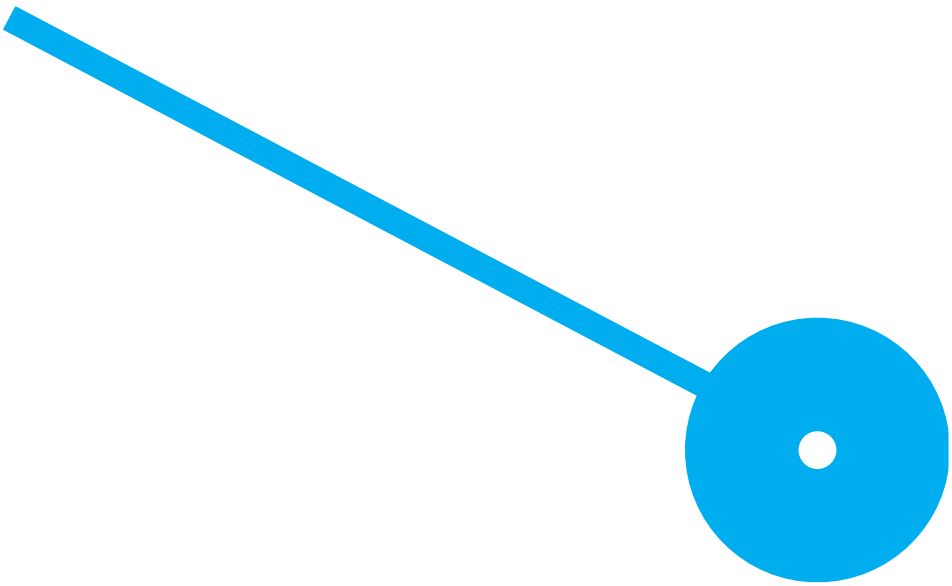
ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

MESTRADO
Engenharia Informática

Abordagem Híbrida para Classificação da Doença de Parkinson através de Voz

Luís Pedro Magalhães da Silva

07/2024



ESCOLA
SUPERIOR
DE TECNOLOGIA
E GESTÃO
POLITÉCNICO
DO PORTO

MESTRADO
ENGENHARIA INFORMÁTICA

Abordagem Híbrida para Classificação da Doença de Parkinson através de Voz

Luís Pedro Magalhães da Silva
8220025

Orientador

Prof. Doutor João Ricardo Martins Ramos

Dissertação apresentada para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Informática pela Escola Superior de Tecnologia e Gestão do Instituto Politécnico do Porto.

07/2024

Declaração de Integridade

Eu, Luís Pedro Magalhães da Silva, estudante nº 8220025, do Mestrado de Engenharia Informática, da Escola Superior de Tecnologia e Gestão do Instituto Politécnico do Porto, declaro que não fiz plágio nem auto-plágio, pelo que o trabalho intitulado "Abordagem Híbrida para a Classificação da Doença de Parkinson através da voz"é original e da minha autoria, não tendo sido usado previamente para qualquer outro fim. Mais declaro que todas as fontes usadas estão citadas, no texto e na bibliografia final, segundo as regras de referenciação adotadas na instituição.

Agradecimentos

Como nota da felicidade por mim expressa, quero deixar umas notas de agradecimento às diversas pessoas que me ajudaram ao longo de todo o meu percurso académico, nomeadamente Professores e Colegas.

Primeiramente, quero agradecer de forma mais específica, ao Professor Doutor João Ricardo Martins Ramos pela sua inestimável assistência no fornecimento de *insights*, nomeadamente na escolha do tema, bem como em conselhos durante a fase de revisão da literatura, bem como no fornecimento de *feedback* durante o desenvolvimento, bem como na descrição dos passos a realizar.

De seguida, gostaria também de deixar um agradecimento especial aos meus colegas de Mestrado e amigos com os quais tive a oportunidade de partilhar experiências realizadas nesta dissertação, bem como através dos quais foi possível obter bastante conhecimento.

Por fim, gostaria de expressar a minha mais profunda gratidão a todos aqueles que me ajudaram a manter-me motivado e inspirado durante esta dissertação, sendo que nem sempre é um processo fácil, sendo exaustivo e desafiante.

Esta dissertação é dedicada aos Familiares mais próximos, dos quais destaco os meus pais e irmão, cujo apoio, incentivo e o amor será para sempre a minha força, resiliência e abrigo para tudo o que conquisto e poderei conquistar na vida.

Resumo

A doença de Parkinson é a segunda doença neurodegenerativa mais presente, apenas superada pela doença de *Alzheimer*, e atualmente estima-se que apresente uma incidência entre 7 a 10 milhões de pessoas, estando presente em pessoas com uma idade mais avançada, uma vez que raramente acontece antes dos 50 anos.

À medida que a população mundial envelhece, a sua prevalência aumenta de forma diretamente proporcional. Sabe-se que não existe nenhuma forma efetiva de realizar o diagnóstico da doença de *Parkinson*, sendo que o presente estudo representa a possibilidade de ser feito um diagnóstico prévio, através de algoritmos de *Machine Learning* baseados num conjunto de dados da voz.

Como o conjunto de dados adquirido é desbalanceado e apresenta um problema de elevada dimensão, conjunto de *features* bastante numeroso, estudou-se o conjunto de dados em 3 vertentes distintas: *dataset* Completo, *dataset* dividido por género e *dataset* dividido por conjunto de *features*.

Nas 3 divisões do conjunto de dados, estudaram-se diversos algoritmos de forma individual e também se utilizou um *Ensemble*, com a utilização dos diversos classificadores, de forma a tornar o modelo mais robusto.

Nos resultados, obteve-se as melhores métricas no estudo com o *dataset* completo, em que se promoveu um sistema híbrido de classificação com a utilização de *Synthetic Minority Oversampling Technique* para balanceamento do *dataset*, seleção de *features* para a redução da dimensionalidade através da importância de *features* do *XGBoost* e *Ensemble Stacking* com *Random Forest*, *Gradient Boosting*, *Support Vector Machine* e *K-Nearest Neighbors* como classificadores base e *XGBoost* como classificador meta, sendo que o resultado apresentou 98.7% de *accuracy*.

Os resultados indicam que a utilização de técnicas de *Machine Learning* baseadas num conjunto de dados da voz pode ser uma boa possibilidade para a deteção prévia da doença de Parkinson, permitindo desta forma, um tratamento mais especializado e eficaz para o paciente.

Palavras-chaves: *Parkinson*, *Machine Learning*, *SMOTE*, *Feature Selection*, *ENSEMBLE*, *Classification*.

Abstract

Parkinson's disease is the second most common neurodegenerative disease, only surpassed by Alzheimer's disease, and it is currently estimated that it has an incidence of between 7 and 10 million people, occurring in older people (it rarely happens before the age of 50).

As the world's population ages, its prevalence increases in direct proportion. It is known that there is no effective way of diagnosing Parkinson's Disease, and this study represents the possibility of making a prior diagnosis using Machine Learning algorithms based on a set of voice data.

As the acquired dataset is unbalanced and presents a high-dimensional problem (a very large set of features), the dataset was studied in 3 different ways: complete dataset, dataset divided by gender and dataset divided by set of features.

In the 3 divisions of the dataset, various algorithms were studied individually and an Ensemble was also used, using the various classifiers, in order to make the model more robust.

The best results were obtained in the study with the complete dataset, in which a hybrid classification system was promoted using the Synthetic Minority Oversampling Technique to balance the dataset, selection of features for dimensionality reduction through the importance of features from XGBoost and Ensemble Stacking with Random Forest, Gradient Boosting, Support Vector Machine and K-Nearest Neighbors as Base classifiers and XGBoost as Meta classifier, and the result was 98.7% accuracy.

The results indicate that the use of *Machine Learning* techniques based on a voice data set may be a good possibility for the early detection of Parkinson's Disease, thus allowing for a more specialized and effective treatment for the patient.

***Keywords*:** *Parkinson*, *Machine Learning*, *SMOTE*, *Feature Selection*, *Ensemble*, *Classification*

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

ACC *Accuracy*.

AdaBoost *Adaptative boosting*.

ANOVA *Analysis of variance*.

AUC *Area under the ROC curve value*.

BT *Bagged tree*.

CRISP-DM *Cross industry standard process for data mining*.

CSV *Comma-separated values*.

CV *Cross-validation*.

DBS *Deep brain stimulation*.

DFA *Detrended fluctuation analysis*.

DP Doença de *Parkinson*.

DT *Decision tree*.

ECFS *Eigenvector centrality feature selection*.

EMD *Empirical mode decomposition*.

EMG Eletromiografia.

FN Falso negativo.

FP Falso positivo.

GB *Gradient boosting*.

GNE *Glottal to noise excitation*.

GQ *Glottis quotient*.

GUI *Graphical user interface*.

IA Inteligência artificial.

L-Dopa *Levodopa.*

LD *Linear descriminant.*

LOSO *Leave one subject out.*

LR *Logistic regression.*

MAMa *Minimum average maximum tree.*

MFCCs *Mel-frequency cepstral coefficients.*

ML *Machine learning.*

MLP *Multilayer perceptron.*

MRI *Magnetic resonance imaging.*

mRMR *Minimum redundancy-maximum relevance.*

NB *Naive Bayes.*

NN *K-Nearest neighbors.*

PCA *Principal component analysis.*

PPE *Pitch period entropy.*

RBF *Radial based function.*

RF *Random forest.*

RFE *Recursive feature elimination.*

RL *Reinforcement learning.*

ROC *Receiver operating characteristics.*

RPDE *Recurrence period density entropy.*

SL *Supervised learning.*

SMOTE *Synthetic minority oversampling technique.*

SVD *Singular value decomposition.*

SVM *Support vector machine.*

TAC Tomografia Computorizada.

TQWT *Tunable wavelet transform approach.*

UL *Unsupervised learning.*

VFER *Vocal fold excitation ratio.*

VIF Variância inflacionária de fator.

VN Verdadeiro negativo.

VP Verdadeiro positivo.

WT *Wavelet transform*.

XGBoost *Extreme gradient boost*.

Capítulo 1

Introdução

O Primeiro Capítulo apresenta um breve enquadramento do tema em estudo, ou seja, uma contextualização e posterior enquadramento, onde se apresentam os diversos tópicos fundamentais para a compreensão do trabalho proposto, bem como a motivação para o estudo, os objetivos delineados e a metodologia de investigação aplicada. O capítulo subsequente descreve a estrutura da dissertação e apresenta uma visão geral da preparação da dissertação, detalhando cada parte.

1.1 Contextualização

A aplicação da *Inteligência artificial (IA)* na área da saúde tem demonstrado um impacto significativo, uma vez que permite uma cooperação entre a saúde e a tecnologia para uma finalidade comum, demonstrando-se através de diversos avanços, tais como em sistemas de tratamento de diagnósticos, sistemas de registos de pacientes, sistemas de gestão de ficheiros, sistemas de libertação de Fármacos, Genética, Imagem Médica, entre outros [2] [3]. A Figura 1.1, apresenta um esquema de possíveis aplicações da *IA* na área da saúde, nomeadamente sistemas de tratamento de diagnósticos, sistemas de registos de pacientes, e sistemas de libertação de Fármacos. o sentido das setas presentes na imagem, representa o contexto ideal de uma aplicação informático de *IA*, onde se começa por um sistema de registo de pacientes, dos quais se obtém a informação e histórico através do sistema de Gestão de ficheiros. Com base na informação e nos algoritmos de *IA*, é possível a toma de decisão em sistemas de libertação de fármacos e o diagnóstico e posterior tratamento.

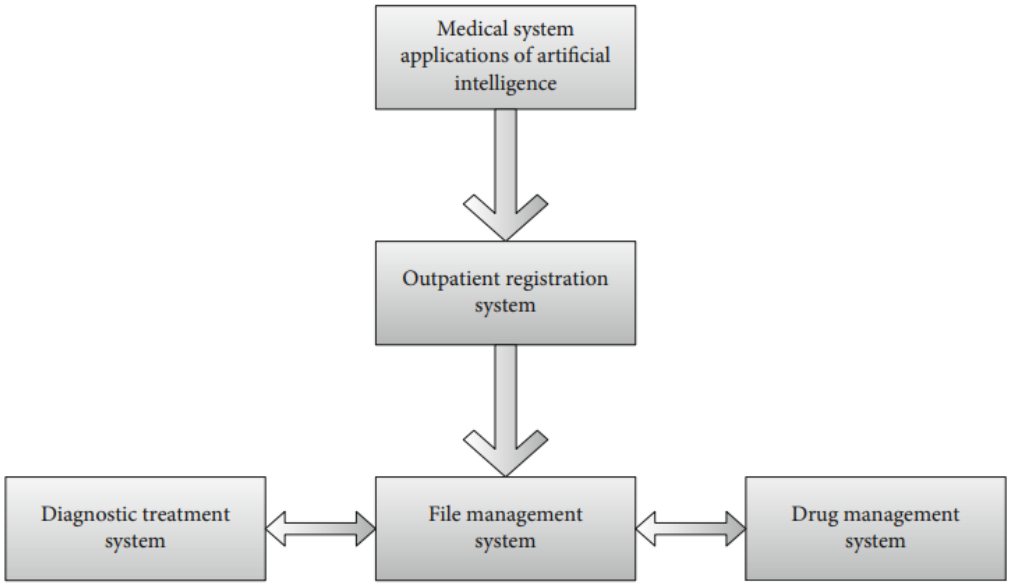


Figura 1.1: Aplicações da Inteligência Artificial na Saúde [2].

No contexto do presente trabalho, destaca-se a utilização de algoritmos de *Machine learning (ML)* para a realização de diagnósticos prévios de doenças, tais como através da interpretação de imagens médicas, tais como exames de *Magnetic resonance imaging (MRI)*, de Tomografia Computorizada (TAC), radiografias, dados provenientes de Medições quotidianas dos pacientes, dados fisiológicos, bem como de dados elétricos provenientes do funcionamento vital do organismo [3] [4].

Os estudos ao nível da *IA* na saúde têm evoluído bastante, sendo que através de dados provenientes da última década (2013, 2014, 2015 e 2016) retirado do *PubMed*, se verifica uma evolução ao nível de estudo de Diagnóstico e Terapêutica de diversas doenças (Figura 1.2) [3].

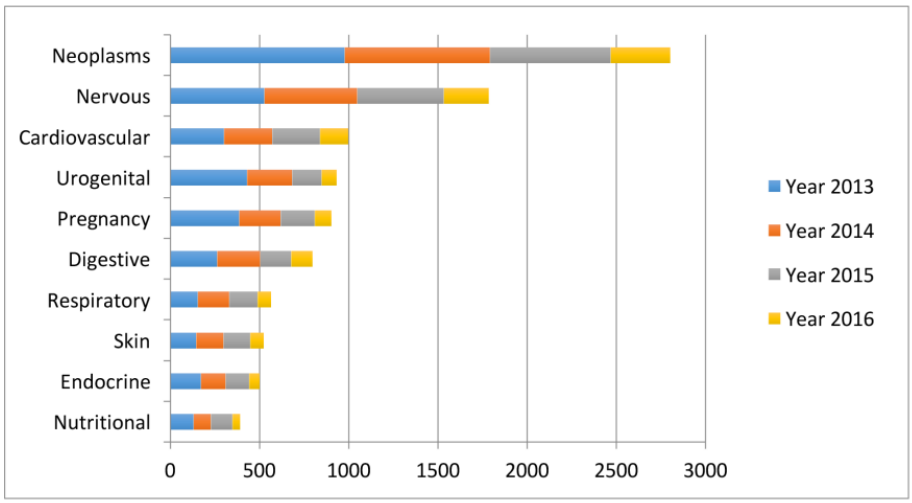


Figura 1.2: Quantidade de estudos por Doença na Saúde através da aplicação de Inteligência Artificial [3].

Da Figura 1.2, verifica-se que o Estudo do sistema nervoso é bastante frequente, apresenta-se em segundo lugar, apenas menos recorrente do que *Neoplasms*. No presente trabalho, o estudo é relativo a uma doença neurodegenerativa (*Doença de Parkinson (DP)*), pertencente ao Sistema Nervoso [3].

A *DP* é a segunda doença neurodegenerativa mais presente, apenas sendo superada pela doença de *Alzheimer*, e sendo esta doença com uma maior incidência em pessoas com uma idade mais avançada (raramente acontece antes dos 50 anos), à medida que a população mundial envelhece a sua prevalência aumenta de forma diretamente proporcional. Atualmente, verifica-se que a estimativa de existência de pessoas diagnosticadas com a Doença de *Parkinson* é de um valor de entre os 7 a 10 milhões de pessoas [5]. No contexto nacional, a Doença também se tem prosperado, sendo que em Portugal, num estudo publicado no ano de 2017, se estima uma existência de 180 casos em cada 100 mil habitantes, resultando numa estimativa total de 18 a 20 mil pessoas [5] [6].

Apesar da sua causa ainda continuar desconhecida, estudos indicam que a sua origem esteja associada a efeitos existentes no Sistema nervoso central [7]. Na sua origem está uma perturbação crónica do Sistema nervoso central, onde os neurónios dopaminérgicos, que apresentam a função de realizar a produção de dopamina são afetados. A dopamina é um neurotransmissor, isto é, uma substância química que faz a transmissão de sinais elétricos entre o cérebro e as restantes partes do corpo, associados a funções de movimento e memória. Trata-se de uma doença progressiva, que não apresenta cura onde o indivíduo acaba por atingir um estado de dependência dos demais, uma vez que apresenta diversas consequências motoras, tais como perda da função muscular que resulta em bradicinesia (lentidão de movimento), rigidez dos membros, postura prejudicada e problemas de marcha e equilíbrio. Contudo, também existem sintomas não-motores que se podem manifestar nas diversas etapas da doença, tais como depressão, ansiedade e demência [7] [8].

Apesar de não existir uma cura para a *DP*, e esta se encontrar longe de ser uma possibilidade, vários esforços têm sido realizados ao nível da Indústria Farmacêutica, bem como ao nível de Cirurgias, nomeadamente ao nível da estimulação elétrica, com a intenção de reduzir os seus sintomas, promovendo uma melhoria na qualidade de vida aos pacientes [9] [10].

No que concerne ao tratamento da DP, não existe um método que possa ser extrapolado e generalizado em todos os portadores da doença [11]. Desta forma, geralmente o tratamento varia de indivíduo para indivíduo, sendo diferente, personalizado e característico de cada utente, bem como diferente para os vários estágios da doença. Apesar deste pressuposto, um frequente primeiro contexto da prática do tratamento é a utilização de *Levodopa (L-Dopa)*, medicamento que tem por objetivo o retardamento da progressão dos efeitos motores que envolve a doença, através da redução rápida de sintomas como tremores e rigidez muscular [11].

Atualmente, o diagnóstico mais utilizado é a visualização do desempenho motor do paciente, contudo, acredita-se sempre que um diagnóstico mais prévio da Doença de *Parkinson* permita uma maior previsão do tratamento eficaz, bem como uma melhor capacidade de tratamento do utente. Posto isto, torna-se necessário o investimento em mecanismos que permitam um diagnóstico prévio da *DP*.

Contudo para este processo de diagnóstico de doença ser efetivo é necessário um processo

de pré-processamento e tratamento de dados intenso, uma vez que os dados na área de medicina são frequentemente poucos e bastante desbalanceados.

1.2 Motivação e objetivo principal

Em 2015, estudos relataram que aproximadamente 177.000 pessoas morreram devido à doença de *Parkinson*, a segunda doença neurodegenerativa mais prevalecente e que causa complicações significativas, em grande parte pela ausência de um diagnóstico precoce eficaz [12]. Tendo isto em conta, o diagnóstico e o prognóstico da doença nas suas fases iniciais são de extrema importância, pois permitem um tratamento mais eficaz.

Acredita-se que dados da voz podem ser uma maneira possível para este diagnóstico, uma vez que os pacientes com *DP* apresentam alterações ao nível da voz, tais como, redução da intensidade vocal, monotonia na entoação, fala arrastada, hesitações e dificuldades na articulação das palavras [13].

De acordo com a literatura, cerca de 90% dos indivíduos com *DP* apresentam alterações na fala nas fases iniciais da doença, sendo este um dos principais sintomas que pode ser utilizado para o diagnóstico [14][15].

Sabe-se que não existe uma forma efetiva de diagnosticar a doença de *Parkinson*; contudo, diversas pesquisas realizadas por investigadores têm explorado várias abordagens com o intuito de aprimorar este diagnóstico. Algoritmos de *Machine Learning* têm sido bastante utilizados por investigadores na área da Medicina e de Biomédica, como alternativa ou como método de apoio ao diagnóstico e tratamento de doenças, dado que podem ser utilizados para problemas de classificação e apresentam resultados precisos e confiáveis.

Posto isto e tendo em conta os pressupostos definidos nos parágrafos prévios, o principal objetivo da presente dissertação passa pelo estudo da possibilidade da realização de um diagnóstico prévio da Doença de *Parkinson* através de um conjunto de dados da fala, recorrendo a algoritmos de *Machine Learning*. No que concerne ao processo em si, tem-se como objetivo a realização do levantamento do estado de arte relativo a projetos em que se utilizou o mesmo conjunto de dados com o mesmo objetivo, bem como o processo em si e os algoritmos utilizados. Ainda neste sentido pretende-se a elaboração de um processo que englobe a análise dos dados existentes, avaliação dos dados e do desbalanceamento existente, pré-processamento dos dados, bem como o processo da seleção das melhores *features*. Espera-se ainda a utilização de diversos Modelos de *Machine learning*, bem como a respetiva otimização.

1.3 Metodologia de investigação

Durante o processo de desenvolvimento do presente trabalho, optou-se por uma estrutura de trabalho denominada por *Cross industry standard process for data mining (CRISP-DM)*, uma metodologia independente do domínio da aplicação.

Este tipo de metodologia é usualmente utilizada em contexto profissional, sendo vista como a abordagem mais comum aos diversos problemas, que podem ser resolvidos através de *Data Mining* [16]. A Figura 1.3, apresenta um esquema da *framework CRISP-DM*, bem como do seu processo iterativo com uma sequência de atividades.

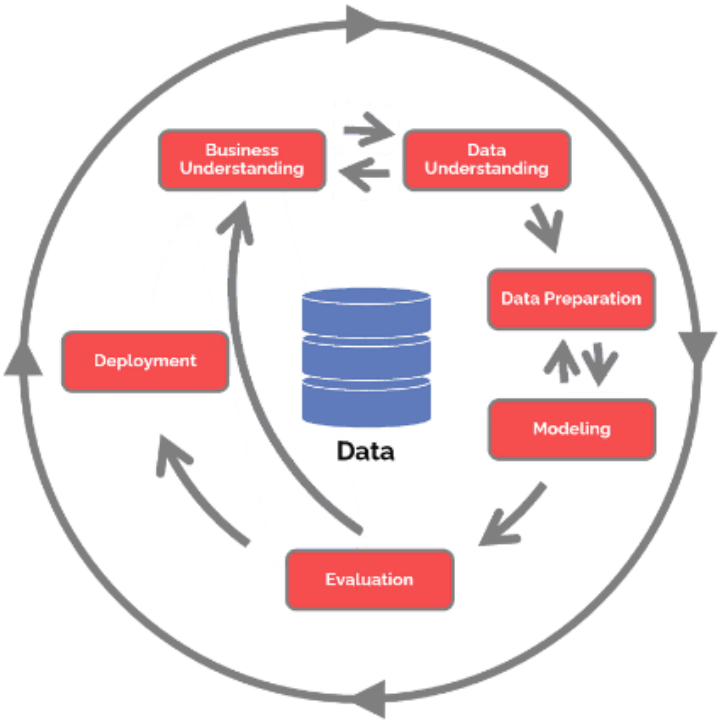


Figura 1.3: *Cross Industry Standard Process for Data Mining.*

Neste sentido, o *CRISP-DM* é um processo iterativo com uma sequência de atividades.

- *Business Understanding*;
- *Data Understanding*;
- *Data Preparation*;
- *Modeling*;
- *Evaluation*;
- *Deployment*.

Business understanding:

- Identificação do objetivo do projeto: classificação da presença ou ausência da *DP* com base em características extraídas de conjuntos de dados de voz.
- Conhecimento relativo ao estudo: revisão da literatura médica relacionada com a *DP*, bem como os impactos que a voz pode ter na sua deteção.

Data understanding:

- Recolha de dados de voz relevantes, isto é, de dados presentes em conjuntos de dados públicos ou através da colaboração com instituições médicas ou laboratórios certificados.
- Análise exploratória dos dados para perceber as suas características, identificação de possíveis problemas de qualidade e quantidade dos dados e determinação dos atributos mais relevantes para a classificação da *DP*.

***Data preparation*:**

- Limpeza dos dados, remoção de valores nulos, remoção de *outliers* e dados com erros.
- Realização de etapas de seleção de atributos, identificação e seleção dos atributos mais relevantes para o problema.
- Divisão dos dados em conjuntos de treino e teste para avaliação do desempenho dos modelos.

***Modeling*:**

- Escolha de algoritmos de classificação adequados para o problema em questão, como *Logistic regression (LR)*, *Support vector machine (SVM)*, entre outros.
- Avaliação do uso de Técnicas *Ensemble*, com o intuito de melhoria dos resultados obtidos.
- Treino de vários modelos de *ML* usando o conjunto de dados de treino.
- Otimização dos Hiperparâmetros dos modelos para otimização do desempenho.

***Evaluation*:**

- Avaliação do desempenho dos modelos usando métricas apropriadas para problemas de classificação.
- Comparação do desempenho de vários modelos.

***Deployment*:**

- Desenvolvimento de uma *Graphical user interface (GUI)*, facilitando o acesso e uso pelos profissionais de saúde.

1.4 Estrutura do documento

A presente dissertação retrata a possibilidade de realização de um diagnóstico prévio da *DP* e encontra-se dividida em 6 capítulos, sendo estes capítulos denominados como "Introdução", "A Doença de *Parkinson*", "*Machine Learning*", "O Caso de estudo", "Interface Gráfica"e "Conclusão e Trabalho Futuro", respetivamente.

O presente capítulo é a Introdução, onde se faz uma contextualização do problema que se estuda ao longo do projeto, bem como apresenta os contextos da motivação e a metodologia utilizada.

No segundo capítulo, "A Doença de *Parkinson*", apresenta-se uma explicação do estudo efetuado desta doença, nomeadamente os seus fundamentos, causas, principais sintomas, bem como meios de diagnóstico e tratamento. Ainda neste capítulo apresenta-se uma revisão da literatura da doença, enumera-se estudos em que se utilizou fundamentos de *ML* para o diagnóstico da *DP* com o mesmo conjunto de dados utilizado no presente trabalho.

No Terceiro Capítulo contextualiza-se fundamentos teóricos de *Machine Learning*, dando-se um contexto dos diferentes tipos de algoritmos de *Machine Learning*, bem como as suas principais aplicações. Ainda neste capítulo, estrutura-se as diferentes fases de um processo de *Machine Learning*, bem como o que se realiza em cada uma destas fases. Por último, apresentam-se fundamentos teóricos e explicações acerca dos diferentes métodos e técnicas de preparação dos dados, bem como o seu intuito.

No Quarto capítulo, denominada como "O Caso de Estudo", apresenta-se e descreve-se o caso de estudo, os dados utilizados e onde se faz uma análise aos resultados obtidos dos diferentes métodos utilizados: conjunto de dados completo e conjunto de dados dividido por género. Ainda neste capítulo faz-se a análise a diversos algoritmos e a diversos métodos de seleção de *features*.

No quinto capítulo, aborda-se a interface gráfica para a aplicação de *ML*. Neste capítulo, explora-se uma interface amigável e interativa que permita aos utilizadores interagirem com modelos de *ML* de uma forma intuitiva. Essa interface gráfica facilita a importação de dados, via *Comma-separated values (CSV)*, a configuração de modelos e a interpretação de resultados, tornando o processo de classificação de algoritmos mais facilitada.

No último capítulo, procede-se com as conclusões finais desta dissertação, a apresentação de limitações do trabalho desenvolvido e sugestões de melhorias futuras ao presente trabalho.

Capítulo 2

Estado de arte

Este capítulo serve como uma introdução completa à doença de *Parkinson*, abordando as suas causas, sintomas e os tratamentos potenciais associados. Pretende-se não só examinar a manifestação clínica desta condição neurológica complexa, mas também explorar as nuances da sua etiologia e as abordagens terapêuticas em desenvolvimento. Ao analisar detalhadamente estes aspetos fundamentais da doença de *Parkinson*, procura-se fornecer uma base sólida para a compreensão e o contexto necessário para as discussões subsequentes sobre o seu diagnóstico, tratamento e investigação científica.

2.1 Doença de *Parkinson*

A Doença de *Parkinson* foi descrita pela primeira vez no ano de 1817 por um Cirurgião Inglês de seu nome *James Parkinson*, no seu estudo denominado por "*An Essay on the Shaking Palsy*". Neste estudo, descreve a doença como uma espécie de paralisia agitante [17]. Devido à importância deste estudo, Jean-Martin Charcot, um neurologista Francês, em homenagem ao estudo supramencionado, deu o seu nome à doença [18].

Atualmente, A Doença de *Parkinson* é a segunda doença neurodegenerativa mais comum, sendo que apenas é superada pela Doença de *Alzheimer* [7]. Além de ser uma doença neurodegenerativa crónica, também é uma doença progressiva que piora ao longo do tempo de vida, que afeta principalmente o sistema motor. Esta doença é caracterizada por tremores, rigidez muscular, lentidão de movimentos e instabilidade postural.

No que diz respeito à epidemiologia da doença, observa-se que as taxas de incidência variam de acordo com a localização geográfica. Estima-se que no Continente Europeu exista entre 257 e 1400 casos por cada 100 mil habitantes, enquanto que no conceito nacional, verifica-se que em Portugal, numa amostra populacional com indivíduos acima dos 50 anos de idade, existe uma prevalência da Doença de *Parkinson* de 180/100000 habitantes, onde existe um pico por voltas dos 70 anos com uma proporção mais prevalecente nos homens em relação às mulheres (3:2) [7].

Como a doença de *Parkinson* é progressiva e crónica, o que significa que os sintomas tendem a piorar ao longo do tempo e sem cura, torna-se necessário a existência de tratamentos que possam aliviar significativamente os sintomas e melhorar a qualidade de vida dos pacientes.

2.1.1 Causa

A causa exata da Doença de *Parkinson* ainda não é compreendida na sua totalidade, mas acredita-se que seja uma combinação de fatores genéticos e fatores ligados ao ambiente. Contudo, na sua maioria, os casos de doença de *Parkinson* existentes ocorrem de forma esporádica, isto é, não estão diretamente ligados a fatores genéticos ou ambientais conclusivos.

Aliado ao fator genético, verifica-se que a doença de *Parkinson* tem uma prevalência maior em adultos acima dos 50 anos e que geralmente é mais comum em homens do que em mulheres e que a progressão da doença pode ser mais lenta nas mulheres, comparativamente com os homens. [19].

Apesar da sua origem ainda ser incerta, a fisiopatologia da doença de *Parkinson* tem sido bastante estudada e é uma doença que integra o grupo das sinucleinopatias, associada com a acumulação da proteína alfa-sinucleína de forma anómala no tecido neuronal, o que resulta num complexo denominado por "*Lewy Bodies*", que está associado aos sinais neuro-imagiológicos do processo de morte neuronal, isto é, das células nervosas. Este processo de morte neuronal compreende a degeneração progressiva de células nervosas na parte do cérebro que controla o movimento, conhecida como substância negra. A substância negra liberta dopamina, que atua como mensageira entre partes do sistema nervoso e o cérebro para coordenar os movimentos e a regulação do humor. A quantidade de dopamina no cérebro diminui se essas células nervosas morrem. Isto indica que a região do cérebro responsável pelo controle dos movimentos não está a funcionar de forma adequada, resultando em movimentos lentos e irregulares [20]. A Substância Negra, bem como o seu efeito, pode ser visualizado na Figura 2.1.

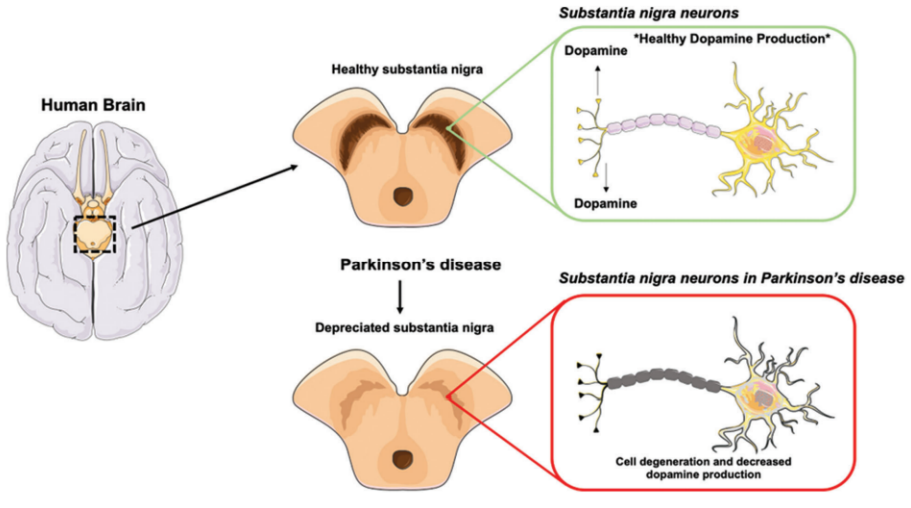


Figura 2.1: Substância Negra [20].

Tendo em conta que esta doença é progressiva e que a sua causa exata ainda é uma incógnita, torna-se necessário os estudos de várias formas de possível diagnóstico, com o intuito de se realizar um tratamento mais eficaz, possibilitando a prevenção e retardamento do progresso da doença.

2.1.2 Sintomas

Embora a Doença de *Parkinson* seja frequentemente associada a manifestações motoras, estes não são os únicos sintomas, bem como as únicas consequências desta doença, uma vez que existem consequências não motoras [21]. Usualmente, tremores em repouso, bradicinesia (movimentos lentos), rigidez e perda de reflexos ao nível de postura são geralmente considerados os sinais mais característicos da Doença de *Parkinson*, e os mais associados com as manifestações da doença, contudo a identificação de sintomas não motores permite também uma melhoria nos cuidados clínicos prestados, monitorização da doença bem como uma melhor compreensão do seu estado evolutivo [7].

Os sintomas da doença de *Parkinson* podem variar de pessoa para pessoa e podem manifestar-se de maneira gradual. Eles geralmente começam de forma leve e pioram com o tempo à medida que a degeneração das células nervosas no cérebro progride. Os principais sintomas estão presentes na Tabela 2.1, onde se faz uma distinção entre Sintomas Motores e Sintomas Não Motores.

Tabela 2.1: Sintomas da Doença de *Parkinson*

Sintomas Motores	**Sintomas Não Motores**
Tremores	Distúrbios do Sono
Rigidez Muscular	Depressão e Ansiedade
Bradicinesia	Diminuição da Capacidade Cognitiva
Instabilidade Postural	Psicose
Dificuldade na Marcha	Apatia
Alterações na Coordenação Motora	Fadiga
Alterações na fala	

Os Sintomas Motores e Não Motores são descritos nos próximos parágrafos:

Tremores: Tremores involuntários, geralmente começando em uma das mãos quando está em repouso é o sintoma mais comum e mais facilmente reconhecível. Estes tremores são unilaterais, involuntários, rítmicos e geralmente ocorrem a uma frequência entre 4 e 6 Hz [7].

Rigidez Muscular: A Rigidez Muscular verifica-se pelo aumento da resistência durante o movimentos dos membros e do pescoço, associado a um fenómeno de dor, levando a desconforto e dificuldade de movimento. Este sintoma, normalmente é avaliado pelos médicos através do movimento passivo dos membros dos pacientes.

Bradicinesia: Refere-se à progressiva lentidão de movimentos e redução da amplitude de movimentos alternados e repetitivos (abrir e fechar da mão, oponência do polegar e indicador, pronação-supinação das mãos ou mesmo o bater repetitivo do calcanhar no chão) [7]. Pode tornar as atividades diárias, como vestir-se ou comer, mais demoradas.

Instabilidade Postural: Este sintoma é um dos mais tardios a ser manifestado, e demonstra a dificuldade em manter o equilíbrio e a postura, o que pode aumentar o risco de quedas.

Dificuldade na Marcha: Pode ocorrer um padrão de caminhar arrastado, com passos curtos e uma postura inclinada para a frente.

Alterações na Coordenação Motora: Dificuldade em realizar movimentos coordenados e precisos. Em indivíduos com Doença de *Parkinson* a postura é um pouco fletida [7].

Alterações na fala: A fala pode tornar-se mais lenta e menos articulada. Caracterizada por uma voz rouca, de volume reduzido, variabilidade de tom restrita (monótono), articulação imprecisa (fala arrastada) e taxa de fala instável.

Alterações do Sono: Pessoas com *Parkinson* podem ter dificuldades em conciliar o sono, apresentar movimentos involuntários durante o sono ou experienciar sonhos vividos.

Depressão e Ansiedade: Estes sintomas podem variar bastante de acordo com a gravidade dos sintomas.

Diminuição da Capacidade Cognitiva: Este fenómeno é observado especialmente em casos avançados da Doença de *Parkinson* ou em pessoas idosas. Inclui limitações no pensamento, assim como dificuldade em encontrar as palavras ou o pensamento correto.

É importante notar que nem todas as pessoas com *Parkinson* experimentam todos estes sintomas, e a gravidade dos sintomas pode variar. Além disso, outros sintomas não relacionados ao movimento podem ocorrer, como alterações cognitivas, depressão, ansiedade e problemas de olfato.

2.1.3 Diagnóstico e tratamento

Apesar de atualmente não existir um teste ou biomarcador definitivo na Doença de *Parkinson*, este pode ser de forma obtido de forma confiável através da avaliação de um Neurologista que esteja treinado no diagnóstico e tratamento de Doenças Neurológicas.

O diagnóstico da doença de *Parkinson* é geralmente feito com base na avaliação clínica dos sintomas pelo médico, através da realização de questionários ao paciente, bem como uma observação neurológica detalhada.

O tratamento visa aliviar os sintomas e melhorar a qualidade de vida do paciente, muitas vezes envolvendo medicamentos, terapia física e ocupacional, e, em alguns casos, cirurgias como a estimulação cerebral profunda.

No que diz respeito ao tratamento da Doença de *Parkinson*, a utilização de medicamentos dopaminérgicos são a base da terapia sintomática para sintomas motores na doença de *Parkinson* [22].

Após a sua descoberta, a levodopa foi o primeiro tratamento sintomático para a doença de *Parkinson*, seguida pela utilização de agonistas da dopamina e inibidores da monoamina oxidase B [22]. A utilização de agonistas da dopamina, que atuam nos recetores de dopamina no corpo estriado, inibidores da monoamina oxidase B e/ou inibidores da catecol-o-metil-transferase têm como intuito a prevenção da degradação periférica da levodopa [22].

A levodopa, também conhecida como L-DOPA, é um precursor natural da dopamina, ou seja, a dopamina é produzida a partir da levodopa no cérebro, um neurotransmissor no cérebro que desempenha um papel fundamental no controlo do movimento, na tentativa

de reverter a diminuição da produção de dopamina consequente da doença de *Parkinson*. Para a obtenção de dopamina, a levodopa, após a passagem da barreira hematoencefálica, entra no cérebro, onde é convertida em dopamina [22][23].

Por outro lado, "*Deep brain stimulation*"é um tipo de terapia mais invasiva, terapia sintomática neuro-cirúrgica segura para pacientes elegíveis com doença avançada, especialmente em casos em que os sintomas não respondem adequadamente aos medicamentos dopaminérgicos ou quando os efeitos colaterais dos medicamentos tornam-se problemáticos [23].

A *Deep brain stimulation (DBS)* consiste na implantação de elétrodos extremamente finos em áreas específicas do cérebro, elétrodos estes que são conectados a um neuroestimulador, que é colocado sob a pele na região do peito ou na parede abdominal. Este dispositivo gera impulsos elétricos que ajudam a modular a atividade neuronal nas áreas-alvo, proporcionando alívio dos sintomas motores associados à doença de *Parkinson* [23]. A título de exemplo, na Figura 2.2, mostra-se a zona específica da colocação dos elétrodos.

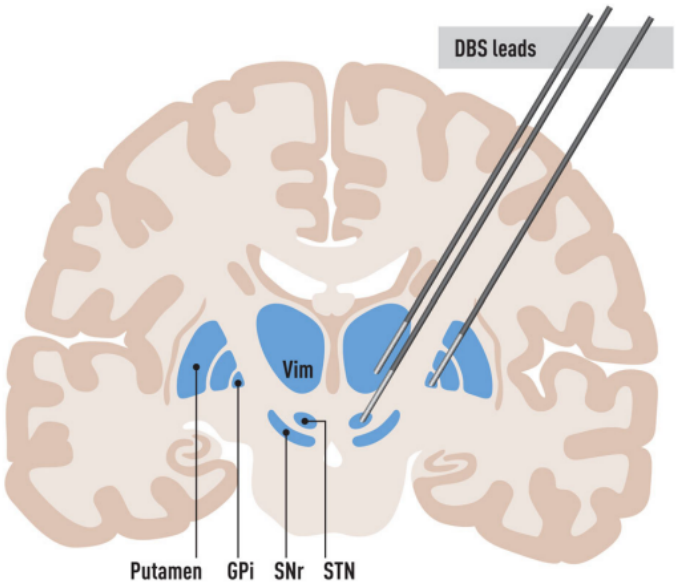


Figura 2.2: Colocação dos elétrodos na *Deep Brain Stimulation*.

2.2 Estudos prévios

O diagnóstico automático da *DP* tem despertado o interesse de muitos como uma forma de permitir um diagnóstico mais preciso e eficiente da *DP*, ao mesmo tempo que reduz a carga de trabalho dos profissionais médicos. Neste sentido, diversos domínios de deteção têm sido utilizados de forma a tentar realizar uma deteção prévia, tais como a deteção a partir de postura/movimento através do uso de *wearable systems*, sistemas de deteção a partir da voz, desenho e escrita, bem como através de sistemas de processamento de imagem médica.

Estudos recentes demonstraram que existem anormalidades na fala, que podem ser utilizadas como um indicador quantificável para o diagnóstico automático e precoce da Doença

de *Parkinson*. Nos estágios iniciais da *DP*, a maioria das pessoas apresenta problemas vocais. Os sinais de voz são sinais oscilatórios não lineares e não estacionários. Como cerca de 90% dos pacientes apresentam anomalias vocais no início da progressão da doença, esses sintomas podem ser úteis na deteção da doença [24].

Mostafa et al.[25] utilizaram um sistema multiagente para selecionar 11 recursos dos 23 recursos, no seu conjunto de dados. Uma vez realizada a etapa de processamento de dados, utilizaram diversos algoritmos para o treino e teste: *Decision tree (DT)*, *Naive Bayes (NB)*, *Multilayer perceptron (MLP)*, *Random forest* (RF) e SVM, antes e após a aplicação do sistema multiagente de seleção de recursos. Na etapa de testes utilizou-se um *Cross-Validation* com 10 *folds*. A nível de resultados mostraram que a utilização do Sistema multiagente para seleção de *features* melhorou a *performance* dos modelos, uma vez que permitiu uma melhor seleção do conjunto de *features*. No conjunto de dados final, ou seja, após a seleção das *features*, obtiveram-se valores acima de 89% de *accuracy* em todos os modelos, sendo que este valor em média foi cerca de 10% superior quando comparado com o conjunto de dados inicial, tal como pode ser verificado na Figura 2.3.

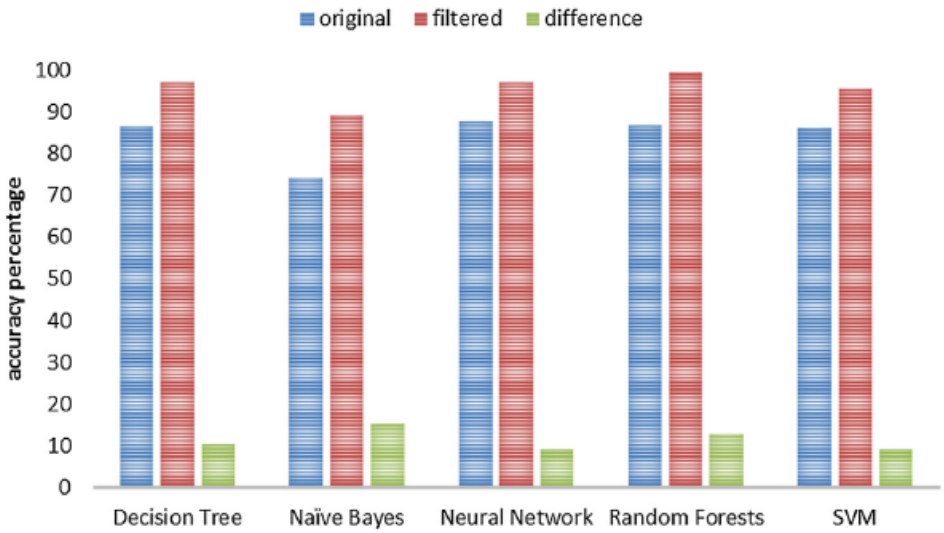


Figura 2.3: Resultados apresentados no estudo [25].

Posto isto, a nível de performance o *NB* foi o que aumentou mais a sua performance tendo em conta a comparação entre conjunto de dados original e conjunto de dados com seleção de features, no entanto o algoritmo que apresentou melhores resultados foi o *RF*, com uma *accuracy* de 99,492%.

Sakar et al. [26] apresentaram um estudo metódico de processamento de sinais de voz para a deteção da doença de *Parkinson*. Os autores extraíram recursos de sinais de voz de 252 pacientes, sendo estes pacientes com ou pacientes sem *DP* usando *Tunable wavelet transform approach (TQWT)*. Os autores concluíram que o desempenho da utilização de *TQWT* é superior a outras abordagens de processamento de sinais de voz de ponta que são utilizados na classificação da doença de *Parkinson*. Neste estudo, *subsets* de *features* são utilizadas como *inputs* de classificadores múltiplos, sendo as previsões destes classificadores utilizadas como *input* do método *Ensemble*.

Neste estudo, na etapa de Pré-processamento, procedeu-se a uma estandardização dos dados, sendo que posteriormente procedeu-se com a seleção das *features* através de *Minimum*

redundancy-maximum relevance (mRMR), onde se obtiveram as *features* mais relevantes e permitiu a redução da dimensionalidade do problema. Uma vez obtidas as *features* finais, procedeu-se à utilização de diversos métodos de classificação, dos quais as previsões foram o *input* de métodos de *Ensemble Voting* e *Ensemble Stacking*.

Şule Yücelbaş [27] devido à prevalência da Doença de *Parkinson* ser superior nos indivíduos do sexo Masculino, propôs a utilização de um *Simple Logistic Hybrid System Based on Greedy Stepwise Search Algorithm (SLGS)*, que através da análise de *features* (redução para o número de *features* mínimo possível) permite a identificação da Doença de *Parkinson*, por género. As etapas do sistema proposto estão presentes na Figura 2.4, onde basicamente através de *Greedy Stepwise Search Algorithm*, se realizou a etapa de seleção de *features*, que posteriormente foram utilizadas para as etapas de treino e teste do algoritmo. Ainda neste processo, utilizaram *Cross-validation (CV)* de forma a reduzir a possibilidade de ocorrência de fenómenos de *overfitting*.

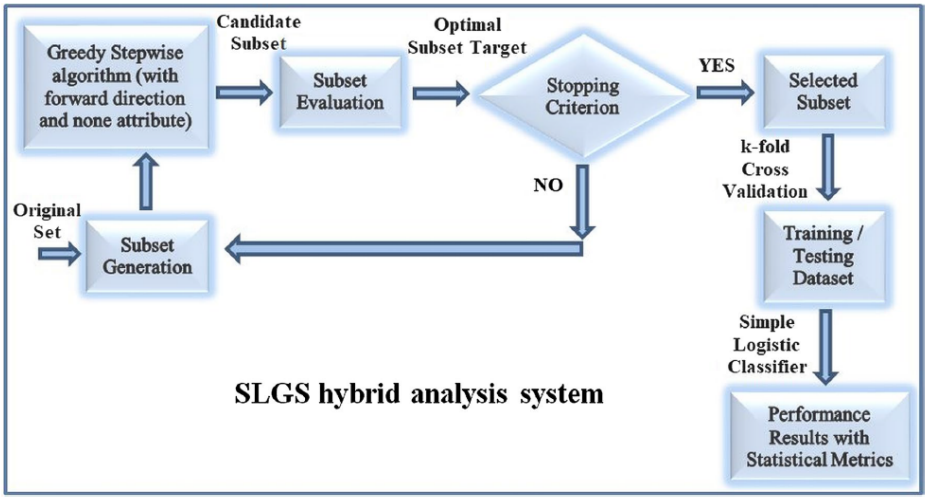


Figura 2.4: Sistema proposto para a classificação de *Parkinson*, por género [27].

No que concerne aos resultados apresentados, o sistema através da utilização de 11 *features* foi capaz de realizar a deteção no sexo Masculino com uma *accuracy* de 88,71%, enquanto que no sexo Feminino, com 9 *features* foi capaz de realizar a classificação com uma *accuracy* de 87,15%.

Tuncer et al. [28] utilizaram o conjunto de dados com 753 características da voz, retirado do repositório público de *UCI Machine Learning Database*. Como consideram que o *dataset* é pequeno, decidiram-se pela utilização de métodos de *Machine Learning* não convencionais, isto é *MAMa tree* (método de seleção de características baseado em árvores de decisão, especificamente projetado para identificar e selecionar as características mais relevantes num conjunto de dados) é utilizada no conjunto de dados proveniente, de forma a extrair as *features*, sendo que os dados presentes nos diferentes nodos são combinados, sendo posteriormente utilizado um *Singular value decomposition (SVD) features extraction* para a seleção de *features*. Na fase de seleção de *features*, 50 *features* são extraídas como as mais importantes, usando seleção de *features* através de *relief*. Posteriormente, 8 algoritmos de classificação (*Linear descriminant (LD)*), *SVM* linear, SVM com *kernel* de *Radial based function (RBF)*, e *kernel* cúbico, *LR*, *NN with city block distance and k = 1*, e *Bagged tree (BT)*) são utilizados em 2 casos:

- Caso 1: Pré-processamento, extração de *features* e classificação.
- Caso 2: Pré-processamento, extração de *features*, classificação e etapa de pós-processamento.

Ao nível de resultados, no caso 1, o resultado superior foi uma *accuracy* de 92,46%, enquanto que no caso 2 foi uma *accuracy* de 96,83%, ambos através da utilização do *NN*.

Nissar, Rizvi, Masood, e Mir [29] procederam à análise de um estudo no qual se utilizou um conjunto de dados retirado da *UCI Machine Learning Database* (*UCI Machine Learning Repository*, 2018).

Este conjunto de dados é constituído por dados de pacientes normais e por dados de pacientes com doença de *Parkison*. Apresenta 756 registos e 753 características, especialmente utilizadas para classificação da doença de *Parkinson* através da voz. Neste artigo, a metodologia incorpora as etapas de aquisição de dados, pré-processamento de dados através da normalização dos mesmos através de *Min-Max Scalling*, seleção de *features* mais relevantes através de dois métodos (*Recursive feature selection* e *minimum redundancy*), aplicação de Modelos de *Machine Learning* e avaliação da performance do mesmo. Para a aplicação de Modelos utilizaram-se 9 algoritmos distintos: *NB*, *LR*, *NN*, MLP, *RF*, SVM (linear), SVM (*RBF*) e *Extreme gradient boost (XGBoost)* de 3 formas distintas:

- Seleção de características com a técnica *Recursive feature selection* e *minimum redundancy feature selection*, exceto o grupo de características *TQWT*.
- Seleção de características com a técnica *Recursive feature selection* e *minimum redundancy feature selection*, exceto o grupo de características *Mel-frequency cepstral coefficients (MFCCs)*.
- Técnicas de seleção de características *Recursive feature selection* e *minimum redundancy feature selection* com todos os grupos de características.

Comparando todos os resultados das 3 formas distintas, verificou-se que o algoritmo que obteve melhores resultados foi o *XGBoost*, com uma precisão de 95.39%, quando utilizada a técnica de seleção de características *minimum redundancy feature selection*.

Ashour et al. [30] propuseram uma *framework* que fazia a seleção das *features* em dois estágios, permitindo a identificação de pacientes com perda de voz em doentes com *Parkinson*. Devido ao conjunto de dados ser altamente dimensional, isto é, as 753 características presentes aumentarem bastante a dimensão do espaço das *features* e aumentarem a possibilidade de existência de *features* irrelevantes, torna-se necessária a redução desta dimensionalidade, sendo que o processo de seleção de *features* é a melhor maneira de reduzir esta dimensão. Os autores utilizaram métodos de *Principal component analysis (PCA)* e *Eigenvector centrality feature selection (ECFS)* como seleção de *features*, permitindo tirar vantagens de ambos os métodos. Uma vez selecionadas as *features*, utilizou-se como método de classificação o algoritmo de *SVM* com o *kernel* cúbico. A nível de resultados, verificou-se que existiu uma melhoria na *accuracy* com e sem a realização da seleção de *features* proposta, verificando-se um aumento de 88% para 94%.

Omar Barukab et al. [24] recorreram a um conjunto de dados disponibilizado por C. O. Sakar et al. (2019), conjunto de dados que apresenta 756 registos e 753 características, onde 65 dos pacientes são saudáveis (41 do sexo Feminino e 23 do sexo Masculino) e 188 pacientes (81 do sexo Feminino e 107 do sexo Masculino) apresentam doença de *Parkinson*, ou seja, uma *dataset* desbalanceado. Neste artigo, os autores propõem a utilização

de diferentes métodos de *ensemble*, com ou sem a utilização de métodos de *oversampling* ou *undersampling*. Os resultados mostraram que *AdaBoost*, *RF* e *DT* apresentam excelentes métricas, tais como precisão, *recall*, *F1-score*, *area under the receiver operating characteristic curve* (AUROC). Utilizaram-se algoritmos de seleção de *features*, tais como o *Lasso* e *Information gain*, de forma a fazer a seleção das 10 melhores *features*. Por fim, o *Adaptative boosting (AdaBoost)* com método de *Information gain* na seleção de *features* é o método de conjunto de melhor desempenho com uma pontuação F1 de 0,903.

Mohammadi et al.[31] recorreram a um conjunto de dados disponibilizado por C. O. Sakar et al. (2019), onde afirma que este conjunto tem um tamanho de amostra limitado e recursos desequilibrados. Neste processo fizeram o uso de *autoencoders* para o processo de seleção de *features*. O resultado mostrou que o uso de *autoencoders* como extratores de *features* pode ser benéfico quando o total de amostras é menor que o número de recursos, especialmente quando a entrada está desequilibrada.

Uma vez obtido o conjunto de dados, primeiramente realizaram a normalização de todas as *features*, de forma a que estas apresentem todas a mesma escala. Uma vez normalizados os dados, desenvolveram 2 abordagens distintas para o mesmo problema: treino e avaliação com os dados simplesmente normalizados e treino e avaliação com os dados normalizados e com o processo de seleção de *features* através de *autoencoders*.

Para a primeira maneira, utilizaram-se os modelos de *SVM*, XGBoost e *MLP* com a utilização de *cross-validation* com 5 *folds* e *tuning* de hiperparâmetros através do *GridSearch*, de onde foram extraídas as métricas de *Accuracy* e *F1-Score*. Neste sentido, obtiveram-se os seguintes resultados na primeira abordagem: *SVM* com o *kernel poly* e o *degree* a 23, obteve-se uma precisão de 94.07% e um *F1-Score* de 96,08%, *XGBoost*, obteve uma precisão de 92.19% e um *F1-Score* de 94.92%, com os parâmetros: *colsample_bytree* = 0.35, *n_estimators* = 325, *max_depth* = 4, *learning rate* = 0.1, *alpha* = 1e-2, *subsample* = 0.75 e *MLP* obteve-se uma precisão de 90.61% e um *F1-Score* 93.72%, com os parâmetros: camada intermédia = 160 e nodos = 25.

Na segunda maneira, utilizaram-se os mesmos modelos, apenas diferindo que após a normalização dos dados ocorreu um processo de seleção de *features* através de *autoencoders*. Por fim, utilizou-se Métodos de *Ensemble* para o aumento do desempenho dos algoritmos.

A nível de resultados, mostraram que os modelos de classificação tradicionais superam as técnicas de aprendizagem profunda, mostrando desta forma que os Modelos de *Machine Learning* conseguem obter resultados bons em problemas de classificação, quando comparados com modelos de *Deep Learning*. Nos resultados demonstraram que *Ensemble Learning* apresenta um papel fundamental na melhoria da classificação, uma vez que conseguiram *accuracy* entre 95 e 97% através da utilização de algoritmo como *SVM*, XGBoost, *MLP* seguidos de *autoencoders*.

Qasim et al. [32] recorreram a um conjunto de dados disponibilizado por C. O. Sakar et al. (2019), onde afirma que este conjunto tem um tamanho de amostra limitada e recursos desequilibrado, representando o *Bias* existente nos *datasets* ligados a Medicina. Neste projeto utilizou-se o algoritmo de *Synthetic minority oversampling technique (SMOTE)* para contornar o desbalanceamento existente no *dataset*. Neste contorno de desbalanceamento teve-se como produto final um *dataset* com 564 registos nas duas classes de classificação. De seguida, procederam com a normalização do *dataset* balanceado.

Younis Thanoun and Yaseen [33] propuseram a utilização de 2 técnicas de *Ensemble* para

fazer a deteção de doença de *Parkinson* através de *Machine Learning*: Classificadores *Stacking* e classificadores *Voting*, após o balanceamento do conjunto de dados através de *SMOTE*, tal como se verifica na Figura 2.5.

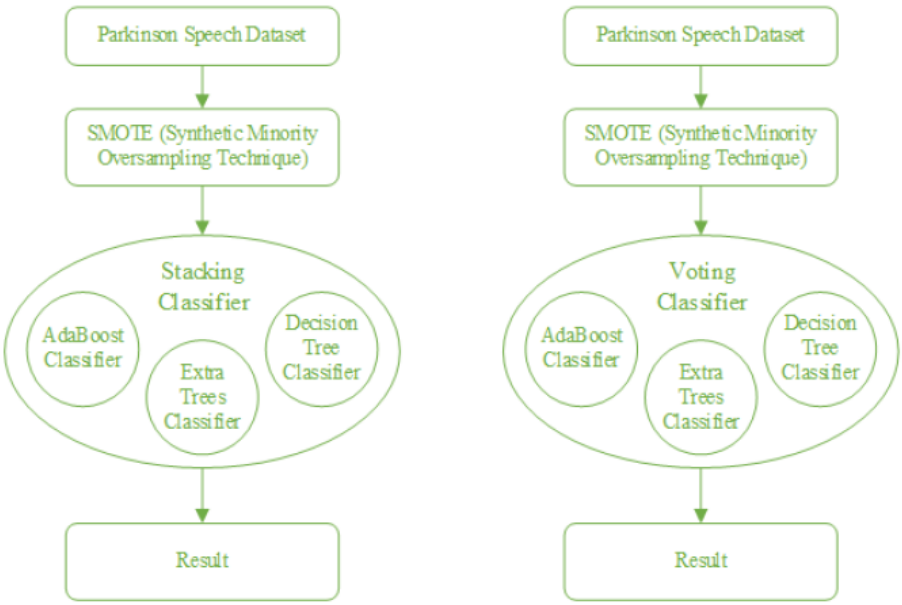


Figura 2.5: Metodologia apresentada no estudo de Younis Thanoun and Yaseen [33].

Ensemble Voting combina as previsões de vários modelos por meio de votação, escolhendo a classe com a maioria dos votos ou a média das probabilidades, enquanto que *Ensemble Stacking* treina um modelo adicional (meta-modelo) para aprender a partir das previsões de múltiplos modelos base, fazendo a previsão final.

Através dos resultados, verificou-se que os Classificadores *Stacking* apresentaram resultados superiores aos classificadores *Voting*, sendo a *accuracy* de 92,2% e 83.57%, respetivamente.

Prasad et al. [15] recorreram a conjunto de dados com 753 *features* da voz, e utilizaram uma *framework* de classificação em duas etapas, sendo a primeira etapa a utilização de múltiplas *Analysis of variance (ANOVA)* nas *features* vocais independentes separadamente (*MFCCs*, *Wavelet transform (WT)* e *TQWT*) para seleção das melhores *features*, que são ligadas com as *Baseline features*, e a segunda etapa a utilização de classificadores *XGBoost*, tal como se verifica na Figura 2.6.

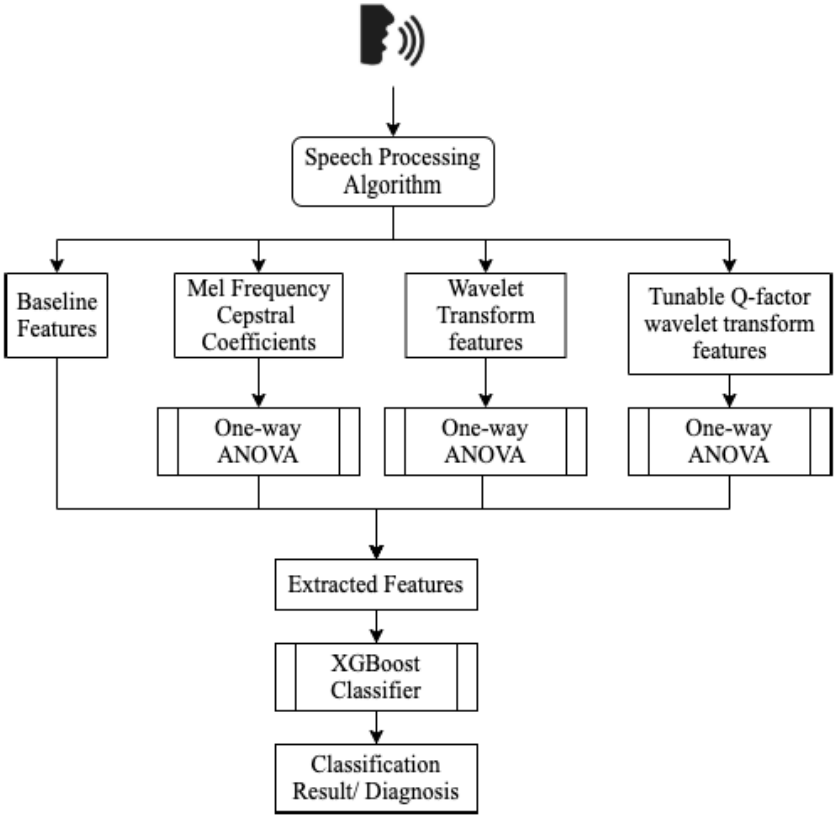


Figura 2.6: Metodologia apresentada no estudo de Prasad et al. [15].

No que concerne aos resultados do modelo, a *framework* proposta apresenta uma *accuracy* de 94,71% e um *F-1 score* de 96%.

Uma vez balanceado e normalizado o *dataset* procedeu-se com a seleção das *features*, de forma a evitar contradições existentes, bem como diminuir o tempo de processamento, através de algoritmos de *Recursive feature elimination (RFE)* e *PCA*. Esta etapa foi bastante importante na redução da quantidade de *features* existentes, uma vez que através do *RFE* fez-se uma redução para exatamente 329 *features*. Após o *RFE*, fez-se novamente uma redução do *dataset* através do *PCA*, o que resultou num valor final de 18 diferentes *features*.

A nível de algoritmos, utilizaram os seguintes Modelos: *Bagging*, *K-nearest neighbour*, *Multilayer perceptron* e *SVM*. Posteriormente aplicaram métodos de *fine-tuning* através de *GridSearch* obtendo-se os parâmetros presentes na Tabela 2.2.

Tabela 2.2: Algoritmos e parâmetros parâmetros utilizados em Quasim et al. [32]

Algoritmo	**Parâmetros do algoritmo**
Bagging	*Classifier* = *NN*, *Number of iterations* = 100, *max samples* = 0.9
K-nearest neighbour	*Number of neighbors* = 1, *Leaf size* = 40
Multilayer perceptron	*Number of iterations* = 500, *learning rate* = 0.01, *Solver for optimum weight* = *adam*
SVM	*Regularization parameter* = 1, *kernel* = *rbf*, *Gamma* = 2

Por último, procederam à avaliação dos algoritmos de 4 formas distintas:

- Conjunto de dados desbalanceado.
- Conjunto de dados equilibrado e com a utilização do algoritmo de seleção de *features* *RFE*.
- Conjunto de dados equilibrado e com a utilização dos algoritmos de seleção de *features* *RFE* e *PCA*.
- Conjunto de dados equilibrado, com a utilização dos algoritmos de seleção de *features* *RFE* e *PCA* e com a utilização do *GridSearch* para otimização.

Para a avaliação dos algoritmos foram utilizadas cinco Métricas: *accuracy*, *precision*, *sensitivity*, *specificity*, and *G-Mean*. Por conseguinte, para o primeiro caso o algoritmo que obteve um melhor desempenho foi o *NN* com uma precisão de 87.4% e com as 753 características do conjunto de dados. No segundo caso o algoritmo que obteve um melhor desempenho foi o MLP com uma precisão de 93.3% e com 329 características. No terceiro caso, o algoritmo que obteve um melhor desempenho foi também o *MLP* com uma precisão de 95.1% e com 18 características. Por fim, no quarto caso foi adicionado o método de *Fine-Tuning* (*GridSearch*) com o objetivo de encontrar os valores ótimos para cada parâmetro dos algoritmos e o algoritmo que teve um melhor desempenho foi o *SVM* com uma precisão de 98%.

Abdurrahman et al. [14] propuseram a utilização de *XGBoost* para a tarefa de classificação da Doença de *Parkinson*, uma vez que este algoritmo apresenta elevada escalabilidade, ou seja, apresentando uma velocidade de processamento superior e consumo de memória inferior aos algoritmos tradicionais de *Machine Learning*. Ainda neste estudo, seguiu-se a metodologia de trabalho presente na Figura 2.7.

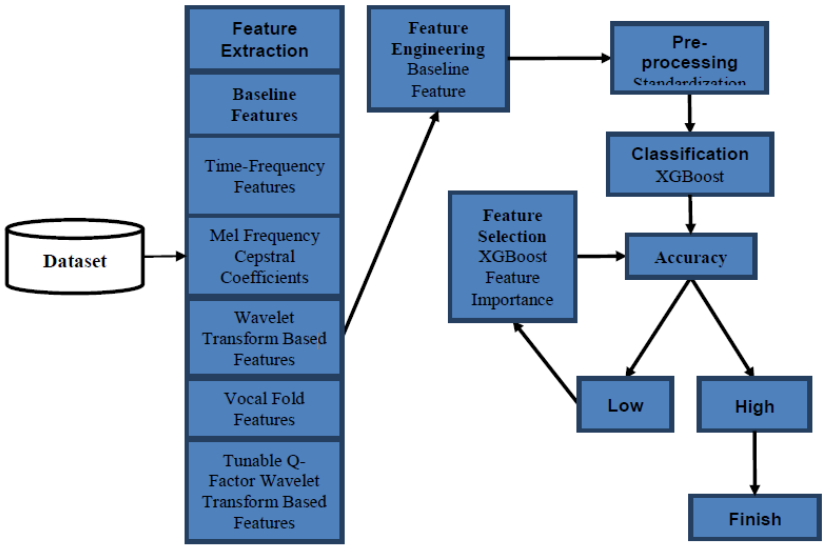


Figura 2.7: Explicação do processo seguido no estudo de Abdurrahman et al. [14].

Como se pode verificar na Figura 2.7, partindo de um conjunto de dados com 754 *features*, nomeadamente *baseline features, intensity parameters, format frequencies, bandwidth parameters, vocal fold, MFCC, Wavelet features,* e *TQWT features*, os autores apenas procederam com a utilização das *Baseline features* no estudo.

Na etapa de classificação, apenas se utilizou o algoritmo de *XGBoost*, onde previamente foi realizada a distinção entre conjunto de dados de treino e conjunto de dados de teste,

sendo que 33% dos dados foram selecionados para o conjunto de teste, sendo a restante quantidade selecionada para os dados de treino. Numa primeira fase, o algoritmo de *XGBoost* foi usado para classificação, envolvendo todas as *Baseline Features*, onde se obteve uma *accuracy* de 84.80%. Posteriormente, utilizaram a *Feature Importance* do algorimo *XGBoost* para se realizar uma redução no número de *Features*, sendo realizados mais dois testes, um primeiro onde se removeu as *features ddaShimmer* e *locShimmer*, obtendo-se uma *accuracy* de 85.60%. Num segundo teste, removeu-se as *features locDbShimmer, meanNoiseToHarmHarmonicity, ppq5Jitter,apq5Shimmer, ddpJitter, rapJiter, PPE*, obtendo-se uma *accuracy* de 84.40%.

Biswajit Karan1 [34] propôs a utilização de uma *framework* de *Machine Learning* baseada em duas etapas: seleção de *features* com mais informação através de algoritmo de *XGBoost*, seguida da utilização de algoritmos *ensemble* na etapa de classificação, onse combinaram diversos classificadores base (*SVM*, NN, RF, *XGBoost* e *MLP*), de forma a criar um classificador mais forte. Nesta etapa, de forma resumida, as probabilidades dos classificadores mais fracos são utilizadas como novas features para diminuir o erro de previsão do classificador principal. Na etapa de treino dos modelos utilizou-se o *Leave one subject out (LOSO)* como técnica de validação cruzada. De forma a classificar a performance dos modelos utilizou-se as métricas de *Accuracy*, sensibilidade, especificidade e *Area under the ROC curve value (AUC).*

A etapa de seleção de *features* foi realizada com o *XGBoost* variando o valor do *threshold* entre 0.001 e 0.02, onde se obteve o valor ótimo com o *threshold* de 0.009, onde se selecionam 21 *features*, sendo 1 da categoria *Baseline Features*, 4 da categoria de *MFCCs*, 3 da categoria de *DWT-based features* e 13 da categoria *TQWT features*. Estas 21 *features* foram utilizadas como *features* de entrada da *stack* de métodos *ensemble*. Na etapa de testes de classificadores *ensemble* obteve-se o valor máximo de 95.07% de *accuracy* quando se utilizou o *SVM*, *NN*, *RF* e *MLP* como métodos base e o método *XGBoost* utilizado como método de melhoria de performance.

Kemal Polat [35] propôs um método híbrido para a deteção da doença de *Parkinson* através da combinação do algoritmo de *SMOTE*, de forma a lidar com um conjunto de dados desbalanceado (192 dados são relativos a utentes sem a doença de *Parkinson*, enquanto que 564 dados pertencem a utentes com a doença de *Parkinson*) conjugado com a utilização de *RF* como algoritmos de classificação. Na etapa de treino e de teste, utilizou-se uma divisão de 50% de dados para treino e 50% de dados para teste, com a utilização de *10-fold Cross-Validation*. A nível de resultados, verificou-se que a utilização unitária do método de classificação de *RF*, isto é, sem uma etapa de *SMOTE* apresentou uma *accuracy* de 87,037%, enquanto que a utilização do método híbrido proposto resultou numa *accuracy* bem superior com o valor de 94.89%.

Vários estudos foram referidos, com o intuito de classificar a doença de *Parkinson* através características vocais dos indivíduos. Na Tabela 2.3, apresenta-se um resumo das principais características de cada um desses estudos.

Tabela 2.3: Quadro resumo dos estudos realizados

Estudos		
Principais Características	**Algoritmos**	**Avaliação do Estudo**
Şule Yücelbaş [27]: Conjunto de dados com 753 características da voz, onde se dividiu a classificação por género e procedeu com a seleção de *features* através de *Greedy Stepwise Search Algorithm*.	*Logistic Regression.*	No que concerne aos resultados apresentados, o sistema através da utilização de 11 *features* foi capaz de realizar a deteção no sexo Masculino com uma *accuracy* de 88,71%, enquanto que no sexo Feminino, com 9 *features* foi capaz de realizar a classificação com uma *accuracy* de 87,15%.
Tuncer et al. [28]: Conjunto de dados com 753 características da voz, onde se procedeu com a seleção de *features* através de uma combinação entre *Minimum average maximum tree (MAMa)* e *SVD*.	*K-Nearest Neighbors with 10-fold Cross-Validation.*	92,46% e 96,83% (com 50 *features*).

Principais Características	**Algoritmos**	**Avaliação do Estudo**
Nissar, Rizvi, Masood, e Mir [29]: Conjunto de dados com 753 características da voz, onde se aplicou pré-processamento através da normalização das features entre 0 e 1 e através de duas diferentes técnicas de seleção de *features* (*Minimum redundancy maximum relevance feature selection* e *Recursive Feature selection*. O treino e teste do modelo foi realizado de 3 formas separadas: - Seleção de *features* com a técnica de *Recursive Feature Selection* e *Minimum redundancy maximum relevance feature selection*, exceto as *features* de *TWTQ*. - Seleção de *features* com a técnica de *Recursive Feature Selection* e *Minimum redundancy maximum relevance feature selection*, exceto as *features* de *MCFF*. - Seleção de *features* com a técnica de *Recursive Feature Selection* e *Minimum redundancy maximum relevance feature selection* com todas as *features*.	*NB*, *LR*, *NN*, *MLP*, *RF*, *SVM* (linear), $SVM(RBF)$ e *XGBoost*.	- Técnica seleção de características *Recursive feature selection*: o modelo XGBoost tem uma precisão de 95.39%. - Técnica seleção de características *mRMR*: o modelo *LR* tem uma precisão de 86,84%. - Nas duas técnicas o modelo *SVM* com o *kernel RBF* tem uma precisão de 88,15%.
Younis Thanoun and Yaseen [33] propuseram a utilização de 2 técnicas de *Ensemble* para fazer a deteção de doença de *Parkinson* através de *Machine Learning*: Classificadores *Stacking* e classificadores *Voting*, após o balanceamento do conjunto de dados através de *SMOTE*.	Classificadores *Stacking* e classificadores *Voting*, com os algoritmos *AdaBoost, Extra Trees e Decision Tree (DT)*.	Classificadores *Stacking* apresentaram resultados superiores aos classificadores *Voting*, sendo a *accuracy* de 92,2% e 83.57%, respetivamente.

Principais Características	**Algoritmos**	**Avaliação do Estudo**
Prasad et al. [15] recorreram a conjunto de dados com 753 *features* da voz, e utilizaram uma *framework* de classificação em duas etapas, sendo a primeira etapa a utilização de múltiplas *ANOVA* nas *features* vocais independentes separadamente (MFFCCs, WTs e TQWTs) para seleção de *features*, que são ligadas com as *Baseline features*, e a segunda etapa a utilização de classificadores *XGBoost*, tal como se verifica na Figura 2.6.	*XGBoost*	No que concerne aos resultados do modelo, a *framework* proposta apresenta uma *accuracy* de 94.71% e um *F-1 score* de 0,96.
Ashour et al. [30]: *Framework* que faz a seleção das *features* em dois estágios, permitindo a identificação de pacientes com perda de voz em doentes com *Parkinson*. Os autores afirmam que as 753 características aumentam bastante a dimensão do espaço das *features* e que aumentam a possibilidade de existência de *features* irrelevantes, sendo que processo de seleção de *features* é a melhor maneira de reduzir esta dimensão. Os autores utilizaram métodos de *PCA* e *ECFS* como seleção de *features*, permitindo tirar vantagens de ambos os métodos.	*SVM* com o *kernel* cúbico.	nível de resultados, verificou-se que existiu uma melhoria na *accuracy* com e sem a realização da seleção de *features* proposta, verificando-se um aumento de 88% para 94%.

Principais Características	**Algoritmos**	**Avaliação do Estudo**
Mohammadi et al.[31] - Conjunto de dados contém 753 registos. - Pré-processamento dos dados: normalização dos dados no intervalo [0,1] e técnica de seleção de características *Autoencoder*. - Treino e teste dos modelos da seguinte forma: 1. Com todas as características do conjunto de dados. 2. Apenas com as características selecionadas pela técnica *Autoencoder*. - Implementação de *cross validation* com k=5. - Otimização dos parâmetros através da técnica *GridSearch*.	*SVM*, *XGBoost* e *MLP*.	- Com todas as características do conjunto de dados o modelo *SVM* tem uma precisão de 94.07%. - Técnica de seleção de características *Autoencoder* o modelo *SVM* tem uma precisão de 91.93%.
Qasim et al. [32]: Conjunto de dados com 753 registos. - Pré-processamento: técnica de *SMOTE* para equilibrar o conjunto de dados, normalização das características no intervalo [0,1] e aplicação das técnicas de seleção de características *Recursive feature selection* (selecionou 329) e a *principal component analysis* (18 componentes) - Divisão do conjunto de dados, num conjunto de dados de treino 80% e um conjunto de dados de teste 20%, com a validação através da técnica *cross validation*, com k=10. - Treino e teste dos modelos da seguinte forma: 1. Conjunto de dados desequilibrado; 2. Conjunto de dados equilibrado e técnica de seleção de características *Recursive feature selection*; 3. Conjunto de dados equilibrado e as duas técnicas de seleção de características *Recursive feature selection* e *Principal Component Analysis*; 4. Conjunto de dados equilibrado e as duas técnicas de seleção de características *Recursive feature selection* e *Principal Component Analysis* com os parâmetros otimizados através da técnica *GridSearch*.	MLP, *SVM*, *NN* e *Bagging*.	No primeiro caso o modelo *NN* tem uma precisão de 87.4%. - No segundo caso o modelo *MLP* tem uma precisão de 93.3%. - No terceiro caso o modelo *MLP* tem uma precisão 95.1% - No quarto caso o modelo SVM tem uma precisão de 98%.

Principais Características	**Algoritmos**	**Avaliação do Estudo**
Abdurrahman et al. [14]: Conjunto de dados com 753 características da voz, onde se aplicou pré-processamento da seleção de apenas as *Baseline Features* e posteriormente seleção através da *Feature Importance* do Algoritmo de *XGBoost*. O treino e teste do modelo foi realizado de 3 formas separadas: - Com todas as *Baseline Features*. - Remoção das *features ddaShimmer* e *locShimmer*. - Remoção das *features locDbShimmer, meanNoiseToHarmHarmonicity, ppq5Jitter, apq5Shimmer, ddpJitter, rapJiter* e *PPE*.	*XGBoost*.	No primeiro caso obteve-se uma *accuracy* de 84.80%. - No segundo caso obteve-se uma *accuracy* de 85.60%. - No terceiro caso obteve-se uma *accuracy* de 84.40%.
Biswajit Karan1 [34]: Conjunto de dados com 753 características da voz, onde se aplicou uma *framework* de *ML* baseada em duas etapas: seleção de *features* com mais informação através de algoritmo de *XGBoost*, seguida da utilização de algoritmos de *stacking ensemble* na etapa de classificação, através da combinação de vários classificadores fracos (SVM, *NN*, *RF*, *XGBoost* e MLP). Na seleção de *features*, o *threshold* máximo foi conseguido com 0.09 de onde se extraíram 21 *features*.	SVM, *NN*, *RF*, *XGBoost* e MLP.	Na etapa de testes de classificadores *ensemble* obteve-se o valor máximo de 95.07% de *accuracy* quando se utilizou o SVM, *NN*, *RF* e MLP como métodos base e o método *XGBoost* utilizado como método de melhoria de performance.
Kemal Polat [35] : Propôs um método híbrido para a deteção da doença de *Parkinson* através da combinação do algoritmo de *SMOTE*, de forma a lidar com o problema de conjunto de dados desbalanceado (192 dados são relativos a utentes sem a doença de *Parkinson*, enquanto que 564 dados pertencem a utentes com a doença de *Parkinson*) conjugado com a utilização de *RF* como algoritmos de classificação. Na etapa de treino e de teste, utilizou-se uma divisão de 50% de dados para treino e 50% de dados para teste, com a utilização de *10-fold Cross-Validation*.	*RF*.	A nível de resultados, verificou-se que a utilização unitária do método de classificação de *RF*, isto é, sem uma etapa de *SMOTE* apresentou uma *accuracy* de 87,037%, enquanto que a utilização do método híbrido proposto resultou numa *accuracy* bem superior com o valor de 94.89%.

Os resultados apresentados na Tabela 2.3 fornecem uma visão abrangente de diferentes estudos focados na deteção da *DP* através de um conjunto de dados da voz e com a utilização de uma variedade de algoritmos de *ML*, com diversos métodos de pré-processamento dos dados.

O estudo de Şule Yücelbaş (2020) [27] utilizou um conjunto de dados com 753 características da voz e aplicou o algoritmo *Greedy Stepwise Search* para a seleção de *features*, utilizando regressão logística para a classificação. A precisão alcançada foi de 88,71% para homens e 87,15% para mulheres, usando 11 e 9 *features*, respetivamente.

Tuncer et al. [28] utilizaram uma combinação de técnicas *MAMa* e *SVD* para a seleção de *features* e utilizaram o algoritmo *K-Nearest Neighbors* com *CV* em 10 partes, alcançando uma precisão de 92,46% e 96,83% com 50 *features*.

Nissar et al. [29] aplicaram normalização e usaram técnicas de seleção de *features* como mRMR e *RFE*, testando múltiplos algoritmos. O XGBoost com *RFE* alcançou uma precisão de 95,39%, enquanto o *LR* com mRMR teve uma precisão de 86,84%, e a combinação de *RFE* e SVM com *kernel RBF* obteve 88,15%.

Younis Thanoun e Yaseen [33] utilizaram técnicas de *ensemble* como *Stacking* e *Voting* após o balanceamento dos dados com SMOTE, onde o método *Stacking* obteve uma precisão de 92,2%, superior ao método *Voting*, que alcançou 83,57%.

Prasad et al. [15] implementaram uma *framework* de classificação em duas etapas usando ANOVA e XGBoost, alcançando uma precisão de 94,71% e um *F1-score* de 0,96.

Ashour et al. [30] utilizaram PCA e ECFS para a seleção de características, e o algoritmo SVM com *kernel cúbico*, observando uma melhoria na precisão de 88% para 94%.

Mohammadi et al. [31] normalizaram os dados e usaram *autoencoder* para a seleção de características, testando *SVM*, *XGBoost* e *MLP*. O *SVM* com todas as características teve uma precisão de 94,07%, enquanto com a seleção de características pelo *autoencoder*, a precisão foi de 91,93%.

Qasim et al. [32] utilizaram técnicas de balanceamento e seleção de *features* como *SMOTE*, *RFE* e *PCA*. O modelo *MLP* alcançou uma precisão de 95,1% com *RFE* e *PCA*, enquanto o *SVM* com otimização de parâmetros via *GridSearch* atingiu 98% de precisão.

Abdurrahman et al. [14] usaram apenas as *Baseline Features* e a seleção de *features* pelo *Feature Importance* do *XGBoost*, alcançando uma precisão de 85,60%.

Biswajit Karan [34] aplicou uma *framework* baseada em *XGBoost* para a seleção de características e *stacking ensemble* na classificação, obtendo uma precisão máxima de 95,07%.

Kemal Polat [35] propôs um método híbrido combinando *SMOTE* e *Random Forest*, resultando numa precisão de 94,89%.

Os estudos demonstram que a precisão na deteção da DP pode ser significativamente melhorada por meio de técnicas eficazes de seleção de *features* e modelos *ML* robustos e eficazes. Técnicas como *RFE*, *mRMR*, *PCA*, e métodos de *ensemble* como *Stacking*, combinadas com algoritmos como *SVM*, *XGBoost* e *RF*, mostraram-se particularmente eficazes. A normalização e o pré-processamento dos dados são etapas críticas que contribuem para o desempenho dos modelos. A seleção adequada de *features* e a otimização

de parâmetros são essenciais para alcançar elevadas métricas, conforme evidenciado pelos resultados variados, mas geralmente altos, dos estudos analisados.

Capítulo 3

Machine Learning

Neste capítulo, explora-se os conceitos teóricos fundamentais de *Machine Learning*, fornecendo uma base sólida para entender os algoritmos e técnicas utilizados nesta área em rápida evolução.

3.1 Tipos de aprendizagem

Machine Learning é uma área da *IA* que tem como intuito o uso de dados e algoritmos de computação que pretende a imitação do pensamento humano, aumentando a sua *performance* desta forma [36]. Ainda assim, o objetivo passa pela criação e utilização de modelos, de forma a retirar conhecimento de um conjunto de dados[37].

De forma a se proceder à obtenção de conhecimento, estes modelos utilizam conceitos da Matemática, tais como Álgebra Linear, de Estatística e de Probabilidades [37]. Através destes modelos, obtém-se um resultado, que incorpora a previsão ou classificação de um determinado domínio.

Existem 4 diferentes tipos de Algoritmos de *Machine Learning*: *Supervised learning (SL)*, *Unsupervised learning (UL)*, *Semi-supervised Learning* e *Reinforcement learning* (RL), tal como pode ser visualizado na Figura 3.1 [38].

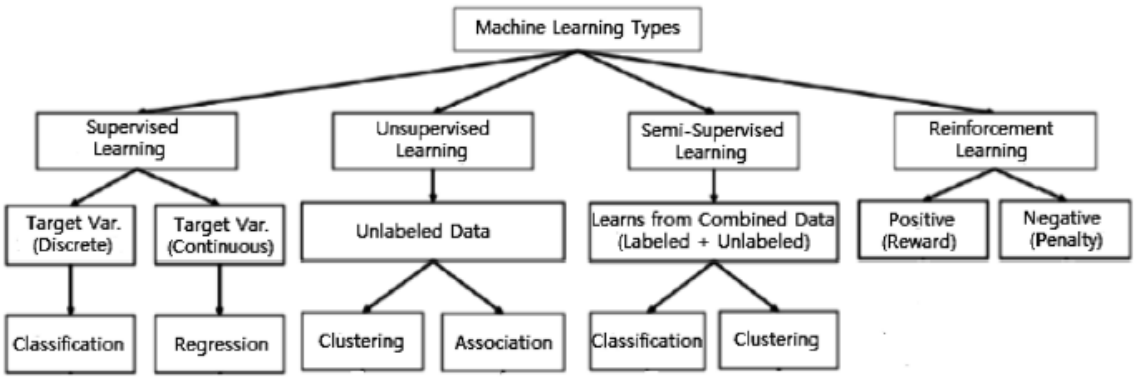


Figura 3.1: Tipo de de Algoritmos de *Machine Learning* [38].

Por outro lado, é importante de salientar que muitas vezes quando se fala em *Machine Learning*, referem-se ao procedimento de utilização de algoritmo de modelação para o problema em concreto, esquecendo-se de etapas essenciais de normalização e tratamento dos dados. Estas etapas são muito importantes, uma vez que a utilização de variáveis uniformizadas e escaladas é essencial para o melhor processo de treino dos algoritmos. Assim como a utilização de variáveis escaladas, é bastante importante a seleção das variáveis mais relevantes, isto é, aquelas que efetivamente têm importância na construção dos modelos, uma vez que permite diminuir a dimensionalidade e complexidade dos classificadores.

3.1.1 Aprendizagem supervisionada

A Aprendizagem Supervisionada, ou *Supervised Learning*, é um dos tipos de algoritmo de *Machine Learning*, sendo este um dos mais comuns e com mais casos de sucesso. Neste tipo de aprendizagem, o algoritmo é treinado de forma a que encontre uma relação entre as entradas e saída, ou seja entre as variáveis independentes e a variável dependente. Contudo não se pode aplicar este tipo de Aprendizagem em qualquer conjunto de dados, uma vez que é necessário que o conjunto de dados de saída esteja rotulado, seja este um rótulo categórico ou numérico [39]. O modelo é criado a partir dos pares de variáveis independentes e dependentes, sendo posteriormente testado com um conjunto de dados nunca antes visto [40].

Este tipo de Aprendizagem é amplamente utilizada em diversos tipos de problemas, tais como os problemas de Classificação, problemas de regressão, bem como de deteção de anomalias [39], sendo que neste tipo de classificação o objetivo passa por fazer a previsão de um rótulo de uma lista pré-definida de possibilidades. Os problemas de classificação dividem-se ainda em problemas de classificação binária (divisão entre duas classes) e problemas de classificação multi-classe, onde a distinção ocorre entre mais de duas classes. Nos problemas de regressão, o objetivo passa por fazer a previsão de um número contínuo, tais como a previsão de idade de uma pessoa ou do seu salário anual [40]. Por outro lado, nos problemas de deteção de anomalias, pretende-se, por exemplo, a deteção de transações fraudulentas em tempo real, em que através de redes Neuronais *Autoencoders* ou algoritmos de *clustering*, consegue-se analisar padrões de transações passadas, com o intuito de verificar comportamentos estranhos que possam indicar a presença de uma fraude.

Generalização, *Overfitting* e *Underfitting*

Como mencionado previamente, o objetivo dos problemas de Aprendizagem supervisionada é a construção de modelos de previsão a partir de um conjunto de dados de treino. Posteriormente, esses modelos são avaliados utilizando-se um conjunto de dados de teste, que possui as mesmas características do conjunto de treino, mas que não foi utilizado durante o processo de treino, de forma a garantir que a avaliação reflita a capacidade do modelo de generalização para dados novos e não vistos previamente [40]. Neste processo, quando o conjunto de dados de teste apresenta previsões precisas, pode-se afirmar que existiu uma generalização para o conjunto de dados de teste a partir do conjunto de dados de treino, sendo que o objetivo passa por ter um modelo que faça esta generalização com a maior precisão possível.

Esta medida da generalização apenas pode ser medida no conjunto de dados de teste, pois é neste conjunto que se visualiza o quão bem o modelo se adapta a novos dados, e para

isto acontecer a melhor forma é através da elaboração de modelo simples (não demasiado simplista), dado que a criação de modelos complexos ou demasiado simplistas pode fazer com que os modelos caiam em fenómenos de *overfitting* ou *underfitting*, respetivamente.

O fenómeno de *Overfitting* acontece quando o modelo apresenta resultados muito positivos no conjunto de dados de treino, contudo no conjunto de dados de teste estes resultados são inferiores. Este problema acontece quando o modelo se adapta demasiado ao conjunto de dados de treino, sendo este incapaz de fazer a previsão da mesma forma quando testado num conjunto de dados novo, ou seja, o modelo é muito específico para os dados de treino e não consegue realizar a generalização para os novos dados que possam surgir.

Por outro lado, também existe o problema de *Underfitting*, que acontece quando o modelo é demasiado simples, não sendo capaz de fazer a captura de todos os aspetos e da variabilidade do conjunto de dados, onde o modelo se adapta mal, inclusive no conjunto de dados de treino. Desta forma, torna-se necessário encontrar o ponto perfeito intermediário (*sweet spot*) entre estes dois possíveis fenómenos, que é onde se encontra a melhor performance ao nível de generalização, tal como se pode visualizar na Figura 3.2.

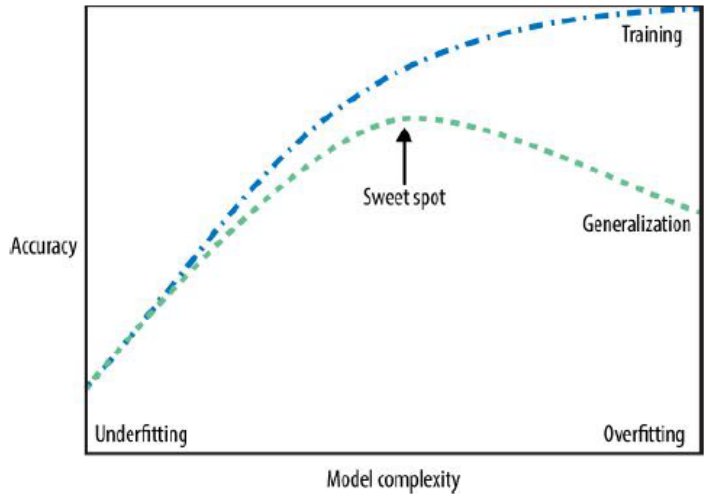


Figura 3.2: *Trade-off* entre a Generalização do modelo, *Overfitting* e *Underfitting* [40].

3.1.2 Aprendizagem não supervisionada

Ao contrário da Aprendizagem supervisionada, a Aprendizagem não supervisionada (*Unsupervised Learning*) não utiliza rótulos, ou seja, não envolve uma variável dependente específica. Uma vez que estes dados não apresentam rótulo, ao contrário da previsão ou classificação, a aprendizagem não supervisionada visa encontrar estruturas ou padrões subjacentes aos dados. O principal objetivo é descobrir informações e *insights* intrínsecos nos dados, sem orientação prévia sobre o que procurar [38]. Nesta aprendizagem, ao modelo apenas são apresentados os dados de *input*, ao qual se espera que o modelo faça a extração de conhecimento, isto é, uma espécie de criação de rótulo ou categoria, de acordo com os dados provenientes.

Este tipo de aprendizagem é amplamente utilizada em diversos tipos de problemas, sendo o *Clustering* e o *PCA* dois exemplos de algoritmos de Aprendizagem Não Supervisionada

[38], ou seja, em problemas de agrupamento de dados e problemas de transformação de dados. Um exemplo dos problemas de transformação de dados prende-se com redução da dimensão, em que existe um *dataset* altamente dimensional, com bastantes *features* como *input*. Nestes problemas, o *output* é um *dataset* muito mais conciso, representado por muito menos *features*, onde apesar de existir perda de informação, tenta-se manter as *features* mais explicativas, com uma melhoria no tempo de execução, sem bastante impacto na performance. Ao nível do *Clustering*, este pode ser utilizado para realização de segmentação de clientes num supermercado, tendo por base os seus hábitos de compras, ou por exemplo, para segmentação de análise de sentimentos, tendo por base os *tweets* no *Twitter*. Desta forma, verifica-se que o *Clustering* é usado para identificar grupos de dados semelhantes, facilitando a análise e a tomada de decisões.

Na Figura 3.3 elabora-se uma representação da dualidade entre a Aprendizagem supervisionada (*Supervised learning*) e a Aprendizagem Não Supervisionada (*Unsupervised learning*), onde é possível a verificação da diferença ao nível da existência e não existência de rótulos nos dados, uma vez que em *Unsupervised learning* nos dados de input estes estão todos a cinza, representativo da ausência de rótulo, enquanto que no *Supervised learning* estes apresentam um rótulo (representado a verde ou vermelho), bem como uma diferenciação ao nível dos *outputs* de ambos os algoritmos, em que num se faz uma classificação em duas *labels* (verde ou vermelho) e noutro se faz um agrupamento pelas características dos dados.

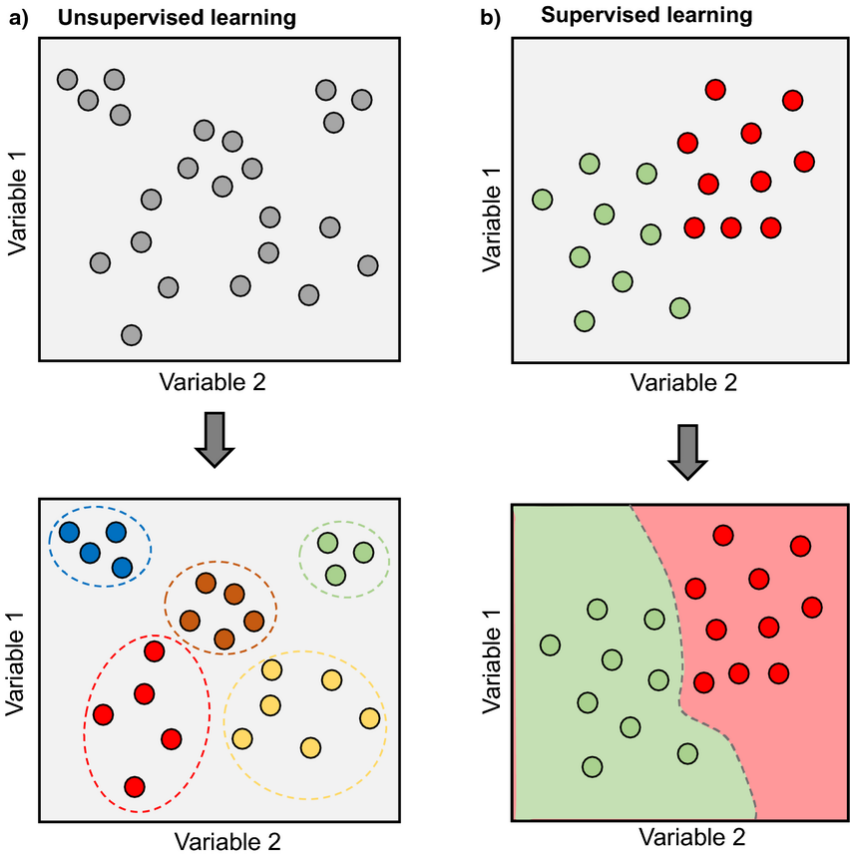


Figura 3.3: Diferenciação entre Aprendizagem Supervisionada e Aprendizagem Não Supervisionada [41].

3.1.3 Aprendizagem semi-supervisionada

A Aprendizagem semi-supervisionada, ou *Semi-Supervised Learning*, bem como o seu próprio nome indica, trata-se um paradigma de aprendizagem que combina elementos de Aprendizagem Supervisionada com elementos de Aprendizagem Não Supervisionada. Este tipo de Aprendizagem caracteriza-se pela utilização de um conjunto de dados misto, ou seja, uma mistura de exemplo com rótulos e dados sem rótulos, dados característicos de Aprendizagem Supervisionada e Não-Supervisionada, respetivamente [38].

Tendo em conta o contexto de aplicação do presente trabalho, a aprendizagem Semi-supervisionada, pode ser aplicada no hospital, em casos em que existam alguns registos que se encontram rotulados, contudo exista uma grande quantidade de registos não rotulada. Através de uma aprendizagem Semi-supervisionada é possível treinar os modelos com os registos rotulados e utilizar os registos que não têm rótulo para a deteção de doenças, o que permite um auxílio aos médicos, permitindo a realização de diagnósticos mais precisos.

3.1.4 Aprendizagem por reforço

A Aprendizagem por reforço (*Reinforcement Learning*) é um tipo de aprendizagem distinta dos modelos de aprendizagem supervisionada ou não supervisionada, onde um agente aprende a tomar decisões e ações através da interação com o ambiente que o rodeia. O agente avalia e ajusta automaticamente seu comportamento com base no contexto e nas mudanças do ambiente, o que constitui a principal diferença em relação aos tipos de aprendizagem mencionados anteriormente. Na aprendizagem por reforço, a ênfase está na interação contínua do agente com o ambiente, permitindo que o agente aprenda e melhore suas ações ao longo do tempo [38].

Neste tipo de aprendizagem, o modelo aprende de forma autónoma, baseando-se em recompensas ou penalidades recebidas como resultado de suas ações. O objetivo final é a maximização acumulativa da recompensa e a minimização do risco ao longo do tempo. Este tipo de algoritmo é fundamental em processos de automação e em sistemas de robótica autónomos, onde a capacidade de adaptação e otimização contínua das ações é crucial para o desempenho eficaz [38].

3.1.5 Análise dos tipos de aprendizagem de *Machine Learning*

Aprendizagem supervisionada:

Na aprendizagem supervisionada os algoritmos são treinados usando um conjunto de dados que possui rótulos, ou seja, entradas acompanhadas de suas saídas desejadas. Essa abordagem é particularmente útil quando se pretende realizar previsões ou classificações com base em exemplos históricos. É eficaz em situações em que se deseja fazer previsões através da utilização de dados previamente rotulados.

Vantagens:

- Ideal para previsão e classificação com dados rotulados.
- Eficaz em cenários de previsões baseadas em dados históricos.

Desvantagens:

- Dependência de dados rotulados.

- Limitado à qualidade e representatividade do conjunto de treino.

Aprendizagem não supervisionada:

Na aprendizagem não supervisionada os algoritmos são treinados com dados que não possuem *labels* predefinidas, tendo por objetivo a descoberta de padrões intrínsecos nos dados, valiosos para a segmentação de dados e redução de dimensionalidade. Esta aprendizagem é frequentemente utilizada na análise exploratória de dados e geração de *insights* a partir de grandes conjuntos não estruturados.

Vantagens:

- Descobre padrões em dados não rotulados.
- Útil para análise exploratória e geração de *insights*.

Desvantagens:

- Requer análise mais aprofundada do conjunto de dados.
- Menos eficaz em tarefas que exigem conhecimento prévio dos dados.

Aprendizagem por reforço:

Na aprendizagem por reforço, os algoritmos aprendem a tomar decisões interagindo com um ambiente, recebendo recompensas ou penalizações com base em suas ações. Essa abordagem é adequada para otimizar processos sequenciais, como jogos, controle de robôs e tomada de decisões em tempo real, onde as ações impactam o ambiente e o agente precisa aprender com a experiência.

Vantagens:

- Otimização de processos sequenciais.
- Adequada para ambientes interativos e tomada de decisões em tempo real.

Desvantagens:

- Pode exigir um grande número de interações para aprender efetivamente.
- A complexidade do ambiente pode aumentar a dificuldade de aprendizagem.

3.2 Etapas de processo de *Machine Learning*

O processo de *Machine Learning* envolve diversas etapas, que são geralmente organizadas em uma sequência lógica. As etapas a seguir descrevem o ciclo típico de desenvolvimento de um modelo de *Machine Learning* utilizado no presente trabalho.

3.2.1 Recolha, compreensão e identificação dos dados

A primeira etapa é a recolha de dados relevantes para a área de negócio envolvida no problema, sendo que os dados podem ser obtidos de diversas fontes, tais como base de dados, sensores, arquivos ou pedidos feitos via *web*.

Uma vez adquiridos os dados, é necessário realizar a sua compreensão, bem como a sua identificação, através da descrição dos mesmos, exploração, bem como a verificação da sua qualidade.

3.2.2 Preparação dos Dados

Os conjuntos de dados nunca são perfeitos, e a maioria destes apresenta valores omissos, apresentam *outliers*, bem como valores que não apresentem qualquer sentido para o contexto do negócio. Desta forma, os *datasets* obtidos necessitam de serem lidos e formatados frequentemente, de forma a puderem ser utilizados para a obtenção de conhecimento.

Nesta etapa de Preparação ou pré-processamento dos dados utilizam-se como processos: remoção de valores ausentes ou substituição de valores ausentes, normalização ou estandardização dos dados, tratamento de *outliers*, verificação de correlação entre atributos, de forma a promover a redução de atributos, e também muitas vezes a geração de novos atributos a partir dos dados já existentes.

3.2.3 Divisão dos dados

Normalmente, os dados são divididos antes da aplicação dos algoritmos. Assim, os dados são geralmente separados em 2 ou 3 conjuntos: treino e teste, ou treino, validação e teste, respetivamente. Nesta prática, o conjunto de dados de treino é utilizado para o treino do modelo, o conjunto de dados de validação é utilizado para o ajuste de hiperparâmetros e avaliação do modelo, sendo que o conjunto de dados de teste é utilizado para a avaliação final do modelo.

3.2.4 Seleção do algoritmo

Nesta fase, tendo em conta o contexto do problema em estudo, bem como a organização dos conjuntos de dados em questão, procede-se com a seleção dos algoritmos adequados. Este processo não é um processo fácil, pois os algoritmos tanto apresentam bons resultados para um problema, como apresentam resultados muito mais satisfatórias na previsão de outro problema.

3.2.5 Treino do algoritmo selecionado

Nesta etapa, o algoritmo é treinado utilizando o conjunto de dados de treino, permitindo que aprenda a relação entre as entradas e as saídas, ou seja, as classes correspondentes. O modelo ajusta seus parâmetros de modo a minimizar a diferença entre as suas previsões e os valores reais das classes.

3.2.6 Avaliação do algoritmo selecionado

Uma vez terminada a fase de treino do algoritmo, torna-se necessária a avaliação do algoritmo selecionado. Esta fase utiliza o conjunto de dados de teste e avalia assim a precisão e *accuracy* do modelo, tentando que se garanta a maior taxa de sucesso, evitando fenómenos de *underfitting* ou *overfitting*.

3.2.7 Melhoria e ajuste de parâmetros

Após a fase de avaliação do algoritmo, é necessário que se verifique se o algoritmo implementado pode ser melhorado, isto é, se este pode obter melhores métricas de avaliação. Tendo em conta este pressuposto, alteram-se os hiperparâmetros do modelo de forma a que sejam ajustados para a otimização máxima do desempenho do modelo. Uma vez alterado o conjunto dos hiperparâmetros, procede-se novamente ao treino e avaliação do modelo.

3.3 Técnicas de preparação dos dados

Normalização

A normalização, na área de *Machine Learning* é uma das técnicas utilizadas na etapa de Pré-processamento dos dados, que tem por finalidade o ajuste da escala das *features* numéricas, uma vez que diversas técnicas de *Machine Learning* são sensíveis à escala dos dados, tendo assim impacto na performance dos modelos. Neste sentido, um atributo que apresente unidades mais pequenas terá uma menor importância, quando comparado com um atributo com unidades maiores, contudo isto será ultrapassado recorrendo a duas técnicas existentes de normalização de dados: *Min-Max Scaling* ou *Z-Score Normalization*.

A *Min-Max Scaling* é uma técnica estatística que coloca os valores dos atributos num intervalo de valores específico, geralmente entre 0 e 1. Esta normalização assenta na equação 3.1:

$$x_{\text{norm}} = \frac{x - x_{\text{min}}}{x_{\text{max}} - x_{\text{min}}} \tag{3.1}$$

Por outro lado, *Z-Score Normalization* é uma técnica de pré-processamento de dados que ajusta os valores dos atributos de forma a que estes tenham uma média de 0 e um desvio padrão de 1, tornando assim os atributos comparáveis e tornando a deteção de *outliers* mais fácil. Esta normalização assenta na equação 3.2:

$$x_{\text{norm}} = \frac{x - \mu}{\sigma} \tag{3.2}$$

Correlação de *Pearson*

De forma a avaliar a independência entre as *features* de *input*, verifica-se a correlação entre as *features* através da utilização do coeficiente de Correlação de *Pearson*, uma vez que permite reduzir a redundância.

A correlação de *Pearson* trata-se de uma medida estatística que estuda a correlação linear existente entre duas variáveis, sendo que os seus valores variam entre -1 e 1, que representam uma correlação negativa perfeita (quando uma variável aumenta a outra variável diminui) e uma correlação positiva perfeita (quando uma variável aumenta a outra variável também aumenta), respetivamente. No que diz respeito ao valor, o valor de 0 representa a ausência de relação linear entre as variáveis, onde não existe nenhum tendência entre as duas variáveis [42]. A existência de correlação positiva (*Positive correlation*), correlação

negativa (*Negative correlation*) e ausência de correlação (*No correlation*) encontra-se apresentada na Figura 3.4 por esta mesma ordem.

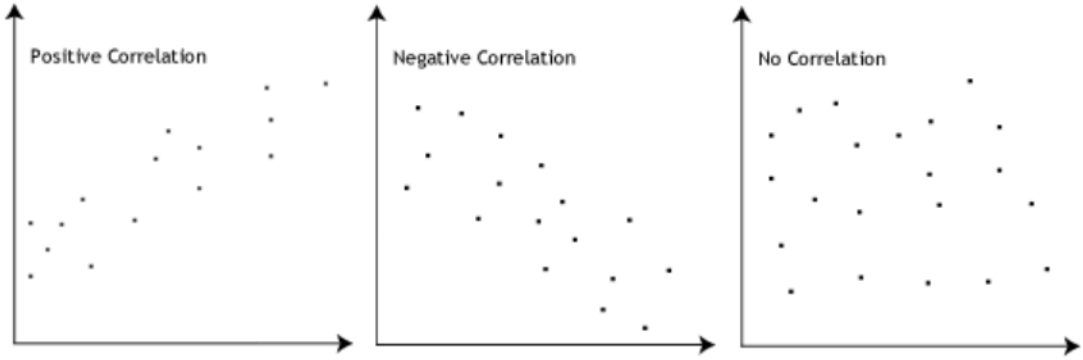


Figura 3.4: Correlação de *Pearson* [43].

A correlação de *Pearson* é importante na seleção de *features*, uma vez que permite fazer a seleção de *features* que devem ser incluídas num modelo, uma vez que *features* altamente correlacionadas podem ser redundantes, e a correlação pode ajudar a identificar essas redundâncias.

Variância inflacionária de fator

É importante a análise de correlação entre atributos, de forma a promover a redução do número de atributos como *Input* dos Algoritmos de *Machine Learning*, bem como a promoção de redução de fenómenos de *Overfitting*.

Para análise de correlação entre os diferentes atributos, pode optar-se pela utilização do fator *Variância inflacionária de fator (VIF)*.

Este Algoritmo, é um indicador de multicolinearidade usado para a avaliação de correlação entre os diferentes atributos, onde multicolinearidade ocorre quando uma ou mais variáveis independentes têm uma forte correlação entre si.

O valor de fator *VIF* divide-se em 3 possíveis acontecimentos:

- Se *VIF* é igual a 1, não existe correlação;
- Se *VIF* é maior que 1 e menor que 5, existe correlação intermédia;
- Se *VIF* é maior que 10, existe alta correlação;

Quando o *VIF* de uma variável é maior que um determinado limiar (geralmente 10), isso indica que a variável tem uma correlação alta com outras variáveis independentes e pode estar causando problemas de multicolinearidade, pelo que pode ser removida, de forma a não gerar redundância.

Principal Component Analysis

Atualmente, muitos dos conjuntos de dados apresentam uma quantidade enorme de informações, com um número considerável de variáveis (*features*), o que dificulta a análise. Muitas dessas variáveis são redundantes ou contêm informações duplicadas. Para facilitar

a análise e melhorar a eficiência dos modelos de *ML*, é necessário reduzir a dimensionalidade dos dados, aplicando técnicas de seleção de *features*, como a *PCA*.

O *PCA* é uma técnica de aprendizagem não supervisionada amplamente utilizada para redução de dimensionalidade. Ele transforma o conjunto de dados original num novo conjunto de variáveis, chamadas componentes principais, que são combinações lineares das variáveis originais. Essas componentes principais são ordenadas de forma que a primeira componente explique a maior parte da variância dos dados, a segunda componente explique a maior parte da variância residual, e assim por diante. O *PCA* permite a redução da dimensionalidade do conjunto de dados, mantendo o máximo possível de variância presente nos dados originais, minimizando a perda de informação relevante e aumentando a sua interpretação [44].

Por meio do *PCA*, é possível projetar os dados em um espaço de menor dimensão, simplificando a análise e, ao mesmo tempo, preservando as características mais importantes do conjunto de dados. Essa técnica é especialmente útil para visualizar dados de alta dimensão e para melhorar a eficiência de algoritmos de *ML*, que podem se beneficiar da eliminação de *features* redundantes.

Balanceamento de *datasets*

Muitos dos conjuntos de dados existentes encontram-se desbalanceados, representados por uma classe maioritária e uma ou mais classes minoritárias. No que diz respeito ao desbalanceamento em dados relativos à saúde, a classe maioritária usualmente é a ausência de uma doença.

Posto isto, é necessária a existência de um processo que permita o balanceamento dos conjuntos de dados, acabando assim com a existência de dados desequilibrados e que permita uma distribuição homogénea por classe. Este equilíbrio é importante, uma vez que os modelos de *Machine Learning* apresentam dificuldades em aprender a classe com menos representações. Este tipo de problema tem bastante impacto em problemas de classificação. Usualmente, existem duas possíveis formas de combater o problema de desbalanceamento dos conjuntos de dados:

- ***Oversampling (Supersampling)***: Criação de cópias adicionais de dados da classe minoritária até que o equilíbrio entre as classes seja alcançado.
- ***Undersampling (Subsampling)***: Redução do número de dados da classe maioritária até que o equilíbrio entre as classes seja alcançado. A redução dos dados pode ser feita de forma aleatória ou através da utilização de determinados critérios.

Contudo, esta duas técnicas também apresentam desvantagens, sendo que o *Oversampling* aumenta o risco de *overfitting*, ao mesmo tempo que aumenta o tempo de treino dos algoritmos. Por outro lado, o *undersampling* pode levar à diminuição da qualidade, uma vez que se podem retirar informações pertinentes da classe maioritária, apesar de aumentar a rapidez do algoritmo. Posto isto, atualmente existem algoritmos mais avançados que permitem este equilíbrio, tal como o *SMOTE*, que faz a criação sintética de dados da classe minoritária, contudo não replicados dados existentes, o que reduz a probabilidade de *overfitting*.

A técnica de *SMOTE*, como se trata de uma técnica de geração de dados, é uma técnica de *oversampling*, contudo esta criação é baseada na semelhança entre os dados da classe

minoritária.

Seleção de *features*

A Etapa de Seleção de *features* apresenta bastante impacto na elaboração e na *performance* de modelos preditivos de *ML*, nomeadamente no ponto de vista de precisão, como do ponto de vista temporal. Revela-se uma etapa bastante importante, nomeadamente, em conjuntos de dados que apresentem elevada dimensionalidade, ou seja, em conjunto de dados em que a quantidade de *features* é bastante elevada, aumentando assim a complexidade do modelo. Posto isto, torna-se necessária esta etapa, nomeadamente devido aos seus benefícios:

- Redução da Complexidade do problema.
- Aumento da Performance do modelo.
- Diminuição do risco de fenómenos de *overfitting*, através da diminuição de *features* redundantes.
- Diminuição do tempo de treino do algoritmo.

Apesar desta etapa estar diretamente ligada com a eficácia dos modelos, esta etapa é demasiadas vezes ignorada, devido à ausência de conhecimento acerca de métodos de seleção de *features*, ou ao baixo tempo necessário para a implementação de projetos. Tendo em conta estes pressupostos, o objetivo nesta etapa passa pela implementação de métodos de seleção de *features* robustos, reutilizáveis e praticamente automatizados, dos quais são exemplo os seguintes métodos:

- Análise de *features* categóricas e correlação entre *features*.
- Importância das features através de modelos como o *RF* e o XGBoost.
- Coeficientes de *Lasso*.
- *RFE*

De um ponto de vista visual, esta etapa ao longo de uma *pipeline* de projetos de *ML*, representa a etapa em que existe a diminuição de quantidade de *features* existente, sem existir redução de quantidade de informação pertinente, ou seja, não afetando a precisão do modelo, apenas retirando as *features* redundantes, tal como se visualiza na Figura 3.5, onde a quantidade de *features* é reduzida, devido à redundância entre as mesmas.

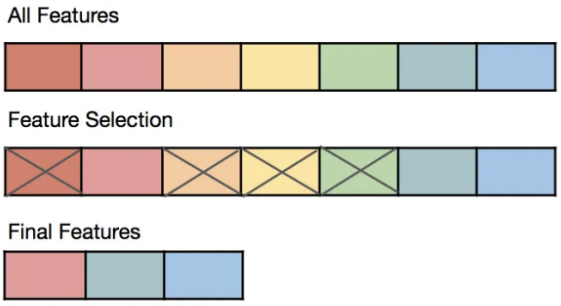


Figura 3.5: Redução do Número de *Features* [45].

Estratégia de validação de modelos

Relembrando que o objetivo da divisão do conjunto de dados total, em conjunto de dados de treino e conjunto de dados de teste, e que o pretendido é que o modelo se adapte corretamente a novos dados, surge o processo de *cross-validation*.

O *cross-validation* é um método estatístico de avaliação e de comparação de resultados de algoritmos de *Machine Learning*, que é mais estável do que uma simples divisão em conjunto de dados de treino e de teste, uma vez que permite caracterizar se o modelo é capaz de generalizar bem para diferentes conjuntos de dados. Enquanto que a simples divisão em conjunto de dados de treino e de teste é feita de forma aleatória, através de *cross-validation*, cada exemplo vai estar no conjunto de dados de treino e de teste uma vez, assegurando a generalização. Ao nível de desvantagens do uso de *cross-validation*, destaca-se o aumento do custo computacional, uma vez que faz o treino de k modelos e não o treino de apenas 1 modelo.

Neste processo, o conjunto de dados é dividido continuamente e são utilizados diversos modelos. A forma mais comum de utilização de *cross-validation* é denominada de *K-fold cross-validation*, onde k é um número, usualmente 5 ou 10, que representa o número de divisões. Por exemplo, quando se utiliza o processo de *5-fold cross-validation*, o conjunto de dados é dividido em 5 sub-conjuntos de tamanho igual. Uma vez realizada esta divisão, são procedidos ao treino e teste, sendo o primeiro *fold* utilizado como conjunto de teste e os restantes como conjunto de treino, ou seja, o modelo é construído com os *folds* 2-5 e depois é avaliado recorrendo ao primeiro *fold*. Posteriormente, utiliza-se o segundo *fold* como conjunto de dados de testes, e o restantes *folds* como conjunto de dados de treino. O processo é repetido iterativamente com os seguintes *folds* como junto de teste. Por último, extrai-se o valor da *accuracy* para cada um das divisões em conjunto de dados de treino e de teste [40]. Este processo encontra-se ilustrado na Figura 3.6.

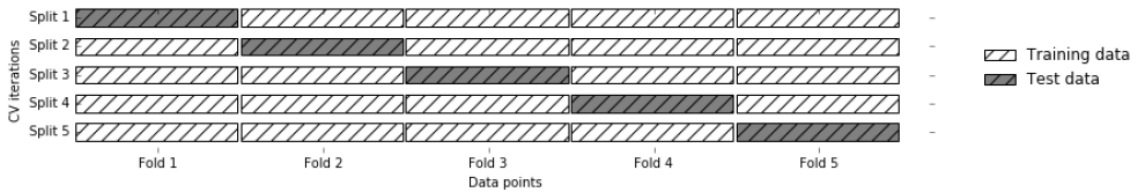


Figura 3.6: Processo de *Cross-Validation* [40].

Em problemas de classificação, a biblioteca *Scikit-learn* utiliza a técnica de *stratified k-fold cross-validation* para avaliar a performance dos modelos de maneira mais robusta e confiável. Nessa abordagem, os dados são divididos em k subconjuntos (*folds*) de forma estratificada, garantindo que cada *fold* preserve a mesma proporção de classes presentes no conjunto de dados original. Isso é particularmente importante em cenários de classificação com classes desbalanceadas, pois assegura que cada *fold* represente adequadamente todas as classes, evitando *bias* na validação do modelo.

Durante o processo de *k-fold cross-validation*, o modelo é treinado k vezes, cada vez utilizando $k-1$ *folds* para treinamento e o *fold* restante para validação. A média das métricas de desempenho dos k modelos fornece uma estimativa mais confiável da performance do

modelo em dados não vistos, permitindo ajustes finos e escolhas informadas dos hiperparâmetros. Este processo é ilustrado na Figura 3.7, onde se observa a manutenção das proporções de classes em cada *fold*, assegurando uma avaliação justa e representativa do modelo.

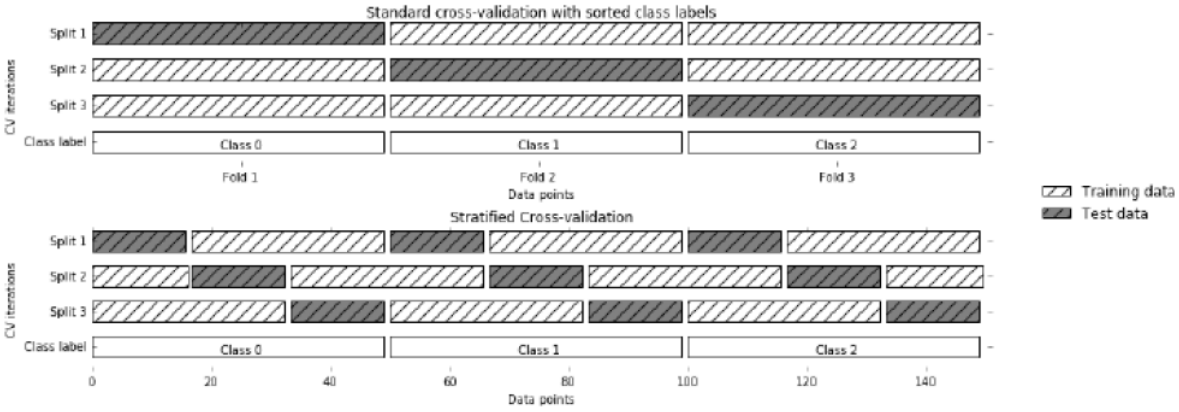


Figura 3.7: Processo de *Stratified k-fold cross-validation* [40].

Tuning de hiperparâmetros

O processo de *tuning* de hiperparâmetros, ou seja, a otimização dos parâmetros de um modelo de *Machine Learning*, não é uma tarefa que seja facilmente realizada de forma manual. Contudo, apresenta resultados significativos no desempenho do modelo nos dados de validação e de teste. Estes parâmetros controlam aspetos importantes dos modelos, sendo que diferentes modelos apresentam diferentes parâmetros. Caso estes parâmetros não sejam escolhidos corretamente, pode-se cair em fenómenos de *overfitting* ou *underfitting*.

Dada a complexidade dos modelos e a dificuldade em se conseguir obter o valor dos parâmetros ótimos de forma manual, existem métodos computacionais capazes de fazer este processo de forma mais eficiente. Como esta tarefa é bastante comum, já existem métodos padrão na biblioteca *Scikit-learn* que facilitam sua execução. Os métodos mais utilizados são o *Grid Search* e o *Random Search*.

O *Grid Search* é um método que realiza uma busca exaustiva sobre um espaço especificado de parâmetros. Ele testa todas as combinações possíveis dos parâmetros fornecidos, avaliando o desempenho de cada combinação com base numa métrica de validação. Embora seja um método robusto, pode ser computacionalmente intensivo, especialmente para grandes espaços de parâmetros.

Por outro lado, o *Random Search* seleciona aleatoriamente combinações de parâmetros do espaço especificado e as avalia, sendo geralmente mais eficiente em termos de tempo, pois não testa todas as combinações possíveis, mas ainda assim pode encontrar bons valores de hiperparâmetros.

3.4 Algoritmos de aprendizagem supervisionada

Nesta secção, explora-se diversos algoritmos fundamentais de aprendizagem supervisionada, que são essenciais para a análise preditiva e classificação de dados em problemas

reais.

Começa-se pela regressão Linear e logística, que são métodos tradicionais na modelação de relacionamentos entre *features* e previsão de resultados numéricos ou resultados rotulados, passando pelo *NB*, que utiliza o teorema de *Bayes* para classificação probabilística. Posteriormente, analisa-se a *DT*, que pode ser aplicada tanto em problemas de regressão como em problemas de classificação, e a sua respetiva extensão, o algoritmo *RF*, que contempla múltiplas árvores para melhorar a precisão, através da redução da probabilidade de ocorrência de *overfitting*.

Além disso, abordaremos o *NN*, que se baseia na proximidade entre instâncias para realizar previsões, e o SVM, ideal para classificação binária e multi-classe por meio da busca por hiperplanos otimizados no espaço de atributos.

Por último, explora-se as técnicas de *ensemble*, como o *XGBoost*, que utiliza *gradient boosting* para melhoraria do desempenho do modelo, e as abordagens de *voting* e *stacking*, que combinam as previsões de múltiplos modelos para obtenção de resultados mais robustos.

Cada algoritmo será apresentado com exemplos visuais e aplicações práticas, destacando-se as suas características distintas.

3.4.1 *Linear regression* e *logistic regression*

O algoritmo mais simples de aprendizagem supervisionada é a regressão Linear, que representa uma reta com uma variável independente (x) e uma variável dependente (y), sendo que o objetivo desta reta é a obtenção do menor erro possível na previsão do valor da variável dependente (y = mx + b).

Num ponto de vista mais realista dos problemas existentes, esta regressão transforma-se numa regressão multivariada, onde existem várias variáveis independentes ($y = m_1x_1 + m_2x_2 + m_3x_3 + \ldots + m_nx_n$), onde n representa o número de atributos, sendo que o objetivo é a descoberta dos melhores valores para m_i, de forma a reduzir a soma dos quadrados dos erros.

A regressão logística é uma técnica de *Machine Learning* recomendada para situações em que a variável dependente é binária, ou seja, em problema de aprendizagem supervisionada de classificação binária, ou seja em problemas de resposta binária.

Nos problemas de regressão logística, estima-se a probabilidade de ocorrência de um evento com base nas variáveis independentes do conjunto de dados. Como o resultado é uma probabilidade, a variável dependente encontra-se limitada entre 0 e 1, posto isto deve-se definir um *threshold* para diferenciar entre classes, por exemplo se for superior a 50% o algoritmo pertence à classe 1, enquanto se for inferior a 50% representa a classe 0. A probabilidade é calculada de acordo com um função logística, onde o *output* está entre 0 e 1, representada pela equação 3.3.

$$P(Y = 1|X) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n)}} \tag{3.3}$$

A equação 3.3 representa a probabilidade de um determinado evento acontecer, dividido pela probabilidade de não ocorrência do mesmo evento, através de uma função sigmóide.

A nível de problemas de classificação utiliza-se a regressão logística, uma vez que na regressão linear univariada existem valores superiores a 1 e valores inferiores a 0, o que não é o pretendido. De forma a distinguir os objetivos de ambas as regressões, verifica-se esta dualidade na representação da Figura 3.8.

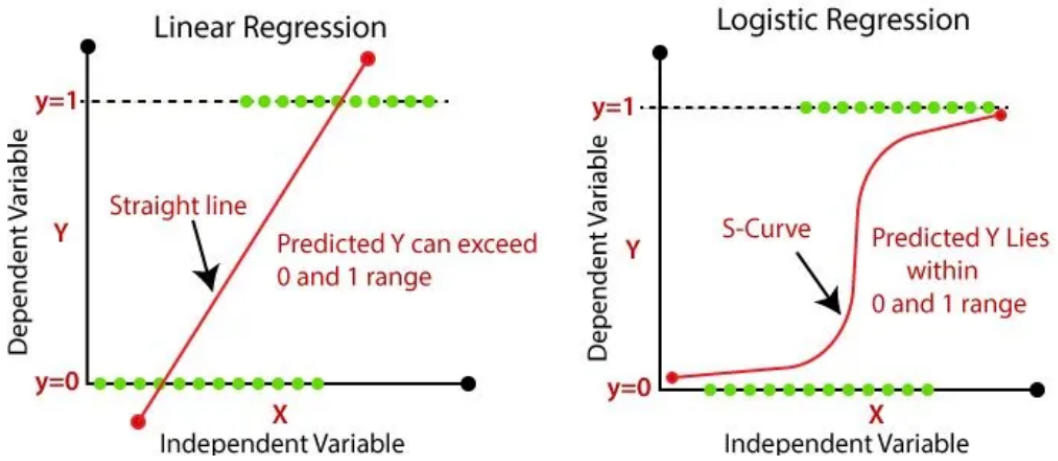


Figura 3.8: Representação visual do Algoritmo de Regressão Linear e Regressão Logística [1].

3.4.2 *Naive Bayes*

Naive Bayes é um algoritmo baseado no teorema de *Bayes*. Este teorema descreve a probabilidade de um evento acontecer, com base no conhecimento prévio de condições que podem estar relacionadas para esse evento. O termo *naive* (ingénuo) deve-se ao fato de o algoritmo não ter como base de partida uma certeza inicial, a partir do qual se constrói esse grau analisando as frequências presentes nos dados que temos disponíveis. Este algoritmo assenta na seguinte equação:

$$P(C_k|X) = \frac{P(C_k) \cdot P(X|C_k)}{P(X)} \tag{3.4}$$

Onde:

$P(C_k|X)$ é a posterior probabilidade da classe $C_k$dadas as *features* de entrada$X$.
$P(C_k)$ é a probabilidade da classe C_k.
$P(X|C_k)$ é a probabilidade de ter certas *features* X sabendo a classe C_k.
$P(X)$ é a probabilidade de ter certas *features* de entrada.

3.4.3 *Decision Tree*

DT é um algoritmo de aprendizagem supervisionada que tanto pode ser utilizado em problemas de regressão, como em problemas de classificação. Neste algoritmo as árvores de decisão são geradas tendo por base o conjunto de dados de treino, sendo que resulta numa estrutura em forma de árvore para classificar os novos dados.

A árvore criada é composta por múltiplos níveis de *nodes* que representam as *features* do conjunto de dados e por *branches*, que contém os possíveis valores que cada nó pode ter, sendo estes *nodes*: *root node*, *decision nodes* (*internal nodes*) e *leaf nodes* (*terminal nodes*).

Para cada novo caso, o processo começa a partir do *root node* e utilizando os valores de cada *feature*, a árvore é percorrida até chegar ao *leaf node*, onde é realizada a sua classificação. A figura 3.9 apresenta a representação visual de uma árvore de decisão [46].

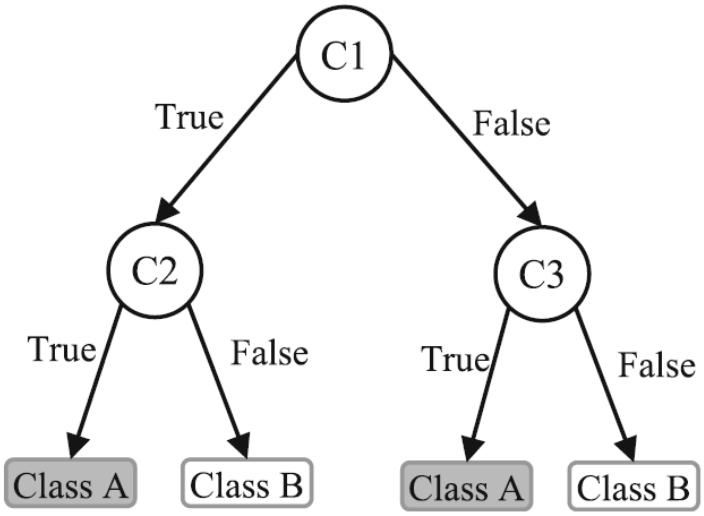


Figura 3.9: Representação visual do algoritmo de *Decision Tree* [46].

Este tipo de algoritmo pode ser utilizado com diversos intuitos, contudo para demonstração do seu potencial, tendo em conta a área da saúde, escolheu-se a classificação do risco de prevenção de um ataque cardíaco, representado na Figura 3.10.

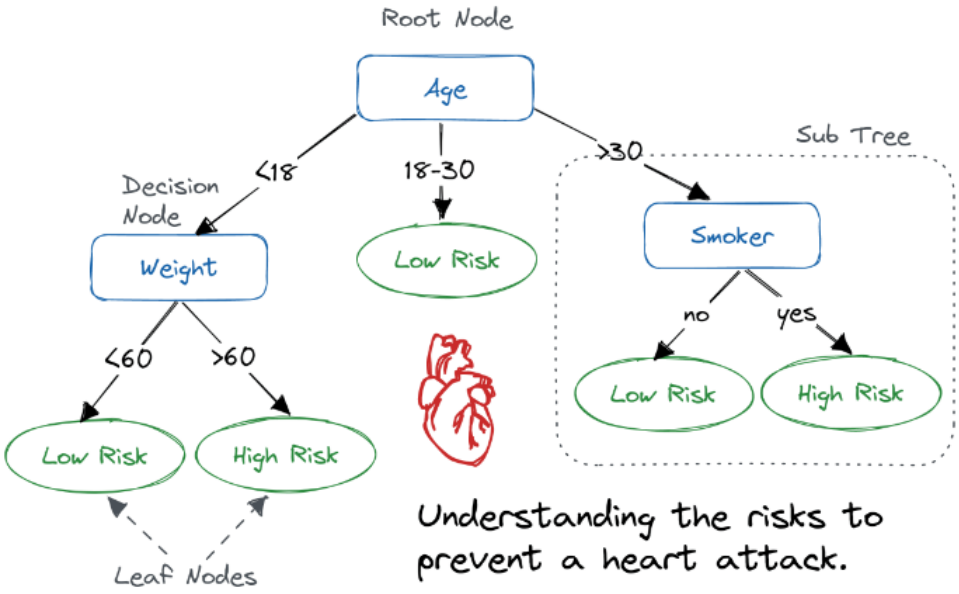


Figura 3.10: Representação visual do algoritmo de *Decision Tree* aplicado na vertente da saúde [47].

Pela análise da árvore de decisão na Figura 3.10, verifica-se o *root node* e os *decision nodes* são as características da pessoa, tais como a idade, peso e se é fumadora, sendo que o *leaf node* é o resultado final, ou seja, o baixo ou alto risco de previsão de um ataque cardíaco [47].

3.4.4 *Random Forest*

O algoritmo *Random Forest* é uma técnica de aprendizagem supervisionada usada principalmente para problemas de classificação e regressão. Ele combina os resultados de várias árvores de decisão, formando uma "floresta"de árvores. A previsão final é obtida calculando a média (para regressão) ou a moda (para classificação) dos resultados das árvores, o que melhora a precisão e reduz o risco de *overfitting*. Cada árvore de decisão na floresta tem sua própria estrutura, mas todas contribuem para o mesmo resultado final. A precisão do algoritmo aumenta à medida que se adicionam mais árvores.

Primeiro, o algoritmo constrói cada árvore de decisão usando uma amostra aleatória dos dados de treino. Em seguida, seleciona a melhor *feature* para dividir o nó raiz e cria nós filhos com mini árvores de decisão. Este processo continua até que o número desejado de árvores seja alcançado. Cada mini árvore testa os dados aleatórios em seus ramos. Finalmente, a classificação é feita para os dados de teste através da votação maioritária das árvores de decisão.

Um exemplo de como este algoritmo funciona, pode ser visualizado na Figura 3.11.

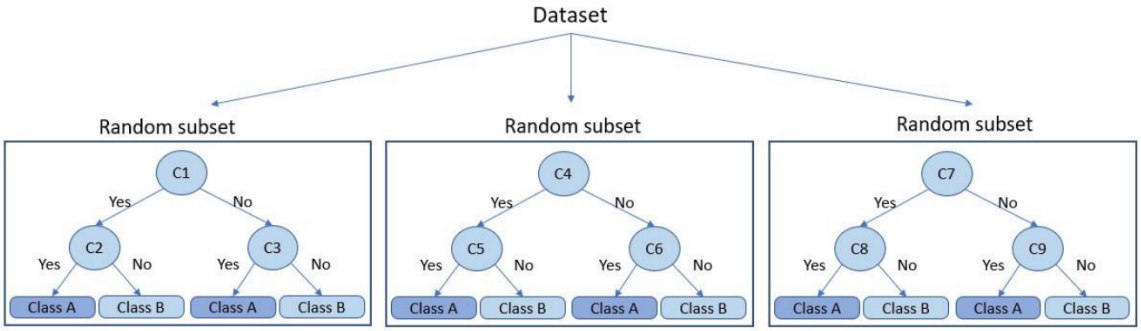


Figura 3.11: Representação visual do algoritmo de *Random Forest* [46].

3.4.5 *K-Nearest Neighbors*

O *K-nearest neighbour* é um dos algoritmos mais simplista utilizados em problemas de regressão e classificação em aprendizagem supervisionada. Este algoritmo é baseado no princípio de proximidade, onde objetos semelhantes devem estar próximo de outros semelhantes. O k presente no nome do algoritmo representa o número de vizinhos próximos que são considerados no problema de classificação, onde a utilização de diferentes números de k, origina diferentes resultados no processo de classificação, tal como se verifica na Figura 3.12, onde por exemplo para k = 3, o objeto é classificado como preto, enquanto que no k = 5, o objeto é classificado como vermelho, uma vez que se utiliza a moda [46].

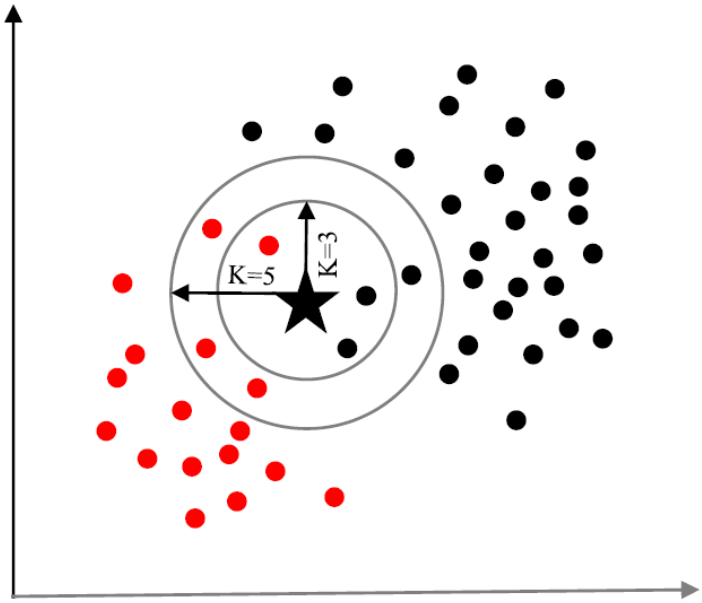


Figura 3.12: Representação visual do algoritmo de *NN* [46].

3.4.6 *Support Vector Machine*

O *SVM* é um algoritmo que tanto permite realizar a classificação de dados linear, como de dados não lineares e tanto pode ser utilização em classificação binária como em classificação multi-classe. Este algoritmo começa pela realização de uma mapeamento do conjunto de dados para um espaço de n dimensões, onde n é o número de *features* no conjunto de dados. Posteriormente, o *SVM* representa estes ponto num espaço bidimensional, e procura um hiperplano ideal, que separe os dados em duas classes enquanto maximiza a distância marginal (distância entre o hiperplano de decisão e a instância que representa a classe) entre as duas classes, diminuindo assim os erro de classificação. Na Figura 3.13 pode-se verificar um exemplo visual do algoritmo de *SVM* [46].

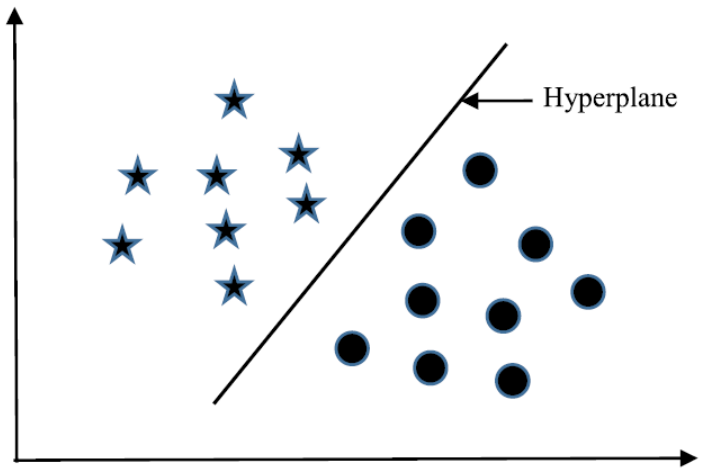


Figura 3.13: Representação visual do algoritmo *Support Vector Machine* [46].

3.4.7 *Ensemble*

Ensemble learning é uma técnica de *Machine Learning* em que existe a combinação de múltiplos modelos de forma a melhorar a performance de modelos de aprendizagem supervisionada de previsão ou de classificação, ou mesmo reduzir a probabilidade de realizar uma previsão ou classificação errada. Neste contexto, esta técnica em vez de depender de apenas um modelo, utiliza vários modelos para conseguir melhorar o algoritmo, tornando este mais preciso [33].

XGBoost

XGBoost é um algoritmo de *Machine Learning* baseado em árvores de decisão, bastante popular em problemas de Aprendizagem Supervisionada, tais como problemas de regressão e de classificação. Trata-se de uma algoritmo com a técnica de *gradient boosting*, onde os modelos são construídos de forma sequencial, e são constantemente melhorados.

Além disto, o *XGBoost* é um excelente algoritmo, uma vez que evita o fenómenos de *overfitting* através da técnica de regularização, ou seja, limita o crescimento das árvores. O *XGBoost* implementa técnicas de regularização tanto nos nós das árvores de decisão individuais, como na combinação das diversas árvores. Isso inclui termos de penalidade adicionados à função objetivo que o modelo otimiza durante o treino. As formas comuns de regularização incluem penalidades L1 (lasso) e L2 (ridge).

Técnica de *Voting*

O *Voting ensemble* é um método de *ensemble learning*, que funciona de forma parecido com um sistema eleitoral, onde vários modelos de *Machine Learning* são treinados de forma individual, mas as suas previsões são combinadas através de votação [33]. Neste sentido, neste tipo de *ensemble*, diferenciam-se dois tipos de votação: *Soft* e *Hard Voting*.

- *Hard Voting*: O *hard voting* é aplicado nas classes das previsões de cada modelo, em que as previsões de cada modelo são consideradas como votos e a classe que recebe mais de metade dos votos é selecionada como a classe da previsão final.
- *Soft Voting*: Cada modelo fornece uma estimativa de probabilidade para cada classe e a média dessas probabilidades é calculada para determinar a previsão final.

Técnica de *Stacking*

O *Stacking* é uma técnica de *ensemble learning* onde as previsões de diversos modelos de *Machine Learning* são adicionadas a um conjunto de dados. O primeiro passo neste tipo de técnica é a definição dos modelos de *Machine Learning* utilizados como modelos base, ou seja, os modelos dos quais as previsões servem para a criação do conjunto de dados que será o *input* do modelo meta. Uma vez definidos estes modelos, faz-se o seu treino recorrendo ao conjunto de dados de treino, sendo que posteriormente realizam-se as previsões recorrendo ao conjunto de dados de teste, sendo estas previsões as *features* de *input* para o modelo Meta. Por fim, o modelo meta é treinado com recurso ao conjunto de dados com as previsões dos modelos base e testado. Neste sentido, o objetivo deste tipo de técnica é melhorar o desempenho, de forma a que este seja superior ao que aos resultados dos modelos de forma individual [33].

3.5 Métricas de avaliação

Uma vez pensado o *workflow* até ao processo de utilização do conjunto de dados de teste, segue-se a etapa de medição do desempenho do modelo, verificando-se se este efetua uma boa generalização do problema, ou seja, a qualidade das previsões obtidas. Posto isto, apresentam-se as principais métricas utilizadas em problemas de classificação, bem como as respetivas fórmulas.

Nos modelos de classificação, mais concretamente de classificação binária (entre duas classes), o objetivo é a previsão de acordo com as duas classes possíveis, geralmente em casos de positivo ou negativo, presença ou ausência de um determinado evento. No âmbito da saúde, um exemplo é a presença ou ausência de uma determinada doença, possivelmente a doença de *Parkinson*. Esta avaliação em problemas de classificação é realizada através da comparação entre a classe real e a classe obtida através dos modelos de previsão.

3.5.1 Matriz de confusão

A qualidade de um modelo de classificação pode ser visualizada através de uma matriz de confusão, matriz esta que apresenta os seguinte tipos de dados:

- *Verdadeiro positivo (VP)*: Número de exemplo da classe positiva classificados corretamente.
- *Falso positivo (FP)*: Número de exemplo da classe negativa classificados como sendo da classe positiva.
- *Verdadeiro negativo (VN)*: Número de exemplo da classe negativa classificados corretamente.
- *Falso negativo (FN)*: Número de exemplo da classe positiva classificados como sendo da classe negativa.

Uma Matriz de Confusão de classificação binária é um *array* de 2 por 2, onde as linhas representam as classes reais, enquanto que as colunas representam as classes previstas, onde cada entrada apresenta uma frequência de amostra, tal como se verifica na Figura 3.14. Esta visualização é relevante, uma vez que facilita a representação de quantos casos foram classificados de forma errada em cada uma das classes. Na figura 3.14, *TP* (*True positive*), *TN* (*True negative*), *FP* (*False positive*) e *FN* (*False negative*) representam os verdadeiros positivos, os verdadeiro negativos, os falsos positivos e os falsos negativos, respetivamente.

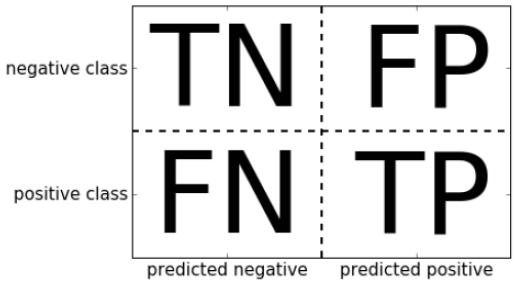


Figura 3.14: Matriz de classificação Binária [40].

Outro fator bastante importante da matriz de confusão, é que através desta representação e dos valores dos dados *VP*, *FP*, *VN* e *FN*, se conseguem calcular outras métricas de avaliação: Taxa de falsos positivos (percentagem de acerto na classe negativa), taxa de verdadeiros positivos (percentagem de acerto na classe positiva), *accuracy*, *precision*, *recall*, *F1-Score*, sensibilidade (erro do tipo 2) e especificidade (erro do tipo 1).

3.5.2 Sensibilidade

A sensibilidade corresponde à taxa de acerto na classe positiva. No contexto do problema, representa a possibilidade de previsão da doença de *Parkinson*, e é calculada pela quação 3.5.

$$\text{Sensibilidade} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}} \tag{3.5}$$

3.5.3 Especificidade

A especificidade corresponde à taxa de acerto na classe negativa. No contexto do problema, representa a possibilidade de previsão da não existência da doença de *Parkinson*, e é calculada pela equação 3.6.

$$\text{Especificidade} = \frac{\text{Verdadeiros Negativos}}{\text{Verdadeiros Negativos} + \text{Falsos Positivos}} \tag{3.6}$$

3.5.4 *Accuracy*

A *accuracy* é calculada pela equação 3.7.

$$\text{Accuracy} = \frac{\text{Verdadeiros Positivos} + \text{Verdadeiros Negativos}}{\text{Total de Amostras}} \tag{3.7}$$

3.5.5 Precisão

A precisão é a proporção de verdadeiros positivos em relação ao número total de previsões positivas feitas pelo modelo. A *precisão* é calculada pela equação 3.8.

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Positivos}} \tag{3.8}$$

3.5.6 *Recall*

O *recall* é a proporção de verdadeiros positivos em relação ao número total de exemplos positivos, sendo que é calculado pela equação 3.9.

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos}}{\text{Verdadeiros Positivos} + \text{Falsos Negativos}} \tag{3.9}$$

3.5.7 *F1-Score*

O *F1-Score*, uma média harmónica, que varia entre 0 e 1, calculada com base na precisão e no *recall*. Esta métrica é calculada pela equação 3.10.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precisão} \cdot \text{Sensibilidade}}{\text{Precisão} + \text{Sensibilidade}} \tag{3.10}$$

3.5.8 *Receiver operating characteristics*

A curva *Receiver operating characteristics (ROC)* é um gráfico que permite proceder com a avaliação de um classificador binário, sendo este uma das métricas mais utilizadas para esta avaliação.

Essa visualização leva em consideração a taxa de verdadeiros positivos e a taxa de falsos positivos, onde no eixo dos X está representada a especificidade, enquanto que no eixo dos Y está representa a sensibilidade. Neste gráfico podem-se representar diversos classificadores e verificar qual deles apresenta melhores resultados, sendo que quanto mais próximo o classificador estiver do topo do eixo dos Y melhor será o classificador. O desempenho total é gerado pela *AUC*, que representa a área de forma dimensional debaixo da curva, onde um valor de aproximadamente 0,5 representa um classificador aleatório, enquanto que um valor de aproximadamente 1 representa um classificador ideal, que é capaz de diferenciar duas classes, tal como se pode verificar na Figura 3.15.

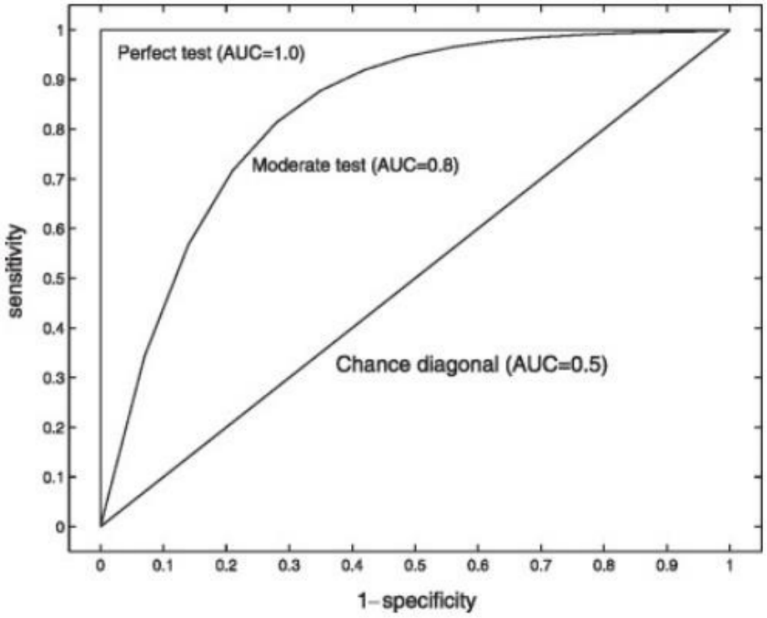


Figura 3.15: Representação gráfica da curva ROC.

Capítulo 4

O Caso de estudo

Neste capítulo apresenta-se uma descrição detalhada do *dataset* utilizado, que foi obtido a partir do repositório da Universidade da Califórnia em *Irvine*. Explica-se as diferentes *features* vocais existentes no mesmo, abrangendo seis grupos principais relacionados ao processamento de sinais de voz. Além disso, detalha-se a metodologia de recolha dos dados e processamento de dados, bem como se apresenta uma discussão dos resultados obtidos a partir das análises realizadas.

4.1 Levantamento dos dados

O *dataset* utilizado foi obtido num repositório da Universidade de Califórnia *Irvine*, dataset este criado por C. Sakar et al.[1] Os dados utilizados neste estudo foram recolhidos de 188 pacientes, sendo 107 destes do sexo masculino e 81 do sexo feminino, com doença de *Parkinson*, com idades entre os 33 e os 87 anos (65 ± 10.9), do Departamento de Neurologia da Faculdade de Medicina da Universidade de Istambul (İstanbul Medipol Üniversitesi, 2018). Neste estudo utilizou-se um grupo de controlo com cerca de 64 indivíduos saudáveis, sendo 23 do sexo masculino e 41 do sexo feminino, com idades compreendidas entre 41 e 82 anos (61.1 ± 8.9). No processo de levantamento dos dados, utilizou-se um microfone com uma frequência de 44.1 Hz, onde cada indivíduo pronunciou 3 vezes a vogal "a", sendo este processo de captura de dados acompanhado por profissionais. Da recolha de dados resultou um *dataset* composto por 6 grupos de *features* relacionadas com processamento de sinal denominadas por "*Baseline Features*", "*Time Frequency Features*", "*Mel Frequency Cepstral Coefficients*", "*Wavelet Transform based Features*", "*Vocal Fold Features*" e "*Tunable Q-factor wavelet transform Features*".

Dentro do conjunto das *Baseline Features*, existem as seguinte *features*: *Jitter Variants*, *Shimmer Variants*, *Fundamental Frequency Parameters*, *Harmonicity Parameters*, *Recurrence period density entropy (RPDE)*, *Detrended fluctuation analysis (DFA)* e *Pitch period entropy (PPE)*.

Jitter Variants apresenta um total de 5 *features* e representa a variação fundamental da frequência a partir de um ciclo periódico para o próximo ciclo. Os valores desta variável

[1] Repositório da Universidade de Califórnia, disponível em `https://archive.ics.uci.edu/dataset/470/parkinson+s+disease+classification`.

mudam consoante a desordem da voz, isto significa que é responsável por uma qualidade de voz rouca.

Shimmer Variants apresenta 6 *features* e representa a variação de amplitudes de períodos consecutivos.

Fundamental Frequency Parameters apresenta 5 *features* e representa a frequência da vibração das cordas vocais. *Harmonicity Parameters* apresenta 2 *features* e representa o ruído causado pelo parcial fecho das cordas vocais. *RPDE* apresenta 1 *feature* e representa a capacidade de as cordas vocais sustentarem oscilações constantes e quantifica os desvios F0. *DFA* apresenta 1 *feature* e é utilizado para avaliar a similaridade do ruído produzido por um fluxo de ar nas cordas vocais. *PPE* apresenta 1 *feature*, que representa o controlo da frequência fundamental F0 utilizando uma escala logarítmica.

No que concerne as *Time Frequency Features*, existem 3 tipos *features* relativas: *Intensity Parameters*, *Formant Frequencies* e *Bandwith*.

Intensity parameters apresenta 3 *features* e é uma característica relacionada com a potência do processamento de sinal da fala e é medida em Db. Estão presentes os valores mínimos, médios e máximos de intensidade.

Formant frequencies apresenta 4 *features* e representa as frequências amplificadas pelo trato vocal. *Bandwidth* apresenta 4 *features* e representa a diferença entre a frequência superior e inferior numa banda contínua de frequências.

No que concerne à *feature* de *MFCCs*, esta apresenta 84 *features* e baseia-se na perceção auditiva humana e não consegue captar frequências superiores a 1 Khz. O tom padrão é representado numa escala de frequências de MEL, com a finalidade de captar características importantes na fonética da fala.

No que concerne a *feature* de *Wavelet Transform-Based*, esta apresenta 182 *features* e permite a análise de sinais oscilatório da transformada discreta de *wavelet*.

A *TQWT* é uma transformada de *wavelet* flexível, onde os seus parâmetros são o *Q-Factor*, a redundância e o número de níveis. O *Q-Factor*, o Q, afeta o comportamento oscilatório da *wavelet*, que representa o número de oscilações que a *wavelet* exibe.

No que concerne as *Vocal Fold Features*, esta apresenta 4 tipos de *Features*: *Glottis quotient (GQ)*, *Glottal to noise excitation (GNE)*, *Vocal fold excitation ratio (VFER)* e *Empirical mode decomposition (EMD)*.

GQ apresenta 3 *features* e fornece informações sobre a duração da abertura e fecho da *glottis*. Sendo representada como uma medida de periodicidade nos movimentos da *glottis*.

GNE apresenta 6 *features* e quantifica a extensão do ruído causado pelo fecho incompleto das cordas vocais, no sinal da fala.

VFER apresenta 7 *features* e quantifica a quantidade de ruído produzido devido á vibração patológica das cordas vocais, recorrendo a conceitos de energia não-linear e entropia.

EMD apresenta 6 *features* e representa a decomposição do sinal da fala em componentes de sinal elementares recorrendo a funções de base adaptativa e os valores obtidos de energia/entropia obtidos a partir destes componentes são utilizados para quantificar o ruído.

O último grupo de *features* é conhecido como *TQWT*, que apresenta 432 *features*. Estas *features* são designadas como *TQWT Features* e são usadas para decompor o sinal em sub-faixas. As sub-faixas são então empregadas na extração de características estatísticas.

A tabela 4.1 apresenta uma lista detalhada das diferentes *features* utilizadas na análise de sinais de voz, incluindo suas descrições e o número de *features* associadas a cada categoria. As *features* são divididas em dois grupos principais: *features primárias* e *features derivadas*. As *features primárias* são diretamente extraídas dos sinais de voz, enquanto as *features derivadas* são calculadas a partir de transformações ou decomposição desses sinais.

Tabela 4.1: Descrição das *features* presentes no conjunto de dados

Features		
Nome	**Descrição**	**Quantidade**
	Features Primárias	
	Baseline Features	
Jitter Variants	Variação fundamental da frequência a partir de um ciclo periódico para o próximo ciclos. Os valores desta variável mudam consoante a desordem da voz, isto significa que é responsável por uma qualidade de voz rouca.	5
Shimmer	Variação de amplitudes de períodos consecutivos.	6
Fundamental frequency parameters	Frequência da vibração das cordas vocais.	6
Harmonicity parameters	Ruído causado pelo parcial fecho das cordas vocais.	2
Recurrence Period Density Entropy	Capacidade de as cordas vocais sustentarem oscilações constantes e quantifica os desvios F0.	1
Detrended Fluctuation Analysis	Avaliação da similaridade do ruído produzido por um fluxo de ar nas cordas vocais.	1
Pitch Period Entropy	Controlo da frequência fundamental F0 utilizando uma escala logarítmica.	1
	Time Frequency Features	
Intensity Parameters	Característica relacionada a potência do processamento de sinal da fala e é medida em Db. Estão presentes os valores mínimos, médios e máximos de intensidade.	3
Formant Frequencies	Frequências amplificadas pelo trato vocal.	4
Bandwidth	Diferença entre a frequência superior e inferior numa banda contínua de frequências.	4
	Vocal Fold Features	
GQ	Informações sobre a duração da abertura e fecho da glottis. Sendo representada como uma medida de periodicidade de movimentos da glottis.	3
GNE	Quantifica a extensão do ruído causado pelo fecho incompleto das cordas vocais, no sinal da fala.	6
VFER	características importantes na fonética da fala	7
EMD	Quantifica a quantidade de ruído produzido devido á vibração patológica das cordas vocais, recorrendo a conceitos de energia não-linear e entropia.	6
	Features derivadas	
	Mel Frequency Cepstral Coefficients	
MFCCs	Baseia-se na perceção auditiva humana e não consegue captar frequências superiores a 1 Khz. O tom padrão é representado numa escala de frequêmcias de MEL, com a finalidade de captar características importantes na fonética da fala.	84
	Wavelet transform-based Features	
WT	Permite analisar sinais oscilatórios da transformada discreta de wavelet.	182
	Tunable Q-Factor wavelet transform	
TQWT	decomposição do sinal *Eletromiografia (EMG)* em subfaixas e estas são utilizadas para a extração de característica estatísticas.	432

4.2 Análise dos dados

O conjunto de dados utilizado é composto por 756 linhas, as quais representas as 3 vezes que cada indivíduo pronunciou a vogal "a", sendo assim representantes dos 252 indivíduos do estudo (188 pacientes com *Parkinson* e 64 indivíduos do grupo de controlo). Neste sentido, verificou-se ainda que o conjunto de dados apresenta elementos do tipo *int64* e do tipo *float64*, não apresentando assim variáveis categóricas. Esta informação está presente na figura 4.1.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 756 entries, 0 to 755
Columns: 755 entries, id to class
dtypes: float64(749), int64(6)
memory usage: 4.4 MB
```

Figura 4.1: Informação geral acerca do Conjuntos de dados, incluindo tipo de variáveis.

Uma parte sempre importante na análise do conjunto de dados, é a verificação da existência da valores omissos. Na Figura 4.2 demonstra-se que não existe nenhum valor omisso no conjunto de dados, em qualquer uma das suas *features*.

```
any_null_in_dataframe = df.isnull().any().any()
any_null_in_dataframe
```

```
]: False
```

Figura 4.2: Verificação de valores omissos no conjunto de dados.

A Tabela 4.2 fornece uma visão geral das características fundamentais do conjunto de dados analisado, evidenciando a informação previamente referida nas Figuras 4.1 e 4.2, relativa à quantidade de *features*, quantidade de linhas, quantidade de valores nulos e tipo de variáveis existentes, respetivamente.

Tabela 4.2: Descrição do conjunto de dados

Característica	**Detalhes**
Número de *features*	755
Número de linhas	756
Número de valores nulos	0
tipos de variáveis	2 numéricas (*int64* e *float64*)

Num problema de classificação, é crucial que o conjunto de dados esteja equilibrado. Com base neste princípio, foram analisados os atributos de uma classe binária (0 ou 1), onde 0 representa a ausência e 1 a presença da doença de *Parkinson*. Conforme evidenciado na Figura 4.3 , verifica-se um desequilíbrio significativo nos registos das classes: aproximadamente 192 casos na classe 0 (sem *Parkinson*) e 565 casos na classe 1 (com *Parkinson*), o que representa 25,40% de casos na classe 0 e 75,60% de casos na classe 1, respetivamente. Com esta representação dos dados, verifica-se que o *dataset* se encontra desbalanceado, necessitando de uma etapa de balanceamento dos dados.

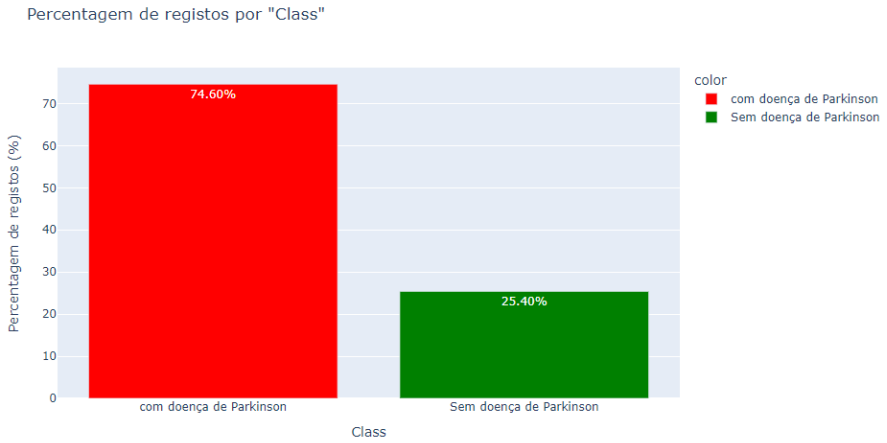


Figura 4.3: Percentagem de casos da não existência e da existência da doença de *Parkinson*.

Ainda na perspetiva de analisar o balanceamento do conjunto de dados, analisou-se a variável representativa do género dos pacientes, sendo esta binária (0 ou 1), representa indivíduos do sexo feminino e do sexo masculino, respetivamente. Na Figura 4.4, verifica-se que a quantidade de registos por género se encontra balanceado, verificando-se 366 pessoas do sexo feminino e 490 pessoas do sexo masculino, ou seja, 48,41% do sexo feminino e 51,59% do sexo masculino.

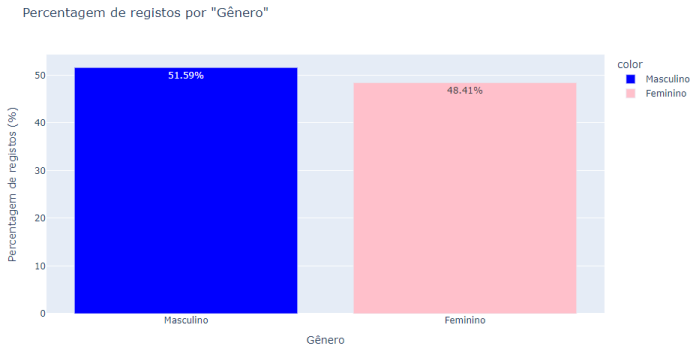


Figura 4.4: Percentagem de pessoas do sexo feminino e do sexo masculino.

A tabela 4.3 apresenta estatísticas descritivas como média, desvio padrão, mínimo, mediana, 1º e 3º quartis para as *Jitter features*, incluindo *stdDevPeriodPulses*, *locPctJitter*, *locAbsJitter*, *rapJitter* e *ppq5Jitter*. Essas métricas são fundamentais na análise de distúrbios vocais e são frequentemente utilizadas na caracterização de padrões de voz. Note que os valores apresentados não foram normalizados e demonstram variações significativas em escalas diferentes, o que representa a necessidade de pré-processamento adequado, como a utilização de técnicas como *Min-Max Scaler*, de forma a que se garanta que todas as características contribuem igualmente nas análises subsequentes.

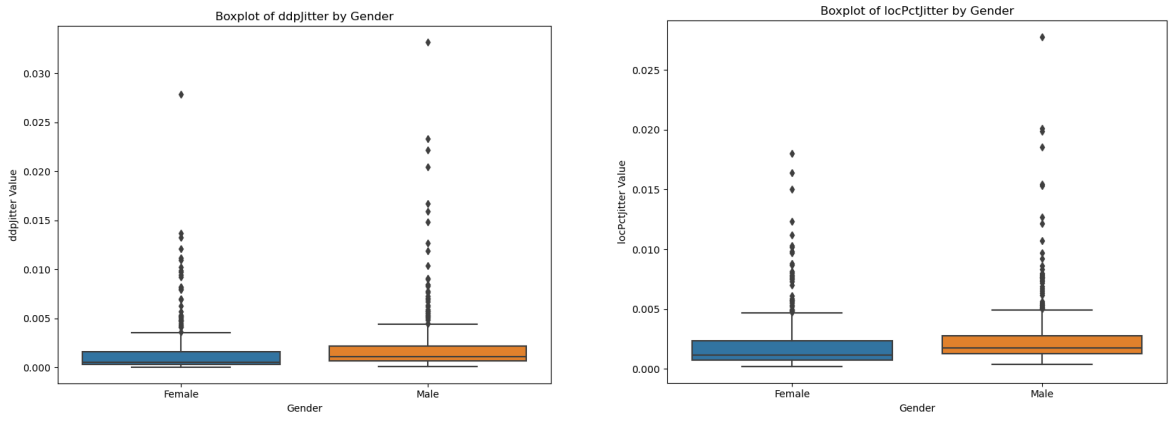
No Apêndice A apresenta-se a informação das estatísticas descritivas com detalhe das *baseline features*, com o intuito de se mostrar a necessidade de utilização do *Min-Max Scaler*.

Tabela 4.3: Estatísticas descritivas das *Jitter features*

Feature	**Mean**	**Std**	**Min**	**25%**	**50%**	**75%**
locPctJitter	0.002324	0.002628	0.000210	0.000970	0.001495	0.002520
locAbsJitter	0.000017	0.000023	0.000001	0.000005	0.000010	0.000018
rapJitter	0.000605	0.000981	0.000020	0.000150	0.000280	0.000650
ppq5Jitter	0.001159	0.001677	0.000050	0.000370	0.000650	0.001253
ddpJitter	0.001815	0.002942	0.000050	0.000450	0.000840	0.001952

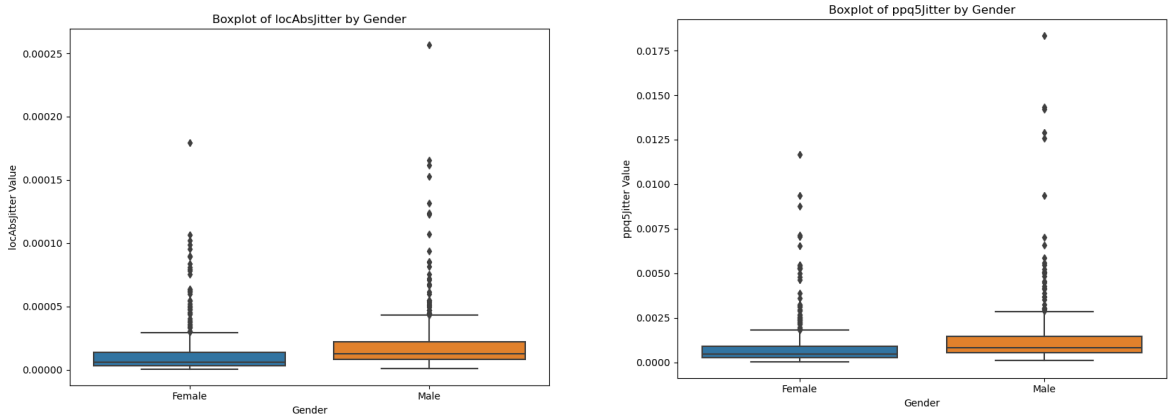
De forma a analisar-se as variações nas *Jitter features* entre os géneros feminino e masculino neste estudo, foram criados *boxplots* para cada *feature* relevante. Os *boxplots* revelaram diferenças significativas na distribuição nas *Jitter features* entre os dois grupos. Por exemplo, ao observar os valores máximos de *jitter* para cada género, nota-se que o valor máximo para os homens geralmente corresponde aproximadamente ao terceiro quartil para as mulheres, tal como se verifica nas Figuras 4.5, 4.6. Essas diferenças destacam a importância de uma análise diferenciada por género, pois sugerem que as características acústicas medidas podem variar substancialmente entre homens e mulheres.

Além disso, as medianas e os quartis também exibiram variações significativas entre os géneros, indicando uma possível divergência nas propriedades acústicas das vozes masculina e feminina. Essas observações são fundamentais para o desenvolvimento de modelos mais precisos e adaptados, especialmente em contextos médicos e de diagnóstico, onde diferenças de género podem influenciar na interpretação dos resultados.



(a) *Boxplot* da *feature ddpJitter*. (b) *Boxplot* da *feature locPctJitter*.

Figura 4.5: *Boxplots* das primeiras 2 *Jitter features*.



(a) *Boxplot* da *feature locAbsJitter* (b) *Boxplot* da *feature ppq5Jitter*

Figura 4.6: *Boxplots* das terceira e quarta *Jitter features*.

4.3 Metodologias

No tratamento dos dados, foram adotadas várias abordagens para estudar o problema de diferentes perspetivas. Inicialmente, verificou-se a não existência de valores omissos e a não existência de variáveis categóricas, o que facilitou o processo de tratamento de dados, não sendo necessária a eliminação ou substituição dos valores omissos, bem como a realização de *One-Hot Encoding* ou *LabelEncoding* ao nível das variáveis categóricas.

Ainda neste sentido dividiu-se o problema em duas abordagens distintas: utilização de todo o conjunto de dados e divisão do conjunto de dados por género.

Apesar da divisão ou não existência de divisão de dados por género, a nível de metodologia esta é semelhante no intuito geral, isto é, na etapa de pré-processamento dos conjuntos de dados, de divisão de conjunto de dados de treino e de teste, modelação através de diversas técnicas de *Machine Learning*, otimização dos hiperparâmetros e avaliação da performances dos diversos modelos com as métricas, tal como se verifica na Figura 4.7.

4.3.1 Conjunto de dados completo

Nesta abordagem utiliza-se o conjunto de dados completo como ponto inicial, sendo que mesmo dentro destas abordagens existem diversos testes que são realizados.

Uma primeira abordagem, sendo uma *baseline* ao nível das métricas resultantes do modelo, ou seja, sem grandes etapas de processamento de dados (apenas se procedeu com a eliminação da variável "*id*", uma vez que apenas é representativa da identificação do registo), foi através da utilização do conjunto de dados fornecido completo, ou seja, uma conjunto de dados desbalanceado, em que se realizou a etapa de Normalização através do *MinMaxScaler*, não se realizaram algoritmos de seleção de features, sendo que se aplicaram 7 algoritmos distintos (*SVM*, RF, *NN*, *NB*, DT, XGBoost e *AdaBoost*), com validação cruzada e otimização dos hiperparâmetros.

Uma vez que o conjunto de dados original apresenta 755 *features* e caso todas estas *features* sejam utilizadas no processo de treino existe um aumento significativo no tempo de processamento, além de que pode resultar em problemas de *features* duplicadas, *features* efetivas, torna-se necessária a existência de diferentes técnicas de seleção de *features*, que

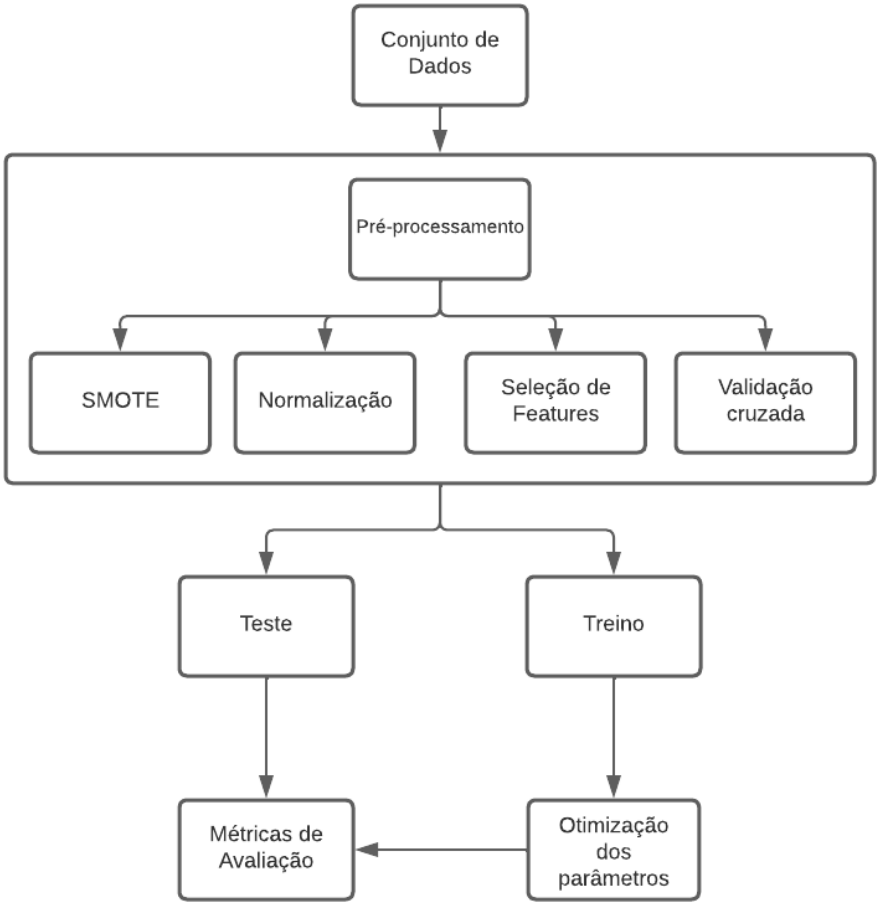


Figura 4.7: Metodologia aplicada.

serão explicadas nas próximas abordagens.

Numa segunda abordagem, com o conjunto de dados desbalanceado, procedeu-se com a etapa de normalização através do *MinMaxScaler* e ao nível de seleção de *features* utilizou-se a Correlação de *Pearson*, removendo-se assim todas as *features* com uma correlação superior a 0.8, resultando numa quantidade final de 262 *features*. Nesta abordagem testaram-se os mesmos 7 algoritmos com validação cruzada e otimização dos hiperparâmetros.

Numa terceira abordagem, ainda com o conjunto de dados desbalanceado, procedeu-se com a etapa de Normalização através do *MinMaxScaler* e ao nível de seleção de *features* utilizou-se a correlação de *Pearson* e o algoritmo de *PCA*, removendo-se assim todas as *features* com uma correlação superior a 0.8, resultando numa quantidade final de 262 *features*, que através do *PCA* reduziu-se para 111 *features*. Nesta abordagem testaram-se os mesmos 7 algoritmos com Validação Cruzada e Otimização dos Hiperparâmetros.

A quarta abordagem já apresenta um *dataset* balanceado através da utilização do algoritmo de *SMOTE*, resultando num conjunto de dados com 1128 registos, sendo metade destes registos referentes a indivíduos sem a doença de *Parkinson* e outra metade referente a indivíduos com a doença de *Parkinson*. A importância deste balanceamento de dados é diretamente proporcional com a proporção de dados entre as duas diferentes *labels* da

classe está presenta na Tabela 4.4.

Tabela 4.4: Comparação entre conjunto de dados desbalanceado e conjunto de dados balanceado

Dataset	**Número de casos de PD**	**Número de casos sem PD**	**Total**
Original	564	192	756
SMOTE	564	564	1128

A este conjunto de dados balanceado aplicou-se uma etapa de normalização através do *MinMaxScaler*, não se realizaram algoritmos de seleção de *features*, sendo que se aplicaram 7 algoritmos distintos (*SVM*, RF, *NN*, *NB*, DT, XGBoost e *AdaBoost*), com validação cruzada e otimização dos hiperparâmetros.

A quinta abordagem apresenta um *dataset* balanceado através da utilização do algoritmo de *SMOTE*, resultando num conjunto de dados com 1128 registos, sendo metade destes registos referentes a indivíduos sem a doença de *Parkinson* e outra metade referente a indivíduos com a doença de *Parkinson*. A este conjunto de dados balanceado aplicou-se uma etapa de normalização através do *MinMaxScaler* e uma redução das *features* através da eliminação das *features* com coeficiente de *Pearson* superior a 0.8, sendo que se aplicaram 7 algoritmos distintos (*SVM*, RF, *NN*, *NB*, DT, XGBoost e *AdaBoost*), com validação cruzada e otimização dos hiperparâmetros.

A sexta abordagem apresenta um *dataset* balanceado através da utilização do algoritmo de *SMOTE*, resultando num conjunto de dados com 1128 registos, sendo metade destes registos referentes a indivíduos sem a Doença de *Parkinson* e outra metade referente a indivíduos com a doença de *Parkinson*. A este conjunto de dados balanceado aplicou-se uma etapa de normalização através do *MinMaxScaler* e uma redução das *features* através da eliminação das *features* com coeficiente de *Pearson* superior a 0.8, resultando numa quantidade final de 262 *features*, que através do *PCA* reduziu-se para 111 *features*. Por fim, aplicaram-se 7 algoritmos distintos (*SVM*, RF, *NN*, *NB*, DT, XGBoost e *AdaBoost*), com validação cruzada e otimização dos hiperparâmetros.

A Sétima, oitava, nova, décima, décima primeira e décima segunda abordagem compreendem o estudo individual de cada um dos conjuntos de *Features*, *baseline Features, time frequency features, vocal fold features, MFCCs features, wavelet features* e *TQWT features*, respetivamente. Inicialmente procedeu-se à divisão do conjunto de dados entre as diferentes conjuntos de *features*, sendo que este já se apresentava balanceado com a utilização do *SMOTE* e normalizado.

A décima terceira abordagem diferenciou-se pela utilização da *feature importance* do algoritmo *XGBoost* para a seleção de *features*, após o balanceamento e normalização do conjunto de dados, de onde resultou um conjunto de dados final com 44 *features*. Posteriormente, utilizou-se um *stacking ensemble* como classificador, com os algoritmos *RF*, *Gradient boosting* (GB), *SVM* e *NN* como classificadores base e dois diferentes algoritmos como classificador meta (*LR* e *XGBoost*).

Na décima quarta abordagem também se compreendeu a uma divisão do conjunto de dados nos diferentes conjuntos de *Features*, *baseline features, time frequency features, vocal fold features, MFCCs features, wavelet Features* e *TQWT features*, respetivamente.

Uma vez feita esta divisão, treinou-se individualmente cada um dos conjuntos de *features* com o algoritmo *RF* e as suas previsões serviram de *input* para um algoritmo de *soft voting ensemble*.

Na décima quinta abordagem, procedeu-se a um balanceamento do conjunto de dados e a uma normalização antes da seleção de *features* com o algoritmo de *XGBoost*, o que resultou num conjunto de dados com 44 *features*. Por fim, aplicaram-se 7 algoritmos distintos (*SVM*, RF, *NN*, *NB*, DT, XGBoost e *AdaBoost*), com validação cruzada e otimização dos hiperparâmetros.

Na décima sexta abordagem, procedeu-se a um balanceamento do conjunto de dados e a uma normalização antes da seleção de *features* com a utilização do *VIF*, o que resultou num conjunto de dados com 125 *features*. Posteriormente, utilizou-se um *stacking ensemble* como classificador, com os algoritmos *NN*, GB, *SVM* e *XGBoost* como classificadores base e o algoritmo de *LR* como classificador meta.

Na décima sétima abordagem, procedeu-se a um balanceamento do conjunto de dados e a uma normalização antes da seleção de *features* com a utilização do *VIF*, o que resultou num conjunto de dados com 125 *features*, ao qual foi aplicado posteriormente um *PCA*, resultando num total de 60 *features*. Posteriormente, utilizou-se um *stacking ensemble* como Classificador, com os algoritmos *NN*, GB, *SVM* e *XGBoost* como classificadores base e o algoritmo de *LR* como classificador meta.

Na décima oitava abordagem, procedeu-se a um balanceamento do conjunto de dados e a uma normalização antes da seleção de *features* com a utilização do *VIF*, o que resultou num conjunto de dados com 125 *features*, das quais se selecionou 96 *features* finais através da *feature importance* do *XGBoost*. Posteriormente, utilizou-se um *stacking ensemble* como classificador, com os algoritmos *NN*, GB, *SVM* e *XGBoost* como classificadores base e o algoritmo de *LR* como classificador meta.

A fim de realizar uma otimização do desempenho dos classificadores na tarefa de potenciar o diagnóstico prévio da *DP*, foi realizado um processo de validação cruzada, no sentido de encontrar as melhores combinações de parâmetros para cada algoritmo.

A tabela 4.5 resume os resultados dessa validação cruzada, destacando-se os diferentes parâmetros considerados para cada classificador, juntamente com os valores testados para cada parâmetro. Esta abordagem sistemática permite uma exploração abrangente do espaço de hiperparâmetros e facilita a identificação das configurações mais promissoras para cada algoritmo. Ao ajustar adequadamente os parâmetros do classificador, esperamos melhorar significativamente sua capacidade de generalização e, consequentemente, sua precisão na classificação de pacientes com *DP*.

A validação cruzada é uma técnica fundamental no desenvolvimento de modelos de *ML*. Esta técnica é usada para o aumento de generalização de um modelo num conjunto de dados não conhecido previamente. Isso é crucial para garantir que o modelo seja capaz de fazer previsões precisas em dados que não foram usados durante o treino.

No contexto do nosso trabalho, implementamos a validação cruzada utilizando a função `GridSearchCV` da biblioteca *scikit-learn*. Esta função executa um processo de busca dos melhores hiperparâmetros para cada modelo de classificação. Durante esse processo, a validação cruzada é usada para avaliar o desempenho do modelo em várias divisões dos dados de treino.

Tabela 4.5: Parâmetros do classificador

Classificador	**Parâmetro**	**Valores**
SVM	C	[0.1, 1, 10]
	kernel	[linear, rbf, poly]
	gamma	[scale, auto]
RF	n_estimators	[100, 200, 300]
	criterion	[gini, entropy]
	max_depth	[None, 5, 10]
KNN	n_neighbors	[3, 5, 7]
	weights	[uniform, distance]
	p	[1, 2]
NB	priors	[None, [0.1, 0.9], [0.3, 0.7], [0.5, 0.5], [0.7, 0.3], [0.9, 0.1]]
DT	criterion	[gini, entropy]
	max_depth	[None, 5, 10]
XGBoost	learning_rate	[0.1, 0.01]
	max_depth	[3, 5, 7]
	n_estimators	[100, 200, 300]
AdaBoost	n_estimators	[50, 100, 200]
	learning_rate	[0.1, 1, 10]

O procedimento de validação cruzada funciona da seguinte maneira:

1. **Divisão dos dados:** Os dados de treino são divididos em várias partes chamadas *folds*. Por padrão, a função `GridSearchCV` utiliza a estratégia de validação cruzada *k-Fold*, onde os dados são divididos em k partes iguais. Cada parte é usada como conjunto de avaliação uma vez, enquanto as outras k-1 partes são usadas para treino.

2. **Treino e avaliação:** Para cada combinação de hiperparâmetros no espaço de procura definido, o modelo é treinado em cada uma das k-1 partições e avaliado na partição de validação. Isso é repetido k vezes, até que cada partição tenha sido usada como conjunto de validação.

3. **Média das métricas:** Ao final do processo de validação cruzada, as métricas de desempenho são calculadas para cada combinação dos hiperparâmetros. Por último, essas métricas são agregadas, faz-se o cálculo da média, para fornecer uma estimativa geral do desempenho do modelo.

4. **Escolha dos melhores hiperparâmetros:** Finalmente, o modelo com os melhores hiperparâmetros, ou seja, aqueles que produzem o melhor desempenho médio durante a validação cruzada (*accuracy*, *F1-score*, precisão e AUC), é selecionado como o modelo final.

4.3.2 Conjunto de dados dividido por género

Numa outra abordagem, dividiu-se o conjunto de dados pelo seu género, procedendo-se ao seu treino e teste, de forma separada, sendo que a nível de etapas e algoritmos utilizados no pré-processamento de dados, procedeu-se com os mesmos algoritmos utilizados na conjunto de dados completo. A nível de objetivos pretendeu-se avaliar a possível influência ao nível de resultados quando estes estão agrupados pelo mesmo género.

4.4 Análise dos resultados

A análise dos resultados será feita em duas partes: primeiro, com o conjunto de dados completo e, depois, com o conjunto de dados dividido por género.

4.4.1 Conjunto de dados completo

No Conjunto de dados completo, foram estudadas as abordagens presentes na seguinte lista:

- Conjunto de dados desbalanceado, com *MinMaxScaler* com todas as *features*, 5 *fold cross-validation* e *GridSearch*
- Conjunto de dados desbalanceado, com *MinMaxScaler* com remoção de variáveis com correlação superior a 0.8 (262 *features*), 5 *fold cross-validation* e *GridSearch*
- Conjunto de dados desbalanceado, com *MinMaxScaler* com remoção de variáveis com correlação superior a 0.8 e *PCA*, *5 fold cross-validation* e *GridSearch*
- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*
- Conjunto de dados balanceado com SMOTE, remoção de variáveis com correlação superior a 0.8 (263), *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*
- Conjunto de dados balanceado com SMOTE, remoção de variáveis com correlação superior a 0.8 e *PCA*, *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*
- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*. Utilização de apenas as *baseline features*
- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*. Utilização de apenas as *Time Frequency features*
- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*. Utilização de apenas as *Vocal Fold features*
- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*. Utilização de apenas as *MFCCs features*
- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*,*5 fold cross-validation* e *GridSearch*.Utilização de apenas as *Wavelet features*
- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*. Utilização de apenas as *TQWT features*

- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*, feature selection com XGBoost, *5 fold cross-validation* e *GridSearch*. *RF*, *GB*, *SVM* e *NN* como Classificadores Base. *Ensemble Stacking*: *XGBoost* como Classificador Meta.
- Conjunto de dados balanceado com SMOTE e *MinMaxScaler*. *RF* aplicado em cada conjunto de *features* e as previsões são utilizadas no *Voting Ensemble*.
- Conjunto de dados balanceado com SMOTE, *MinMaxScaler*, *feature selection* com *XGBoost*, *5 fold cross-validation* e *GridSearch*.
- Conjunto de dados balanceado com SMOTE e *MinMaxScaler*, redução de Features através da utilização do índice de *VIF* para 125 features, *5 fold cross-validation* e *GridSearch*. NN, *GB*, *SVM* e *XGBoost* como Classificadores Base.
- Conjunto de dados balanceado com SMOTE e *MinMaxScaler*, redução de *Features* através da utilização do índice de *VIF* e *PCA*, *5 fold cross-validation* e *GridSearch*. *NN*, *GB*, *SVM* e *XGBoost* como Classificadores Base e *LR* como Classificador Meta.
- Conjunto de dados balanceado com SMOTE e *MinMaxScaler*, redução de Features através da utilização do índice de *VIF* e *XGBoost Feature Importance* para 96 features, *5 fold cross-validation* e *GridSearch*. *RF*, *GB*, *SVM* e NN como Classificadores Base e XGBoost como Classificador Meta

Os modelos foram avaliados utilizando validação cruzada *k-fold*. Para a Abordagem I, II e III incluíram-se as métricas de desempenho para cada *fold* no Apêndice B. Estes resultados indicam que os modelos são capazes de generalizar os resultados com conjuntos de dados de teste não conhecidos pelo treino, mantendo um desempenho consistente ao longo dos diferentes *folds*, evitando assim fenómenos de *overfitting*.

Na Tabela 4.6 apresentam-se os resultados obtidos em estudo com diversas características de processamento e normalização dos dados, no caso em que se considera todo o conjunto de dados, isto é, não existindo uma diferenciação entre género.

Tabela 4.6: Análise de resultados com o conjunto de dados completo

Estudos					
Abordagem	**Algoritmos**	**AUC**	**F1**	**ACC**	**Tempo (s)**
I	SVM	0.772	0.926	0.882	10.082
	RF	0.759	0.922	0.875	161.787
	KNN	0.860	0.930	0.895	2.313
	NB	0.741	0.862	0.796	0.595
	DT	0.796	0.864	0.796	6.714
	XGBoost	0.785	0.920	0.875	532.757
	AdaBoost	0.838	0.936	0.901	192.909
II	SVM	0.741	0.905	0.849	4.335
	RF	0.701	0.903	0.842	103.169
	KNN	0.895	0.966	0.947	1.098
	NB	0.732	0.828	0.757	0.233
	DT	0.754	0.854	0.789	1.778
	XGBoost	0.750	0.914	0.862	133.619

Abordagem	Algoritmos	AUC	F1	ACC	Tempo (s)
	AdaBoost	0.763	0.917	0.868	39.323
III	SVM	0.855	0.944	0.914	1.385
	RF	0.614	0.883	0.803	39.256
	KNN	0.886	0.957	0.934	0.519
	NB	0.644	0.864	0.782	0.154
	DT	0.662	0.828	0.743	1.030
	XGBoost	0.750	0.914	0.862	68.663
	AdaBoost	0.692	0.894	0.829	22.221
IV	SVM	0.944	0.942	0.942	29.770
	RF	0.947	0.947	0.946	230.486
	KNN	0.936	0.932	0.934	15.969
	NB	0.792	0.798	0.792	2.279
	DT	0.886	0.885	0.885	16.509
	XGBoost	0.961	0.961	0.960	671.184
	AdaBoost	0.947	0.948	0.947	275.578
V	SVM	0.935	0.933	0.934	0.505
	RF	0.930	0.930	0.929	85.232
	KNN	0.944	0.941	0.942	1.476
	NB	0.813	0.794	0.810	0.258
	DT	0.881	0.882	0.880	2.689
	XGBoost	0.939	0.938	0.938	152.633
	AdaBoost	0.907	0.909	0.907	75.791
VI	SVM	0.952	0.952	0.951	3.269
	RF	0.908	0.909	0.907	67.916
	KNN	0.910	0.901	0.907	0.935
	NB	0.787	0.796	0.788	0.170
	DT	0.816	0.809	0.814	1.487
	XGBoost	0.904	0.901	0.903	78.830
	AdaBoost	0.882	0.877	0.880	35.929
VII	SVM	0.816	0.809	0.814	2.922
	RF	0.827	0.842	0.845	53.365
	KNN	0.791	0.769	0.788	0.684
	NB	0.630	0.555	0.624	0.110
	DT	0.767	0.762	0.765	0.447
	XGBoost	0.869	0.865	0.867	23.198
	AdaBoost	0.841	0.845	0.841	16.644
VIII	SVM	0.808	0.796	0.805	2.981
	RF	0.875	0.864	0.872	47.341
	KNN	0.827	0.806	0.823	0.568
	NB	0.742	0.760	0.743	0.127
	DT	0.786	0.772	0.783	0.319
	XGBoost	0.871	0.858	0.867	15.379
	AdaBoost	0.773	0.857	0.770	12.209
	SVM	0.776	0.767	0.774	3.070

Abordagem	Algoritmos	AUC	F1	ACC	Tempo (s)
IX	RF	0.877	0.877	0.876	53.097
	KNN	0.832	0.810	0.827	0.633
	NB	0.665	0.699	0.668	0.126
	DT	0.719	0.719	0.717	0.478
	XGBoost	0.904	0.902	0.903	25.688
	AdaBoost	0.740	0.740	0.739	18.485
X	SVM	0.939	0.938	0.938	3.546
	RF	0.895	0.893	0.894	81.300
	KNN	0.833	0.802	0.827	0.965
	NB	0.729	0.659	0.721	0.165
	DT	0.820	0.818	0.818	1.461
	XGBoost	0.887	0.883	0.885	55.515
	AdaBoost	0.889	0.888	0.889	30.739
XI	SVM	0.659	0.672	0.659	5.464
	RF	0.842	0.841	0.841	101.497
	KNN	0.789	0.749	0.783	1.496
	NB	0.651	0.695	0.655	0.278
	DT	0.771	0.771	0.770	2.956
	XGBoost	0.828	0.830	0.827	132.535
	AdaBoost	0.783	0.788	0.783	74.969
XII	SVM	0.659	0.672	0.659	5.464
	RF	0.842	0.841	0.841	101.597
	KNN	0.780	0.749	0.783	1.496
	NB	0.651	0.695	0.655	0.278
	DT	0.771	0.772	0.770	2.956
	XGBoost	0.828	0.830	0.827	132.535
	AdaBoost	0.783	0.788	0.783	74.969
XIII	Essemble Stacking: LR como Classificador Meta	0.983	0.983	0.982	70.497
	Essemble Stacking: XGBoost como Classificador Meta	0.987	0.987	0.987	71.723
XIV	Voting Ensemble	0.909	0.906	0.907	0.854
XV	SVM	0.948	0.947	0.947	3.431
	RF	0.948	0.947	0.947	60.663
	KNN	0.961	0.960	0.960	0.799
	NB	0.767	0.789	0.769	0.139
	DT	0.868	0.866	0.867	0.908
	XGBoost	0.969	0.969	0.969	35.416
	AdaBoost	0.917	0.916	0.916	26.767

Abordagem	Algoritmos	AUC	F1	ACC	Tempo (s)
XVI	*Ensemble Stacking*: *LR* como Classificador Meta	0.978	0.979	0.978	203.329
XVII	*Ensemble Stacking*: *LR* como Classificador Meta	0.973	0.974	0.973	92.953
XVIII	*Ensemble Stacking*: *XGBoost* como Classificador Meta	0.982	0.982	0.982	139.426

Ao comparar várias abordagens de classificação usando métricas como *AUC*, F1-score, *Accuracy (ACC)* e tempo de execução, destacam-se diferentes padrões de desempenho e eficiência computacional.

Melhores desempenhos:

- *AUC*: A abordagem VI liderou com o melhor desempenho médio de *AUC* (0.952), seguida por IV (0.948) e V (0.926).
- F1-score: Novamente, a abordagem VI teve o melhor desempenho médio de F1-score (0.952), seguida por V (0.924) e IV (0.947).
- *ACC*: Aqui, a abordagem IV mostrou o melhor desempenho médio de *ACC* (0.947), seguida por V (0.925) e VI (0.951).
- Tempo de execução: A abordagem V foi a mais eficiente com o menor tempo médio de execução (1.476 segundos), seguida por III (1.030 segundos) e IX (0.633 segundos).
- A abordagem III geralmente apresentou resultados inferiores em *AUC*, F1-score e ACC, enquanto a abordagem VII mostrou consistentemente desempenho mais baixo em todas as métricas.

Impactos das diferentes abordagens:

- *SVM* vs Ensemble (*RF*, *AdaBoost*, *XGBoost*): Métodos como *SVM* demonstraram desempenho competitivo em *AUC* e *ACC*, embora com tempos de execução mais longos. Enquanto isso, ensembles como *RF*, *AdaBoost* e *XGBoost* alcançaram bons resultados variando entre desempenho e tempo de execução, com *XGBoost* sendo mais lento, mas robusto em métricas como *AUC* e F1-score.
- *NN* e *NB*: *NN* foi eficaz com bom desempenho em *AUC* e *ACC* e tempos de execução baixos. *NB* teve execução extremamente rápida, mas desempenho moderado a baixo em comparação com outras abordagens.
- *DT*: Exibiram desempenho intermediário, com resultados variados em todas as métricas e tempos de execução relativamente rápidos.

Considerações finais:

- A escolha da abordagem ideal depende das necessidades específicas do problema, por exemplo, máximo desempenho em *AUC* ou equilíbrio entre desempenho e tempo de execução. Para conjuntos de dados onde desempenho é crítico e recursos permitem, métodos como *XGBoost* ou *SVM* podem ser preferíveis, enquanto *NN* pode ser uma escolha eficiente para equilibrar desempenho e tempo.
- No que diz respeito a nível de melhores métricas, as abordagem que utilizam métodos *Ensemble* (XIII, XVI, XVIII, e XVIII) apresentam as melhores métricas, contudo também apresentam mais tempo de execução.

4.4.2 Conjunto de dados dividido por género

No Conjunto de dados dividido por género, foram estudadas as abordagens presentes na seguinte lista:

- Conjunto de dados desbalanceado, com *MinMaxScaler* com todas as *features*, 5 *fold cross-validation* e *GridSearch*
- Conjunto de dados desbalanceado, com *MinMaxScaler* com remoção de variáveis com correlação superior a 0.8 (262 *features*), 5 *fold cross-validation* e *GridSearch*
- Conjunto de dados desbalanceado, com *MinMaxScaler* com remoção de variáveis com correlação superior a 0.8 e PCA, *5 fold cross-validation* e *GridSearch*
- Conjunto de dados balanceado com *SMOTE*, *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*
- Conjunto de dados balanceado com *SMOTE*, remoção de variáveis com correlação superior a 0.8 (263), *MinMaxScaler*, *5 fold cross-validation* e *GridSearch*

Na Tabela 4.7 apresentam-se os resultados obtidos em estudo com diversas características de processamento e normalização dos dados, no caso em que se considera o conjunto de dados divido por género

Tabela 4.7: Análise de resultados com o conjunto de dividido por género

Estudos						
Abordagem	**Sexo**	**Algoritmos**	**AUC**	**F1**	**ACC**	**Tempo (s)**
I	F	SVM	0.760	0.848	0.797	0.797
		RF	0.835	0.898	0.865	29.732
		KNN	0.903	0.923	0.905	0.442
		NB	0.766	0.765	0.743	0.174
		DT	0.796	0.866	0.824	0.678
		XGBoost	0.867	0.903	0.878	86.713
		AdaBoost	0.845	0.879	0.851	19.291
I	M	SVM	0.803	0.898	0.846	0.811
		RF	0.611	0.896	0.821	51.068
		KNN	0.800	0.918	0.872	0.876
		NB	0.678	0.739	0.654	0.334
		DT	0.783	0.900	0.846	1.442
		XGBoost	0.852	0.952	0.923	107.075

Abordagem	Sexo	Algoritmos	AUC	F1	ACC	Tempo (s)
		AdaBoost	0.761	0.920	0.872	39.355
		SVM	0.900	0.938	0.919	2.043
		RF	0.750	0.868	0.811	27.205
		KNN	0.896	0.925	0.905	0.343
II	F	NB	0.823	0.815	0.797	0.151
		DT	0.698	0.750	0.703	0.360
		XGBoost	0.796	0.866	0.824	46.485
		AdaBoost	0.741	0.804	0.757	20.567
		SVM	0.892	0.950	0.923	1.049
		RF	0.555	0.882	0.795	56.976
		KNN	0.744	0.903	0.846	0.730
II	M	NB	0.650	0.892	0.820	0.306
		DT	0.558	0.790	0.679	0.821
		XGBoost	0.625	0.866	0.782	68.971
		AdaBoost	0.717	0.896	0.833	25.996
		SVM	0.878	0.870	0.878	4.926
		RF	0.931	0.925	0.929	104.924
		KNN	0.961	0.957	0.959	3.242
III	F	NB	0.790	0.762	0.796	0.816
		DT	0.856	0.844	0.857	5.410
		XGBoost	0.921	0.915	0.918	491.822
		AdaBoost	0.869	0.860	0.767	145.878
		SVM	0.962	0.960	0.961	3.818
		RF	0.962	0.960	0.961	77.242
		KNN	0.931	0.926	0.930	1.805
III	M	NB	0.759	0.783	0.760	0.551
		DT	0.852	0.859	0.853	4.914
		XGBoost	0.954	0.952	0.953	359.854
		AdaBoost	0.962	0.960	0.961	143.641
		SVM	0.900	0.891	0.898	2.405
		RF	0.880	0.872	0.878	73.609
		KNN	0.878	0.861	0.888	1.147
IV	F	NB	0.741	0.713	0.745	0.380
		DT	0.818	0.809	0.816	1.645
		XGBoost	0.899	0.891	0.898	200.352
		AdaBoost	0.867	0.847	0.867	55.476
		SVM	0.946	0.944	0.946	4.150
		RF	0.961	0.961	0.961	101.975
		KNN	0.953	0.952	0.953	1.729
IV	M	NB	0.768	0.737	0.767	0.440
		DT	0.869	0.864	0.868	2.690
		XGBoost	0.961	0.961	0.961	208.071
		AdaBoost	0.946	0.946	0.946	58.888
		SVM	0.899	0.891	0.898	2.168

Abordagem	Sexo	Algoritmos	AUC	F1	ACC	Tempo (s)
		RF	0.880	0.872	0.878	73.901
		KNN	0.878	0.861	0.888	1.253
V	F	NB	0.741	0.713	0.745	0.400
		NB	0.818	0.809	0.816	1.629
		XGBoost	0.890	0.891	0.898	255.769
		AdaBoost	0.867	0.857	0.867	54.520
		SVM	0.946	0.944	0.946	3.777
		RF	0.961	0.961	0.961	102.780
		KNN	0.954	0.952	0.953	1.663
V	M	NB	0.768	0.737	0.767	0.471
		NB	0.866	0.864	0.868	2.701
		XGBoost	0.961	0.961	0.961	187.435
		AdaBoost	0.946	0.946	0.946	36.650

No conjunto de dados do sexo masculino, os resultados mostraram que todos os modelos tiveram desempenho consistente em conjuntos de dados masculinos. O *NN*, *XGBoost* e *RF* apresentaram altos valores de *AUC*, F1 e ACC, indicando uma capacidade robusta de distinguir entre as classes de interesse. Especificamente, o *NN* destacou-se pela simplicidade e eficácia em capturar padrões nos dados masculinos.

Em contraste, observou-se uma variação nos resultados dos modelos quando aplicados a conjuntos de dados femininos. O *NN* manteve um desempenho relativamente estável, embora tenha mostrado uma ligeira queda nas métricas de avaliação em comparação com os dados masculinos. Isso sugere que, embora ainda eficaz, o *NN* pode não ser tão robusto ao lidar com as nuances específicas dos dados femininos.

O *XGBoost* e *RF*, por outro lado, exibiram uma queda mais acentuada no desempenho ao lidar com dados femininos. Isso pode ser atribuído à complexidade desses modelos em capturar padrões sensíveis dos conjuntos de dados femininos a características não totalmente capturadas pelos algoritmos.

O *NB* mostrou a maior disparidade de desempenho entre conjuntos de dados masculinos e femininos, com métricas notavelmente mais baixas nos dados femininos. Isso indica uma limitação significativa do modelo *NB* em generalizar para diferentes distribuições de dados demográficos, possivelmente devido à sua suposição simplificada de independência entre variáveis.

Esta análise destacou a importância de considerar o género ao desenvolver modelos de *ML*. As diferenças no desempenho entre conjuntos de dados masculinos e femininos ressaltam que as características demográficas podem influenciar significativamente a capacidade do modelo de generalizar e capturar padrões relevantes.

Modelos que mostraram consistência em ambos os conjuntos de dados podem ser mais robustos e generalizáveis em contextos mais amplos, enquanto aqueles que demonstraram variação significativa podem exigir ajustes adicionais ou abordagens personalizadas para cada grupo demográfico

Capítulo 5

Interface gráfica

Na última fase do projeto, desenvolveu-se uma aplicação *Web*, uma *GUI* com o intuito de demonstrar a utilização dos diversos algoritmos em prática. Esta interface foi construída através de *Flask*, *HTML* e *CSS*, tendo como principal objetivo a utilização de modelos de *ML* treinados, ou seja, facilitando a utilização destes modelos, permitindo fazer uma previsão de resultados baseados num conjunto de dados de entrada, sem a necessidade de utilização de código.

A arquitetura da interface de *ML* que utiliza *Flask*, *HTML*, *CSS* e importa os modelos via *pickle* é projetada para fornecer uma experiência amigável e intuitiva aos utilizadores enquanto aproveita a eficácia dos modelos treinados. *Flask* atua como o servidor *web*, onde se gere as solicitações dos clientes e se faz a comunicação entre a interface do utilizador e os modelos de *ML*. O *HTML* e o *CSS* são empregados para criar uma interface visual atraente e responsiva, que permite aos utilizadores interagir facilmente com a aplicação.

A integração dos modelos de *ML* via pickle simplifica o processo de implementação, permitindo que os modelos treinados sejam facilmente carregados e utilizados pelo *Flask*. Com o uso do formato *pickle*, os modelos são armazenados em arquivos, garantindo que eles possam ser facilmente importados e utilizados pela aplicação *web*. A Figura 5.1 representa a arquitetura da aplicação.

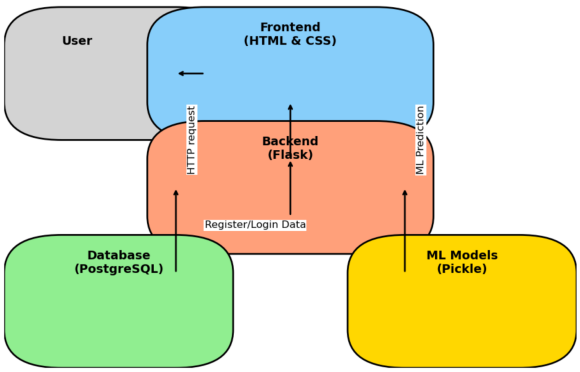


Figura 5.1: Arquitetura da Aplicação.

Essa arquitetura modular e flexível permite que a aplicação *web* aproveite os poderosos recursos de *ML* de forma eficiente, ao mesmo tempo em que fornece uma experiência de utilizador suave e amigável. A importação de modelos via *pickle* simplifica o processo de

desenvolvimento e manutenção da aplicação, garantindo sua escalabilidade e adaptabilidade.

5.1 Tipos de estudos

A aplicação pode ser utilizada de duas formas distintas: através da gravação de voz ou através da Importação de dados via *CSV*.

5.1.1 Gravação de voz

- Os utilizadores podem gravar sua voz usando um dispositivo de entrada de áudio, como microfone embutido no computador ou um dispositivo externo.
- A gravação de voz é então processada e enviada para o servidor *Flask* para análise.
- O servidor utiliza o modelo de *ML* previamente treinado para fazer previsões com base na entrada de voz.
- Os resultados da previsão são exibidos na interface do utilizador, juntamente com qualquer *feedback* adicional relevante.

5.1.2 Importação de CSV

- Os utilizadores podem fazer upload de um arquivo *CSV* contendo os dados de entrada para previsão.
- O servidor *Flask* processa o arquivo *CSV* e utiliza o modelo de *ML* para fazer previsões com base nos dados fornecidos.
- Os resultados da previsão são exibidos na interface do utilizador, permitindo que os utilizadores visualizem e interpretem facilmente os resultados.

É importante notar que a parte de gravação por voz não é precisa, nem possível de comparação neste sentido, quando comparada com a importação de dados via *CSV*, uma vez que se utiliza o microfone do PC e não há garantia de que as *features* calculadas sejam equiparadas aos cálculos das *features* num ambiente real e controlado. No entanto, o objetivo é demonstrar a viabilidade de um protótipo desse tipo, que pode ser aprimorado para produzir resultados mais precisos no futuro.

5.2 Requisitos

5.2.1 Requisitos funcionais

Os requisitos funcionais são aqueles que descrevem as funcionalidades específicas que a aplicação deve oferecer para atender às necessidades dos utilizadores. No contexto de uma interface para *ML*, é fundamental fornecer ferramentas e recursos que facilitem a interação dos utilizadores com os modelos de *ML*, permitindo a execução de tarefas como teste de modelos, avaliação de desempenho e inferência de dados.

Os requisitos funcionais apresentados para a aplicação são os seguintes:

- **Avaliação de desempenho:** Oferecer métricas de avaliação de desempenho, como precisão, *recall* e *F1-score*, para que os utilizadores possam analisar e comparar o desempenho dos modelos treinados.
- **Visualização de dados:** Possibilitar a visualização de dados de treino e resultados de inferência por meio de gráficos, tabelas ou outras representações visuais intuitivas.
- **Inferência de dados:** Permitir que os utilizadores realizem inferência em novos dados usando os modelos treinados, fornecendo previsões ou classificações para os mesmos.
- **Gestão de modelos:** Permitir aos utilizadores guardar, carregar e gerir modelos treinados, disponibilizando a reutilização dos mesmos.

Esses requisitos funcionais são essenciais para garantir que a nossa interface para *ML* seja robusta, intuitiva e capaz de atender às diversas necessidades dos utilizadores, desde o treino inicial de modelos até a análise avançada de resultados e *insights*.

5.2.2 Requisitos não funcionais

Ao projetar e desenvolver uma aplicação, além dos requisitos funcionais, é relevante a consideração dos requisitos não funcionais da mesma, ou seja, requisitos que permitam uma melhor experiência ao utilizador, tais como a usabilidade, disponibilidade, desempenho e facilidade, bem como a compatibilidade entre diversos sistemas.

De seguida, detalha-se os principais requisitos não funcionais que orientam o desenvolvimento e a implementação da presente aplicação:

- **Usabilidade:** A aplicação deverá ser de uso fácil, com *endpoints* claramente definidos, permitindo aos utilizadores navegar de forma intuitiva e realizar suas tarefas de maneira eficiente.
- **Disponibilidade:** A aplicação estará disponível durante os 7 dias da semana, 24 horas por dia, garantindo acesso contínuo e ininterrupto aos utilizadores, independentemente do horário ou localização.
- **Desempenho:** O tempo de resposta da aplicação deve ser minimizado, garantindo uma experiência ágil e responsiva que não comprometa a eficácia na realização de tarefas ou a satisfação do utilizador.
- **Compatibilidade:** O sistema deve ser suportado por todos os *browsers* modernos, assegurando uma experiência consistente e sem falhas, independentemente do navegador utilizado pelos utilizadores.

Esses requisitos não funcionais desempenham um papel crítico na definição da qualidade e do sucesso da nossa aplicação, garantindo que ela não apenas atenda às necessidades funcionais dos utilizadores, mas também ofereça uma experiência robusta, acessível e satisfatória em todos os aspetos.

5.3 Menus

A nível da aplicação, uma vez efetuado o registo e posterior *login*, apresenta então a sua página inicial, onde é apresentada informação geral acerca da *DP*, bem como informação

relativa aos restantes menus existentes:

- *Real time*: Onde é possível fazer uma previsão em tempo real, através da gravação da própria voz.
- *Data analysis*: Menu que se dividi em dois sub-menus (Geral e *Distribution*) onde são apresentadas característica gerais e características sobre a distribuição dos dados do *dataset*.
- *All dataset*: Menu que se divide em três sub-menus (*Normalized*, *Normalized and Correlated* e *PCA*) onde é possível, através da importação de um ficheiro *CSV*, realizar a previsão da *DP*, através da utilização de vários modelos.
- *All Dataset with SMOTE*: Menu que se dividi em três sub-menus (*Normalized*, *Normalized and Correlated* e *PCA*) onde é possível, através da importação de um ficheiro *CSV*, realizar a previsão da *DP*, através da utilização de vários modelos.
- *XGBoost Feature Selection*: Menu onde é possível, através da importação de um Ficheiro *CSV*, realizar a previsão da *DP*, através da utilização de vários modelos.

5.3.1 Registo e *login*

Nesta página, os utilizadores têm a opção de criar uma conta na aplicação ou iniciar sessão caso já possuam uma conta. O registo permite o acesso a funcionalidades personalizadas e conteúdos exclusivos, ou seja, apenas se pode visualizar o resto da aplicação caso tenho o *login* efetuado. A Figura 5.2 representa a página de registo e *login* da aplicação.

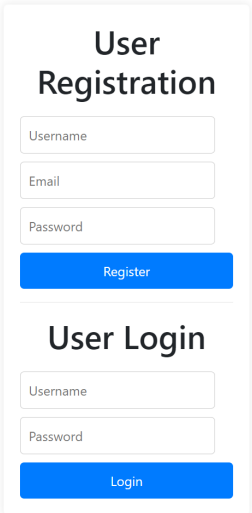


Figura 5.2: Registo e *login* na Aplicação.

5.3.2 Página inicial

A página inicial da aplicação oferece uma visão geral das funcionalidades disponíveis, uma descrição da *DP*, bem como acesso a informações relevantes e opções de navegação entre

os diferentes menus existentes na aplicação. A Figura 5.3 representa a página inicial da aplicação.

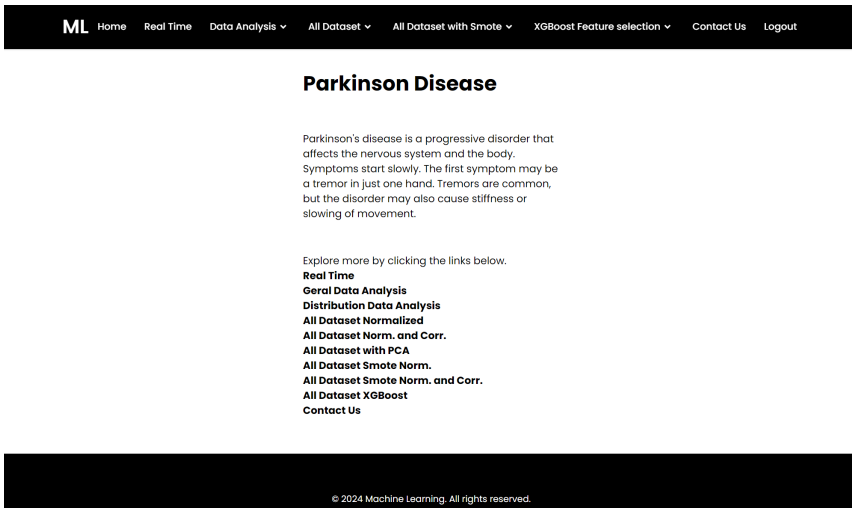


Figura 5.3: Página inicial da Aplicação.

5.3.3 *Real time*

Neste menu, os utilizadores podem realizar previsões em tempo real, isto é, podem realizar a classificação em tempo real, utilizando os dados e recursos disponíveis na aplicação para tomar decisões informadas. Desta forma, o utilizador pode gravar a sua voz durante 3 segundos, depois pode realizar a visualização do áudio gravado, bem como realizar a classificação do áudio gravado.

No final da previsão aparece um texto com a indicação da presença ou ausência da doença de *Parkinson*. A Figura 5.4 representa a página de previsão em tempo real.

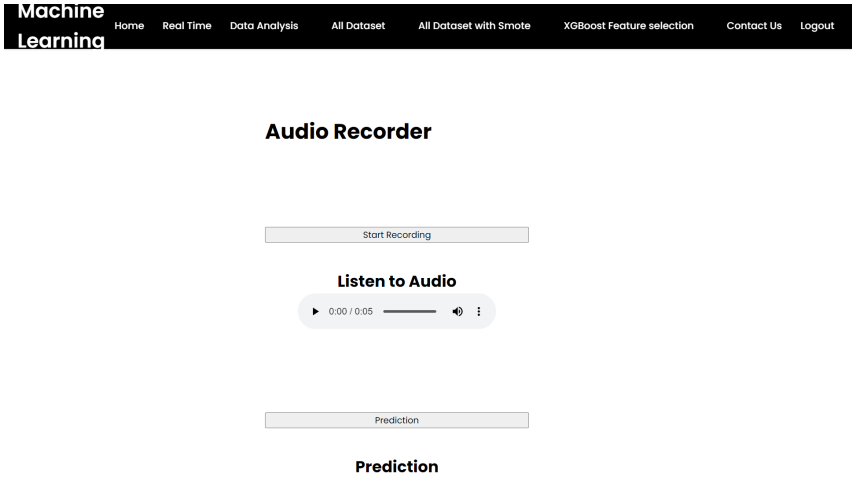


Figura 5.4: Previsão em tempo real da aplicação.

5.3.4 Menus de previsão por importação CSV

Esta página permite aos utilizadores revisar e realizar previsões com base em dados importados através de um ficheiro *CSV*. Essa funcionalidade facilita a análise de conjuntos

de dados externos, uma vez que o conjunto de dados de treino é de elevada dimensionalidade, necessitando de um ficheiro com bastantes *features* para teste, sendo mais fácil a sua importação por *CSV* do que o seu preenchimento através de um formulário.

Este menu serve como base para a maioria dos menus existentes na aplicação, uma vez que o que difere entre os menus são os modelos utilizados para a realização da previsão. A Figura 5.5 representa a página de previsão por importação *CSV* da aplicação.

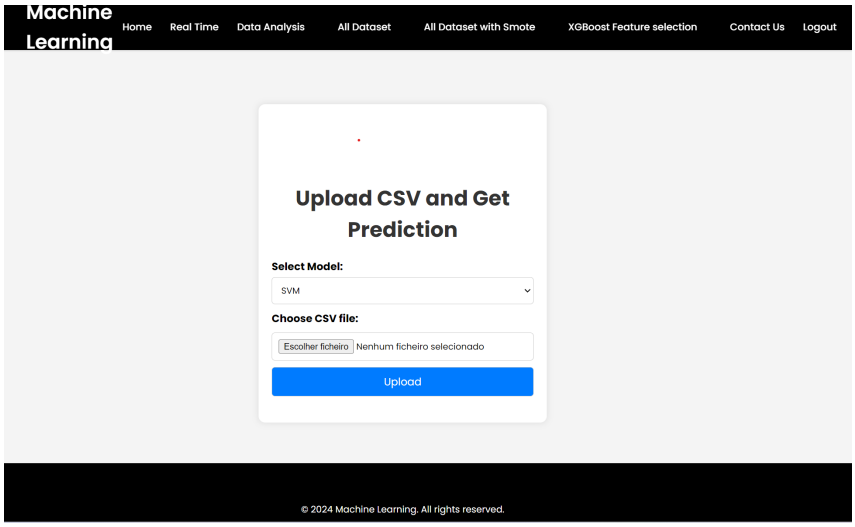


Figura 5.5: Previsão por importação CSV da aplicação.

Capítulo 6

Conclusão e trabalho futuro

6.1 Conclusão

A deteção prévia da doença de *Parkinson* assume uma importância crucial no contexto de tratamento da segunda doença neurodegenerativa mais frequente, uma vez que esta possibilita uma intervenção mais prévia e a elaboração de métodos de tratamento personalizado, sendo assim, por sua vez mais eficazes. Esta maior eficácia deve-se à possibilidade de elaboração de terapias individuais que retardam de forma eficaz a progressão dos sintomas da doença, possibilitando desta forma uma melhor qualidade de vida aos utentes, o mais perto possível da condição normal.

Neste sentido, torna-se essencial investir em métodos que permitam a deteção precoce da doença de *Parkinson*, pois isso possibilita uma intervenção médica mais personalizada e eficaz. Nesse contexto, a utilização de algoritmos de *Machine Learning* tem sido amplamente estudada em diversos conjuntos de dados de utentes, incluindo dados da marcha, caligrafia, imagens médicas e registos de voz. No presente trabalho, foram utilizados registos de voz, que são característicos da área da saúde. Este conjunto de dados é desbalanceado, com uma quantidade limitada de registos, apresentando ainda uma predominância da classe referente à existência da doença e um número elevado de *features*.

Tendo em conta que o conjunto de dados apresenta as características previamente referidas, foi necessária a elaboração de uma método que permitisse a realização de um excelente tratamento de dados, através do balanceamento, normalização e redução da dimensionalidade do conjunto de dados, onde se utilizaram algoritmo como o *SMOTE*, Normalização, *PCA* e diversos algoritmos de seleção de *Features*.

Para verificar como a deteção prévia pode ser afetada pelo género, estudou-se o problema em duas vertentes, sendo que na primeira vertente estudou-se com o conjunto de dados completo, e numa segunda vertente analisou-se através de uma divisão do conjunto de dados por género. Ainda dentro desta divisão, fez-se um estudo exaustivo a diferentes pré-processamentos e a diferente algoritmos, bem como com a utilização de algoritmo de *Ensemble*, dos quais tiram partido de diversos algoritmos mais fracos, para desenvolvimento de um modelo final mais completo e eficiente.

A nível de resultados, verificou-se que as etapas de balanceamento de dados, bem como a normalização dos dados e a seleção de *features* promoveu uma melhor eficiência ao

nível dos algoritmos, bem como permitiu uma redução no tempo para a execução dos algoritmos. Neste estudo destacam-se os resultados de 3 métodos:

- *Ensemble stacking* com *LR* como classificador meta: Conjunto de dados balanceado com *SMOTE* e *MinMaxScaler*, redução de *features* através da utilização do índice de *VIF* para 125 *features*, 5 *fold cross-validation* e *GridSearch*. *NN*, *GB*, *SVM* e *XGBoost* como classificadores base
- *Ensemble stacking* com *LR* como classificador meta: Conjunto de dados balanceado com *SMOTE* e *MinMaxScaler*, redução de *features* através da utilização do índice de *VIF* e *PCA*, 5 *fold cross-validation* e *GridSearch*. *NN*, *GB*, *SVM* e *XGBoost* como classificadores base
- *Ensemble stacking* com *LR* como classificador meta: Conjunto de dados balanceado com *SMOTE* e *MinMaxScaler*, redução de *features* através da utilização do índice de *VIF* e *XGBoost feature importance* para 96 *features*, 5 *fold cross-validation* e *GridSearch*. *NN*, *GB*, *SVM* e *XGBoost* como classificadores base

Os 3 métodos obtiveram resultados bastante satisfatórios, apresentando *AUC*, *F1* e *ACC* perto dos 98%, sendo que o último método apresentou os resultados superiores, sendo 98.7% a percentagem presentes nas 3 métricas.

Contudo, tal como se verificou previamente, o conjunto de dados é um conjunto de dados característico da área da saúde, apresentando limitações, tais como o conjunto de dados apenas apresentar 754 registos, e também apresentar um conjunto de dados desbalanceado.

Por outro lado, o conjunto de dados é complexo, sendo necessário o conhecimento de eletrónica para o seu entendimento, nomeadamente ao nível do processamento de sinal, para conhecimento da transformada utilizada.

6.2 Trabalho futuro

O presente estudo proporcionou *insights* valiosos sobre a deteção precoce da doença de *Parkinson*, destacando a eficácia de abordagens baseadas em algoritmos de *Machine Learning* e pré-processamento de dados. No entanto, há várias direções que podem ser exploradas no futuro para aprimorar ainda mais a pesquisa e suas aplicações práticas:

6.2.1 Aumento do conjunto de dados

Embora os resultados tenham sido promissores com o conjunto de dados disponível, a inclusão de mais dados pode melhorar a robustez e a generalização dos modelos desenvolvidos. Recolhas de dados abrangentes, envolvendo múltiplos centros médicos e diferentes populações, ajudariam a capturar uma variedade maior de *features* da doença e dos pacientes.

6.2.2 Exploração de outras fontes de dados

Além dos registos de voz, explorar outras fontes de dados como dados de imagem cerebral, dados genéticos ou dados de sensores *wearable* pode enriquecer a análise e proporcionar uma compreensão mais abrangente da doença de *Parkinson*. A integração de múltiplas

modalidades de dados pode revelar padrões e correlações ocultas, melhorando a precisão da deteção prévia.

6.2.3 Desenvolvimento de modelos de interpretação

Embora os modelos de *Machine Learning* tenham demonstrado alta performance, muitas vezes estes não são de fácil compreensão, especialmente em contextos clínicos onde a transparência é crucial. Portanto, o desenvolvimento de modelos interpretáveis, como árvores de decisão ou modelos baseados em regras, pode facilitar a compreensão dos fatores que contribuem para a deteção da doença de *Parkinson*.

6.2.4 Validação clínica

Uma etapa fundamental para a aplicação prática desses modelos é a validação clínica em larga escala. Colaborações com profissionais de saúde e instituições médicas podem ajudar a avaliar a eficácia dos modelos em ambientes do mundo real, determinar sua utilidade clínica e identificar possíveis desafios de implementação.

6.2.5 Integração em sistemas de saúde

Uma vez validados, os modelos de deteção precoce da doença de *Parkinson* podem ser integrados em sistemas de saúde existentes para auxiliar no diagnóstico e triagem de pacientes. Isso pode melhorar o acesso ao tratamento precoce e personalizado, proporcionando melhores resultados clínicos e qualidade de vida para os pacientes.

6.2.6 Investigação contínua

Dada a natureza complexa e dinâmica da doença de *Parkinson*, a pesquisa contínua é essencial para acompanhar os avanços na área médica e de ciência de dados. Explorar novas técnicas de modelagem, adaptar-se às mudanças na disponibilidade de dados e atualizar constantemente os modelos são componentes-chave para o sucesso contínuo na deteção precoce e tratamento da doença.

Em resumo, o futuro da deteção precoce da doença de *Parkinson* está repleto de oportunidades emocionantes para avanços significativos. Com um compromisso contínuo com a inovação e a colaboração entre diversas disciplinas, podemos trabalhar em direção a abordagens mais eficazes e acessíveis para enfrentar essa importante questão de saúde pública.

Bibliografia

[1] Nirajan Acharya. Why is it called logistic regression and not logistic classification? https://medium.com/@nirajan.acharya777/why-is-it-called-logistic-regression-and-not-logistic-classification-c201889d268c, 2023. Accessed: 2024-06-11.

[2] Fang Zhang, Zhen Zhang, and Hui Xiao. Research on medical big data analysis and disease prediction method based on artificial intelligence. *Computational and Mathematical Methods in Medicine*, 2022:1–10, 09 2022.

[3] Fei Jiang, Yong Jiang, Hui Zhi, Yi Dong, Hao Li, Sufeng Ma, Yilong Wang, Qiang Dong, Haipeng Shen, and Yongjun Wang. Artificial intelligence in healthcare: past, present and future. *Stroke and Vascular Neurology*, 2(4):230–243, 2017.

[4] L. D. Jones, D. Golan, S. A. Hanna, and M. Ramachandran. Artificial intelligence, machine learning and the evolution of healthcare. *Bone & Joint Research*, 7(3):223–225, 2018.

[5] J. J. Ferreira, N. Gonçalves, A. Valadas, C. Januário, M. R. Silva, L. Nogueira, J. L. M. Vieira, and A. B. Lima. Prevalence of parkinson's disease: a population-based study in portugal. *European Journal of Neurology*, 24(5):748–750, 2017.

[6] A. L. Doença de parkinson afeta entre 18 a 20 mil pessoas em portugal: Spn - sociedade portuguesa de neurologia, 2022.

[7] Verónica Cabreira and João Massano. Parkinson's disease: Clinical review and update. *Acta Médica Portuguesa*, 32(10):661–670, Oct. 2019.

[8] Silvia Del Din, Alan Godfrey, Brook Galna, Sue Lord, and Lynn Rochester. Free-living gait characteristics in ageing and parkinson's disease: Impact of environment and ambulatory bout length. *Journal of NeuroEngineering and Rehabilitation*, 13, 2016.

[9] Günther Deuschl, Carmen Schade-Brittinger, Paul Krack, Jens Volkmann, Helmut Schäfer, Kai Bötzel, Christine Daniels, Angela Deutschländer, Ulrich Dillmann, Wilhelm Eisner, Doreen Gruber, Wolfgang Hamel, Jan Herzog, Rüdiger Hilker, Stephan Klebe, Manja Kloß, Jan Koy, Martin Krause, Andreas Kupsch, Delia Lorenz, Stefan Lorenzl, H. Maximilian Mehdorn, Jean Richard Moringlane, Wolfgang Oertel, Marcus O. Pinsker, Heinz Reichmann, Alexander Reuß, Gerd-Helge Schneider, Alfons Schnitzler, Ulrich Steude, Volker Sturm, Lars Timmermann, Volker Tronnier, Thomas Trottenberg, Lars Wojtecki, Elisabeth Wolf, Werner Poewe, and Jürgen Voges. A randomized trial of deep-brain stimulation for parkinson's disease. *New England Journal of Medicine*, 355(9):896–908, 2006.

[10] Andres M. Lozano, Nir Lipsman, Hagai Bergman, Peter Brown, Stéphan Chabardès, Jin Woo Chang, Keith Matthews, Cameron C. McIntyre, Thomas E. Schlaepfer, Michael Schulder, Yasin Temel, Jens Volkmann, and Joachim K. Krauss. Deep brain stimulation: current challenges and future directions. *Nature Reviews Neurology*, 15:148–160, 2019.

[11] Nicola Tambasco, Michele Romoli, and Paolo Calabresi. Levodopa in parkinson's disease: Current status and future developments. *Current Neuropharmacology*, 16:1239 – 1252, 2017.

[12] Theo Vos et Al. Global, regional, and national incidence, prevalence, and years lived with disability for 310 diseases and injuries, 1990–2015: a systematic analysis for the global burden of disease study 2015. *The Lancet*, 388(10053):1545–1602, 2016.

[13] Aileen Ho, John Bradshaw, Robert Iansek, and Robin Alfredson. Volume regulation in parkinsonian speech. In *Proceedings of the 5th International Conference on Spoken Language Processing (ICSLP 1998)*, 11 1998.

[14] Ginanjar Abdurrahman and Mukti Sintawati. Implementation of xgboost for classification of parkinson's disease. *Journal of Physics: Conference Series*, 1538:012024, 05 2020.

[15] Gaurang Prasad, Thilanka Munasinghe, and Oshani Seneviratne. A two-step framework for parkinson's disease classification: Using multiple one-way anova on speech features and decision trees. In *Proceedings of the International Workshop on Artificial Intelligence for Health (AI4Health 2020)*, volume 2884, 2020.

[16] R. Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 01 2000.

[17] I. M.L. Donaldson. James parkinson's essay on the shaking palsy, 2015.

[18] André Parent. A tribute to james parkinson, 2018.

[19] Glenda E. Gillies, Ilse S. Pienaar, Shiv Vohra, and Zahi Qamhawi. Sex differences in parkinson's disease. *Frontiers in Neuroendocrinology*, 35(3):370–384, 2014. Sex Differences in Neurological and Psychiatric Disorders.

[20] Muhammad Shakeel Khan Imran Ahmed, Sultan Aljahdali and Sanaa Kaddoura. Classification of parkinson disease based on patient's voice signal using machine learning. *Intelligent Automation & Soft Computing*, 32(2):705–722, 2022.

[21] J Jankovic. Parkinson's disease: clinical features and diagnosis. *Journal of Neurology, Neurosurgery & Psychiatry*, 79(4):368–376, 2008.

[22] Christopher Kobylecki. Update on the diagnosis and management of parkinson's disease. *Clinical Medicine*, 20(4):393–398, 2020.

[23] Marwan Hariz and Patric Blomstedt. Deep brain stimulation for parkinson's disease. *Journal of Internal Medicine*, 292(5):764–778, 2022.

[24] Omar Barukab, Amir Ahmad, Tabrej Khan, and Mujeeb Rahiman Thayyil Kunhumuhammed. Analysis of parkinson's disease using an imbalanced-speech dataset by employing decision tree ensemble methods. *Diagnostics*, 12(12), 2022.

[25] Salama A. Mostafa, Aida Mustapha, Mazin Abed Mohammed, Raed Ibraheem Hamed, N. Arunkumar, Mohd Khanapi Abd Ghani, Mustafa Musa Jaber, and Shihab Hamad Khaleefah. Examining multiple feature evaluation and classification methods for improving the diagnosis of parkinson's disease. *Cognitive Systems Research*, 54:90–99, 2019.

[26] C. Okan Sakar, Gorkem Serbes, Aysegul Gunduz, Hunkar C. Tunc, Hatice Nizam, Betul Erdogdu Sakar, Melih Tutuncu, Tarkan Aydin, M. Erdem Isenkul, and Hulya Apaydin. A comparative analysis of speech signal processing algorithms for parkinson's disease classification and the use of the tunable q-factor wavelet transform. *Applied Soft Computing*, 74:255–263, 2019.

[27] Şule Yücelbaş. Simple logistic hybrid system based on greedy stepwise algorithm for feature analysis to diagnose parkinson's disease according to gender. *Arabian Journal for Science and Engineering*, 45, 2020.

[28] Turker Tuncer, Sengul Dogan, and Udyavara Rajendra Acharya. Automated detection of parkinson's disease using minimum average maximum tree and singular value decomposition method with vowels. *Biocybernetics and Biomedical Engineering*, 40, 2020.

[29] Iqra Nissar, Danish Raza Rizvi, Sarfaraz Masood, and Aqib Nazir Mir. Voice-based detection of parkinson's disease through ensemble machine learning approach: A performance study. *EAI Endorsed Transactions on Pervasive Health and Technology*, 5(19), 8 2019.

[30] Amira S. Ashour, Majid Kamal A. Nour, Kemal Polat, Yanhui Guo, Wafaa Alsaggaf, and Amira El-Attar. A novel framework of two successive feature selection levels using weight-based procedure for voice-loss detection in parkinson's disease. *IEEE Access*, 8:76193–76203, 2020.

[31] Ashena Gorgan Mohammadi, Pouya Mehralian, Amir Naseri, and Hedieh Sajedi. Parkinson's disease diagnosis: The effect of autoencoders on extracting features from vocal characteristics. *Array*, 11:100079, 2021.

[32] Hayder Mohammed Qasim, Oguz Ata, Mohammad Azam Ansari, Mohammad N. Alomary, Saad Alghamdi, and Mazen Almehmadi. Hybrid feature selection framework for the parkinson imbalanced dataset prediction problem. *Medicina*, 57(11), 2021.

[33] Mohammed Younis Thanoun and Mohammad T. Yaseen. A comparative study of parkinson disease diagnosis in machine learning. In *Proceedings of the 2020 International Conference on Software Engineering and Information Management (ICSIM 2020)*, 2020.

[34] Biswajit Karan. Speech-based parkinson's disease prediction using xgboost-based features selection and the stacked ensemble of classifiers. *Journal of The Institution of Engineers (India): Series B*, 104, 2023.

[35] Kemal Polat. A hybrid approach to parkinson disease classification using speech signal: The combination of smote and random forests. In *2019 Scientific Meeting on Electrical-Electronics and Biomedical Engineering and Computer Science, EBBT 2019*, 2019.

[36] Shai Shalev-Shwartz and Shai Ben-David. *Understanding Machine Learning - From Theory to Algorithms.* Cambridge University Press, 2014.

[37] Joel Grus. *Data Science from Scratch: First Principles with Python.* O'Reilly, Beijing, 2015.

[38] Iqbal H. Sarker. Machine learning: Algorithms, real-world applications and research directions. *SN Comput. Sci.*, 2(3), mar 2021.

[39] Padraig Cunningham, Matthieu Cord, and Sarah Delany. *Supervised Learning*, pages 21–49. Springer, 01 2008.

[40] A.C. Müller and S. Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists.* O'Reilly Media, 2016.

[41] Georg Langs, S. Röhrich, Johannes Hofmanninger, F. Prayer, J. Pan, C. Herold, and Helmut Prosch. Machine learning: from radiomics to discovery and routine. *Der Radiologe*, 58, 06 2018.

[42] Haldun Akoglu. User's guide to correlation coefficients, 2018.

[43] Laerd Statistics. Pearson product-moment correlation. `https://statistics.laerd.com/statistical-guides/pearson-correlation-coefficient-statistical-guide.php`, 2023. Accessed: November 24, 2023.

[44] Ian Jolliffe and Jorge Cadima. Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374:20150202, 04 2016.

[45] Doruk Canga. Automated feature selection for machine learning in python, 2024. Accessed: 2024-05-15.

[46] Shahadat Uddin, Arif Khan, Md Ekramul Hossain, and Mohammad Ali Moni. Comparing different supervised machine learning algorithms for disease prediction. *BMC Medical Informatics and Decision Making*, 19, 2019.

[47] DataCamp. Decision tree classification in python, 2024.

Apêndice A

Estatísticas descritivas das *baseline features*

A Tabela A.1 apresenta as estatísticas descritivas das *baseline features* do conjunto de dados presente no estudo efetuado. Como o conjunto de dados é altamente dimensional, isto é, apresenta um número elevado de *features*, apenas foram selecionadas estas no sentido de proceder com a demonstração da necessidade de normalização do conjunto de dados.

Estas estatísticas são calculadas sobre os dados antes da aplicação do *MinMaxScaler*, que é uma técnica de normalização frequentemente utilizada para colocar os dados para um intervalo de valores específico, isto é, entre 0 e 1. Esta prática é crucial para se garantir que todas as *features* contribuam de forma igual nos modelos de *Machine Learning* utilizados.

Por exemplo, *features* como e *meanHarmToNoiseHarmonicity* têm escalas muito distintas, quando comparadas com as restantes *features*, o que pode influenciar negativamente o desempenho de modelos que se baseiam em cálculos de distância ou regularização.

Tabela A.1: Estatísticas Descritivas das *Baseline Features*

Feature	**Mean**	**Std**	**Min**	**25%**	**50%**	**75%**	**Max**
PPE	0.746	0.169	0.042	0.763	0.810	0.834	0.908
DFA	0.700	0.070	0.544	0.647	0.701	0.755	0.853
RPDE	0.489	0.137	0.154	0.387	0.484	0.587	0.871
numPulses	323.97	99.22	2.00	251.00	317.00	384.25	907.00
numPeriodsPulses	322.68	99.40	1.00	250.00	316.00	383.25	905.00
meanPeriodPulses	0.006	0.002	0.002	0.005	0.006	0.008	0.013
stdDevPeriodPulses	0.0004	0.0007	0.00001	0.00005	0.00008	0.00017	0.003
locPctJitter	0.0023	0.0026	0.0002	0.0010	0.0015	0.0025	0.028
locAbsJitter	0.00002	0.00002	0.00000	0.00001	0.00001	0.00002	0.00026
rapJitter	0.0006	0.0010	0.00002	0.00015	0.00028	0.00065	0.011
ppq5Jitter	0.0012	0.0017	0.00005	0.00037	0.00065	0.00125	0.018
ddpJitter	0.0018	0.0029	0.00005	0.00045	0.00084	0.00195	0.033
locShimmer	0.0675	0.0430	0.0066	0.0361	0.0557	0.0855	0.251
locDbShimmer	0.606	0.383	0.057	0.319	0.503	0.797	2.114
apq3Shimmer	0.0344	0.0224	0.0034	0.0178	0.0280	0.0447	0.131
apq5Shimmer	0.0412	0.0272	0.0042	0.0219	0.0337	0.0530	0.200
apq11Shimmer	0.0554	0.0340	0.0004	0.0310	0.0475	0.0714	0.278
ddaShimmer	0.1031	0.0671	0.0100	0.0535	0.0839	0.1340	0.392
meanAutoCorrHarmonicity	0.962	0.064	0.590	0.963	0.984	0.993	0.999
meanNoiseToHarmHarmonicity	0.0511	0.1046	0.0006	0.0072	0.0163	0.0401	0.762
meanHarmToNoiseHarmonicity	18.860	5.576	1.655	15.713	19.310	22.878	33.197

Apêndice B

Métricas de cada *fold* da abordagem I, II e III com o *dataset* completo

As Tabelas B.1, B.2 e B.3 apresentam as métricas de desempenho (*accuracy*, precisão, *recall*, *F1-score* e AUC) para cada *fold* de *cross-validation* da abordagem I, abordagem II e abordagem III com o *dataset* completo, respetivamente. Utilizou-se *cross-validation* com *k-folds* (k = 5) com o intuito de verificar a capacidade dos modelos de generalização com dados não utilizados no treino, e desta forma evitar o fenómeno de *overfitting*.

Observando as métricas para cada modelo, pode-se concluir que:

- Os modelos apresentam *accuracy* consistente entre os *folds* de *cross-validation*, indicando que são capazes de classificar corretamente a maioria das instâncias nos conjuntos de dados de validação.
- A precisão, *recall* e *F1-score* também mostram pouca variação entre os diferentes *folds*, sugerindo que os modelos mantêm um bom equilíbrio entre a capacidade de identificar corretamente as classes positivas e minimizar os falsos positivos.
- A AUC mostra que os modelos têm boa capacidade de discriminação entre as classes, o que é crucial para problemas de classificação binária.

Essas observações indicam que os modelos são robustos e capazes de generalizar bem para novos conjuntos de dados, evitando o fenómeno de *overfitting*. Neste sentido, e como se apresenta várias abordagens, os resultados apresentados nas Tabelas B.1, B.2 e B.3, na abordagem I, II e III com o *dataset* completo são suficientes para demonstrar que os modelos são capazes de generalização dos resultados e confiáveis para as restantes abordagens, tanto como o conjunto de dados completo, como com o conjunto de dados dividido por género.

Tabela B.1: Métricas de cada *fold* de cada modelo na abordagem I

Modelo	**Métricas de cada *fold***				
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
SVM	0.876	0.864	0.989	0.922	0.769
	0.876	0.879	0.967	0.921	0.790
	0.876	0.871	0.978	0.921	0.779
	0.926	0.918	0.989	0.952	0.865
	0.833	0.837	0.967	0.897	0.700
Mean	0.877	0.874	0.978	0.923	0.781
Std	0.029	0.026	0.010	0.017	0.053
Random Forest	0.884	0.880	0.978	0.926	0.795
	0.851	0.846	0.978	0.907	0.731
	0.835	0.850	0.944	0.895	0.730
	0.893	0.874	1.000	0.933	0.790
	0.850	0.853	0.967	0.906	0.733
Mean	0.863	0.861	0.973	0.913	0.756
Std	0.022	0.014	0.018	0.014	0.030
KNN	0.909	0.916	0.967	0.941	0.854
	0.884	0.913	0.933	0.923	0.838
	0.884	0.913	0.933	0.923	0.838
	0.884	0.896	0.956	0.925	0.816
	0.892	0.881	0.989	0.932	0.794
Mean	0.891	0.904	0.956	0.929	0.828
Std	0.010	0.013	0.021	0.007	0.021
Naive Bayes	0.818	0.895	0.856	0.875	0.783
	0.802	0.884	0.844	0.864	0.761
	0.777	0.871	0.822	0.846	0.734
	0.835	0.898	0.878	0.888	0.794
	0.817	0.862	0.900	0.880	0.733
Mean	0.810	0.882	0.860	0.870	0.761
Std	0.019	0.014	0.027	0.015	0.025
Decision Tree	0.793	0.849	0.878	0.863	0.713
	0.818	0.878	0.878	0.878	0.761
	0.826	0.888	0.878	0.883	0.778
	0.793	0.857	0.867	0.862	0.724
	0.808	0.894	0.844	0.869	0.772
Mean	0.808	0.873	0.869	0.871	0.750
Std	0.013	0.017	0.013	0.008	0.026
XGBoost	0.851	0.860	0.956	0.905	0.752
	0.868	0.863	0.978	0.917	0.763
	0.893	0.897	0.967	0.930	0.822
	0.909	0.899	0.989	0.942	0.833
	0.858	0.854	0.978	0.912	0.739
Mean	0.876	0.875	0.973	0.921	0.782
Std	0.022	0.019	0.011	0.013	0.038

Modelo	Métricas de cada *fold*				
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
AdaBoost	0.835	0.865	0.922	0.892	0.751
	0.843	0.859	0.944	0.899	0.746
	0.901	0.915	0.956	0.935	0.849
	0.909	0.907	0.978	0.941	0.844
	0.858	0.876	0.944	0.909	0.772
Mean	0.869	0.884	0.949	0.915	0.793
Std	0.030	0.023	0.018	0.019	0.045

Tabela B.2: Métricas de cada *fold* de cada modelo na abordagem II

Modelo	Métricas de cada *fold*				
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
SVM	0.860	0.854	0.978	0.912	0.747
	0.893	0.881	0.989	0.932	0.801
	0.835	0.843	0.956	0.896	0.720
	0.893	0.914	0.944	0.929	0.843
	0.800	0.817	0.944	0.876	0.656
Mean	0.856	0.862	0.962	0.909	0.753
Std	0.035	0.033	0.018	0.021	0.065
Random Forest	0.868	0.856	0.989	0.918	0.753
	0.826	0.835	0.956	0.891	0.704
	0.810	0.825	0.944	0.881	0.682
	0.893	0.874	1.000	0.933	0.790
	0.833	0.837	0.967	0.897	0.700
Mean	0.846	0.845	0.971	0.904	0.726
Std	0.030	0.017	0.021	0.019	0.040
KNN	0.884	0.913	0.933	0.923	0.838
	0.893	0.889	0.978	0.931	0.811
	0.851	0.875	0.933	0.903	0.773
	0.909	0.899	0.989	0.942	0.833
	0.867	0.870	0.967	0.916	0.767
Mean	0.881	0.889	0.960	0.923	0.804
Std	0.020	0.016	0.023	0.013	0.030
Naive Bayes	0.818	0.915	0.833	0.872	0.804
	0.719	0.859	0.744	0.798	0.695
	0.752	0.841	0.822	0.831	0.685
	0.810	0.924	0.811	0.864	0.809
	0.750	0.875	0.778	0.824	0.722
Mean	0.770	0.883	0.798	0.838	0.743
Std	0.038	0.032	0.033	0.027	0.053

Modelo	Métricas de cada *fold*				
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
Decision Tree	0.719	0.833	0.778	0.805	0.663
	0.810	0.853	0.900	0.876	0.724
	0.769	0.844	0.844	0.844	0.696
	0.777	0.846	0.856	0.851	0.702
	0.833	0.865	0.922	0.892	0.744
Mean	0.782	0.848	0.860	0.854	0.706
Std	0.039	0.010	0.050	0.030	0.027
XGBoost	0.884	0.896	0.956	0.926	0.842
	0.860	0.861	0.967	0.911	0.828
	0.851	0.860	0.956	0.904	0.778
	0.884	0.880	0.978	0.929	0.850
	0.850	0.853	0.967	0.906	0.767
Mean	0.866	0.870	0.965	0.915	0.813
Std	0.016	0.018	0.008	0.011	0.035

Tabela B.3: Métricas de cada *fold* de cada modelo na abordagem III

Modelo	Métricas de cada *fold*				
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
SVM	0.917	0.944	0.944	0.944	0.892
	0.901	0.906	0.967	0.935	0.838
	0.876	0.887	0.956	0.920	0.800
	0.959	0.947	1.000	0.973	0.919
	0.875	0.887	0.956	0.920	0.794
Mean	0.906	0.914	0.964	0.938	0.849
Std	0.031	0.027	0.019	0.020	0.049
Random Forest	0.802	0.789	1.000	0.882	0.613
	0.835	0.818	1.000	0.900	0.677
	0.810	0.802	0.989	0.886	0.640
	0.826	0.811	1.000	0.896	0.661
	0.800	0.789	1.000	0.882	0.600
Mean	0.815	0.802	0.998	0.889	0.638
Std	0.014	0.011	0.004	0.007	0.029
KNN	0.893	0.914	0.944	0.929	0.843
	0.934	0.936	0.978	0.957	0.892
	0.851	0.900	0.900	0.900	0.805
	0.851	0.867	0.944	0.904	0.763
	0.850	0.860	0.956	0.905	0.744
Mean	0.876	0.895	0.944	0.919	0.809
Std	0.033	0.029	0.025	0.021	0.054

Modelo	Métricas de cada *fold*				
	Accuracy	**Precision**	**Recall**	**F1 Score**	**AUC**
Naive Bayes	0.793	0.828	0.911	0.868	0.681
	0.802	0.837	0.911	0.872	0.697
	0.793	0.822	0.922	0.869	0.671
	0.793	0.842	0.889	0.865	0.703
	0.725	0.794	0.856	0.824	0.594
Mean	0.781	0.825	0.898	0.860	0.669
Std	0.028	0.017	0.024	0.018	0.039
Decision Tree	0.719	0.811	0.811	0.811	0.631
	0.744	0.847	0.800	0.823	0.690
	0.760	0.835	0.844	0.840	0.680
	0.744	0.864	0.778	0.819	0.711
	0.733	0.802	0.856	0.828	0.611
Mean	0.740	0.832	0.818	0.824	0.665
Std	0.014	0.023	0.029	0.010	0.038
XGBoost	0.851	0.846	0.978	0.907	0.731
	0.835	0.850	0.944	0.895	0.730
	0.826	0.835	0.956	0.891	0.704
	0.901	0.890	0.989	0.937	0.817
	0.825	0.835	0.956	0.891	0.694
Mean	0.848	0.851	0.964	0.904	0.735
Std	0.028	0.020	0.016	0.017	0.043
AdaBoost	0.802	0.811	0.956	0.878	0.655
	0.826	0.829	0.967	0.892	0.693
	0.818	0.847	0.922	0.883	0.719
	0.868	0.863	0.978	0.917	0.763
	0.800	0.811	0.956	0.878	0.644
Mean	0.823	0.832	0.956	0.889	0.695
Std	0.025	0.020	0.019	0.015	0.043