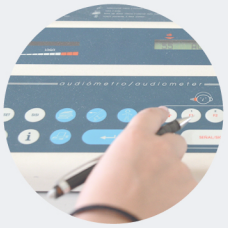
INSTITUTO SUPERIOR DE ENGENHARIA DO PORTO
MESTRADO EM ENGENHARIA BIOMÉDICA






Codificação Neural e Plasticidade Sinática em Modelos Animais de Doenças do Neurodesenvolvimento

GUILHERME LARANJEIRA MIRANDA
Novembro de 2022

P.PORTO

Politécnico do Porto
Instituto Superior de Engenharia do Porto

Codificação Neural e Plasticidade Sinática em Modelos Animais de Doenças do Neurodesenvolvimento

Guilherme Miranda

MEBIOM
Mestrado em Engenharia Biomédica

Departamento de Física
Instituto Superior de Engenharia do Porto

Novembro, 2022

Mestrado em Engenharia Biomédica.

Candidato: Guilherme Miranda, Nº 1170935, 1170935@isep.ipp.pt

Orientação Científica: Prof. Dr. Luís Coelho, lfc@isep.ipp.pt

Coorientação Científica: Dr. João Martins, jfmartins@fmed.uc.pt

Departamento de Física
Instituto Superior de Engenharia do Porto
Rua Dr. António Bernardino de Almeida, 431, 4200–072 Porto

Novembro, 2022

"It's not about better time management. It's about better life management" - Alexandra of The Productivity Zone

Agradecimentos

Este trabalho é o culminar de um percurso académico rico em conquistas, desilusões, amizades, felicidade, tristeza e sobretudo de muito crescimento e aprendizagem.

Termino o meu caminho no ISEP com a certeza de que foi a instituição certa para mim, repleta de profissionais de excelência a quem muito agradeço pelo que me ensinaram. Por isso, começo por agradecer a eles, em especial ao Professor Doutor Luís Coelho do ISEP e ao Professor Doutor João Martins da FMUC por terem aceite acompanhar-me e guiar-me nesta aventura que foi o desenvolvimento desta tese.

Agradeço aos meus amigos, família e namorada pela companhia nesta estrada sinuosa e cheia de obstáculos que é a vida. Ao Pedro, Rui, Inês, Nelson, Rosário, Fábio, Sílvia e Micaela, obrigado pelos bons momentos que partilhamos e pelo companheirismo. Ao meus pais e irmão, obrigado por sempre me ouvirem, acreditarem em mim e me darem força para seguir o caminho que escolhesse. À minha namorada Rita, por estar lá em todos os momentos, bons e maus, e por ter sempre a palavra que precisava de ouvir.

A todos vocês, obrigado, está feito!

Resumo

A neurofibromatose tipo 1 (NF1) é uma doença hereditária associada a perturbações do desenvolvimento neurológico sendo um dos impactos desta doença as alterações da neuroplasticidade, incluindo a do córtex visual. Surge então a necessidade de aferir a integridade da neuroplasticidade, através dos potenciais evocados visuais (VEP) sob a forma de potenciação da resposta seletiva a estímulos (SRP). A SRP está presente nos VEP quando o potencial obtido tem maior intensidade num estímulo familiar do que num estímulo novel.

Assim, o principal objetivo deste trabalho foi a classificação de estímulos novel e familiar presentes no eletroencefalograma (EEG), após cada estímulo. Foram então estudados métodos que conseguissem extrair do sinal EEG características diferenciadoras de cada tipo de estímulo que permitissem a sua classificação. Para atingir este objetivo foi então desenvolvido um programa com recurso a técnicas de *Machine Learning* que dado um período temporal após cada estímulo fosse capaz de classificar o estímulo que resultou na resposta presente no período temporal, como novel ou familiar.

Com o presente trabalho, foi possível concluir que é possível um algoritmo de *machine learning* classificar corretamente cada estímulo, uma vez que pelos resultados obtidos os valores da *accuracy* são bastante bons e o modelo apresenta robustez.

Palavras-Chave: Classificação, Eletroencefalograma, *Machine Learning*, Neuroplasticidade, Potenciação da Resposta Seletiva a Estímulos, Potenciais Evocados Visuais

Abstract

Neurofibromatosis type 1 (NF1) is an inherited disease associated with neurodevelopmental disorders, and one of the impacts of this disease is neuroplasticity changes, including that of the visual cortex. The need arises to assess the integrity of neuroplasticity, through visual evoked potentials (VEP) in the form of stimulus-selective response plasticity (SRP). SRP is present in VEP when the obtained potential has higher intensity in a familiar stimulus than in a novel stimulus.

Thus, the main goal of this work was the classification of novel and familiar stimuli present in the electroencephalogram (EEG) after each stimulus. Methods were then studied that could extract from the EEG signal differentiating characteristics of each stimulus type that would allow their classification. To achieve this goal a program using techniques of *Machine Learning* was then developed, that given a time period after each stimulus was able to classify the stimulus that resulted in the response present in the time period, as novel or familiar.

With this work, it was possible to conclude that it is possible for a *machine learning* algorithm to correctly classify each stimulus, since by the results obtained the values of *accuracy* are quite good and the model presents robustness.

Keywords: Classification, Electroencephalogram, *Machine Learning*, Neuroplasticity, Stimulus-Selective Response Potentiation, Visual Evoked Potentials

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Contextualização

O Eletroencefalograma (EEG) é um sinal biológico que representa a atividade elétrica das células cerebrais em localizações diferentes do cérebro. Um EEG pode ser adquirido de maneira invasiva, utilizando elétrodos intracranianos ou de maneira não invasiva com o recurso a elétrodos colocados na superfície da cabeça [1]. Os potenciais evocados visuais (VEP) são um tipo de sinal que ocorre em resposta a um estímulo visual e podem estar presentes no EEG [2]. Os VEPs são uma medição quantitativa da integridade do sistema visual, por exemplo, a latência reflete a velocidade da propagação do sinal pelos neurónios e é uma medição fiável do nível de mielinização do nervo óptico enquanto que a amplitude reflete o nível de danos dos axónios [3].

A neuroplasticidade, inclui, a plasticidade no desenvolvimento, aprendizagem e memória, plasticidade compensatória e reparação do cérebro adulto. A neuroplasticidade pode ser afetada por condições patológicas como tumor cerebral, epilepsia e doenças neurodegenerativas como o Alzheimer. A nível celular, a plasticidade sináptica é responsável por manter as conexões neuronais e a integridade das estruturas cerebrais [4]. Quando a plasticidade é impactada por alguma condição, a utilização de VEPs pode ser de grande importância para diagnosticar precocemente uma doença que cause uma degradação cerebral.

Assim, é importante perceber as alterações da neuroplasticidade em doenças do neurodesenvolvimento uma vez que a retificação destas alterações poderá trazer

benefícios para o doente. Tal situação leva-nos à motivação do presente trabalho que passamos a descrever na secção seguinte.

1.2 Motivação e Objetivos

O diagnóstico das alterações da neuroplasticidade causada por doenças neurodegenerativas pode ter um grande impacto na qualidade da vida dos doentes. Para tal, é importante que existam ferramentas capazes de realizar esta avaliação com a maior fiabilidade e rapidez possível, e que consigam encontrar características não visíveis ao olho humano. Com os recentes avanços no campo da *machine learning*, espera-se que resultados interessantes possam ser obtidos. Assim, o grande objetivo deste trabalho é perceber a possibilidade da utilização de técnicas de *machine learning* para a correta classificação do tipo de estímulo presente no eletroencefalograma em estudo. Pretende-se desenvolver um programa capaz de realizar o processamento de sinal, a extração de *features* representativas de cada estímulo e a sua classificação. Pretende-se por isso estudar vários métodos para cada etapa do programa, de maneira a encontrar o melhor modelo possível para resolver o problema em estudo.

1.3 Organização do Relatório

O presente documento encontra-se estruturado em seis capítulos. No primeiro capítulo é feita uma contextualização do estudo efetuado e são apresentados a motivação e os seus objetivos. No Capítulo 2, são introduzidos alguns conceitos fundamentais associados à memória, aos potenciais evocados e à sua relação com a potenciação da resposta seletiva a estímulos. Já no terceiro capítulo, é descrito o estado da arte relativo aos métodos atualmente utilizados num problema de *machine learning* e são indicados os métodos adotados por diferentes autores na resolução de problemas semelhantes ao problema em estudo no presente documento. Os materiais e métodos adotados na realização deste estudo encontram-se detalhados no Capítulo 4 e no Capítulo 5 apresentam-se os resultados do pré-processamento de dados, da redução de *features*, uma avaliação do desempenho para diferentes conjuntos de dado, métodos de normalização e da *cross validation*. Para finalizar, no último capítulo todo o trabalho é posto em perspetiva e são tiradas conclusões do que foi desenvolvido.

Capítulo 2

Conceitos Fundamentais

2.1 Cortéx Cerebral e a Memória

Entre os conhecimentos atribuídos a Donald Olding Hebb, estão as noções de que as memórias de experiências sensoriais são armazenadas por alterações sinápticas, e que estas alterações ocorrem nas mesmas regiões do cérebro onde é processada a informação sensorial. Ou seja, as memórias de experiências visuais seriam armazenadas na região do córtex visual, as experiências auditivas seriam armazenadas no córtex auditivo, e assim por diante. Dentro de cada uma destas regiões do córtex, aquilo a que nos referimos como memória de uma experiência sensorial será o resultado da permanente modificação das sinapses entre os neurónios corticais que foram ativados por essa experiência.[5]

Foi descoberto que a memória inclui três categorias fundamentais: sensorial, a curto-prazo, e a longo-prazo (Figura 2.1)[6]. Cada um destes tipos de memória tem diferentes atributos: A **memória sensorial** refere-se à retenção de informação proveniente dos sentidos e é constituída pela memória ecóica, icónica e háptica. A memória icónica retém informação obtida através de um estímulo visual, a memória ecóica retém informação obtida através de um estímulo auditivo e a háptica através do toque. A **memória a curto-prazo** refere-se à informação processada num curto período sendo a memória de trabalho responsável por executar este processamento. A memória de trabalho é constituída por quatro elementos que processam informação: o executivo central (controlo da atenção), o *visuospatial sketchpad* (cria e

mantém uma representação visio-espacial), o *phonological buffer* (armazena e consolida novas palavras), e o *buffer* episódico (armazena e integra informação de diferentes fontes). A **memória a longo-prazo** permite armazenar informação durante longos períodos. Esta informação pode ser recuperada conscientemente (memória explícita) ou inconscientemente (memória implícita). A memória explícita é constituída pela memória episódica (eventos relacionados com o tempo) e pela memória semântica (conceitos e significados). A memória implícita tem, por sua vez, memória processual (capacidades motoras e executivas), memória associativa (clássica e condicionamento operante), memória não-associativa (sensibilização e habituação), e *priming* (um estímulo primário que influencia um secundário) [7].

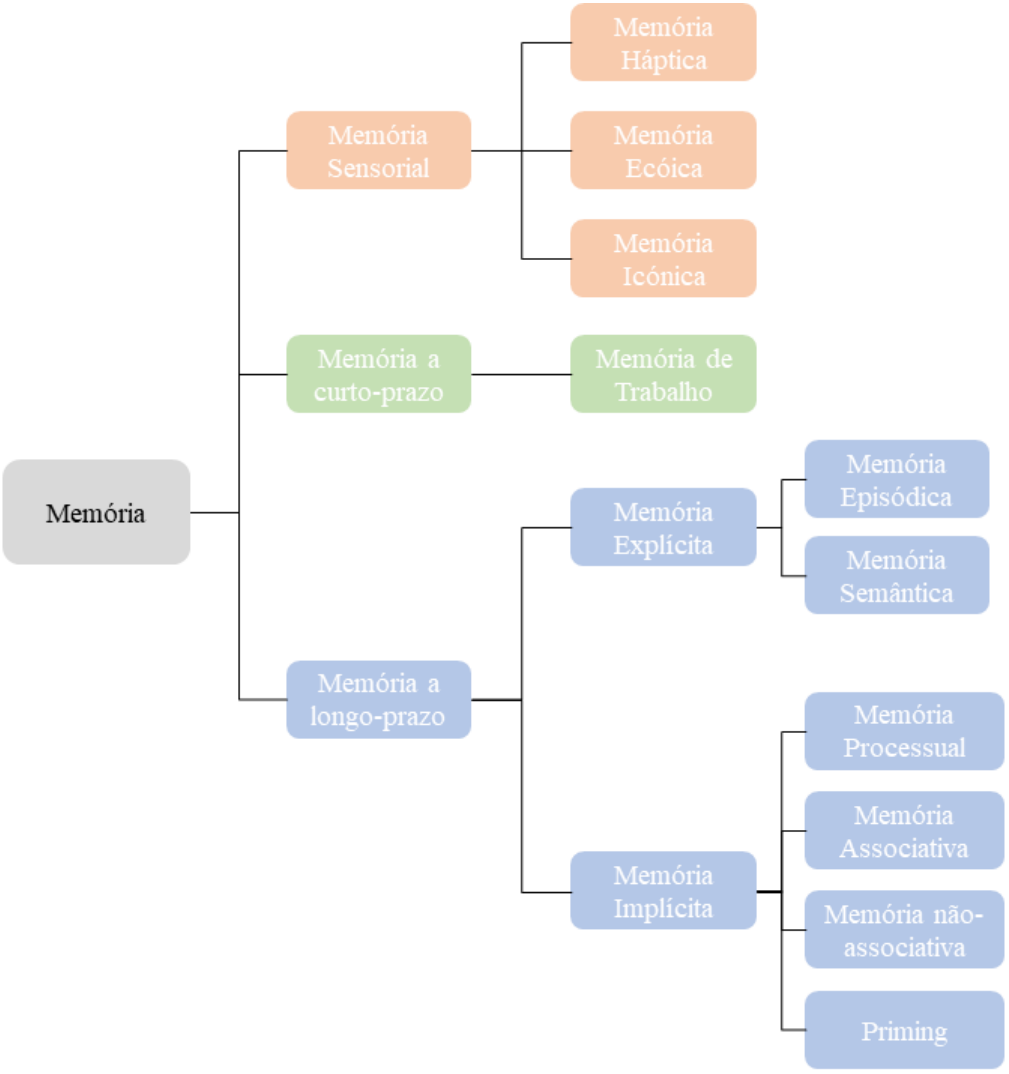


Figura 2.1: Classificação da Memória. Adaptado de [7].

2.1.1 Córtex Visual

O córtex visual é a região cortical primária do cérebro que recebe, integra e processa a informação visual com origem nas retinas. Encontra-se no lóbulo occipital do córtex cerebral primário, localizado na região mais posterior do cérebro. O córtex visual divide-se em cinco áreas diferentes (V1 a V5) com base na sua função e estrutura [8]. Correspondendo V1 à área 17 de Brodmann, V2 à área 18 e V3 à 19 (Figura 2.2).

A principal função do córtex visual é o processamento da informação visual. A teoria é que, à medida que a informação visual é transmitida, cada área cortical subsequente é mais especializada do que a anterior. Os neurónios no córtex visual respondem regularmente a estímulos dentro de um campo recetivo fixo, a área do campo visual a que respondem, e os neurónios em cada área visual respondem a diferentes tipos de estímulos. As células simples, que se encontram principalmente em V1, respondem a tipos específicos de estímulos visuais, tais como a orientação das arestas e linhas [8].

V1, também conhecido como córtex visual primário, é a primeira das regiões corticais a receber e processar informação, assim como, a porção mais bem compreendida do córtex visual. V1 está dividido em seis camadas distintas, cada uma compreendendo diferentes tipos de células e funções, a camada 4 é o local que recebe a informação das retinas. A camada 4 é também a camada que tem o maior concentração de células simples [8].

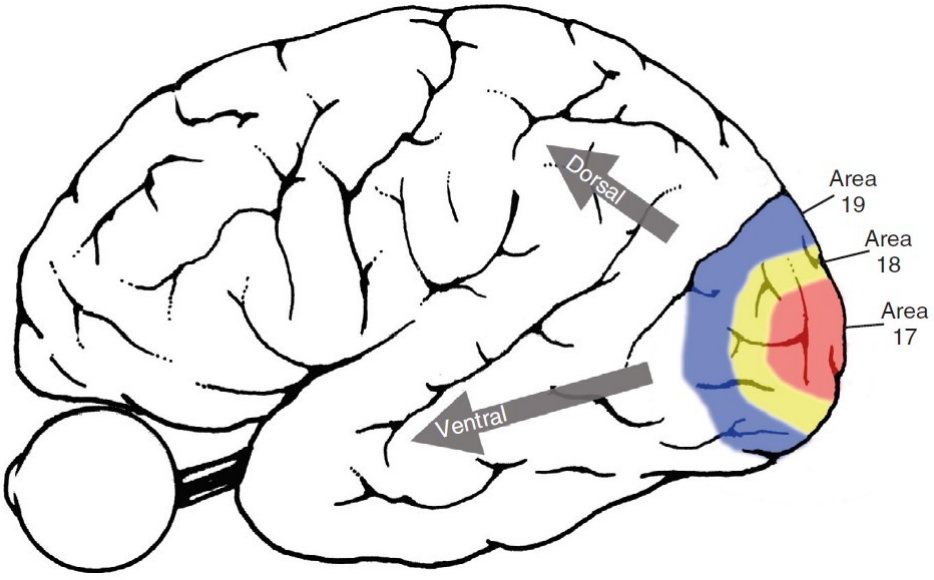


Figura 2.2: Áreas corticais visuais, incluindo a Área 17 de Brodmann (V1), a Área 18 de Brodmann (V2) e a Área 19 de Brodmann (V3) [9].

2.2 Potenciais Evocados

Os potenciais evocados (EPs) representam uma atividade relacionada a eventos externos que é expresso através da resposta elétrica do cérebro ou do tronco encefálico a vários tipos de estimulação sensorial dos tecidos nervosos ([10], [11]). O termo potenciais evocados foi originalmente utilizado como referência a respostas causadas por estímulos sensoriais [10]. Os EPs sensoriais podem ser subdivididos em 3 categorias relativamente à sua latência, curta (menos de 10 ms), média (10-50 ms) e longa (mais de 50 ms). Os EPs de latência curta são normalmente robustos, não sendo afetados pelo nível de atenção do sujeito e pela anestesia. Os EPs de latência média apresentam diferenças entre sujeitos acordados ou adormecidos e são bastante afetados pela anestesia de nível cirúrgico. Os EPs de longa latência são os que melhor exibem o estado de alerta e os efeitos da anestesia [10]. Os EPs normalmente manifestam-se como uma forma de onda transitória cuja morfologia depende do tipo e da força do estímulo [11]. O estado mental do sujeito, ou seja, o seu nível de atenção, vigília e expectativa, tem também influencia na morfologia da onda [11]. Por convenção, os componentes de pico dos EPs são identificados pelas letras P (amplitude positiva) e N (amplitude negativa). O número adjacente à letra reflete a latência em milissegundos desde o estímulo. Por exemplo, P300 significa que ocorreu um pico positivo 300 ms após o estímulo [11].

Os EPs sensoriais podem ser registados após a estimulação em qualquer modalidade sensorial, mas os EPs somatossensoriais (SEPs), os EPs auditivos (AEPs) e os EPs visuais (VEPs), são os mais utilizados para fins de diagnóstico e testes clínicos, monitorização intra-operatório (IOM) e investigação neuro-fisiológica [10].

2.2.1 Potencias Evocados Somatossensoriais

Os EPs somatossensoriais (Figura 2.3) são provocados pela estimulação elétrica de nervos periféricos. Os componentes do sistema nervoso com origem nos nervos periféricos, são analisados e por isso, os SEPs podem detetar anomalias que afetam as vias somatossensoriais desde o local do estímulo ao córtex somatossensorial primário. Assim, os SEPs podem identificar disfunções do nervo periférico, da medula espinhal, tronco cerebral, tálamo, substância branca e do córtex somatossensorial primário [10].

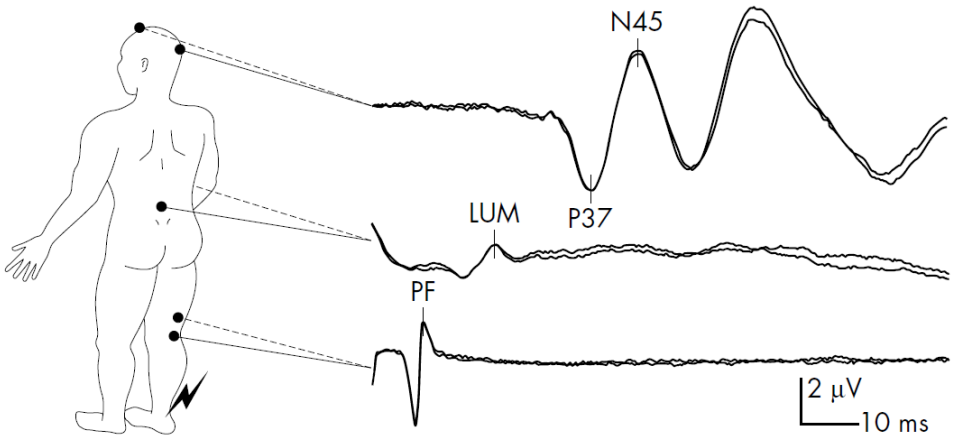


Figura 2.3: Potenciais evocados somatossensoriais normais após estimulação do nervo tibial posterior [12].

2.2.2 Potencias Evocados Auditivos

Os EPs auditivos (Figura 2.4) são causados pelo presença de um curto estímulo auditivo como cliques ou pips. Os AEPs com mais utilidade clínica são os de latência curta devido a serem de fácil registo e à sua alta consistência entre sujeitos. Os AEPs podem ser utilizados para aferir o funcionamento dos ouvidos, nervos auditivos e das vias auditivas [10]. Uma outra aplicabilidade é a IOM durante uma cirurgia espinhal, sendo que não ocorrendo alteração da forma de onda durante a cirurgia indica que não houve mudanças no funcionamento neurológico do paciente [11].

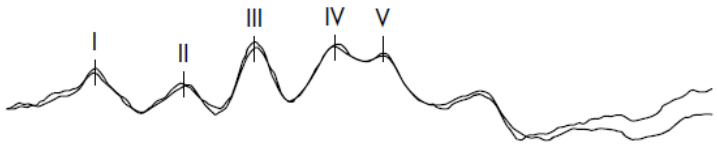


Figura 2.4: Potencial evocado auditivo normal [12].

2.2.3 Potencias Evocados Visuais

Os EPs visuais (Figura 2.5) são normalmente resultado da estimulação por *flashes* ou por inversão de padrões de alto contraste. Os VEPs causados por inversão de padrões apresentam menos variação entre sujeitos do que os por *flashes* e por isso têm maior sensibilidade no diagnóstico de irregularidades do sistema visual. Os VEPs são EPs de latência longa [12] e por isso, altamente afetados pela anestesia.

Os VEPs podem ser utilizados no meio clínico para quantificar a integridade funcional do sistema visual, desde a retina, passando pelos nervos ópticos, tratos ópticos até ao tálamo e do córtex visual e occipital. Podem ainda ser uma ferramenta

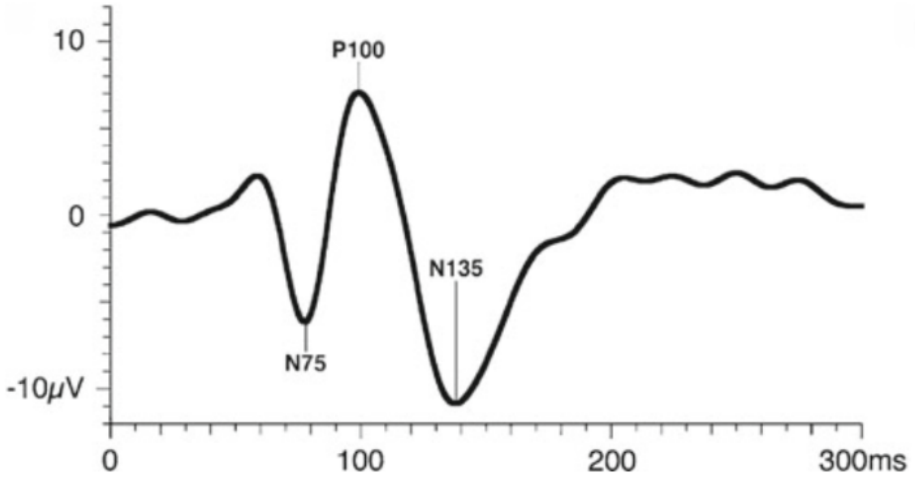


Figura 2.5: Potencial evocado visual normal causado por inversão de padrões [13].

de diagnóstico precoce de certas patologias. Os VEPs são melhores a quantificar o estado das vias visuais do que técnicas de imagem como a ressonância magnética (RM). Qualquer anormalidade que tenha impacto num dos componentes do sistema visual enunciados, pode afetar o VEP. De um modo geral, a desmielinização causa um atraso nos picos do VEP. A compressão das vias óticas, reduz a amplitude dos VEP. E a disfunção macular retinal pode atrasar o VEP e reduzir a sua amplitude [14].

Num paciente com esclerose múltipla (EM) em estado inicial incluindo neurite ótica, o comum é apenas um nervo óptico apresentar VEPs normais. O outro nervo, afetado com neurite ótica, apresenta um P100 com atraso. Os pacientes com EM podem vir a desenvolver neurite ótica no outro nervo ainda saudável. À medida que a doença progride os VEPs destes doentes tornam-se progressivamente mais lentos, eventualmente diminuindo a amplitude à medida que a desmielinização aumenta [14]. Um outro exemplo de desmielinização são os pacientes com a doença de Parkinson, estando provado o aumento da latência do N75, P100 e N145 mas sem alterações na amplitude [15].

Crianças com neurofibromatose do tipo I (NF1) são suscetíveis de desenvolver gliomas do nervo óptico. Neste cenário, o VEP é uma ferramenta com melhor custo benefício para acompanhar o progresso da patologia do que uma RM. De facto, é observável em todas as crianças com NF1 um atraso dos VEPs, mesmo sem alterações morfológicas das vias visuais visíveis numa RM [14]. Os VEPs foram demonstrados como sendo uma ferramenta viável no diagnóstico precoce de glaucoma, ocorrendo um aumento significativo da latência do N145 nos sujeitos afetados pela doença [16]. Ainda, os VEPs podem ser úteis a quantificar a progressão da hipoplasia do nervo óptico, atrofia ótica e neuropatia ótica [14].

Alguns medicamentos são tóxicos para o nervo óptico, podendo alterar o seu correto funcionamento. Medicamentos utilizados no tratamento da taquicardia ventricular ou da fibrilhação ventricular, como a amiodarone, podem causar neuropatia ótica isquémica. Os VEPs destes pacientes apresentam uma latência dos picos mais elevada [14].

Os VEPs podem ser ainda utilizados para avaliar a qualidade da visão binocular após a cirurgia refrativa LASIK [17]. Um aumento da latência do P100 foi associado com doentes que sofrem de hipertiroidismo [18] e doentes com retinopatia diabética apresentaram um aumento da latência do P100 e N75 [19].

2.3 Potenciação da Resposta Seletiva a Estímulos

A potenciação da resposta seletiva a estímulos (SRP) é uma plasticidade de longa duração, dependente da experiência, que ocorre no V1 de murganhos (Figura 2.6). Em animais conscientes e com a cabeça fixa, os potenciais evocados visuais (VEPs) causados por estímulos de grelha sinusoidal de inversão de fase, registados a partir da camada 4 do V1, sofrem um aumento gradual mas significativo da amplitude durante um período em que ocorre a apresentação repetida do mesmo estímulo [20]. O VEP reflete a soma de correntes sinápticas estereotipada em função da latência desde o início do estímulo e a profundidade no córtex [21]. Após a sua expressão, o SRP dura pelo menos várias semanas, e ainda não existem indicações de que se degrada com o tempo, mesmo na ausência da apresentação contínua dos estímulos [20]. Existe evidência de que o SRP é seletivo para um leque de propriedades dos estímulos e não se transfere para estímulos *novel*. Mesmo uma ligeira mudança na orientação do estímulo (de apenas 5 graus) irá provocar um VEP significativamente menor, assim como mudanças no contraste do estímulo e na frequência espacial [21]. Ainda, se apenas um olho for exposto ao estímulo, o SRP pode ser induzido através deste olho sem transferência para o outro olho [20].

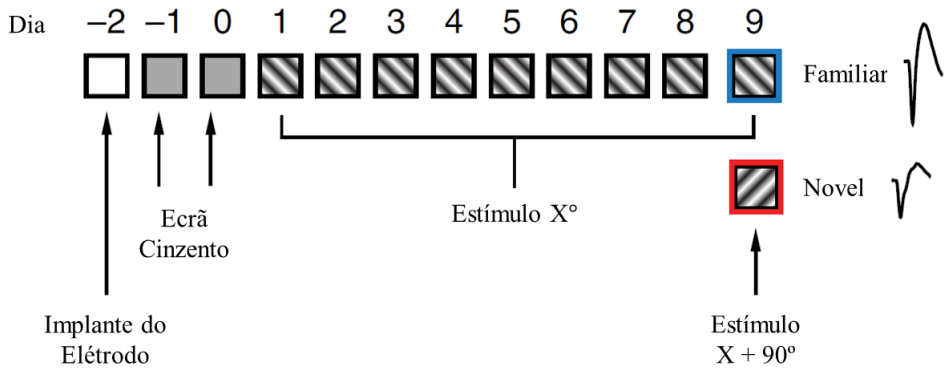


Figura 2.6: Exemplo de um processo de treino de SRP em murganhos. Adaptado de [22].

Capítulo 3

Estado da Arte

Antes de podermos aplicar um algoritmo de *machine learning* ao problema em causa, é necessário traduzi-lo para um problema de ciência dos dados primeiro. Este processo pode ser visto como uma *pipeline* de *machine learning* (Figura 3.1) que consiste em várias etapas [23]:

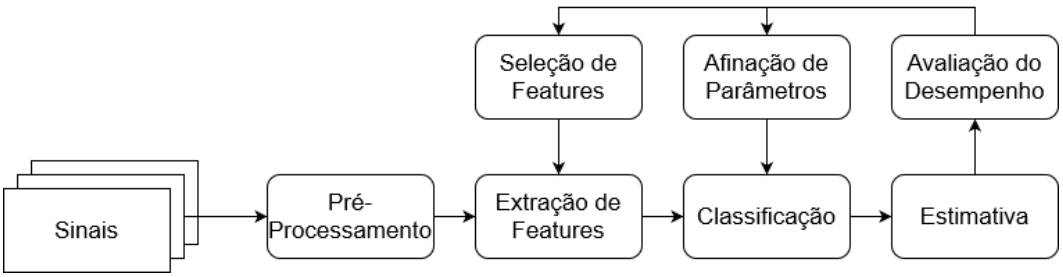


Figura 3.1: Modelo básico de um sistema de *machine learning*. Adaptado de [23] e [24]

1. **Registo de dados:** Etapa onde os dados necessários para resolver o problema são identificados e obtidos. Sendo o resultado dados *raw* [23].

2. **Preparação dos dados:** Após os dados *raw* serem recolhidos, são preprocessados para serem limpos. São depois transformados num novo espaço que representa os padrões dos dados, chamado vetor de *features*. Durante o processo de extração de features é normal se começar com um grande vetor de

features e eventualmente serem escolhidas as mais representativas durante o processo de seleção de *features* [23].

3. **Treino do modelo:** Após termos as *features* bem definidas, é necessário treinar um algoritmo de *machine learning* para obter um modelo. Após ser escolhido o algoritmo e os seus hiperparâmetros terem sido afinados, o modelo está pronto a ser utilizado num cenário real [23].

Estas etapas serão descritas com mais detalhe nas próximas secções.

3.1 Metodologias de Pré-Processamento

Tal como discutido anteriormente no início deste capítulo, o primeiro passo de um modelo de *machine learning* é o pré-processamento. Como o problema tem como dados de entrada os eletroencefalogramas (EEG) será no pré-processamento deste tipo de sinal que se irá focar esta secção.

Um dos maiores desafios de estudar sinais cerebrais é que as gravações são contaminadas por ruído e artefactos. Podem ser ruído do ambiente, ruído instrumental ou sinais provenientes de partes do corpo que não sejam o objeto em estudo. A presença de ruído pode tornar o sinal em estudo invisível ou interferir com a análise. Contudo, quando este ruído está presente numa região do espetro diferente da do sinal em estudo, é possível melhorar a relação entre o ruído e sinal (SNR) através da aplicação de um filtro [25]. O SNR é a razão entre a potência do sinal desejado e o ruído presente no sinal de entrada e é expressado pela Equação 3.1. Por observação da equação deduzimos que um SNR mais elevado é desejável uma vez que significa uma maior redução do ruído [26].

$$SNR_{dB} = 20.log_{10}\frac{V_{signal}}{V_{noise}} \tag{3.1}$$

Relembrando que o sinal que se pretende extrair do EEG são os VEPs, é de interesse saber que o VEP tem um SNR de -5 dB, o que é bastante pequeno [15]. Este valor comprova o problema do ruído discutido e a importância de filtrarmos o sinal de entrada, assim, serão agora apresentadas técnicas de filtragem e exemplos da literatura.

O melhoramento do SNR obtido através da aplicação de um filtro é uma mais valia, contudo os efeitos de um filtro também afetam o sinal em estudo. Existem 4 tipos comuns de filtros, o passa-baixo, passa-alto, passa-banda e *notch* [25]:

1. **Passa-baixo:** Quando os fenómenos de interesse correspondem a frequências baixas, podemos categorizar as componentes de alta frequência como irrelevantes e serão atenuadas pelo filtro passa-baixo. Este filtro tem um efeito suavizador [25].

2. **Passa-alto:** Os EEG são suscetíveis a mudanças na componente contínua sobre as quais ocorrem os sinais de interesse com maior frequência. Por exemplo, na implantação de elétrodos para a gravação do EEG podem ocorrer potenciais de junção entre o elétrodo e o tecido cerebral. Os filtros passa-alto são por isso a ferramenta utilizada para remover estas componentes de baixa frequência [25].

3. **Passa-banda:** Consiste na combinação de um filtro passa-baixo com um passa-alto. É útil na neurociência isolar bandas específicas do EEG, como a alfa, beta, teta, entre outras, sendo nessas situações particularmente útil este filtro [25].

4. ***Notch*:** O *notch* é utilizado para remover frequências específicas do sinal, como as do ruído da rede elétrica (50 ou 60 Hz) [25].

O uso de filtros pode trazer alterações indesejadas ao sinal, algumas graves outras nem tanto. Uma alteração é a perda de informação útil que é filtrada juntamente com o ruído. Menos óbvia é a distorção das características temporais do sinal em estudo, os picos podem aparecer suavizados e artefactos podem surgir. Mais graves são o enfraquecimento da relação causal entre um estímulo externo e o sinal, e o atraso temporal. É por isso importante ter atenção no momento de implementar um filtro para não serem removidas componentes do sinal essenciais ao que se pretende estudar [25].

Os filtros aplicados por diferentes autores encontram-se na Tabela 3.1 para melhor visualização. Como se pode observar não existe um consenso na frequência de corte, embora existe no tipo de filtro, sempre um tipo passa-banda. Dos autores citados, é importante referir o trabalho de Oikonomou et. al [1] e Montalvo-Aguillar et. al [27]. Os primeiros testaram vários filtros no classificador que estavam a desenvolver e obtiveram a melhor *accuracy* para o filtro Elíptico 5-48 Hz. Montalvo-Aguillar et. al [27] testarem os filtros *Butterworth*, *Chebyshev* e Elíptico, e constataram que o *Butterworth* e Elíptico tiveram resultados semelhantes enquanto que o Chebyshev foi o pior dos três.

Tabela 3.1: Tipos de filtros e respetivas frequências de corte utilizados por diversos autores.

Tipo de Filtro	**Frequência de Corte**	**Autores**
Butterworth	5-25 Hz	[28]
Butterworth	5-40 Hz	[28]
Elíptico	5-48 Hz	[24][1]
Passa-banda	0.1-100 Hz	[29][30]
Passa-banda	1-100 Hz	[31]
Passa-banda	1-90 Hz	[32]
Passa-banda	0.1-80 Hz	[33]
Butterworth, *Chebyshev* e Elíptico	4-40 Hz	[27]

Um outro método de pré-processamento utilizado para a extração dos VEPs é o *signal averaging* [34][15][35]. Nesta técnica, é feita a média de um conjunto de segmentos de EEG, denominados *epochs*, onde se obtém uma onda onde características consistentes do sinal (os VEPs) são preservadas e as características que variam (o ruído) são atenuadas. Dependendo do tamanho do VEP e do SRP, o número de epochs necessárias para aplicar esta técnica pode oscilar entre 10 a mais de 1000 [36]. Assim, tendo um sinal x, repetido N vezes e sendo k um ponto em x, podemos aplicar a Fórmula 3.2 para realizar *signal averaging*:

$$\overline{x(k)_N} = \frac{1}{N}\sum_{j=1}^{N} x_j(k) \tag{3.2}$$

As técnicas de pré-processamento apresentadas, embora tenham um resultado idêntico (extrair o sinal em estudo pretendido) não podem ser aplicadas a qualquer sistema. Isto é, a primeira técnica, a filtragem, é melhor empregada num sistema em que se pretenda analisar de maneira independente cada *epoch* enquanto que o *signal averaging*, devido a ser necessário a utilização de vários *epochs* não poderá ser utilizado no mesmo tipo de sistema.

3.2 Metodologias de Extração de *Features*

Após o pré-processamento do sinal, segue-se a extração de *features*. Uma *feature* é uma representação alternativa do sinal e normalmente de menor dimensionalidade. Esta etapa de um algoritmo de *machine learning* é de grande importância num

sistema de reconhecimento de padrões, já que este processo determina vários descritores do sinal. É também importante reter que a escolha das *features* a utilizar tem uma elevada influência na precisão e no custo computacional do modelo [1].

Existem duas grandes categorias para a extração de *features*: Não transformadas (potência, amplitude, energia, etc) e transformadas (espectro de frequência e amplitude, etc). De um modo geral, a categoria das transformadas é a mais importante para o processamento de sinal biomédico. A ideia das *features* transformadas é encontrar uma transformação (Método de Welch, Transformada de Fourier, etc) dos dados pré-processados, que melhor representa a informação relevante para a etapa da classificação [1].

Podem ser aplicadas diversas técnicas para a extração de *features* relevantes para a análise dos dados e segue-se agora uma breve descrição dos métodos aplicados na literatura atual:

Embora existam autores que utilizem *features* não transformadas [29], [37], [38], apenas [29] utilizou unicamente *features* não transformadas (amplitude e fase), enquanto que [37] e [38] utilizaram uma combinação de não transformadas e transformadas. Em contra-partida, parece haver uma quantidade muito superior de autores que recorrem ás *features* transformadas [37], [38], [39], [24], [40], [1], [41] e [30].

Começando pelos autores que utilizaram a combinação de categorias: [37] utilizou como *features* não transformadas o tempo e a sua derivada (declive), assim como a amplitude e latência do pico. Como *features* transformadas recorreu à transformada Wavelet, obtendo como *features* pontuais o seu valor máximo, frequência e timing, e os valores do espectograma como *feature* contínua. [38] utilizou 4 *features* diferentes, o desvio padrão, entropia, potência e energia tanto dos sinais no domínio do tempo como no domínio da frequência através da transformada rápida de Fourier (FFT).

Em relação aos autores que apenas utilizaram *features* transformadas, temos por exemplo [39] que utilizou os valores obtidos através da FFT. [24] e [1], utilizaram os valores do método de Welch e tendo sido provado por [40] como um método eficaz para a extração de *features* para problemas relacionados com VEPs. Já [41] utilizou também o método de Welch mas na forma dos valores da média e desvio padrão. Por último, [30] desenvolveu um novo método de extração de *features* que recorre ao uso da transformada Wavelet.

3.3 Metodologias de Seleção de *Features*

Os métodos de seleção de *features* destinam-se a reduzir o número de variáveis de entrada para aquelas que se acredita serem mais úteis para um algoritmo de *machine learning* prever a variável de destino. Alguns algoritmos possuem um grande número de variáveis que podem atrasar o desenvolvimento e treino dos modelos e

requerem uma grande quantidade de memória. Além disso, o desempenho de alguns modelos pode diminuir ao incluir *features* que não são relevantes para o classificador. Existem duas categorias de métodos de seleção de *features*: os supervisionados e não supervisionados. O que os distingue é se as *features* são selecionadas com base na variável de destino ou não. Quando a variável de destino é ignorada, o método é não supervisionado e o objetivo é remover *features* com elevada correlação com outras *features*. Os métodos supervisionados utilizam a variável de destino, e são utilizados para quantificar a importância das *features* [42].

Podemos ainda dividir os métodos de seleção de *features* em 3 outras categorias (Figura 3.2): os métodos intrínsecos, os métodos baseados em filtro e os métodos *wrapper*. Os métodos intrínsecos são aqueles cuja seleção de *features* já é realizada de maneira automática pelo algoritmo de *machine learning*. Exemplos destes algoritmos são árvores de decisão, *Multivariate adaptive regression spline* (MARS) e Lasso. [43] Os métodos baseados em filtro avaliam a relevância das *features* fora dos modelos e de seguida modelam apenas as *features* que cumprem algum critério. Por exemplo, num problema de classificação, cada *feature* poderia ser avaliada individualmente para se perceber se existe uma relação entre si e as classe na saída. Apenas as *features* com uma boa e importante relação seriam utilizadas no modelo. Por último, os métodos *wrapper* criam vários modelos com diferentes grupos de *features* e selecionam as *features* que resultam na melhor no melhor desempenho do modelo de acordo com uma métrica de avaliação definida [42].

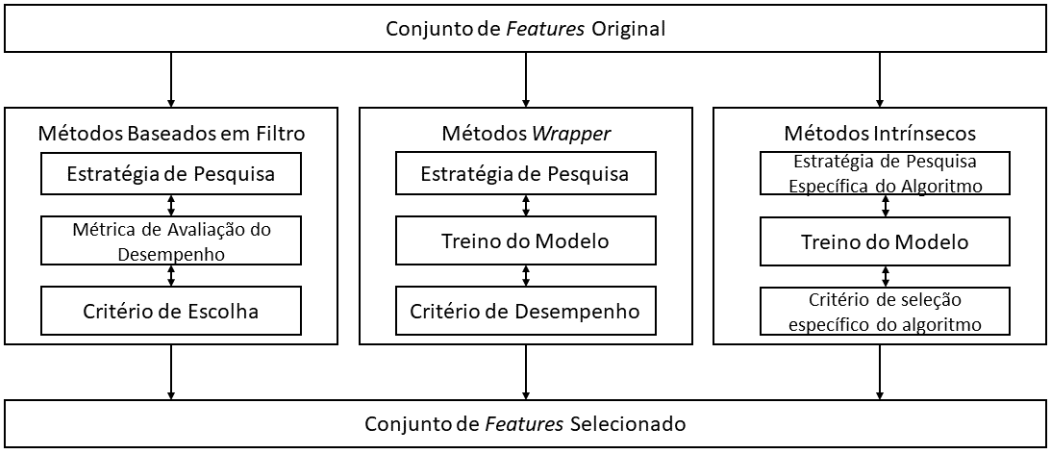


Figura 3.2: Seleção de *features* utilizando métodos com base em filtro, métodos *wrapper* e métodos intrínsecos. Adaptado de [44].

Existem 5 principais métodos de seleção de *features*:

1. **Correlação de *Pearson*:** Este é um método baseado em filtro. É verificado o valor absoluto da correlação de *Pearson* entre a variável de saída e as *features* do dataset. Guarda-se as melhores n *features* baseadas neste critério [45].

2. **Chi-Quadrado:** Outro método baseado em filtro. É calculado o chi-quadrado entre a a variável de saída e cada *feature* e são selecionadas apenas o número desejado de *features* com o melhor valor do chi-quadrado [45].

3. **Eliminação de *Features* Recursiva (RFE):** Este é um método *wrapper*. O objetivo do RFE é selecionar *features* considerando recursivamente conjuntos cada vez menores de *features*. O modelo é inicialmente treinado com o conjunto total das *features* e a importância de cada *feature* é obtida através de um coeficiente. As *features* com menor importância são por isso removidas do conjunto atual de *features*. Este processo repete-se até se obter o número desejado de *features* [45].

4. **Lasso:** Este é um método intrínseco. O Lasso força muitas *features* a terem uma importância de zero [45].

5. **Árvores de decisão:** Este é outro método de seleção de *features* intrínseco. É calculado a importância de cada *feature* utilizando as impurezas do nó em cada árvore de decisão. Nas *Random Forest*, a importância final de cada *feature* é a média das importâncias de cada árvore de decisão [45].

Quando se olha para a literatura relacionada com o problema em causa, vemos que nem todos os autores utilizam métodos de seleção de *features* e os métodos aplicados diferem. [37] testou vários métodos, *minimum redundancy maximum relevance* (MRMR), Chi-Quadrado, relief e seleção aleatória de features, tendo obtido melhores resultados com o MRMR. [38] aplicou 3 algoritmos Principal Component Analysis (PCA), Independent Component Analysis (ICA) e PCA+ICA, tendo obtido no geral um melhoramento do desempenho em relação à não utilização de nenhum método. [40] testou diferentes métodos, dois baseados em filtro, Correlação de Pearson e David Bouldin, e um em wrappers, tendo concluído que o uso de técnicas de seleção de *features* tem um impacto muito significativo no desempenho do modelo. [1] avaliou o impacto de doze métodos e concluiu que o Singular Value Decomposition (SVD) foi o método que resultou num maior aumento do desempenho do modelo. [30] utilizou um método baseado em informação mútua para selecionar os melhores coeficientes da transformada *Wavelet*. Por último, [24] aplicou seleção de *features* com base nos valores dos coeficientes de correlação entre *features*, eliminando as *features* com um alto coeficiente de correlação.

Assim, conclui-se que a utilização de um método de seleção de *features* pode trazer melhorias ao desempenho do modelo, contudo, o melhor método a utilizar no processo de seleção de *features* só é conseguido por meio de experimentação.

3.4 Metodologias de Classificação

Tendo concluído as etapas anteriormente descritas, de acordo com o necessário, os dados que representam o nosso problema estão agora prontos para serem utilizados num algoritmo de *machine learning*.

Podemos dividir a aprendizagem pela quantidade de *feedback* que é dada ao algoritmo:

1. **Supervisionada:** A aprendizagem supervisionada, utiliza dados de *input* e *output* pré-definidos. Estes dados formam o *dataset* de treino que é utilizado para ensinar ao algoritmo como prever *outputs* com base em *inputs* a que ainda não foi exposto [23].

2. **Não Supervisionada:** Os algoritmos de aprendizagem não supervisionada tentam encontrar padrões ocultos em dados não etiquetados. O algoritmo recebe apenas *inputs* sem *outputs* conhecidos, a aprendizagem é realizada ao procurar semelhanças nos dados de entrada [23].

3. **Semi-Supervisionada**: Existem vários modelos que combinam os dois tipos de aprendizagem já falados. A aprendizagem semi-supervisionada é utilizada em cenários onde existe uma pequena quantidade de dados etiquetados e uma grande quantidade não etiquetada. Este tipo de aprendizagem tem grande valor prático uma vez que pode aliviar o custo de computacional de etiquetar todo o *dataset* [23].

A aprendizagem pode ainda ser dividida em três outras categorias de acordo com a maneira como a informação é fornecida ao algoritmo.

1. ***Offline*:** Na aprendizagem *offline* o algoritmo é treinado utlizando o *dataset* completo. Utiliza-se a aprendizagem *offline* quando o sistema que se pretende modular não altera as suas propriedades dinamicamente. Os modelos que recorrem a este tipo de aprendizagem são de fácil implementação devido a não requererem uma constante aprendizagem e podem ainda ser facilmente retreinados. Uma utilização deste tipo de aprendizagem é em sistemas de reconhecimento de atividade humana, onde um classificador *offline* é treinado com dados *raw* lidos por sensores e depois é implementado para reconhecimento de atividade *online* [23].

2. ***Online*:** Nos algoritmos de aprendizagem *online* os dados utilizados pelo algoritmo são fornecidos de maneira sequencial e são utilizados para atualizar a representação do modelo em cada iteração. Este tipo de aprendizagem é útil em problemas onde os dados de treino se tornam disponíveis um de cada vez ou quando devido a limitações computacionais, treinar o algoritmo com o *dataset* inteiro se torna inviável ou impossível [23].

3. **Ativa:** Uma outra forma de aprendizagem online é a aprendizagem ativa, onde se decide quais exemplos seriam melhores para o treino, tentando escolher o menor número possível, e depois o algoritmo é treinado com esses exemplos [23].

Passemos agora à descrição de alguns algoritmos de *machine learning* em específico. Comecemos pelos algoritmos de aprendizagem supervisionada, para tal é importante dividirmos os os problemas em duas categorias: classificação e regressão.

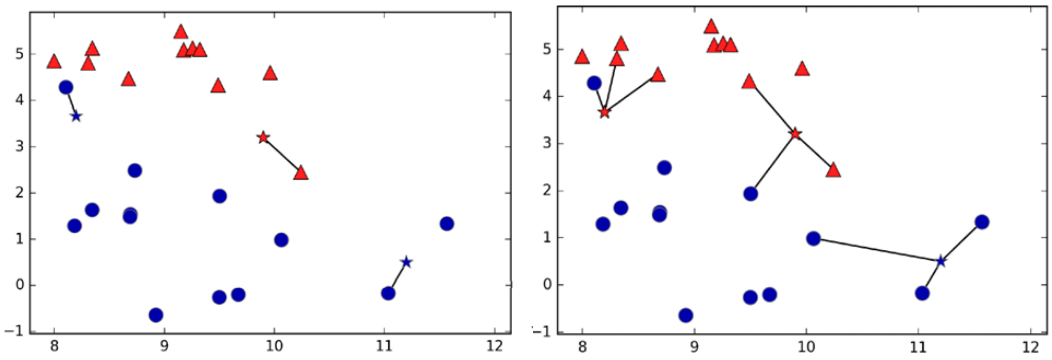
Na classificação o objetivo é prever uma etiqueta de classe, que é uma escolha de entre uma lista de possibilidades predefinidas. A classificação é normalmente composta por classificação binária, que é quando se pretende distinguir apenas entre duas classes, e por classificação multiclasse, que é a classificação de mais do que duas classes. Na regressão, pretende-se prever um número real. Como por exemplo, o rendimento anual de uma pessoa com base no seu nível de educação, idade e onde vive. Alguns dos algoritmos de aprendizagem supervisionada são:

1. **k-Nearest Neighbors:** O k-NN é provavelmente o algoritmo de *machine learning* mais simples. A construção deste modelo implica apenas guardar o *dataset* de treino. Para prever o valor de um novo dado, o algoritmo procura os pontos mais próximos no *dataset* de treino, ou seja, os seus *nearest neighbors*. Na sua forma mais simples, o algoritmo k-NN considera apenas um *nearest neighbor* que é o ponto do *dataset* de treino mais perto do ponto sobre o qual pretendemos fazer uma estimativa (Figura 3.3a). Pode-se ainda considerar um número, k, de *neighbors*. Quando k>1, utiliza-se a votação para atribuir uma etiqueta. Ou seja, para cada ponto que pretendemos testar, são contados quandos *neighbors* existem da classe 1 ou 0. A classe com mais prevalência é a classe atribuída (Figura 3.3b). Embora a Figura 3.3 apenas represente um problema binário de duas classes, podemos aplicar este algoritmo a um problema multiclasse [46].

2. **Regressão Linear:** A regressão linear é o modelo linear mais simples e clássico para regressão. A regressão linear calcula os parâmetros w e b que minimizam o quadrado médio do erro entre as previsões e os valores reais da regressão, y, no *dataset* de treino. A regressão linear não tem parâmetros, sendo um benefício mas por outro lado não permite nenhum controlo sobre a complexidade do modelo [46].

$$y = w * x + b \tag{3.3}$$

3. **Regressão Logística:** A regressão logística é algoritmo comum para a implementação de classificação linear. Num modelo de classificação linear, a



(a) Classificação com k=1.

(b) Classificação com k=3.

Figura 3.3: Representação do k-NN com diferentes valores de k. As bolas e triângulos representam diferentes valores de diferentes classes do *dataset* de treino. As estrelas são os pontos em teste, com a cor da classe atribuída. Adaptado de [46].

fronteira de decisão é uma função linear do *input*. Ou seja, um classificador binário que separa duas classes utilizando uma linha, plano ou hiperplano. O parâmetro que permite restringir o modelo de maneira a evitar *overfitting* é o C. Se for utilizado um valor baixo de C, o algoritmo vai tentar se ajudar à maioria dos dados, enquanto que um valor alto de C eleva a necessidade de cada dado ser classificado corretamente (Figura 3.4) [46].

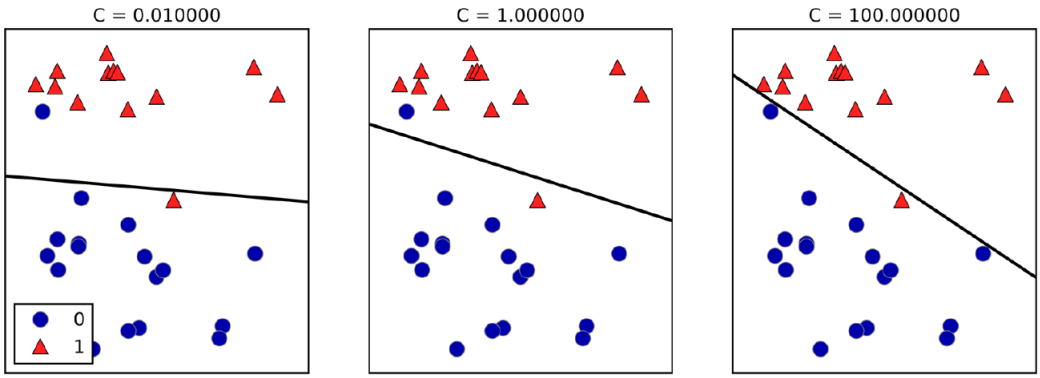


Figura 3.4: Fronteiras de decisão para diferentes valores de C para um algoritmo de Regressão Logística. Adaptado de [46].

4. **Árvores de decisão:** As árvores de decisão são um modelo bastante utilizado para classificação e regressão. De uma maneira simplista, aprendem uma hierarquia de questões *if/else* que levam a uma decisão [46]. A árvore consiste num nó raiz, nós internos chamados de nós decisão que vão testando as questões *if/else* e nós de folha que correspondem a uma classe [23]. Embora estes algoritmos sejam de bastante fácil compreensão e invariáveis com o aumento

dos dados, o *overfitting* é normal ocorrer e por isso, na maioria das aplicações utilizam-se as *random forests* [46].

5. **Random forests:** Uma *random forest* é um conjunto de árvores de decisão, onde cada árvore é ligeiramente diferente das outras. A filosofia por trás das *random forests* é que cada árvore pode realizar uma boa previsão, mas o mais provável é ocorrer *overfitting*. Se forem construídas muitas árvores, todas a funcionar bem e com *overfitting* a ocorrer de maneiras diferentes, podemos reduzir a quantidade de *overfitting* ao calcularmos a média dos resultados (Figura 3.5) [23].

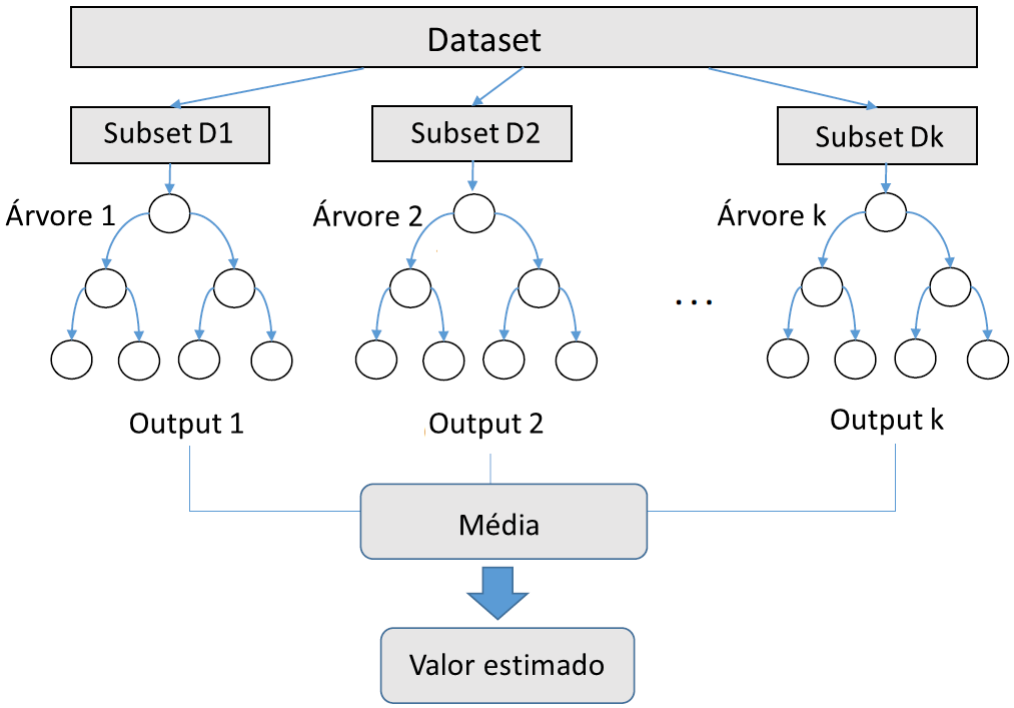


Figura 3.5: Representação gráfica do funcionamento do algoritmo *random forests*. Adaptado de [46].

6. **Support Vector Machine (SVM):** A SVM é um algoritmo de resolução de problemas de classificação ao mapear os dados de entrada num espaço de *features* de maior dimensão, onde é possível separar linearmente por um hiperplano, semelhante ao da Figura 3.4. É ainda possível realizar o mapeamento não linear com recurso a funções *kernel*. Diferentes funções *kernel* funcionam melhor em diferentes aplicações. As funções *kernel* mais comuns são as lineares, polinomiais e o kernel da função de base radial (RBF) [23].

7. **Redes neuronais:** As redes neuronais são um algoritmo de aprendizagem supervisionada inspirado no funcionamento do cérebro, normalmente utilizado para derivar limites de decisão complexos e não lineares para a construção

de um modelo de classificação mas também funcionam para problemas de regressão caso se pretenda prever um valor real. As redes neuronais são famosas pela sua capacidade de identificar padrões complexos e detetar relações não-lineares nas variáveis de entrada. A arquitetura de uma rede neuronal (Figura 3.6) é composta por uma camada de entrada, um número de camadas ocultas e uma camada de saída. A camada de entrada é a camada das variáveis de entrada. Cada camada oculta contém um número de elementos chamados de neurónios responsáveis pelo processamento dos inputs usando uma função de ativação ou transferência. Os elementos da rede são conectados por linhas que têm um valor numérico associado que é aprendido pelo algoritmo. A camada de saída é a camada que nos fornece uma previsão tendo em conta as conexões definidas pela rede [23].

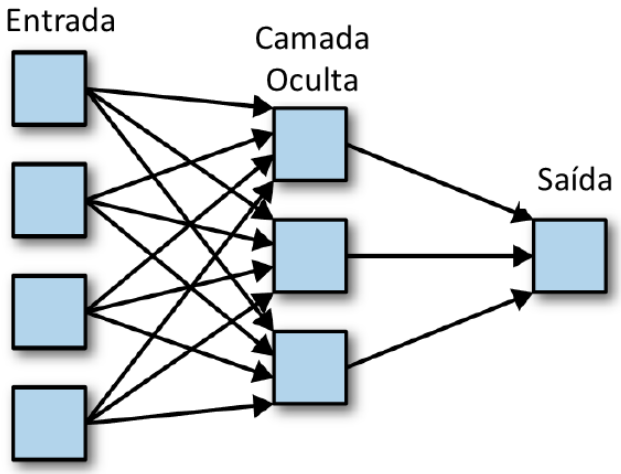


Figura 3.6: Representação gráfica de uma rede neuronal.

Por último temos os algoritmos de aprendizagem não supervisionada, os chamados algoritmos de *clustering*, semelhantes aos algoritmos de classificação, estes algoritmos atribuem um número a cada dado indicando a qual grupo pertence:

1. **k-Means:** O k-means é o algoritmo de *clustering* mais simples e mais usado. O objetivo é tentar encontrar grupos que representam regiões dos dados. O algoritmo consiste em dois passos: atribuir pontos ao centro do grupo mais próximo e depois recalcular o centro como sendo a média dos valores atribuídos a esse mesmo centro (Figura 3.7). O algoritmo termina quando a atribuição dos dados aos centros dos grupos não se altera. Podemos olhar para o *clustering* como um algoritmo de classificação, uma vez que a cada dado é atribuído uma etiqueta, que neste caso é representado por um grupo [46].

   Tendo agora um pequeno entendimento dos algoritmos mais comuns de *machine learning* e da sua maneira de funcionamento, podemos falar de quais

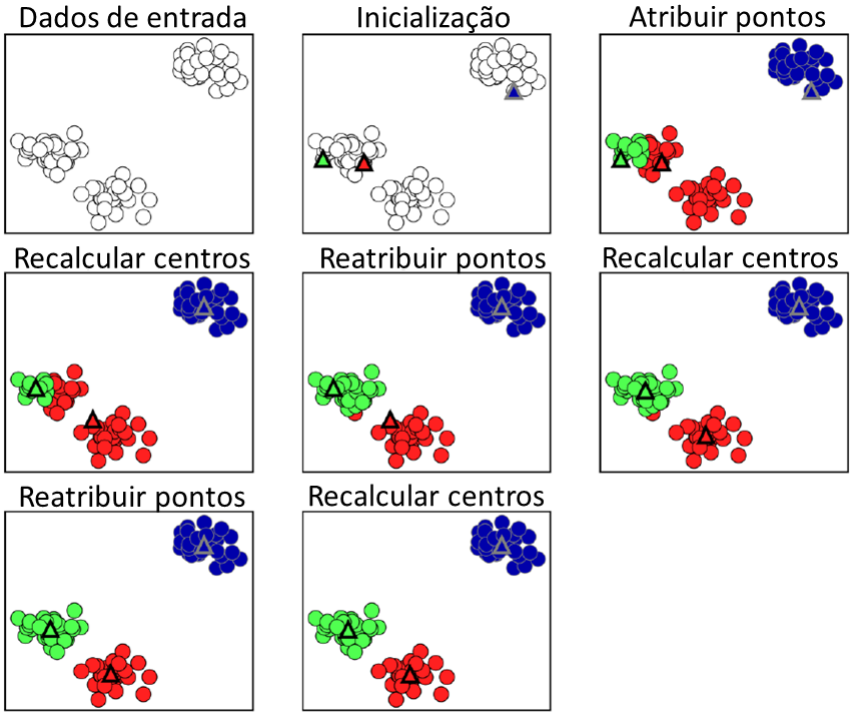


Figura 3.7: Dados de entrada e os passos até ao cálculo final do algoritmo k-means. Os triângulos representam os centros dos grupos e os pontos dados. Adaptado de [46].

são utilizados pelos autores no estado da arte atual e do desempenho obtido com cada modelo. [40], testou três algoritmos diferentes, LDA, ELM e SVM. Em cada algoritmo testou três métodos de extração e três métodos de seleção de *features*. Embora todos os modelos criados tenham tido um desempenho entre os 80-90 %, o modelo com melhor resultado foi o LDA, utilizando como *features* o método de Welch e como algoritmo de redução de *features* o Davies Bouldin. À semelhança de [40], [24] testou 16 algoritmos diferentes de *machine learning*, dos referidos anteriormente neste subcapítulo destacam-se o k-NN com um desempenho de 51,03 %, Multi Layer Perception (MLP) que é uma forma de rede neuronal com 65,14 %, SVM com kernel gaussiano com 65,13 % e Convolutional Neural Network (CNN) também um tipo de rede neuronal, com um desempenho de 69,03 %. [39] aplicou três algoritmos aos seus *datasets* sendo eles a regressão logística, árvores de decisão e extra trees. Destes três modelos o algoritmo extra trees foi o que obteve melhor desempenho, superior a 99 % para ambos os *datasets*. [38] obteve um desempenho de 99,83 % com um algoritmo do tipo rede neuronal, [37] obteve 100 % de desempenho utilizando a classificação com recurso a SVM e [33], utilizando o mesmo método obteve 87 %. [29] utilizou SVM para um problema multiclasse de quatro classes, tendo obtido uma média de 93,49 % de desempenho das quatro classes para 20 sujeitos. Por último, [1] testou inicialmente 9 algoritmos de *machine*

learning no seu *dataset* para tentar perceber qual o algoritmo que lhe daria melhor desempenho. Dos 9 algoritmos, salienta-se o SVM, árvores de decisão e k-NN por serem os anteriormente mencionados neste subcapítulo, tendo o SVM obtido o melhor desempenho dos 9, com 72,47 %, as árvores de decisão 50,92 % e o k-NN 49,40 %. Tendo decidido qual o algoritmo a utilizar, [1] testou de seguida 10 kernels diferentes, tendo obtido a melhor classificação para o kernel de spearman com 73,74 %.

3.5 Metodologias de Avaliação do Desempenho

Antes de falarmos sobre as diferentes metodologias de avaliação de desempenho, é importante perceber em que situações utilizar apenas o desempenho é enganador. Existem dois tipos de erros, os falsos negativos e falsos positivos. Quando fazemos uma previsão negativa incorreta, chamamos de falso negativo, quando a previsão é uma previsão positiva incorreta chamamos de falso positivo. Estes dois tipos de erros têm um papel importante em problemas binários onde uma das classes é mais frequente que a outra. Um *dataset* onde uma classe é muito mais frequente que a outra são chamados de desequilibrados ou *datasets* com classes desequilibradas. Num cenário real, o normal é termos um *dataset* não balanceado. Vamos agora imaginar que construímos um algoritmo de *machine learning* com 99% de desempenho, numa tarefa onde o *dataset* é altamente desequilibrado. Embora o desempenho pareça muito bom, não tem em conta o desequilíbrio das classes. Assim, é importante utilizar outros métodos de avaliação do desempenho para ser possível observar comportamentos do modelo que ficam escondidos se olharmos apenas para o valor do desempenho [46].

1. **Matrizes de confusão:** Uma das representações de mais fácil compreensão de classificação binário são as matrizes de confusão (Figura 3.8). O *output* de uma matriz de confusão é uma matriz de 2 por 2, onde as linhas correspondem às classes verdadeiras e as colunas às classes previstas. O valor de cada célula corresponde à quantidade de vezes que um dado que pertence à coluna onde foi colocado foi classificado como correspondendo à classe da coluna. A diagonal principal corresponde a classificações corretas, enquanto que as da outra diagonal correspondem à quantidade de previsões erradas [46].

2. **Precisão, *Recall* & f-score:** Existem algumas maneiras de sumarizar a matriz de confusão, sendo a precisão e o *recall* os mais utilizados. A precisão corresponde à quantidade de dados previstos como positivos que são realmente positivos e é utilizada quando o objetivo é limitar o número de falsos positivos [46].

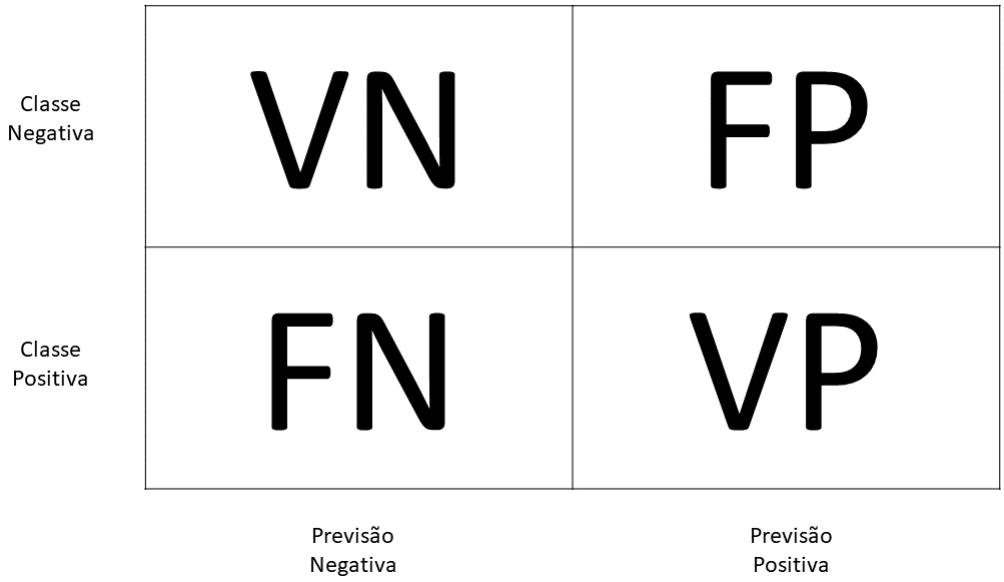


Figura 3.8: Matriz de confusão para um problema de classificação binária. VP-Verdadeiro positivo; VN-Verdadeiro negativo; FP-Falso positivo; FN-Falso negativo. Adaptado de [46].

$$Precisão = \frac{VP}{VP + FP} \tag{3.4}$$

O *recall* é uma medida de quantos elementos relevantes foram selecionados. Por outras palavras, o *recall* pode ser visto como a capacidade do modelo em encontrar todos os dados da classe de interesse num *dataset* [46].

$$Recall = \frac{VP}{VP + FN} \tag{3.5}$$

Existe uma troca entre a otimização do *recall* e da precisão. Se criarmos um modelo cujos dados sejam atribuídos todos como sendo verdadeiros positivos, o nosso recall seria perfeito (1.0), enquanto que a precisão seria muito baixa. Pelo contrário, se um modelo apenas fizer uma previsão sobre um ponto como positivo e esse ponto for efetivamente positivo, a precisão seria ótima mas o recall muito mau [46].

Embora a precisão e o *recall* forneçam informação valiosa acerca do desempenho dum modelo, se considerarmos apenas um deles, podemos cair no erro de não ter uma visão real do desempenho devido às razões ainda agora comentadas. Uma maneira de ter em conta tanto o valor do *recall* como da precisão é o f-score, que é a média harmónica destes dois indicadores [46].

$$f\text{-}score = 2 * \frac{precisão * recall}{precisão + recall} \tag{3.6}$$

3. **Curva de precisão-*recall*:** Decidir o limite entre o valor da precisão e do *recall* num modelo, é uma maneira de ajustar o equilíbrio entre a precisão e o *recall* de um dado modelo. Se pretendermos um modelo que falha menos dos 10% de dados positivos, basta selecionar um *recall* de 90%. É sempre possível definir um limite para um determinado objetivo, contudo a dificuldade está em obter um modelo que também tenha uma boa precisão tendo este valor de *recall*. A definição de um limite, como o de um *recall* de 90% é chamado de definir um ponto de operação [46].

   Quando se desenvolve um novo modelo, este ponto de operação ainda não está claro. Por isso, é necessário observar todos os limites possíveis de uma vez. Isto é possível ao utilizarmos uma curva de precisão-*recall* (Figura 3.9). Cada ponto da curva corresponde a um valor limite. Quanto mais a curva se aproxima do canto superior direito, melhor é o classificador. Um ponto no canto superior direito significa uma alta precisão e um alto *recall* [46].

   É importante ainda fazer-se uma comparação entre diferentes algoritmos de *machine learning* uma vez que diferentes algoritmos apresentam diferentes curvas de precisão-*recall*. Comparar estes modelos utilizando apenas este método de avaliação de desempenho, embora forneça uma boa introspeção, é um processo bastante manual. Uma maneira de resumir estes dados é calcular o integral (área sob a curva), também conhecido como precisão média [46].

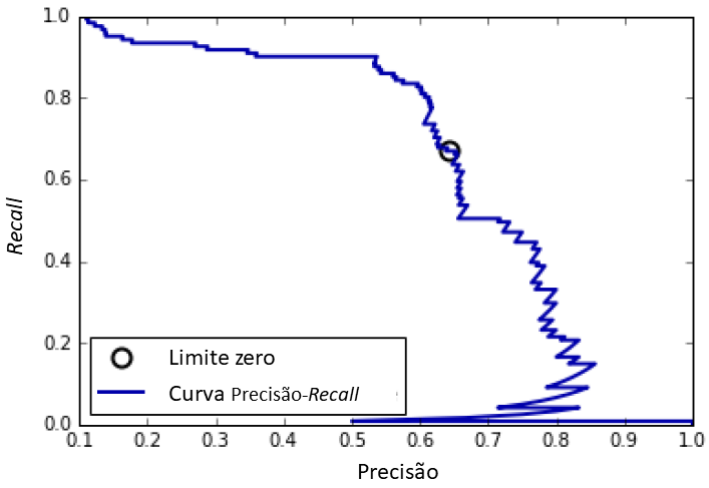


Figura 3.9: Exemplo de uma curva de precisão-*recall*. Adaptado de [46].

4. **Curva ROC:** A curva de *receiver operating characteristics* (ROC) é outra ferramenta que pode ser utilizada para avaliar o desempenho de um algoritmo de *machine learning*. À semelhança da curva de precisão-*recall*, também a curva ROC considera todos os limites possíveis de um algoritmo só que em vez de utilizar a precisão e o *recall*, mostra a taxa de falsos positivos e o

recall (Figura 3.10). A curva ROC ideal é aquela que se aproxima mais do canto superior esquerdo, ou seja, um *recall* alto e mantendo uma taxa de falsos positivos baixa. Tal como falado anteriormente para a curva de precisão-*recall*, também aqui é importante resumir os dados da curva através de um único valor, a área sob a curva (AUC) [46].

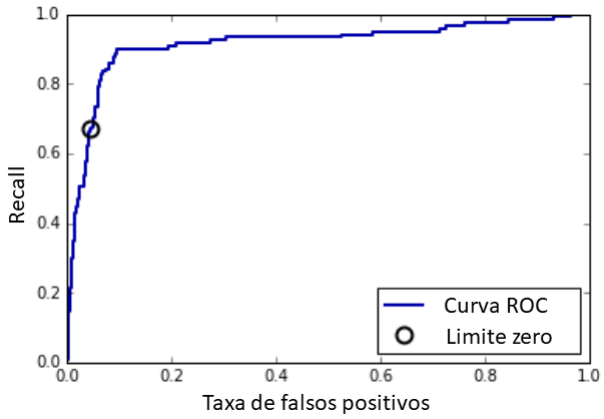


Figura 3.10: Exemplo de uma curva ROC. Adaptado de [46].

Capítulo 4

Materiais e Métodos

4.1 Aquisição de Dados

O *dataset* utilizado para a execução do presente trabalho é constituído por um total de 17 ratos, 8 ratos do sexo masculino, sendo 6 deles do tipo wild-type (WT) e 2 com neurofibromatose 1 (NF1), e por 9 ratos do sexo feminino, tendo 4 o genótipo WT e 5 o NF1. Os dados presentes neste *dataset* são as gravações de sessões de eletrofisiologia, onde os animais foram expostos a estímulos familiares e *novel*, assim como as respetivas indicações temporais em que cada estímulo ocorreu.

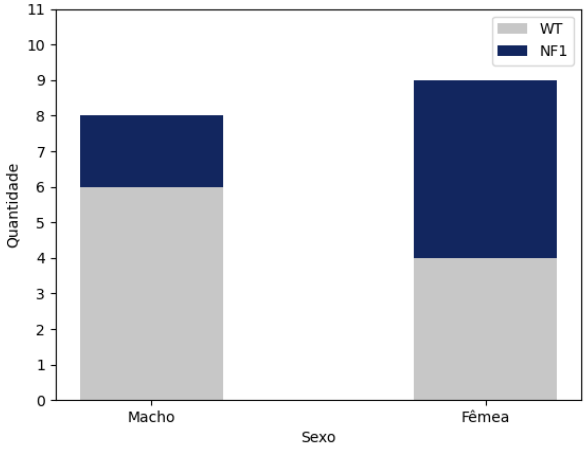


Figura 4.1: Distribuição demográfica dos ratos utilizados.

Durante o processo experimental, toda a legislação foi cuidadosamente aplicada (Directiva UE 2010/63/EU e Decreto-Lei 13/2013). O projeto é licenciado pela autoridade nacional responsável pela experimentação animal, a DGAV (Direção-Geral de alimentação e veterinária) e monitorizada pela ORBEA (Órgão responsável pelo bem-estar animal) do ICNAS (Instituto de Ciências Nucleares aplicadas à saúde). São aplicados todos os esforços necessários para minimizar o número de animais utilizados assim como o seu sofrimento. Os animais foram mantidos em grupos de 2 a 5 indivíduos durante um ciclo de 12 h de luz e escuridão. Os ratos trangénicos utilizados (NF1+/-) têm uma mutação específica para um alelo do gene que codifica a neurofibromina. Estes ratos foram obtidos através do cruzamento de ratos NF1+/- (origem C57BL/6N) com ratos 129/Sv.

Antes do protocolo de estimulação visual, os animais são preparados durante uma semana. Nos primeiros 2 dias, os ratos são colocados na arena com um estímulo cinzento para que ocorra a habituação ao ambiente da arena, durante 20 minutos. O monitor é colocado a 30 cm de distância da arena. Durante 6 dias, o animal é colocado numa arena onde irá ocorrer a estimulação com uma orientação predefinida. O estímulo consiste numa grelha sinusoidal de 100 % de contraste que inverte a fase a uma frequência de 2 Hz com uma frequência espacial de 0,05 ciclos por grau. Os estímulos visuais são apresentados em cinco blocos de 100 inversões de fase por bloco com um intervalo entre blocos de 30 s em que ocorre a estimulação com ecrã cinzento. No dia 7, o animal foi anestesiado por uma injecção intraperitoneal (IP) de uretano (1,5 g/kg de uretano em soro fisiológico). Também foi administrada Xilazina através de uma injeção IP (5 mg/kg). Além disso, atropina (0,1 mg/kg) e glucose (5 % de glucose em soro fisiológico) foram administradas por via subcutânea. Se necessário também era administrada dexametasona (2 mg/kg) por via subcutânea. Quando, o animal estava profundamente anestesiado, era colocado no aparelho estereotáxico para fazer a cirurgia de craniotomia seguindo as coordenadas do córtex visual primário (+0,5 anterior a lambda, 2,8 ML hemisfério direito, ângulo de 10 graus). Foi obtida uma janela craniana onde foi colocada uma sonda de silicone com um eléctrodo de 16 canais (Neuronexus) (Figura 4.2). No outro hemisfério, foi feita outra perfuração onde foi colocado o eléctrodo de referência que consistia num fio de níquel-crómio. O eléctrodo terra foi implantado no músculo do animal. Com o aparelho de electrofisiologia montado, as gravações foram obtidas utilizando o software Multichannel Experimenter. As gravações foram obtidas enquanto uma estimulação visual era apresentada com blocos do estímulo familiar e blocos do estímulo *novel*. As orientações utilizadas foram -30º, 15º e 60º.

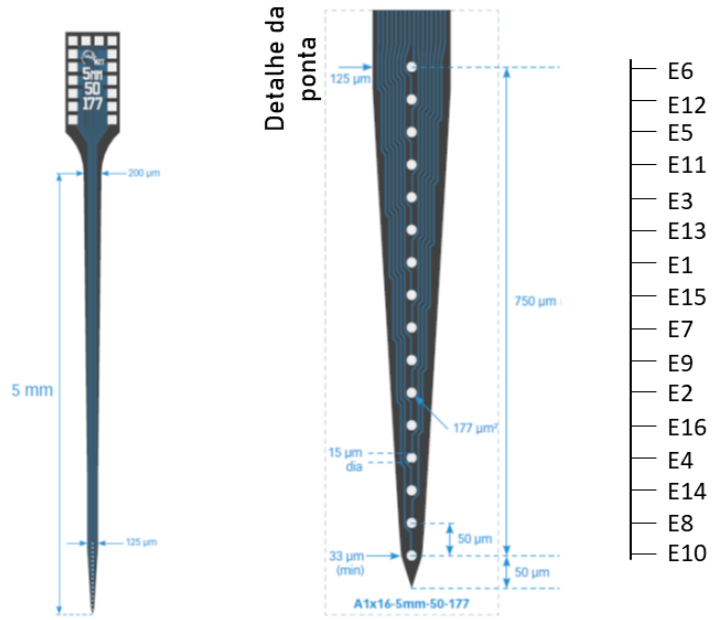


Figura 4.2: Especificações, arquitetura e distribuição dos canais do tipo de elétrodo utilizado. No topo temos o canal mais superficial (6) e no fundo o mais profundo (10).

4.2 Pré-Processamento

Antes de se iniciar o pré-processamento é necessário escolher o canal que se irá utilizar, no presente trabalho o canal escolhido foi o 2 devido a ser um dos canais com maior amplitude. A amplitude é de particular importância neste caso uma vez que esta é uma das diferenças entre um sinal correspondente a um estímulo *novel* e a um estímulo familiar. O pré-processamento é iniciado com a verificação da frequência a que os EEG foram registados e efetuado o *downsample* para 10 kHz caso a frequência fosse diferente. Esta etapa é essencial para se obter ficheiros uniformizados de maneira a ser possível a realização de cálculos utilizados na obtenção das *epochs*, filtragem e extração de *features*.

Após a verificação da frequência, os EEG são agora passíveis de análise. Tal como descrito anteriormente, cada ficheiro de EEG tem um ficheiro correspondente com as marcas temporais dos estímulos. Assim, a filosofia adotada para a extração das *epochs* foi o cruzamento das marcas temporais presentes em ambos os ficheiros até 200 ms após cada marca temporal, tal como utilizado por [22] (Figura 4.3). Devido ao downsample efetuado, algumas marcas temporais não tinham correspondência direta com nenhuma marca temporal do EEG. Assim, a solução encontrada foi procurar o ponto seguinte mais próximo nestes casos, o que daria uma diferença desde o início do estímulo de 50 μs. Como a ordem de magnitude da janela temporal utilizada é em ms, este adiantamento corresponde apenas a 0.05 ms, ou seja, uma pequena perda de 0.025 % do sinal. Recorde-se que uma *epoch* é a janela temporal

corresponde a um intervalo de tempo pré-definido após o estímulo. Tendo todas as *epochs* extraídas, aplicou-se um filtro Butterworth de ordem 8 passa-banda 5-48 Hz, de acordo com o trabalho de [1]. Por último, para cada vetor realizou-se o mapeamento binário dos estímulos "Novel" e "Familiar" a 0 e 1, respetivamente.

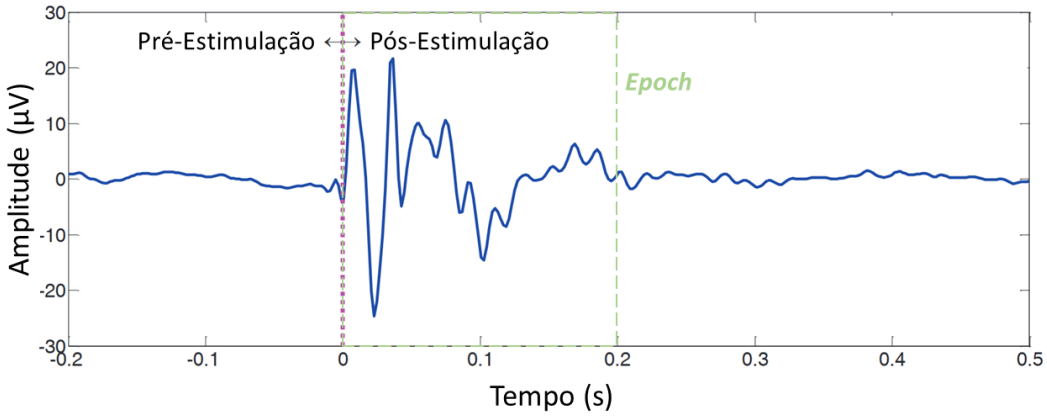


Figura 4.3: Exemplificação da janela utilizada para a extração da *epoch*; A tracejado rosa: o momento em que ocorre o estímulo; A tracejado verde: a janela extraída. Adaptado de [47]

4.3 Extração de Características

No processo de extração de *features* foram considerados dois conjuntos, que se baseavam nos componentes do EEG. Inicialmente, desenvolveu-se o método 1 que consistia na aplicação do método de Welch a cada *epoch* para obter a densidade espectral de potência. O método foi aplicado com uma frequência de amostragem de 10 kHz, sendo esta uma das razões para o *downsample* efetuado anteriormente na etapa de pré-processamento. A escolha do método de Welch deveu-se à sua utilização por vários autores referidos no capítulo 3.2. O método 2 teve por base os valores obtidos anteriormente no método 1. Destes valores calculou-se a média, o desvio padrão, a mediana, a variância, a assimetria e a curtose. Estas *features* estatísticas são, de acordo com [48] as normalmente utilizadas no domínio da frequência para a análise de EEG e foram por essa mesma razão as escolhidas.

Tabela 4.1: Conjuntos de *features* utilizados.

Conjunto	Features
Conjunto 1	Método de Welch
Conjunto 2	Média
	Desvio Padrão
	Mediana
	Variância
	Assimetria
	Curtose

4.4 Classificação

A etapa da classificação é o culminar de todo o trabalho até então realizado e o seu objetivo é classificar o estímulo presente em cada *epoch* como *novel* ou familiar, através de um algoritmo de *machine learning*. Assim, três algoritmos de classficação foram implementados, SVM, RF e ANN, utilizando ambos os conjuntos de *features* anteriormente descritos. O *dataset* construído foi dividido em 2 conjuntos, um de treino e um de teste, em 75 % e 25 % respetivamente. Esta divisão foi a mesma para os vários classificadores, de maneira a ser possível a intercomparabilidade dos resultados. Um dos parâmetros que a função responsável pela criação dos conjuntos de treino e teste utiliza é o *random state*. Este parâmetro é responsável por garantir que a divisão dos dados é sempre a mesma, ou seja, se este parâmetro não fosse definido, os conjuntos criados seriam diferentes em cada iteração. É importante esta fixação, de maneira a permitir que os valores da *accuracy* obtidos com cada alteração num componente da *pipeline* seriam devido a essa alteração e não à alteração da divisão dos dados.

Devido ao elevado número de *features* do conjunto 1, tornou-se necessário encontrar métodos que tornassem estes dados mais facilmente comparáveis entre si. Assim as etapas Redução de *Features* e Normalização foram apenas aplicadas ao conjunto de *features* 1. Uma vez que os valores do vetor de *features* do conjunto 1 são os valores obtidos através do método de Welch, teremos presentes neste vetor valores que serão irrelevantes para a resolução do problema em causa. A informação que se pretende retirar daqui é referente ao pico existente na onda VEP. Se olharmos para uma onda típica do VEP (Figura 2.5) e sabendo que uma onda é composta por diferentes componentes no domínio da frequência, percebemos que existirão valores do método de Welch que serão irrelevantes por não serem relacionados com componentes de frequência do sinal referentes ao estímulo. Assim, uma remoção destas *features* através de um método de redução de *features* é essencial. Aqui, o método escolhido foi o *truncated* SVD por ser eficaz em dados dispersos e por permitir um

maior controlo uma vez que é possível definir o número de *features* que se pretende manter.

Foram ainda testados vários algoritmos de normalização de maneira a tentar obter uma melhor *accuracy*, tais como o MinMaxScaler, RobustScaler e o StandardScaler. O MinMaxScaler dimensiona e traduz cada *feature* individualmente de modo a que esteja no intervalo determinado no conjunto de treino, por exemplo entre zero e um. O RobustScaler remove a mediana e dimensiona os dados de acordo com a amplitude interquartil (o padrão é IQR: intervalo interquartil). O IQR é o intervalo entre o 1º quartil (25º quantil) e o 3º quartil (75º quantil). O objetivo desta normalização é remover *outliers*. O StandardScaler padroniza as *features* removendo a média e dimensionando para a unidade de variação.

O sucesso de um classificador do tipo SVM, está relacionado diretamente com a organização espacial dos dados. Quanto melhor agrupados estiverem, melhor será o desempenho do classificador, por isso perceber o impacto que certas mudanças sortiam nessa mesma distribuição é fundamental. Assim, tornou-se útil a utilização de um método que permitisse visualizar num mapa bidimensional a distribuição destes dados. Para este efeito, utilizou-se o t-SNE (*t-Distributed Stochastic Neighbor Embedding*). O t-SNE é um método estatístico para a visualização de dados, atribuindo a cada dado uma localização num mapa bi ou tridimensional.

A abordagem adotada para a afinação dos parâmetros foi a utilização de métodos de avaliação do desempenho, como a *accuracy* (sendo este o método mais utilizado), a precisão e o recall através das curvas precision-recall, e a curva ROC. A *accuracy* foi calculada segundo a expressão 4.1:

$$Accuracy = \frac{NúmeroDePrevisõesCorretas}{NúmeroTotalDePrevisões}.100\,\% \tag{4.1}$$

Uma última consideração a ter no momento da avaliação do desempenho do modelo é a seguinte: Ao se dividir o *dataset* em dois conjuntos, um de treino e um de teste, está-se a garantir que o classificador irá ser avaliado num conjunto de dados que nunca antes viu, ou seja, o conjunto de teste. Contudo, este conjunto de teste representa apenas uma pequena porção dos dados presentes num cenário de implementação real. Por isso, a utilização de uma estratégia que permita aproveitar ao máximo os dados disponíveis para se realizarem múltiplos treinos e testes, é necessária. Tendo esta consideração em mente, foi implementado um método de *cross validation* (CV). Na CV é definido um número fixo de *folds* (partições) dos dados, é feita a análise aos dados em cada *fold* e depois é feita uma estimativa final do desempenho do modelo através da média do desempenho em cada *fold*. Dentro da CV existem vários métodos, tais como o K-fold, Stratified K-fold, Leave-One-Out ou Leave-P-Out. Sendo o dataset balanceado, o método escolhido foi o K-fold, uma vez que garante que cada dado presente no *dataset* original tem a chance de

aparecer em ambos os conjuntos de treino e teste ao longo das iterações, resultando num modelo menos enviesado.

A *pipeline* desenvolvida encontra-se na Figura 4.4.

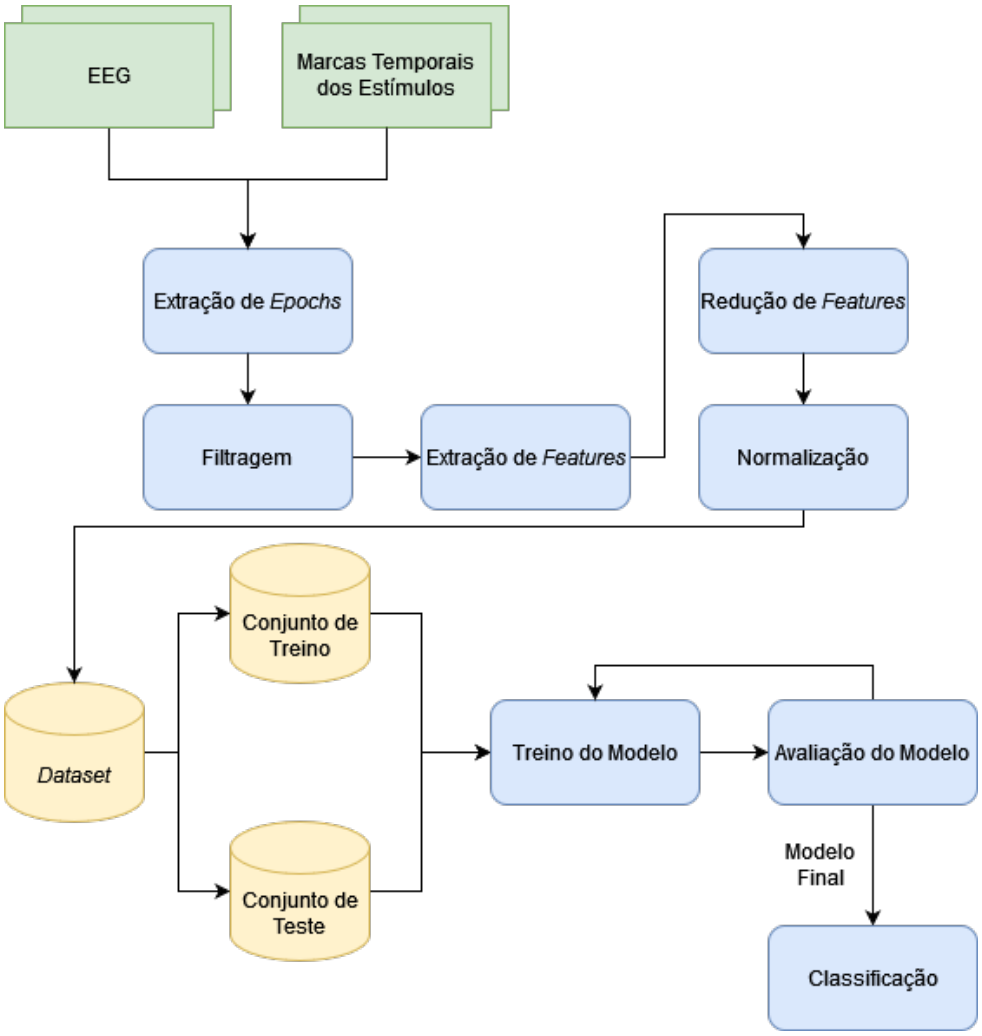


Figura 4.4: *Pipeline* desenvolvida para o problema de classificação binária em estudo.

Capítulo 5

Discussão de Resultados

Da aplicação dos métodos descritos anteriormente para o pré-processamento obtiveram-se 6666 *epochs*, com uma distribuição homogénea entre os vários murganhos como se pode confirmar pelo gráfico 5.1. Note-se que existem algumas exceções, os murganhos 1653.1 e 1650.1 têm uma maior representatividade do que os restantes, com 606 *epochs* cada um, e o 2091.0 tem a menor representação de todos com 202 *epochs*. Salienta-se ainda a importância da solução descrita no capítulo 4.2 para resolver o problema do cruzamento das marcas temporais das *epochs* com as marcas temporais dos estímulos, fator que tornou possível esta homogeneidade dos dados. Assim, o *dataset* obtido é totalmente balanceado, apresentando 50 % de dados relativos a *epochs* do tipo estímulo novel e 50 % dos dados relativos a estímulos familiares.

Dos métodos de extração de *features* utilizados, resultaram dois vetores, um vetor de 129 *features* para o método 1 e um vetor de 6 *features* para o método 2. Focando agora no maior vetor de *features*, verificou-se que o elevado número de *features* estava a causar uma *accuracy* próxima de 50 %, mesmo quando utilizando os diferentes método de normalização, significando uma elevada redundância de algumas *features*. Isto em termos de representação espacial significava uma baixa organização em *clusters* o que era indesejado. Quando se reduziu a quantidade de *features* para 15, notaram-se diferenças significativas ao utilizar o método de normalização MinMaxScaler. Na figura 5.2 temos representado o impacto da utilização dos métodos de normalização na distribuição espacial dos dados relativos às 15 *features* selecionadas pelo *truncated SVD*. O papel do t-SNE neste processo foi conseguir criar uma relação causal entre o agrupamento das *features* e o desempenho do classificador SVM.

Ou seja, permitiu diminuir a abstratividade dos dados enquanto decorria o processo de afiniação do modelo. Na figura 5.2b, observa-se uma organização em pequenos *clusters* das classes novel (pontos vermelhos) e familiar (pontos azuis). De facto, devido à SVM tirar partido desta organização em *clusters* é nesta configuração que se obtém a melhor classificação para este classificador, tabela 5.1.

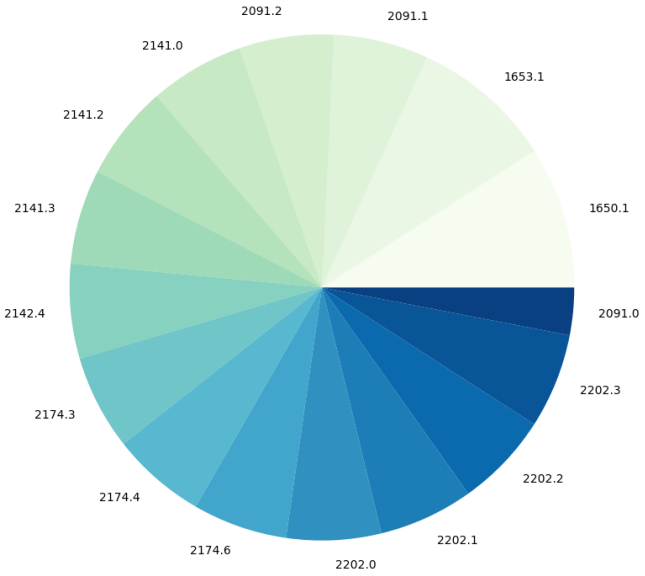


Figura 5.1: Distribuição das *epochs* por murganho. A quantidade de *epochs* recolhidas é de magnitude semelhante para cada murganho, em grande parte devido à solução adotada para resolver o facto de algumas marcas temporais não terem correspondência direta no EEG aquando do *downsample*. n=6666

De maneira a testar qual seria o método de normalização e conjunto de dados que resultavam num melhor desempenho para cada classificador, aplicaram-se os três métodos de normalização a cada classificador e a cada conjunto de dados. O método de avaliação aqui utilizado foi a *accuracy* com uma divisão de dados tal como descrito na secção anterior e sem recorrer ao K-fold, uma vez que o propósito desta avaliação não era ganhar confiança no modelo mas sim definir qual o método e conjunto a utilizar.

Os resultados obtidos para os 3 métodos de classificação e para cada método de normalização encontram-se descritos na tabela 5.1. Podemos observar que o algoritmo com melhores resultados no geral foi o RF. Para o conjunto 1 obteve-se um máximo de 72 % tanto com o método de normalização MinMaxScaler como com o StandardScaler. Já com o conjunto 2 conseguiu-se chegar à melhor *accuracy* do presente trabalho, 81 % com o RobustScaler. Em relação à SVM obteve-se o melhor resultado, 73 % com o conjunto 1 e utilizando o MinMaxScaler. Por último, quando utilizando o RobustScaler e o conjunto de dados 1, a ANN obteve a sua accuracy mais alta de 72 %.

Tabela 5.1: *Accuracy* obtida por classificador, método de normalização e conjunto de *features* utilizado.

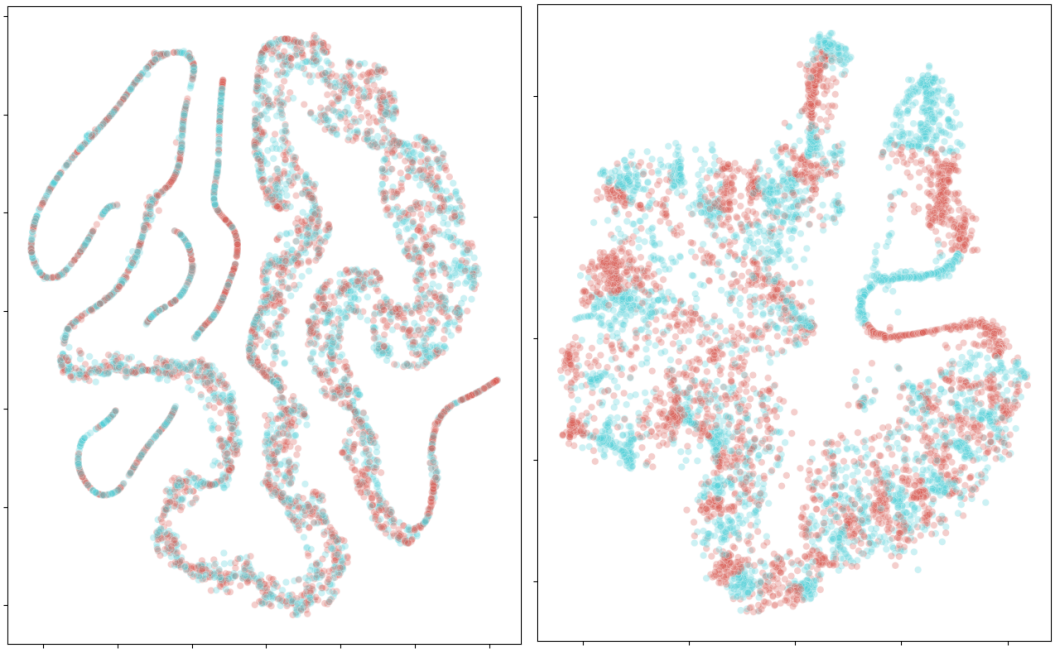
Classificador	Método de Normalização	Conjunto 1	Conjunto 2
SVM	MinMaxScaler	**73 %**	54 %
	RobustScaler	50 %	58 %
	StandardScaler	57 %	65 %
RF	MinMaxScaler	72 %	57 %
	RobustScaler	60 %	**81 %**
	StandardScaler	72 %	80 %
ANN	MinMaxScaler	58 %	62 %
	RobustScaler	**72 %**	60 %
	StandardScaler	60 %	53 %

Tendo definido anteriormente, com recurso à tabela 5.1, que o classificador SVM tinha melhor desempenho quando utilizando o MinMaxScaler e o Conjunto 1 de *features*, e o RF utilizando o RobustScaler com o Conjunto 2, testou-se de seguida a robustez dos modelos recorrendo à CV. Aqui definiram-se três tipos de *folds*, k=2, 5 e 10 , tendo no final calculado a média dos 3 *folds* para cada classificador. À ANN não foi aplicado este teste devido ao elevado custo computacional.

Na Tabela 5.2 podemos ver que a *accuracy* de ambos os classificadores não divergiram muito da *accuracy* obtida no teste anterior (Tabela 5.1) para os três diferentes *folds*. Assim, a RF continua a ter o melhor desempenho de 81 % quando k=10, ao utilizar o RobustScaler e o Conjunto 2.

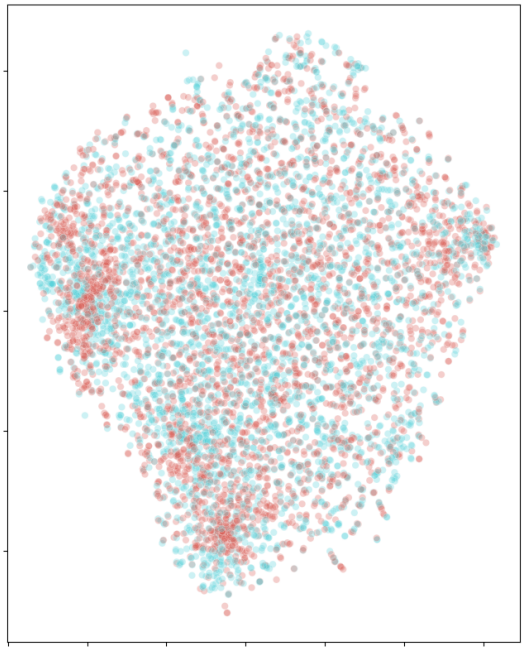
Tabela 5.2: *Accuracy* obtida utilizando o método de CV K-fold para k=2, 5 e 10.

Classificador	k=2	k=5	k=10	Média
SVM	70 %	72 %	72%	71 %
RF	78 %	80 %	**81 %**	**80 %**

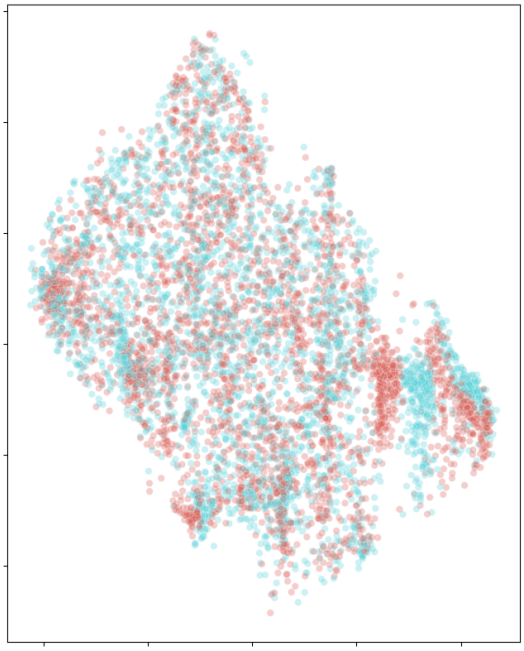


(a) Sem normalização.

(b) MinMaxScaler



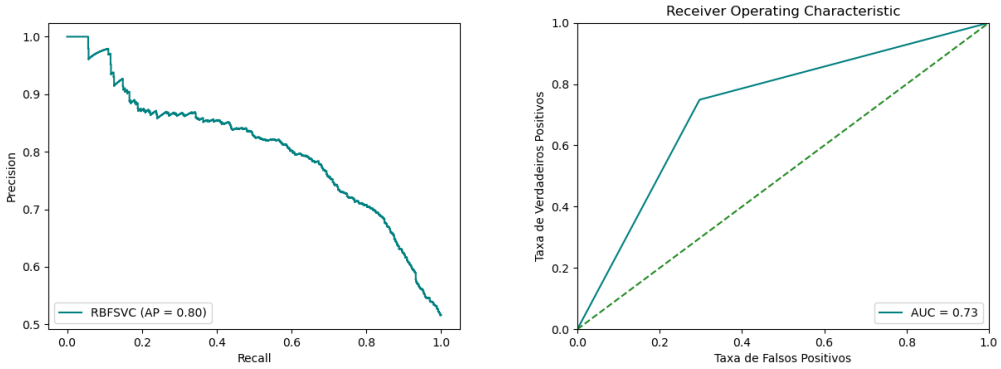
(c) RobustScaler



(d) StandardScaler

Figura 5.2: Impacto da utilização dos métodos de normalização na distribuição espacial dos dados do Conjunto 1 e para as 15 *features* selecionadas. Cada ponto vermelho corresponde a um dado da classe novel e cada ponto azul a um dado da classe familiar.

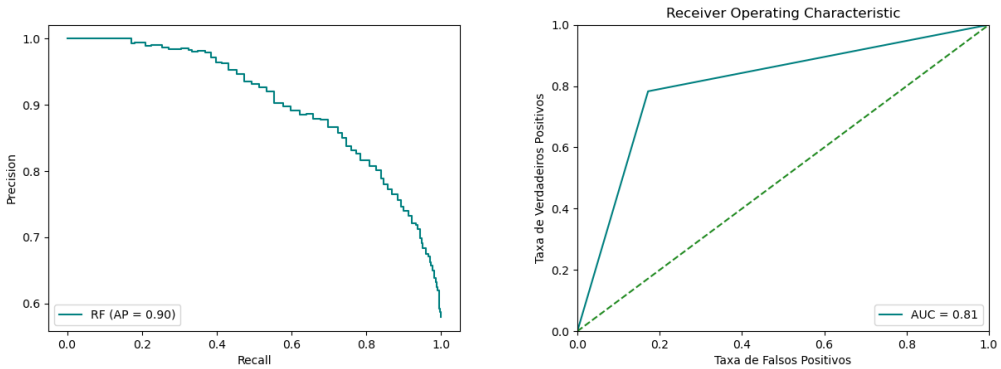
Por último, de maneira a complementar a análise dos modelos desenvolvidos, traçaram-se as curvas PR e ROC dos classificadores SVM e RF, encontrando-se nas figuras 5.3 e 5.4 respetivamente. Em relação às curvas PR (Figuras 5.3a e 5.4a), quanto mais próximas do (1,1), melhor o comportamento do classificador. Quanto às curvas ROC (Figuras 5.3b e 5.4b), uma curvatura mais próxima do (0,1) representa um melhor classificador e a linha a tracejado significa um classificador aleatório. Dito isto, podemos verificar que em ambas as curvas o RF apresenta visivelmente melhores resultados, garantindo-nos que para além da elevada *accuracy* obtida (81 %) é também um bom classificador.



(a) Curva PR do classificador SVM utilizando o método de normalização MinMaxScaler.

(b) Curva ROC do classificador SVM utilizando o método de normalização MinMaxScaler.

Figura 5.3: Curvas de avaliação de desempenho do classificador SVM para a maior *accuracy* obtida.



(a) Curva PR do classificador RF utilizando o método de normalização RobustScaler.

(b) Curva ROC do classificador RF utilizando o método de normalização RobustScaler.

Figura 5.4: Curvas de avaliação de desempenho do classificador RF para a maior *accuracy* obtida.

Capítulo 6

Conclusões

A neurofibromatose tipo 1 (NF1) é uma doença hereditária associada a perturbações do desenvolvimento neurológico sendo um dos impactos desta doença as alterações da neuroplasticidade, incluindo a do córtex visual. É essencial a avaliação da neuroplasticidade como método de diagnóstico uma vez que as correções das alterações podem trazer benefícios aos doentes. Uma das maneiras de realizar esta avaliação é através dos potenciais evocados visuais (VEP) sob a forma de potenciação da resposta seletiva a estímulos (SRP). Assim, foi através da necessidade de ferramentas capazes de um avaliação com a maior fiabilidade e rapidez possível, e que consigam encontrar características não visíveis ao olho humano, que surgiu o presente trabalho. De facto, o objetivo principal desta tese foi perceber a possibilidade da utilização de técnicas de *machine learning* para a correta classificação do tipo de estímulo presente num segmento de EEG, novel ou familiar.

Começou-se por realizar uma preparação apropriada do sinal em estudo através de diversas técnicas de pré-processamento. Inicialmente a frequência do sinal foi ajustada para 10 kHz para permitir uma uniformidade no processo de análise, seguiram-se a extração das *epochs* através do cruzamento das marcas temporais de cada sinal com as marcas temporais dos estímulos presentes e a filtragem do sinal com recurso a um filtro *Butterworth* de ordem 8 passa-banda 5-48 Hz.

Para ser possível a classificação do sinal como novel ou familiar, foi necessária a seleção de um conjunto de *features* do sinal que representassem os fatores diferenciadores de cada tipo de estímulo. Aqui dois conjuntos de *features* foram utilizados, um sendo os valores obtidos após a aplicação do método de Welch ao sinal e o outro

sendo os valores do desvio padrão, mediana, variância, assimetria e curtose. Devido ao elevado número de *features* presentes no conjunto 1, tornou-se necessária uma redução de maneira a eliminar *features* redundantes, para tal recorreu-se ao *truncated SVD*.

Três algoritmos de *machine learning* foram utilizados: Support-Vector Machine, Random Forest e Artificial Neural Network. Para cada algoritmo testou-se qual o conjunto de dados e o melhor método de normalização dos dados, com uma divisão dos dados de 75 % treino e 25 % teste. Deste teste resultou que o algoritmo que obtinha melhores resultados, 81 %, foi o Random Forest utilizando o conjunto 2 e o RobustScaler. Para avaliar a robustez dos resultados aplicou-se *cross validation* com 3 tipos de *folds* diferentes, e os resultados mantiveram-se muito próximos dos obtidos anteriormente. Portanto, conclui-se que o trabalho obtido conseguiu provar que é possível a utilização de métodos de *machine learning* como ferramenta auxiliar de diagnóstico da neuroplasticidade.

Assim, embora os métodos utilizados ao longo deste trabalho tenham obtido bons resultados, poderiam ser melhorados e estudados outros que por ventura acrescentassem mais informação útil para a classificação do sinal. Embora a quantidade de dados em análise fosse elevada, uma maior diferenciação de sujeitos aumentaria a robustez e fiabilidade dos resultados.

Referências

[1] V. P. Oikonomou, G. Liaros, K. Georgiadis, E. Chatzilari, K. Adam, S. Nikolopoulos, and I. Kompatsiaris, "Comparative evaluation of state-of-the-art algorithms for SSVEP-based BCIs," no. January, pp. 1–33, 2016. [Citado nas páginas 1, 13, 14, 15, 17, 23, 24 e 32]

[2] E. Cerri, C. Fabiani, C. Criscuolo, and L. Domenici, "Chapter 7," vol. 1695, pp. 69–80, 2018. [Citado na página 1]

[3] S. Marenna, V. Castoldi, R. D'Isa, C. Marco, G. Comi, and L. Leocani, "Semi-invasive and non-invasive recording of visual evoked potentials in mice," *Documenta Ophthalmologica*, vol. 138, no. 3, pp. 169–179, 2019. [Citado na página 1]

[4] J. Dorszewska, W. Kozubski, W. Waleszczyk, M. Zabel, and K. Ong, "Neuroplasticity in the Pathology of Neurodegenerative Diseases," *Neural Plasticity*, vol. 2020, 2020. [Citado na página 1]

[5] M. F. Bear, "A synaptic basis for memory storage in the cerebral cortex," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 93, no. 24, pp. 13453–13459, 1996. [Citado na página 3]

[6] G. Zlotnik and A. Vansintjan, "Memory: An Extended Definition," *Frontiers in Psychology*, vol. 10, no. November, pp. 1–5, 2019. [Citado na página 3]

[7] E. Camina and F. Güell, "The neuroanatomical, neurophysiological and psychological basis of memory: Current models and their origins," *Frontiers in Pharmacology*, vol. 8, no. JUN, pp. 1–16, 2017. [Citado na página 4]

[8] T. Huff, N. Mahabadi, and P. Tadi, "Neuroanatomy, Visual Cortex," *StatPearls*, jul 2021. [Citado na página 5]

[9] V. S. Pelak, "Disorders of Higher Cortical Visual Function," *Liu, Volpe, and Galetta's Neuro-Ophthalmology*, pp. 341–364, jan 2019. [Citado na página 5]

[10] A. D. Legatt, "Evoked Potentials," *Encyclopedia of the Neurological Sciences*, vol. 2, no. 1974, pp. 228–231, 2014. [Citado nas páginas 6 e 7]

[11] L. Sörnmo and P. Laguna, "Chapter 4 - Evoked Potentials," in *Bioelectrical Signal Processing in Cardiac and Neurological Applications* (L. Sörnmo and P. Laguna, eds.), Biomedical Engineering, pp. 181–336, Burlington: Academic Press, 2005. [Citado nas páginas 6 e 7]

[12] P. Walsh, N. Kane, and S. Butler, "The clinical role of evoked potentials," *Neurology in Practice*, vol. 76, no. 2, pp. 16–22, 2005. [Citado na página 7]

[13] J. N. V. Hoeve, R. J. Munger, C. J. Murphy, and T. M. Nork, *Emerging Electrophysiological Technologies for Assessing Ocular Toxicity in Laboratory Animals*, pp. 123–157. Totowa, NJ: Humana Press, 2013. [Citado na página 8]

[14] D. J. Creel, "Visually evoked potentials," *Handbook of Clinical Neurology*, vol. 160, pp. 501–522, 2019. [Citado nas páginas 8 e 9]

[15] C. Liu, Y. Zhang, W. Tang, B. Wang, B. Wang, and S. He, "Evoked potential changes in patients with Parkinson's disease," *Brain and Behavior*, vol. 7, no. 5, pp. 1–8, 2017. [Citado nas páginas 8, 12 e 14]

[16] A. M. Firan, "Visual evoked potential in the early diagnosis of glaucoma. Literature review," *Romanian Journal of Ophthalmology*, vol. 64, no. 1, pp. 15–20, 2020. [Citado na página 8]

[17] R. Amini Vishteh, A. Mirzajani, E. Jafarzadehpur, and A. Taghieh, "Evaluation of visual evoked potential binocular summation after corneal refractive surgery," *Documenta Ophthalmologica*, vol. 140, no. 2, pp. 181–188, 2020. [Citado na página 9]

[18] A. Azimi, S. Bonakdaran, J. Heravian, P. Layegh, N. Yazdani, and M. Alborzi, "Pattern visual evoked potential in hypothyroid patients," *Documenta Ophthalmologica*, vol. 138, no. 2, pp. 77–84, 2019. [Citado na página 9]

[19] A. Corduneanu, V. Chişca, N. Ciobanu, and S. Groppa, "Evaluation of visual pathways using visual evoked potential in patients with diabetic retinopathy," *Romanian Journal of Ophthalmology*, vol. 63, no. 4, pp. 367–371, 2019. [Citado na página 9]

[20] S. F. Cooke and M. F. Bear, "Stimulus-selective response plasticity in the visual cortex: An assay for the assessment of pathophysiology and treatment of cognitive impairment associated with psychiatric disorders," *Biological Psychiatry*, vol. 71, no. 6, pp. 487–495, 2012. [Citado na página 9]

[21] D. P. Montgomery, D. J. Hayden, F. A. Chaloner, S. F. Cooke, and M. F. Bear, "Stimulus-Selective Response Plasticity in Primary Visual Cortex: Progress and Puzzles," *Frontiers in Neural Circuits*, vol. 15, no. January, pp. 1–18, 2022. [Citado na página 9]

[22] S. F. Cooke, R. W. Komorowski, E. S. Kaplan, J. P. Gavornik, and M. F. Bear, "Visual recognition memory, manifested as long-term habituation, requires synaptic plasticity in V1," *Nature Neuroscience*, vol. 18, no. 2, pp. 262–271, 2015. [Citado nas páginas 10 e 31]

[23] M. Kulin, T. Kazaz, E. De Poorter, and I. Moerman, "A survey on machine learning-based performance improvement of wireless networks: PHY, MAC and network layer," *Electronics (Switzerland)*, vol. 10, no. 3, pp. 1–64, 2021. [Citado nas páginas 11, 12, 18, 19, 20, 21 e 22]

[24] J. Thomas, T. Maszczyk, N. Sinha, T. Kluge, and J. Dauwels, "Deep Learning-based Classification for Brain-Computer Interfaces," pp. 234–239, 2017. [Citado nas páginas 11, 14, 15, 17 e 23]

[25] A. de Cheveigné and I. Nelken, "Filters: When, Why, and How (Not) to Use Them," *Neuron*, vol. 102, no. 2, pp. 280–293, 2019. [Citado nas páginas 12 e 13]

[26] T. Dinh, T. Nguyen, H.-P. Phan, V. Dau, D. Dao, and N.-T. Nguyen, "Physical Sensors: Thermal Sensors," *Reference Module in Biomedical Sciences*, jan 2021. [Citado na página 12]

[27] J. A. Montalvo-aguilar and I. B. A. Ramírez-garcía, "Evaluación de técnicas de filtrado aplicadas a señales electroencefalográficas simuladas para la detección de Potenciales Evocados Visuales Evaluation of Filtering Techniques applied to simulated Electroencephalogram signals for Visual Evoked Potential De," vol. 12, no. 2, pp. 1–30, 2020. [Citado nas páginas 13 e 14]

[28] Z. Gao, X. Sun, M. Liu, W. Dang, C. Ma, and G. Chen, "Attention-Based Parallel Multiscale Convolutional Neural Network for Visual Evoked Potentials EEG Classification," *IEEE Journal of Biomedical and Health Informatics*, vol. 25, no. 8, pp. 2887–2894, 2021. [Citado na página 14]

[29] C. L. Yeh, P. L. Lee, W. M. Chen, C. Y. Chang, Y. T. Wu, and G. Y. Lan, "Improvement of classification accuracy in a phase-tagged steady-state visual evoked potential-based brain computer interface using multiclass support vector machine," *BioMedical Engineering Online*, vol. 12, no. 1, pp. 1–17, 2013. [Citado nas páginas 14, 15 e 23]

[30] V. Lopes-dos Santos, H. G. Rey, J. Navajas, and R. Quian Quiroga, "Extracting information from the shape and spatial distribution of evoked potentials," *Journal of Neuroscience Methods*, vol. 296, pp. 12–22, 2018. [Citado nas páginas 14, 15 e 17]

[31] J. Castelhano, J. Rebola, B. Leitão, E. Rodriguez, and M. Castelo-Branco, "To Perceive or Not Perceive: The Role of Gamma-band Activity in Signaling Object Percepts," *PLoS ONE*, vol. 8, no. 6, pp. 35–37, 2013. [Citado na página 14]

[32] J. Castelhano, P. Tavares, S. Mouga, G. Oliveira, and M. Castelo-Branco, "Stimulus dependent neural oscillatory patterns show reliable statistical identification of autism spectrum disorder in a face perceptual decision task," *Clinical Neurophysiology*, vol. 129, no. 5, pp. 981–989, 2018. [Citado na página 14]

[33] A. X. Stewart, A. Nuthmann, and G. Sanguinetti, "Single-trial classification of EEG in a visual object task using ICA and machine learning," *Journal of Neuroscience Methods*, vol. 228, pp. 1–14, 2014. [Citado nas páginas 14 e 23]

[34] R. Palaniappan and P. Raveendran, "Single trial VEP extraction using digital filter," *IEEE Workshop on Statistical Signal Processing Proceedings*, no. 2, pp. 249–252, 2001. [Citado na página 14]

[35] B. Benchabane, M. Benkherrat, and S. Djelel, "Statistical method to extract evoked potentials from noise," *E3S Web of Conferences*, vol. 170, pp. 8–11, 2020. [Citado na página 14]

[36] M. D. Rugg, "Event-related/Evoked Potentials," in *International Encyclopedia of the Social & Behavioral Sciences* (N. J. Smelser and P. B. Baltes, eds.), pp. 4962–4966, Oxford: Pergamon, 2001. [Citado na página 14]

[37] V. Gaillet, E. Borda, E. G. Zollinger, and D. Ghezzi, "A machine-learning algorithm correctly classifies cortical evoked potentials from both visual stimulation and electrical stimulation of the optic nerve," *Journal of Neural Engineering*, vol. 18, no. 4, 2021. [Citado nas páginas 15, 17 e 23]

[38] V. Vijean, H. M, S. Yaacob, and M. N. B. Sulaiman, "Application of clustering techniques for visually evoked potentials based detection of vision impairments," *Biocybernetics and Biomedical Engineering*, vol. 34, no. 3, pp. 169–177, 2014. [Citado nas páginas 15, 17 e 23]

[39] M. I. Vanegas, M. F. Ghilardi, S. P. Kelly, and A. Blangero, "Machine learning for EEG-based biomarkers in Parkinson's disease," in *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 2661–2665, 2018. [Citado nas páginas 15 e 23]

[40] S. N. Carvalho, T. B. Costa, L. F. Uribe, D. C. Soriano, G. F. Yared, L. C. Coradine, and R. Attux, "Comparative analysis of strategies for feature extraction and classification in SSVEP BCIs," *Biomedical Signal Processing and Control*, vol. 21, pp. 34–42, 2015. [Citado nas páginas 15, 17 e 23]

[41] S. Sarraf, "EEG-Based Movement Imagery Classification Using Machine Learning Techniques and Welch's Power Spectral Density Estimation," *American Scientific Research Journal for Engineering, Technology, and Sciences*, vol. 33, no. 1, pp. 124–145, 2017. [Citado na página 15]

[42] M. Kuhn and K. Johnson, *Applied predictive modeling*. 2013. [Citado na página 16]

[43] M. Kuhn and K. Johnson, *Feature Engineering and Selection*. 2019. [Citado na página 16]

[44] A. S. Struchtrup, D. Kvaktun, and R. Schiffers, "Comparison of feature selection methods for machine learning based injection molding quality prediction," *AIP Conference Proceedings*, vol. 2289, no. November, 2020. [Citado na página 16]

[45] R. Agarwal, "The 5 Feature Selection Algorithms every Data Scientist should know | by Rahul Agarwal | Towards Data Science," 2019. [Citado nas páginas 16 e 17]

[46] A. Müller C. and S. Guido, *Introduction to Machine Learning with Python*. O'Reilly Media, Inc, 2016. [Citado nas páginas 19, 20, 21, 22, 23, 24, 25, 26 e 27]

[47] D. Kim, C. Yeon, and K. Kim, "Development and experimental validation of a dry non-invasive multi-channel mouse scalp EEG sensor through visual evoked potential recordings," *Sensors (Switzerland)*, vol. 17, no. 2, pp. 18–22, 2017. [Citado na página 32]

[48] V. T. Sai Sandeep Raju and M. Belwal, "Driver Drowsiness Detection," in *Lecture Notes on Data Engineering and Communications Technologies*, vol. 58, pp. 975–983, 2021. [Citado na página 32]