagem automática.
  - O Capítulo 4 visa apresentar uma análise geral dos requisitos do sistema e a abordagem utilizada para satisfazer esses requisitos.
  - O Capítulo 5 descreve todo o processo de concretização da solução abordada no capítulo anterior.
  - (Falar do capítulo 6)
  - (Falar do capítulo 7)



# Capítulo 2

## Fundamentos

Neste capítulo serão abordados os conceitos de carácter teórico, que sustentam o trabalho realizado.

### 2.1 Conceitos de Processamento de Sinal

Um sinal corresponde a uma grandeza física que varia ao longo do tempo. Em termos matemáticos, um sinal varia em função de uma ou mais variáveis, que podem ser contínuas (sinais contínuos) ou discretas (sinais discretos) [Sampaio et al., 2006].

Neste trabalho, o áudio será extraído do vídeo através de um processo de amostragem e por isso corresponde a um sinal discreto no tempo.

#### 2.1.1 Amostragem

O processo de amostragem converte um sinal contínuo num sinal discreto. Para isso, escolhe pontos equi-espaçados do sinal contínuo para gerar os pontos do sinal discreto. A amostragem de um sinal contínuo com um período específico permite obter um sinal discretizado no tempo [LibreTexts, 2022a]. Cada um dos pontos resultantes é designado por amostra.

O intervalo entre as amostras consecutivas denomina-se período de amostragem e é dado em segundos(s). O inverso deste valor corresponde à frequência de amostragem, que é dada em *Hertz*(Hz), e refere-se ao número de amostras que ocorre no sinal a cada segundo. A informação removida pelo processo de amostragem pode ser parcialmente recuperada através de interpolação ou reconstrução do sinal [LibreTexts, 2022b].

O ouvido humano deteta sons na gama de frequências compreendida entre 20Hz e 20kHz, pelo que, de acordo com o teorema de *Nyquist* [Evia, 2022], é necessário que a frequência de amostragem dos sons que chegam ao aparelho auditivo humano sejam amostrados a uma frequência duas vezes igual ou superior a 20kHz.

A frequência de amostragem a ser considerada neste projeto é de 44100Hz (cerca de 43kHz), que é um dos valores padrão utilizados [Brown, 2019].

### 2.1.2 Transformada de Fourier

A transformada de Fourier (*Fourier Transform* – FT) é uma função matemática que decompõe um sinal nas suas várias frequências (sinusóides) constituintes [Semmlow, 2012]. O sinal de entrada pertence ao domínio do tempo, e o sinal de saída pertence ao domínio da frequência. Esta função (nos domínios do tempo e frequência) é designada por espectro.

A transformada de *Fourier* é útil para verificar, por exemplo, em que zonas do espectro existe mais ou menos energia concentrada e será utilizada no cálculo das características a extrair do áudio.

### 2.1.3 Características extraídas do áudio

Neste trabalho, a partir do áudio são extraídas as seguintes características: deteção do início de um som (*Onset*) e o valor eficaz (RMS).

O *Onset* divide-se em duas componentes: a deteção do início (*Onset Detect*) e o fluxo espectral (*Spectral Flux*). A primeira componente, corresponde a detetar o instante em que se inicia um determinado som [Rosão, 2012]. A segunda corresponde a verificar variações no espectro do sinal, no sentido de encontrar diferenças entre *frames* consecutivas, o que permite detetar também o início de um som. O fluxo espectral pode também ser definido como uma medida do quão depressa o espectro de um sinal varia [Meyda, 2022].

O valor eficaz (*Root Mean Square* – RMS) refere-se à energia (intensidade) média concentrada numa *frame* [Room, 2021].

A figura 2.1 constitui uma representação das três características enunciadas.

Observando a figura, verifica-se que os picos das duas componentes do *Onset*, bem como o RMS estão correspondem aos instantes no áudio original

onde a amplitude é maior.

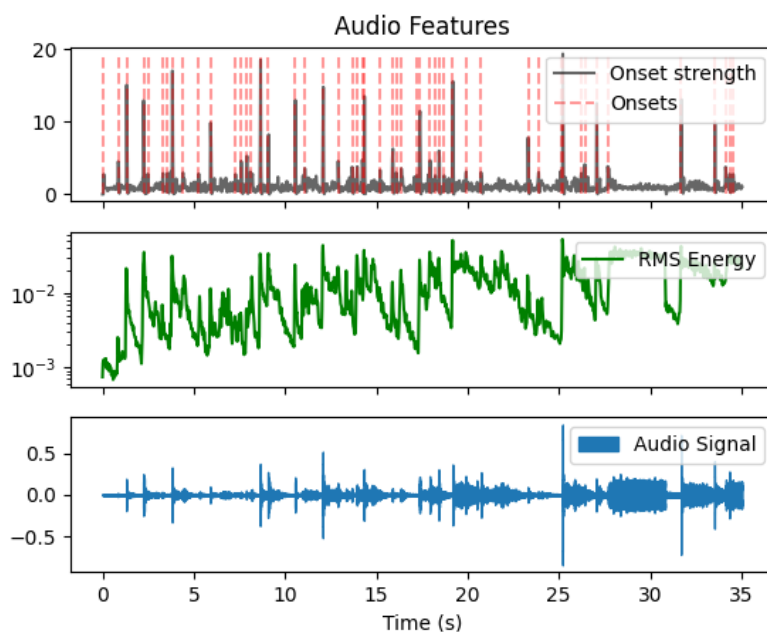


Figura 2.1: Características (cima e meio) e áudio (baixo) correspondente.

As características enunciadas acima podem ser calculadas com o auxílio da biblioteca *librosa* [McFee et al., 2015]. Este módulo realiza processamento sobre sinais de áudio, no sentido de recolher informações sobre o mesmo.

## 2.2 Conceitos de Aprendizagem Automática

A aprendizagem automática (*Machine Learning* - ML) é ramo da inteligência artificial que recorre à análise de (grandes volumes de) dados visando extrair padrões e representá-los num modelo [Janiesch et al., 2021].

Existem três tipos de aprendizagem automática [Sah, 2020]:

- Supervisionada – O modelo é construído com base num conjunto de dados (*dataset*) que associa uma lista de características (*features*) à classe correspondente;
- Não Supervisionada – O modelo é construído com base num conjunto de dados (*dataset*) que apenas contém as características;

- Por Reforço – O modelo é contruído com base na interação com um sistema que gera os dados necessários à extração de padrões.

Este trabalho utiliza aprendizagem supervisionada para extrair os padrões de áudio (e.g., batida de bola em raquete).

### 2.2.1 *Dataset*

Um *dataset* é um conjunto de dados usado no processo de construção dos modelos gerados através de métodos de aprendizagem automática. O *dataset* é estruturado em colunas e linhas (formato de tabela). No caso da aprendizagem supervisionada as colunas representam as características (*features*) e a classe. As linhas representam sempre instâncias (exemplos ou observações).

Considerando uma matriz de características,  $X$ , e o vetor de valores de classes correspondente,  $y$ , o problema de classificação em questão tem como objetivo obter a função,  $f$ , tal que:

$$f(X) = y \quad (2.1)$$

A função  $f$  corresponde a um modelo que deve receber a matriz de características e, classificar de maneira correta cada um dos seus exemplos.

### 2.2.2 *Desiquilíbrio no Dataset*

O desequilíbrio no *dataset* ocorre quando a distribuição de exemplos pelas classes é desigual [Badr, 2019]. Esse desequilíbrio pode resultar na construção de um modelo enviesado que reconhece melhor determinadas classes.

Num problema de classificação binária o desequilíbrio pode abordar-se aumentando o número de exemplos da classe em minoria (sobre-amostragem) ou reduzindo o número da classe em maioria (sub-amostragem) [Badr, 2019].

Neste trabalho será utilizada a técnica de sub-amostragem para equilibrar o *dataset*.

### 2.2.3 *Hiper-parâmetros*

Em aprendizagem automática, um hiper-parâmetro é um valor utilizado pelo modelo para controlar o processo de treino. Os hiper-parâmetros de



um modelo não são utilizados diretamente no modelo, mas determinam os parâmetros internos utilizados diretamente. O prefixo “hiper” advém do facto de estes parâmetros serem os escolhidos para definir outros, sugerindo que estão num nível hierárquico superior [Nyuytiymbiy, 2021].

(Falar dos hiper-parâmetros utilizados neste trabalho).

## 2.2.4 Redes Neurais

Uma rede neuronal artificial (*Artificial Neural Network* – ANN) é uma estrutura usada em métodos de aprendizagem automática que tem como inspiração a biologia humana e a forma como os neurónios comunicam entre si para compreender os dados percecionados [Rauber, 2022].

Uma rede neuronal está organizada em camadas, que são constituídas por unidades (nós). A primeira camada (de *input*), tem um número de nós igual ao número de características do *dataset*. A última camada (de *output*), possui os nós que são usados para representar os valores da classe. Entre a camada de *input* e a camada de *output* existem uma ou mais camadas escondidas. A figura 2.2 representa de uma rede neuronal, onde estão delimitadas as ligações entre os vários nós constituintes.

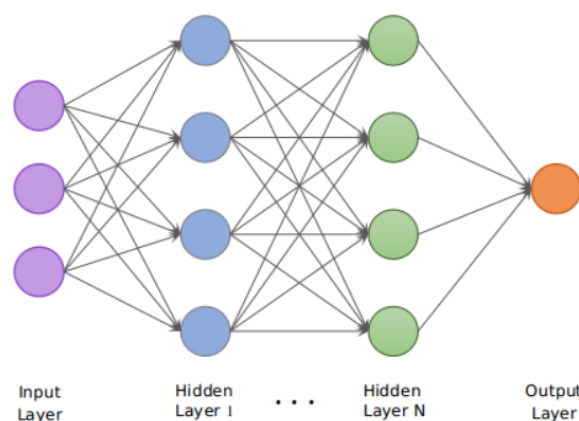


Figura 2.2: Representação típica de uma rede neuronal.

O número de camadas, bem como o número de nós em cada camada são definidos com o objetivo de obter o melhor classificador possível [Krishnan, 2021].

Tal como se verifica na figura 2.2, cada nó recebe os valores de saída dos nós da camada anterior e associa-lhes um peso que é atualizado ao longo do

processo de aprendizagem [Medium, 2019].

Os valores de saída são calculados com base numa função de ativação que define a forma como os pesos a serem atualizados são calculados numa determinada camada da rede. Neste trabalho serão utilizadas as funções de ativação [Brownlee, 2021]:

- ReLU (*Rectified Linear Activation*) – recebe um valor e devolve outro de acordo com a expressão 2.2. Pelo que valores negativos são mapeados no valor 0. Esta função será utilizada nas camadas escondidas.

$$outputValue = \max(0, inputValue) \quad (2.2)$$

- Sigmóide (*Sigmoid*) – recebe um valor e devolve outro no intervalo  $[0, 1]$  de acordo com a expressão 2.3. Esta função será utilizada na camada de *output*.

$$outputValue = \frac{1}{1 + e^{-inputValue}} \quad (2.3)$$

O processo de aprendizagem é realizado ao longo de várias épocas. Uma época (*epoch*) corresponde a um ciclo de passagem, pela rede neuronal, de todos os exemplos do *dataset*. Uma época é constituída por várias iterações. Cada iteração corresponde a uma atualização de pesos após a passagem pela rede de um número pré-definido (*batch*) de exemplos.

O uso de métodos de otimização permite atualizar os pesos e a taxa de aprendizagem da rede neuronal de forma a reduzir o erro associado e aumentar a taxa de sucesso. Neste trabalho, será utilizado o método de otimização *Adam* [Doshi, 2019].

### 2.2.5 Validação Cruzada

No caso da aprendizagem supervisionada, a classificação é utilizada para atribuir uma classe (categoria) a um conjunto de exemplos. Após este processo, é construído um modelo que através dos padrões presentes nos dados de *input*, prevê a classe à qual esses dados pertencem. Existem três tipos de classificação [Brownlee, 2022]:

- Multi-Classe – quando existem mais do que duas classes nos dados. Um exemplo não pode pertencer, simultaneamente, a mais do que uma classe.
- Binária – quando existem apenas duas classes nos dados. Um exemplo não pode pertencer, simultaneamente, a mais do que uma classe.
- Multi-Label – quando existem duas ou mais classes e um exemplo pode pertencer, simultaneamente, a mais do que uma classe.

Neste trabalho, o problema de classificação é binário, uma vez que o modelo deve distinguir exemplos que são batidas de bola (positivos) de exemplos que não são batidas de bola (negativos).

Para avaliar o desempenho de um classificador é necessário verificar quais são as hipóteses de este acertar independentemente da classe, mas também verificar a distribuição de acertos em cada classe. Essas probabilidades podem ser obtidas através da matriz de confusão. Neste trabalho, entre as várias métricas de desempenho que podem ser obtidas da matriz de confusão, serão consideradas as seguintes [Brownlee, 2020]:

- Taxa de sucesso (*Accuracy*) – percentagem de exemplos bem classificados independentemente da classe;
- Precisão (*Precision*) – percentagem de exemplos que o classificador indicou serem de uma determinada classe e que pertencem a essa classe;
- Cobertura (*Recall*) – percentagem de exemplos que pertencem a uma determinada classe e o classificador indicou como sendo dessa classe;
- *F1-score* – corresponde à média harmónica entre a precisão e a cobertura [MachineLearning, 2017].

Em geral, num problema de classificação os dados são divididos em treino e teste. O grupo de exemplos de treino é utilizado no processo de aprendizagem, e o grupo de teste é utilizado no processo de classificação. Embora esta abordagem permita verificar como o classificador se comporta na predição sobre dados novos, tem como desvantagem a hipótese de existirem exemplos mais fáceis de identificar no conjunto de teste. Como solução pode ser utilizada validação cruzada.

A validação cruzada é uma técnica que permite avaliar a capacidade de generalização de um classificador. Os dados são divididos em  $K$  partições (*folds*) e ao longo de  $K$  iterações são treinados modelos diferentes, onde  $K-1$  *folds* são utilizados para treino e um *fold* é utilizado para validação [Alhamid, 2020].

Dividir os dados em várias partições pode não ser suficiente, uma vez que a distribuição dos exemplos de cada classe pelos *folds* pode não ser igual à original.

De forma a manter essa distribuição de exemplos de cada classe nos *folds* será utilizada a técnica de validação, partição estratificada (*Stratified K-Fold*) [Muralidhar, 2021].

# Capítulo 3

## Trabalho Relacionado

(Por fazer)



# Capítulo 4

## Modelo Proposto

Neste capítulo será realizada uma descrição geral do sistema desenvolvido, dos requisitos e da abordagem adotada para o desenvolver.

### 4.1 Descrição Geral do Sistema

O sistema desenvolvido reconhece batidas de bola, permite auxiliar no processo de construção do dataset e, conseqüentemente no processo de melhoria de desempenho do classificador. O funcionamento do sistema pode ser representado como se mostra na figura 4.1.

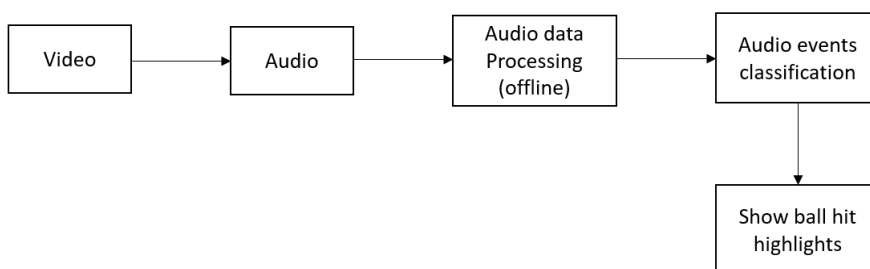


Figura 4.1: Sistema de anotação de eventos desejado.

### 4.2 Requisitos

Os requisitos do sistema podem ser categorizados como funcionais (funções de sistema) e não funcionais (atributos do sistema). As funções de sistema,

referem-se a funcionalidades desempenhadas pelo sistema. Os atributos do sistema descrevem a forma como o sistema deve ser disponibilizado.

Neste projeto, os requisitos funcionais foram organizados em dois grupos:

- Construção do *dataset*;
- Construção do classificador.

A tabela 4.1 contém os requisitos funcionais que dizem respeito à criação do *dataset*:

Requisito	Função
R1.1	Extrair o áudio do vídeo
R1.2	Anotar as batidas de bola presentes no áudio (A)
R1.3	Extrair as características do áudio (B)
R1.4	Combinação dos dois conjuntos A e B

Tabela 4.1: Requisitos funcionais do sistema em desenvolvimento para a construção do *dataset*.

A tabela 4.2 contém os requisitos funcionais que dizem respeito ao processo de classificação:

Requisito	Função
R1.1	Extrair o áudio do vídeo
R1.2	Extrair as características do áudio
R1.3	Aplicar o classificador
R1.4	Apresentar resultados dos eventos etiquetados

Tabela 4.2: Requisitos funcionais respeitantes ao processo de classificação.

A tabela 4.3 representa os atributos do sistema.

Atributo	Detalhe/ Restrição Fronteira
Interação Homem-Máquina	Detalhe Interface fácil de aprender a usar

Tabela 4.3: Requisitos não funcionais do sistema.



## 4.3 Abordagem

Nesta secção será descrita a abordagem utilizada para realizar cada uma das etapas inerentes ao sistema em desenvolvimento. As etapas são as seguintes:

- Construção do *dataset*;
- Construção de um classificador que reconhece batidas de bola;
- Desenvolvimento de uma aplicação *web* que contribui para aumentar o *dataset*.

### 4.3.1 Construção do *Dataset*

O *dataset* construído, corresponde a um conjunto de dados com padrões identificativos de batidas de bola.

#### Obtenção da matriz de características – $X$

No processo de extração das características (cf., 2.1.3) é realizado um varrimento sobre o áudio com uma janela temporal de dimensão fixa. Este processo é realizado ao longo de várias iterações, onde em cada iteração a janela desliza um número específico de amostras, a partir das quais são extraídas as características pretendidas.

No cálculo dessas características, o *package librosa* considera um salto (“*hop*”) de 512 amostras. Pelo que, este módulo divide o sinal em segmentos, com 512 amostras de dimensão, e sobre cada um deles calcula o valor da característica [McFee et al., 2015]. No entanto, podem ser escolhidos múltiplos do valor do salto como outras abordagens, no sentido de verificar que alterações produzem no desempenho do classificador. O mesmo se aplica ao tamanho da janela: uma batida de bola dura em média meio segundo, mas podem ser realizados experimentos com durações maiores (na tabela 4.4). Neste trabalho, como primeira abordagem, será considerada uma janela de 0.5 segundos e um salto de 1024 amostras.

A matriz  $X$  é constituída por  $M$  linhas ou exemplos. Por sua vez, cada exemplo tem  $N$  colunas ou valores de características, pelo que a matriz  $X$  pode ser representada na figura 4.2 da seguinte forma:

$$X = \begin{bmatrix} X_1 & X_2 & X_3 & \cdots & X_N \\ X_2 & X_3 & X_4 & \cdots & X_{N+1} \\ X_3 & X_4 & X_5 & \cdots & X_{N+2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ X_{N-1} & X_N & X_{N+1} & \cdots & X_{2N-2} \\ X_N & X_{N+1} & X_{N+2} & \cdots & X_{2N-1} \\ X_{N+1} & X_{N+2} & X_{N+3} & \cdots & X_{2N} \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix}$$

Figura 4.2: Matriz de características.

O valor de  $N$  na figura 4.2, corresponde ao número de segmentos de amostras abranjidos pela janela fixa para uma característica e pode ser dado pela seguinte expressão:

$$N = \frac{eventLength \times samplingRate}{hopLength} \quad (4.1)$$

onde na equação, *eventLength* é a duração de um evento, *samplingRate* é a frequência de amostragem (a que os áudios são obtidos), e *hopLength* refere-se ao número de amostras deslizadas a cada iteração do varrimento. Na tabela 4.4, estão evidenciados os valores de  $N$  obtidos para as várias combinações de *eventLength* e *hopLength*:

<i>eventLength</i>	<i>hopLength</i>	<i>N</i>
0.5	1024	22
0.5	2048	11
0.5	4096	5
0.7	1024	30
0.7	2048	15
0.7	4096	8
1.0	1024	43
1.0	2048	22
1.0	4096	11

Tabela 4.4: Valores de duração (em segundos), amostras deslizadas e  $N$  a considerar, para um valor de *samplingRate* igual a 44100Hz.

Observando ainda a figura 4.2, a primeira iteração do processo de var-

rimento permite obter a primeira linha da matriz  $X$ . A segunda segunda iteração permite obter a segunda linha da matriz e, assim sucessivamente até à iteração  $M$ . Note-se que o valor de  $N$ , corresponde ao número de segmentos de amostras obtidos para uma característica. Por isso, caso se pretenda considerar as três características enunciadas na secção 2.1.3, o número de elementos em cada linha da matriz  $X$ , é igual a três vezes o valor de  $N$ .

### Obtenção do vetor de classes – $y$

Cada um dos exemplos da matriz de características pertence a uma classe ou categoria. O vetor de classes pode ser representado da seguinte forma:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_M \end{bmatrix}$$

Figura 4.3: Vetor de classes.

onde cada elemento  $y_i$  corresponde à classe etiquetada. Para obtenção deste vetor, é necessário etiquetar manualmente o áudio. O processo resume-se a ouvir o áudio e anotar num ficheiro (formato.csv), em que intervalos de tempo ocorrem batidas de bolas.

Note-se que apenas as batidas de bola são registadas no ficheiro. Por isso, todos os eventos que não são registados constituem ruído. Cada um dos exemplos,  $y_i$ , do vetor  $y$  terá um valor dependente da expressão 4.2:

$$y_i = \begin{cases} \text{ball-hit} & \forall X_i: X_i \in [beginSample, endSample] \\ \text{noise} & \forall X_i: X_i \notin [beginSample, endSample] \end{cases} \quad (4.2)$$

onde  $[beginSample, endSample]$  é o intervalo de tempo em que a batida de bola ocorre.

### Criação do *dataset*

O processo de construção do *dataset* refere-se à junção da matriz de características ao vetor de classes. A janela desliza ao longo do áudio por isso, um conjunto de exemplos ou instâncias,  $X_i$ , corresponderão a uma classe,  $y_i$ , tal como se verifica na figura 4.4:

$$X = \begin{bmatrix} X_1 & X_2 & X_3 & \cdots & X_N & y_1 \\ X_2 & X_3 & X_4 & \cdots & X_{N+1} & y_2 \\ X_3 & X_4 & X_5 & \cdots & X_{N+2} & y_3 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \\ X_{N-1} & X_N & X_{N+1} & \cdots & X_{2N-2} & y_{N-1} \\ X_N & X_{N+1} & X_{N+2} & \cdots & X_{2N-1} & y_N \\ X_{N+1} & X_{N+2} & X_{N+3} & \cdots & X_{2N} & y_{N+1} \\ \vdots & \vdots & \vdots & & \vdots & \vdots \end{bmatrix}$$

Figura 4.4: Matriz de características.

O vídeo de *input*, a partir do qual o *dataset* foi construído é um vídeo obtido de forma não profissional com uma duração de 1 hora e 32 minutos. e constitui uma gravação de um treino realizado por um conjunto de atletas em ambiente *indoor*.

#### 4.3.2 Construção do Classificador

A escolha do algoritmo de aprendizagem automática a utilizar foi realizada através da ferramenta *Orange Data Mining* – ODM. Esta ferramenta permite aplicar de vários algoritmos aos dados, no sentido de verificar quais destes produzem os melhores resultados [Baptista, 2019].

A construção do classificador envolve também a escolha dos hiper-parâmetros que produzem os melhores resultados.

#### 4.3.3 Desenvolvimento da Aplicação *web*

A aplicação *web* desenvolvida tem duas páginas: página (1) que permite escolher o vídeo sobre o qual se pretende visualizar os resultados; página (2) que permite visualizar os resultados devolvidos pelo classificador (para um

determinado vídeo). O *design* destas páginas encontra-se nas figuras ?? e 4.5. Todo o processamento efetuado sobre o áudio é realizado antes de apresentar os resultados nesta aplicação.

(Colocar aqui o design da página 1).

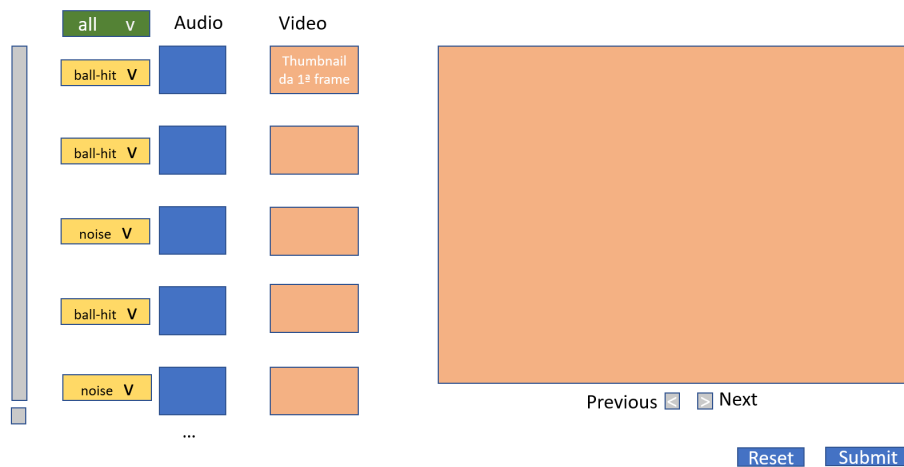


Figura 4.5: *Design* desejado da página *web* 2.



# Capítulo 5

## Implementação do Modelo

Este capítulo descreve todo o processo de implementação das etapas enunciadas na secção abordagem (4.3).

### 5.1 Construção do *Dataset*

O processo de criação do *dataset* pode ser representado pelo esquema que se encontra na figura 5.1.

Para simplificar o processo de anotação das batidas de bola (“*Human ear annotation*”), optou-se por segmentar o vídeo de longa duração em vídeos com 1 minuto. O processo de segmentação deu origem a cerca de 90 vídeos de curta duração. Para cada um desses vídeos, foram gerados os áudios correspondentes. Foram anotados 24 áudios.

A anotação das batidas de bola foi realizada com o auxílio da aplicação *Adobe Audition* [Adobe, 2013]. Esta ferramenta permite efetuar uma análise sobre áudio e verificar em que instantes ocorrem batidas de bola. A anotação é realizada num ficheiro *Excel*, que posteriormente é convertido para *CSV*.

Cada evento está registado no ficheiro *Excel* com os seguintes dados:

- O tipo de batida de bola (em raquete, no chão, na rede metálica, entre outros). Todos estes tipos pertencem à classe batida de bola;
- O intervalo de tempo em que ocorre a batida de bola (amostras inicial e final no áudio);
- O tipo de ambiente em que o treino é praticado (*indoor* ou *outdoor*);

- O lado do campo em que batida de bola ocorre.

O tipo de ambiente onde o desporto é praticado e o lado do campo onde ocorrem as batidas de bola estão fora do contexto do presente trabalho.

Na fase de extração de características iterou-se apenas sobre os áudios anotados. Sobre estes áudios realizou-se o processo descrito na secção 4.3.1 para obtenção da matriz de características com o auxílio da biblioteca *tensorflow*. Esta biblioteca permite realizar indexação e varrimento sobre *arrays* de forma simples.

Na fase de “*Labeling*” combinou-se a informação registada nos ficheiros *CSV* (batidas de bola) com as características tal como se verifica na figura 4.4, o que permitiu obter o *dataset* a utilizar na construção do modelo (“*Input data*”). Este processo de etiquetação foi realizado com base na equação 4.1.

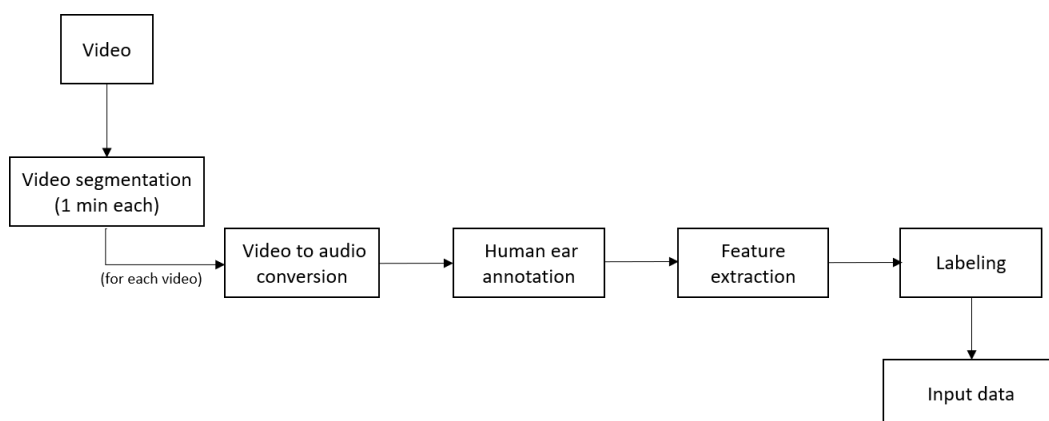


Figura 5.1: Processo detalhado de construção do *dataset*.

Considerando como primeira abordagem a primeira linha da tabela 4.4, o *dataset* obtido contém cerca de 65 mil exemplos.

## 5.2 Construção do Classificador

A construção do modelo pode ser descrita através da figura 5.3.

Através da ferramenta *Orange Data Mining* obtiveram-se os resultados que se encontram na figura 5.2. Esta figura contém os resultados (independentes das classes) referentes à aplicação de vários algoritmos ao *dataset* e sugere que os modelos que produzem os melhores resultados são o regressor



logístico e a rede neuronal. Os valores das métricas de avaliação (cf., 2.2.5) estão acima dos 85% e o erro (“*LogLoss*”) associado à classificação é igual ou inferior 35%.

As figuras B.2 e B.3 correspondem aos resultados para cada uma das classes (positivos e negativos). Observando separadamente estes resultados, verifica-se que a rede neuronal obtém os melhores resultados em termos de taxa de sucesso, média harmónica e erro associado. Por este motivo, a rede neuronal será a técnica de aprendizagem automática a utilizar para construir o classificador.

Devido a um desequilíbrio no *dataset*, os valores obtidos no caso da classe dos positivos (batidas de bola) são menos eficientes. Como solução, será aplicada a técnica de sub-amostragem na qual se diminui o número de exemplos da classe dos negativos (ruído).

Para verificar a qualidade dos modelos é utilizada a partição estratificada (“*Stratified*”). Neste processo de validação os dados são divididos em 5 *folds*.

Model	CA	F1	Precision	Recall	LogLoss
Logistic Regression	0.868	0.850	0.867	0.868	0.351
Naive Bayes	0.783	0.798	0.833	0.783	2.668
Neural Network	0.893	0.889	0.889	0.893	0.319
SVM	0.736	0.749	0.769	0.736	0.479
Tree	0.873	0.869	0.867	0.873	2.684

Figura 5.2: Resultados (independentes das classes) da aplicação de vários algoritmos aos dados.

Para construir e treinar o modelo utilizou-se a biblioteca *Python Keras*. Esta biblioteca usa internamente o módulo *tensorflow* e permite criar modelos de aprendizagem automática de forma simples e flexível [Keras, 2022].

A capacidade de generalizar do modelo foi verificada através do método de validação cruzada, *Stratified KFold*. Os dados foram divididos em 5 *folds*.

Os hiper-parâmetros considerados foram os seguintes:

- Temporais – duração da janela de dimensão fixa (*eventLength*) e número de amostras deslizadas a cada iteração (*hopLength*);
- Na rede rede neuronal – taxa de aprendizagem, método de otimização, número de épocas, número de *batches*, número de camadas na rede e de neurónios em cada camada.

Os parâmetros temporais têm influência na estrutura do *dataset* por isso também têm impacto na construção do modelo. Os parâmetros da rede neuronal têm impacto na forma como o modelo se ajusta aos dados recebidos.

Os modelos a serem analisados no Capítulo 6 foram construídos com base na tabela (referenciar a tabela)  
(Tabela).

O objetivo é escolher os hiper-parâmetros do modelo que produzem os melhores resultados:

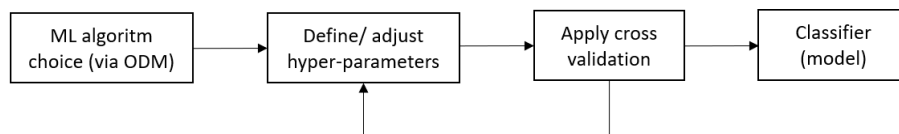


Figura 5.3: Processo de construção do classificador.

(Na imagem trocar a etapa de validação cruzada com a definição e ajuste dos hiper-parâmetros).

## 5.3 Desenvolvimento da Aplicação *web*

(Por fazer)

# Capítulo 6

## Validação e Testes

(Por fazer)



# Capítulo 7

## Conclusões e Trabalho Futuro

(Por fazer)



# Apêndice A

## Um Detalhe Adicional

O “apêndice” utiliza-se para descrever aspectos que tendo sido desenvolvidos pelo autor constituem um complemento ao que já foi apresentado no corpo principal do documento.

Neste documento utilize o apêndice para explicar o processo usado na **gestão das versões** que foram sendo construídas ao longo do desenvolvimento do trabalho.

É especialmente importante explicar o objetivo de cada ramo (“branch”) definido no projeto (ou apenas dos ramos mais importantes) e indicar quais os ramos que participaram numa junção (“merge”).

É também importante explicar qual a arquitetura usada para interligar os vários repositórios (e.g., Git, GitHub, DropBox, GoogleDrive) que contêm as várias versões (e respetivos ramos) do projeto.

Notar a diferença essencial entre “apêndice” e “anexo”. O “apêndice” é um texto (ou documento) que descreve trabalho desenvolvido pelo autor (e.g., do relatório, monografia, tese). O “anexo” é um texto (ou documento) sobre trabalho que não foi desenvolvido pelo autor.

Para simplificar vamos apenas considerar a noção de “apêndice”. No entanto, pode sempre adicionar os anexos que entender como adequados.





## Apêndice B

### Outro Detalhe Adicional

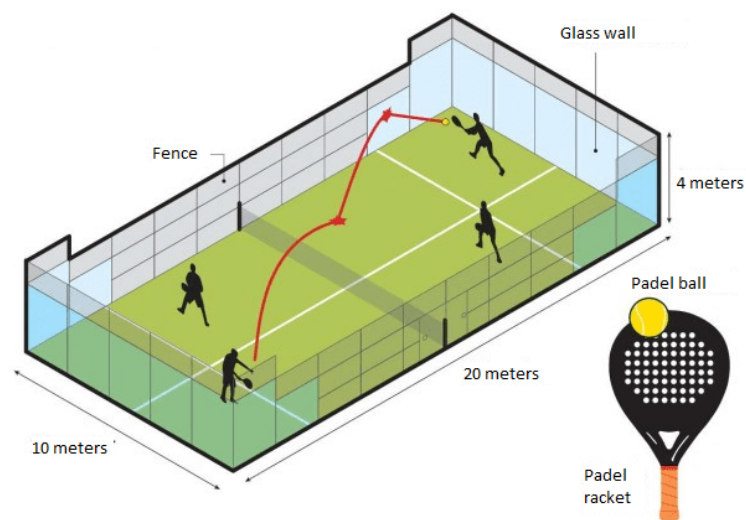


Figura B.1: Descrição pormenorizada do campo de padel. [Tennisnerd, 2015]

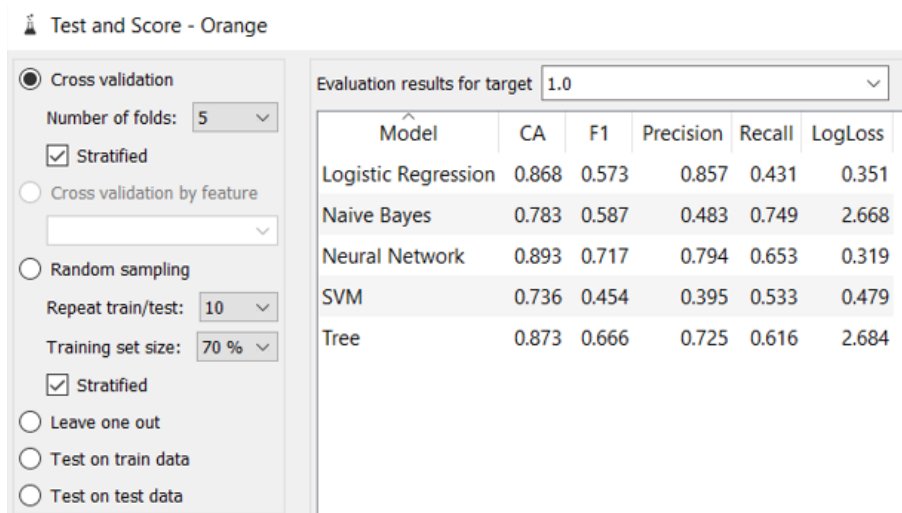


Figura B.2: Resultados da aplicação de vários algoritmos aos dados para a classe dos positivos.

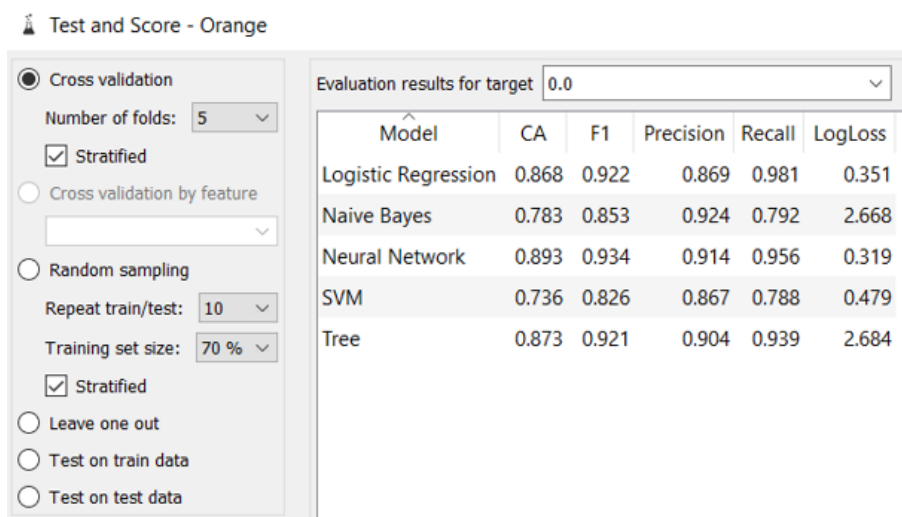


Figura B.3: Resultados da aplicação de vários algoritmos aos dados para a classe dos negativos.

# Bibliografia

[Adobe, 2013] Adobe, U. (2013). Adobe audition.

[Alhamid, 2020] Alhamid, M. (2020). What is cross-validation? <https://towardsdatascience.com/what-is-cross-validation-60c01f9d9e75>.

[Badr, 2019] Badr, W. (2019). Having an imbalanced dataset? here is how you can fix it. *Towards Data Science*, 22.

[Baptista, 2019] Baptista, B. (2019). Machine learning sem código. <https://medium.com/ensina-ai/machine-learning-sem-c%C3%B3digo-636d1a8f9081>.

[Brown, 2019] Brown, G. (2019). Digital audio basics: sample rate and bit depth. *Dostupno na: https://www.izotope.com/en/learn/digital-audio-basics-sample-rate-and-bit-depth.html [15. rujna 2020.]*.

[Brownlee, 2020] Brownlee, J. (2020). Tour of evaluation metrics for imbalanced classification. *Vermont Victoria*.

[Brownlee, 2021] Brownlee, J. (2021). How to choose an activation function for deep learning. *Machine Learning Mastery*.

[Brownlee, 2022] Brownlee, J. (2022). Types of classification tasks in machine learning. *Machine Learning Mastery*, 4p. Available online at: <https://machinelearningmastery.com/types-of-classification-in-machine-learning> (accessed November 25, 2021).

[Council et al., 2004] Council, N. R. et al. (2004). Hearing loss: Determining eligibility for social security benefits.

- [Courel-Ibáñez et al., 2019] Courel-Ibáñez, J., Martinez, B. J. S.-A., e Marín, D. M. (2019). Exploring game dynamics in padel: Implications for assessment and training. *The Journal of Strength & Conditioning Research*, 33(7):1971–1977.
- [Doshi, 2019] Doshi, S. (2019). Various optimization algorithms for training neural network. *Towards data science*, 13.
- [Evia, 2022] Evia, A. (visitado em Jul.2022). Nyquist sampling theorem.
- [Janiesch et al., 2021] Janiesch, C., Zschech, P., e Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3):685–695.
- [Keras, 2022] Keras (visitado em Ago, 2022). About keras. <https://keras.io/about/>.
- [Krishnan, 2021] Krishnan, S. (2021). How to determine the number of layers and neurons in the hidden layer. *URL: <https://medium.com/geekculture/introduction-to-neuralnetwork-2f8b8221fbd3>*.
- [LibreTexts, 2022a] LibreTexts, E. (2022a). Signal sampling. [https://eng.libretexts.org/Bookshelves/Electrical\\_Engineering/Signal\\_Processing\\_and\\_Modeling/Signals\\_and\\_Systems\\_\(Baraniuk\\_et\\_al.\)/10%3ASampling\\_and\\_Reconstruction/10.01%3ASignal\\_Sampling](https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Signal_Processing_and_Modeling/Signals_and_Systems_(Baraniuk_et_al.)/10%3ASampling_and_Reconstruction/10.01%3ASignal_Sampling).
- [LibreTexts, 2022b] LibreTexts, E. (2022b). Signal sampling. [https://eng.libretexts.org/Bookshelves/Electrical\\_Engineering/Signal\\_Processing\\_and\\_Modeling/Signals\\_and\\_Systems\\_\(Baraniuk\\_et\\_al.\)/10%3ASampling\\_and\\_Reconstruction/10.03%3ASignal\\_Reconstruction](https://eng.libretexts.org/Bookshelves/Electrical_Engineering/Signal_Processing_and_Modeling/Signals_and_Systems_(Baraniuk_et_al.)/10%3ASampling_and_Reconstruction/10.03%3ASignal_Reconstruction).
- [MachineLearning, 2017] MachineLearning (2017). Harmonic precision-recall mean (f1 score). <https://machinelearning.wtf/terms/harmonic-precision-recall-mean-f1-score/>.
- [McFee et al., 2015] McFee, B., Raffel, C., Liang, D., Ellis, D. P., McVicar, M., Battenberg, E., e Nieto, O. (2015). librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, p. 18–25. Citeseer.

- [Medium, 2019] Medium (2019). A beginner intro to neural networks. <https://purnasaigudikandula.medium.com/a-beginner-intro-to-neural-networks-543267bda3c8>.
- [Meyda, 2022] Meyda (visitado em Jul.2022). Audio feature extraction for javascript. <https://meyda.js.org/audio-features.html>.
- [Muralidhar, 2021] Muralidhar, K. (2021). What is stratified cross-validation in machine learning? *Medium*. Retrieved December, 12:2021.
- [Nyuytiymbiy, 2021] Nyuytiymbiy, K. (2021). Parameters and hyperparameters in machine learning and deep learning. *Medium*. <https://towardsdatascience.com/parameters-and-hyperparameters-aa609601a9ac> (April 22, 2021).
- [Rauber, 2022] Rauber, T. W. (visitado em Ago, 2022). Redes neuronais artificiais. *Documento de Apoio (visitado em Jul, 2022)*.
- [Room, 2021] Room, C. (2021). Audio feature extraction. *machine learning*, 16(17):51.
- [Rosão, 2012] Rosão, C. M. T. (2012). *Onset detection in music signals*. PhD thesis.
- [Sah, 2020] Sah, S. (2020). Machine learning: a review of learning types.
- [Sampaio et al., 2006] Sampaio, R., Cataldo, E., e BRANDÃO, A. (2006). Análise e processamento de sinais. *Sociedade Brasileira de Matemática Aplicada e Computacional*. São Paulo.
- [Semmlow, 2012] Semmlow, J. (2012). Chapter 3 - fourier transform: Introduction. In Semmlow, J., editor, *Signals and Systems for Bioengineers (Second Edition)*, Biomedical Engineering, p. 81–129. Academic Press, second edition edition.
- [Tennisnerd, 2015] Tennisnerd (2015). What is padel tennis? <https://tennisnerd.net/padel/what-is-padel-tennis/15893>.