



**Politécnico  
de Viseu**

Escola Superior  
de Tecnologia  
e Gestão de Viseu

# **Modelo machine learning para padrões alimentares de doentes com Alzheimer**

Tiago Ribeiro Cardoso

## **Dissertação**

Mestrado em Engenharia Informática - Sistemas de Informação

Trabalho efetuado sob a orientação de  
Professor Doutor Carlos Augusto da Silva Cunha  
Professor Doutor Rui Pedro Monteiro Amaro Duarte

Fevereiro de 2023



**Politécnico  
de Viseu**

Escola Superior  
de Tecnologia  
e Gestão de Viseu

# **Modelo machine learning para padrões alimentares de doentes com Alzheimer**

Tiago Ribeiro Cardoso

## **Dissertação**

Mestrado em Engenharia Informática - Sistemas de Informação

Trabalho efetuado sob a orientação de

Professor Doutor Carlos Augusto da Silva Cunha  
Professor Doutor Rui Pedro Monteiro Amaro Duarte

Fevereiro 2023

## **DEDICATÓRIA**

Aos meus pais.



## **AGRADECIMENTOS**

Gostaria de agradecer à minha família e amigos, em especial aos meus pais que, sempre me ajudaram e me apoiaram, não só durante o desenvolvimento desta tese, mas também em todo o meu percurso académico.

Um forte agradecimento para o professor Doutor Carlos Augusto da Silva Cunha e para o professor Doutor Rui Pedro Monteiro Amaro Duarte que me ajudaram e aconselharam sempre.

Muito obrigado a todos.



## RESUMO

Este trabalho pretende explorar a aplicação de técnicas de *machine learning* na análise de padrões de alimentares. A área de *machine learning* pode ser aplicável a qualquer indústria e que, se corretamente aplicada, poderá trazer benefícios que até agora eram impossíveis de alcançar para a qualidade da vida humana. Por estas razões a utilização de técnicas de *machine learning* tem vindo nos últimos anos a ganhar importância no âmbito do reconhecimento de padrões para suporte á tomada de decisões.

A doença de Alzheimer leva as pessoas a perder a autonomia de exprimir as suas necessidades. Com isto os doentes de Alzheimer tornam-se dependentes de outras pessoas, mesmo para as tarefas mais básicas, nomeadamente a escolha de alimentos. Tornando assim o controlo alimentar destas pessoas uma tarefa bastante importante executada pela pessoa que apoia a sua rotina diária, denominada de Cuidador Informal (CI). Este acompanhamento é sem dúvida um processo que exige do CI uma grande responsabilidade, disciplina e habilidade para lidar com possíveis adaptações circunstanciais, como a substituição de alimentos prescritos no plano alimentar por outros equivalentes ou a combinação cuidadosa de alimentos.

A combinação de alimentos é um fator sobretudo ligado à preferência dos próprios doentes, para além das regras de combinação dos alimentos recomendados pelos especialistas da doença, existem padrões particulares a cada indivíduo que podem variar ao longo do tempo. Pretende-se assim, desenvolver um modelo de *machine learning* para detetar padrões a partir do registo prévio alimentar, podendo substituir o doente nas suas escolhas e auxiliar a pessoa que o acompanha. Pretende-se explorar várias combinações de configurações, adaptando os valores dos diferentes hiperparâmetros na execução do trabalho experimental, para comparar e obter resultados que servirão para retirar conclusões sobre a aplicação destas técnicas neste contexto.

**Palavras-chave:** Doença de Alzheimer; Cuidador Informal; Machine learning, ML



## **ABSTRACT**

This work intends to explore the application of machine learning techniques to the analysis of eating patterns. The area of machine learning can be applicable to any industry and, if correctly applied, can bring benefits that until now were impossible to achieve for the quality of human life. For these reasons, the use of machine learning techniques has gained importance in recent years in the field of pattern recognition to support decision making.

Alzheimer's disease causes people to lose the autonomy to express their needs. With this, Alzheimer's patients become dependent on other people, even for the most basic tasks, namely choosing food. Thus, making the dietary control of these people a very important task performed by the person who supports their daily routine, called Informal Caregiver (IC). This follow-up is undoubtedly a process that demands great responsibility, discipline and ability from the IC to deal with possible circumstantial adaptations, such as replacing foods prescribed in the food plan with equivalent ones or carefully combining foods.

The combination of foods is a factor mainly linked to the preference of the patients themselves, in addition to the rules for combining foods recommended by specialists in the disease, there are particular standards for each individual that can vary over time. The aim is thus to develop a machine learning model to detect patterns from the previous food record, being able to replace the patient in his choices and help the person accompanying him. Explore various combinations of configurations, adapting the values of the different hyperparameters in the execution of the experimental work, to allow drawing conclusions from each test and obtaining results that will also serve to draw conclusions about the application of these techniques in this context.

**Keywords:** Alzheimer's disease; Informal caregiver; Machine learning, MLP



# ÍNDICE GERAL

ÍNDICE DE FIGURAS .....	xiv
ÍNDICE DE GRÁFICOS .....	xvi
LISTA DE SIGLAS / ABREVIATURAS .....	xviii
<b>1. INTRODUÇÃO .....</b>	<b>1</b>
<b>1.1 Objetivos .....</b>	<b>2</b>
<b>1.2 Questões de investigação .....</b>	<b>3</b>
<b>1.3 Resultados Esperados .....</b>	<b>3</b>
<b>1.4 Plano de trabalhos .....</b>	<b>3</b>
<b>1.5 Estrutura do documento .....</b>	<b>6</b>
<b>2. REVISÃO DE LITERATURA .....</b>	<b>7</b>
<b>2.1 Alzheimer .....</b>	<b>7</b>
<b>2.1.1 Fases da doença .....</b>	<b>8</b>
<b>2.1.2 O cuidador informal .....</b>	<b>9</b>
<b>2.1.3 Nutrição e alimentação .....</b>	<b>10</b>
<b>2.2 Recomendação de alimentos .....</b>	<b>12</b>
<b>2.3 <i>Machine learning</i> .....</b>	<b>17</b>
<b>2.3.1 Tipos de <i>Machine learning</i> .....</b>	<b>18</b>
<b>2.3.2 <i>Random Forest</i> .....</b>	<b>19</b>
<b>2.3.3 <i>Preference Learning</i> .....</b>	<b>21</b>
<b>2.3.4 <i>Deep Learning</i> .....</b>	<b>22</b>
<b>2.3.5 Redes Neurais .....</b>	<b>23</b>
<b>2.3.7 Hiperparâmetros .....</b>	<b>27</b>
<b>2.3.8 Problemas Comuns .....</b>	<b>34</b>
<b>2.4 Medidas de avaliação .....</b>	<b>35</b>
<b>2.4.1 <i>Accuracy</i> .....</b>	<b>36</b>
<b>2.4.2 <i>Cross Validation</i> .....</b>	<b>37</b>
<b>3. METODOLOGIA .....</b>	<b>39</b>
<b>3.1 Seleção dos dados .....</b>	<b>39</b>
<b>3.2 Pré-processamento .....</b>	<b>39</b>
<b>3.2.1 Normalização .....</b>	<b>39</b>
<b>3.2.2 Seleção de atributos .....</b>	<b>40</b>
<b>3.3 Transformação dos dados .....</b>	<b>41</b>

3.4	Mineração dos dados .....	41
3.5	Interpretação e avaliação dos dados .....	41
4.	IMPLEMENTAÇÃO .....	43
4.1	Dados de registos alimentares .....	43
4.1.1	Reestruturação de dados .....	43
4.1.2	Engenharia de atributos .....	44
4.1.3	Seleção de registos .....	45
4.1.4	Normalização dos dados .....	47
4.1.5	Seleção de atributos.....	48
4.1.6	Divisão dos dados .....	49
4.2	Criação do modelo .....	50
4.2.1	Classificação dos alimentos por refeição .....	50
4.2.2	Processo de treino, teste e validação .....	52
5.	ANÁLISE DOS RESULTADOS.....	55
5.1	Resultados da normalização .....	55
5.2	Resultados dos ajustes do modelo.....	56
5.2.1	Função de perda ( <i>Loss Function</i> ).....	56
5.2.2	Função de ativação final .....	57
5.2.3	Optimizadores.....	58
5.2.4	Épocas.....	60
5.2.5	<i>Batch</i> .....	61
5.2.6	Camadas escondidas ( <i>Hidden layers</i> ).....	62
5.2.7	<i>Units</i> (nº de neurónios).....	64
5.3	Validação .....	65
5.4	Resultado da MLP com os hiperparâmetros escolhidos .....	66
5.5	Discussão de resultados .....	67
6.	CONCLUSÃO .....	69
6.1	Limitações e desafios .....	69
6.2	Trabalho futuro.....	70
	REFERÊNCIAS BIBLIOGRÁFICAS .....	71

## ÍNDICE DE TABELAS / QUADROS

Tabela 1.1: Plano de Escalonamento das Tarefas.....	5
Tabela 4.2: Id do alimento.....	51
Tabela 4.3: Identificação dos Ids na refeição.....	51
Tabela 4.4: Valores dos hiperparâmetros.....	53
Tabela 5.1: Hiperparâmetros utilizados nos testes da normalização.....	55
Tabela 5.2: Hiperparâmetros utilizados nos testes da função de perda.....	56
Tabela 5.3: Resultados das funções de perda.....	57
Tabela 5.4: Hiperparâmetros utilizados nos testes com a função de ativação.....	57
Tabela 5.5: Resultados das Funções de ativação.....	58
Tabela 5.6: Hiperparâmetros utilizados nos testes do otimizador.....	59
Tabela 5.7: Resultados dos otimizadores.....	59
Tabela 5.8: Hiperparâmetros utilizados nos testes com épocas.....	60
Tabela 5.9: Hiperparâmetros utilizados nos testes do Batch.....	61
Tabela 5.10: Hiperparâmetros utilizados nos testes das camadas ocultas.....	63
Tabela 5.11: Resultado de uma camada escondida.....	64
Tabela 5.12: Hiperparâmetros utilizados no número de Units.....	64
Tabela 5.13: Melhores resultados de accuracy do número de Units.....	65
Tabela 5.14: Hiperparâmetros utilizados nos testes de k-fold validation.....	65
Tabela 5.15: Accuracy com K-Fold= 5.....	66
Tabela 5.16: Hiperparâmetros escolhidos.....	67



## ÍNDICE DE FIGURAS

Figura 2.1: Estrutura de uma Random Forest.....	20
Figura 2.2: Arquitetura da MLP .....	25
Figura 2.3: Equação num neurónio.....	25
Figura 2.4 Treino supervisionado.....	26
Figura 2.5: Algoritmo de Backpropagation.....	27
Figura 2.6: Funções linear e não linear .....	28
Figura 2.7: Função ReLU .....	28
Figura 2.8: Função Sigmoid .....	29
Figura 2.9: Função Tanh.....	29
Figura 2.10: Função Softmax .....	29
Figura 2.11: Exemplo do Batch.....	34
Figura 2.12: Cenários Underfitting e Overfitting .....	35
Figura 3.1: Etapas do ML.....	39
Figura 3.2: Exemplo K-fold de 5.....	42
Figura 4.1: Tabela com os dados após pré-processamento .....	43
Figura 4.2: Transformação do atributo data .....	44
Figura 4.3: Transformação do atributo food-ids.....	44
Figura 4.4: Metodologia de seleção de registos .....	45
Figura 4.5: Número de registos por utilizador (200 utilizadores) .....	45
Figura 4.6: Contagem de alimentos de 10 utilizadores .....	46
Figura 4.7: Conjunto de dados final .....	46
Figura 4.8: Histograma do valor das calorias não normalizado .....	47
Figura 4.9: Histograma do valor das calorias normalizado .....	48
Figura 4.10: feature_importances usando Random Forest .....	49
Figura 4.11: Esquema de separação dos dados.....	50
Figura 4.12: Previsão de vários rótulos de várias classes.....	51
Figura 4.13: Processo de aprendizagem do modelo .....	52
Figura 5.1: Gráficos de perda com e sem normalização.....	55
Figura 5.2: Gráficos das funções de perda .....	56
Figura 5.3: Gráficos de perda com as funções de ativação final .....	58
Figura 5.4: Gráfico da função de perda com os optimizadores .....	59



## ÍNDICE DE GRÁFICOS

Gráfico 5.1: <i>Accuracy</i> do número de épocas .....	60
Gráfico 5.2: Número de épocas em relação ao tempo de treino .....	61
Gráfico 5.3: <i>Accuracy</i> do tamanho Batch .....	62
Gráfico 5.4: Número de épocas em relação ao tamanho Batch.....	62
Gráfico 5.5: <i>Accuracy</i> do número de camadas escondidas .....	63
Gráfico 5.6: Número de camadas escondidas em relação ao número de épocas.....	63
Gráfico 5.7: <i>Accuracy</i> do número de neurónios.....	64
Gráfico 5.8: <i>Accuracy</i> com K-Fold = 5 .....	66
Gráfico 5.9: <i>Accuracy</i> média ao longo das refeições .....	67



## **LISTA DE SIGLAS / ABREVIATURAS**

DA	Doença de Alzheimer
CI	Cuidador informal
ML	Machine learning
DL	Deep learning
RNA	Redes Neurais Artificiais
MLP	Multilayer Perceptron
LSTM	Long Short-Term Memory
RNN	Recurrent Neural Networks
CNN	Convolutional Neural Networks
RF	Random Forest



# 1. INTRODUÇÃO

A doença de Alzheimer (DA), manifesta-se por uma perda progressiva da função mental, causada pela degeneração do tecido cerebral (Salawu et al, 2011). As pessoas que sofrem desta doença estão mais sujeitadas a condições de subnutrição, sobrenutrição e desidratação por causa das doenças causadoras de demência que os leva a perder a sua autonomia de exprimir necessidades alimentares (Pivi et al, 2017). Desta forma, os doentes de Alzheimer tornam-se dependentes de outras pessoas mesmo para as tarefas mais básicas, nomeadamente a escolha de alimentos. Assim o controlo de alimentação destas pessoas tornou-se uma tarefa bastante importante que é executada pela pessoa que apoia a sua rotina diária, denominada de Cuidador Informal (CI) (Garrido e Almeida, 1999).

O CI desempenha um papel essencial na vida diária dos pacientes com DA, envolvendo-se em praticamente todos os aspetos de cuidado do doente e assumindo responsabilidades adicionais de maneira crescente (Mohide, 1993). À medida que a demência vai progredindo, o CI, além de se envolver em atividades da vida diária (administrar finanças e medicamentos), realiza também tarefas básicas, como tarefas de cuidado pessoal de higiene, banho e alimentação. Em média, os CI despedem 60 horas por semana em responsabilidades de cuidador. O CI geralmente é uma pessoa dentro do círculo familiar, e muitas vezes, a tarefa é assumida de maneira inesperada, sendo ele conduzido a uma sobrecarga física e emocional (Haley, 1997).

Este acompanhamento é sem dúvida um processo que exige do CI uma grande responsabilidade, disciplina e habilidade para lidar com possíveis adaptações circunstanciais, como a combinação cuidadosa de alimentos. O apoio á nutrição com um plano alimentar de acordo com as necessidades e preferências de cada doente, juntamente com o acompanhamento da pessoa que cuida do doente são fundamentais para o bem-estar destes doentes (Mohide,1993).

Acredita-se que os sistemas de recomendação sejam uma solução que ajudará os CI na seleção de alimentos. Nos últimos anos foram desenvolvidos diversos estudos em relação a sistemas de recomendação de alimentos para pacientes, assim como para pessoas que pretendam realizar uma dieta (Cao et al 2018). Tem sido um tema bastante abordado pela comunidade científica devido ao facto de haver cada vez mais interesse em tornar a vida das pessoas mais saudável. Um sistema de recomendação de dietas é um

sistema que monitoriza a alimentação de uma pessoa, para dietas adequadas ou ingestão de alimentos em grandes variedades de seleções prováveis a ingerir, Trang (2018). É um sistema implementado com o objetivo de ajudar a seguir dietas e ingerir alimentos considerados melhores para atender às necessidades de saúde, sabor e preferências alimentares.

Para prever as preferências alimentares, pode-se encontrar soluções na área de ML e nos seus subcampos, que são uma grande ajuda no que toca à tomada de decisões (Murphy, 2012). Os algoritmos de ML permitem treinar os modelos a partir de registos alimentares. Estes modelos têm um potencial para substituir a pessoa, efetuando previsões de qualidade em situações futuras, fazendo a escolha adequada de alimentos quando incorporados num sistema de recomendação alimentar.

O problema que se propõe resolver com a realização deste trabalho, será dar ao CI uma ferramenta, para a seleção de alimentos de acordo com as preferências de uma pessoa.

## **1.1 Objetivos**

Os objetivos propostos para este projeto são:

1. Implementar um modelo de preferências alimentares com recurso a técnicas de ML;
2. Mostrar fatores/atributos que poderão ter mais influência na escolha dos alimentos;
3. Determinar o impacto dos hiperparâmetros. Para o efeito, os modelos de aprendizagem serão testados com variadas configurações de parâmetros. Desta forma, é possível ajustar o modelo para um melhor resultado e saber quais os melhores hiperparâmetros a utilizar;

## 1.2 Questões de investigação

As perguntas a que se pretende responder com a realização deste projeto serão:

1. É possível prever os alimentos que uma pessoa pode ingerir a partir dos objetivos energéticos e de nutrientes definidos para um plano alimentar por um nutricionista?
2. Que técnicas de Pré-Processamento são necessárias?
3. Que algoritmos podem ser adequados a utilizar neste tipo de problemas?

## 1.3 Resultados Esperados

No fim da realização do projeto, é esperado obter os seguintes resultados:

1. Ter uma seleção dos atributos presentes nos dados com maior capacidade de previsão, escolhendo assim aqueles que irão obter melhores resultados.
2. De acordo com as metas nutricionais de um determinado utilizador conseguir recomendar alimentos.
3. Obter melhor precisão nas previsões fazendo uma escolha mais adequada dos hiperparâmetros do modelo.

## 1.4 Plano de trabalhos

O trabalho elaborado segue o seguinte planeamento:

T1. Estudo do tema da doença de Alzheimer e dos seus efeitos na vida do doente e do seu cuidador. Compreender melhor os problemas que este tipo de doença traz tanto como para o paciente como para as pessoas que cuidam do mesmo.

T2. Estudo de métodos de *machine learning* que poderão ser aplicados neste contexto, sobretudo nas áreas de nutrição.

T3. Pesquisa de trabalhos e aplicações que poderão servir de apoio para este estudo/projeto. Para assim ter como suporte para a realização deste trabalho.

T4. Fazer a recolha de dados e uma preparação dos mesmos, o conjunto de dados usado para o processo de ML são compostos por variáveis, que por sua vez são

definidos como características ou atributos. Tanto a qualidade como a quantidade das informações dos dados utilizados para o ML são relevantes para uma melhor performance e acerto de previsões.

T5. Desenvolver o modelo utilizando técnicas de ML, estudando os exemplos anteriores, pode-se encontrar alguma semelhança no que diz respeito às preferências alimentares de uma pessoa, e pode-se enquadrar esse tipo de soluções, por exemplo, pode-se obter as preferências de escolha de alimentos de um doente baseando-se em registos anteriores e assim pode-se ter uma perspectiva de quais alimentos agradaram mais ao doente sem que este se precise de manifestar. Assim sendo pretende-se que o modelo desenvolvido faça esse tipo de previsões acertadamente.

T6. Treinar e testar o modelo com seleção de atributos e diferentes hiperparâmetros, fornecendo dados ao modelo para que este possa aprender com eles, é a ideia do ML, para que assim aumente o seu conhecimento de modo a melhorar o seu desempenho. Após este treino, o modelo irá ter um maior potencial para efetuar previsões de qualidade em situações futuras, que será fazer a escolha adequada de alimentos.

T7. Avaliação de resultados obtidos, avaliar o desempenho do modelo também será uma tarefa importante porque pode ajudar a fazer alterações para que este tenha melhor desempenho.

T8. Escrita da dissertação relacionada com a aprendizagem e resultados obtidos com a realização deste projeto.

## CALENDARIZAÇÃO DAS TAREFAS

As tarefas descritas serão executadas de acordo com a calendarização constante na Tabela 1.1.

Tarefa	Meses												
	2022											2023	
	fev	mar	abril	maio	jun	julho	ago	set	out	nov	dez	jan	fev
T1													
T2													
T3													
T4													
T5													
T6													
T7													
T8													
<b>Metas</b>		INI			M1 M2	M3 M4		M5			M6	M7 M8	FIM

Tabela 1.1: Plano de Escalonamento das Tarefas

Legenda:

- INI Início do trabalho;
- M1 Fim da tarefa 1.
- M2 Fim da tarefa 2.
- M3 Fim da tarefa 3.
- M4 Fim da tarefa 4.
- M5 Fim da tarefa 5.
- M6 Fim da tarefa 6.
- M7 Fim da tarefa 7.
- M8 Fim da tarefa 8.
- FIM Fim do trabalho.

## 1.5 Estrutura do documento

Este documento será a descrição do projeto que se desenvolveu. Consiste na criação de um modelo de ML para prever hábitos alimentares e principais desafios a ultrapassar, permitindo assim demonstrar resultados positivos na aplicação de algoritmos *machine learning* no apoio aos cuidadores informais de doentes com Alzheimer.

Este documento está dividido em 6 capítulos:

No capítulo 1, é apresentado a motivação, a contextualização, a definição do problema, os objetivos, os resultados esperados, o plano de trabalhos e a estrutura do documento.

No capítulo 2 é apresentado o estado da arte na área do ML aplicado a padrões alimentares e consequentemente uma descrição da DA e dos diversos cuidados que um cuidador informal terá de ter no apoio a este tipo de doentes. Numa primeira fase o estudo da doença e dos seus sintomas que levam a que estes doentes tenham a necessidade de um acompanhamento especial por parte de quem está a cuidar deles, e por outro lado os cuidados e o seu papel que estes irão ter na vida destes doentes. Numa segunda fase, é feito o estudo de possíveis técnicas de ML que poderão ser uma solução para ajudar o aumento de qualidade de vida dos doentes com Alzheimer, nomeadamente na sua alimentação e os seus benefícios para o doente.

No capítulo 3 é apresentada a metodologia utilizada e a sua descrição detalhada ao longo do projeto. Uma parte com o processo relativo aos dados e outra parte com o processo relativo ao desenvolvimento do modelo. No capítulo 4 a implementação e o 5 os resultados dos testes obtidos. No capítulo 6 as conclusões do projeto.

## **2. REVISÃO DE LITERATURA**

Neste capítulo é apresentado os fundamentos teóricos relacionados com a DA, os trabalhos relacionados com este projeto, e as técnicas de ML que possam ser soluções ao problema.

### **2.1 Alzheimer**

Cada vez mais a expectativa de vida humana aumenta e as pessoas estão a viver durante mais tempo e com mais qualidade do que nunca. Embora este aumento permita que as pessoas tenham uma vida mais feliz e plena, por outro lado também aproxima as células humanas do seu limite biológico, causando o surgimento de novos tipos de complicações e doenças relacionadas ao envelhecimento (Brandy et al, 2005). Um exemplo de tais complicações é a demência, uma condição que afeta a população idosa cada vez mais a cada ano. A demência é classificada como um transtorno mental orgânico, e é descrita como "um comprometimento global adquirido do intelecto, memória e personalidade, mas sem comprometimento da consciência" (Yeo & Thompson, 2014).

A DA foi descrita pela primeira vez por Alois Alzheimer há mais de cem anos (Cipriani et al, 2011). De acordo com Salawu et al (2011) é uma doença neurodegenerativa progressiva que apresenta duas lesões cardinais características: placas senis e emaranhados neurofibrilares. Estas alterações neuropatológicas podem ser provocadas por alterações genéticas e ambientais. Clinicamente, a DA é caracterizada por perda progressiva de memória e orientação e outros défices cognitivos que incluem afetação do julgamento e da tomada de decisões, dificuldades motoras e perturbações da linguagem. É a demência mais comum e afeta cerca de 75% das mais de 35 milhões de pessoas no mundo com demência (Yiannopoulou e Papageorgiou, 2013). O número de pessoas afetadas por esta demência, em todo o mundo, tem aumentado de forma abismal e estima-se que até 2050 o número possa atingir os 150 milhões (Petersson e Philippou, 2016). Estes dados exigiram uma elevada atenção por parte dos países, muito devido ao peso financeiro que a doença tem nos indivíduos, famílias e serviços de saúde pública. Observou-se uma necessidade de desenvolver estratégias de modo a garantir que pessoas com demência, assim como as suas famílias e cuidadores, conseguissem ter acesso a materiais e equipamentos que permitam, de certo modo, ajudar na realização de tarefas diárias que se tornam cada vez mais difíceis, à medida que a doença progride

(Schmachtenberg et al, 2020). A doença tem um início silencioso, levando lentamente à perda progressiva da memória e funções cognitivas. Também são visíveis alterações de humor, comportamento, linguagem e atenção. No entanto, o sintoma inicial mais comum, é a dificuldade em recordar informação recebida a curto prazo (Singh et al, 2013).

Um fator distinto e característico da DA, é a acumulação de placas  $\beta$ -amilóide (proteína tóxica para os neurónios) e a forma hiperfosforilada da proteína Tau (proteínas que estabilizam os microtúbulos). A presença desta condição leva a que ocorra uma progressiva perda sináptica, morte neuronal e toxicidade vascular. Estão ainda associadas outras alterações como: disfunção mitocondrial, inflamação, acumulação anormal de metais de transição e stress oxidativo. O cérebro sendo altamente suscetível a estes danos, leva a que haja um aumento da produção e deposição das placas  $\beta$ -amilóide no cérebro e, conseqüente hiperfosforilação da proteína Tau, impulsionando o ciclo de desenvolvimento neuropatológico da doença (Fernández et al, 2017). À medida que ocorre a destruição crónica e progressiva das células nervosas, o cérebro deixa de conseguir compensar as alterações sofridas e as placas  $\beta$ -amilóide e proteína Tau acabam por se expandir para outras zonas do cérebro. Os danos são de tal modo significativos que os doentes começam a apresentar alterações comportamentais como: depressão, variações de humor e total perda de interesse nas atividades do quotidiano (Trevisan et al, 2019).

### **2.1.1 Fases da doença**

Numa fase inicial da DA, geralmente, o primeiro sintoma manifesta-se através da memória. O doente tem memória de acontecimentos antigos e consegue relatá-los, no entanto, tem dificuldade em recordar novos acontecimentos. Inicialmente terá dificuldades em lembrar as datas, os locais e as pessoas presentes em determinados acontecimentos. Começam por ficar desorientados, sendo frequente perderem-se na rua e até mesmo dentro de casa. Também é normal perderem a noção do tempo, confundindo o dia, o mês e até o ano em que estão (Silva, 2018). Ao longo do tempo, a forma como estes utentes comunicavam vai ficando afetada. Primeiro manifestam dificuldades em nomear o nome de objetos divagando de forma confusa. O discurso torna-se confuso e perde-se com facilidade a falar. Rapidamente começa a perder capacidade de enfrentar e resolver problemas simples do dia-a-dia. Também a personalidade do doente começa a mudar, age como nunca agiu anteriormente na sua vida, faz comentários inadequados com linguagem imprópria, sem perceber que atinge outras pessoas, nomeadamente os familiares próximos. A vida social dos doentes de Alzheimer pouco a pouco vai se

tornando muito confusa, deixando de cumprir as suas obrigações sociais e profissionais no seu quotidiano. O doente deixa de se preocupar com a sua higiene, a sua forma de se vestir e de comer. As perturbações de humor estão quase sempre ligadas à DA. Os doentes manifestam-se ansiosos e inseguros por não conseguirem realizar tarefas do quotidiano como antes faziam e, na maior parte dos casos, tentam negar perante familiares e amigos que estão a perder capacidades. Geralmente, os doentes ficam pessimistas, podendo até ganhar uma depressão grave, e pensamentos negativos (Silva, 2018).

Numa fase mais avançada da doença, a capacidade motora do doente fica bastante afetada, revelando muita dificuldade em andar, abrir portas ou vestir-se, acabando mais tarde por deixar de fazer estas atividades. Começa a perder a capacidade de associar os objetos às suas funções, assim como a capacidade de reconhecer pessoas, lugares, sons e cheiros. Aos poucos perde a função da fala e deixa de comunicar verbalmente. Começam a alucinar, vendo objetos que não existem e intrusos dentro da sua própria habitação. Por consequência, os doentes tornam-se agitados e agressivos, podendo usar violência. É frequente o doente deambular pela casa, sem razão aparente ou sair para o exterior da habitação podendo correr perigo. Para o doente, comer ou tomar banho é agora necessário controlá-lo de forma constante (Correia et al, 2015). É importante referir que estas alterações podem ser agravadas por medicação com excessiva sedação. Desta forma, passa a ter uma vida dependente de uma terceira pessoa, carecendo de cuidados permanentes, alimentação, higiene pessoal, cuidados de saúde (Silva, 2018).

Numa fase terminal, torna-se muito complicado alimentar estes doentes, sendo que por vezes é necessário recorrer à sonda. Também a incontinência urinária se instalará nesta fase. A atrofia muscular acontece na fase terminal em que o doente perde os reflexos, perde a postura e termina limitado à cadeira e à cama. Depois destas limitações os problemas de saúde agravam-se, nomeadamente, obstipação tenaz, infeções respiratórias ou urinárias, úlceras de decúbito. No final, deixa de reagir a qualquer estímulo e a vida dele torna-se vegetativa (Silva, 2018).

### **2.1.2 O cuidador informal**

À medida que a DA progride, surge a dependência por cuidados especiais, função importante desempenhada pelo cuidador informal (CI). O CI é quem dá suporte físico e psicológico, fornecendo ajuda prática, se necessário. Define-se cuidador como a pessoa

diretamente responsável pelo CI do paciente, normalmente a esposa, um dos filhos ou outro parente, ou, ainda, uma pessoa contratada para a função (Grafstrom et al, 1992). O CI é definido como o principal responsável por prover ou coordenar os recursos requeridos pelo paciente. Na década de 60 iniciaram-se estudos sobre CI de pacientes psiquiátricos; posteriormente, CI de idosos frágeis; finalmente, na década de 80 iniciaram-se estudos sobre CI de pessoas com demência, em especial o CI de pacientes com DA (Mohide, 1993). Os estudos dos CI de idosos com DA indicam que a idade média do CI varia entre 50 e 65 anos (Burns, 2003).

Aproximadamente 80% dos cuidados com os pacientes com DA são providos por membros da família, que não têm qualquer formação ou conhecimento da doença (Garrido e Almeida, 1999).

Cuidar de um portador de DA não é um trabalho fácil, pois requer tempo, energia e paciência. Os cuidadores são expostos a sobrecargas que podem provocar problemas sociais, físicos e psicológicos (Paula, et al 2008). O comprometimento da qualidade de vida dos cuidadores é bastante comum e pode ser evidenciado pela presença de depressão e exaustão devido ao progressivo desgaste. A tarefa exaustiva de cuidar é geralmente repetitiva, contínua e, muitas vezes, uma atividade solitária e sem descanso, que o pode levar a isolamento afetivo e social (Paula, et al 2008). Os cuidadores ainda têm probabilidades de apresentar ansiedade, que é caracterizada pela preocupação, irritabilidade, fadiga, insónia, entre outros sintomas, devido à grande responsabilidade e paciência que a tarefa de cuidar pode trazer (Paula, et al 2008).

### **2.1.3 Nutrição e alimentação**

Um bom estado nutricional ao longo do ciclo de vida, tem um importante papel na prevenção e progressão da DA. Alimentar uma pessoa com demência pode levar bastantes minutos para que se garanta que seja uma experiência calma e agradável. A qualidade da interação do CI que dá assistência na alimentação da pessoa com demência está positivamente associada com a quantidade de comida consumida (Correia et al, 2015). A alimentação pode ser uma tarefa difícil para indivíduos com Alzheimer devido às suas alterações fisiológicas e também psicológicas. Com avançar da doença e da idade ocorre perda de autonomia com impactos negativos na ingestão alimentar e estado nutricional. Por isso, nas refeições de pessoas com DA deve-se ter em conta não só as suas necessidades nutricionais, mas também tornar o prato sensorialmente apelativo (cores,

consistência, aromas...) ou seja deve-se ter em conta a capacidade para estimular o desejo de consumir alimentos (Pouyet et al, 2014).

Uma vez que a doença já se encontra instalada no doente, a suplementação (fontes concentradas de nutrientes em suplementos), não chega para controlar a deterioração do estado mental e cognitivo dos doentes. Assim sendo é importante manter uma dieta equilibrada com alimentos frescos e com propriedades benéficas para a saúde das pessoas (Ferreira, 2019).

Tendo em conta que os défices nutricionais são muito comuns nos doentes com Alzheimer, por causa da falta de capacidade de expressão e de autonomia, é muito importante assegurar a ingestão adequada destes nutrientes pela alimentação, de modo a garantir a melhor qualidade de vida a estas pessoas. Para tal, é importante que haja elevada comunicação entre o CI e outros profissionais de saúde (nutricionistas, psicólogos, terapeutas físicos e da fala), que podem proporcionar as orientações adequadas, de modo a prestar os melhores cuidados possíveis. Educar e treinar o CI numa fase inicial e durante o decorrer da doença, é muito importante para incentivar a adoção de estratégias relativas ao relacionamento social e nutricional (Pivi et al, 2017). Assim, treinar adequadamente o CI em relação à alimentação é vital, uma vez que quanto melhor a interação com o cuidador, melhores os resultados de nutrição relativamente à alimentação por parte dos doentes. Portanto, as medidas de orientação para os CI, são essenciais, uma vez que a perda de apetite e dificuldade na alimentação são inevitáveis no decorrer da doença (Pivi et al, 2017).

Um dos aspetos a ter em conta na DA é a perda de peso ou aumento de peso. Tal deve-se a distúrbios cognitivos e comportamentais, como agitação; medicação para o tratamento da DA, que aumenta o risco de efeitos adversos ao nível do trato gastrointestinal (náuseas, vômitos) e que por sua vez podem dificultar a capacidade de alimentação; degradação fisiológica do cérebro no decurso da doença que afeta a região do cérebro responsável pelos comportamentos alimentares; ou o elevado catabolismo que ocorre nestes doentes (Pivi et al, 2017).

De modo a garantir um bom estado nutricional, é necessário garantir a adequada ingestão de calorias e proteínas diárias (através de suplementos orais), juntamente com uma dieta adequada e recomendada, que garantem a manutenção das necessidades nutricionais específicas. De modo a garantir a melhor qualidade de vida das pessoas a

atravessar todo o processo da doença existem diferentes estratégias alimentares a adotar de acordo com cada caso específico. A ligação entre o doente e o CI é muito importante, mas acima de tudo é importante garantir a saúde e qualidade de vida de ambos (Ferreira, 2019).

Quando o paciente se encontra num estado nutricional não adequado, como é frequente em doentes com Alzheimer, é encaminhado a ter um acompanhamento nutricional com um nutricionista. As tecnologias de informação estão a ser cada vez mais implementadas nas organizações de saúde para responder às necessidades dos profissionais de saúde e também das pessoas que cuidam dos doentes (CI), nas suas atividades diárias de tomada de decisão. Por exemplo, quando um paciente está desnutrido e precisa de monitoramento, as decisões precisam ser tomadas corretamente (portela et al 2015).

Prevenir o risco de desnutrição em pacientes a partir de variáveis preditivas pode ser o primeiro passo para um adequado controle nutricional. Dada a rastreabilidade e monitoramento do estado nutricional de cada paciente, utilizar essa informação para tratar e melhorar o prognóstico de cada um. Com isso, o tempo de internação e os custos hospitalares serão reduzidos, melhorando a qualidade de vida dos pacientes e do seu CI. Diante dessa realidade, o nutricionista desempenha um papel crucial, pois recomenda uma dieta ao paciente, e conseqüentemente, preveni e controla a desnutrição (Reis et al, 2017).

## **2.2 Recomendação de alimentos**

Nos últimos anos foram desenvolvidos diversos estudos em relação a sistemas de recomendação de alimentos. Tem sido um tema bastante abordado pela comunidade científica, com o objetivo de solucionar diversos problemas relacionados com a nutrição. Em seguida irão ser apresentados diversos trabalhos relacionados com sistemas de recomendação de alimentos.

Um sistema de recomendação de dieta personalizado para a nutrição, foi desenvolvido usando o algoritmo de *artificial bee colony* (Karaboga,D 2010). Os autores Raut et al (2018) propuseram um sistema que usa raciocínio baseado em regras para fazer recomendações alimentares e nutricionais de forma eficiente enquanto um algoritmo genético para a geração do menu de uma refeição. O sistema dependia da API do Google FIT do utilizador para obter informações sobre as atividades diárias e a necessidade de

energia. O sistema também se baseia no histórico de doenças anteriores do paciente para fazer recomendações personalizadas de dieta para utilizadores. Com base nos resultados obtidos, os autores concluíram que os algoritmos de DL obtiveram um melhor desempenho em relação aos algoritmos de ML.

Leipold et al (2018) desenvolveram um sistema de assistência nutricional que fornece feedback sobre o comportamento alimentar do paciente e acomoda a mudança de comportamento por meio de diversos elementos persuasivos, como auto monitorização, personalização e implementação de reflexão, recomendações ou rastreamento. Considerando que um sistema automatizado de recomendação de alimentos / dieta pode fornecer grandes benefícios quando comparado a nutricionistas humanos, ele também enfrenta uma série de limitações que variam de usabilidade, eficiência, eficácia e satisfação. O estudo mostrou a necessidade de melhorias em vários aspetos da aplicação, como as recomendações, desempenho e facilidade de rastreamento de ingestão. Ainda assim, 43% dos participantes usariam a aplicação regularmente e a maioria (85%) prefere o rastreamento alimentar diário a um questionário semanal alimentar. Os autores concluíram que, apesar dos resultados, é necessário integrar as informações contextuais e sociais, bem como aumentar a precisão dos dados de entrada recebidos.

Agapito et al (2018) propuseram um sistema organizador de dieta. Eles criaram um perfil usando uma pesquisa dinâmica em tempo real que os médicos prepararam e que foi compilada pelos utilizadores. O sistema denominado DIETOS é capaz de recomendar não apenas alimentos específicos da mesma categoria e com graus de saúde semelhantes, mas também pode fornecer sugestões nutricionais para algumas categorias de problemas de saúde. Todas as informações armazenadas na base de dados são acessíveis ao utilizador por meio de uma interface na web. O acesso aos serviços oferecidos pelo sistema de recomendação foi pensado para ser o mais intuitivo e direto possível. Em particular, os alimentos típicos compatíveis com o perfil de saúde atual do utilizador são representados em verde na interface do sistema, enquanto os alimentos incompatíveis com seu perfil de saúde são representados em vermelho. O conhecimento dos locais de produção dos alimentos típicos ocorre por meio da interação gráfica. Para tal, foi desenvolvido um sistema que permite recuperar informações sobre alimentos e localizações navegando graficamente no mapa. Os resultados (as comidas típicas produzidas no local escolhido), foram exibidos num mapa. Além disso, foi possível conhecer as propriedades nutricionais, benefícios e efeitos colaterais de cada produto pesquisado na base de dados.

Noutro estudo, realizado por Weber e Achananuparp (2016), para apoiar pessoas que estão a tentar perder peso e se manterem saudáveis, usaram dados diários alimentares públicos de mais de 4.000 utilizadores ativos a longo prazo da aplicação do MyFitnessPal. Rebedew (2015), para estudar as características de uma dieta mal sucedida. Concretamente, treinaram um modelo de ML para prever repetidamente o excesso ou a falta de metas de calorias diárias autodefinidas e, em seguida, observaram quais os recursos que contribuem à previsão do modelo. A análise foi centrada num classificador que, dada uma lista de alimentos consumidos por uma pessoa, prevê se eles ficarão abaixo ou acima de sua meta de calorias autodefinida. Os resultados comprovaram a viabilidade de mineração de tais dados para análises relacionadas à saúde. Especialmente com *links* adicionais para a atividade dos utilizadores e padrões gerais de estilo de vida, dietas personalizadas e adaptativas geradas automaticamente parecem um caminho promissor a seguir.

Noutro projeto os autores Iwendi et al (2020), propuseram-se a desenvolver uma solução de DL para um conjunto de dados de saúde, que deteta automaticamente qual alimento deve ser dado a qual paciente com base na doença e outras características como idade, sexo, peso, calorias, proteína, gordura, sódio, fibra, colesterol. A pesquisa foi focada na implementação de algoritmos de DL e de ML, como regressão logística, Naive Bayes, Rede Neural Recorrente (RNN), Multilayer Perceptron (MLP), Unidades Recorrentes Portadas, e Long Short-Term Memory (LSTM). E foram aplicadas num conjunto de dados médicos recolhidos pela internet e hospitais que consistiram em dados de 30 pacientes com 13 características de diferentes doenças e 1000 produtos. O desempenho de várias técnicas de DL e de ML foi testado e treinado e o resultado provou que estas técnicas tiveram eficácia de previsão neste tipo de solução. O modelo LSTM superou outros modelos em termos de *precision*, *recall* e *F1 measure*. Para a classe permitida, o classificador LSTM teve 98% de *precision*, 99% de *recall* e *F1 measure*, respetivamente. Para a classe não permitida, teve 89% de *precision*, 73% de *recall* e 80 % *F1 measure*.

Rostami et al (2022) visaram desenvolver um novo sistema híbrido de recomendação de alimentos. O método proposto envolve duas fases: recomendação baseada no conteúdo alimentar e recomendação baseada no utilizador. O agrupamento de gráficos é usado na primeira fase, e uma abordagem baseada em DL é usada na segunda fase para agrupar utilizadores e alimentos. Os resultados indicaram que o sistema híbrido

de recomendação de alimentos proposto é, em média, cerca de 11,8% mais eficiente do que o caso em que apenas a fase baseada no utilizador é utilizada. E cerca de 11,9% mais eficiente do que o caso em que apenas a fase baseada em alimentos é empregue para recomendação final. Além disso, os resultados relatados mostraram que quando a rede de confiança explícita do utilizador é usada no sistema de recomendação desenvolvido, o desempenho melhora em média 5,3% em comparação com o caso em que a informação de confiança é ignorada. Os testes realizados também mostraram que quando a medida de avaliação leva em consideração o fator tempo, o desempenho final do sistema desenvolvido é em média 9,51% melhor do que quando se ignora esse fator. As médias *Precision*, *Recall*, *F1*, *AUC* e *NDCG* do modelo proposto foram 0,0721, 0,0691, 0,0705, 0,6812 e 0,0497, respetivamente, o que se mostrou significativamente maior do que nos modelos de comparação.

Zhang et al (2022) desenvolveram uma nova abordagem de recomendação baseada na otimização de vários objetivos (*many-objective optimization*) (MaOO), foi desenvolvida para fornecer uma maneira equilibrada e sistemática de lidar com as tarefas de recomendação de alimentos. Foram considerados simultaneamente no método de recomendação as preferências do utilizador, padrão de dieta do utilizador, valores nutricionais dos alimentos e diversidade alimentar. Em seguida, três algoritmos de otimização de vários objetivos, foram aplicados para resolver a tarefa de recomendação apresentada, baseados em conjuntos de dados reais foram conduzidos para verificar a eficácia da estrutura de recomendação proposta. Foi realizada uma comparação de três algoritmos MaOO de acordo com as métricas de desempenho. O indicador de hipervolume funcionou como um quantificador onde valores maiores indicam melhores resultados. Entre os três algoritmos MaOO, o SPEA2 + SDE forneceu o melhor desempenho.

Banerjee et al (2019) utilizaram uma abordagem de previsão usando algoritmos de ML para identificar pacientes com doença renal crónica. Foram usados os algoritmos de *Random Forest*, *SVM* e *Naive Bayes*. Com base nos resultados obtidos, é possível recomendar alimentos para diferentes níveis de pacientes com doença renal crónica usando o nível de potássio no sangue, o que ajudará os pacientes a manter seu nível de sal. O algoritmo *Naive Bayes* fornece uma precisão de 95,5%, o *Support Vector Machine* (SVM) fornece uma precisão de 98,25% e o algoritmo *Random Forest* fornece uma precisão de 99,75%. O algoritmo *Random Forest* obteve o melhor desempenho.

Alian et al (2018) desenvolveram um sistema de recomendação de autocuidado com diabetes especificamente para pacientes na comunidade de índios americanos. Recomenda um estilo de vida saudável aos utilizadores com diabetes. Ao integrar o perfil ontológico dos utilizadores com recomendações e diretrizes clínicas gerais de diabetes, o sistema pode fazer recomendações personalizadas (por exemplo, ingestão de alimentos e exercícios físicos) com base nas características socioeconómicas, culturais, e situação geográfica. O sistema proposto foi implementado como aplicação móvel. Avaliações realizadas por casos de estudos de uso e verificação de especialistas humanos demonstram a eficácia do sistema. Foi medido o desempenho da recomendação em termos de precisão, relevância, que mede se o sistema fala ou não a linguagem dos utilizadores, com palavras, frases e conceitos familiares aos utilizadores. E na adequação, que avalia se a recomendação produziria ou não um resultado positivo/encorajador. Com 100% de precisão, os especialistas médicos confirmaram a exatidão das recomendações.

Phanich et al (2010), propuseram um Sistema de Recomendação Alimentar, por meio da análise de agrupamento de alimentos para pacientes diabéticos. O sistema recomendará os alimentos substitutos adequados no contexto da nutrição e das características dos alimentos. Foi usado o *Self-Organizing Map* e o *K-mean clustering* para análise de agrupamentos de alimentos que é baseado na similaridade de oito nutrientes significativos para pacientes diabéticos. No final, o sistema foi avaliado por nutricionistas e teve um desempenho muito bom e útil para a área de nutrição. Este sistema foi avaliado por nutricionistas convidados para o preenchimento de um questionário. A primeira parte do questionário está relacionada ao formato de apresentação e a outra parte está envolvida com a aceitação do resultado do sistema. Os nutricionistas preencheram os questionários atribuindo um valor inteiro entre 1 e 5 para cada questão, '5' significa ser o mais eficaz, enquanto '1' sugere que o sistema precisa ter uma alteração. Os resultados dos questionários mostraram que os utilizadores estão bastante satisfeitos com o uso do sistema de recomendação. A pontuação total do formato de apresentação e da aceitação são 3,58 e 3,64, respetivamente. Para a pontuação do formato de apresentação do sistema e a clareza dos complementos são 4,3 definidos como um sistema eficaz. O sistema obtém uma pontuação boa de aproximadamente 3,3 na amenidade do formato do resultado. No entanto, os utilizadores precisam ter algum conhecimento nutricional antes de usar o sistema. Portanto, não obteve uma pontuação alta na usabilidade. Para a aceitação do resultado, a rapidez na execução ganha a nota

máxima de 4,3. Isso mostra que pode responder rapidamente aos utilizadores. A precisão do sistema com pontuação 4 também é boa para os utilizadores. A complacência, a perspicácia e a conveniência no uso do sistema recebem pontuação 3,3.

Naresh et al (2020), desenvolveram um sistema de recomendação de alimentos, que é construído de forma inteligente e é útil para os utilizadores escolherem entre a enorme variedade de alimentos. O Sistema de Recomendação Alimentar, proposto neste artigo, tornou-se mais inteligente e avançado com os benefícios dos algoritmos personalizados A\* e BFS personalizados. Este modelo pode entender e prever o gosto dos utilizadores. Modelo de recomendação alimentar, foi integrado com sucesso num chat chamado “FOODIE”, que é usado para pedir comida. À medida que o utilizador pede comida, o chat armazena os dados treina o modelo de ML com esses dados para o próximo pedido de comida e recomenda comida com base nesse resultado. A classificação de utilizadores foi de 92% positiva, o que significa um agrado com a comida recomendada.

### ***2.3 Machine learning***

À medida que as organizações de saúde geram e armazenam grandes volumes de dados, as decisões clínicas podem ser tomadas não apenas com base na intuição e experiência do médico, mas também com base no conhecimento oculto armazenado ao longo do tempo em bases de dados de saúde (Albirami et al, 2013).

O ML é o estudo de algoritmos de computador que melhoram automaticamente com a experiência e podem ser usado para desenvolver sistemas que resultam em maior eficiência e eficácia (Eapen, 2004). O ML fornece a base técnica da mineração de dados. É usado para extrair informações dos dados brutos em bases de dados. A mineração de dados é a aplicação de algoritmos específicos na extração de padrões de dados (Fyyad, 1996). Trata-se de uma área com um potencial de evolução enorme para os próximos anos, tendo uma grande capacidade de processamento e suporte à tomada de decisão, sendo um conceito que pode ser aplicável a qualquer indústria e que, se corretamente aplicada, poderá trazer benefícios que até agora eram impossíveis de alcançar para a qualidade da vida humana (Albirami et al, 2013). E por estas razões a utilização de técnicas de ML tem vindo nos últimos anos a ganhar importância no âmbito do reconhecimento de padrões para suporte á tomada de decisões. Intuitivamente, a

utilização de ML para suportar a escolha de alimentos para as refeições de um doente é uma potencial solução a explorar.

Segundo Murphy (2012), ML é definido por uma modificação ou adaptação de ações de um computador com o objetivo de acertar decisões ou previsões, utilizando um conjunto de dados como forma de experiências para o seu processo de aprendizagem. O conjunto de dados usado para o processo de ML são compostos por variáveis, que por sua vez são definidos como características ou atributos, tal como refere Furnkranz e Hullermeier (2010).

### **2.3.1 Tipos de *Machine learning***

Segundo Lison (2015), o ML pode ser dividida em três tipos: aprendizagem supervisionada, aprendizagem não supervisionada, e aprendizagem por reforço, em que, cada um destes tipos tem propósitos diferentes como irá ser explicado a seguir.

Nos dias de hoje, existem diversos algoritmos de ML que podem ser utilizados de diversas maneiras, de acordo com o seu propósito, são categorizados pelos tipos de aprendizagem referidos no parágrafo anterior (Portugal, Alencar e Cowan, 2018). Alguns exemplos de algoritmos são: Árvores de decisão, *NearestNeighbor*, Redes Neurais e Regressão Linear (Shalev-Shwartz e Ben-David, 2014).

A aprendizagem é supervisionada é quando é fornecido um conjunto de exemplos de dados com as respostas corretas, incluindo dados de entrada e saída. Com base nesse conjunto de dados, o algoritmo utilizado, posteriormente, irá tentar responder a todas as entradas possíveis. (Marsland, 2014). Neste tipo, existem duas subcategorias: regressão e classificação. A regressão é utilizada para prever valores contínuos com base nos dados de entrada e saída. Um exemplo onde se pode utilizar este tipo de aprendizagem será na previsão da quantidade específica de nutrientes que uma pessoa deve ingerir (Chin et al, 2019). A classificação é utilizada quando se pretende retornar um valor categórico (e. g. Sim ou Não, verdadeiro ou falso). Um exemplo de onde se pode utilizar este tipo de categoria é na descoberta se uma pessoa está a atingir as metas para uma dieta ou não (Zellerbach & Ruiz, 2019).

A aprendizagem não supervisionada é utilizada quando há situações onde algumas informações não são implícitas, para encontrar padrões neste conjunto de dados (Sathya e Abraham, 2013). O algoritmo mais conhecido e utilizado deste tipo de aprendizagem,

que é também conhecida por *clustering*, é o *K-means*. É um algoritmo com grande simplicidade e eficiência na sua utilização, este algoritmo tem sido usado em diversas situações, como reconhecimento de padrões, mineração de dados e na área da bioinformática (Peng, Wang, Pérez-Jiménez e Riscos-Núñez, 2015). Este tipo de aprendizagem tem sido cada vez mais utilizado nestes últimos anos. Um exemplo onde este pode ser aplicado é na deteção de desnutrição de crianças. Em Nagari & Inayati (2020), foi implementado um cluster usando o algoritmo de *K-means* para determinar os valores nutricionais de crianças na Indonésia e assim poder agrupar um grupo de crianças com maior necessidade de apoio. Os resultados deste estudo indicam que o estado nutricional dessas crianças pode ser agrupado usando o método *K-Means* através do peso de acordo com a idade em 4 *clusters*. O *cluster* 1 com 23 crianças em situação de desnutrição, *cluster* 2 com 17 crianças em mau estado nutricional, *cluster* 3 com 7 crianças em mau estado nutricional e *cluster* 4 com 10 crianças.

Para além dos outros dois tipos, existe também a aprendizagem por reforço. Esta tem, como principal objetivo, aprender consoante o meio que a envolve e não através de conjuntos de dados inicialmente inseridos. Um exemplo onde se usou a aprendizagem por reforço, foi quando um agente que é colocado em campo, efetua uma ação com base no ambiente em que está. Essa ação é analisada e, quanto mais acertada for a escolha, melhor recompensa o agente recebe (Sutton e Barto, 2018). Um dos principais problemas deste tipo de aprendizagem é que nem sempre se tem um conjunto de dados com todos os dados de entrada e saída devidamente identificados, como acontece com a aprendizagem supervisionada, para se prever com maior precisão os resultados pretendidos, o que torna este tipo de aprendizagem mais desafiante que os dois tipos anteriormente apresentados (Sutton e Barto, 2018).

### **2.3.2 Random Forest**

O *Random Forest* (RF) é um algoritmo de ML que combina um conjunto de árvores de decisão, onde cada árvore é treinada utilizando um subconjunto de amostras selecionadas aleatoriamente e com repetição. A quantidade total de amostras utilizadas no subconjunto é o total de amostras do conjunto original de treino (Breiman, 2001).

A Árvore de Decisão também é um algoritmo ML, que tem como finalidade criar um modelo num formato de árvore (RF). Cada nó dessa árvore contém uma questão, e as possíveis respostas para essa questão são os ramos que apontam para os nós. Ao

classificar um dado, é percorrido o caminho em direção da raiz até algum nó sem “folha”, respondendo a perguntas de acordo com as características do dado. A folha determina a qual classe pertence um dado (Kingsford & Salzberg, 2008). A Figura 2.1 ilustra a representação de uma RF.

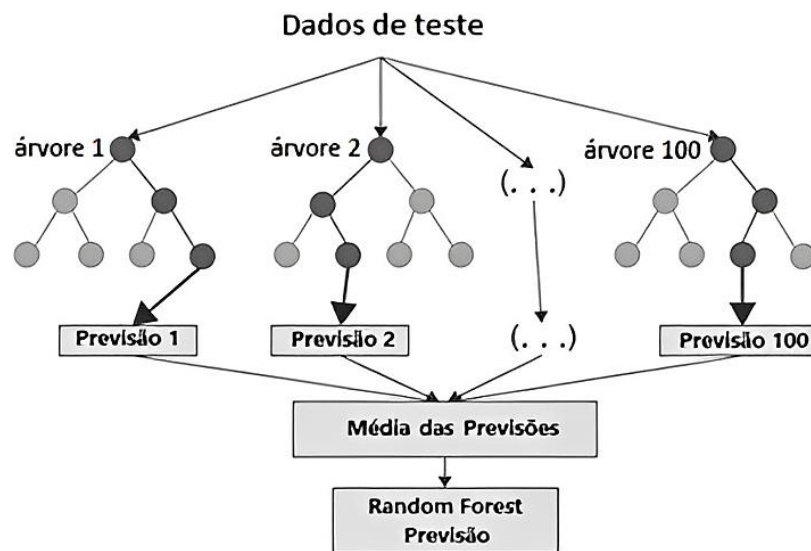


Figura 2.1: Estrutura de uma *Random Forest*

Cada nó “pai” terá 2 nós” filhos”. A árvore irá ter de recorrer ao cálculo da entropia de cada atributo, para escolher o atributo do nó inicial, isto é, o grau de impureza de um dado conjunto de valores, com o objetivo dos nós seguintes serem os mais puros possíveis. Dada uma coleção de dados  $S$  de  $c$  dados, o cálculo da entropia é definido pela expressão (2.1):

$$Entropy(S) = \sum -p(I)\log_2 p(I) \quad (2.1)$$

onde  $p(I)$  é a probabilidade de  $S$  pertencer à coleção de dados  $I$ . No fim, é calculado o ganho, que é a diferença entre o valor da entropia antes da separação e o valor da entropia depois da separação do conjunto de dados. O cálculo do ganho é definido pela expressão seguinte (2.2):

$$Gain(S, A) = Entropy(S) - \sum \frac{|S_v|}{|S|} Entropy(S_v) \quad (2.2)$$

onde  $S$  é o subconjunto de  $S$  e  $A$  o atributo, que tem o valor  $v$ . O atributo que tiver maior ganho é o escolhido para inicializar o nó da árvore (Yang, Li & Song, (2007). No caso do RF, a seleção do atributo do nó inicial não é calculada com base em todos os atributos do conjunto de dados, mas sim de maneira aleatória Genuer, Poggi, e Tuleau-Malot, (2010).

### 2.3.3 *Preference Learning*

*Preference Learning* é um subcampo do *machine learning* que consiste em induzir modelos de preferência preditivos a partir de dados observados. Em geral, uma tarefa de *Preference Learning* consiste num conjunto de itens para os quais as preferências são conhecidas, e a tarefa é aprender uma função que prevê preferências para um novo conjunto de itens ou para o mesmo conjunto de itens num contexto diferente (Doumpos & Zopounidis, 2011).

As técnicas de *Preference learning* são aplicadas em muitas áreas distintas. Por exemplo, um mecanismo de busca classifica as páginas da Web de acordo com as preferências do utilizador, isto é, dada uma consulta C e um conjunto de documentos D, encontra uma classificação dos documentos em D que corresponde à sua relevância em relação a C. Esta classificação é baseada numa preferência desconhecida, que deve ser aprendida a partir do feedback do utilizador em classificações anteriores realizadas pelo mesmo (Furnkranz & Hullermeier 2013).

Da mesma forma, os filmes podem ser classificados de acordo com as preferências do utilizador em características que distinguem diferentes géneros de filmes. Atualmente, os sistemas de recomendação são frequentemente usados por lojas online para recomendar produtos aos seus clientes. Esses sistemas normalmente armazenam uma tabela de dados com produtos dos utilizadores, que regista o grau de preferência de um utilizador por esse produto. Um cliente pode fornecer esse grau de preferência explicitamente, dando algum tipo de feedback (por exemplo, atribuindo uma classificação a um filme ou ver um filme) (Furnkranz & Hullermeier 2013).

Os métodos baseados em preferências são especialmente atraentes do ponto de vista da inteligência artificial, porque permitem especificar os desejos de uma forma declarativa, combinar modos qualitativos e quantitativos de raciocínio e lidar com inconsistências e exceções de uma forma bastante flexível. Uma preferência pode ser considerada como uma restrição, previamente definida com a escolha de um utilizador, a partir dessa escolha irá poder-se fazer a restrição de futuras escolhas. Na verdade, uma vantagem importante de um paradigma de resolução de problemas baseado em preferências é uma maior flexibilidade (Furnkranz & Hullermeier 2013).

Uma abordagem estabelecida para modelar preferências remonta ao conceito de uma função útil. Tal função atribui um grau abstrato de utilidade a cada alternativa em

consideração. Do ponto de vista de ML, um problema óbvio é aprender as funções utilitárias de dados de treino. Dependendo da escala de utilidade subjacente, que é tipicamente numérica ou ordinal, o problema passa a ser de aprendizagem de regressão ou classificação ordinal. Ambos os problemas são bem conhecidos no ML. No entanto, as funções de utilidade muitas vezes implicam requisitos e restrições especiais que devem ser levados em consideração (Furnkranz & Hullermeier 2013).

Outra abordagem é aprender uma relação de preferência binária que compara pares de alternativas (por exemplo, objetos ou rótulos). O treino de um modelo torna-se assim mais simples, principalmente porque as informações de treino comparativas podem ser usadas diretamente em vez de traduzi-las em restrições numa função de utilidade. Por outro lado, a etapa de previsão pode-se tornar mais difícil, uma vez que uma relação de preferência binária aprendida dos dados não é necessariamente consistente no sentido de ser transitiva e, portanto, normalmente não define uma classificação de uma maneira única (Furnkranz & Hullermeier 2013).

Outra abordagem para aprender funções de classificação é proceder a partir de suposições de um modelo específico, isto é, suposições sobre a estrutura das relações de preferência. Esta abordagem é menos genérica do que as anteriores, pois depende muito das suposições concretas feitas (Furnkranz & Hullermeier 2013).

Ainda outra alternativa é recorrer à ideia de técnicas de estimativa local, por exemplo, pelo princípio de estimativa do vizinho mais próximo: Considerando as classificações observadas em situações semelhantes como representativas, uma classificação para a situação atual é estimada com base dessas classificações "vizinhas", normalmente usando um operador de agregação semelhante a uma média. Esta abordagem é em certo sentido ortogonal à anterior baseada no modelo, pois é muito flexível e normalmente não vem com nenhuma suposição de um modelo específico (Furnkranz & Hullermeier 2013).

### **2.3.4 Deep Learning**

O *Deep Learning* (DL) é um subcampo do ML que utiliza um conjunto de algoritmos para aprender diferentes níveis de abstração, utilizando várias camadas de processamentos não lineares. Utilizando essas camadas de processamento não lineares, é possível realizar a deteção de padrões dos dados de entrada durante o seu processo de aprendizagem. Nas técnicas de ML tradicionais, esse processo de extração de padrões e

de características é feito manualmente por um profissional da área. Os modelos de DL podem aprender a classificar um determinado objeto representado por imagens, texto ou som. Para isto são utilizados grande volumes de dados para treinar o modelo, e a utilização de arquiteturas de redes neuronais (LeCun et al, 2015).

O DL tem ganho importância no mundo da tecnologia, tendo aplicações vantajosas em processos onde é necessário um alto nível de precisão de reconhecimento de padrões e tem, recentemente, obtido resultados extremamente positivos, dos quais se destacam alguns superiores até à própria capacidade humana (LeCun et al, 2015).

### 2.3.5 Redes Neuronais

Baseado na estrutura neural do cérebro, as redes neuronais artificiais (RNA) são algoritmos computacionais. O primeiro modelo proposto foi chamado *Perceptron*, composto de apenas um neurónio. Esse modelo era apenas capaz de resolver problemas linearmente separáveis de classificação binária, matematicamente, os neurónios são representados por equações (Haykin, 2009).

As *Convolutional Neural Networks* (CNN) são um tipo de rede neural desenhado para mapear dados de imagens numa variável de saída. Assim sendo, para desenvolver uma representação interna de uma imagem 2D, classificação de modelos de previsão e problemas de previsão de regressão, as CNN são extremamente adequadas. Genericamente, para dados que possuem uma relação de espaço são bastante adequadas (Gehring et al, 2017).

Segundo Sak et al (2014), as *Recurrent Neural Networks* (RNN) foram desenhadas para trabalhar numa sequência de modelos de previsão.

A RNN é uma RN recursiva, onde os valores da entrada são dependentes dos passos que aconteceram nos neurónios anteriores (Bengio et al 1994). O *Long Short-Term Memory* (LSTM) é a RNN mais popular uma vez que resolve os problemas de treinar uma RNN. Uma rede LSTM é composta de RNN com unidades LSTM, e utilizando essas unidades, as redes LSTM conseguem ter uma memória de longo prazo mais eficiente que RNN ao criar uma célula de estado interno de memória que soma a entrada processada ao invés de multiplicá-la (Horn, 1990).

As redes *Multilayer Perceptron* (MLP) são frequentemente usadas para conjuntos de dados em formato tabular e são o tipo de redes mais clássico. Aplicam-se a

classificação de modelos de previsão e a problemas de previsão de regressão. Geralmente, os pixéis de uma imagem podem ser reduzidos a uma linha de dados e podem ser usados como entrada num MLP. As palavras de um documento também podem ser passadas como uma linha de dados de entrada numa MLP. Dados de imagens, dados de texto, dados de séries temporais são também adequadas para este tipo de tratamento (Ramchoun, 2016).

A rede Perceptron de Multicamadas, ou, *Multilayer Perceptron* (MLP), é uma das mais conhecidas e aplicadas arquiteturas de redes neuronais. A arquitetura da MLP consiste numa camada de entrada, uma ou mais camadas ocultas e uma camada de saída (Abirami & Chitra 2020).

A camada de entrada, cujos neurónios pertencentes a essa camada são denominados unidades de entrada, propaga os valores das entradas para as camadas seguintes sem alterações.

As camadas intermediárias (ocultas) transmitem informações por meio das conexões entre as camadas de entrada e saída. Os neurónios pertencentes às camadas intermediárias são denominados unidades ocultas.

A camada de saída, cujos neurónios são chamados unidades de saída, transmite a resposta da rede neuronal à entrada aplicada na camada de entrada. Na Figura 2.2 é apresentado um exemplo da arquitetura com uma camada oculta de uma MLP.

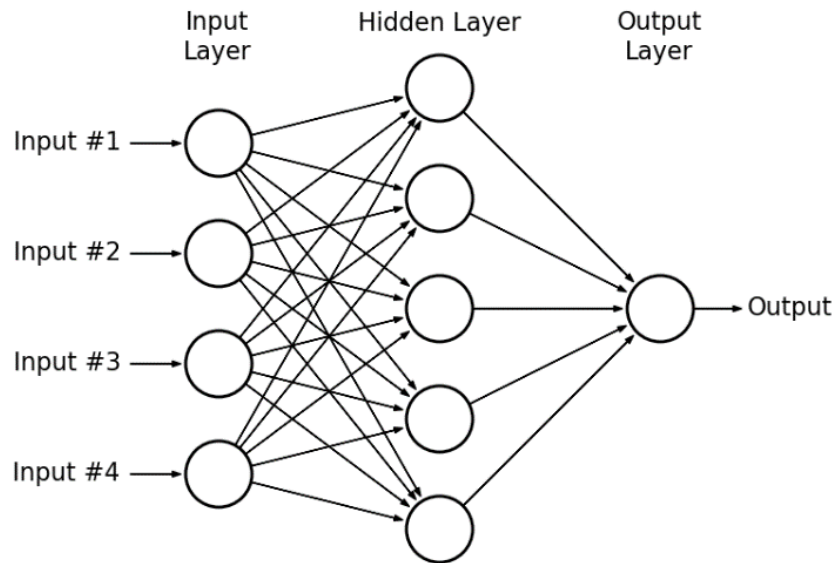


Figura 2.2: Arquitetura da MLP (Abirami &amp; Chitra 2020)

Os neurónios são conectados por arestas e a cada aresta está associado um peso. A comunicação entre os neurónios é realizada por meio dessas interconexões ponderadas. Porém, a comunicação ocorre somente entre as unidades de camadas distintas, portanto, não existe comunicação entre as unidades de uma mesma camada (Menzies et al 2015). A Figura 2.3 exemplifica a operação realizada num neurónio.

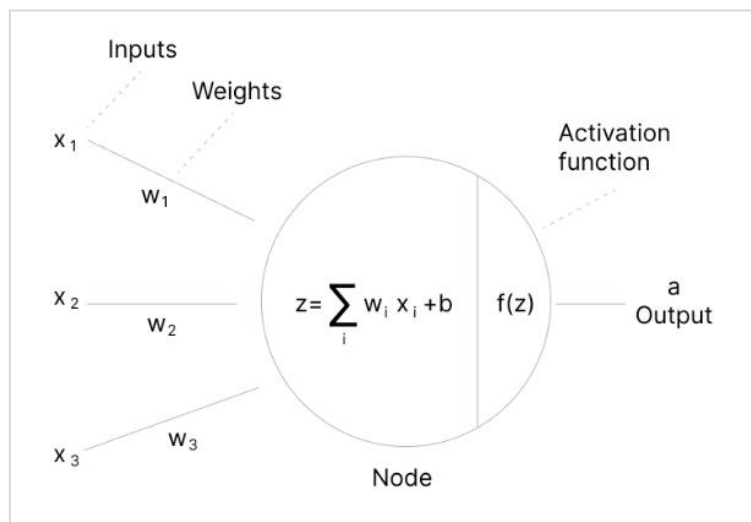


Figura 2.3: Equação num neurónio (Menzies et al 2015)

- Os valores  $x$  referem-se a entradas, sejam os recursos originais ou entradas de uma camada oculta anterior.

- Em cada camada, há também um viés  $b$  que pode ajudar a ajustar melhor os dados.
- O neurónio passa o valor  $a$  para todos os neurónios aos quais está conectado na próxima camada, ou o retorna como valor final.

O cálculo começa com uma equação linear (2.3):

$$Z = X_1 W_1 + X_2 W_2 + X_3 W_3 + b \quad (2.3)$$

Antes de adicionar uma função de ativação não linear (2.4):

$$a = f(z) \quad (2.4)$$

A arquitetura da rede neuronal MLP apresenta três características distintas: variações quanto ao número de camadas e neurónios intermediários da rede, tipo de conexão determinada pelas sinapses e função de ativação.

O modelo MLP apresenta um treino supervisionado demonstrado na Figura 2.4, ou seja, é necessário um “supervisor” para indicar a resposta desejada para um padrão de entrada apresentado à rede durante a fase de aprendizagem. Um erro é calculado pela diferença entre a resposta desejada e a resposta observada. Os hiperparâmetros da rede (pesos e limiares) são ajustados de acordo com esse sinal. Assim, a rede é adaptada ou treinada por um processo iterativo de ajustes sucessivos dos pesos (Mohan Mohanty & Mihir Mohanty, 2022).

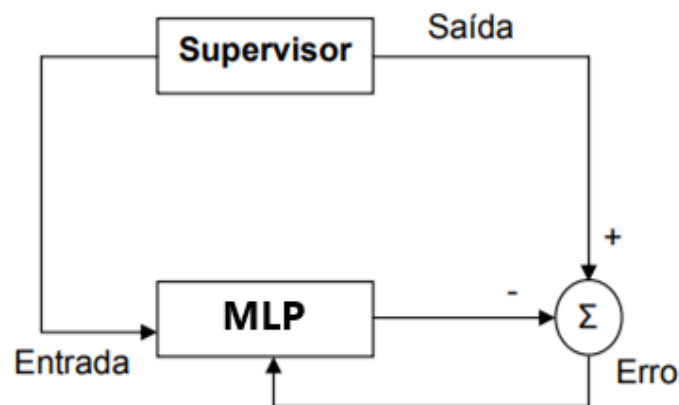


Figura 2.4 Treino supervisionado

Para o treino da rede MLP, o método de aprendizagem utilizado é o algoritmo de retro propagação (*Backpropagation*) apresentado na Figura 2.5. Este algoritmo consiste basicamente em duas fases (Abirami & Chitra 2020). A primeira fase, conhecida como

fase *feed-forward*, as entradas são apresentadas e propagadas por meio da rede, camada por camada, calculando a saída de cada neurónio. Durante essa fase, os pesos são fixos e a saída calculada é comparada com a saída desejada, resultando num erro para cada unidade. Na segunda fase, conhecida como fase *backward* o erro calculado é propagado da camada de saída para a camada de entrada e os pesos são ajustados de acordo com a regra de correção do erro. Originalmente, é denominada de “retro propagação do erro”.

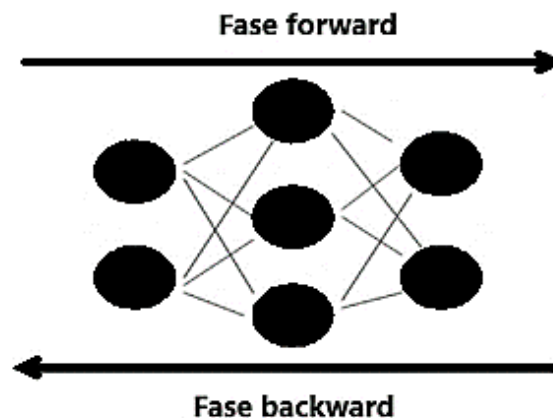


Figura 2.5: Algoritmo de *Backpropagation*

### 2.3.7 Hiperparâmetros

Os algoritmos de ML possuem hiperparâmetros, configurações que se podem usar para controlar o comportamento do algoritmo. Noutras palavras os hiperparâmetros são as variáveis que determinam a estrutura do modelo, como por exemplo, o número de camadas ocultas, função de ativação, etc. Além das variáveis que determinam como o mesmo é treinado, como por exemplo, *batch*, número de épocas de treino etc.

- **Função de ativação**

A função de ativação faz parte de qualquer rede neuronal, tem como objetivo calcular o *output* de um nó que irá servir de *input* para o(s) próximo(s). Esta permite evitar que informação desnecessária passe para os próximos neurónios e, devido a isso, as funções de ativação têm um papel bastante importante e têm influência direta no sucesso ou insucesso do modelo (Hamdan, 2018).

A função de ativação é uma função não linear aplicada por um neurónio para introduzir propriedades não lineares na rede. Uma relação é linear se uma mudança na

primeira variável corresponde a uma mudança constante na segunda variável. Uma relação não linear significa que uma mudança na primeira variável não corresponde necessariamente a uma mudança constante na segunda (Hamdan, 2018). A Figura 2.6 apresenta um exemplo de uma função linear e não linear.

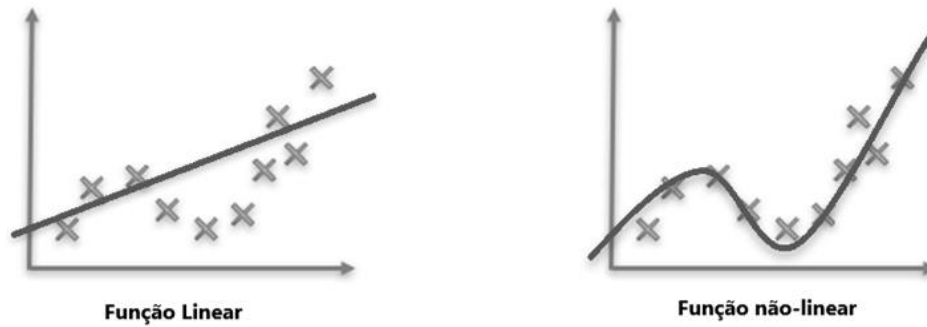
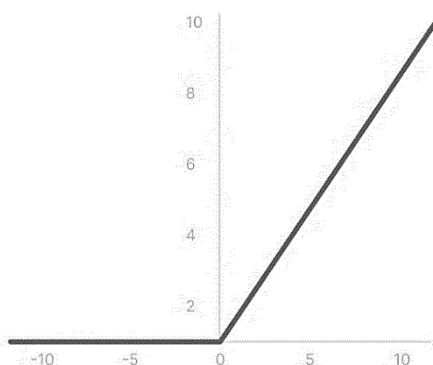


Figura 2.6: Funções linear e não linear

Como afirmam Ramachandran et al (2017), existem diversas funções de ativação, sendo a mais conhecida e utilizada na maior parte das redes neurais a ReLU (*Rectified Linear Units*), devido a ser uma das mais simples.

Funções de ativação final:

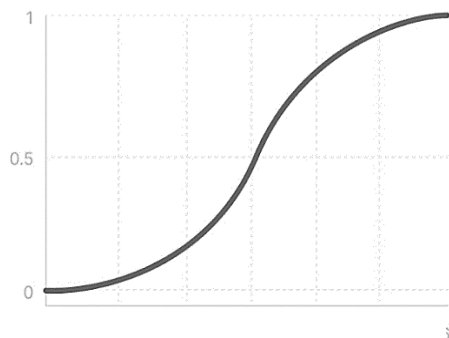
A função ReLU apresenta da Figura 2.7, tem como resultado um valor numérico maior que 0 (2.5).



$$relu(z) = \max(0, z) \quad (2.5)$$

Figura 2.7: Função ReLU

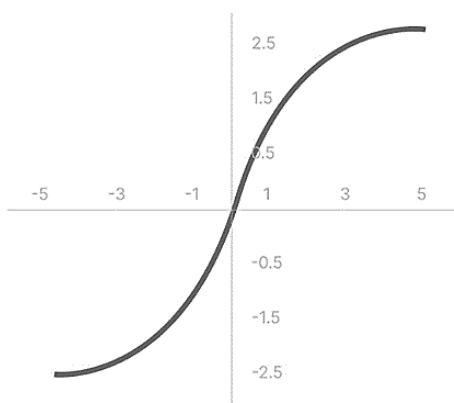
A função *sigmoid* apresentada na Figura 2.8, tem como resultado um valor entre 0 e 1 (2.6).



$$\mathit{sigmoid}(z) = \frac{1}{1 + e^{-z}} \quad (2.6)$$

Figura 2.8: Função *Sigmoid*

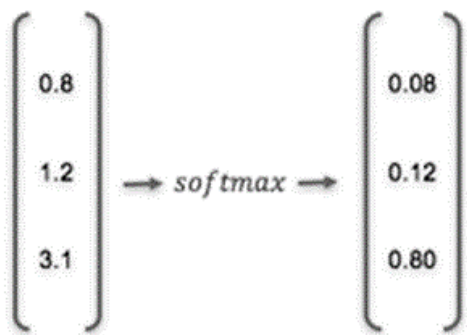
A função *Tanh* apresentada na Figura 2.9, é semelhante à função de ativação *sigmoid*, e ainda tem a mesma forma de S com a diferença na faixa de saída de -1 a 1 (2.7).



$$\mathit{Tanh}(z) = \frac{(e^z - e^{-z})}{(e^z + e^{-z})} \quad (2.7)$$

Figura 2.9: Função *Tanh*

A função *Softmax* apresentada na Figura 2.10, tem como resultado valores entre 0 e 1 para cada uma das saídas que somam 1. Conseqüentemente, isso pode ser inferido como uma distribuição de probabilidade. É usada apenas na camada de saída e não em toda a rede (2.8).



$$\mathit{softmax}(z_i) = \frac{\exp(z_i)}{\sum_j \exp(z_j)} \quad (2.8)$$

(Divide o output para que a soma total seja 1)

Figura 2.10: Função *Softmax*

- **Função de perda (*Loss Function*)**

No contexto de um algoritmo de ML para otimização, a função usada para avaliar uma solução candidata é chamada de função objetivo. Segundo Goodfellow et al (2017), pode-se procurar maximizar ou minimizar a função objetivo, o que significa que se procura uma solução candidata que tenha a pontuação mais alta ou mais baixa, respetivamente. Quando se está a minimizar, também pode-se chamá-la de função de custo, função de perda ou função de erro.

Nas redes neuronais, procura-se minimizar o erro. Como tal, a função objetivo é muitas vezes referida como uma função de perda e o valor calculado pela função de perda é referido simplesmente como “perda”. A função de perda tem um trabalho importante, pois define todos os aspetos do modelo num único valor, de modo que melhorias nesse valor sejam um sinal de um melhoramento. Permite que as soluções candidatas sejam classificadas e comparadas (Reed & Marks, 1999).

Uma função de perda deve ser escolhida quando se vai calcular o erro do modelo durante o processo de otimização. O objetivo será ter os valores da função de perda cada vez menores ao longo do processo de avaliação do modelo. Se os valores forem diminuindo, isso significa que a perda é cada vez menor, portanto o modelo está a conseguir minimizar o erro.

*Mean squared error* (MSE), encontra a diferença quadrática média entre o valor previsto e o valor verdadeiro, dada pela seguinte expressão (2.9):

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - p_i)^2 \quad (2.9)$$

Onde ( $y$ ) é o valor verdadeiro e o ( $p$ ) o valor previsto

*Binary Cross Entropy* (entropia cruzada binária, para prever um resultado binário) quantifica a diferença entre duas distribuições de probabilidade. O modelo prevê uma distribuição de modelo de  $\{p, 1-p\}$ . Usa-se entropia cruzada binária para comparar isso com a distribuição verdadeira  $\{y, 1-y\}$ , dada pela seguinte expressão (2.10):

$$Binary\ Cross\ Entropy = - (y \log(p) + (1 - y) \log(1 - p)) \quad (2.10)$$

Onde ( $y$ ) é o valor verdadeiro e o ( $p$ ) o valor previsto

*Cross Entropy* (entropia cruzada), quantifica a diferença entre duas distribuições de probabilidade. O modelo prevê uma distribuição de modelo de  $\{p_1, p_2, p_3\}$  (onde  $p_1+p_2+p_3 = 1$ ). Usa-se entropia cruzada para comparar isso com a distribuição verdadeira  $\{y_1, y_2, y_3\}$ , dada pela seguinte expressão (2.11):

$$\mathbf{Cross Entropy} = -\sum_i^M y_i \log(p_i) \quad (2.11)$$

Onde ( $y$ ) é o valor verdadeiro e o ( $p$ ) o valor previsto

*Binary Cross Entropy* (Para prever um conjunto de resultados com várias classes) quantifica a diferença entre duas distribuições de probabilidade. O modelo prevê uma distribuição de  $\{p, 1-p\}$  (distribuição binária) para cada uma das classes. Usa-se entropia cruzada binária para compará-los com as distribuições verdadeiras  $\{y, 1-y\}$  para cada classe e somar os seus resultados, dada pela seguinte expressão (2.12):

$$\mathbf{Binary Cross Entropy} = -\sum_i^M (y_i \log(p_i) + (1 - y_i) \log(1 - p_i)) \quad (2.12)$$

Onde ( $y$ ) é o valor verdadeiro e o ( $p$ ) o valor previsto

#### ▪ **Optimizadores**

Quase todos os otimizadores populares em DL são baseados em gradiente descendente. Isto significa que estimam repetidamente a inclinação de uma dada função de perda (*Loss*) e movem os parâmetros na direção oposta (portanto, descendo em direção a um suposto mínimo global).

O exemplo mais simples para um otimizador é provavelmente o *Stochastic Gradient Descent* (ou *SGD*) que tem sido usado desde a década de 1950 por Robbins et al (1951).

Na década de 2010, o uso de métodos de gradiente adaptativos como *AdaGrad* ou *Adam*, por Duchi et al (2011) e por Kingma e Ba (2014), tornaram-se cada vez mais populares. No entanto, em determinadas situações, a comunidade científica voltou a usar *SGD* em vez de métodos de gradiente adaptativo. Além disso, os desafios atuais em DL trazem novas variantes de *SGD* como *LARS* ou *LAMB*, You et al. (2019). Segue-se o estudo de alguns otimizadores usados neste projeto.

O *SGD* (*Stochastic Gradient Descent*), é um método iterativo para otimizar a função de perda, substitui o gradiente real (calculado a partir de todo o conjunto de dados) por uma estimativa do mesmo (calculada a partir de um subconjunto de dados selecionado aleatoriamente). Em problemas de otimização de alta dimensão, isso reduz a carga computacional muito alta, alcançando iterações mais rápidas em troca de uma menor taxa de convergência (Bharath et al, 1999).

$$w_{t+1} = w_t - \alpha \cdot g_t \quad (2.13)$$

*Regra de atualização para SGD (2.13)*

*AdaGrad* é um dos primeiros métodos bem-sucedidos que faz uso de taxas de aprendizagem adaptáveis. O *AdaGrad* dimensiona a taxa de aprendizagem para cada parâmetro com base na raiz quadrada do inverso da soma dos gradientes quadrados. Este procedimento escala direções de gradiente esparsas para cima, o que, por sua vez, permite passos maiores nessas direções. A consequência, o *AdaGrad* pode convergir mais rapidamente em cenários com recursos esparsos (Duchi et al, 2011).

$$G_t = \sum_{\tau=1}^t g_{\tau} \cdot g_{\tau}^T$$

$$w_{t+1} = w_t - \alpha \cdot \text{diag}(G_t)^{-\frac{1}{2}} \cdot g_t \quad (2.14)$$

*Regra de atualização para AdaGrad (2.14)*

*RMSprop* é um otimizador que tem sido imenso usado nos últimos anos. A ideia é semelhante ao *AdaGrad*, mas o escalonamento do gradiente é menos agressivo. É um método no qual a taxa de aprendizagem é adaptada para cada um dos parâmetros. A soma dos gradientes quadrados é substituída por uma média móvel dos gradientes quadrados (Tieleman e Hinton, 2012).

$$v_{t+1} = \beta \cdot v_t + (1 - \beta) \cdot g_t^2$$

$$w_{t+1} = w_t - \frac{\alpha}{\sqrt{v_{t+1}} + \epsilon} \cdot g_t \quad (2.15)$$

*Regra de atualização para RMSprop (2.15)*

*Adam* combina os métodos *AdaGrad*, *RMSprop* e *momentum* num só. A direção do passo é determinada por uma média móvel dos gradientes e o tamanho do passo é aproximadamente limitado superiormente pelo tamanho global do passo. Além disso, cada dimensão do gradiente é redimensionada de forma semelhante ao *RMSprop*.

$$\begin{aligned}
 m_{t+1} &= \beta_1 \cdot m_t + (1 - \beta_1) \cdot g_t \\
 v_{t+1} &= \beta_2 \cdot v_t + (1 - \beta_2) \cdot g_t^2 \\
 m_{t+1} &= \frac{m_{t+1}}{(1 - \beta_1^t)} \\
 v_{t+1} &= \frac{v_{t+1}}{(1 - \beta_2^t)} \\
 w_{t+1} &= w_t - \frac{\alpha}{\sqrt{v_{t+1}} + \epsilon} \cdot m_{t+1}
 \end{aligned} \tag{2.16}$$

*Regra de atualização para Adam (2.16)*

Uma diferença fundamental entre *Adam* e *RMSprop* (ou *AdaGrad*) é o facto de que as estimativas de momento  $m$  e  $v$  são corrigidas em direção a zero. *Adam* é conhecido por alcançar um bom desempenho com poucos ajustes de hiperparâmetros (Kingma & Ba, 2014).

- **Épocas**

Nos modelos de ML é recorrente utilizar-se este hiperparâmetro. O número de épocas (*epochs*) diz respeito ao número de vezes que o modelo vai percorrer o conjunto de dados para treinar. Normalmente o número de épocas a utilizar é alto, como 100 ou 200, até ser capaz de minimizar o erro (Misbahuddin, 2020).

- **Batch**

O hiperparâmetro *Batch* faz parte de diversos modelos de ML, define o número de dados que vão ser processados pelo modelo antes dos pesos da rede neuronal serem atualizados. No final de cada *batch* ser processado, as previsões são calculadas com as variáveis de saída e com base nesses cálculos, é calculado o erro. Através desse erro, são atualizados os pesos do modelo, com o objetivo de ir aumentando a eficácia do modelo. Na Figura 2.19 é apresentado um exemplo onde se entende melhor o processo de leitura de cada batch por época:

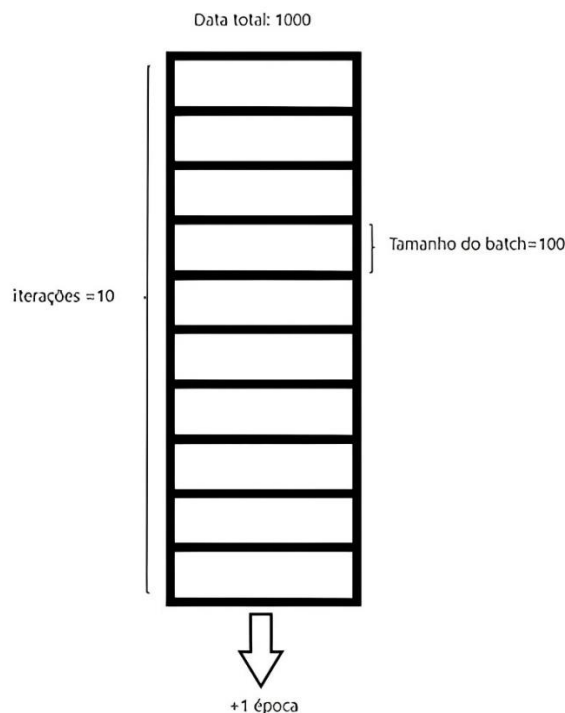


Figura 2.11: Exemplo do Batch

Na Figura 2.11 o *batch* é igual a 100, numa base de dados com 1000 registos, o que acontece é que o modelo irá fazer 10 iterações por época, se o nº de épocas for 10 irá ser feito um total de 100 iterações.

### 2.3.8 Problemas Comuns

Existem vários problemas na implementação de algoritmos de DL (Schmidhuber, 2015) tais como o *overfitting*, qualidade e consistência de dados disponíveis para alimentar os modelos e necessidade de altas capacidades de processamento.

O principal problema de *overfitting* acontece normalmente quando as taxas de aprendizagem testadas são elevadas, podendo causar um cenário onde o algoritmo aprende depressa demais e assume rapidamente ter encontrado o resultado ótimo, em vez de explorar continuamente uma melhor ação a executar num determinado estado. Pode-se entender que, neste caso, o modelo aprendeu tão bem as relações existentes no treino, que acabou apenas decorando o que deveria ser feito, e ao receber as informações das variáveis de previsão nos dados de teste, o modelo tenta aplicar as mesmas regras decoradas, porém com dados diferentes esta regra não tem validade, e o desempenho é afetado, pois modelo treinado não tem capacidade de generalização. Num cenário de *underfitting* o desempenho do modelo já é mau no próprio treino. O modelo não consegue

encontrar relações entre as variáveis e o teste nem precisa acontecer. Um modelo assim já pode ser descartado, pois não terá utilidade. A Figura 2.12 apresenta as três situações possíveis.

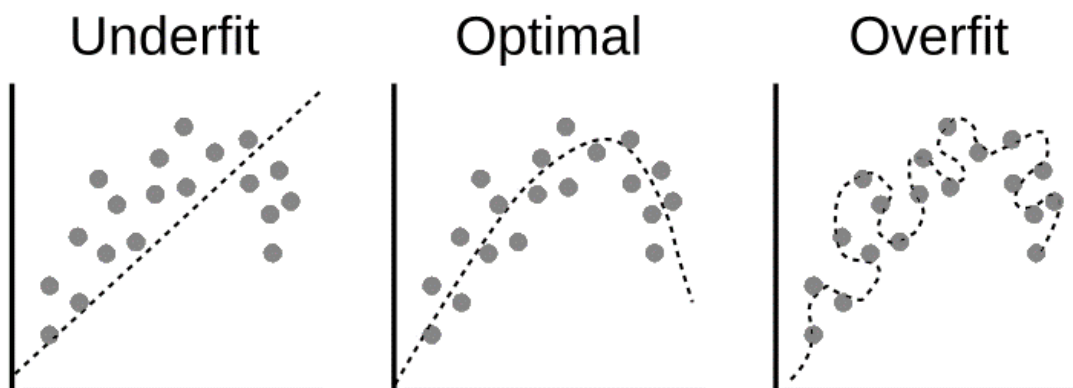


Figura 2.12: Cenários *Underfitting* e *Overfitting*

Existem várias técnicas para contrariar estes fenómenos, nomeadamente quer testando diferentes taxas de aprendizagem, quer através de técnicas de penalização, que na prática são a aplicação de penalizações aos agentes de inteligência artificial na forma de pontos descontados na sua performance geral e que assim desencorajam a repetição consecutiva dos mesmos erros no processo de aprendizagem (Domingos, 2012).

A generalização continua a ser outro dos principais desafios na implementação de DL dos modelos de aprendizagem, ou seja, a complexidade de garantir que os modelos de aprendizagem tenham a capacidade de se generalizar. Mais concretamente se estes modelos têm a capacidade de adaptar-se a ambientes onde são aplicados sem treino prévio nesse ambiente em particular, tendo apenas acesso ao conhecimento obtido em sessões de treino anteriores, não relacionadas especificamente com o problema atual (Tassa et al., 2012).

## 2.4 Medidas de avaliação

Segundo Wang et al (2007), muitos investigadores acham que a métrica de desempenho mais razoável é a razão entre o número de amostras classificadas corretamente e o número total de amostras. Essa medida é chamada de *accuracy* e, por definição, também funciona quando são mais de dois rótulos (mais que uma classe).

Num modelo de classificação é previsto a classe de cada instância de dados, atribuindo a cada amostra um valor previsto (positivo ou negativo): assim, no final do procedimento de classificação, cada amostra irá se enquadrar num dos quatro casos a seguir:

- Os positivos reais que são positivos corretamente previstos são chamados de *True positives* (TP);
- Os falsos negativos que são negativos previstos incorretamente são chamados de *false negatives* (FN);
- Os negativos reais que são negativos previstos corretamente são chamados de *True negatives* (TN);
- Os falsos positivos que são positivos previstos incorretamente são chamados de *false positives* (FP).

Esta partição pode ser apresentada numa tabela  $2 \times 2$  chamada matriz de confusão  $M = \begin{pmatrix} TP & FN \\ FP & TN \end{pmatrix}$  que descreve completamente o resultado da tarefa de classificação.

### 2.4.1 Accuracy

Movendo-se para métricas globais com três ou mais entradas, muitos investigadores consideram calcular a *accuracy* como o caminho padrão a seguir, como Akosa (2017). A *accuracy*, na verdade, representa a razão entre as instâncias previstas corretamente e todas as instâncias no conjunto de dados (2.17):

$$\mathbf{accuracy} = \frac{TP}{TP+FP+FN} \quad (2.17)$$

Por definição, a *accuracy* é definida para cada matriz de confusão  $M$  e intervalos no intervalo real da unidade  $[0,1]$ ; o melhor valor de 1 corresponde à classificação perfeita

$M = M = \begin{pmatrix} n^+ & 0 \\ 0 & n^- \end{pmatrix}$  e o pior valor 0 corresponde a um erro de classificação perfeito  $M = \begin{pmatrix} 0 & n^+ \\ n^- & 0 \end{pmatrix}$ .

### 2.4.2 *Cross Validation*

Normalmente antes de treinar um modelo separa-se de forma aleatória os dados em base de treino e teste. Após o modelo treinado utiliza-se a parte corresponde ao teste para avaliar o desempenho do modelo. É uma boa abordagem, porém pode não ser suficiente para avaliar o modelo e é aqui que entra o *Cross Validation*.

*Cross Validation* é uma técnica muito utilizada para avaliação de desempenho de modelos de ML. Esta técnica consiste em particionar o *dataset* em conjuntos (partes), onde um conjunto é utilizado para treino e outro conjunto é utilizado para teste e avaliação do desempenho do modelo. A sua utilização tem altas chances de detetar se o modelo está só ajustado aos dados de treino, ou seja, sofrendo *overfitting*. (Wong & Yeh 2019).

Um dos métodos é o *K-fold*, que consiste em dividir a base de dados de forma aleatória em K subconjuntos (em que K é definido previamente) com aproximadamente a mesma quantidade de amostras em cada um deles. A cada iteração, treino e teste, um conjunto formado por K-1 subconjuntos são utilizados para treino e o subconjunto restante será utilizado para teste gerando o resultado da métrica para avaliação. Esse processo garante que cada subconjunto será utilizado para teste no momento da avaliação do modelo (Wong & Yeh 2019).



### 3. METODOLOGIA

Este capítulo apresenta a metodologia de desenvolvimento do projeto referente às atividades de seleção de dados, pré-processamento, transformação, mineração e avaliação de modelos de ML, ilustrada na Figura 3.1.

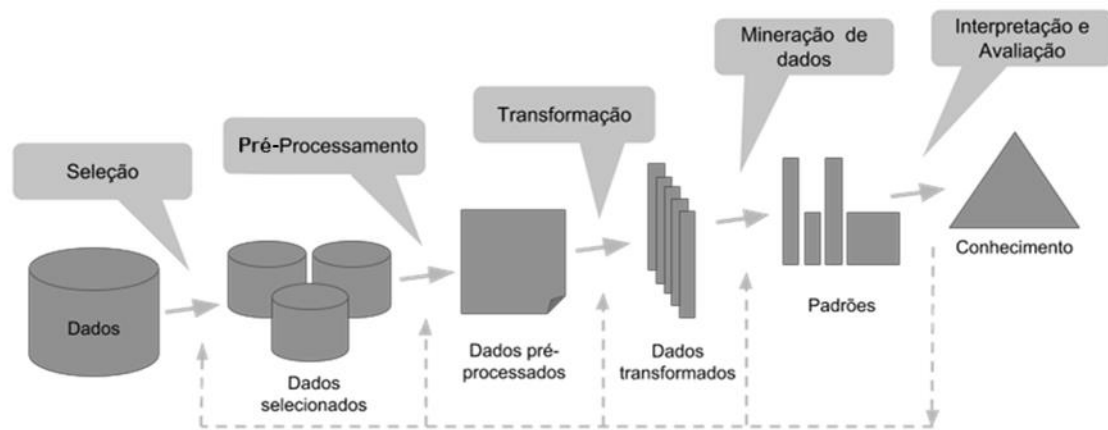


Figura 3.1: Etapas do ML<sup>1</sup>

#### 3.1 Seleção dos dados

A etapa de seleção dos dados desta metodologia, tem como objetivo selecionar os dados recolhidos. Tanto a quantidade e a qualidade dos dados recolhidos determinam o quão preciso o modelo vai ser (Mohri et al, 2018).

#### 3.2 Pré-processamento

Na etapa de pré-processamento dos dados é feita a limpeza, seleção de atributos e a normalização dos dados, para que possam ser usados para treinar o modelo. O pré-processamento dos dados influencia significativamente a qualidade dos modelos gerados (Dougherty, 2012).

##### 3.2.1 Normalização

A normalização é uma operação em dados brutos que os redimensionam ou os transformam de forma que cada registo tenha uma contribuição uniforme. Esta trata de uma das questões principais de dados que dificulta o processo de aprendizagem

<sup>1</sup> Fonte: [https://www.researchgate.net/figure/Figura-1-Fases-do-Processo-de-KDD-8-O-produto-esperado-da-extracao-de-conhecimento-e\\_fig1\\_308749622](https://www.researchgate.net/figure/Figura-1-Fases-do-Processo-de-KDD-8-O-produto-esperado-da-extracao-de-conhecimento-e_fig1_308749622) [consultado a 22/10/2022]

de algoritmos de ML, ou seja, a presença de características dominantes. A normalização de dados é uma etapa essencial da atividade de pré-processamento dados que envolve a transformação de parâmetros num intervalo comum para que valores numéricos com maiores valores não possam dominar os menores valores. O objetivo principal é minimizar a presença das características cuja contribuição numérica é maior na discriminação de classes de padrões. Existem vários métodos para normalizar os dados dentro de um intervalo especificado com base em medidas estatísticas dos dados brutos (não normalizados) (García et al, 2015).

#### **3.2.2 Seleção de atributos**

Por vezes, uma grande quantidade de atributos não contribui para uma melhor performance do modelo. Os atributos são as variáveis de entrada de um modelo preditivo. São variáveis independentes utilizadas para a construção de um modelo de ML. A seleção de variáveis de entrada para modelos de ML é de grande importância.

Tendo em conta um conjunto de dados, se uma ou mais variáveis de entrada influenciarem negativamente a precisão do modelo, muito provavelmente seria melhor retirá-las para evitar previsões erradas. Devido a isso, foi necessário escolher os atributos que realmente influenciam na previsão final. Para efetuar a seleção dos atributos, foi utilizado o algoritmo de *Random Forest* para saber quais as variáveis de entrada com maior importância na identificação de alimentos.

As técnicas para atribuir pontuações a variáveis de entrada para um modelo de previsão, permitem escolher os atributos com maior impacto na performance de previsão dos algoritmos. As pontuações de importância do atributo podem ser calculadas para problemas que envolvem a previsão de uma variável contínua, e os problemas que envolvem a previsão de um rótulo de uma classe (classificação) (Kuhn & Johnson, 2013).

As pontuações são úteis e podem ser usadas para reduzir o número de variáveis de entrada. Isso pode ser alcançado usando as pontuações de importância para selecionar os atributos a serem excluídos (pontuações mais baixas) ou os atributos a serem mantidos (pontuações mais altas). Este é um tipo de seleção de atributos que pode simplificar o problema que está a ser modelado, acelerar o processo de modelação. A exclusão de atributos pode, em alguns casos, melhorar o desempenho do modelo.

### 3.3 Transformação dos dados

Na etapa de transformação dos dados é definido o processo que modifica os dados para o formato apropriado, geralmente do formato de um sistema de origem para o formato exigido de um sistema novo. Esta etapa também pode ser necessária para uniformizar dados provenientes de várias fontes.

Os dados podem ser transformados de várias maneiras, dependendo do objetivo e dos requisitos do sistema de destino. As transformações criadas podem não ser apenas usadas para limpar, filtrar, dividir e juntar dados, mas também para enriquecê-los.

### 3.4 Mineração dos dados

Na fase de mineração de dados, retiram-se padrões dos dados recolhidos, com o objetivo de retirar informações úteis. Aqui são utilizados algoritmos de *machine learning* para construir modelos.

O modelo de ML adotado para treinar os dados, utiliza um algoritmo. Esse algoritmo irá calcular as variáveis de saída. O resultado do modelo é dependente dos dados de entrada que o alimentam, logo ser necessário todo o processo de preparação dos dados já anteriormente apresentado. O modelo utilizado, irá ser de classificação.

### 3.5 Interpretação e avaliação dos dados

Na fase de interpretação e avaliação verifica-se a precisão do modelo. Se este tiver o desempenho pretendido será o modelo escolhido, caso contrário começa-se o processo novamente.

Para avaliar classificações binárias, a *Accuracy* foi calculada em matrizes de confusão e é uma das métricas adotadas mais utilizadas em tarefas de classificação binária.

Para também a avaliar o desempenho do modelo, irá se considerar um cenário de *5-fold cross validation* apresentado na Figura 3.2. Os dados serão divididos em 5 partes. Na primeira iteração a primeira parte será utilizada para teste e as partes restantes serão utilizadas para treino do modelo gerando uma métrica de avaliação. Na segunda iteração, a segunda parte será utilizada para teste enquanto as restantes para treino. Esse processo será repetido 5 vezes até que toda a base de dados passe pelo processo de treino e teste gerando uma métrica de avaliação média para o modelo.

iteração 1	teste	treino	treino	treino	treino
iteração 2	treino	teste	treino	treino	treino
iteração 3	treino	treino	teste	treino	treino
iteração 4	treino	treino	treino	teste	treino
iteração 5	treino	treino	treino	treino	teste

Figura 3.2: Exemplo K-fold de 5

O valor de K foi escolhido cuidadosamente de forma que cada grupo de dados para treino e teste sejam grandes o suficiente para representarem estatisticamente o *dataset* original. É possível encontrar na literatura a orientação para a utilização de  $k=5$  ou  $k=10$ .

Um valor de K mal escolhido pode resultar numa interpretação errada do desempenho do modelo como medidas de avaliação do modelo com alta variância (as medidas podem variar bastante de acordo com os dados utilizados para treino).

## 4. IMPLEMENTAÇÃO

Este capítulo apresenta a implementação do projeto referente às atividades de reestruturação de dados, engenharia de atributos, seleção de registos, normalização, seleção de atributos e divisão de dados. Irá ser descrito o modelo utilizado para o processo de treino, teste e validação dos dados, e os diferentes hiperparâmetros utilizados.

### 4.1 Dados de registos alimentares

O conjunto de dados utilizado para criação e validação do modelo contém 587.187 dias de registos alimentares, de 9,9 mil utilizadores do *MyFitnessPal* (Evans, 2017) de setembro de 2014 a abril de 2015 (disponível online<sup>2</sup>). A tabela de dados é constituída por quatro colunas: *Id de utilizador*, *data do registo*, *alimentos ingeridos* (com os respetivos valores nutricionais e a sequência) e os *goals* (agregação diária da quantidade de nutrientes recomendada). Os dados encontram-se armazenados em formato JSON.

#### 4.1.1 Reestruturação de dados

Na fase de pré-processamento foram decompostas as colunas dos *alimentos ingeridos* e dos *goals*, de maneira que possam ser lidas e processadas. Este processo teve como resultado uma estrutura com 16 colunas, apresentada na Figura 4.1.

- Contexto: *Id de utilizador*; *data*, *Ids dos alimentos*, *sequência* (nº da refeição do dia);
- Consumo total: *calorias*, *hidratos de carbono*, *gordura*, *proteína*, *sódio* e *açúcar*;
- Objetivo diário: *calorias*, *hidratos de carbono*, *gordura*, *proteína*, *sódio* e *açúcar*.

	date	food_ids	goal_calories	goal_carbs	goal_fat	goal_protein	goal_sodium	goal_sugar	sequence	total_calories	total_carbs	total_fat	total_protein	total_sodium
0	2014-09-15	[1, 2, 3, 4, 4]	1572.0	196.0	52.0	79.0	2300.0	59.0	1	2430	96	37.0	50.0	2300.0
1	2014-09-16	[5, 1, 2, 3, 6, 7]	1832.0	229.0	61.0	92.0	2300.0	69.0	1	1862	158	54.0	114.0	2300.0
2	2014-09-17	[1, 2, 3, 6, 8, 9, 10]	1685.0	210.0	56.0	85.0	2300.0	63.0	1	2251	187	60.0	98.0	2300.0
3	2014-09-18	[1, 6, 2, 3, 11, 12]	1597.0	199.0	53.0	80.0	2300.0	60.0	1	2001	113	81.0	202.0	2300.0
4	2014-09-19	[1, 7, 13, 12, 2, 3, 12, 12]	1589.0	198.0	53.0	80.0	2300.0	60.0	1	2158	180	89.0	115.0	2300.0
5	2014-09-20	[1, 5, 2, 3, 14, 15, 16, 11, 17, 18]	2823.0	352.0	93.0	142.0	2300.0	106.0	1	2691	282	92.0	216.0	2300.0

Figura 4.1: Tabela com os dados após pré-processamento

<sup>2</sup> <https://www.kaggle.com/datasets/zvikinozadze/myfitnesspal-dataset/discussion/194277> [consultado a 12/3/2022]

### 4.1.2 Engenharia de atributos

A engenharia de atributos permite obter um conhecimento maior do conjunto de dados, podendo a partir da informação já existente, extrair dados importantes que possam contribuir para uma melhor previsão (Zaidi, 2015).

Na fase da engenharia de atributos foram criados atributos novos, que serão utilizados como variáveis de entrada. A coluna da *date* foi decomposta nas colunas do *year*, *mm*, *day* e pôde-se obter a informação para criar a coluna do *weekday* que não existia, como está representado na Figura 4.2. Esta fase pode contribuir para encontrar padrões alimentares, uma vez que as pessoas têm padrões de consumo específicos em determinados dias (ex.: fim de semana).

	date		weekday	year	mm	day
0	2014-09-15		0	2014	09	15
1	2014-09-16		1	2014	09	16
2	2014-09-17	→	2	2014	09	17
3	2014-09-18		3	2014	09	18
4	2014-09-19		4	2014	09	19

Figura 4.2: Transformação do atributo *date*

Outra transformação efetuada foi a separação do atributo dos *Ids dos alimentos*, que pode ser observada na Figura 4.3. Foi feita uma binarização, representando a presença do alimento atribuindo o valor 1 se estiver presente na refeição e o valor 0 se estiver ausente. Esta representação permite estruturar o input da rede neuronal associada.

food_ids	1	10	100	1000	1001	1002	1003	1004	1005	1006	...
0 [2, 3, 4, 4]	1	0	0	0	0	0	0	0	0	0	...
1 [5, 1, 2, 3, 6, 7]	1	1	0	0	0	0	0	0	0	0	...
2 [1, 2, 3, 6, 8, 9, 10]	1	0	0	0	0	0	0	0	0	0	...
3 [1, 6, 2, 3, 11, 12]	0	0	0	0	0	0	0	0	0	0	...
4 [1, 7, 13, 12, 2, 3, 12, 12]	0	0	0	0	1	0	0	0	0	0	...
	716	0	0	0	0	0	0	0	0	0	...
	717	0	0	0	0	0	0	0	0	0	...
	718	0	0	0	0	0	0	0	0	0	...
	719	0	0	0	0	0	0	0	0	0	...

Figura 4.3: Transformação do atributo *food-ids*

### 4.1.3 Seleção de registos

De forma a reduzir o *dataset*, foi feita uma seleção de dados para uma base de dados reduzida (Figura 4.4). Este processo teve como objetivo fazer uma filtragem de utilizadores e de alimentos para remover *outliers* do *dataset*, permitindo assim extrair padrões alimentares num *dataset* reduzido.

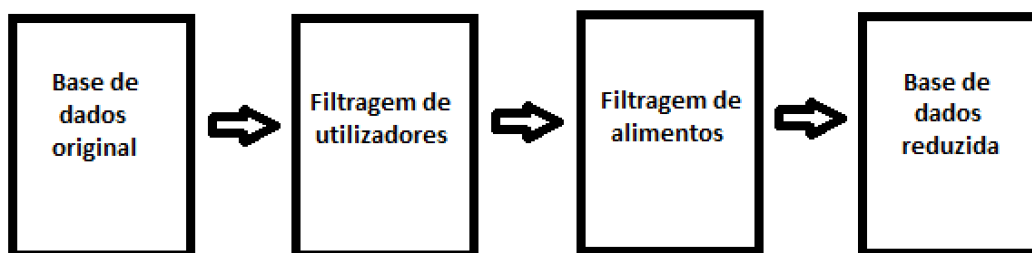


Figura 4.4: Metodologia de seleção de registos

Da análise dos dados da Figura 4.5, é possível constatar uma distribuição não homogénea dos registos presentes no *dataset* pelos utilizadores. De acordo com esta observação, foi feita uma seleção de utilizadores com um número maior de registos, acima do percentil 90. Esta seleção permite remover os *outliers* no *dataset* que podem comprometer a qualidade do modelo de previsão, passando a contar só com os utilizadores que contribuem para um melhor reconhecimento de padrões.

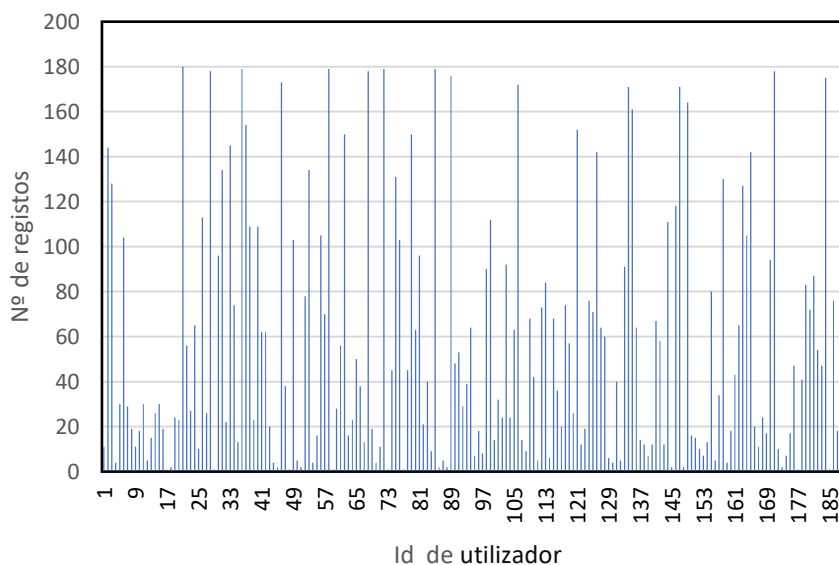


Figura 4.5: Número de registos por utilizador (200 utilizadores)

Da análise dos dados da Figura 4.6, é possível visualizar uma distribuição não homogénea do número de registos por alimento. Não tendo uma distribuição semelhante na escolha, foram seleccionados os alimentos mais consumidos. Para esta seleção, foram escolhidas as refeições onde se encontram presentes os alimentos ingeridos acima do percentil 10. Esta seleção permite remover os alimentos de menor consumo que podem comprometer a qualidade do modelo de previsão.

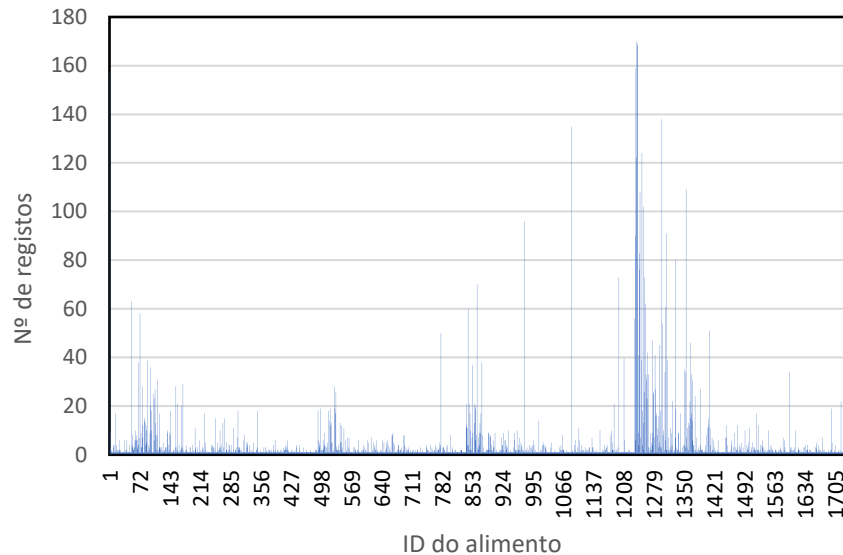


Figura 4.6: Contagem de alimentos de 10 utilizadores

O resultado das seleções anteriores foi uma base de dados reduzida, com um conjunto de dados com utilizadores com maior número de registos e refeições com alimentos de consumo mais frequente. A Figura 4.7 apresenta o número total de refeições e o número de alimentos diferentes seleccionados.



Figura 4.7: Conjunto de dados final

#### 4.1.4 Normalização dos dados

Na fase de normalização de dados foram redimensionados os valores presentes no *dataset*. As colunas dos alimentos já estavam normalizadas, e assinalados por 1 e 0 em cada refeição. Observando a Figura 4.8 pode-se constatar a distribuição dos valores das calorias por intervalos, o processo consistiu em redimensionar esses valores em intervalos diferentes, proporcionando uma melhor aprendizagem ao modelo.

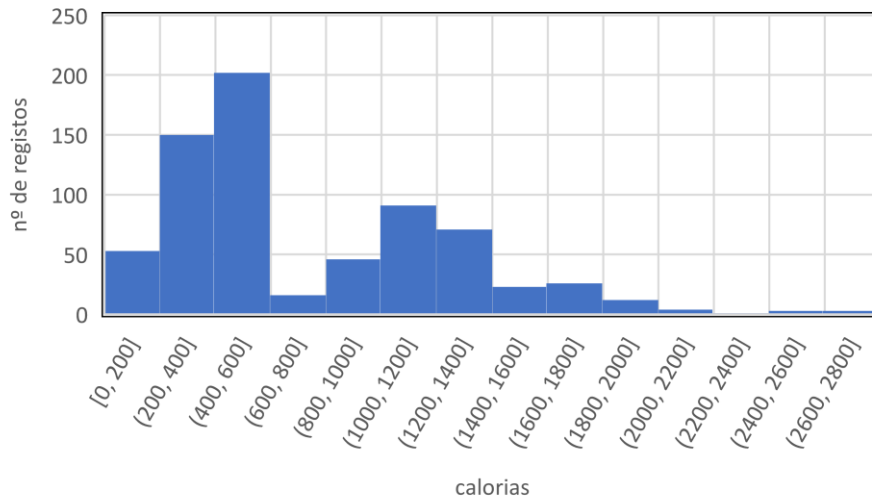


Figura 4.8: Histograma do valor das calorias não normalizado

O método utilizado para normalizar os dados teve como base o valor mínimo e o valor máximo de cada coluna. O método Mín-Máx (MMN) dimensiona os dados não normalizados para limites inferiores e superiores predefinidos linearmente (Han et al, 2011). Os dados são geralmente redimensionados dentro do intervalo de 0 a 1 ou -1 a 1. Na equação (4.1), os limites inferior e superior dos valores são indicados com  $nMin$  e  $nMax$  respetivamente:

$$x'_{i,n} = \frac{x_{i,n} - \min(x_i)}{\max(x_i) - \min(x_i)} (nMax - nMin) + nMin \quad (4.1)$$

As escalas [0, 1] ou [-1, 1] são consideradas para analisar o desempenho da classificação.

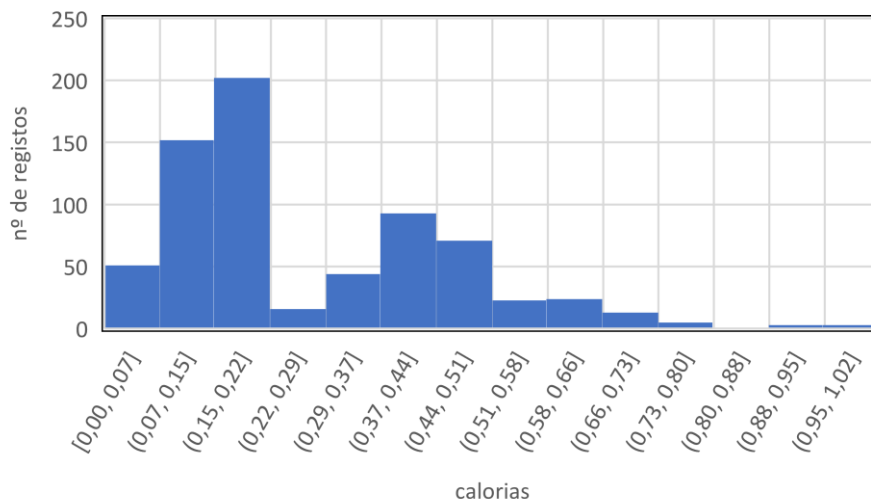


Figura 4.9: Histograma do valor das calorias normalizado

O resultado da normalização pode ser observado no histograma apresentado na Figura 4.9, onde mostra a distribuição dos registros em função dos intervalos das calorias.

#### 4.1.5 Seleção de atributos

Na atividade de seleção de atributos, não se considerou o consumo de nutrientes em cada refeição, mas sim, os objetivos diários (*goals*) de cada utilizador, pois estes valores representam as suas metas diárias.

O algoritmo de *Random Forest* permite avaliar o *dataset* com a importância de cada atributo (*feature\_importances*). As pontuações mais altas, representam os atributos com maior importância na previsão. Na figura 4.10 pode-se observar as colunas utilizadas e as suas relevâncias. Para evitar os atributos de menos importância e que poderiam influenciar negativamente o desempenho do modelo, foram escolhidos aqueles com uma importância acima dos 5%. A coluna *year* foi a única abaixo desse valor, então não se considerou como variável de entrada.

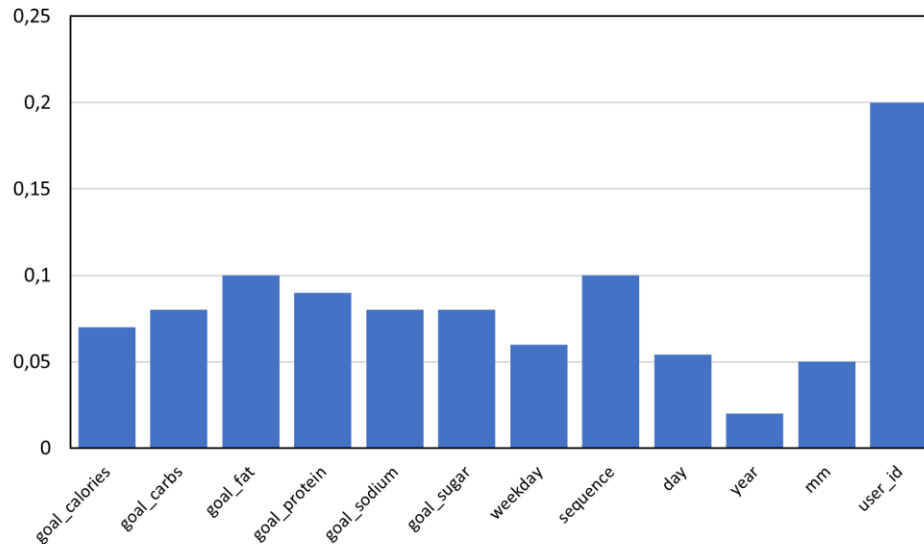


Figura 4.10: *feature\_importances* usando *Random Forest*

### 4.1.6 Divisão dos dados

Como apresentado na Figura 4.11, o *dataset* foi dividido nos datasets de treino, validação e teste:

- **Dados de treino (70%):** treinar e ajustar o modelo.
- **Dados de validação (10%):** fornecer uma avaliação com outros dados que não sejam os de treino, para evitar o *overfitting* do modelo. A avaliação torna-se mais parcial à medida que o conjunto de dados de validação é incorporado à configuração do modelo.
- **Dados de teste (20%):** fornecer uma avaliação ajustada no conjunto de dados de teste.

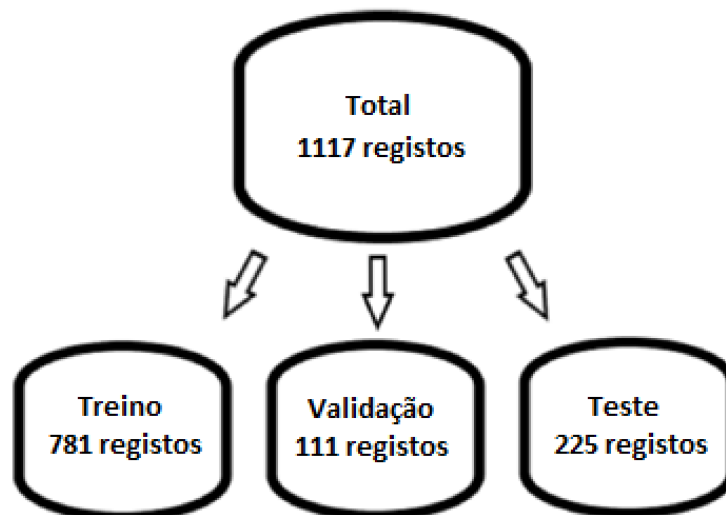


Figura 4.11: Esquema de separação dos dados

## 4.2 Criação do modelo

O modelo de ML criado deve sugerir os alimentos adequados para ingestão de cada utilizador. Para isso o modelo irá realizar uma classificação dos alimentos por cada refeição, processando o *dataset* de treino, validação e teste.

### 4.2.1 Classificação dos alimentos por refeição

O modelo de classificação deve prever a presença do alimento a partir das variáveis de entrada. Dentro de cada classe (refeição), encontram-se os alimentos que irão ser classificados como presentes ou ausentes.

A Tabela 4.2 apresenta a identificação dos alimentos, com a atribuição de um id que irá representar o respetivo alimento. A Tabela 4.3 apresenta as refeições, em que cada refeição estão assinalados os alimentos como presentes ou ausentes, de acordo com os seus ids.

Alimento	ID
	1
	2
	3

Tabela 4.2: Id do alimento




Refeição	1	2	3
	1	1	1
	1	1	0
	0	1	1

Tabela 4.3: Identificação dos Ids na refeição

Para o processo de classificação apresentado na Figura 4.12, foi utilizada uma MLP. O *output* da classificação da rede neuronal foi a refeição com todos os alimentos previstos. Em cada classe (refeição) os alimentos obtiveram uma probabilidade de serem escolhidos (valor entre 0 e 1). Para comparar o resultado do *output*, as probabilidades de cada alimento ser ingerido irão ser comparadas aos valores verdadeiros. Para a previsão ser correta, os alimentos com probabilidade de escolha têm de estar presentes nos valores verdadeiros.

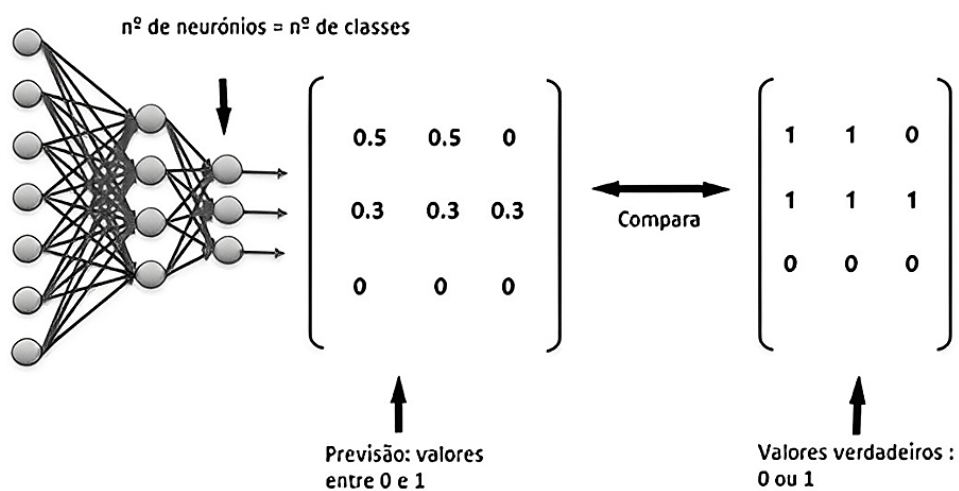


Figura 4.12: Previsão de vários rótulos de várias classes

### 4.2.2 Processo de treino, teste e validação

O processo de treino começa com os valores das variáveis de entrada a serem inseridos no modelo, onde irão ser treinados. Enquanto os dados de treino estão a ser treinados, os dados de validação estão a ser utilizados para fazer validações iniciais para que o modelo possa fazer uma verificação daquilo que está a aprender e assim evitar o efeito de *overfitting*.

No processo de teste, são retirados os rótulos dos dados de teste, para que em seguida o modelo preveja as respectivas classes. Para avaliar se o modelo aprendeu corretamente é utilizada a métrica de avaliação (*accuracy*), que compara os valores verdadeiros com os valores previstos, obtendo o resultado da classificação dos dados de teste.

O resultado da métrica de avaliação irá ser útil para ajustar os hiperparâmetros e repetir o processo novamente até se obter o conhecimento dos mais adequados a utilizar. A Figura 4.13 ilustra esquematicamente todo este processo de aprendizagem do modelo.

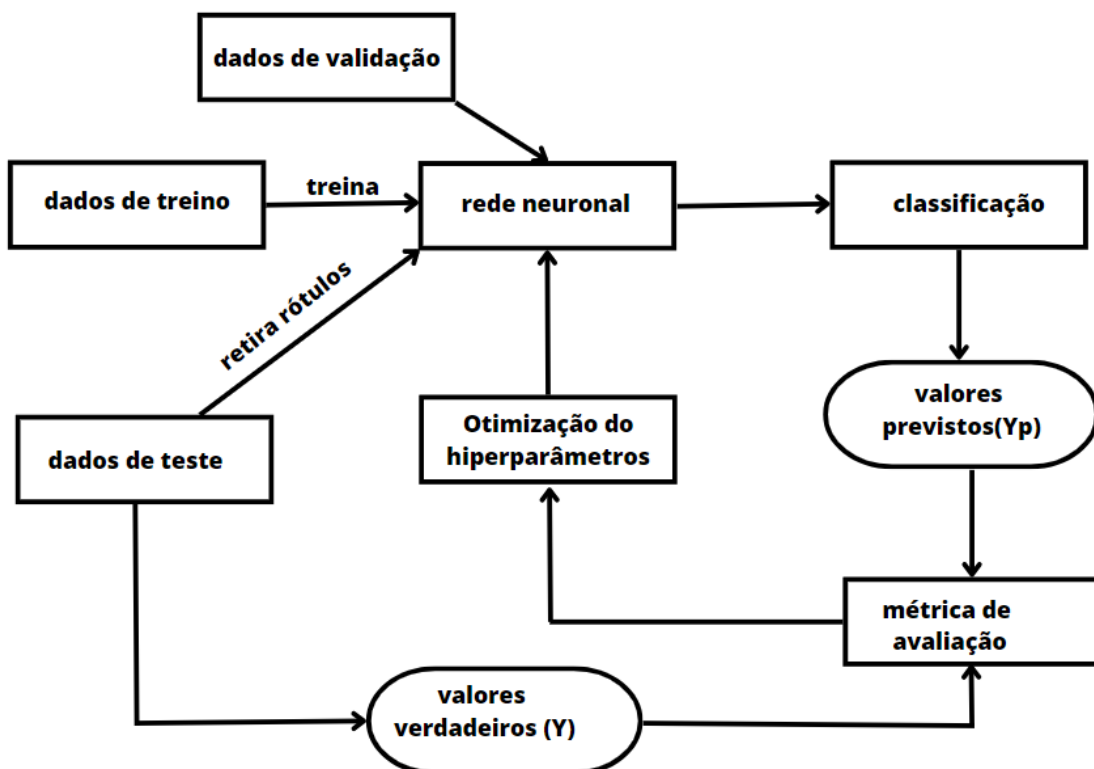


Figura 4.13: Processo de aprendizagem do modelo

Na fase de teste foram realizadas várias experiências com diferentes hiperparâmetros, com o objetivo de avaliar o impacto de cada um. Foi também importante definir a métrica utilizada para avaliar o modelo, para que a escolha seja adequada. Na tabela 4.4 são apresentados os hiperparâmetros utilizados.

<b>Hiperparâmetros</b>	<b>Testes elaborados (valores utilizados)</b>
<b>Épocas</b>	1-1000
<b><i>Batch</i></b>	1-128
<b>Camadas escondidas</b>	1-30
<b><i>Units</i> (nº de neurónios)</b>	1-3000
<b>Função de ativação</b>	<i>ReLU, Sigmoid, Tanh e Softmax</i>
<b>Função de perda</b>	<i>Binary Crossentropy, MSE, Cross Entropy</i>
<b>Optimizadores</b>	<i>SGD, AdaGrad, RMSprop, Adam</i>

Tabela 4.4: Valores dos hiperparâmetros



## 5. ANÁLISE DOS RESULTADOS

Este capítulo apresenta os resultados dos testes elaborados da fase de validação e da fase de teste dos dados. Foi utilizado um computador com um processador i7-8565U, 8GB de RAM e com uma placa gráfica Geforce GTX 1050Ti. O desempenho foi avaliado com a métrica de *accuracy* (2.17).

### 5.1 Resultados da normalização

Os testes elaborados foram realizados com os hiperparâmetros da Tabela 5.1.

Épocas	200
<i>Batch</i>	30
Camadas escondidas	1
<i>Units</i> (nº de neurónios)	10
Função de ativação	<i>Sigmoid</i>
Função de perda	<i>Binary Crossentropy</i>
Optimizadores	<i>RMSprop</i>

Tabela 5.1: Hiperparâmetros utilizados nos testes da normalização

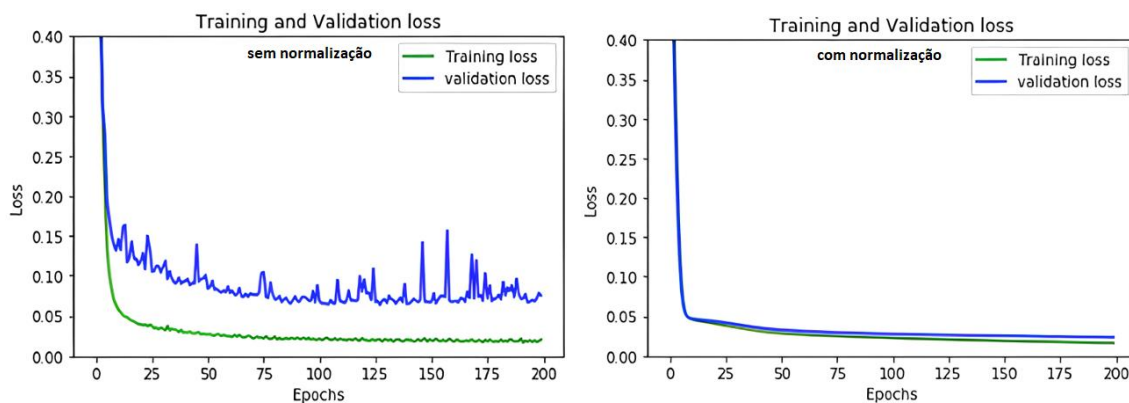


Figura 5.1: Gráficos de perda com e sem normalização

Analisando os dois gráficos das perdas de validação e teste da Figura 5.1, pode-se verificar, que o gráfico de perda sem normalização apresenta picos de valores de perda, enquanto, que com a normalização dos dados o modelo consegue diminuir a perda gradualmente. A normalização teve um impacto positivo.

## 5.2 Resultados dos ajustes do modelo

Para ajustar o modelo irá ser apresentado os resultados dos vários hiperparâmetros utilizados e desses quais irão ser os melhores a utilizar. Para facilitar a comparação dos mesmos irá ser apresentado um gráfico da função de perda e uma tabela com os valores da *accuracy* de cada teste.

### 5.2.1 Função de perda (*Loss Function*)

Os testes elaborados para testar as várias funções de perda, foram realizados com os hiperparâmetros da Tabela 5.2.

Épocas	200
<i>Batch</i>	30
Camadas escondidas	1
<i>Units</i> (nº de neurónios)	10
Função de ativação	<i>Sigmoid</i>
Optimizadores	<i>RMSprop</i>

Tabela 5.2: Hiperparâmetros utilizados nos testes da função de perda

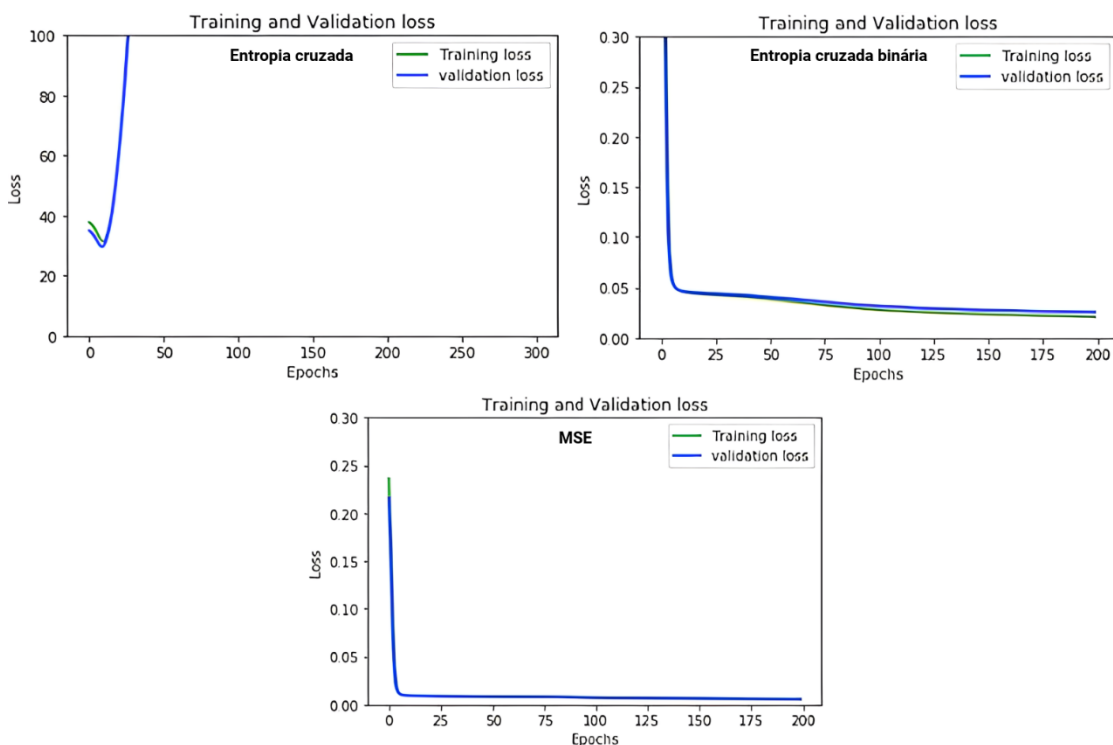


Figura 5.2: Gráficos das funções de perda

<b>Métrica</b>	<b><i>MSE</i></b>	<b><i>Cross Entropy</i></b>	<b><i>Binary Cross Entropy</i></b>
<i>Accuracy (%)</i>	56,8%	1,8%	58,3%
Tempo (s)	19.1s	19,2s	19s

Tabela 5.3: Resultados das funções de perda

Analisando a Figura 5.2 e a Tabela 5.3, pode-se verificar que a função *Binary Cross Entropy* e a *Mean squared error (MSE)*, obtiveram melhores resultados de *accuracy*. A função *Cross Entropy* obteve um resultado de *accuracy* inferior e valores da perda superiores. Optou-se por usar a a função *Binary Cross Entropy*, justificado pelo desempenho superior.

### 5.2.2 Função de ativação final

Os testes elaborados para testar as várias funções de ativação final foram feitos com hiperparâmetros da Tabela 5.4.

Épocas	200
<i>Batch</i>	30
Camadas escondidas	1
<i>Units</i> (nº de neurónios)	10
Função de perda	<i>Binary Crossentropy</i>
Optimizadores	<i>RMSprop</i>

Tabela 5.4: Hiperparâmetros utilizados nos testes com a função de ativação

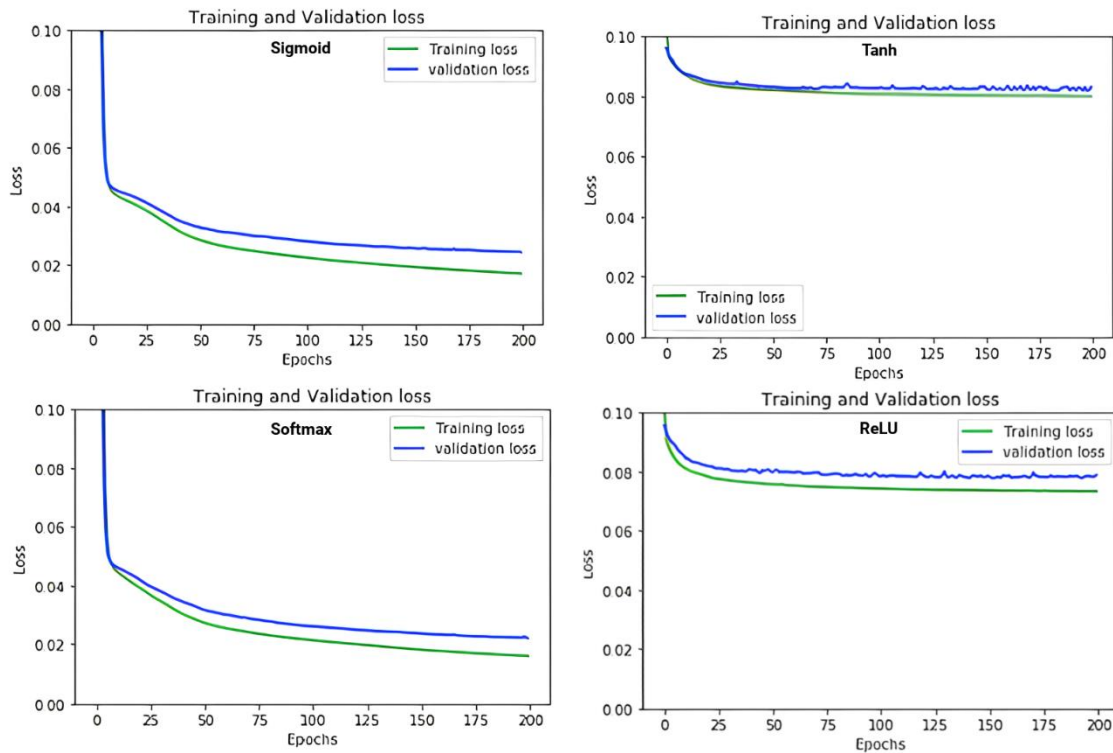


Figura 5.3: Gráficos de perda com as funções de ativação final

Métrica	<i>ReLU</i>	<i>Sigmoid</i>	<i>Tanh</i>	<i>Softmax</i>
Accuracy (%)	48,6%	58,5%	23,7%	51%
Tempo (s)	19s	19,1s	19,3s	19s

Tabela 5.5: Resultados das Funções de ativação

Com base nos gráficos obtidos na Figura 5.3 e da *accuracy* na Tabela 5.5 pode-se verificar que a função *Sigmoid* se adapta melhor ao problema conseguindo diminuir o valor da perda e obtendo um valor de *accuracy* superior. A função de ativação a *Sigmoid* foi a escolhida.

### 5.2.3 Optimizadores

Os testes com optimizadores foram feitos com os hiperparâmetros da Tabela 5.6.

Épocas	200
<i>Batch</i>	30
Camadas escondidas	1
<i>Units</i> (n° de neurónios)	10
Função de ativação	<i>Sigmoid</i>
Função de perda	<i>Binary Crossentropy</i>

Tabela 5.6: Hiperparâmetros utilizados nos testes do otimizador

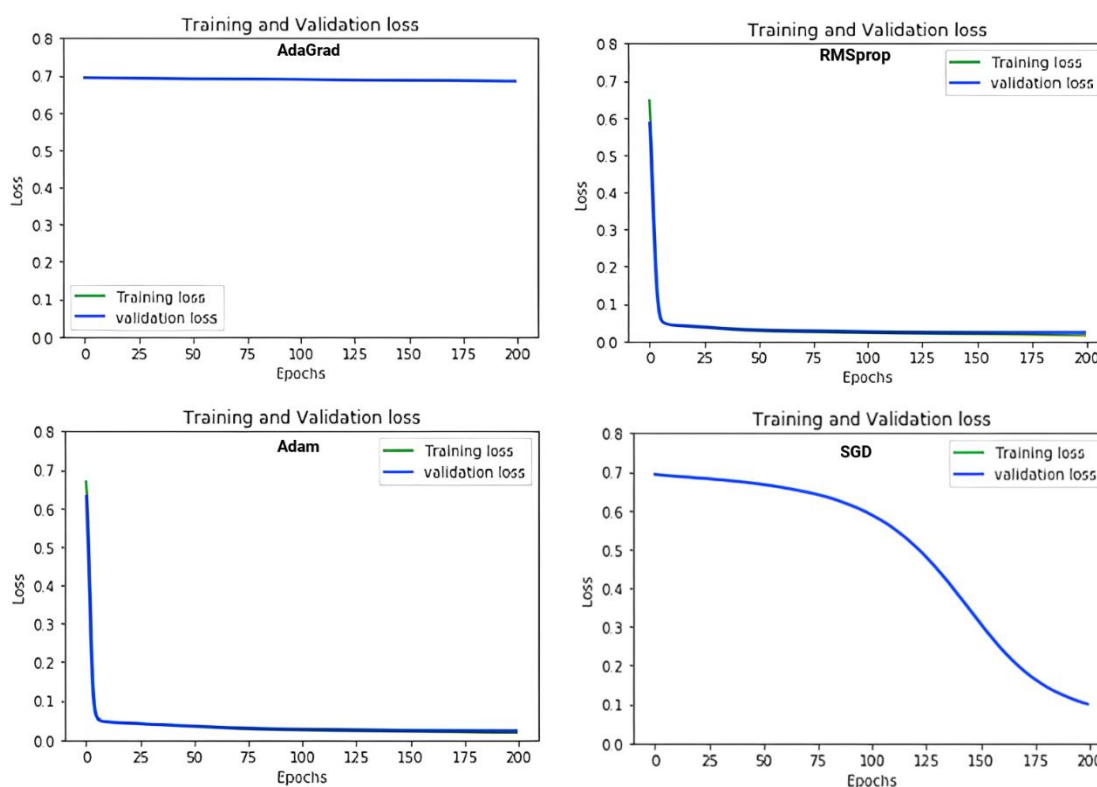


Figura 5.4: Gráfico da função de perda com os otimizadores

Métrica	<i>SGD</i>	<i>AdaGrad</i>	<i>RMSprop</i>	<i>Adam</i>
Accuracy (%)	12,7%	1,6%	58,3%	49,5%
Tempo (s)	23.2s	23.8s	19.3s	19.2s

Tabela 5.7: Resultados dos otimizadores

De acordo com os resultados obtidos na Figura 5.4, os otimizadores *SGD* e *AdaGrad* obtiveram valores de perda maiores. Em relação ao *RMSprop* e ao *Adam*, os resultados são melhores, conseguindo diminuir os valores da perda de forma gradual e

obtendo valores de *accuracy* superiores, na Tabela 5.7. O otimizador escolhido foi o *RMSprop*, por ser aquele com *accuracy* superior.

### 5.2.4 Épocas

Os testes elaborados para avaliar o impacto do número de épocas foram realizados com os hiperparâmetros da Tabela 5.8.

<i>Batch</i>	30
Camadas escondidas	1
<i>Units</i> (nº de neurónios)	10
Função de ativação	<i>Sigmoid</i>
Função de perda	<i>Binary Crossentropy</i>
Otimizadores	<i>RMSprop</i>

Tabela 5.8: Hiperparâmetros utilizados nos testes com épocas

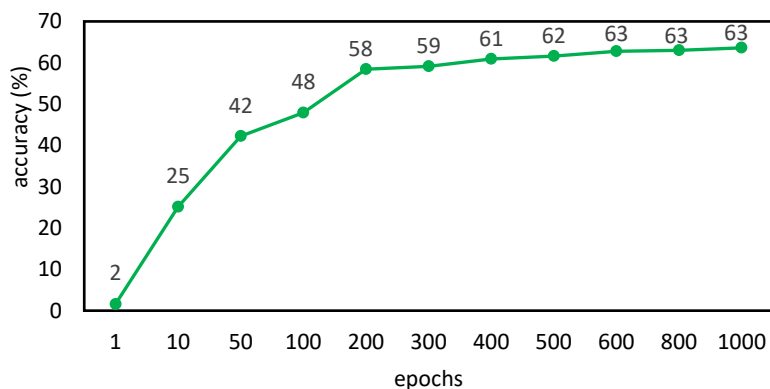


Gráfico 5.1: *Accuracy* do número de épocas

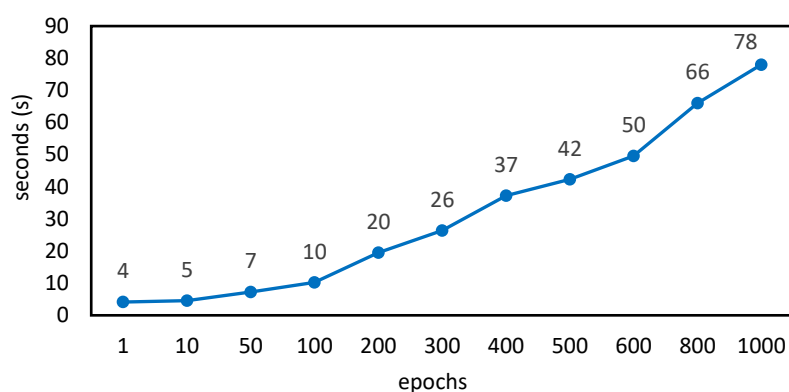


Gráfico 5.2: Número de épocas em relação ao tempo de treino

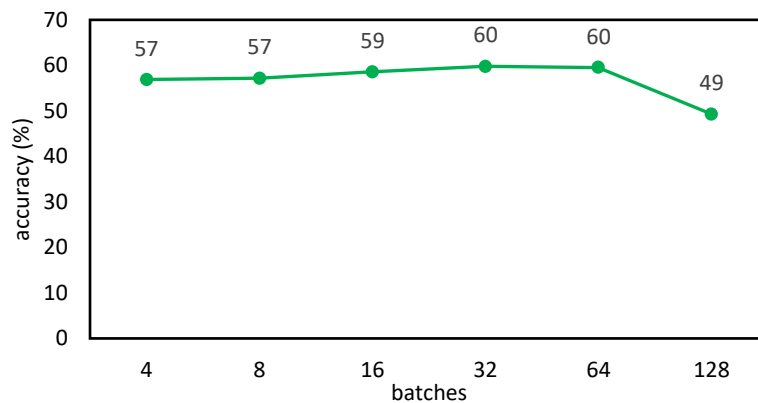
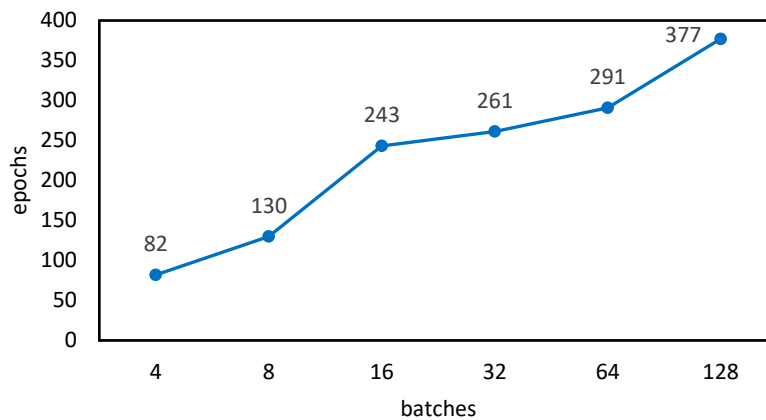
De acordo com o gráfico 5.1 pode-se verificar que, quanto maior o nº de épocas melhor vai ser a *accuracy*. Pode-se verificar no Gráfico 5.2, que quanto maior o número de épocas, maior o tempo de treino. Contudo, nos testes seguintes foi definido que se os valores da perda subirem o modelo pára de treinar, evitando o excesso de vezes que o modelo percorre o conjunto de dados de treino para atingir o valor de *accuracy* de mais de 60%.

### 5.2.5 Batch

Os testes elaborados para avaliar o impacto do tamanho do *Batch* foram realizados com os hiperparâmetros da Tabela 5.9.

Camadas escondidas	1
<i>Units</i> (nº de neurónios)	10
Função de ativação	<i>Sigmoid</i>
Função de perda	<i>Binary Crossentropy</i>
Optimizadores	<i>RMSprop</i>

Tabela 5.9: Hiperparâmetros utilizados nos testes do *Batch*

Gráfico 5.3: Accuracy do tamanho *Batch*Gráfico 5.4: Número de épocas em relação ao tamanho *Batch*

De acordo com os valores obtidos no gráfico 5.4, se o valor do *Batch* for maior o número de épocas também aumenta (o número de vezes que o modelo vai percorrer o conjunto de dados para treinar irá ser maior). No Gráfico 5.3 pode-se verificar que os resultados de *accuracy* aumentaram ligeiramente até ao 64, descendo depois no 128. Quanto maior for o *Batch* será necessário mais épocas para atingir um valor de *accuracy* de cerca 60%. Foi escolhido o valor 32 para o *batch* justificado pelo melhor resultado.

### 5.2.6 Camadas escondidas (*Hidden layers*)

Os testes com camadas ocultas foram realizados com os hiperparâmetros da Tabela 5.10.

<i>Batch</i>	32
<i>Units</i> (nº de neurónios)	10
Função de ativação	<i>Sigmoid</i>
Função de perda	<i>Binary Crossentropy</i>
Optimizadores	<i>RMSprop</i>

Tabela 5.10: Hiperparâmetros utilizados nos testes das camadas ocultas

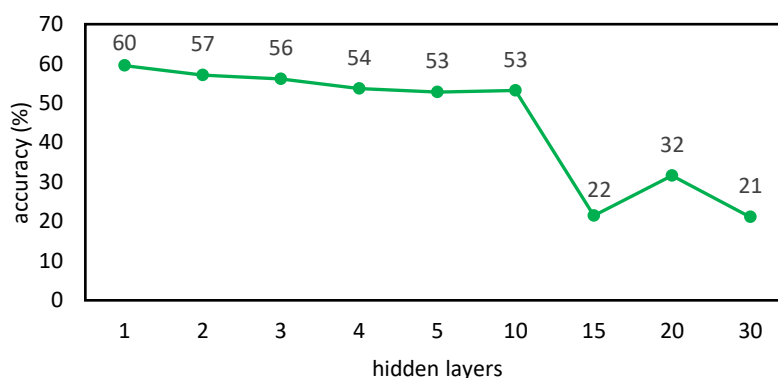


Gráfico 5.5: Accuracy do número de camadas escondidas

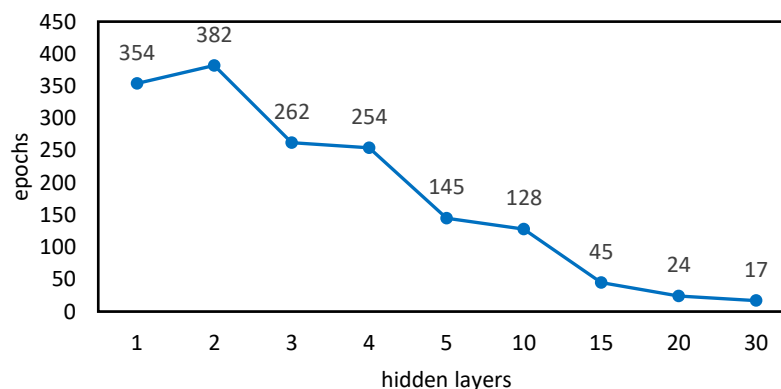


Gráfico 5.6: Número de camadas escondidas em relação ao número de épocas

Analisando o Gráfico 5.5 pode-se verificar que quando se aumenta o nº de camadas ocultas, a *accuracy* não aumenta. Neste modelo não é necessário aumentar as camadas ocultas. O gráfico 5.6 apresenta a relação entre o número de camadas ocultas da rede MLP e o número de épocas, verifica-se que quando se aumenta o valor de camadas ocultas, menor o número de épocas irá ser. A tabela 5.11 apresenta o melhor registo com uma camada.

<b>Hidden layers</b>	<b>Accuracy (%)</b>	<b>Tempo(s)</b>	<b>Épocas</b>	<b>Parâmetros Totais</b>
1	59,5	19,3	354	7,556

Tabela 5.11: Resultado de uma camada escondida

### 5.2.7 Units (nº de neurónios)

Os testes elaborados foram realizados com os hiperparâmetros da tabela 5.12.

Camadas escondidas	1
<i>Batch</i>	32
Função de ativação	<i>Sigmoid</i>
Função de perda	<i>Binary Crossentropy</i>
Optimizadores	<i>RMSprop</i>

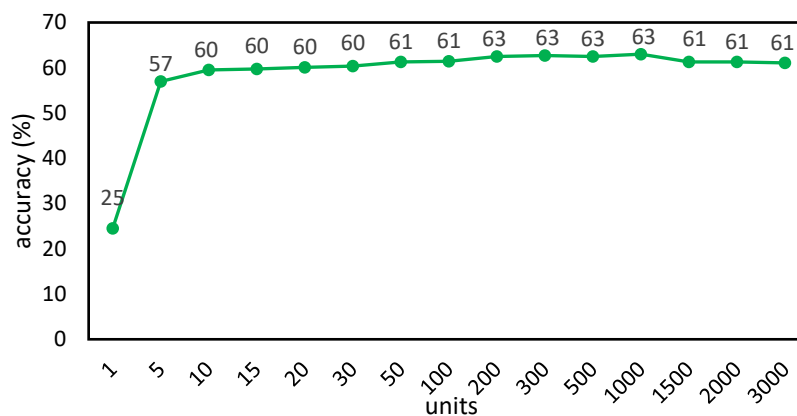
Tabela 5.12: Hiperparâmetros utilizados no número de *Units*

Gráfico 5.7: Accuracy do número de neurónios

<i>Units</i>	<i>Accuracy (%)</i>	<i>Tempo(s)</i>	<i>Épocas</i>	<i>Parâmetros Totais</i>
200	62,5	25,3	191	134286
300	62,7	26,1	195	200986
500	62,5	25,9	143	334386
1000	63	26,7	64	667886

Tabela 5.13: Melhores resultados de *accuracy* do número de *Units*

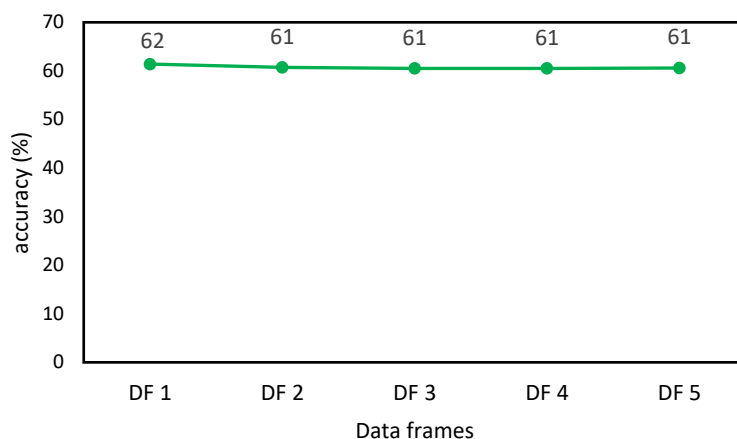
Os valores da *accuracy* da Tabela 5.13 e do gráfico 5.7, foram aumentado com o aumento do nº de units até 1000. Posteriormente, começaram a descer ligeiramente, levando a concluir que o aumento de neurónios teve uma influência positiva aumentando o nº de parâmetros significativamente, atingindo cerca de 63% de *accuracy*.

### 5.3 Validação

O método *k-fold validation* foi aplicado na fase de validação para um k igual a 5. Como tal o *dataset* foi particionado, dando origem a 5 *dataframes*. A Tabela 5.14 apresenta os hiperparâmetros utilizados.

Camadas escondidas	1
<i>Units</i> (nº de neurónios)	750
<i>Batch</i>	32
Função de ativação	<i>Sigmoid</i>
Função de perda	<i>Binary Crossentropy</i>
Optimizadores	<i>RMSprop</i>

Tabela 5.14: Hiperparâmetros utilizados nos testes de *k-fold validation*

Gráfico 5.8: Accuracy com  $K\text{-Fold} = 5$ 

Teste	Accuracy (%)	Tempo(s)	Épocas	Parâmetros Totais
DF 1	61,5	28,4s	45	1,164,386
DF 2	60,7	33,1s	58	1,164,386
DF 3	60,5	25,1s	45	1,164,386
DF 4	60,5	23,9s	39	1,164,386
DF 5	60,6	28,5s	45	1,164,386

Tabela 5.15: Accuracy com  $K\text{-Fold} = 5$ 

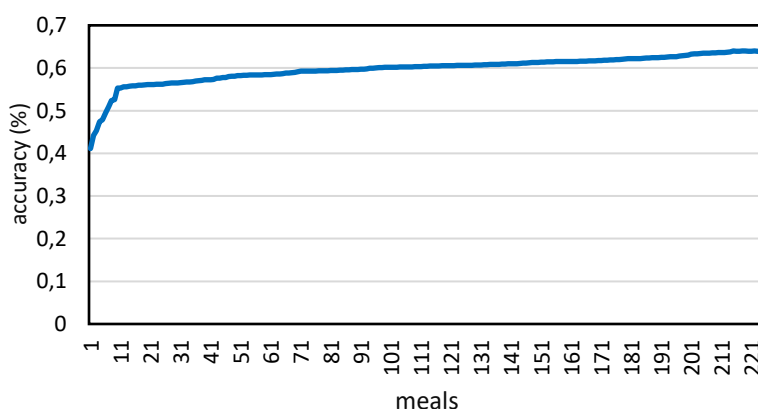
Analisando a Tabela 5.15 e o gráfico 5.8, pode-se verificar que os resultados de *accuracy* foram semelhantes, o que significa que mesmo alterando os dados de treino e teste o modelo consegue adaptar-se e aprender com dados novos.

#### 5.4 Resultado da MLP com os hiperparâmetros escolhidos

Para pôr em prática os resultados dos testes realizados, foi realizado um teste após a escolha dos hiperparâmetros que se encontram na Tabela 5.16.

Camadas escondidas	1
<i>Units</i> (nº de neurónios)	1000
<i>Batch</i>	32
Função de ativação	<i>Sigmoid</i>
Função de perda	<i>Binary Crossentropy</i>
Optimizadores	<i>RMSprop</i>

Tabela 5.16: Hiperparâmetros escolhidos

Gráfico 5.9: *Accuracy* média ao longo das refeições

O gráfico 5.9 permite visualizar a *accuracy* média obtida por refeição. Pode-se verificar que os valores obtidos estiveram quase sempre acima dos 60%.

## 5.5 Discussão de resultados

A fase de pré-processamento dos dados influenciou significativamente a qualidade do modelo gerado. A transformação dos dados permitiu organizar e gerar novos atributos que se relevaram importantes para a recomendação de alimentos. A seleção de variáveis de entrada permitiu selecionar os atributos com um impacto superior na previsão. Pode-se concluir que a aplicação da normalização no *dataset* teve um impacto positivo, permitindo um aumento de *accuracy* do modelo.

A escolha de hiperparâmetros da MLP permitiu definir os valores para o modelo obter melhores resultados de *accuracy*, oferecendo uma maior quantidade de previsões corretas. *Binary Cross Entropy* destacou-se das outras funções de perda, tendo sido a escolhida. A função de ativação final que obteve a melhor *accuracy* foi a *Sigmoid* e o otimizador foi o *RMSprop*. O aumento do número de neurónios teve um impacto positivo no modelo fazendo subir o score da *accuracy*. O aumento do número de camadas ocultas não teve uma influência positiva no resultado da *accuracy*, fazendo diminuir a *accuracy* à medida que se aumentava o número de camadas ocultas.

A validação feita com o método de *k-fold* provou que mesmo alterando os dados de treino e teste o modelo consegue adaptar-se e aprender com dados novos. O que permite concluir que modelo não sofre de *overfitting*, já que a sua performance com dados não utilizados no *dataset* de treino se revelou “positiva”.

A *accuracy* global para o modelo treinado é de aproximadamente 63%. O *dataset* é bastante heterogéneo e o objetivo deste trabalho foi a recomendação de alimentos de acordo com as preferências globais dos utilizadores e não de um utilizador específico. A utilização de dados específicos de um utilizador, quando existentes, poderão potencialmente melhorar a *accuracy* da previsão de alimentos.

## 6. CONCLUSÃO

O objetivo do presente trabalho centrou-se na aplicação de um modelo *machine learning* em dados de registos alimentares, para fazer previsões a partir das metas nutricionais de cada utilizador e auxiliar o cuidador informal. Com estas previsões, seria possível ajudar os doentes de alzheimer, permitindo que o cuidador informal obtivesse uma melhor tomada de decisão.

O processamento dos dados teve uma grande influência, não só nos resultados do modelo, mas também, na análise dos respetivos dados, permitindo selecionar os dados que proporcionaram uma previsão mais eficaz.

A exploração de várias combinações de configurações, adaptando os valores dos diferentes hiperparâmetros abordados na execução do trabalho experimental, permitiu retirar conclusões junto de cada teste e considerando a métrica definida que, conjuntamente, demonstrou os resultados da performance da MLP.

Com a resolução deste projeto foram adquiridas diversas competências, não só na área do *machine learning*, mas também em outras como a mineração de dados, análise de dados, ciência de dados e também conhecimentos sobre a doença de alzheimer. Foi bastante trabalhoso, mas muito enriquecedor, pois a recomendação de alimentos recorrendo a técnicas de *machine learning* com base no registo prévio alimentar é uma solução. Devido a isso, foi possível aprender diversas técnicas utilizadas pela comunidade científica, que tendem a crescer cada vez mais.

### 6.1 Limitações e desafios

O principal desafio deste trabalho esteve relacionado com a disponibilidade de *datasets* de análise que possam ser representativos do grupo de utilizadores da solução proposta.

Outra limitação foi a utilização de dados relacionados com uma comunidade de utilizadores sem articulação com as preferências específicas de um utilizador.

## **6.2 Trabalho futuro**

Como trabalho futuro, pretende-se continuar a aplicar outros algoritmos de *machine learning* neste tipo de dados, com base no registo prévio de alimentos, com o objetivo de encontrar e testar diferentes técnicas que poderão ser úteis, e melhorando, assim, a qualidade de vida dos doentes.

Seria interessante também aplicar o modelo com a rede MLP a outros dados relacionados com o registo alimentar, por exemplo, dados com a temperatura ou com o estado emocional do doente em cada refeição, com o objetivo de saber se consegue ter uma maior capacidade de previsão com outros atributos.

Também seria interessante fazer a articulação de um modelo geral com um modelo de preferências individuais de utilizadores específicos. Fazendo recomendações com base nas preferências de um utilizador com necessidades específicas juntamente com as recomendações dos profissionais de saúde.

## REFERÊNCIAS BIBLIOGRÁFICAS

Salawu FK, Umar JT, Olokoba AB (2011). Alzheimer's disease: a review of recente developments. *Ann Afr Med*, 10(2):73-9.

Pivi, G. A. K., Vieira, N. M. D. A., da Ponte, J. B., de Moraes, D. S. C., & Bertolucci, P. H. F. (2017). Nutritional management for Alzheimer's disease in all stages: mild, moderate, and severe. *Nutrire*, 42(1), 1-6.

Garrido, R. & Almeida, O. P. (1999). Distúrbios de comportamento em pacientes com demência: impacto na vida do cuidador. *Arquivos de Neuropsiquiatria*, 57(2B), 427-434.

Mohide, E. A. (1993). Informal care of community- dwelling patients with Alzheimer's disease: Focus on the family caregiver. *Neurology*, 43(Supl. 4), 16-19

Haley, W. (1997). The family caregiver's role in Alzheimer's disease. *Neurology*, 48(5), 25-29

Cao, C., Liu, F., Tan, H., Song, D., Shu, W., Li, W., ... & Xie, Z. (2018). Deep learning and its applications in biomedicine. *Genomics, proteomics & bioinformatics*, 16(1), 17-32

Trang Tran, T. N., Atas, M., Felfernig, A., & Stettinger, M. (2018). An overview of recommender systems in the healthy food domain. *Journal of Intelligent Information Systems*, 50(3), 501-526.

Murphy, K. P. (2012). *Machine learning: a probabilistic perspective*. MIT press

Brady, S., Siegel, G., Albers, R. W., & Price, D. L. (Eds.). (2005). *Basic neurochemistry: molecular, cellular and medical aspects*. Elsevier.

Yeo, G., & Thompson, D. G. (2014). *Ethnicity and dementias*. Taylor & Francis.

Cipriani, G., Dolciotti, C., Picchi, L., & Bonuccelli, U. (2011). Alzheimer and his disease: a brief history. *Neurological Sciences*, 32(2), 275-279.

Yiannopoulou, K. G., & Papageorgiou, S. G. (2013). Current and future treatments for Alzheimer's disease. *Therapeutic advances in neurological disorders*, 6(1), 19-33.

Petersson, S. D., & Philippou, E. (2016). Mediterranean diet, cognitive function, and dementia: a systematic review of the evidence. *Advances in Nutrition*, 7(5), 889-904.

Schmachtenberg, T., Monsees, J., Hoffmann, W., van den Berg, N., Stentzel, U., & Thyrian, J. R. (2020). Comparing national dementia plans and strategies in Europe—is there a focus of care for people with dementia from a migration background? *BMC Public Health*, 20(1), 1-10.

Singh, M., Kaur, M., Kukreja, H., Chugh, R., Silakari, O., & Singh, D. (2013). Acetylcholinesterase inhibitors as Alzheimer therapy: from nerve toxins to neuroprotection. *European journal of medicinal chemistry*, 70, 165-188.

- Fernández, S. S. M., Ivanauskas, T., & Ribeiro, S. M. L. (2017). Nutritional strategies in the management of Alzheimer disease: systematic review with network meta-analysis. *Journal of the American Medical Directors Association*, 18(10), 897-e13.
- Trevisan, K., Cristina-Pereira, R., Silva-Amaral, D., & Aversí-Ferreira, T. A. (2019). Theories of Aging and the Prevalence of Alzheimer's Disease. *BioMed research international*, 2019.
- Silva, J. L. M. M. D. (2018). A criação de um programa de atividades para reabilitar doentes com alzheimer (Doctoral dissertation).
- Correia, A., Filipe, J., Santos, A., & Graça, P. (2015). Nutrição e doença de Alzheimer.
- Grafstrom, M., Fratiglioni, L., Sandman, P. O. & Winblad, B. (1992). Health and social consequences for relatives of demented and non-demented elderly: A population study. *Journal Clinical of Epidemiology*, 45(8), 861-870.
- Burns, M. D, Nichols, L. O, Adams, J. M, Graney, M. J & Lummus, A. (2003). Primary Care Interventions for Dementia Caregivers: 2- Year Outcomes From the REACH Study. *The Gerontologist*, 43(4), 547-555.
- Paula JA, Roque FP, Araujo FS. Qualidade de vida em cuidadores de idosos portadores de demência de Alzheimer. *J bras psiquiatr*. 2008; 57:283-7.
- Pouyet, V., Giboreau, A., Benattar, L., & Cuvelier, G. (2014). Attractiveness and consumption of finger foods in elderly Alzheimer's disease patients. *Food Quality and Preference*, 34, 62-69.
- Ferreira, D. M. (2019). A Importância da Alimentação na Doença de Alzheimer (Doctoral dissertation, Universidade de Lisboa (Portugal)).
- Portela, F., Santos, M. F., Machado, J., Abelha, A., Rua, F., & Silva, Á. (2015). Real-time decision support using data mining to predict blood pressure critical events in intensive medicine patients. In *Ambient Intelligence for Health* (pp. 77-90). Springer International Publishing.
- Reis, R., Mendonça, A., Ferreira, D. L. A., Peixoto, H., & Machado, J. (2017). Business Intelligence for Nutrition Therapy. In *NextGeneration Mobile and Pervasive Healthcare Solutions* (pp. 203- 218). IGI Global.
- Karaboga, D. (2010). Artificial bee colony algorithm. *scholarpedia*, 5(3), 6915.
- Raut, M., Prabhu, K., Fatehpuria, R., Bangar, S., & Sahu, S. (2018). A personalized diet recommendation system using fuzzy ontology. *Int. J. Eng. Sci. Invention*, 7(3), 51-55.
- Leipold, N., Madenach, M., Schäfer, H., Lurz, M., Terzimehic, N., Groh, G., ... & Krcmar, H. (2018). Nutrilize a Personalized Nutrition Recommender System: an Enable Study. *HealthRecSys@ RecSys*, 2216, 24-29.
- Agapito, G., Calabrese, B., Guzzi, P. H., Cannataro, M., Simeoni, M., Caré, I., ... & Pujia, A. (2016, October). DIETOS: A recommender system for adaptive diet monitoring and personalized food suggestion. In *2016 IEEE 12th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)* (pp. 1-8). IEEE.

- Weber, I., & Achananuparp, P. (2016). Insights from machine-learned diet success prediction. In *Biocomputing 2016: Proceedings of the Pacific Symposium* (pp. 540-551).
- Iwendi, C., Khan, S., Anajemba, J. H., Bashir, A. K., & Noor, F. (2020). Realizing an efficient IoMT-assisted patient diet recommendation system through machine learning model. *IEEE Access*, 8, 28462-28474.
- Rostami, M., Oussalah, M. & Farrahi, V. A Novel Time-Aware Food Recommender-System Based on Deep Learning and Graph Clustering. *Ieee Access* 10, 52508–52524 (2022).
- Zhang, J., Li, M., Liu, W., Lauria, S. & Liu, X. Many-objective optimization meets recommendation systems: A food recommendation scenario. *Neurocomputing* 503, 109–117 (2022).
- Banerjee, A., Noor, A., Siddiqua, N. & Uddin, M. N. Food Recommendation using Machine Learning for Chronic Kidney Disease Patients. *2019 Int Conf Comput Commun Informatics Iccci 00*, 1–5 (2019).
- Alian, S., Li, J. & Pandey, V. A Personalized Recommendation System to Support Diabetes Self-Management for American Indians. *Ieee Access* 6, 73041–73051 (2018).
- Phanich, M., Pholkul, P. & Phimoltares, S. Food Recommendation System Using Clustering Analysis for Diabetic Patients. *2010 Int Conf Information Sci Appl* 1–8 (2010) doi:10.1109/icisa.2010.5480416.
- Naresh, A., Shaastry, M. S. S., Yadav, B. P., & Bhaskar, K. (2020). Understanding user
- Abirami, N., Kamalakannan, T., & Muthukumaravel, A. (2013). A Study on Analysis of Various Data Mining Classification Techniques on Healthcare Data. *International Journal of Emerging Technology and Advanced Engineering*, 3(7), 604-607.
- Eapen, A. G. (2004). *Application of Data mining in Medical Applications*.
- Fayyad, U., Piatetsky-Shapiro, G., & Smyth, P. (1996). From data mining to knowledge discovery in databases. *AI magazine*, 17(3), 37.
- Fürnkranz, J., & Hüllermeier, E. (2010). Preference learning and ranking by pairwise comparison. In *Preference learning* (pp. 65-82). Springer, Berlin, Heidelberg.
- Lison, P., 2015. An introduction to machine learning. *Language Technology Group (LTG)*, 1, 35.
- Portugal, I., Alencar, P. & Cowan, D., 2018. The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, 97, pp.205-227.
- Shalev-Shwartz, S. & Ben-David, S., 2014. *Understanding machine learning: From theory to algorithms*. Cambridge university press.
- Marsland, S., 2014. *Machine learning: an algorithmic perspective*. Chapman and Hall/CRC.

- Chin, E. L., Simmons, G., Bouzid, Y. Y., Kan, A., Burnett, D. J., Tagkopoulos, I., & Lemay, D. G. (2019). Nutrient estimation from 24-hour food recalls using machine learning and database mapping: a case study with lactose. *Nutrients*, *11*(12), 3045.
- Zellerbach, K., & Ruiz, C. (2019, November). Machine learning to predict overeating from macronutrient composition. In 2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM) (pp. 1268-1270). IEEE.
- Sathya, R. & Abraham, A., 2013. Comparison of supervised and unsupervised learning algorithms for pattern classification. *International Journal of Advanced Research in Artificial Intelligence*, *2*(2), pp.34-38.
- Peng, H., Wang, J., Pérez-Jiménez, M.J. & Riscos-Núñez, A., 2015. An unsupervised learning algorithm for membrane computing. *Information Sciences*, *304*, pp.80-91.
- Nagari, S. S., & Inayati, L. (2020). Implementation of Clustering Using K-Means Method To Determine Nutritional Status. *J. Biometrika dan Kependud*, *9*(1), 62.
- Sutton, R.S. & Barto, A.G., 2018. Reinforcement learning: An introduction. MIT press.
- Breiman, Leo. "Random forests." *Machine learning* 45.1 (2001): 5-32.
- Kingsford, C., & Salzberg, S. L. (2008). What are decision trees?. *Nature biotechnology*, *26*(9), 1011-1013.
- Yang, N., Li, T., & Song, J. (2007, October). Construction of decision trees based entropy and rough sets under tolerance relation. In *International Conference on Intelligent Systems and Knowledge Engineering 2007*. Atlantis Press.
- Genuer, R., Poggi, J. M., & Tuleau-Malot, C. (2010). Variable selection using random forests. *Pattern recognition letters*, *31*(14), 2225-2236.
- Doumpos, M., & Zopounidis, C. (2011). Preference disaggregation and statistical learning for multicriteria decision support: a review. *European Journal of Operational Research*, *209*(3), 203–214.
- Fürnkranz, J. & Hüllermeier, E. (2013). Preference learning and ranking. *Machine Learning*, *93*(2), 185-189.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, *521*(7553), 436-444.
- Haykin, S. (2009). *Neural networks and learning machines*. [sl] pearson upper saddle river. NJ, USA, 3.
- Gehring, J., Auli, M., Grangier, D., Yarats, D., & Dauphin, Y. N. (2017, July). Convolutional sequence to sequence learning. In *International conference on machine learning* (pp. 1243-1252). PMLR.
- Sak, H., Senior, A. W., & Beaufays, F. (2014). Long short-term memory recurrent neural network architectures for large scale acoustic modeling.
- Bengio, Y., Simard, P., & Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, *5*(2), 157-166.

- Horn, R. A. (1990, May). The hadamard product. In Proc. Symp. Appl. Math (Vol. 40, pp. 87-169).
- Ramchoun, H., Idrissi, M. A. J., Ghanou, Y., & Ettaouil, M. (2016). Multilayer Perceptron: Architecture Optimization and Training. *IJIMAI*, 4(1), 26-30.
- Abirami, S., & Chitra, P. (2020). Energy-efficient edge based real-time healthcare support system. In *Advances in computers* (Vol. 117, No. 1, pp. 339-368). Elsevier.
- Menzies, T., Kocaguneli, E., Turhan, B., Minku, L., & Peters, F. (2014). *Sharing data and models in software engineering*. Morgan Kaufmann.
- Mohanty, M. D., & Mohanty, M. N. (2022). Verbal sentiment analysis and detection using recurrent neural network. In *Advanced Data Mining Tools and Methods for Social Computing* (pp. 85-106). Academic Press.
- Hamdan, M. K. (2018). *VHDL auto-generation tool for optimized hardware acceleration of convolutional neural networks on FPGA (VGT)* (Doctoral dissertation, Iowa State University).
- Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv preprint arXiv:1710.05941.
- Goodfellow, I., Bengio, Y., & Courville, A. (2017). *Deep learning (adaptive computation and machine learning series)*. Cambridge Massachusetts, 321-359.
- Reed, R., & MarksII, R. J. (1999). *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press.
- Robbins, H., & Monro, S. (1951). A stochastic approximation method. *The annals of mathematical statistics*, 400-407.
- Duchi, J. C.; Hazan, E. & Singer, Y. (2011), 'Adaptive Subgradient Methods for Online Learning and Stochastic Optimization.', *J. Mach. Learn. Res.* 12 , 2121–2159.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
- You, Y., Li, J., Reddi, S., Hseu, J., Kumar, S., Bhojanapalli, S., ... & Hsieh, C. J. (2019). Large batch optimization for deep learning: Training bert in 76 minutes. arXiv preprint arXiv:1904.00962.
- Bharath, B., & Borkar, V. S. (1999). Stochastic approximation algorithms: Overview and recent trends. *Sadhana*, 24(4), 425-452.
- Tieleman, T., & Hinton, G. (2012). Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural networks for machine learning*, 4(2), 26-31.
- Misbahuddin, M. (2020, October). Estimating petrophysical properties of shale rock using conventional neural networks CNN. In *SPE Annual Technical Conference and Exhibition*. OnePetro.

- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural networks*, 61, 85- 117.
- Domingos, P. M. (2012). A few useful things to know about machine learning. *Commun. acm*, 55(10), 78-87
- Tassa, Y., Erez, T., & Todorov, E. (2012, outubro). Synthesis and stabilization of complex behaviors through online trajectory optimization. In 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (pp. 4906-4913). IEEE.
- Wang, L., Chu, F., & Xie, W. (2007). Accurate cancer classification using expressions of very few genes. *IEEE/ACM Transactions on computational biology and bioinformatics*, 4(1), 40-53.
- Akosa, J. (2017, April). Predictive accuracy: A misleading performance measure for highly imbalanced data. In *Proceedings of the SAS global forum* (Vol. 12, pp. 1-4).
- Wong, T. T., & Yeh, P. Y. (2019). Reliable accuracy estimates from k-fold cross validation. *IEEE Transactions on Knowledge and Data Engineering*, 32(8), 1586-1594.
- Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.
- Dougherty, G. (2012). *Pattern recognition and classification: an introduction*. Springer Science & Business Media.
- García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72, pp. 59-139). Cham, Switzerland: Springer International Publishing.
- Kuhn, M., & Johnson, K. (2013). *Applied predictive modeling* (Vol. 26, p. 13). New York: Springer.
- Evans, D. (2017). MyFitnessPal. *British Journal of Sports Medicine*, 51(14), 1101-1102.
- Zaidi, Nayyar. (2015). *Feature Engineering in Machine Learning*. 10.13140/RG.2.1.3564.3367.
- Han, J., Kamber, M., & Pei, J. (2011). *Data Mining: Concepts and Techniques* [Internet]. Waltham.