



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

---

## Título do Trabalho

(eventual) imagem ilustrativa do trabalho – *dimensão*: até 13cm x 4cm

Nome do aluno (número)

Nome do aluno (número)

---

*Orientador(es)*

*Professor [Doutor]* Nome do orientador

*Professor [Doutor]* Nome do orientador

---

*Mês, ano*



# Resumo

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

A ferramenta desenvolvida neste projeto, permite extrair e reconhecer eventos relevantes (i.e. períodos em que ocorra uma maior troca de bolas) com base no vídeo da atividade desportiva do atleta (em treinos ou competições de padel /ténis). O processamento é realizado com base no áudio extraído do vídeo, e tem como base a extração de características ou padrões identificativos dos eventos, com o auxílio de técnicas de aprendizagem automática, baseadas em séries temporais.

No final, são realizadas estatísticas, que permitem obter um resumo detalhado do que foi registado no vídeo, dando uma perspetiva mais abrangente e objetiva do desempenho do atleta. A ferramenta identifica corretamente cerca de 85% dos eventos em análise, sendo necessários alguns ajustes para melhorar o processo de reconhecimento.

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.

The tool developed in this project, allows extracting and recognizing relevant events (i.e., periods when a greater exchange of balls occurs) based on the video of the athlete's sport activity (in the padel/tennis sport activity). The processing is done based on the audio extracted from the video, and is based on the extraction of patterns identifying the events, with the help of machine learning techniques, based on time series.

At the end, statistics are performed, allowing a detailed summary of what was recorded in the video, 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 . . . . .	2
1.2 Principais Contributos . . . . .	3
1.3 Validação e Testes . . . . .	4
1.4 Estrutura do Relatório . . . . .	4
<b>2 Fundamentos</b>	<b>7</b>
2.1 Conceitos de Processamento de Sinais . . . . .	7
2.1.1 Amostragem . . . . .	7
2.1.2 Transformada de Fourier . . . . .	8
2.2 Conceitos de Aprendizagem Automática . . . . .	8
2.2.1 <i>Dataset</i> . . . . .	9
2.2.2 Características . . . . .	9
2.2.3 Classes . . . . .	11
2.2.4 Desiquilíbrio no <i>Dataset</i> . . . . .	11
2.2.5 Hiper-parâmetros . . . . .	11

2.2.6	Classificação e Validação Cruzada . . . . .	12
2.2.7	Redes Neurais . . . . .	13
<b>3</b>	<b>Trabalho Relacionado</b>	<b>15</b>
<b>4</b>	<b>Modelo Proposto</b>	<b>17</b>
4.1	Descrição Geral do Sistema . . . . .	17
4.2	Requisitos . . . . .	18
4.3	Abordagem . . . . .	20
4.3.1	Construção do <i>dataset</i> . . . . .	20
4.3.2	Construção do classificador . . . . .	23
4.3.3	Criação da interface . . . . .	25
<b>5</b>	<b>Implementação do Modelo</b>	<b>27</b>
<b>6</b>	<b>Validação e Testes</b>	<b>29</b>
<b>7</b>	<b>Conclusões e Trabalho Futuro</b>	<b>31</b>
<b>A</b>	<b>Um Detalhe Adicional</b>	<b>33</b>
<b>B</b>	<b>Outro Detalhe Adicional</b>	<b>35</b>
	<b>Bibliografia</b>	<b>37</b>

# Lista de Tabelas

4.1	Requisitos funcionais do sistema em desenvolvimento para a construção do <i>dataset</i> . . . . .	18
4.2	Requisitos funcionais respeitantes ao processo de classificação. . . . .	18
4.3	Requisitos não funcionais do sistema. . . . .	19
4.4	Valores de duração (em segundos), amostras delizadas e N a considerar, para um valor de <i>samplingRate</i> igual a 44100Hz. . . . .	22
6.1	Uma tabela . . . . .	29



# Lista de Figuras

1.1	Raquete e bola de padel . . . . .	1
2.1	Características (cima e meio) e áudio (baixo) correspondente. .	10
2.2	Representação típica de uma rede neuronal. . . . .	13
4.1	Representação geral da ferramenta em desenvolvimento. . . . .	17
4.2	Matriz de características. . . . .	21
4.3	Vetor de classes. . . . .	22
4.4	Matriz de características. . . . .	23
4.5	Resultados da aplicação de vários algoritmos aos dados. . . . .	24
6.1	Uma figura . . . . .	29
B.1	Representação de um neurónio de uma rede neuronal. . . . .	35





# Capítulo 1

## Introdução

Para que a atividade desportiva de um atleta, numa determinada modalidade, melhore é necessário que este seja monitorizado de uma perspetiva externa. Para esse efeito, é necessário que o atleta seja supervisionado durante a sua atividade física, ou que a última seja registada e posteriormente analisada (registo em papel ou em vídeo).

A anotação de eventos sonoros em vídeo consiste no processo de análise do áudio extraído de um vídeo previamente gravado, no sentido de recolher informações relativas a possíveis eventos relevantes. Este processo, constitui uma tarefa complexa, já que envolve o processamento sobre séries temporais, a partir das quais se podem extrair características e identificar os eventos. Embora esse processamento possa ter aplicações nas mais variadas componentes, este trabalho destina-se apenas à análise da componente desportiva de padel, no sentido de fornecer a atletas e/ou treinadores um complemento no processo de treino. Neste sentido, os eventos sonoros a considerar são batidas de bola em raquetes de padel [Courel-Ibáñez et al., 2019]. A figura que se segue, corresponde à raquete e bola utilizadas no desporto de padel:



Figura 1.1: Raquete e bola de padel

Uma figura com mais detalhe 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*). No entanto, este trabalho analisa apenas vídeos em ambiente *indoor*.

É ainda importante referir que, dada a complexidade de todo o processamento a realizar, bem como o volume de dados e o custo computacional associado, o primeiro será realizado em modo *offline*.

## 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. De acordo com [Council et al., 2004], o ouvido humano consegue fazer a distinção entre sons, considerando as características dos sons emitidos: duração, intensidade, tom, entre outras. Pelo que, é necessário proporcionar à máquina a capacidade de reconhecer batidas de bola de forma semelhante à humana. Neste sentido, o que se pretende, é utilizar a máquina para simular o ouvido humano. Para esse efeito, far-se-á uso de aprendizagem automática.

A aprendizagem automática permite aplicar algoritmos aos dados recebidos pela máquina, e reconhecer nestes as batidas de bola.

Antes do processo de aprendizagem, não existindo dados relativos aos eventos pretendidos, ou que estejam etiquetados nas condições desejadas, procedeu-se à construção de um *dataset* de raiz.

Neste sentido, pretende-se desenvolver uma ferramenta que auxilie na atividade desportiva do atleta ou na monitorização de quem o treina. A aplicação processa o vídeo introduzido, realiza um processamento interno e devolve uma análise intensiva dos períodos onde existem eventos que possam interessar aos utilizadores, como por exemplo, períodos onde existem mais batidas de bola. Esta aplicação 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 a construção

do modelo.

## 1.2 Principais Contributos

Os contributos do sistema em desenvolvimento dividem-se em três componentes: a criação do *dataset*; a construção de um modelo; elaboração de uma interface que auxilia no processo de aumento do *dataset*. Nesta secção será feita uma descrição geral dessas três componentes.

Os dados serão colocados numa estrutura de dados denominada *dataset*. O processo de construção do *dataset* envolve os seguintes passos:

- Extração do áudio a partir do vídeo – resume-se à conversão do vídeo no áudio correspondente;
- Anotação dos eventos relevantes presentes no áudio – refere-se à etiquetagem de batidas de bola no áudio obtido no ponto anterior;
- Extração das características do áudio – refere-se à obtenção de padrões no áudio que possam identificar potenciais batidas de bola;
- Junção dos dados resultantes dos dois pontos anteriores.

O *dataset* é introduzido num algoritmo de dados que identifica batidas de bola. O processo de construção do modelo pode ser descrito da seguinte forma:

- Escolha do algoritmo de aprendizagem automática – escolha do algoritmo que melhor se adequa ao reconhecimento de batidas de bola;
- Escolha dos hiper-parâmetros que produzem os melhores resultados no treino do modelo;
- Aplicação de validação cruzada, no sentido de avaliar o desempenho do modelo em construção, fazendo uso de todo o *dataset* disponível.

O sistema em desenvolvimento permitirá também que o utilizador valide os resultados devolvidos pelo classificador, o que poderá contribuir para melhorar a qualidade do próprio modelo e do *dataset*:

- Disponibilizar numa interface, os resultados (em termos de eventos) retornados pelo classificador, evidenciando na série temporal onde ocorrem os eventos correspondentes a batidas de bola;
- Nessa mesma interface disponibilizar a opção de validar os eventos considerados pelo modelo como sendo ou não batidas de bola.

## 1.3 Validação e Testes

Validação e testes sobre o classificador: avaliação do classificador (validação cruzada) Validação e testes sobre a ferramenta final (testes de usabilidade)

Falar um pouco sobre os resultados obtidos em termos de percentagens e a forma como o sistema se comporta face a novos dados de input.

Fazer uma apreciação global dos resultados alcançados (como está no Resumo e Abstract).

## 1.4 Estrutura do Relatório

O relatório está organizado da seguinte forma:

(A descrição dos capítulos não precisa de ser muito detalhada. Deve se ser o mais resumida possível.)

- O Capítulo 1 é introduzida a intenção de utilizar a aprendizagem automática para reconhecer padrões identificativos de batidas de bola em jogos/ treinos de padel. Neste capítulo, são também enunciados os principais contributos da aplicação em desenvolvimento. No final, é realizada uma apreciação global dos resultados obtidos.
- O Capítulo 2 descreve e faz referência a trabalhos relacionados com o trabalho corrente. Estes trabalhos que foram sendo analisados durante o desenvolvimento do trabalho e baseiam-se também no uso da inteligência artificial para reconhecer eventos em modalidades desportivas.
- O Capítulo 3 visa apresentar uma análise geral dos requisitos do sistema e a abordagem considerada para desenvolver a aplicação.
- (Falar do capítulo 4)

- (Falar do capítulo 5)
- (Falar do capítulo 6)



# Capítulo 2

## Fundamentos

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

### 2.1 Conceitos de Processamento de Sinais

De acordo com [SAMPAIO et al., 2006], um sinal corresponde a uma grandeza física que varia ao longo do tempo. Um sinal pode também ser descrito como uma série temporal. Em termos matemáticos, um sinal varia em função de uma ou mais variáveis, que podem ser contínuas (valores contínuos) ou discretas (valores discretos). Neste sentido, podemos ter dois tipos de sinais: contínuos ou discretos. Neste trabalho, serão analisados apenas sinais discretos, uma vez que se pretende realizar processamento sobre áudio, que foi previamente amostrado numa determinada frequência (a abordar na secção seguinte, ). O presente trabalho também efetuará a análise de sinais no domínio do tempo e no domínio da frequência.

#### 2.1.1 Amostragem

O processo de amostragem refere-se à conversão de um sinal contínuo, num sinal discreto. Este algoritmo escolhe pontos equi-espaçados do sinal contínuo para gerar os pontos do sinal discreto [LibreTexts, 2022a]. O sinal (discreto) obtido evolui, no domínio do tempo, em função de amostras ou valores discretos, e não de valores contínuos.

O intervalo entre as amostras denomina-se período de amostragem e é dado em segundos(s). O inverso deste valor corresponde à frequência de

amostragem, e refere-se ao número de amostras que ocorre no sinal a cada segundo. A frequência de amostragem é dada em Hertz(Hz). Neste algoritmo de amostragem, é removida informação do sinal original (contínuo), que em parte pode ser recuperada no processo inverso com o auxílio da interpolação ou reconstrução do sinal [LibreTexts, 2022b]. O ouvido humano deteta sons na gama de frequências compreendida entre 20Hz e 20kHz. Neste sentido e de acordo com o teorema de *Nyquist* [Evia e Arnold, 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 44100Hz, que é o valor padrão, tendo em conta a gama de frequência do ouvido humano.

### 2.1.2 Transformada de Fourier

A transformada de Fourier ou Fourier Transform (FT), é uma função matemática que decompõe um sinal nas várias frequências ou sinusóides constituintes [Semmlow, 2012]. Assim, o sinal de entrada é um sinal no domínio do tempo, ao passo que o sinal de saída é um sinal no domínio da frequência ou espectro. A transformada de *Fourier* é útil para verificar, por exemplo, em que zonas do espectro, existe mais ou menos energia concentrada. Um caso particular da transformada de *Fourier* é a transformada de *Fourier* a curto prazo ou *Short-time Fourier transform* (STFT), que divide o sinal em grupos de amostras sobre os quais aplica a transformada de *Fourier*.

## 2.2 Conceitos de Aprendizagem Automática

A aprendizagem automática ou *Machine Learning* (ML) é ramo da inteligência artificial, que se refere à capacidade que os sistemas computacionais adquirem para realizar determinadas tarefas de forma automática [Janiesch et al., 2021]. Esses sistemas aprendem a resolver uma determinada tarefa, recorrendo à análise de padrões existentes em dados específicos do problema a resolver. Ao analisarem esses dados, é construído um modelo, que, a grosso modo, corresponde ao “cérebro” do sistema computacional, e o auxilia na resolução da tarefa, para a qual foi designado.

De acordo com [Sah, 2020], existem três tipos de aprendizagem automática:



- Aprendizagem Supervisionada – tipo de aprendizagem em que os dados de *input* utilizados para o processo de treino do algoritmo, contêm as características e a classe correspondente;
- Aprendizagem Não Supervisionada – tipo de aprendizagem em que os dados de *input* utilizados para o processo de treino do algoritmo, contêm apenas as características;
- Aprendizagem por Reforço – tipo de aprendizagem utilizado quando o sistema computacional tem como objetivo realizar uma tarefa que envolve uma sequência de decisões. O sistema ganha reforços positivos ou negativos, mediante a ação realizada.

No caso deste trabalho, a aprendizagem automática irá permitir aplicar algoritmos aos dados recebidos pela máquina, e reconhecer nestes, padrões identificativos de batidas de bola. Pelo que, o reconhecimento de padrões nos dados constitui uma forma de a máquina aprender a identificar os eventos relevantes ou batidas de bola. O tipo de aprendizagem automática a considerar será a supervisionada.

### 2.2.1 *Dataset*

Um *dataset* corresponde a um conjunto de dados, que é utilizado para treinar ou ensinar algoritmos de inteligência artificial a resolver tarefas [House, 2022]. Um *dataset* é constituído por características e classes. Esta estrutura de dados é organizada em forma de tabela (com colunas e linhas). Cada linha corresponde a um exemplo ou observação, ao passo que cada uma das colunas (à exceção de uma delas, que corresponde à classe) representa uma característica.

### 2.2.2 Características

As características correspondem a colunas no *dataset* e representam os dados de *input* utilizados para o treino do algoritmo ou modelo.

Neste trabalho, pretende-se extrair a partir do áudio, as seguintes características: deteção de picos (*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, tal como

o nome indica, corresponde a detetar o instante em que se inicia um determinado som ou evento [Rosão, 2012]. Já 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 evento. O fluxo espectral pode também ser definido como uma medida do quão depressa o espectro de um sinal varia [Meyda, 2022].

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

A figura abaixo constitui uma representação das três características enunciadas:

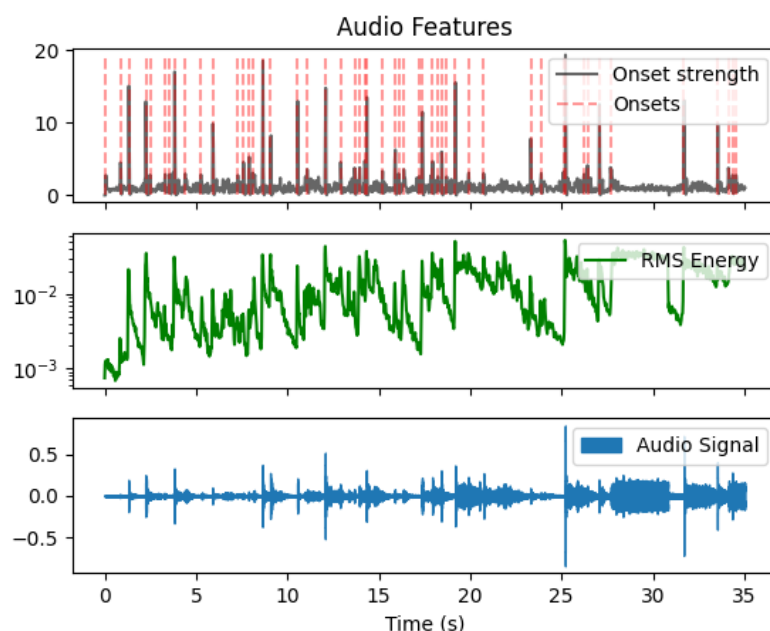


Figura 2.1: Características (cima e meio) e áudio (baixo) correspondente.

Observando a figura acima, verifica-se que os picos das duas componentes do *Onset*, bem como o RMS estão em concordância com o início dos sons no áudio original. O gráfico representativo do fluxo espectral, apresenta picos com maior amplitude, onde o áudio original tem mais intensidade.

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

### 2.2.3 Classes

Os exemplos ou observações de um *dataset* estão divididos em categorias. Essas categorias denominam-se classes. No caso da aprendizagem supervisionada, o algoritmo vai processando cada uma desses exemplos e faz a análise dos padrões encontrados nos dados. Depois de realizar essa análise, escolhe uma classe (*output*), que pode estar correta ou não em relação à classe de *input* ou real. Visto que este trabalho se destina ao reconhecimento de batidas de bola, serão apenas consideradas as classes: batida de bola e não batida de bola.

### 2.2.4 Desequilíbrio no *Dataset*

O desequilíbrio no *dataset* ocorre quando existe uma diferença considerável no número de exemplos que existe nas categorias do conjunto de dados. Por outras palavras, existe desequilíbrio no dataset quando a distribuição de exemplos pelas classes é desigual [Badr, 2019]. O processo de treino com dados, cuja distribuição é desigual, pode resultar na criação de um modelo que reconhece melhor certas classes em relação a outras.

Ainda de acordo com [Badr, 2019] e assumindo que se trata de um problema de classificação binária, é possível solucionar este problema das seguintes formas:

- Amostragem – aumentar o número de exemplos da classe em minoria (sobre-amostragem) ou reduzir o número da classe em maioria (sub-amostragem), no sentido de obter uma proporção de 50:1 entre as classes.
- Juntar vários algoritmos de treino – para obter um desempenho melhor, comparativamente à utilização de apenas um dos algoritmos.

### 2.2.5 Hiper-parâmetros

De acordo com [Nyuytiymbiy, 2021], em aprendizagem automática, um hiper-parâmetro constitui um valor utilizado pelo algoritmo ou 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.

### 2.2.6 Classificação e Validação Cruzada

O processo de classificação é utilizado pelos algoritmos de *Machine Learning* para atribuir uma classe ou categoria a um conjunto de exemplos [Brownlee, 4]. Após este processo, é construído um modelo ou classificador, que através dos padrões presentes nos dados de *textinput*, prevê a classe à qual esses dados pertencem. Ainda de acordo com a referência anterior, existem três tipos de classificação:

- Classificação Multi-Classe – aplica-se a tarefas em que existem mais do que duas classes. Um exemplo não pode pertencer, simultaneamente, a mais do que uma classe.
- Classificação Binária – aplica-se a problemas em que existem apenas duas classes. Um exemplo não pode pertencer, simultaneamente, a mais do que uma classe.
- Classificação Multi-Label – aplica-se a problemas em que existem duas classes ou mais classes. Um exemplo pode pertencer, simultaneamente, a mais do que uma classe.

Neste trabalho, o problema de classificação é binário, já que o sistema computacional deve apenas distinguir eventos que são batidas de bola de eventos que não o são.

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 ou treino, ao passo que o grupo de teste é utilizado no processo de classificação. Esta distribuição dos dados pode não ser suficiente para que o modelo aprenda a generalizar, no caso de ter de analisar dados novos. Como solução, pode ser utilizada a validação cruzada.

A validação cruzada é uma técnica utilizada para avaliar o desempenho de um modelo, quando analisa dados novos [Alhamid, 2020].

### 2.2.7 Redes Neurais

Uma rede neuronal ou *Artificial Neural Network* (ANN) é um algoritmo ou técnica de aprendizagem automática que tem como objetivo imitar o cérebro humano na realização de uma determinada tarefa, simulando a forma como os neurónios comunicam entre si [Burns, 2022].

À semelhança do cérebro humano, o neurónio é a unidade básica da rede neuronal. Um neurónio funciona como uma função que recebe um determinado *input*, processa-o e devolve um *output*. No capítulo B encontra-se uma figura ilustrativa de um neurónio consituente de uma rede neuronal.

Uma rede neuronal está organizada em camadas, que são constituídas por neurónios. A primeira camada ou a camada de *input*, tem a mesma quantidade de neurónios que o número de características do *dataset*. A última camada ou a camada de *output*, contém tantos neurónios quanto o número de classes ou categorias. Entre a camada de *input* e a camada de *output* existem uma ou mais camadas intermédias. A figura 2.2 representa de uma rede neuronal, onde estão delimitadas as ligações entre os vários neurónios constituintes:

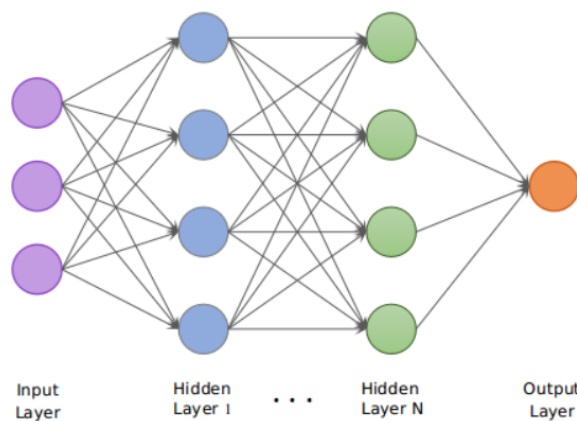


Figura 2.2: Representação típica de uma rede neuronal.

O número de camadas, bem como o número de neurónios constituinte devem ser escolhidos com rigor, a fim de garantir que o modelo obtém a melhor classificação possível [Krishnan, 2021].

Tal como se verifica na figura 2.2, quando um neurónio recebe os vários *inputs* (de outros neurónios), associa a cada um destes um peso, que vai

atualizando ao longo do processo de aprendizagem [Medium, 2019].

De acordo com [Baeldung, 2022], durante a aprendizagem, os dados propagam-se ao longo da rede neuronal várias vezes. O processo de propagação de todos os dados ao longo da rede neuronal denomina-se época. Neste sentido, uma época corresponde a um ciclo de passagem de todos os dados do *dataset* ao longo da rede neuronal. Uma época é realizada ao longo de várias iterações. Uma iteração é o número de grupos de dados (*batches*), constituintes do *dataset*, necessários para completar uma época. Pelo que, o (*batch*) é o número de exemplos ou observações que atravessam a rede neuronal numa iteração.

## Capítulo 3

# Trabalho Relacionado

Este capítulo faz referência a artigos e outros trabalhos de cariz técnico que se relacionam com o trabalho em desenvolvimento: uso de algoritmos de aprendizagem para reconhecimento de eventos em modalidades desportivas.

No fundo, pegar nos artigos científicos e relacionar/ comparar com o que se pretende no trabalho: tecnologias utilizadas, métodos de desenvolvimento, área a que se aplica (desporto).

Trabalho relacionado aqui . . .

Aqui terá certamente necessidade de citar (fazer referência) a vários trabalhos anteriormente publicados e que foi analisando ao longo de todo o seu projeto. Esses trabalhos devem ser apresentados com os seguintes objetivos essenciais:

- delimitar o contexto onde o seu projeto se insere,
- definir claramente os aspetos diferenciadores (inovadores) do seu projeto,
- identificar e caracterizar os pressupostos (teóricos ou tecnológicos) em que o projeto se baseia.

Cada trabalho a que fizer referência precisa de ser corretamente identificado. Essa identificação depende do tipo de publicação do trabalho. Um trabalho terá sido publicado em revista científica, e.g., [Elzinga e Mills, 2011], outro em ata de conferência internacional, e.g., [Boutilier et al., 1995], outro em livro, e.g., [Bellifemine et al., 2007], ou apenas em capítulo de livro, e.g., [Wooldridge, 2000], ou pode ainda incluído numa coleção, e.g.,

[Howard e Matheson, 1984] e há também a hipótese de ser uma “publicação de proveniência diversa”, como no caso em que o “o sítio na Internet” é a principal forma de publicação, e.g., [Python3.2.3, 2012] e, por fim, a publicação pode ser um relatório técnico, e.g., [Marin, 2006].

Para conseguir lidar de forma adequada com as referências é importante construir um acervo e ter um mecanismo para geração automática (e correta) das referências que vai fazendo ao longo do texto.

Atualmente, as publicações têm também informação sobre o modo como devem ser corretamente citadas; em geral essa informação segue o formato `BIBTEX`.

Para fazer referência a um trabalho é necessário seguir as boas regras (sintáticas) para uma citação correta, mas isso não é suficiente; falta a “semântica”. Ou seja, é também preciso descrever o essencial do trabalho que está a citar. É necessário explicar esse trabalho e enquadrá-lo, no texto, de modo a tornar clara a relação entre esse trabalho e o seu projeto.



# Capítulo 4

## Modelo Proposto

Neste capítulo, será realizada uma descrição geral de todo o sistema. De seguida, serão abordados os requisitos da ferramenta em desenvolvimento. A seguir, será feita uma descrição de qual a abordagem adotada para desenvolver o sistema.

### 4.1 Descrição Geral do Sistema

De forma geral, o funcionamento do sistema pode ser representado como se mostra na figura 4.1:

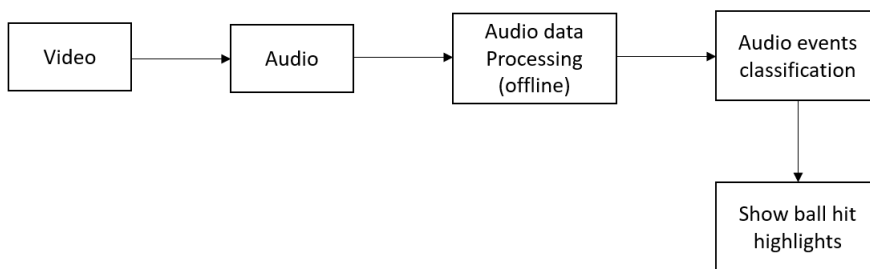


Figura 4.1: Representação geral da ferramenta em desenvolvimento.

Assim, quando um utilizador submete um vídeo, este é convertido no áudio correspondente. De seguida, o áudio é processado e, posteriormente, são classificados os eventos nele presentes. Numa fase final, através de uma interface, são visualizados os resultados da classificação.

## 4.2 Requisitos

Do ponto de vista do utilizador, a ferramenta em desenvolvimento destina-se a atletas ou treinadores que tenham como objetivo analisar os treinos. Pretende-se desenvolver um sistema que reconheça os eventos correspondentes a batidas de bola, e que contribua para auxiliar o processo de classificação.

Os requisitos do sistema podem ser funcionais e não funcionais. Os primeiros, que também podem ser designados por funções de sistema, referem-se a funcionalidades desempenhadas pelo sistema. Neste caso, os requisitos funcionais podem ainda ser divididos em dois grupos:

- Grupo das funcionalidades associadas à construção do *dataset*;
- Grupo das funcionalidades associadas ao processo de classificação.

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

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

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

Tabela 4.2: Requisitos funcionais respeitantes ao processo de classificação.

Note-se que as funcionalidades enunciadas acima constituem grande parte dos contributos deste projeto (referidos na secção 1.2).

Os requisitos não funcionais, também designados atributos do sistema, correspondem a algo que o sistema deve realizar. A tabela 4.3 representa os atributos não funcionais do sistema.

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

Tabela 4.3: Requisitos não funcionais do sistema.

## 4.3 Abordagem

Neste 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;
- Criação de uma interface que contribui para aumentar o *dataset*.

### 4.3.1 Construção do *dataset*

Tal como referido anteriormente, pretende-se, realizar processamento *offline* sobre áudio com batidas de bola, e identificar em que instantes temporais estas ocorrem. Para esse efeito, é necessário numa fase inicial, criar um *dataset*. O *dataset* em questão, corresponde a um conjunto de dados com padrões identificativos desses eventos.

Neste sentido, 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:

$$f(X) = y \quad (4.1)$$

A função  $f$  corresponde a um modelo que deve receber a matriz de características e, classificar de maneira correta cada um dos exemplos desse mesmo vetor. Esse modelo é treinado com o *dataset*, que constitui uma forma de permitir que o primeiro seja capaz de reconhecer as batidas de bola.

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

No sentido de obter as características ou padrões presentes no áudio (extraído do vídeo), será realizado um varrimento sobre o mesmo com uma janela 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.

De acordo com [McFee et al., 2015], no cálculo das características enunciadas na secção 2.2.2, o *package* *librosa* considera um salto (“*hop*”) ou deslizamento 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. 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.

A matriz  $X$  é constituída por  $M$  linhas ou exemplos. Por sua vez, cada exemplo tem  $N$  colunas ou características, pelo que a matriz  $X$  pode ser representada na figura 4.4 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 4.4, corresponde ao número de segmentos abrangidos pela janela fixa. Admitindo que o áudio é obtido a uma frequência de amostragem igual a 44100Hz (secção 2.1.1), o valor de  $N$  pode ser dado pela seguinte expressão:

$$N = \frac{eventLength \times samplingRate}{hopLength} \quad (4.2)$$

onde na equação *eventLength*, corresponde à duração de um evento ou batida de bola, *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	512	43
0.5	1024	22
0.5	2048	11
0.7	512	60
0.7	1024	30
0.7	2048	15
1.0	512	86
1.0	1024	43
1.0	2048	22

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

Voltando ao processo de deslizamento da janela e observando ainda figura 4.4, a primeira iteração corresponderá à primeira linha da matriz  $X$ . A segunda linha corresponderá à segunda iteração e, assim sucessivamente até à iteração  $M$ . Os elementos da matriz correspondem às características obtidas da janela deslizante a cada iteração.

### 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:

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

Figura 4.3: Vetor de classes.

onde cada elemento corresponde à classe etiquetada. Para obtenção deste vetor, é necessário etiquetar o áudio, a partir do qual se obtém a matriz de características,  $X$ . O processo resume-se a ouvir o áudio e anotar num ficheiro .csv, em que instantes de tempo (em termos de amostras) ocorrem batidas de bolas.

Note-se que apenas eventos correspondentes a batidas de bola são registados no ficheiro. Desta maneira, admite-se que os eventos que não são registados, são interpretados como não sendo batidas de bola. Assim, cada um dos exemplos do vetor  $y$  terá um valor dependente da expressão 4.3:

$$y = \begin{cases} \text{Ball hit} & \forall X_{m,n}: X_{m,n} \in [\text{beginSample}, \text{endSample}] \\ \text{Non ball hit} & \forall X_{m,n}: X_{m,n} \notin [\text{beginSample}, \text{endSample}] \end{cases} \quad (4.3)$$

### Criação do *dataset*

Após a obtenção da matriz de características e o vetor de classes correspondente, o processo de criação do *dataset* refere-se à junção destas duas componentes. Considerando que a janela desliza ao longo do áudio, um conjunto de exemplos ou instâncias,  $X_i$ , corresponderão a uma mesma classe,  $y_i$ :

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

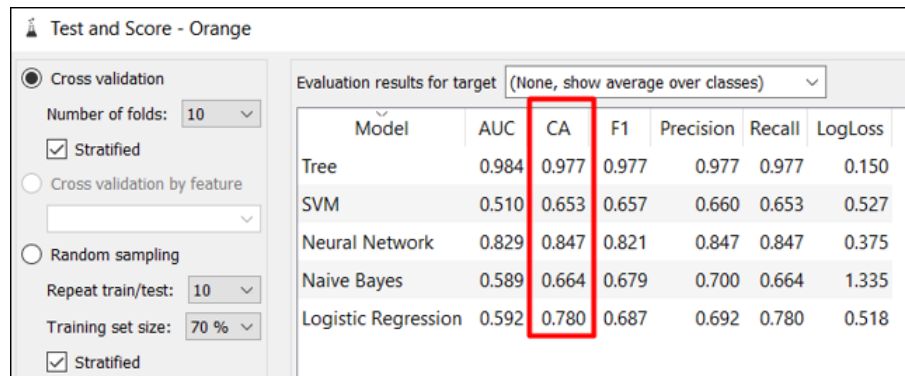
Figura 4.4: Matriz de características.

As linhas ou instâncias seguintes, que não estão evidenciadas na matriz, estão associadas a outra classe e, assim sucessivamente.

#### 4.3.2 Construção do classificador

Tal como referido na secção 2.2.6, um classificador permite construir um modelo que realiza a discriminação entre classes, às quais os exemplos pertencem.

No entanto antes de realizar o processo de classificação, é necessário definir qual a técnica de aprendizagem automática se deve utilizar. Para esse efeito, fez-se uso da ferramenta **Orange Data Mining** (ODM), que permite simular a aplicação de vários algoritmos aos dados, no sentido de verificar quais destas produzem os melhores resultados.



Model	AUC	CA	F1	Precision	Recall	LogLoss
Tree	0.984	0.977	0.977	0.977	0.977	0.150
SVM	0.510	0.653	0.657	0.660	0.653	0.527
Neural Network	0.829	0.847	0.821	0.847	0.847	0.375
Naive Bayes	0.589	0.664	0.679	0.700	0.664	1.335
Logistic Regression	0.592	0.780	0.687	0.692	0.780	0.518

Figura 4.5: Resultados da aplicação de vários algoritmos aos dados.

A figura 4.5 acima, contém os resultados referentes à aplicação de vários modelos a um conjunto de dados. No processo de treino é utilizada validação cruzada estratificada (*“Stratified”*), em que os dados são divididos em 10 folds e é mantida a distribuição dos dados nos grupos. Os dados de *input* são resultado da etiquetagem e extração de características evidenciada na secção (referenciar secção da implementação onde está a construção do dataset). Esta figura sugere que as técnicas que produzem os melhores resultados são a árvore de decisão e as redes neuronais. A árvore de decisão produz os melhores resultados, particularmente a classificação e o erro associado. No entanto, as redes neuronais serão escolhidas em detrimento da árvore de decisão, já que constituem uma área da aprendizagem automática em evolução nos últimos anos (referenciar na bibliografia).

Para o processo de treino do modelo, será utilizado o *package Keras*. Esta biblioteca disponibiliza uma interface em *Python* para desenvolver e avaliar algoritmos de redes neuronais [Brownlee, 2020].

Depois de escolhido o algoritmo e a API que permite criar o modelo, devem ser escolhidos os hiper-parâmetros que produzem os melhores resultados de classificação.

No decurso da construção do classificador e para poder fazer uso de todo



---

o *dataset*, será utilizada a validação cruzada. Como referido no Capítulo 2.2.6, a validação cruzada constitui também uma forma de verificar qual o desempenho ou qualidade do modelo.

### 4.3.3 Criação da interface

A interface a desenvolver contém uma síntese dos eventos mais relevantes do ponto de vista do utilizador. Numa primeira fase, serão elaborados os *mockups* que representam o *design* das páginas *web* a serem visualizadas pelo utilizador.



# Capítulo 5

## Implementação do Modelo

Implementação do modelo aqui ...; pode precisar de referir o capítulo 4 ...

Aqui identifica as opções teóricas e justifica as dependências tecnológicas assumidas neste projeto. Descreva com rigor formal e detalhe adequado e faça evidência de tudo o que foi proposto e desenvolvido especificamente no contexto deste projeto. Aqui a ênfase está naquilo que foi de facto concretizado neste projeto.

O leitor quer detalhes de concretização. Ele já está enquadrado no tema (cf., capítulo 3), já conhece os aspetos mais abstratos da sua proposta (cf., capítulo 4) e agora precisa de entender os detalhes para conseguir também interpretar as validações e testes que posteriormente (cf., capítulo 6) lhe irá apresentar.



# Capítulo 6

## Validação e Testes

Validação e testes aqui ...; pode precisar de referir o capítulo 4 ou alguma das suas secções, e.g., a secção ?? ...

Pode precisar de apresentar tabelas. Por exemplo, a tabela 6.1 apresenta os dados obtidos na experiência ...

$c_1$	$c_2$	$c_3$	$\sum_{i=1} c_i$
1	2	3	6
1.1	2.2	3.3	6.6

Tabela 6.1: Uma tabela

Para além de tabelas pode também precisar de apresentar figuras. Por exemplo, a figura 6.1 descreve ...

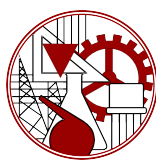


Figura 6.1: Uma figura

**Atenção.** Todas as tabelas e figuras, e.g., diagramas, imagens ilustrativas da aplicação em funcionamento, têm que ser devidamente enquadradas no texto antes de serem apresentadas e esse enquadramento inclui uma explicação da imagem apresentada e eventuais conclusões (interpretações) a tirar dessa imagem.



## Capítulo 7

# Conclusões e Trabalho Futuro

Conclusões e trabalho futuro aqui . . .

Quais as principais mensagens a transmitir ao leitor deste trabalho? O leitor está certamente interessado nos temas aqui abordados. Em geral procurará, neste projeto, pistas para algum outro objetivo. Assim, é muito importante que o leitor perceba rapidamente a relação entre este trabalho e o seu próprio (do leitor) objetivo.

Aqui é o local próprio para condensar a experiência adquirida neste projeto e apresentá-la a outros (futuros leitores).

O pressuposto é o de que este projeto é um “elemento vivo” que recorreu a outros elementos (cf., capítulo 3) para ser construído e que poderá servir de suporte à construção de futuros projetos.





# 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

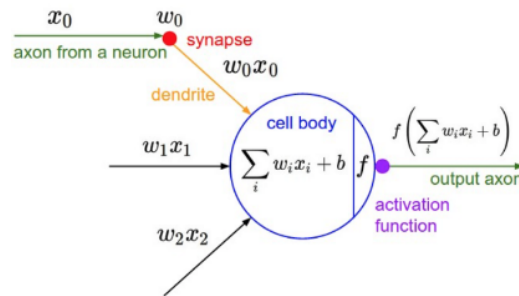


Figura B.1: Representação de um neurónio de uma rede neuronal.

Escrever aqui o detalhe adicional que melhor explique outro aspecto (diferente do que está no apêndice A) descrito no corpo principal do documento

...



# Bibliografia

- [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.
- [Baeldung, 2022] Baeldung (2022). The difference between epoch and iteration in neural networks. <https://www.baeldung.com/cs/neural-networks-epoch-vs-iteration#3-batch>.
- [Bellifemine et al., 2007] Bellifemine, F. L., Caire, G., e Greenwood, D. (2007). *Developing Multi-Agent Systems with JADE*. Wiley Series in Agent Technology. Wiley.
- [Boutilier et al., 1995] Boutilier, C., Dearden, R., e Goldszmidt, M. (1995). Exploiting structure in policy construction. In *Proceedings of the IJCAI-95*, p. 1104–1111.
- [Brownlee, 2020] Brownlee, J. (2020). Your first deep learning project in python with keras step-by-step. *Machine Learning Mastery*.
- [Brownlee, 4] Brownlee, J. (4). 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).
- [Burns, 2022] Burns, E. (2022). What is a neural network? explanation and examples. <https://www.techtarget.com/searchenterpriseai/definition/neural-network>.

- [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.
- [Elzinga e Mills, 2011] Elzinga, K. e Mills, D. (2011). The lerner index of monopoly power: Origins and uses. *American Economic Review: Papers & Proceedings*, 101(3).
- [Evia e Arnold, 2022] Evia e Arnold (visitado em Jul.2022). Nyquist sampling theorem.
- [House, 2022] House, D. (2022). Datasets. <https://www.digitalhouse.com/br/blog/dataset/>.
- [Howard e Matheson, 1984] Howard, R. e Matheson, J. (1984). Influence diagrams. In *Readings on the Principles and Applications of Decision Analysis*, volume 2, p. 721–762. Strategic Decision Group, Menlo Park, CA.
- [Janiesch et al., 2021] Janiesch, C., Zschech, P., e Heinrich, K. (2021). Machine learning and deep learning. *Electronic Markets*, 31(3):685–695.
- [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).

- [Marin, 2006] Marin, D. (2006). A formalization of RDF (applications de la logique á la sémantique du Web). Technical report, Dept. Computer Science, Ecole Polytechnique, Universidad de Chile, TR/DCC-2006-8.
- [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>.
- [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).
- [Python3.2.3, 2012] Python3.2.3 (2012). Python programming language. <http://docs.python.org/py3k/>.
- [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.

- [Wooldridge, 2000] Wooldridge, M. (2000). *Reasoning About Rational Agents*, cap.: Implementing Rational Agents. The MIT Press.