Universidade do Minho
Escola de Engenharia

Maria Inês da Cunha Ferreira

Utilização de técnicas de Machine Learning na classificação da doença de Parkinson

Dezembro de 2022

Utilização de técnicas de Machine Learning na classificação da doença de Parkinson
Maria Inês da Cunha Ferreira
UMinho | 2022

Universidade do Minho
Escola de Engenharia

Maria Inês da Cunha Ferreira

Utilização de técnicas de Machine Learning na classificação da doença de Parkinson

Dissertação de Mestrado

Mestrado em Engenharia de Sistemas

Trabalho realizado sob a orientação de

Professor Doutor Orlando Manuel de Oliveira Belo

Dezembro de 2022

AGRADECIMENTOS

Em primeiro lugar, quero agradecer ao Professor Orlando Belo, por todo o apoio que me transmitiu durante a dissertação e pela sua paciência.

Quero também agradecer à minha família, em especial aos meus pais e às minhas irmãs, pelo apoio incondicional e coragem que sempre me transmitiram.

Aos meus amigos por todos os momentos vividos enquanto comunidade académica.

Por fim, a todas as pessoas que cruzaram o meu caminho e que de alguma forma o tornaram especial.

A todos, muito obrigada!

DECLARAÇÃO DE INTEGRIDADE

Declaro ter atuado com integridade na elaboração do presente trabalho académico e confirmo que não recorri à prática de plágio nem a qualquer forma de utilização indevida ou falsificação de informações ou resultados em nenhuma das etapas conducente à sua elaboração.

Mais declaro que conheço e que respeitei o Código de Conduta Ética da Universidade do Minho.

Utilização de técnicas de Machine Learning na classificação da doença de Parkinson

RESUMO

A Parkinson é uma doença neurodegenerativa, progressiva que afeta principalmente o sistema nervoso central, afetando milhões de pessoas em todo o Mundo. Surge habitualmente em indivíduos acima de 65 anos e comumente diagnosticada através da apresentação dos seus sintomas motores. Segundo a Organização Mundial de Saúde (WHO), os números de óbitos devido à doença de Parkinson têm aumentado exponencialmente quando comparado com as outras doenças neurológicas. Algumas estimativas globais afirmam que, em 2019, esta doença sofreu um aumento de 81% desde 2000, causando cerca de 329000 óbitos. Deste modo, a implementação de mecanismos que sejam capazes de realizar um diagnóstico desta doença é importante para desenvolvimento da sociedade, e consequentemente para salvar vidas. Neste trabalho de dissertação implementaram-se três algoritmos de *Machine Learning*, com o objetivo de descobrir, através do seu desempenho, o quão eficientes são na deteção desta doença. Para a concretização da classificação proposta, numa primeira etapa foram escolhidos o conjunto de dados referentes à voz e os algoritmos que iriam ser aplicados neste contexto. Os resultados obtidos demonstram que o conjunto de dados apresenta precisões muito elevadas, como aconteceu com o algoritmo *Logistic Regression*, que demonstrou uma "perfeita" classificação do número de casos. No entanto, com aplicação da técnica *SMOTE* a precisão dos modelos baixou, mas, mesmo assim, apresentou um conjunto de resultado bastante confiável. Em ambas as abordagens o modelo que apresentou um melhor desempenho foi o baseado em *Support Vector Machines.* O sistema desenvolvido é capaz de classificar o número de casos com e sem doença, mas necessita da implementação híbrida de técnicas de seleção de características na primeira abordagem.

Palavras-Chave: *Data Mining, Logistic Regression, Machine Learning, Parkinson, Random Forest, Support Vector Machines.*

Applying Machine Learning techniques for Parkinson disease classification

ABSTRACT

Parkinson's is a progressive neurodegenerative disease that primarily affects the central nervous system, affecting millions of people worldwide. It usually appears in individuals over the age of 65 and is commonly diagnosed through the presentation of its motor symptoms. According to the World Health Organization (WHO), the number of deaths due to Parkinson's disease has increased exponentially when compared to other neurological diseases. Some global estimates say that by 2019 this disease will have increased by 81% since 2000, causing 329000 deaths. Thus, the implementation of mechanisms that are able to perform a diagnosis of this disease is important for the development of society, and consequently to save lives. In this dissertation work, three Machine Learning algorithms were implemented with the objective of discovering, through their performance, how efficient they are in detecting this disease.

To perform the proposed classification, in a first stage the dataset referring to the voice and the algorithms that would be applied in this context were chosen. The results obtained show that the data set presents very high accuracy, as is the case of the Logistic Regression algorithm, which demonstrated a "perfect" classification of the number of cases. However, with application of the SMOTE technique the accuracy of the models dropped, but still showed a very reliable set of results. In both approaches the model that performed best was Support Vector Machines.

The developed system is able to classify the number of cases with and without disease but needs the hybrid implementation of feature selection techniques as a first approach.

Keywords: *Data Mining, Logistic Regression, Machine Learning, Parkinson, Random Forest, Support Vector Machines*

ÍNDICE

LISTA DE FIGURAS

LISTA DE TABELAS

1. INTRODUÇÃO

1.1 Contextualização

Nos últimos anos, houve um aumento do número de investigações em temas como o *Big Data*, *Business Analytics*, ou *Internet of things.* Várias organizações geram volumes de dados cada vez maiores, tendo os seus sistemas de armazenamento evoluído para sistemas de armazenamento maiores. Deste modo, as mais variadas indústrias do setor público e privado gerem, armazenam e analisam um grande volume de dados com o objetivo de melhorar os serviços que prestam aos clientes. Por conseguinte, os sistemas de informação têm demonstrado às indústrias uma nova forma de acrescentar valor ao seu negócio.

Particularmente na indústria da saúde, surgiu a necessidade de compreender e implementar software que ajude em contexto médico a definir um melhor diagnóstico e tratamento de doenças, tendo surgido a informática médica para estudar questões que aliem a computação à saúde. Na prática, a informática médica é um campo multidisciplinar que estuda a informação de dados biomédicos, tendo evoluído positivamente nos últimos anos. A sua aplicação estende-se a muitas aplicações na área da saúde, tais como: aplicações médicas para uso no quotidiano do ser humano, sistemas hospitalares, seguradoras de saúde ou desenvolvimento de dispositivos médico, entre muitas outras.

Uma das principais razões para esta evolução corresponde à integração de algoritmos de *Machine Learning* em sistemas de apoio à decisão, com o objetivo de prever doenças, de forma a auxiliar a equipa médica na deteção do diagnóstico numa fase precoce da doença. A extração dos dados para aplicação de algoritmos de *Machine Learning* pode ser proveniente de diversas fontes, das quais fazem parte, as bases de dados online e os equipamentos mais sofisticados, como por exemplo, sensores e dispositivos móveis. Sendo que na maioria dos casos, os dados extraídos apresentam uma baixa qualidade devido a valores em falta, valores duplicados, desequilíbrio do número de registos e formato das variáveis. Desta forma, o resultado de um algoritmo de *Machine Learning* depende diretamente da qualidade do conjunto de dados utilizado. Essa qualidade é melhorada através de técnicas de pré-processamento, onde o conjunto de dados sofre diversas transformações de limpeza até se verificar que se pode avançar para a aplicação do algoritmo.

As doenças neurológicas são uma preocupação crescente na sociedade. De acordo com um relatório da Organização Mundial de Saúde (World Health Organization, 2022) as doenças neurológicas,

tais como a epilepsia, o Alzheimer ou o Parkinson afetam cerca de 1 bilião de pessoas em todo o mundo, estimando-se que 6,8 milhões de pessoas morrem todos os anos em consequência dessas perturbações neurológicas (Siuly & Zhang, 2016).

Quanto à doença de Parkinson, a qual é alvo de estudo neste trabalho, é uma doença progressiva que, por enquanto, não tem uma cura, e que não é possível prever a sua progressão. Os indivíduos portadores desta doença, num estado terminal, apresentam uma baixa qualidade de vida. O tratamento desta doença varia de indivíduo para indivíduo, sendo definido de forma personalizada e gerido de diferentes formas. No entanto, existe uma abordagem de tratamento muito comum que é a administração do medicamento *Levedopa* com o objetivo de atrasar a progressão dos sintomas motores desta doença.

De acordo com o PubMed (PubMed, 2022) (Figura 1), entre o período de 2012 e 2022 é notório um interesse crescente dos investigadores nesta doença.

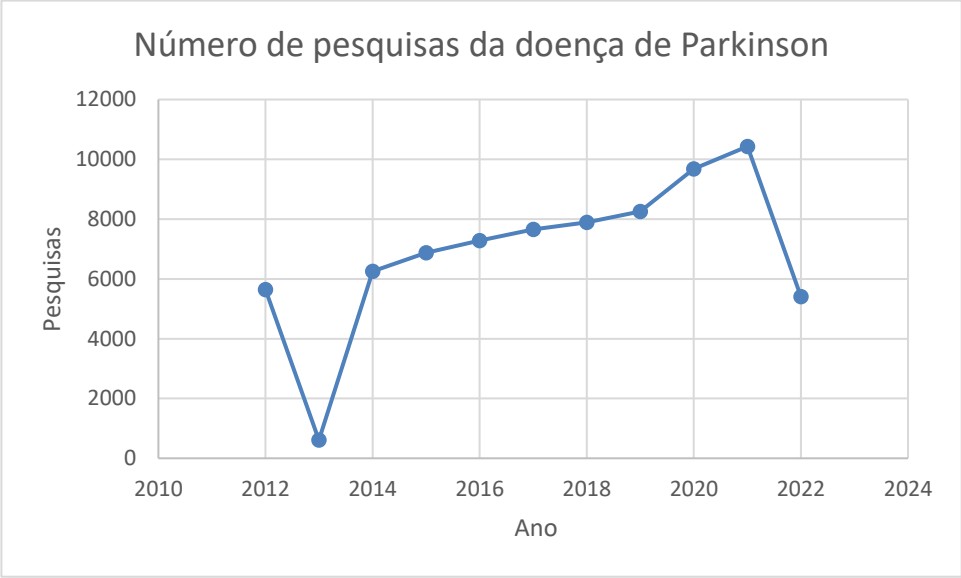

Figura 1 - Verificação do número de pesquisas da doença de Parkinson no *Pubmed* (PubMed, 2022).

Existe uma forte aposta na alocação de recursos de investigação para a descoberta de sistemas que melhorem o diagnóstico desta doença, que aumentem o sucesso de tratamento da doença, com o principal objetivo de salvar vidas, reduzir custos e tempos de diagnóstico.

1.2 Motivação e Objetivos

A seguir ao Alzheimer, a doença de Parkinson é a segunda doença neurodegenerativa que causa mais complicações nos indivíduos que com ela são diagnosticados. A maioria dos casos não são detetados precocemente devido à falta de métodos padrão capazes de realizar o seu diagnóstico. As aplicações de *Machine Learning* exercem um impacto considerável na saúde. Vários sistemas de apoio à decisão já utilizam estas técnicas, em problemas de diagnóstico e tratamento, registos eletrónicos, descoberta de novos medicamentos, ou questões relacionadas com a imagem médica, tal como a dermatologia e a radiologia. Além disso, temos a sua aplicação na monitorização de pacientes na sua própria casa a partir de dispositivos inteligentes. Outro aspeto a considerar nesta área é a descoberta de ferramentas, que pertençam ao nosso dia-a-dia e nas quais são desenvolvidas aplicações capazes de detetar uma doença.

Por enquanto, esta doença não tem cura o que causa inquietação junto da comunidade científica a nível Mundial. Os investigadores têm realizado principalmente estudos ao nível de terapias genéticas. Estas terapias genéticas têm como objetivo o reabastecimento de dopamina através de tratamentos personalizados para a restauração da integridade, anatómica e funcional do circuito cerebral (Ntetsika, Papathoma, & Markaki, 2021). O propósito dos investigadores é encontrar formas de minimizar os sintomas motores e não motores aos portadores da doença de Parkinson com o objetivo de lhes proporcionar um envelhecimento mais confortável. Nos últimos tempos, têm sido desenvolvidas aplicações de baixo custo, acessíveis à generalidade dos indivíduos, tal como o desenvolvimento de aplicações para smartphones que permitem detetar a doença de Parkinson, através da voz, ou para a monitorização da sua progressão a partir de casa. Sendo que estes dispositivos são usualmente conhecidos como "*Smart Devices*". Estas aplicações proporcionam ao indivíduo uma maior comodidade.

A principal motivação deste trabalho de dissertação foi a demonstração da utilidade de técnicas de *Machine Learning* no processo de diagnóstico mais rápido da doença de Parkinson. Além disso, pretendia-se identificar, entre os algoritmos aplicados, quais é que apresentariam um melhor desempenho e que pudessem ser melhorados e aplicados à deteção/classificação de diferentes doenças tais como depressão, Alzheimer, stress pós-traumático e até mesmo doenças cardiovasculares. Para a realização deste trabalho de dissertação foram estabelecidos os seguintes objetivos:

- revisão literária da doença de *Parkinson* e estudos impactantes na área;
- seleção e estudo dos modelos de *Machine Learning* para sua implementação, bem como estudar a configuração dos parâmetros de cada um dos algoritmos;

- avaliação crítica dos resultados obtidos e sua comparação com resultados obtidos noutras investigações/estudos.

1.3 Metodologia de Trabalho

A metodologia de investigação utilizada nesta dissertação foi a *Cross Industry Standard Process for Data Mining* (CRISP-DM). Esta metodologia é frequentemente aplicada em projetos de *Data Mining* (Langley & Carbonell, 1984).

Figura 2 - Metodologia CRISP-DM. Adaptado de Langley e Carbonell (1984).

A metodologia CRISP-DM desenvolve-se em 6 fases (Figura 2), nomeadamente (Schröer, Kruse, & Gómez, 2021):

- **Compreensão do negócio**. Nesta fase deve ser avaliada a situação do negócio, de forma a se obter uma visão geral do mesmo, tal como recursos disponíveis e necessários. Nesta fase é importante definir-se o objetivo do projeto, quais dados necessários para o seu desenvolvimento e elaborar-se um plano para atingir os pontos referidos.
- **Compreensão dos dados**. Aqui, procede-se à recolha dos dados necessários na fonte de informação selecionada e realiza-se a exploração, descrição e verificação da sua qualidade.

- **Preparação dos dados**: Esta fase corresponde à seleção dos dados e os quais são selecionados através de critérios de inclusão e exclusão. De forma sucinta, esta fase diz respeito ao processo que engloba todas as atividades de transformação de dados, tais como: correção de problemas, limpeza, integração e formato dos dados.
- **Modelação:** A fase de modelação consiste em selecionar diferentes técnicas de modelação, realizar a sua implementação e construir o modelo propriamente dito. Além disso, os parâmetros de cada algoritmo devem ser parametrizados para encontrar os valores ótimos para cada parâmetro.
- **Avaliação:** Corresponde à fase de avaliação dos resultados obtidos do modelo (ou modelos). Por conseguinte, os resultados devem ser avaliados minuciosamente para se verificar se é necessário rever mais alguma medida decidida anteriormente, antes do modelo passar para a fase de implementação.
- **Implementação:** Nesta fase pretende-se organizar todo o conhecimento extraído, bem como apresentá-lo de forma clara.

1.4 Estrutura da dissertação

Para além do presente capítulo, esta dissertação está estruturada em mais cinco capítulos. O segundo capítulo contém a revisão bibliográfica do tema abordado, sendo o capítulo que apresenta uma componente teórica mais sólida. De seguida, no terceiro capítulo, aborda-se as áreas de *Data Mining* e *Machine Learning,* bem como a sua relação, descrevendo-se as diferentes técnicas de *Machine Learning* existentes, bem como as diferentes etapas de um processo de *Machine Learning*. Por fim, apresentam-se as técnicas de pré-processamento utilizadas no estudo. No quarto capítulo apresenta-se o levantamento teórico dos algoritmos de classificação que foi realizado, bem como a descrição dos seus parâmetros, em particular aqueles que serão aplicados na classificação da doença de Parkinson. No capítulo seguinte, o quinto capítulo, apresenta-se e descreve-se o caso de estudo, os dados utilizados e a análise dos resultados obtidos. Para terminar esta dissertação, desenvolvemos o capítulo 6 que enuncia as conclusões alcançadas e enumera algumas linhas para trabalho futuro.

2. A DOENÇA DE PARKINSON

2.1 A Doença

A doença de Parkinson é uma doença neurodegenerativa progressiva, descoberta em 1817 pelo Doutor James Parkinson (Fröhlich, 2016). É uma doença que se caracteriza pelos sintomas motores causados pelas proteínas que se desenvolvem no interior das células nervosas denominadas por "*Lewy Bodies*", mas também devido à perda de neurónios dopaminérgicos na substância *Nigra* do cérebro (Kalia & Lang, 2015). No entanto esta doença também pode ser diagnosticada através de sintomas não motores como, por exemplo, alterações no olfato e fadiga.

Vários fatores de risco são apontados para o desenvolvimento da doença de Parkinson, por exemplo, a idade, o género ou fatores ambientais (pesticidas e poluição atmosférica). O que causa a doença de Parkinson ainda permanece desconhecido, mas supõe-se que surge de uma interação complexa entre fatores genéticos e fatores ambientais ao longo da vida (Kalia & Lang, 2015). Relativamente ao fator de risco do género a probabilidade de desenvolver esta doença é duas vezes maior no género masculino do que no género feminino. No entanto as mulheres têm uma taxa de mortalidade mais elevada e uma progressão de Parkinson mais acelerada (Cerri, Mus, & Blandini, 2019).

Os sintomas de Parkinson e a sua manifestação variam de indivíduo para indivíduo, tanto ao nível de intensidade como da evolução. A(s) causa(s) desta doença ainda continuam desconhecidas, embora se conheça já algumas evidências de possíveis fatores de risco, como fatores genéticos, fatores ambientais ou a combinação de ambos. Embora possam surgir vários sintomas não motores, os sintomas típicos desta doença são os motores e envolvem distúrbios ao nível do movimento, tremores, rigidez e instabilidade ao nível da postura, isto é, problemas de equilíbrio (Stoker & Greenland, 2018). Na Tabela 1 são apresentados os vários sintomas associados ao Parkinson, que estão classificados como sintomas motores e não motores, respetivamente.

Tabela 1 - Os sintomas mais comuns da doença de Parkinson.

Sintomas Motores	Sintomas Não Motores
Tremores	Distúrbios de sono
Lentidão dos movimentos	Depressão e Ansiedade
Rigidez muscular	Diminuição da capacidade cognitiva
Alterações na fala	(-)
Instabilidade na postura	(-)

2.2 Diagnóstico e tratamento

Não existe um procedimento de diagnóstico específico para a deteção da doença de Parkinson, mas é frequente realizar-se um exame neurológico detalhado, no qual é incluído uma revisão do histórico médico de cada indivíduo e se realiza uma avaliação clínica dos sintomas motores e não motores. Os neurologistas podem utilizar duas escalas, a escala "*Movement Disorder Society – sponsered revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS)*" (Goetz et al., 2007) ou a escala de "*Hoehn and Yahr*" (Poewe, 2012). Tipicamente estas escalas baseiam-se nos sintomas motores apresentados por cada indivíduo. No entanto, em casos mais difíceis de estabelecer um diagnóstico para a doença de Parkinson, é possível recorrer a exames de imagem médica como, por exemplo, o *DaTscan* (Gayed et al., 2015), o qual permite detetar a presença de dopamina no cérebro. Contudo, existem outros métodos de imagem médica como MRI (*Magnetic Ressonance Imaging*) (Pyatigorskaya et al., 2014) ou o PET (*Positron emission tomography*) (Loane & Politis, 2011), no entanto estes métodos não são muito utilizados.

Quanto ao tratamento da doença de Parkinson os medicamentos dopaminérgicos são o método de tratamento mais típico para o tratamento do Parkinson, sendo o *Levedopa*/L-DOPA (Tambasco, Romoli, & Calabresi, 2018) o medicamento mais aplicado no tratamento dos sintomas motores de Parkinson. Sendo considerado o medicamento "padrão" de tratamento desta doença. Quando tomado o medicamento *Levedopa* este entra no corpo do Ser Humano e é convertido nos gânglios basais onde aumenta a dopamina nessa região, funcionando como um tipo de neurotransmissor entre os neurónios e coordenação motora do corpo. Este medicamento pretende atenuar os sintomas motores causados por esta doença, proporcionando ao indivíduo uma maior qualidade de vida. No entanto, existe um método de tratamento mais invasivo no tratamento da doença de Parkinson: o *Deep Brain Stimulation*

(DBS) (Neilson, Zande, & Abboud, 2020). Este método cirúrgico é frequentemente utilizado para reduzir os sintomas motores do Parkinson. De acordo com Benabid (2003), o *Deep Brain Stimulation* é um método de tratamento de neurocirurgia funcional, que consiste numa contínua estimulação elétrica da estrutura neural do cérebro. Esta estimulação é feita através de elétrodos, os quais são implantados e ligados a um estimulador, que é programável ao nível da amplitude, largura e frequência. A afinação do estimulador pode ser personalizada em intervalos de tempo, dependendo de cada paciente.

2.3 Alguns Estudos realizados

A doença de Parkinson tem sido investigada através da aplicação de algoritmos de *Machine Learning*, em particular aqueles que envolvem classificação. Diferentes domínios de deteção têm sido investigados, tais como a deteção a partir da voz, movimento/postura, imagem médica (ressonância magnética), desenho e escrita dos pacientes. Os estudos abordados nesta secção são todos referentes à classificação da doença de Parkinson, a partir de conjuntos dados relativos à voz.

2.3.1 A adequação das medidas de disfonia para a telemonitorização da doença de Parkinson.

Um dos primeiros conjuntos de dados disponibilizado para estudo foi introduzido por Max (2009). Este conjunto de dados é constituído por 195 registos de voz, recolhidos de 31 indivíduos, entre os quais 23 têm a doença de Parkinson e os restantes 8 não têm a doença e o qual contém 24 características (Tabela 2) relacionadas com a voz. Todas estas características são fundamentais para deteção da doença de Parkinson em todos os estudos.

Tabela 2 - Características do estudo de Sharanyaa et al. (2020).

Frequência	Jitter	Shimmer	Tonalidade da voz	Outras
MDVP: Fo(Hz)	MDVP: Jitter (%)	MDVP: Shimmer	Harmonic to Noise Ratio	RPDE (Recurrent Period Density Entropy)
MDVP: Fhi(Hz)	MDVP: Jitter (Abs)	MDVP: Shimmer (dB)	Noise to Harmonic Ratio	DFA, PPE (Pitch Period Entropy)
MDVP: Flo(Hz)	MDVP: RAP	Shimmer: APQ3, APQ5,	(-)	(-)
(-)	MDVP: PPQ	MDVP: APQ	(-)	(-)

Sharanyaa et al. (2020) utilizaram esse conjunto de dados para classificação da doença de Parkinson utilizando os seguintes algoritmos de *Machine Learning*: *Naive Bayes* (Yang, 2018), *Logistic Regression* (Park, 2013), *Random Forest* (Breiman, 2001), *K-Nearest Neighbors* (Cunningham & Delany, 2020), com objetivo de compreender qual deles é que apresentaria um melhor desempenho no estudo em questão. O seu estudo segue as seguintes etapas: aquisição do conjunto de dados, pré-processamento dos dados, aplicação dos algoritmos, classificação da doença e comparação entre o desempenho dos modelos. Na etapa de pré-processamento, inicialmente aplicou-se a standardização, a qual normaliza cada variável do conjunto de dados com base na média do conjunto de dados, selecionando assim 13 das 24 variáveis. De seguida, ainda na mesma etapa e apenas nas 13 variáveis selecionadas, aplicou uma normalização no intervalo [0,1].

Após a etapa de pré-processamento os autores aplicaram algoritmos de classificação para classificar todos indivíduos que têm a doença de Parkinson ao qual é atribuída a classe 1 e aos indivíduos que não têm a doença de Parkinson é atribuída a classe 0. Os autores aplicaram os seguintes algoritmos *Nayve Bayes*, *Logistic Regression*, *Random Forest* e o *K-Nearest Neighbor.* Relativamente aos resultados obtidos, o *Nayve Bayes* obteve uma precisão de 71.5%, o *Logistic Regression* teve uma precisão de 80%, o *Random Forest* obteve uma precisão de 87.27% e o *K-Nearest Neighbor* obteve uma precisão de 90.2% em que o parâmetro definido que devolveu esta precisão foi para um k=5. Este k, é calculado através da distância de cada um dos pontos já classificados em relação à amostra que se pretende classificar. Assim, os modelos que obtêm um melhor desempenho são o *Random Forest* e o *K-Nearest Neighbor.* O estudo foi desenvolvido através do software *Rstudio*.

2.3.2 Deteção da doença de Parkinson através de características vocais, com métodos *ensemble* de *Machine Learning*: Um estudo de desempenho.

Nissar, Rizvi, Masood, & Mir (2019) realizaram um estudo no qual utilizaram um conjunto de dados retirado da *UCI Machine Learning Database* (UCI Machine Learning Repository, 2018) este conjunto de dados é constituído por 756 registos e 753 características, especialmente utilizadas para classificação da doença de Parkinson através da voz. O estudo assenta no impacto da escolha do método de "*Feature Selection*" (seleção das características) na classificação da doença de Parkinson. Neste trabalho os autores seguem as seguintes etapas: aquisição dos dados, pré-processamento dos dados, "*Feature Selection*", aplicação de modelos de *Machine Learning* e avaliação da sua eficiência. Na etapa de pré-processamento dos dados apenas colocaram as variáveis todas à mesma escala, através da normalização mínimo-máximo correspondendo ao intervalo [0,1]. De seguida, treinaram 9 algoritmos, nomeadamente *Nayve Bayes, Logistic Regression, K-Nearest Neighbors, Multilayer Perceptron, Random Forest, Support Vector Machine* (linear), *Support Vector Machine* (rbf) e *XGBoost*, da seguinte forma:

1. Seleção de características com a técnica *Recursive feature selection* e *minimum redundancy feature selection*, exceto o grupo de características TWQT.
2. Seleção de características com a técnica *Recursive feature selection* e *minimum redundancy feature selection*, exceto o grupo de características MCFF.
3. Técnicas de seleção de características *Recursive feature selection* e *minimum redundancy feature selection* com todos os grupos de características.

No entanto, de entre todos os algoritmos aplicados, os autores verificam que o algoritmo que obteve um melhor desempenho foi o *XGBoost,* com uma precisão de 95.39%, quando utilizada a técnica de seleção de características minimum redundancy feature selection. Relativamente aos restantes algoritmos aplicados, o *Logistic Regression* providenciou um modelo *com* uma precisão de 86.18%, utilizando a seleção de características *minimum redundancy feature selection* e o algoritmo *Random Forest* obteve uma precisão de 86.84% nas duas técnicas de seleção de características. O algoritmo *Multilayer Perceptron* obteve uma precisão de 84.86% quando treinado com todas as características, com a técnica de *minimum redundancy feature selection*. O algoritmo *Support Vector Machine* foi treinado com os dois tipos de *Kernels,* o linear e o rbf. Com o *Kernel* linear o algoritmo obteve uma precisão de 84.21% nas duas técnicas de seleção de características, o que comprovou que tanto a técnica de seleção de características *Recursive feature selection* e *minimum redundancy feature selection* não causam impacto

nos resultados da classificação da doença de Parkinson neste modelo. Quanto ao *Kernel* rbf obteve uma precisão de 88.15% com a técnica de seleção de características *minimum redundancy feature selection*.

Por conseguinte, os autores concluem que o algoritmo *XGBoost* é o algoritmo que devolve a melhor precisão quando aplicada a técnica de seleção de características *minimum redundancy feature selection*. Destacam ainda, que embora o modelo funcione de forma eficiente é um pouco limitado devido ao tamanho do conjunto de dados.

2.3.3 Análise da voz para reconhecimento de padrões da doença de Parkinson

Tai, Bryan, Loayza e Peláez (2021) utilizaram um conjunto de dados proveniente do projeto "*Mobile Parkinson Disease Study*" introduzido por Bot et al. (2016). O conjunto de dados utilizado continha cerca de 65000 áudios adquiridos através de um *smartphone*. O áudio continha a gravação da vogal "aaaa", com uma duração de 10 segundos a uma frequência de 44,1 kz. Além disso, o conjunto de dados contêm ainda um conjunto de vozes de um grupo de controlo. Na seleção dos áudios os autores recorrem à implementação de critérios tais como: a divisão dos indivíduos em grupos (pacientes e grupo de controlo). No entanto, no grupo de pacientes escolhe os áudios com base nas respostas recolhidas do seguinte questionário:

- "Quem fez a gravação antes de tomar a medicação para o *Parkinson*?", "Tem um diagnóstico de *Parkinson* positivo e realizado por um profissional especializado?"
- "Consome algum tipo de medicação para tratar a doença?"
- "Pertence à faixa entre os 50 e 75 anos?".

No grupo de controlo escolhe apenas os registos dos indivíduos que pertencem a uma faixa etária entre os 50 e 75 anos. Relativamente ao pré-processamento de sinal filtrou apenas os áudios compreendidos entre as frequências -20db a -3db. Após a aplicação deste critério ficou com apenas 1400 áudios, em que 700 das amostras utilizadas correspondem aos indivíduos que têm a doença de Parkinson e equivalem ao grupo de controlo. Tai et al., 2021 aplicaram dois métodos diferentes ao nível de seleção das características o *High Correlation Filter* (Bharadwa, 2021) e método *Principal Component Analysis* (Pearson, 1901) em que as componentes principais foram selecionadas com base na variância acumulada de 97% corresponde a 27 componentes principais. Tai et al. (2021) aplicaram os seguintes modelos de classificação: *Multilayer Perceptron*, *Support Vector Machine*, *Logistic Regression*, *Random Forest*, nos quais os seus parâmetros foram otimizados através do método de *Fine-Tuning* para selecionar os valores ótimos de cada parâmetro.

Os modelos foram treinados, validados e testados, sendo que os modelos *Multilayer Perceptron*, *Random Forest* e *Support Vector Machine* apresentam um melhor desempenho em termos de precisão quando a técnica de seleção de características é a *Principal Component Analysis*, apresentando uma precisão de 86%, 82% e 87%. O algoritmo *Logistic Regression* apresenta um melhor desempenho quando utilizada a técnica *High Correlation Filter*, com uma precisão de 80%. No estudo realizado por Tai et al. (2021), concluiu-se que o modelo que melhor classifica a doença de Parkinson é o *Support Vector Machine* com a técnica de seleção de características *Principal Component Analysis*. O estudo foi realizado no software *Python* na versão 3.0, em particular, as suas bibliotecas *Librosa* e *Parselmouth* para a extração das características do áudio.

2.3.4 Análise da voz como instrumento de auxílio na deteção da doença de Parkinson e subsequente interpretação clínica

Por sua vez, Solana-Lavalle e Rosas-Romero (2021) recorreram a um conjunto de dados disponibilizado por C. O. Sakar et al. (2019), o qual contêm 756 características e 252 indivíduos que repetiram três vezes a vogal "a". O conjunto de dados é constituído pelos seguintes grupos de características: características fundamentais da voz, características de tempo-frequência, características das cordas vocais, características dos coeficientes cepstrais de Mel (MFCC), características da Transformada de *Wavelet*, características da transformada de *Wavelet* com o fator de qualidade Q (TQWT). No seu estudo aplicaram o método conhecido por *Wrapper feature selection* para seleção das características mais importantes, sendo excluído o grupo de características de tempo-frequência e as características das cordas vocais. De seguida, aplicaram a técnica de *Principal Component Analysis* para verificarem quais as características do grupo TQWT possuem uma maior variância. Neste trabalho a técnica de *Principal Component Analysis* foi exclusivamente aplicada para a demonstração e comparação estatística do comportamento das 4 características selecionadas no grupo de indivíduos que apresentam a doença de Parkinson e no grupo de controlo. Essa comparação é feita através de um gráfico diagrama de caixa por meio de intervalos de quartis, no qual se avalia a média e o desvio padrão dessas características. Os autores estudaram três abordagens diferentes.

Primeiramente aplicaram a técnica de *Principal Component Analysis* no conjunto de dados balanceado, com todos os indivíduos do sexo feminino e do sexo masculino. O grupo de características TQWT selecionado integrava as propriedades: LT TKEO valor médio para o oitavo coeficiente TQWT com variância de 4,93%, LT TKEO desvio padrão do sexto coeficiente TQWT com variância de 4,873%, LT TKEO valor médio do sétimo coeficiente TQWT com variância de 4,866%, det TKEO valor médio para o

trigésimo terceiro coeficiente TQWT com variância de 4,699%. Na segunda experiência utilizaram o conjunto de dados, balanceado e não balanceado, do sexo masculino no qual foram selecionadas as seguintes características: o valor médio det TKEO do trigésimo coeficiente TQWT com uma variância de 4,699%, a entropia *Shannon* para o quinto coeficiente TQWT com uma variância de 5,52%, o valor mínimo para o quinto coeficiente TQWT com uma variância de 5,43%, o valor médio TKEO para o quinto coeficiente TQWT com uma variância de 5,27%.

Na terceira experiência, os autores utilizaram um conjunto de dados balanceado e não balanceado, contendo apenas os indivíduos do sexo feminino. Nesta experiência foram selecionadas as seguintes características: a energia do trigésimo coeficiente TQWT com uma variância de 5,99%, o valor médio de TKEO para o coeficiente trigésimo segundo de TQWT com uma variância de 5,98%, a entropia Shannon do trigésimo segundo coeficiente de TQWT com uma variância de 5,89%, a entropia Shannon do trigésimo terceiro coeficiente de TQWT com uma variância de 5,83%.

Posteriormente, os autores aplicaram os algoritmos de *Machine Learning*: *K-Nearest Neighbors*, *Random Forest*, *Multilayer Perceptron*, *Support Vector Machine* nos quatro grupos de características escolhidas pelo método *wrapper feature selection* (Kohavi & John, 1997) em três procedimentos diferentes. Primeiro, aplicaram os algoritmos na população que continha todos os indivíduos do sexo feminino e do sexo masculino, com o conjunto de dados balanceado. O modelo que obteve um melhor desempenho foi o *Multilayer Perceptron, com* uma precisão de 89%. Na segunda experiência foram utilizados apenas os indivíduos masculino com o conjunto de dados balanceado e não balanceado. No conjunto de dados balanceado, o modelo com melhor desempenho foi o *Random Forest,* com uma precisão de 92.7%, e, no conjunto de dados não balanceado, o modelo com melhor desempenho foi o *K-Nearest Neighbor* com uma precisão de 94.3%.

Na terceira experiência utilizou apenas os indivíduos do sexo feminino com o conjunto de dados balanceado e não balanceado. No conjunto de dados balanceado o modelo que devolve uma maior precisão é o *Multilayer Perceptron* com uma precisão de 89%. Quanto ao conjunto não balanceado o modelo que apresenta uma maior precisão é o *K-Nearest Neighbor* com 95%. Através do seu estudo os autores concluíram que obtiveram melhores resultados comparativamente com a literatura que estudaram, e que, as características relativas à deteção do Parkinson a partir da voz dependem do género do indivíduo. Apontam ainda, que características relacionadas com a frequência apresentam uma maior contribuição para a deteção de Parkinson e que no caso do sexo masculino corresponde à gama de baixa frequência e no caso do sexo feminino corresponde à gama de alta frequência. Outra conclusão retirada é que o grupo de características mais importante é o TQWT.

Por fim, o modelo que apresenta melhores resultados comparando com a literatura que o autor estudou é o *K-Nearest Neighbor* com uma precisão de 95,9%. O software utilizado pelos autores no seu estudo foi o software *Weka* (Hall et al., 2009), o qual permite construir modelos de *Machine Learning* sem conhecimento em programação, utilizaram ainda a biblioteca de *Wrapper feature selection* deste software para selecionarem as características mais importantes.

2.3.5 Modelo de seleção de características híbrido para a classificação da doença de Parkinson num conjunto de dados desequilibrado

Qasim et al. (2021) também estuda uma metodologia de seleção de características híbrida para classificação da doença de Parkinson num conjunto de dados desequilibrado. O conjunto de dados selecionado foi introduzido em C.O. Sakar et al. (2019). Neste trabalho, o autor segui as seguintes etapas: pré-processamento do conjunto de dados, divisão do conjunto de dados num conjunto de dados de treino e num conjunto de dados de teste, aplicação dos algoritmos de *Machine Learning*, aplicação do método de *Fine-Tuning (GridSearch)* e avaliação de desempenho dos algoritmos. Relativamente à etapa de pré-processamento o autor aplicou a técnica de SMOTE a qual equilibrou o conjunto de dados com 564 registos na classe 0 e na classe 1, de seguida colocou todas as variáveis na mesma escala entre 0 e 1. Quanto às técnicas de seleção de características o autor aplicou a técnica *Recursive Feature Selection* a qual selecionou 329 características, de seguida, aplicou a técnica *Principal Component Analysis* a qual selecionou dentro das 329 características apenas 18. Posteriormente, segue-se a etapa de divisão do conjunto de dados num conjunto de dados de treino e num conjunto de dados de teste. Ao conjunto de dados de treino é atribuído 80% do conjunto de dados inicial e ao conjunto de dados de teste é atribuído 20% do conjunto de dados inicial. Para validarem cada um dos conjuntos de dados, os autores aplicaram o método de *cross-validation,* com k=10. Isto é, dividiram os dados em 10 subconjuntos, utilizando nove desses subconjuntos para treino e um para teste, obtendo a precisão inicial do modelo criado com os dados de teste. Após essa avaliação, os restantes subconjuntos são iterados repetidamente até que todos os subconjuntos tenham sido testados e a precisão final do modelo é obtida através da média da precisão dos 10 subconjuntos testados. Os autores aplicaram os algoritmos *Multilayer Perceptron*, *Support Vector Machine*, *K-Nearest Neighbor* e o *Bagging* e os quais foram treinados e testados 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* e o método de *Fine-Tuning (*Tabela 3*)*.

Por conseguinte, para o primeiro caso o algoritmo que obteve um melhor desempenho foi o *K-Nearest Neighbor* 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 *Multilayer Perceptron* com uma precisão de 93.3% e com 329 características. No terceiro caso, o algoritmo que obteve um melhor desempenho foi também o *Multilayer Perceptron* 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 (Tabela 3) e o algoritmo que teve um melhor desempenho foi o *Support Vector Machine* com uma precisão de 98%.

Tabela 3 - Parâmetros otimizados para modelo de Machine Learning Qasim et al. (2021).

Algoritmos de classificação	Parâmetros otimizados
Multilayer Perceptron	Número de iterações = 500 Taxa de aprendizagem = 0.01 Peso ótimo = adam
Support Vector Machine	Parâmetro de regularização = 1 Kernel = rbf Gamma = 2
K-Nearest Neighbor	Número de vizinhos = 1 Tamanho da folha = 40
Bagging	Classificador = K-NN Número de iterações = 100 Máximo de amostras = 0.9

2.3.6 Melhorar o diagnóstico da doença de Parkinson através de algoritmos de *Machine Learning*

Celik e Ilhan Omurca (2019) realizaram um estudo sobre formas de como melhorar o diagnóstico da doença de Parkinson utilizando modelos de *Machine Learning.* O conjunto de dados utilizado foi o de B. E. Sakar et al. (2013) e o qual é constituído por 40 indivíduos dos quais 20 têm a doença de Parkinson (6 género feminino e 14 do sexo masculino) dentro da faixa etária 43 a 77 anos. Os restantes indivíduos correspondem aos indivíduos saudáveis não apresenta a doença (10 género feminino e 10 género masculino) na faixa etária entre os 45 e 83 anos. Foram gravadas 26 amostras através de um microfone regulado entre os 96 kHz e 30db a 10 cm de distância do indivíduo, em que amostras gravadas incluem: vogais, números, palavras e frases e foram extraídas as características relativas à voz (Tabela 4).

Tabela 4 - Conjunto de dados: Características extraídas.

Característica	Definição
1-5	Jitter
6-11	Shimmer
12-14	Parâmetros de harmonicidade
15-19	Frequência máxima, Frequência mínima, Desvio Padrão, Frequência Média, Mediana da Frequência.
20-23	Desvio padrão do período, Média do período, Número de períodos, Número de impulsos
24-26	Grau de falha da voz, Fração local de um frame não vocalizado, Número de falhas da voz

Posteriormente, os autores recolheram amostras de 28 indivíduos com a doença de Parkinson que continha apenas gravações de três repetições de vogais “a” e “o”, o que representa no total 168 indivíduos. Assim, nesse estudo utilizaram 1040 registos nos conjuntos de dados de treino e 168 registos no conjunto de dados de teste. Ao nível da etapa de pré-processamento do conjunto de dados para seleção das características mais importantes aplicaram a matriz de correlação, a técnica *Principal Component Analysis* e a técnica *Information Gain*. Através da matriz de correlação visualizaram a correlação entre as características e aplicaram a técnica *Principal Component Analysis* para a seleção das características mais importantes, através da qual o conjunto de dados foi reduzido de 26 características para 5 características, com um *threshold*(filtro) de 0.8. O mesmo raciocínio foi aplicado com a aplicação da técnica de *Information Gain* que das 26 características selecionou 9, com um *threshold*(filtro)de 0.4. O conjunto de dados foi dividido num conjunto de dados de treino com 80% de

dados do conjunto inicial e num conjunto de dados de teste 20% de dados do conjunto inicial. Quanto aos modelos que obtiveram um melhor desempenho neste trabalho, foram o *Logistic Regression* com uma precisão de 76.03% e o *Support Vector Machine* com uma precisão de 75.49% no conjunto de dados em que foi aplicada a matriz de correlação com características originais. No entanto, os modelos que obtiveram um melhor desempenho quando utilizada a técnica matriz de correlação e técnica de *Information Gain*, foram o modelo *Random Forest* que obteve uma precisão de 72.69% e o *Gradient Boosting* uma precisão de 72.28%.

2.3.7 Diagnóstico da doença de Parkinson: O efeito de autocodificadores na extração de características vocais

Outro dos trabalhos que estudámos foi o realizado por Mohammadi, Mehralian, Naseri e Sajedi (2021). Estes efetuaram um estudo, cujo objetivo era o de comprovar que os modelos de *Machine Learning* clássicos de classificação conseguem obter bons resultados quando comparados com algoritmos de *Deep Learning*. O conjunto de dados que foi utilizado nesse trabalho foi extraído da base de dados *UCI Machine Learning Repository* (UCI Machine Learning Repository, 2018). O conjunto de dados contêm registos vocais de 252 indivíduos, entre os quais 188 têm a doença de Parkinson. O conjunto de dados incorpora valores de 756 características, do qual se destacam os seguintes grupos de características: características fundamentais da voz, características de tempo-frequência, características das cordas vocais, características dos coeficientes cepstrais de Mel (MFCC), características da Transformada de *Wavelet*, características da transformada de *Wavelet* com o fator de qualidade Q (TQWT).

De seguida, na etapa de pré-processamento os autores normalizaram todas as variáveis entre o intervalo [0,1], para ficarem todas à mesma escala. O estudo que realizaram seguiu duas abordagens. Na primeira, fez-se o treino e o teste dos modelos sem que tenha sido feita a extração das características mais importantes no modelo. Depois, aplicaram-se os modelos de classificação *Support Vector Machine*, *XGBoost* e *Multilayer Perceptron.* Para validarem cada um dos subconjuntos de dados, os autores aplicaram o método de *cross-validation,* com k=5. Isto é, dividiram os dados em 5 subconjuntos, utilizando quatro desses subconjuntos para treino e um para teste, obtendo a precisão inicial do modelo criado com os dados de teste. Após essa avaliação, os restantes subconjuntos são iterados repetidamente até que todos os subconjuntos tenham sido testados e a precisão final do modelo é obtida através da média da precisão dos 5 subconjuntos testados.

Nesta primeira abordagem, com os modelos unicamente treinados com o conjunto de dados normalizado, obtiveram-se os seguintes resultados: o modelo *Support Vector Machine*, com o *kernel* poly e o *degree* a 23, obteve-se uma precisão de 94.07% e um F1-Score de 96,08%, enquanto com o modelo *XGBoost,* obteve uma precisão de 92.19% e um F1-Score de 94.92%, com a seguinte definição de parâmetros: *colsample_bytree* = 0.35, *n_estimators* = 325, *max_depth* = 4, *learning rate* = 0.1, *alpha* = 1e-2, *subsample* = 0.75. Por fim, com o modelo *Multilayer Perceptron* obteve-se uma precisão de 90.61% e um F1-Score 93.72%, com os parâmetros definidos da seguinte forma: *camada intermédia* = 160 e *nodos* = 25. Os parâmetros que melhor se ajustam a cada um dos modelos implementados foram descobertos a partir da técnica de *fine-tuning GridSearch.* Na segunda abordagem, para além do conjunto de dados ser normalizado num intervalo [0,1], foi aplicada a técnica de seleção de características "*Autoencoder*" e, posteriormente, aplicados os algoritmos de classificação. Com esta técnica os parâmetros definidos para treinar o algoritmo *Multilayer Perceptron* são as camadas intermédias = 3,9,27,81,243 e cada neurónio têm definida cada camada intermédia uma função de ativação a *tanh.* A saída possuía uma camada final com a função de ativação *sigmoid* e a qual foi treinada em 100 épocas em lotes de 100 com o otimizador *Adam*, obtendo uma precisão de 91.53% e um F1-Score de 94.36%. Com esta técnica foi ainda treinado o modelo *Support Vector Machine*, configurado da seguinte forma: camadas intermédias = 500, 250, 25, 250, 500, 753, *gamma*=0.01 e o *kernel* = *rbf*. Este modelo obteve uma precisão de 91.93% e um F1-Score de 94.71%.

Por fim, os autores concluíram que os algoritmos aplicados são capazes de distinguir indivíduos que apresentam a doença de Parkinson e indivíduos que não apresentam a doença, apresentando uma precisão entre os 95% e 97%, com a aplicação de algoritmos como o *Support Vector Machine, XGBoost, Multilayer Perceptron* e *Support Vector Machine* com a extração de características através do método *Autoencoder*. Ainda que os algoritmos de *Deep Learning* sejam muito eficientes na classificação desta doença, Mohammadi et al. (2021) referem que os modelos de *Machine Learning* clássicos apresentam bons resultados e podem ser utilizados *à priori* no caso do conjunto de dados apresente um baixo número de registos. Isso deve-se, por exemplo, a razões como a eficiência no tempo de execução dos algoritmos *Support Vector Machine* e *XGBoost* quando comparados com algoritmos de *Deep Learning, que* são interpretáveis relativamente à sua complexidade e têm uma boa capacidade de generalização. No entanto, no estudo desenvolvido existe um erro associado à classificação da doença de Parkinson de 7% a 10% na deteção de indivíduos que não apresentem a doença de Parkinson. Para concretizarem este estudo os autores utilizaram o software *Python*, particularmente as bibliotecas *sklearn.preprocessing,*

sklearn.svm,sklearn.neural_network, sklearn.ensemble, sklearn.linear_model, sklearn.model_selection, sklearn.metrics.

2.4 Principais características de cada estudo

Nas secções anteriores (2.3.1 a 2.3.7) foram abordados vários estudos com o objetivo de classificar a doença de Parkinson através características vocais dos indivíduos. Na Tabela 5, 6 e 7 podemos ver um resumo das principais características de cada um desses estudos.

Tabela 5 - Quadro resumo dos estudos na subsecção 2.3.1 e 2.3.2.

Subsecção de cada estudo	Principais características	Algoritmos	Melhor Modelo
2.3.1	- Conjunto de dados com 195 registos de voz e 24 características. - Pré-processamento standardização das características e seleção de apenas 13. De seguida. Aplicação da normalização das 13 características no intervalo [0,1]. - Treino e teste dos modelos e análise dos resultados obtidos. - Software utilizado *RStudio*.	*Naive Bayes, Logistic Regression, Random Forest, K-Nearest Neighbors*	*K-Nearest Neighbor* com k=5, com uma precisão de 90.2%.
2.3.2	- Conjunto de dados com 753 características da vocais. -Pré -processamento: aplicação as técnicas de seleção de características *Minimum redundancy maximum relevance feature selection* e *Recursive feature selection*. Normalização das características no intervalo [0,1]. - Treino e teste dos modelos da seguinte forma: 1. Seleção de características com a técnica *Recursive feature selection* e *Minimum redundancy maximum relevance feature selection*, exceto as características TWQT. 2. Seleção de características com a técnica *Recursive feature selection* e *Minimum redundancy maximum relevance feature selection*, exceto as características MCFF. 3. Técnicas de seleção de características *Recursive feature selection* e *Minimum redundancy maximum relevance feature selection* com todas as características.	*Nayve Bayes, Logistic Regression, K-Nearest Neighbors, Multilayer Perceptron, Random Forest, Support Vector Machine (linear), Support Vector Machine (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 *Logistic Regression* tem uma precisão de 86.84%. - Nas duas técnicas o modelo *Support Vector Machine* com o *kernel rbf* tem uma precisão de 88.15%.

Tabela 6 - Quadro resumo dos estudos nas subsecções 2.3.3 e 2.3.4.

Subsecção de cada estudo	Principais características	Algoritmos	Melhor Modelo
2.3.3	- Conjuntos de dados com 65000 áudios da voz. - Seleção de 700 áudios com base em respostas de um questionário. - Técnica de seleção de características *High Correlation Filter* e *Principal Component Analysis* que seleciona 27 componentes principais. - Treino e teste dos modelos e otimização dos seus parâmetros através da técnica *GridSearch*.	*Multilayer Perceptron, Support Vector Machine, Logistic Regression, Random Forest.*	- Técnica de seleção de características *High Filter Correlation* o modelo *Logistic Regression* tem uma precisão de 80%. - Técnica de seleção de características *Principal Component Analysis* o modelo *Support Vector Machine* tem uma precisão de 87%.
2.3.4	- Conjunto de dados tem 753 registos; - Pré-processamento foram eliminados os grupos de características de tempo-frequência e características das cordas vocais através do método *Wrapper Feature Selection*. - Treino e teste dos modelos da seguinte forma: 1. população que continha todos os indivíduos do sexo feminino e do sexo masculino, com o conjunto de dados balanceado 2. os indivíduos do sexo masculino com o conjunto de dados balanceado e não balanceado. 3. indivíduos do sexo feminino com o conjunto de dados balanceado e não balanceado. indivíduos do sexo feminino com o conjunto de dados balanceado e não balanceado. - Software utilizado *Weka*.	*K-Nearest Neighbors, Random Forest, Multilayer Perceptron, Support Vector Machine.*	- No primeiro caso o modelo *Multilayer Perceptron* tem uma precisão de 89%. - No segundo caso o modelo Random Forest tem uma precisão de 92.7% no conjunto de dados balanceado e no conjunto de dados não balanceado o modelo o *K-Nearest Neighbor* com uma precisão de 94.3%. - No terceiro caso, no conjunto de dados balanceados o modelo *Multilayer Perceptron* tem uma precisão de 89% e no conjunto de dados não balanceado o modelo o *K-Nearest Neighbor* com uma precisão de 95%

Tabela 7 - Quadro resumo com os estudos nas subsecções 2.3.5, 2.3.6 e 2.3.7.

Subsecção de cada estudo	Principais características	Algoritmos	Melhor Modelo
2.3.5	- 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*.	*Multilayer Perceptron, Support Vector Machine, K-Nearest Neighbors, Bagging.*	- No primeiro caso o modelo *K-Nearest Neighbor* tem uma precisão de 87.4%. - No segundo caso o modelo *multilayer percetron* tem uma precisão de 93.3%. - No terceiro caso o modelo *multilayer perceptron* tem uma precisão 95.1% - No quarto caso o modelo *Support Vector Machine* tem uma precisão de 98%.
2.3.6	- Conjunto de dados com 1040 registos de voz; - Pré-processamento: Técnica de seleção de *Principal Component Analysis* (selecionou 5 características) e *Information Gain* (selecionou 9 características) - Divisão em treino e teste 80% e 20%.	*Logistic regression, Support Vector Machine, Random Forest e Gradient Boosting.*	- Matriz de correlação das características originais do conjunto de dados, o modelo *Logistic Regression* tem uma precisão de 76.03%.
2.3.7	- 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*.	*Support Vector Machine, XGBoost e Multilayer Perceptron.*	- Com todas as características do conjunto de dados o modelo *Support Vector Machine* tem uma precisão de 94.07%. - Técnica de seleção de características *Autocencoder* o modelo *Support Vector Machine* tem uma precisão de 91.93%

Assim da análise da Tabela 5, 6 e 7 podemos ver que todos os estudos foram realizados por autores que utilizaram recursos vocais para a classificação da doença de *Parkinson*. Além disso, note-se que todos os estudos investem na etapa de pré-processamento, na qual predomina principalmente a normalização das características do conjunto de dados na mesma escala entre 0 e 1. Em todos os estudos, a técnica de seleção de características *Principal Component Analysis* é muito utilizada. Porém, a aplicação da combinação de duas ou mais técnicas de seleção de características num mesmo conjunto de dados também é muito aplicada neste contexto da classificação da doença de Parkinson. Por exemplo, o estudo da subseção 2.3.2 com a seleção de características através das técnicas *Recursive Feature*

Selection e *Minimum redundancy maximum relevance*, o modelo *XGBoost* obteve uma precisão de 95.39%. Também o estudo na subsecção 2.3.5 são aplicadas duas técnicas de seleção de características *Recursive Feature Selection* e *Principal Component Analysis*, com o conjunto de dados equilibrado e com os parâmetros otimizados e o modelo que obteve a melhor precisão foi *Support Vector Machine* com uma precisão de 98%.

Quanto aos modelos que foram aplicados nos estudos relatadas nas subsecções 2.3.1 a 2.3.7 pode-se verificar que o estudo abordado na subsecção 2.3.6 é o único que utiliza como técnica de seleção das características a matriz de correlação entre as mesmas e o modelo que obteve um melhor desempenho foi o Logistic Regression com uma precisão de 76.03%. Que comparando com os restantes estudos é o que devolve uma precisão inferior a 80%.

Por fim, deve-se referir que alguns dos estudos relatados nas subsecções 2.3.1 a 2.3.7 foram implementados em software *Python*, *Weka* e *RStudio*, muito aplicados na área de *Machine Learning*.

3. DATA MINING E MACHINE LEARNING

3.1 Relação entre os domínios – Data Mining vs Machine Learning

Hoje em dia, é recorrente falar-se nos domínios de *Data Mining* e *Machine Learning* devido à sua evolução e aplicação em vários contextos tais como: cuidados de saúde, Indústria 4.0, retalho, serviços financeiros, entre outros. Porém, estes dois domínios, apresentam uma correlação associativa já que ambos estão profundamente relacionados com a vertente da "*Data Science*" e com a descoberta de novos padrões a partir de um conjunto de dados. O *Data Mining* é definido como sendo um processo de extração de conhecimento a partir de um grande volume de dados. O principal objetivo deste domínio é encontrar padrões num determinado conjunto de dados que inicialmente não tenham sido analisados e tratados e que depois de sofrerem um processo tratamento sejam encontrados padrões, que auxiliem no apoio à tomada de decisões nos negócios. A Figura 3 mostra a diversidade de domínios nos quais o *Data Mining* está integrado, o que revela a sua multidisciplinariedade.

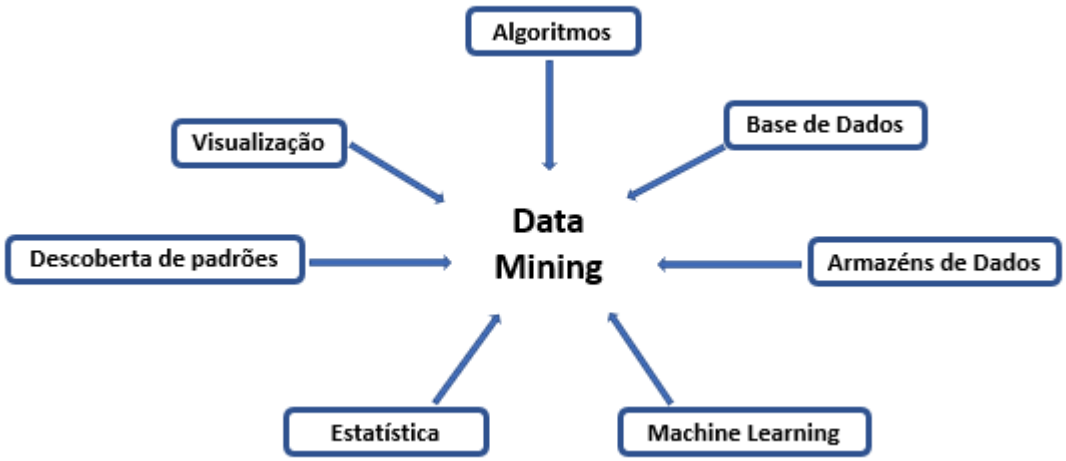

Figura 3 - Ramos relacionados com o Data Mining.

O *Machine Learning* integra-se no domínio da Inteligência Artificial e, de acordo com Mitchell (1997) pode ser definido como o "campo de estudo que fornece a capacidade do computador aprender e automaticamente melhorar através da experiência". Quando se aborda o conceito de *Machine Learning* é importante também compreender o conceito de aprendizagem. Segundo Portugal, Alencar e Cowan (2018) a aprendizagem é o processo de aquisição de conhecimento, no qual os humanos aprendem naturalmente através da experiência e do seu raciocínio, enquanto que os computadores não aprendem por raciocínio, mas sim através de algoritmos. Um algoritmo de *Machine Learning* recorre a um conjunto

de dados de entrada, que servem de *input* ao algoritmo e que são utilizados numa fase de treino com o intuito que o sistema adquira conhecimento com base nesses dados e aprenda a generalizar sem que seja necessária uma programação manual. O qual tem como objetivo reproduzir modelos que sejam capazes de se adaptar e de adquirir conhecimento em experiências passadas, para que seja possível fazer uma previsão ou classificação de casos futuros.

O *Machine Learning* é um subcampo do *Data Mining* e recorre sobretudo às suas técnicas de exploração e visualização de dados. O mesmo, está presente em vários domínios tais como: psicologia, neurociência, filosofia e teoria computacional. Na Tabela 8 podemos algumas das principais diferenças entre estas duas áreas de trabalho.

Tabela 8 - Principais diferenças entre o Data Mining e o Machine Learning.

Data Mining	*Machine Learning*
Extração de informação a partir de um grande volume de dados.	Aplicação de um algoritmo num conjunto de dados, o qual aprende com os dados inseridos e é capaz de prever eventos futuros.
Utilizado para compreensão do processo de fluxo dos dados.	Ensina o computador a aprender e compreender o fluxo de dados.
Recorre a base de dados com dados não estruturados.	Recorre a dados existentes.
Extração dos dados orientada a armazéns de dados.	No *Machine Learning* o computador lê o ficheiro de dados.
O *Data Mining* necessita de intervenção humana para que os dados possam ser processados pelo ser humano.	Recorre ao conceito de uma aprendizagem mecânica para que não haja dependência da intervenção do ser humano.

3.2 Tipos de Machine Learning

De acordo com Stoker e Greenland (2018) existem vários tipos de algoritmos de aprendizagem em *Machine Learning*, que estão divididos em três categorias principais: *Supervised Learning, Unsupervised Learning* e *Reinforcement Learning*. Por vezes, considera-se uma quarta categoria, *Semi-Supervised Learning,* que envolve a combinação dos modelos anteriores. A Figura 4 a presenta um esquema dos diversos tipos de *Machine Learning*.

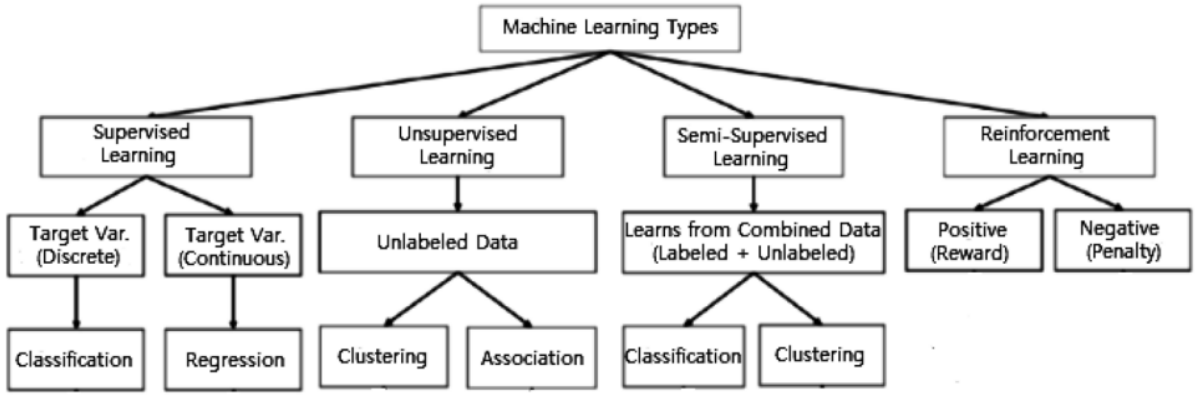

Figura 4 - Tipos de Machine Learning. Adaptado de Sarker (2021).

3.2.1 Supervised Learning

A técnica *Supervised Learning* (aprendizagem supervisionada) começa tipicamente com um conjunto de dados rotulados, destinando-se a encontrar padrões. O modelo de *machine learning* aprende através de um conjunto de dados de entrada com base nos resultados previamente conhecidos, sendo estes são utilizados para treinar o modelo. Os algoritmos mais conhecidos de *Supervised Learning* são a classificação e regressão. Na classificação o objetivo é prever valores de variáveis qualitativas, por exemplo: {0,1}, {apresenta a doença, não apresenta a doença}. Na regressão o objetivo é prever valores de variáveis contínuas, como por o exemplo o preço de ações no mercado financeiro.

3.2.2 Unsupervised Learning

No caso da *Unsupervised Learning* não existem dados de entrada rotulados. Não há informação sobre a qual grupo pertence a variável que se pretende classificar. Neste tipo são apresentados alguns dados ao algoritmo e este aprende com esses dados de forma autónoma (Celebi & Aydin, 2016). Neste tipo de aprendizagem os algoritmos de *Machine Learning* não passam pela fase de treino. Os algoritmos

de *Unsupervised Learning,* focam-se em encontrar padrões através da separação dos dados em grupos com características semelhantes/segmentação em *clusters*.

3.2.3 Semi-Supervised Learning

Outros algoritmos são capazes de lidar, em simultâneo, com uma combinação de dados rotulados e não rotulados, na qual os dados rotulados são em pequeno número e os dados não rotulados existem em grande número. O *Semi-supervised learning* agrupa primeiro os dados semelhantes recorrendo a um algoritmo de *Unsupervised Learning* e, de seguida, utiliza os dados rotulados para rotular os dados que não estejam rotulados.

3.2.4 Reinforcement Learning

O *Reinforcement Learning* é um tipo de aprendizagem que utiliza um sistema de recompensa e penalização nos algoritmos. Este tipo de aprendizagem têm um sistema que se baseia num agente que tenta aprender qual a melhor ação a ser tomada, dependendo das circunstâncias do ambiente, isto é, a ação é tomada num ambiente interativo. Sempre que o agente executa uma ação recebe uma recompensa devido ao seu desempenho correto e uma penalização por um desempenho incorreto (El Bouchefry & de Souza, 2020). Este processo ocorre de forma constante e espera-se que o sistema seja capaz de aprender ações que geram uma maior recompensa em cada ambiente apresentado e que evite ações que possam gerar uma punição (Géron, 2019).

3.3 Etapas do processo de Machine Learning

Em *machine learning,* o processo de aprendizagem ocorre de forma contínua e a seleção do algoritmo certo para o problema em questão é apenas um passo deste processo. As etapas deste processo estão divididas em seis etapas:

1 Identificação dos dados

A identificação dos dados é a primeira etapa pela qual todos os estudos e implementação de algoritmos de *Machine Learning* necessitam de planear. Esse planeamento passa pela seleção da fonte onde os dados irão ser extraídos, por exemplo: uma base de dados, um repositório na internet no qual são disponibilizados *datasets* para estudo de aplicações de Ciência dos Dados.

Esta é uma etapa crítica pois as qualidades dos dados extraídos têm um grande impacto nos resultados obtidos no modelo preditivo.

2 Preparação dos dados

Nesta etapa é avaliada a qualidade do conjunto de dados recolhido e qual sofre um tratamento de acordo com as condições que apresenta. Esta etapa é fundamental para que o modelo preditivo obtenha bons resultados. Deste modo, é necessário realizar uma análise exploratória do *dataset* e tentar identificar pequenos requisitos que precisem por exemplo de um tratamento de dados mais extenso. Deve-se referir, também, que, ao nível do tratamento de dados, é necessário normalmente aplicar técnicas de redução de características, quando estas apresentam uma forte correlação, este passo deve ser realizado cuidadosamente. Outro passo importante é a divisão do conjunto de dados num conjunto de dados de treino e de teste para treino do modelo preditivo.

3 Seleção do algoritmo

A escolha do algoritmo nem sempre é uma escolha trivial, pois a natureza do problema em estudo pode apresentar algumas barreiras no resultado final. Existem algoritmos que funcionam bem e apresentam bons resultados para um determinado problema e outros que não. Deste modo, convém realizar uma pesquisa rápida com trabalhos de outros autores, que tenham aplicado determinado algoritmo a um caso semelhante. Esta etapa costuma ser uma etapa de rápida execução, uma vez que a grande parte dos algoritmos já estão configurados.

4 Treino do algoritmo selecionado

Nesta etapa é utilizado o conjunto de dados definido para treino. No caso de serem aplicados algoritmos de aprendizagem supervisionada, o conjunto de dados está catalogado com as respetivas classes. Assim, pretende-se que o algoritmo aprenda, através das classes catalogadas, a classificar futuramente o número de casos alocados a cada classe.

5 Avaliação do modelo

Finalizada a fase de treino do algoritmo e dependendo do tipo de algoritmo aplicado, deve ser utilizado um conjunto de dados de teste para avaliar a precisão do modelo.

6 Melhoria e ajuste dos parâmetros

Após a fase de avaliação, é possível que seja necessário melhorar o modelo. Para tal propósito os parâmetros do algoritmo poderão ter de sofrer um ajuste e de seguida, é necessário voltar a treinar e a testar o modelo.

3.4 Técnicas de pré-processamento

3.4.1 Técnica Principal Component Analysis

A técnica *Principal Component Analysis* (Análise das Componentes Principais) foi criada por (Pearson, 1901) e (Hotelling, 1933) e é uma técnica comumente aplicada na análise exploratória de dados e na conceção de modelos preditivos. É, ainda, uma das técnicas mais importantes em análise multivariada, sendo uma técnica do tipo *unsupervised learning* (aprendizagem não supervisionada).

A análise de componentes principais é definida como uma técnica que permite reduzir a dimensionalidade de um determinado conjunto de dados aumentando a sua interpretação e minimizando a perda de informação (Jollife & Cadima, 2016). Surgindo como uma técnica de auxílio na interpretação de grandes conjuntos de dados, esta técnica procede a uma transformação ortogonal na qual converte as variáveis correlacionadas (variáveis originais) em componentes principais (variáveis não correlacionadas). As componentes principais são ordenadas, de forma que as primeiras retenham grande parte da variação presente nas características originais. Na Figura 5 podemos ver uma ilustração das etapas base para a aplicação da técnica *Principal Component Analysis*.

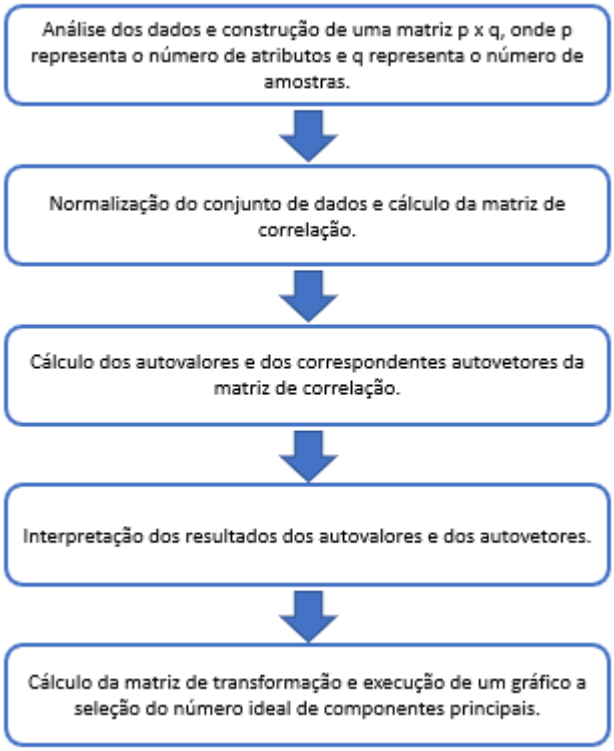

Figura 5 - Esquema das etapas base da técnica Principal Component Analysis.

Resumidamente, a técnica de *Principal Component Analysis* consiste na transformação das características num espaço N-dimensional, preservando a informação das características originais. Esse novo espaço N-dimensional das características é criado através da transformação d-dimensional em k-dimensional através da construção da matriz de correlação. Essa matriz é construída através da seleção dos K autovetores mais importantes, na qual os autovetores mais importantes são selecionados através da reordenação dos autovalores de forma decrescente da sua variância. Estes autovetores e autovalores são criados através da decomposição da matriz de correlação das características originais do conjunto de dados.

Quanto à seleção do número de componentes principais, existem 3 critérios que ajudam na escolha do número de componentes principais ideal, nomeadamente: o critério de *Pearson*, o critério de *Kaiser* e o *Scree Plot*. O Critério de *Pearson* (Pearson, 1901), também conhecido pela regra dos 80%, em que se calcula as matrizes de covariância e de correlação sendo que o número de componentes principais é escolhido até se preservar no mínimo 80% das características iniciais. O Critério de *Kaiser* (Kaiser, 1960) é utilizado com a matriz de correlação. Neste caso são escolhidas as componentes cujo valor próprio seja superior a 1. Este critério é utilizado quando o número de características é inferior ou igual a 30 (Moreira, 2007).

O Critério do *Scree Plot* ou teste de *Cattel* (Cattell, 1966), é um critério gráfico no qual os pontos de maior declive indicam o número de componentes. Através desta análise gráfica visualiza-se as componentes que apresentam maiores autovalores, demonstrando assim as componentes que apresentam uma maior variabilidade. Este critério é utilizado quando o número de características é superior a 30 (Moreira, 2007).

Algumas das vantagens da técnica *Principal Component Analysis* são:

- Remoção das características correlacionadas.
- Melhoria o desempenho do algoritmo de *Machine Learning*, devido á redução do tempo de execução.
- Redução do *overfitting*.

No entanto, esta técnica apresenta algumas desvantagens tais como:

- As variáveis independentes podem tornar-se menos interpretáveis, pois a técnica *Principal Component Analysis* reduz as variáveis em componentes.
- A ocorrência de perda de informação, caso não se selecione o número certo de componentes;
- As variáveis devem estar normalizadas antes de ser aplicado a técnica *Principal Component Analysis*.

3.4.2 Normalização

A normalização é uma das técnicas mais utilizadas em *Machine Learning*. Segundo Han, Kamber, & Pei (2012) a unidade na qual cada atributo se apresenta num determinado conjunto de dados pode afetar a análise de dados. Isto é, um atributo que possua unidades mais pequenas conduzirá a uma maior importância, apresentando, assim, no processo um maior peso, uma maior influência. Para ajudar a evitar a dependência entre atributos com unidades diferentes, algumas técnicas de normalização são aplicadas para fazer com que todos os atributos, dentro do conjunto de dados, tenham igual peso. Essas normalizações são transformações que podem ocorrer entre os seguintes intervalos: [-1,1] e [0,1].

A normalização abordada no presente trabalho foi a normalização Min-Max, o que significa que esta pertence ao intervalo entre [0,1]. Esta efetua uma transformação linear sobre o conjunto de dados original, ou seja, supondo que o mínimo de A e o máximo de A são os valores mínimos e máximos do atributo A. Assim, a normalização Min-Max irá mapear um valor, vi do atributo A para um novo atributo A, $v'i$ variando entre o novo Mínimo de A e o novo máximo de A, como mostra a equação na figura 6.

$$v'i = \frac{vi - minA}{maxA - minA}(new_{maxA} - new_{minA}) + new_{minA}$$

Figura 6 - Equação da normalização Min-Max [0,1].

3.4.3 Otimização dos parâmetros - *GridSearch*

A seleção dos valores ótimos para os parâmetros de um modelo não é uma tarefa fácil. Quando esta é feita de forma manual requer um grande desgaste de recursos computacionais e consumo de tempo. Assim, o método *GridSearch*, é aplicado através da biblioteca *Scikit-Learn* do *Python,* permite a procura dos valores ótimos para um determinado parâmetro. Isto significa que, é necessário criar uma combinação de valores para cada parâmetro que se deseja otimizar - o *GridSearch* irá experimentar e avaliar todas as combinações possíveis para cada parâmetro utilizando o *cross-validation*. De seguida, realiza-se o treino e o teste do modelo com os valores obtidos para cada parâmetro do método *GridSearch* e avalia-se a precisão do modelo.

3.4.4 Cross-validation

O *cross-validation* é um método estatístico de avaliação e de comparação de algoritmos de *Machine Learning*. Este processo tem como objetivo a divisão do conjunto de dados num subconjunto de dados de treino e teste, no qual o subconjunto de dados de treino é utilizado para treinar o modelo e o subconjunto de teste é utilizado para validar o modelo (Bhattacharya, 2014). De acordo Han, Pei, & Tong (2022), o método *cross-validation* consiste na divisão aleatória de um conjunto de dados em K subconjuntos ou *folds* mutuamente exclusivos, por exemplo, em subconjuntos D1, D2,...,DK e cada subconjunto possui um tamanho semelhante. O treino e teste do modelo é realizado K vezes. Na iteração i, a partição Di do conjunto inicial é reservada como conjunto de teste, sendo as restantes partições utilizadas para treino do modelo. Isto é, na primeira iteração são utilizados os subconjuntos {D2, D3, ..., DK}, que são utilizados para treino e, assim, obter-se o primeiro modelo, o qual é testado com recurso ao conjunto D1. Na segunda iteração são utlizados para treino do modelo os subconjuntos {D1,D3,...,DK}, sendo o modelo testado com recurso ao conjunto D2 (Han et al., 2022). Este processo ocorre de forma consecutiva até ser atingido o valor de K estabelecido. A performance do *cross-validation* em cada *fold* pode ser avaliada através da média da precisão em cada subconjunto. Em domínios de *Data Mining* e *Machine Learning* utiliza-se frequentemente o método de *cross-validation* com K=10. A Figura 7 ilustra o processo de *cross-validation*.

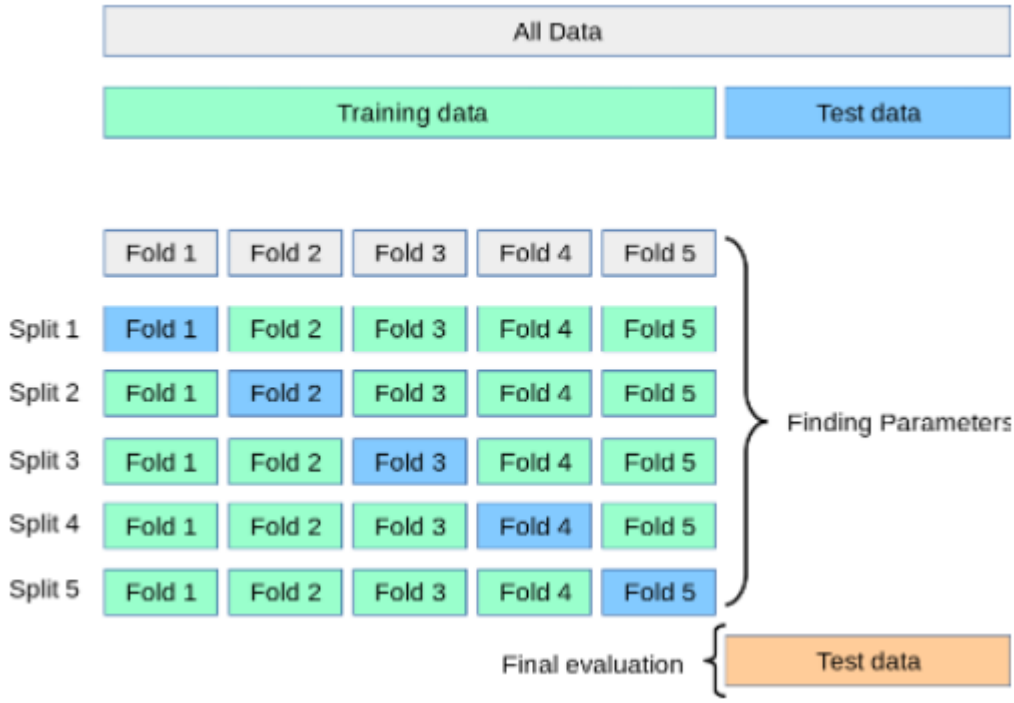

Figura 7 - Ilustração do exemplo de cross validation – imagem extraída de Scikit-learn (2022).

O *Stratified K-fold cross-validation* deriva do método *cross-validation*. Este método assegura que os subconjuntos de dados selecionados apresentam aproximadamente a mesma frequência de valores alocados a cada classe nos seus subconjuntos. Por exemplo, no caso da classificação binária, isto significa que, cada subconjunto tem nas suas classes uma proporção semelhante de casos atribuídos.

3.4.5 Balanceamento dos dados: Técnica *SMOTE* (*Synthetic Minority Oversampling Technique*)

Nem todos os *datasets* disponíveis online ou adquiridos, através de indústrias que disponibilizem os seus dados para estudo de problemas da Ciência dos Dados, apresentam um equilíbrio quanto ao número de casos alocados a cada variável de classificação ou regressão. Portanto, *datasets* (conjunto de dados) desequilibrados tornaram-se num dos problemas mais desafiantes e de reflexão em *data mining*. Deste modo, foram surgindo algumas técnicas com o propósito de colmatar este tipo de problemas e com o intuito de tornar o *dataset* no mais equilibrado possível. Assim, surgiram técnicas de *Undersampling*, *Oversampling* e outras que usufruem da combinação das duas técnicas anteriores.

Quanto às técnicas de *Undersampling*, estas removem elementos da classe maioritária para que sejam o mesmo número de amostras que a classe minoritária, na qual os exemplos podem ser removidos de forma aleatória ou por intermédio de um critério de classificação. A sua principal vantagem é que, caso o *dataset* apresente um elevado número de dados, esta técnica acaba por reduzir o *dataset*. Porém, tem a desvantagem de que se a redução do *dataset* for grande pode ocorrer perda de informação relevante (Tantithamthavorn & Hassan, 2020).

A técnica de *Oversampling*, permite gerar amostras aleatórias, com substituição ou replicação da classe maioritária. Apresentando, assim, como vantagem, o facto de não levar à perda de informação relevante do *dataset* original. Como adiciona casos replicados ao *dataset* original, apresenta como desvantagem a fomentação de redundância do conjunto de dados de treino, pois obtém-se módulos semelhantes, impulsionando o *overfitting* do modelo. Relativamente à conjugação das duas técnicas podem ser aplicadas da seguinte forma: em primeiro lugar, aplica-se a técnica de *oversampling*, pois cria dados artificiais duplicados e, de seguida, aplica-se da técnica *undersampling* com o intuito de remover os dados artificiais desnecessariamente gerados.

A técnica de *SMOTE* (*Synthetic Minority Oversampling Technique*) é uma técnica de *oversampling* proposta por Chawla, Bowyer, Hall e Kegelmeyer (2002), que permite criar dados artificiais tendo por base as semelhanças dos casos da classe minoritária. Assim, para cada caso da classe minoritária existente no conjunto de dados de treino, a presente técnica considera a execução das seguintes etapas (Tantithamthavorn & Hassan, 2020):

1 Cálculo dos K-vizinhos mais próximos.

2 Seleção dos N casos da classe maioritária, com base na menor magnitude das distâncias euclidianas obtidas pelos K-vizinhos mais próximos.

Por fim, o SMOTE combina a amostragem sintética dos casos da classe minoritária com a subamostra dos casos da classe maioritária (Figura 8).

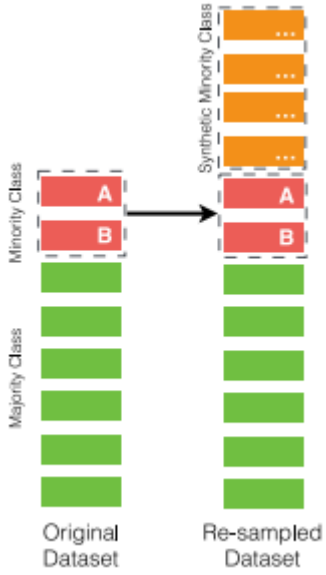

Figura 8 - Técnica SMOTE. Adaptado de Tantithamthavorn e Hassan (2020).

4. ALGORITMOS DE CLASSIFICAÇÃO

A Classificação é uma área de elevada relevância na aprendizagem supervisionada, na qual incidem um elevado número de problemas de *Machine Learning*. A classificação reconhece classes rotuladas dentro de um conjunto de dados e, quando possível, retira conclusões sobre a forma como essas classes devem ser rotuladas. Os algoritmos de classificação mais comuns são: *Logistic regression*, *Support Vector Machine* e *Random Forest.* Vejamos em que consiste cada um deles.

4.1 Random Forest

O algoritmo *Random Forest* é um tipo de algoritmo de *Machine Learning* supervisionado que se baseia na aprendizagem através de conjuntos (Jin et al., 2020). Este algoritmo foi desenvolvido por Breiman (2001) e baseia-se no desenvolvimento de várias árvores de decisão, o que se designa como floresta aleatória em que o resultado final é calculado através da média dos resultados obtidos por cada árvore de decisão.

Numa primeira fase, o algoritmo constrói cada árvore de decisão utilizando uma amostra aleatória do conjunto de dados de treino. De seguida, o algoritmo faz a seleção da melhor característica para o nó de divisão de raiz e gera nós filhos com mini árvores de decisão. Este processo ocorre iterativamente até se atinja o número de árvores desejado. Cada mini árvore testa os dados aleatoriamente presentes em cada um dos seus ramos. Depois, a classificação é realizada para o conjunto de dados de teste através da votação maioritária de cada árvore de decisão. Na Figura 9 está ilustrado, de uma forma sucinta, a forma como atua um algoritmo *Random Forest*.

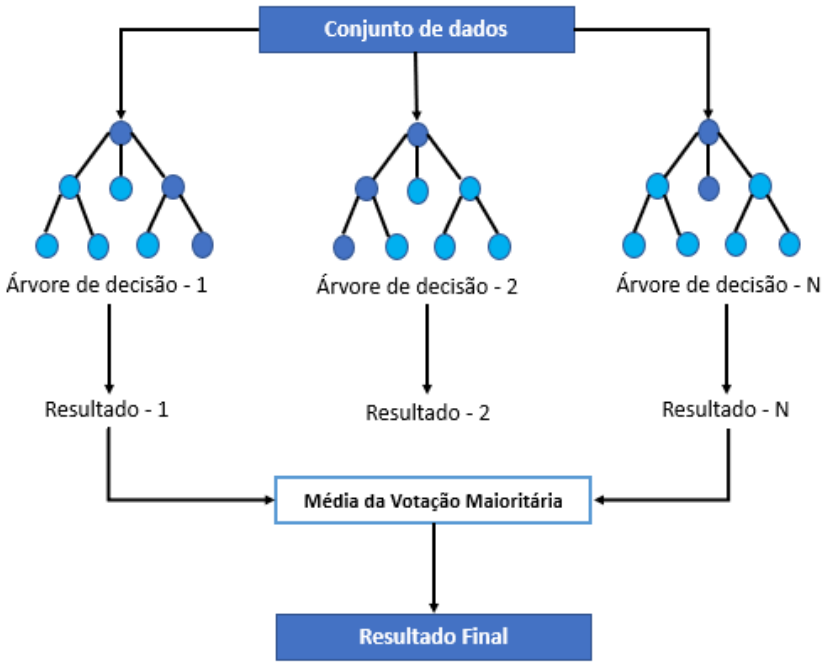

Figura 9 - Diagrama geral do algoritmo Random Forest.

No que diz respeito aos parâmetros, que constituem este algoritmo existem alguns que são importantes de referir, devido ao seu impacto no estudo realizado desta dissertação. De entre todos os parâmetros destacamos os seguintes:

- *Max_depth*: define a profundidade máxima de cada árvore de decisão.
- *Min_sample_split*: fornece a indicação à arvore de decisão sobre o número mínimo de observações exigido num determinado nó de divisão; o valor predefinido para este parâmetro é 2.
- *Max_leaf_nodes*: estabelece uma condição no que diz respeito á divisão dos nós na árvore de decisão, restringindo o crescimento da árvore.
- *Min_samples_leaf*: especifica o número mínimo de amostras que devem estar presentes no nó da extremidade após a divisão de um nó; o valor predefinido para este parâmetro é 1.
- *N_estimators*: define o número de estimadores (árvores de decisão) utilizadas pelo algoritmo *Random Forest*.
- *Max_sample:* representa a fração do conjunto de dados original que é dado a cada árvore de decisão individual.
- *Max_features:* representa o número de características que serão utilizadas em cada árvore de decisão; serão selecionadas diferentes características para cada árvore com o propósito destas serem completamente diferentes.

- ***Bootstrap:*** medida estatística que permite definir se é utilizado todas amostras do conjunto de dados ou se apenas utilizam uma parte das amostras.
- ***Criterion:*** mede a qualidade da construção da árvore de decisão.

Deve-se ainda referir que este algoritmo pode ser aplicado em problemas de classificação e de regressão. Para a implementação deste algoritmo foi utilizado a linguagem *Python, em particular* a biblioteca *Scikit-Learn* (Scikit-Learn biblioteca, 2022). Na Tabela 9 apresentamos algumas das vantagens e desvantagens deste algoritmo.

Tabela 9 - Vantagens e desvantagens do algoritmo Random Forest.

Vantagens	Desvantagens
• Permite tratar um grande volume de dados, o que demonstras uma maior capacidade para tratar dados que apresentam valores em falta. • Comparativamente a outros algoritmos de classificação, devolve uma precisão mais elevada. • Apresenta a capacidade de automaticamente equilibrar os dados do *dataset* introduzido.	• Exige um maior recurso a nível computacional. • Em termos de tempos de execução consome mais tempo, comparativamente ao algoritmo *"Decision Tree"*.

4.2 Logistic Regression

O algoritmo *Logistic Regression* é um método do tipo *Supervised Learning, que* é frequentemente utilizado em problemas de classificação. Este é um algoritmo robusto e flexível na classificação dicotómica, ou seja, é utilizado para prever um resultado binário como por exemplo {sim, não}, {ocorrerá/não ocorrerá} (Seufert, 2014). Este algoritmo também pode ser aplicado em problemas de regressão, como, por exemplo, a previsão do preço de uma casa. O algoritmo *Logistic Regression* é adequado quando a probabilidade da variável a ser classificada pertence a um intervalo binário 0 e 1. Deste modo, neste tipo de modelo, a variável dependente irá adquirir o valor 1 se o seu resultado for "Sucesso" e o valor 0 quando o resultado for "Insucesso". Contudo, para se colocar a variável dependente a variar entre 0 e 1, recorre-se à função *logit* para realizar essa transformação (Figura 10).

$$\log\left(\frac{\pi}{1-\pi}\right) = \beta 0 + \beta 1 x 1 + \beta 2 x 2 + \cdots + \beta m x m$$

Figura 10 - Equação da função Logit.

Em termos gráficos, o modelo *logistic regression* consiste numa transformação não linear do próprio modelo de regressão linear, sendo representada por uma curva em forma de S (Figura 11).

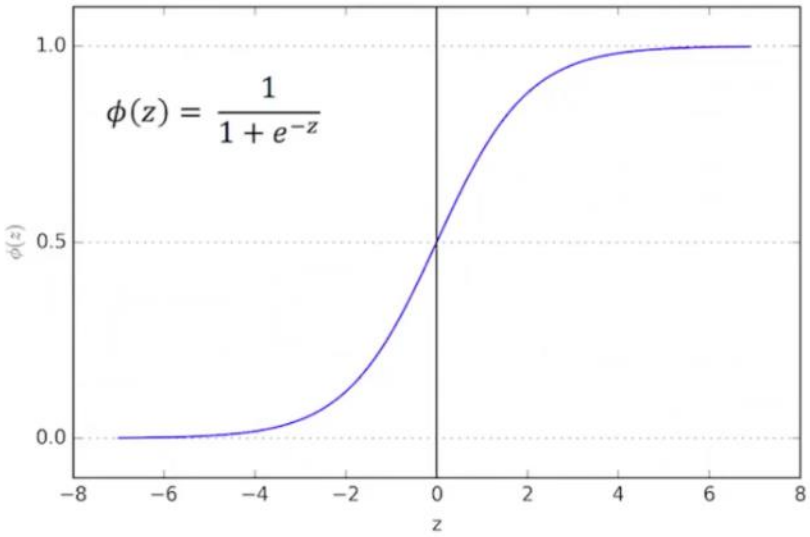

Figura 11 - Exemplo do Logistic regression. Adaptado de Remanan (2018).

Na Tabela 10 apresentamos algumas das vantagens e desvantagens deste algoritmo.

Tabela 10 - Vantagens e Desvantagens do algoritmo Logistic Regression.

Vantagens	Desvantagens
• Fácil implementação, pois é um algoritmo que não necessita de um elevado poder de computação e é simples de atualizar. • Apresenta resultados bem calibrados e em relação a outros algoritmos é menos propenso a *overfitting* quando o utilizado um conjunto de dados de baixa dimensão. • Apresenta uma boa precisão, sendo que o seu desempenho é ainda melhor quando o conjunto de dados tem características linearmente separáveis.	• Quando o conjunto de dados é elevada dimensão impulsiona o *overfitting*, sendo que o parâmetro regularização ajuda a prevenir o mesmo. No entanto se o parâmetro regularização for muito elevado pode resultar num modelo *underfitting* com resultados imprecisos; • Requer um elevado número de amostras, isto significa que o conjunto de dados deve ter um maior número de amostras quando comparado com o número de características. Pois caso contrário poderá levar a um modelo com *overfitting*. • Requer um pré-processamento de dados mais elevado.

4.3 Support Vector Machine

O modelo *Support Vector Machine* foi introduzido por Vapnik (1998) e deriva da teoria da aprendizagem estatística. Este algoritmo é do tipo aprendizagem supervisionada e o qual é aplicado em problemas classificação e regressão. No algoritmo *Support Vector Machine*, o objetivo é encontrar um hiperplano num espaço N-dimensional, no qual N representa o número de características. Os hiperplanos representam os limites de decisão que ajudam na classificação das características. Existem vários hiperplanos que podem ser escolhidos para a separação das classes. O principal objetivo do algoritmo *Support Vector Machine* é descobrir o melhor hiperplano que separe as duas classes, maximizando a distância entre a margem e os vetores de suporte. É através desta maximização que se torna possível que sejam classificadas futuras características com maior confiança. Os vetores de suporte são os pontos que correspondem às características que estão mais próximas do hiperplano de separação entre as classes. A Figura 12 ilustra o processo de *Support Vector Machine*.

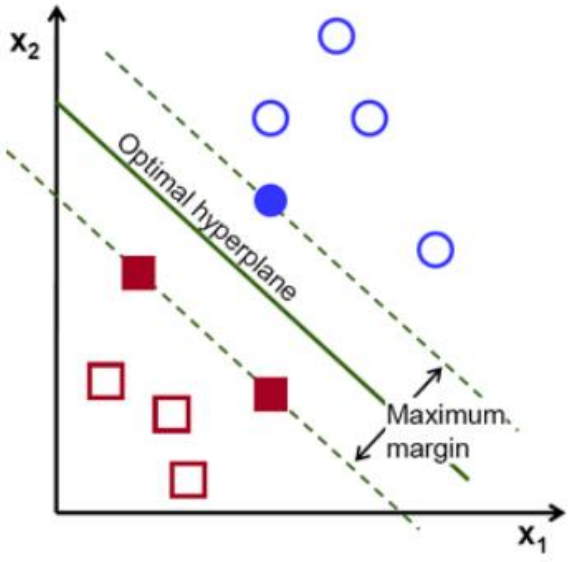

Figura 12 - Visão geral do SVM no espaço bidimensional. Adaptado de Rohith Gandhi (2018)

A performance deste algoritmo é afetada pelos parâmetros que o *Support Vector Machine* possui e que permitem encontrar o melhor hiperplano que separa as características. O algoritmo *Support Vector Machine* assenta em três parâmetros fundamentais. O parâmetro de regularização, também conhecido como "C", informa a otimização do algoritmo do SVM, o quanto é necessário evitar classificação errada de cada exemplo presente em cada conjunto de dados. Assim, quanto maior for o "C" o hiperplano escolhido terá uma margem menor, o que se evidenciará uma baixa taxa de erros relativos à classificação dos dados de treino. Em contrapartida, um baixo valor de "C" apresentará um hiperplano com uma margem maior, devolvendo um maior número de classificações erradas no conjunto de dados de treino.

Outro parâmetro importante é o "*Gamma*", o qual define a curvatura do limite de decisão. Assim, um valor elevado neste parâmetro apresenta uma baixa curvatura do limite de decisão, enquanto um valor baixo neste parâmetro apresenta uma elevada curvatura do limite de decisão. O parâmetro "*Gamma*" está diretamente relacionado com o parâmetro "*Kernel*". O parâmetro "*Kernel*" ajuda na transformação do hiperplano numa dimensão superior, sem que aumente a sua complexidade. Existem quatro tipos de "*Kernel*": *linear*, *rbf*, *poly* e *sigmoid*.

Na Tabela 11 apresentamos algumas das vantagens e desvantagens deste algoritmo.

Tabela 11 - Vantagens e Desvantagens do algoritmo Support Vector Machine.

Vantagens	Desvantagens
• Fácil execução, principalmente quando se verifica de forma clara a margem de separação entre as classes. • É eficiente nos casos em que o número de dimensões é maior do que o número de amostras. • Em termos de memória é relativamente eficiente.	• Para um grande conjunto de dados, apresenta um tempo de treino longo. • A escolha do melhor canal para um determinado conjunto de dados é difícil de selecionar. • Para um grande volume de dados não é o mais eficiente.

4.4 Métricas de avaliação

Após a implementação de um algoritmo de *Machine Learning* é fundamental medir o desempenho do modelo, o que servirá como indicador da qualidade das previsões e dos resultados obtidos.

4.4.1 Matriz de confusão

A matriz de confusão é uma medida de desempenho frequentemente usada em problemas de classificação, que permite, de uma forma generalizada, visualizar a performance do modelo (Figura 13).

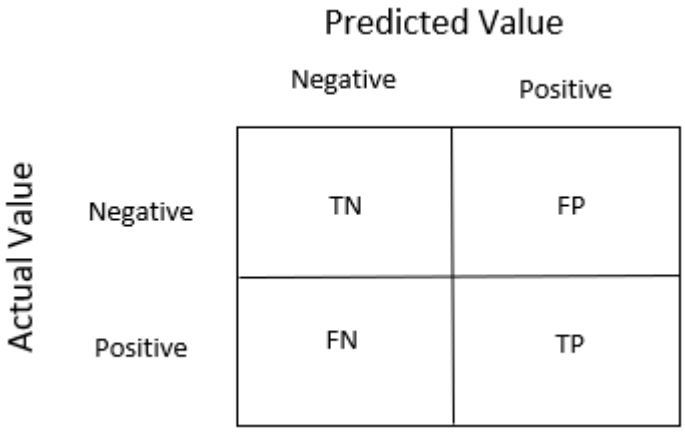

Figura 13 - Exemplo de uma Matriz de confusão.

Da análise da matriz de confusão retira-se informação do número de *True Negative* (TN), *True Positive* (TP), *False Negative* (FN) e *False Positive* (FP), que representam, respetivamente, o número de casos que o modelo prediz corretamente a classe negativa, o número de casos que o modelo prediz corretamente a classe positiva, o número de casos que o modelo prediz incorretamente a classe negativa e o número de casos que o modelo prediz incorretamente a classe positiva. Através da informação retirada da matriz de confusão são calculadas as métricas de avaliação de um modelo, como a *accuracy, precision*, *recall* e *F1-Score.* Estas métricas serão abordadas nas secções seguintes.

4.4.2 Precisão

A precisão é uma das métricas de avaliação mais utilizada em problemas de classificação, que é calculada através da seguinte fórmula:

$$Precisão = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FP} + \mathrm{FN}}$$

No entanto quando o conjunto de dados utilizado é desequilibrado, ou seja, quando o número de observações varia nas diferentes classes, é uma métrica que pode produzir um resultado enganador. Sabendo isso, deve-se consultar as restantes métricas baseadas na matriz de confusão*, uma vez* que podem ser úteis para avaliar o desempenho do modelo.

4.4.3 Classification Report

O *Classification Report* é utilizado para medir a qualidade das previsões do modelo. Este pode incluir as seguintes métricas:

- ***Precisão*** – de entre todas as classes positivas previstas, a precisão mostra quantas das classes é que são realmente positivas. Esta métrica é calculada através da seguinte fórmula:

$$Precisão = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}$$

- ***Recall*** **-** de entre todas as classes positivas previstas, mostra quantas são realmente positivas**.** Esta métrica é calculada através da seguinte fórmula:

$$Recall = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}$$

- ***F1 Score*** **–** é uma média harmónica entre a *precision* e a *recall,* que varia entre [0,1]. Quanto maior for o F1 Score melhor é o desempenho do modelo. A métrica F1 Score é calculada através da seguinte fórmula:

$$F1\ Score = 2 * \frac{Precision * Recall}{\mathrm{Precision} + \mathrm{Recall}}$$

- ***Support*** – é o número de amostras que foram incorretamente classificadas como verdadeiras.
- ***Macro Average*** – é a média aritmética de todos os recall scores para as diferentes classes.
- ***Micro Average*** – Incorpora todas as contribuições de todas as classes para calcular a métrica F1 Score através do somatório de Verdadeiros Positivos, Falsos Negativos e Falsos Positivos. Esta métrica é mostrada apenas quando se têm mais do que duas classes pois, caso contrário, corresponde à precisão. Esta é uma métrica importante quando há desequilíbrios nas classes.

4.4.4 Receiver Operating Characteristics (ROC)

A curva AUC-ROC é uma das métricas mais prestigiadas e utilizadas para avaliar o desempenho de modelos de classificação. Esta permite comparar diferentes classificadores e definir qual o melhor classificador com base nos diferentes pontos de corte. A curva ROC é uma métrica representada graficamente e traçada em função da sensibilidade (TPR-*True Positive Rate*), presente no eixo do y, e da especificidade (*FPR-False Positive Rate*), presente no eixo do x (Figura 14).

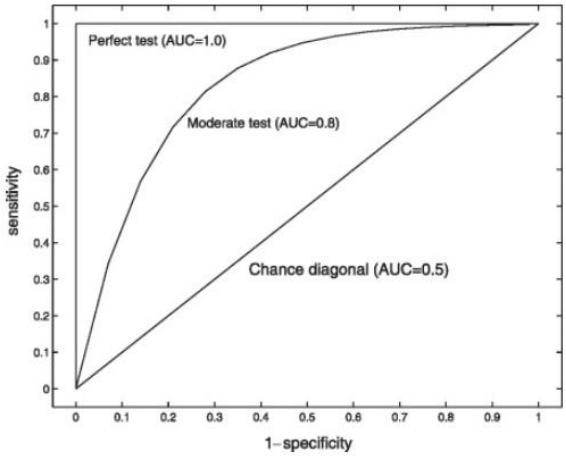

Figura 14 - Curva AUC-ROC. Adaptado de Swets (2001)

A AUC calcula a área da forma bidimensional abaixo da curva de ROC, representando o grau ou medida de separabilidade, e demonstrando quanto o modelo é capaz de diferenciar as duas classes. A curva ROC representa apenas uma curva de probabilidade.

Quanto maior a AUC, melhor é a capacidade de o modelo prever corretamente cada classe. Um ótimo modelo apresenta uma AUC próxima de 1. Esse valor indica que o modelo classifica corretamente todas as classes e, por conseguinte, diferencia corretamente as classes. Um modelo com baixo desempenho apresenta uma AUC próxima de 0, isto é o modelo não é capaz de separar as classes corretamente. No caso em que o modelo apresenta uma AUC próximo de 0.5, significa que o modelo não têm capacidade para separar as respetivas classes.

5. O CASO DE ESTUDO

A Parkinson é uma doença caracterizada como neurodegenerativa e progressiva que se desenvolve com o avançar da idade, afetando a função motora e cognitiva (El-Turabi & Bachmann, 2018). Várias pesquisas têm sido realizadas face ao desenvolvimento da doença, originando ferramentas de apoio ao diagnóstico e acompanhamento contínuo do doença, como, por exemplo, a análise de sinais biomédicos associados á pessoa com a doença (Amato, Borzì, Olmo, & Orozco-Arroyave, 2021). Em termos de metodologias de deteção automática desta doença tem sido utilizada as características vocais do ser humano, através de testes de fonética. A disformidade da fala permite-nos obter informação significativa sobre o estado da doença, permitindo a deteção precoce e acompanhamento da doença.

5.1 Levantamento dos dados

O *dataset* que utilizámos neste trabalho foi obtido num repositório da Universidade da Califórnia *Irvine "UCI Machine Learning Repository"*. Este dataset foi sendo criado por Sakar et al. (2019), que realizou um estudo comparativo de vários algoritmos para classificação da doença de Parkinson através do processamento de sinal.

O *dataset* escolhido é constituído por dados recolhidos de 188 pacientes com Parkinson, dos quais 107 correspondem ao sexo masculino e 81 ao sexo feminino, com idades compreendidas entre os 33 e os 87 anos. Além disso, contêm um grupo de controlo de 64 indivíduos saudáveis, dos quais 23 são do sexo masculino e 41 são do sexo feminino, com idades compreendidas entre os 41 e 82 anos. Os dados foram recolhidos pelo Departamento de Neurologia da Faculdade de Medicina da Universidade de Istambul (İstanbul Medipol Üniversitesi, 2018). O processo de recolha dos dados foi realizado durante um exame médico, no qual foi utilizado um microfone com uma frequência de 44,1 KHz, realizado com 252 indivíduos que pronunciaram 3 vezes a vogal "a" (C. O. Sakar et al., 2019). Dos dados disponíveis extraíram-se 6 grupos de parâmetros, nomeadamente "*Baseline Features*", "*Time Frequency Features*, "*Mel Frequency Cepstral Coefficients*", "*Wavelet Transform based Features*", "*Vocal Fold Features*" e "*Tunable Q-factor wavelet transform Features*", através do software Praat (BOERSMA & P., 2011). Na Tabela 12 podemos ver esses parâmetros, bem com o número de variáveis que cada parâmetro contém.

Tabela 12 - Descrição das variáveis do conjunto de dados.

Features		
Baseline Features	***Number of Features***	***Brief Description of the Features***
Jitter Variants	5	Representa a variação fundamental da frequência a partir de um ciclo periódico para o próximo ciclo. Os valores desta variável mudam consoante a desordem da voz, isto significa que é responsável por uma qualidade de voz rouca.
Shimmer Variants	6	Representa a variação de amplitudes de períodos consecutivos.
Fundamental Frequency Parameters	5	Representa a frequência da vibração das cordas vocais.
Harmonicity Parameters	2	Representa o ruído causado pelo parcial fecho das cordas vocais.
Recurrence Period Density Entropy (RPDE)	1	Representa a capacidade de as cordas vocais sustentarem oscilações constantes e quantifica os desvios F0.
Detrended Fluctuation Analysis (DFA)	1	É utilizado para avaliar a similaridade do ruído produzido por um fluxo de ar nas cordas vocais.
Pitch Period Entropy (PPE)	1	Estima o controlo da frequência fundamental F0 utilizando uma escala logarítmica.
Time Frequency Features		
Intensity parameters	3	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.
Formant frequencies	4	Representam as frequências amplificadas pelo trato vocal.
Bandwidth	4	Representa a diferença entre a frequência superior e inferior numa banda contínua de frequências.
Mel-Frequency Cepstral Coefficients (MFCCs)		
MFCCS	84	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.
Wavelet transform-based Features		
Wavelet Transform-Based Features	182	Permite analisar sinais oscilatórios da transformada discreta de wavelet.
Vocal Fold Features		
Glottis quotient (GQ)	3	Fornece informações sobre a duração da abertura e fecho da glottis. Sendo representada como uma medida de periodicidade nos movimentos da glottis.
Glottal to noise excitation (GNE)	6	Quantifica a extensão do ruído causado pelo fecho incompleto das cordas vocais, no sinal da fala
Vocal fold excitation ratio (VFER)	7	Quantifica a quantidade de ruído produzido devido á vibração patológica das cordas vocais, recorrendo a conceitos de energia não-linear e entropia.
Empirical mode decomposition (EMD)	6	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.
Tunable Q-factor wavelet transform		
TQWT Features	432	Procede à decomposição do sinal EMG em subfaixas e estas são utilizadas para a extração de características estatísticas.

5.2 Abordagem realizada

O *dataset* é constituído por 756 características, num total de 754 registos, com 3 registos por indivíduo. Assim, temos 252 indivíduos em análise no presente estudo. De seguida, verificou-se que tipo de variáveis o *dataset* continha, verificando-se se eram do tipo ***int64*** e do tipo ***float64*** (Figura 15).

```
df_parkinson.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 15 - Tipo de variáveis presentes no conjunto de dados.

Na figura 16 é possível visualizar o número de casos atribuídos a cada classe. Os casos atribuídos à classe 0 pertencem todos a indivíduos (192) que não apresentam a doença de Parkinson. Os casos atribuídos à classe 1 pertencem a indivíduos (564) que apresentam a doença de Parkinson. Também podemos ver que o *dataset* apresentado está bastante desequilibrado.

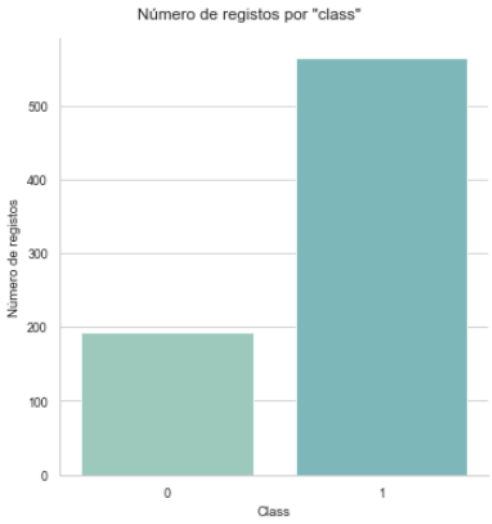

Figura 16 - Número de casos de Parkinson.

Analisando a variável "*gender*" (Figura 17), uma variável é do tipo binária, relativa ao género de cada individuo, vemos que ao valor 0 correspondem todos os indivíduos do género feminino e ao valor 1 correspondem todos os indivíduos do género masculino. Deste modo, contamos com 366 indivíduos do

género feminino e 390 indivíduos do género masculino. Ao nível do género o número de amostras atribuído a cada variável está equilibrado.

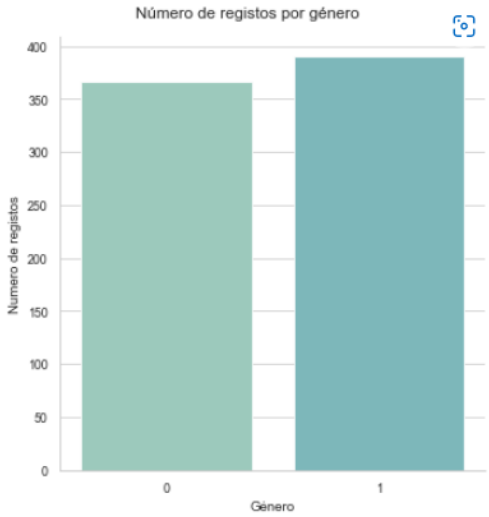

Figura 17 - Número de indivíduos por género.

Durante o processo de análise verificámos também se existiam ou não valores em falta (Figura 18). Comprovámos que todos os registos se apresentaram devidamente preenchidos.

```
df_parkinson.isnull().sum()

id                          0
gender                      0
PPE                         0
DFA                         0
RPDE                        0
                           ..
tqwt_kurtosisValue_dec_33   0
tqwt_kurtosisValue_dec_34   0
tqwt_kurtosisValue_dec_35   0
tqwt_kurtosisValue_dec_36   0
class                       0
Length: 755, dtype: int64
```

Figura 18 - Verificação da existência de valores em falta.

No que toca à correlação das variáveis, foram removidas todas as variáveis com uma correlação superior a 0.9, o que resultou num *dataset* com 432 variáveis. Esta variáveis foram utilizadas no seguimento do estudo. Posteriormente, a variável "id" do conjunto de dados foi removida pois apenas identifica cada registo, o que não provoca qualquer impacto no estudo. De seguida, realizámos o pré-

processamento de dados. Neste processo atribuímos à variável "Y" todos os casos da variável "*class*" na qual estão os indivíduos que têm a doença de Parkinson e os indivíduos que não têm a doença de Parkinson. No entanto, à variável "X" foram atribuídas as restantes características do conjunto de dados que caracterizam esta doença. Depois destas ações, passámos à normalização das variáveis, aplicando a normalização "*MinMaxScaler*", com o objetivo de colocar todas as variáveis na mesma escala – 0 e 1. Após a normalização, foi aplicada a técnica de *Principal Component Analysis*, como técnica de redução do número de características. Quando a técnica *Principal Component Analysis* é utilizada, a principal dificuldade é a seleção do número correto de componentes em que para essa seleção foi utilizado o critério *Screet Plot* em que um gráfico foi traçado para uma variância acumulada de 97% o número ideal de componentes são 150 (Figura 19).

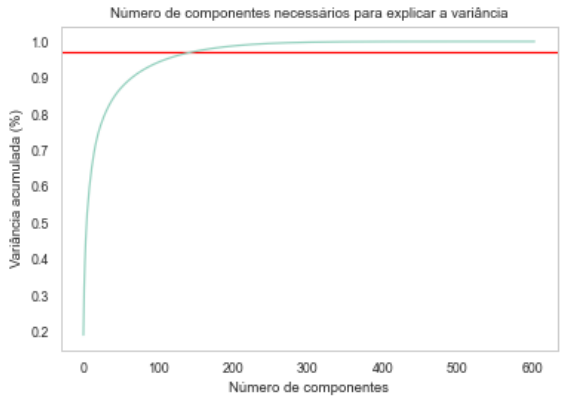

Figura 19 - Número de componentes principais para explicar a variância do dataset.

De seguida, dividimos o *dataset* em dois subconjuntos: um de treino e um de teste. Através deste processo, é possível garantir que o modelo tenha um melhor desempenho no caso de ocorrerem dados desconhecidos. Assim, o subconjunto de dados de treino contém as respostas certas, de forma que o modelo possa aprender com esses dados e que apresente uma generalização para dados posteriores. Quanto ao subconjunto com os dados de teste, este tem como objetivo permitir a verificação de quanto o modelo é consistente, em termos da sua precisão. Assim, ao subconjunto de treino foi atribuído 80% dos dados do conjunto inicial e ao subconjunto de teste foi atribuído 20% do conjunto de dados inicial. Esta implementação foi realizada no software *Python* e foram utilizadas as seguintes bibliotecas: *sklear.model_selection* e *sklearn.train_test_split.*

A divisão do conjunto de dados no subconjunto de dados de treino e de teste foi parametrizada de forma que o número de casos pertencentes à classe que se pretendia classificar mantivesse o número de casos. Em paralelo à etapa de divisão de treino e teste, foi implementado a técnica de *cross-validation* como de medida de complementar de validação dos resultados de precisão obtidos. Como o conjunto de dados era desequilibrado selecionou-se a técnica particular de *cross-validation StratifiedKFold*

Validation pois reorganiza o conjunto de dados de forma que cada subconjunto tenha uma boa representatividade como um todo do conjunto inicial. O único parâmetro a configurar nesta técnica é o K em que selecionamos o valor 10. Terminada a divisão do conjunto de dados, seguiu-se a etapa de implementação do algoritmo de *Machine Learning.* Para isso precisámos de definir os valores dos parâmetros. Estes são específicos a cada algoritmo, pelo que não podem ser calculados a partir do *dataset*. Assim, diferentes parâmetros produzirão diferentes valores. Usualmente, os valores dos parâmetros podem ser configurados quase de forma aleatória e, de seguida, podemos avaliar se os valores apresentam um melhor desempenho. No entanto, essa forma de selecionar os parâmetros é uma tarefa bastante exaustiva, na medida em que é necessário comparar o desempenho dos algoritmos com os diferentes valores dos parâmetros. Deste modo, em vez dos parâmetros serem selecionados de forma aleatória, recorremos à implementação de um processo de seleção automática dos melhores valores para cada algoritmo em específico. Para tal, recorreu-se a um método de "*Fine Tuning*" denominado por "*GridSearch*". O "*GridSearch*" permite criar um dicionário com todos os parâmetros e com os respetivos conjuntos de valores alocados a cada um deles, que irão ser testados para que se possa obter um melhor desempenho do modelo. Para realizar esta ação utilizámos o método "*GridSearch*" da biblioteca *sklearn.model_selection.*

Por fim, para verificarmos se um indivíduo apresenta ou não a doença de Parkinson utilizámos os seguintes algoritmos:

- Classificador *Random Forest*, utilizando a biblioteca *sklearn.ensemble* e o método *RandomForestClassifier*.
- Classificador *Logistic Regression*, utilizando a biblioteca *sklearn.linear_model e* o método *LogisticRegression.*
- Classificador *Support Vector Machine*, utilizando a biblioteca *sklearn.svm e* os métodos *SVC e LinearSVC.*

5.3 Análise dos resultados

Tal como referimos anteriormente, aplicámos os algoritmos *Random Forest*, *Logistic Regression* e *Support Vector Machine* para fazer a classificação do número de indivíduos que apresentavam ou não a doença de Parkinson. Para apresentar os resultados obtidos do estudo adotámos a seguinte estratégia: primeiramente, descreveríamos os resultados obtidos com o *dataset* não balanceado e depois com o *dataset* balanceado.

5.3.1 Primeira abordagem dataset não balanceado

Todos os algoritmos são inicialmente treinados e testados sem os seus parâmetros estarem configurados. De seguida, volta-se a treinar e a testar o algoritmo com os parâmetros configurados em que se aplica a técnica *GridSearch* para selecionar o melhor valor para cada um. O primeiro algoritmo a ser aplicado foi o *Random Forest* e no qual se configuraram os seguintes parâmetros: *n_estimators, max_features, max_depth, criterion, bootstrap* em que ao parâmetro *n_estimators* foram atribuídos os seguintes valores [10, 20, 30, 40, 50, 100] e a técnica *GridSearch* selecionou o valor 100. O parâmetro *max_features* foram atribuídas as funções auto, sqrt e log2 e a que foi selecionada foi a auto. De seguida, o parâmetro *max_depth* também foi configurado com o intervalo de valores [2, 5, 6, 7, 8] e o valor selecionado para este parâmetro foi o 8. Quantos aos parâmetros *criterion* e *bootstrap*, o *criterion* foi o configurado com o critério *Gini* e *Entropy* e foi selecionado o critério *Gini*, o parâmetro *bootstrap* também foi configurado com dois critérios o *True* e *False* e o parâmetro selecionado foi o *false*.

Através da combinação dos parâmetros selecionados pela técnica "*GridSearch*", realizámos o treino e o teste do modelo desenvolvido, obtendo-se uma *"accuracy"* de 95%, tal como é apresentado na Tabela 13.

Tabela 13 - Classification report do modelo com os dados de teste do modelo Random Forest.

	Precision	Recall	F1-score	Support
0	1.00	0.82	0.90	39
1	0.94	1.00	0.97	113
Accuracy			0.95	152
macro avg	0.97	0.91	0.94	152
weighted avg	0.96	0.95	0.95	152

Na Figura 20 podemos ver uma avaliação dos casos classificados. Através da sua análise concluímos que o modelo previu corretamente 113 casos que apresentavam a doença de *Parkinson* e 30 outros casos que não a apresentavam. Quanto aos casos incorretamente previstos, o modelo identificou incorretamente 9 que apresentavam a doença de Parkinson e nenhum caso que não apresentasse a doença.

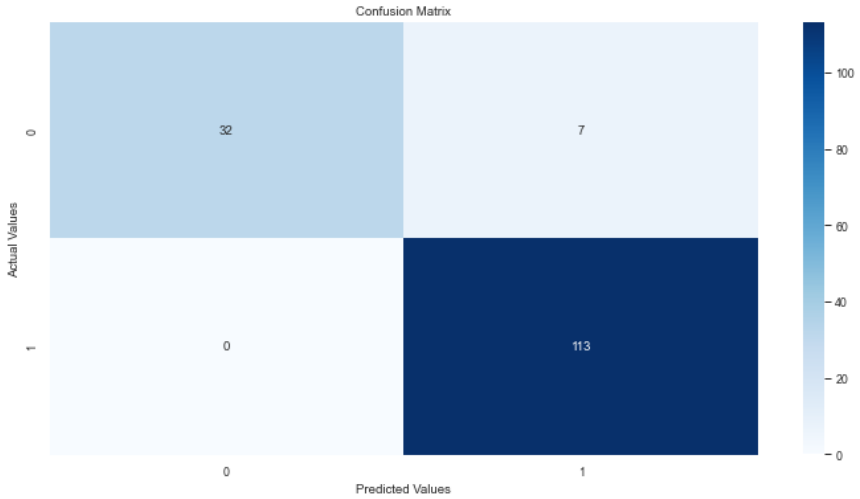

Figura 20 - Matriz de confusão com os resultados obtidos do algoritmo Random Forest.

A seguir, foi aplicado o algoritmo *Logistic Regression* em que os parâmetros selecionados para treinar e testar este modelo foram os seguintes: *penalty*, C, *solver* e o *max_iter*. Em que ao parâmetro *penalty* foram atribuídos dois valores l2 e l1, sendo que o valor selecionado foi o l2. Quanto ao parâmetro C o conjuntos de valores atribuídos foi 0.001, 0.01, 0.1, 1, 10, 100, 1000 e o melhor valor para este parâmetro foi o 1. Para o parâmetro solver que é a função de otimização do algoritmo *Logistic Regression* foram atribuídas duas funções a *liblinear* e *saga*, sendo que a função selecionada como a melhor função para este parâmetro foi a *liblinear.* Por fim, resta o parâmetro *max_iter* foi atribuído apenas o valor de 10000. Após a combinação destes parâmetros no treino e teste do modelo, o modelo obteve uma precisão de 100% (Tabela 14).

Tabela 14 - Classification Report do modelo com os dados de teste do modelo Logistic Regression.

	precision	recall	F1-score	Support
0	1.00	1.00	1.00	39
1	1.00	1.00	1.00	113
accuracy			1.00	152
macro avg	1.00	1.00	1.00	152
weighted avg	1.00	1.00	1.00	152

Quanto ao número corretamente e incorretamente classificados pelo algoritmo *Logistic Regression* a Figura 21, mostra que o modelo previu corretamente 113 casos que apresentavam a doença de *Parkinson e* 39 casos que não apresentavam a doença. Quanto aos casos incorretamente preditos, o modelo não previu incorretamente nenhum caso que apresentasse ou não a doença de Parkinson.

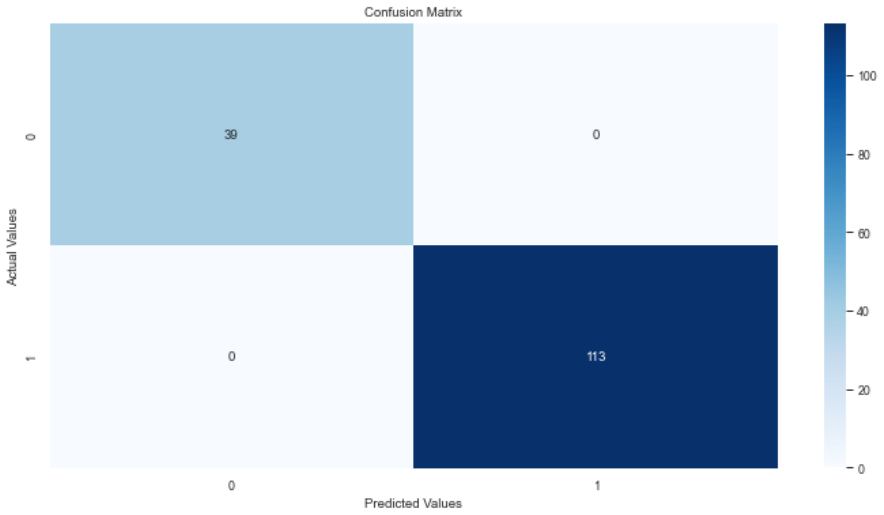

Figura 21 - Matriz de confusão com os resultados obtidos do algoritmo Logistic Regression.

Por fim, foi aplicado o algoritmo *Support Vector Machine* em que foi treinado e testado exclusivamente com o *kernel* linear e o único parâmetro configurado foi o C ao qual se atribuíram os seguintes valores 0.001, 0.01, 0.1, 1, 10, 100, 1000 e o valor selecionado pela técnica *GridSearch* foi 1. A precisão devolvida pelo algoritmo *Support Vector Machine* foi de 99% (Tabela 15).

Tabela 15 - Classification Report do modelo Support Vector Machine com os dados de teste do modelo com Kernel Linear.

	precision	recall	F1-score	Support
0	1.00	0.95	0.97	39
1	0.98	1.00	0.99	113
accuracy			0.99	152
macro avg	0.99	0.97	0.98	152
weighted avg	0.99	0.99	0.99	152

Uma breve análise à Figura 22 permite-nos ver que o modelo previu de forma correta 113 casos com a doença de *Parkinson* e 37 casos sem a doença de *Parkinson*. Por outro lado, o modelo previu incorretamente 2 casos que apresentavam a doença de Parkinson e nenhuma de forma incorreta.

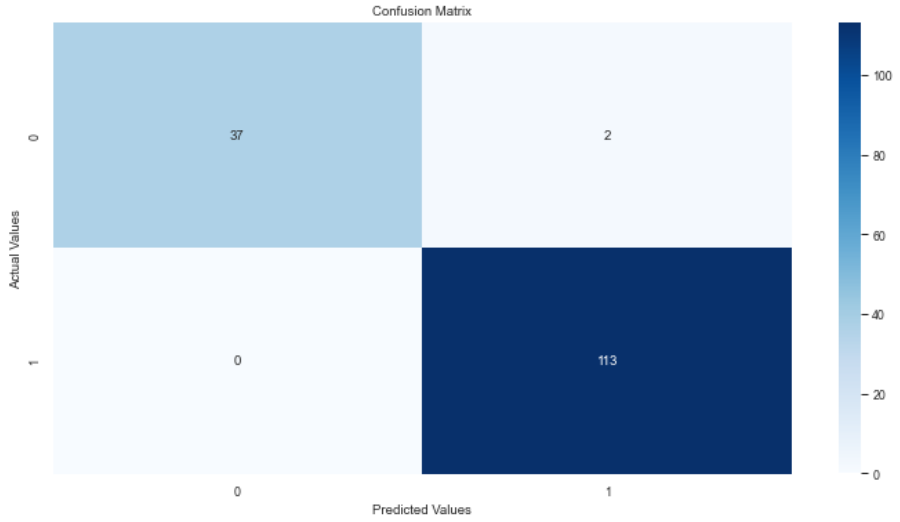

Figura 22 - Matriz de confusão com os resultados obtidos do algoritmo Support Vector Machine com o Kernel Linear.

Seguidamente o algoritmo *Support Vector Machine* foi treinado e testado para os *kernels rbf*, *poly* e *sigmoid* e que a técnica *GridSearch* selecionou o *kernel poly*. E os restantes parâmetros como C, *gamma, tolerance, degree* foram configurados. Ao parâmetro C foram atribuídos os seguintes valores 0.001, 0.01, 0.1, 1, 10, 100 e valor selecionado foi o 0.1, ao parâmetro gamma foram atribuídos os valores 1/753, 1,0.1,0.01,0.001 em que o valor selecionado foi o 1.0. Quanto à tolerância foram atribuídos os valores 1e-3, 1e-4, 1e-5 em que o valor selecionado foi 0.001, por fim como *kernel poly* foi selecionado este tem obrigatoriamente o parâmetro *degree* associado e ao qual foram atribuídos os valores 1, 2, 3, 4, 5 e o valor selecionado foi o 1.0. Após o treino e teste do algoritmo *Support Vector Machine* com esta combinação de parâmetros a precisão devolvida foi de 100% (Tabela 16).

Tabela 16 - Classification Report do modelo com os dados de teste do modelo Support Vector Machine.

	precision	recall	F1-score	Support
0	1.00	1.00	1.00	39
1	1.00	1.00	1.00	113
accuracy			1.00	152
macro avg	1.00	1.00	1.00	152
weighted avg	1.00	1.00	1.00	152

Quanto à representação de casos corretamente e incorretamente classificados a matriz de confusão visualizada na Figura 23, mostra que o modelo previu corretamente 113 casos que apresentam a doença de *Parkinson*. Da mesma forma que o modelo previu corretamente 39 casos que não apresentam a doença de *Parkinson*. Quanto aos casos incorretamente preditos, o modelo não previu incorretamente

nenhum caso que apresentam a doença de Parkinson e não previu incorretamente nenhum caso que não apresente a doença.

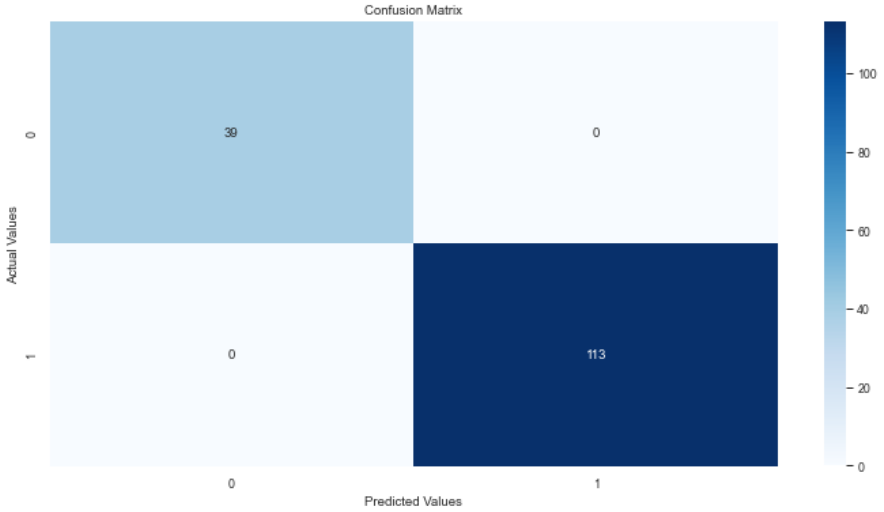

Figura 23 - Matriz de confusão com os resultados obtidos do algoritmo Support Vector Machine.

5.3.2 Segunda Abordagem Dataset Balanceado

A segunda abordagem consistiu na adição de uma técnica de balanceamento do *dataset*. Esta técnica gerou um número igual de amostras para cada uma das classes que se pretendia classificar. Após a aplicação desta técnica, utilizámos o mesmo método que na primeira abordagem: a aplicação da técnica de *Principal Component Analysis*, com o mesmo número de componentes selecionadas (150), a aplicação do algoritmo de aprendizagem e a otimização dos parâmetros dos algoritmos. Na Figura 24 podemos ver o número de registos que ficaram atribuídos a cada classe após a aplicação da técnica de SMOTE, a qual igual gerou registos para serem adicionados à classe minoritária com o objetivo de se partir de um conjunto de dados balanceado, isto é, ambas as classes tem o mesmo número e registos e quando somados os o número de registos da classe 1 com o número de registos da classe 0 obtém-se um total de 1128 registos.

```
print("After OverSampling, counts of label '1': {}".format(sum(y_sm==1)))
print("After OverSampling, counts of label '0': {}".format(sum(y_sm==0)))
```

```
After OverSampling, counts of label '1': 564
After OverSampling, counts of label '0': 564
```

Figura 24 - Aplicação da técnica SMOTE.

Tal como na primeira abordagem, a discussão dos resultados segui a mesma estrutura. Portanto, o primeiro algoritmo aplicado foi o *Random Forest* e os parâmetros configurados foram o *n_estimators, max_features, max_depth, criterion, bootstrap*. Em ao parâmetro *n_estimators* foram atribuídos os

valores 10, 20, 30, 40, 50, 100 e o valor selecionado foi o valor 100, para o parâmetro *max_features* foram atribuídas as funções auto, *sqrt* e logarítmica e a função selecionada foi a Auto. De seguida, seguem-se os parâmetros *max_depth* ao foi atribuído os valores 2, 5, 6, 7, 8 e o valor selecionado foi o 8, ao parâmetro *criterion* foram atribuídos os critérios *Gini* e *Entropy* e o critério selecionado foi o *Entropy*. Por fim, temos o parâmetro *bootstrap* ao qual se atribuíram as condições *True* e *False* e a condição selecionada foi a *True*. Com estes valores selecionados para os parâmetros do algoritmo *Random Forest* este obteve uma precisão de 86% (Tabela 17).

Tabela 17 - Classification Report do modelo Random Forest com os dados de testes com o conjunto de dados balanceado.

	precision	recall	F1-score	Support
0	1.00	0.72	0.83	109
1	0.79	1.00	0.88	117
accuracy			0.86	226
macro avg	0.90	0.86	0.86	226
weighted avg	0.89	0.86	0.86	226

Por conseguinte, foi utilizada a matriz de confusão para a avaliar o número de casos corretamente e incorretamente classificados com a doença de Parkinson. Na Figura 25 é demonstrado que o modelo previu corretamente 117 casos que apresentam a doença de *Parkinson*. Da mesma forma que o modelo previu corretamente 78 casos que não apresentam a doença de *Parkinson*. Quanto aos casos incorretamente preditos, o modelo previu incorretamente 31 que apresentam a doença de Parkinson e não previu incorretamente nenhum caso que não apresente a doença.

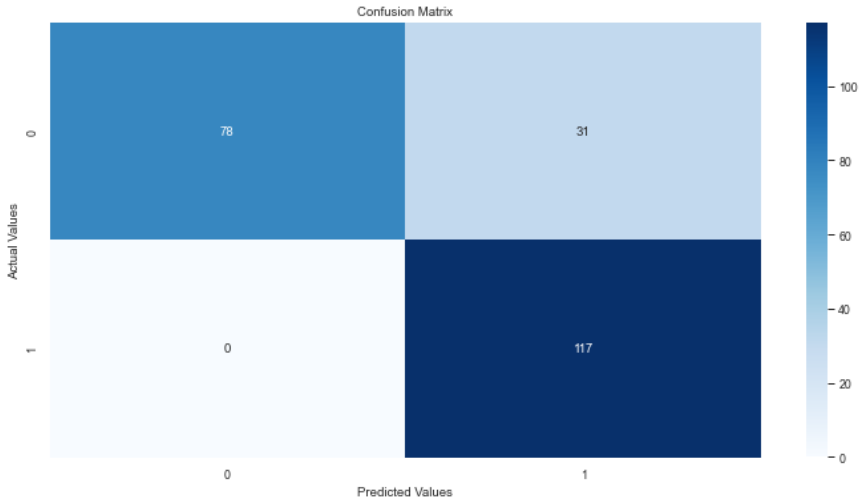

Figura 25 - Matriz de confusão com os resultados obtidos do algoritmo Random Forest com o conjunto de dados balanceado.

Em segundo lugar, foi aplicado o algoritmo *Logistic Regression* com os seguintes parâmetros configurados: *penalty*, C, *solver* e *max_iter*. O parâmetro *penalty* contêm os valores l2 e l1 e o valor selecionado foi o l2, de seguida ao parâmetro C foram atribuídos os valores 0.001, 0.01, 0.1, 1, 10, 100, 1000 e o valor ótimo selecionado foi o valor 100, quanto ao parâmetro solver que é a função de otimização do algoritmo *Logistic Regression* foram apenas configuradas duas funções a *liblinear* e a *saga*, sendo que a função selecionada como função ótima foi a *liblinear*. Por fim, o parâmetro *max_iter* foi atribuído o valor 10000. Posteriormente, o algoritmo foi treinado e testado com estes parâmetros configurados e os valores selecionados pela técnica *GridSearch* e o modelo obteve uma precisão de 91% (Tabela 18).

Tabela 18 - Classification Report do modelo Logistic Regression com os dados de testes com o conjunto de dados balanceado.

	precision	recall	F1-score	Support
0	0.87	0.96	0.91	109
1	0.96	0.96	0.91	117
accuracy			0.91	226
macro avg	0.91	0.91	0.91	226
weighted avg	0.92	0.91	0.91	226

Na Figura 26 podemos ver a avaliação dos casos classificados, que demonstra que o modelo previu corretamente 101 casos apresentando a doença de Parkinson. O modelo previu também corretamente 105 casos que não apresentam a doença de Parkinson. Quanto aos casos que previu incorretamente, o modelo previu incorretamente 4, que apresentam a doença de Parkinson, e incorretamente 16 casos, que não apresentam a doença.

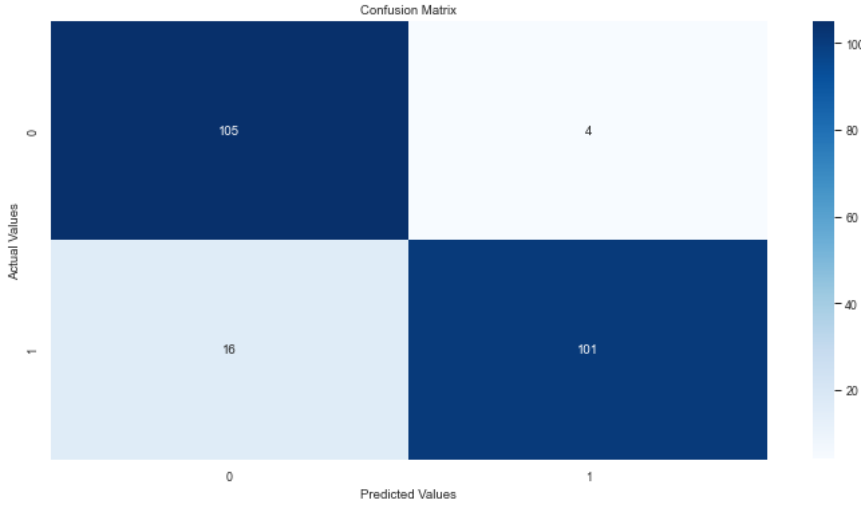

Figura 26 - Matriz de confusão com os resultados obtidos do algoritmo Logistic Regression com o conjunto de dados balanceado.

Por fim, o último algoritmo que foi aplicado foi o *Support Vector Machine* o qual foi primeiro configurado com o *kernel* linear e o única parâmetro utilizado em conjunto com este *kernel* foi o parâmetro C e ao qual se atribuíram os seguintes valores 0.001, 0.01, 0.1, 1, 10, 100, 1000 e o valor selecionado para o C foi o valor 1.0. O modelo obteve uma precisão de 96% (Tabela 19).

Tabela 19 - Classification Report do modelo Support Vector Machine com os dados de teste do modelo com Kernel Linear com o conjunto de dados balanceado.

	precision	recall	F1-score	Support
0	0.94	0.97	0.95	109
1	0.97	0.94	0.96	117
accuracy			0.96	226
macro avg	0.96	0.96	0.96	226
weighted avg	0.96	0.96	0.96	226

A Figura 27 mostra uma representação da avaliação de casos classificados e revela que o modelo previu corretamente 110 casos que apresentam a doença de *Parkinson* e previu corretamente 106 casos que não apresentam a doença de *Parkinson*. Quanto aos casos que o modelo previu incorretamente, o modelo previu incorretamente 3, que apresentam a doença de Parkinson, e previu incorretamente 7 casos, que não apresentam a doença.

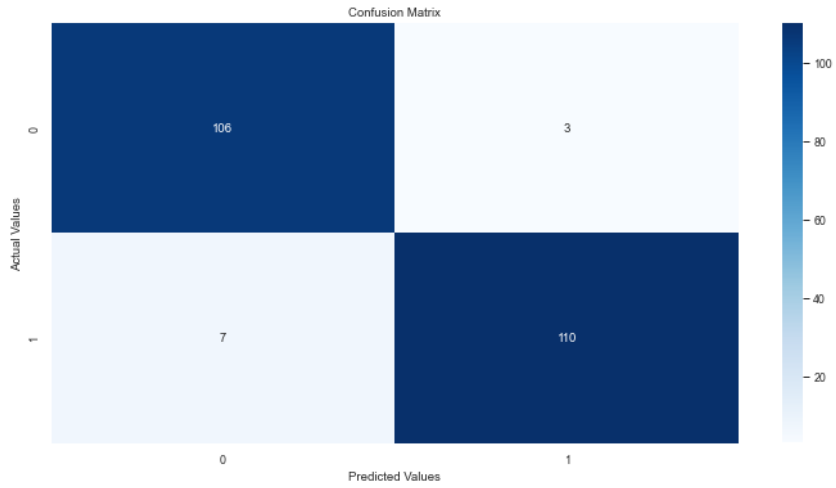

Figura 27 - Matriz de confusão com os resultados obtidos do algoritmo Support Vector Machine com o Kernel Linear com o conjunto de dados balanceado.

No entanto, o modelo *Support Vector Machine* também configurado com os restantes *Kernels* o *rbf*, o *poly* e o *sigmoid*, sendo que o *kernel* selecionado foi o *rbf*. Quanto ao parâmetro C foram atribuídos os seguintes valores 0.001, 0.01, 0.1, 1, 10, 100 e valor ótimo selecionado foi o 10. O parâmetro gamma também foi configurado com os seguintes valores 1/753, 1,0.1,0.01,0.001 em que o valor selecionado

foi o 0.1. De seguida, ao parâmetro da tolerância foram atribuídos os seguintes valores 1e-3, 1e-4, 1e-5 e o valor selecionado foi 0.001. Com estes valores nos vários parâmetros e selecionados pela técnica de *GridSearch* o modelo obteve uma precisão de 96% (Tabela 20).

Tabela 20 - Classification Report do modelo Support Vector Machine com os dados de teste do modelo com o conjunto de dados balanceado.

	precision	recall	F1-score	Support
0	0.95	0.96	0.95	109
1	0.97	0.95	0.96	117
accuracy			0.96	226
macro avg	0.96	0.96	0.96	226
weighted avg	0.96	0.96	0.96	226

A Figura 28 revela-nos a avaliação dos casos classificados. Através da análise dessa figura, podemos ver que o modelo previu 111 casos que apresentam a doença de Parkinson e 105 casos que não apresentam a doença de Parkinson. Quanto aos casos que o modelo previu incorretamente, o modelo previu incorretamente 4 que apresentam a doença de Parkinson e 6 casos que não apresentam a doença.

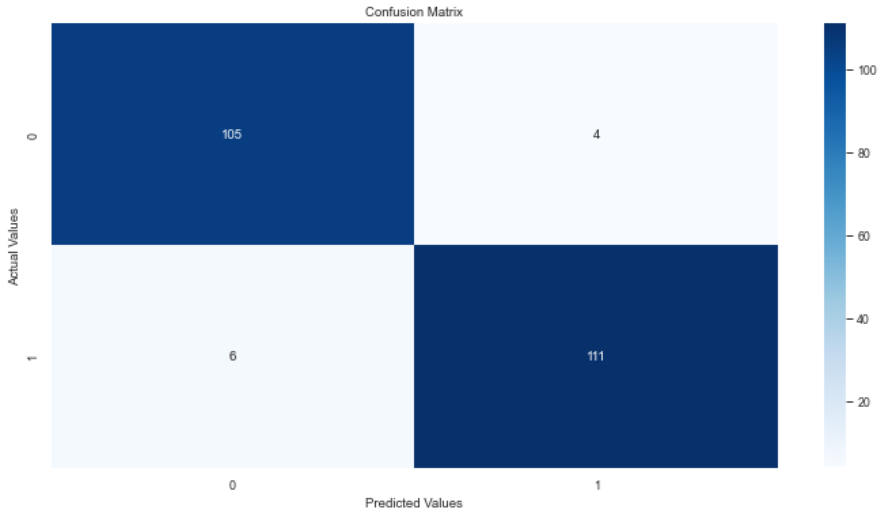

Figura 28 - Matriz de confusão com os resultados obtidos do algoritmo Support Vector Machine com o conjunto de dados balanceado.

Na Tabela 21 temos uma comparação entre os resultados obtidos na primeira abordagem com os resultados obtidos na segunda abordagem, na qual o *dataset* foi balanceado. Com estes resultados, verificamos que na primeira abordagem o algoritmo *Random Forest* é aquele que apresenta uma precisão mais baixa, quando comparado com o modelo *Logistic Regression* e *Support Vector Machine*, enquanto, o modelo *Logistic Regression* apresenta uma maior precisão. No entanto, como o conjunto de dados está desequilibrado, os dados de atribuídos tanto ao conjunto de dados de treino e ao conjunto de dados de teste esteja enviesado no sentido de ter mais registos de uma classe do que da outra. Porém, verificou-

se que quando aplicada a técnica de *SMOTE,* que gerou o mesmo número de registos nas duas classes, os resultados são mais verídicos mesmo com a descida da precisão. Nesta segunda abordagem, o modelo *Random Forest* é aquele que apresenta uma precisão mais baixa comparativamente com os modelos *Logistic Regression* e *Support Vector Machine*. Sendo que o modelo *Support Vector Machine* é o que apresenta um melhor desempenho, tanto quando configurado com o *kernel* linear como com o *rbf*, devolvendo uma precisão de 96%. Quanto ao algoritmo *Logistic Regression* verifica-se que, na primeira para a segunda abordagem, a precisão diminui, comprovando que um conjunto de dados mais equilibrado permite obter um resultado mais fiável.

Tabela 21 - Comparação dos resultados obtidos na classificação de doença de Parkinson.

Modelos	Precisão Dataset inicial	Precisão Dataset Balanceado
Random Forest	95%	86%
Logistic Regression	100%	91%
Support Vector Machine Linear	99%	96%
Support Vector Machine	100%	96%

Na Figura 29 podemos ver, de forma agregada, todas as curvas de ROC associadas a cada modelo que foi aplicado na primeira abordagem versus as curvas de ROC dos modelos aplicados na segunda abordagem, que contêm o conjunto de dados balanceado. Essas curvas têm como objetivo demonstrar o desempenho de cada modelo, de forma a ver quanto conseguem apresentar uma boa segmentação das classes. Deste modo, verifica-se que em ambas as abordagens, o sistema é capaz de diferenciar os casos que apresentam a doença de Parkinson daqueles que não a apresentam. No entanto, na primeira abordagem, os resultados das curvas são mais enviesados, devido ao modelo *Logistic Regression* apresentar um desempenho de 100% e o modelo *Support Vector Machine* apresentar a mesma precisão. Quanto às curvas de ROC dos quatro modelos, com o conjunto de dados balanceado, sabemos que é o modelo *Support Vector Machine* que obtém os melhores resultados de AUC.

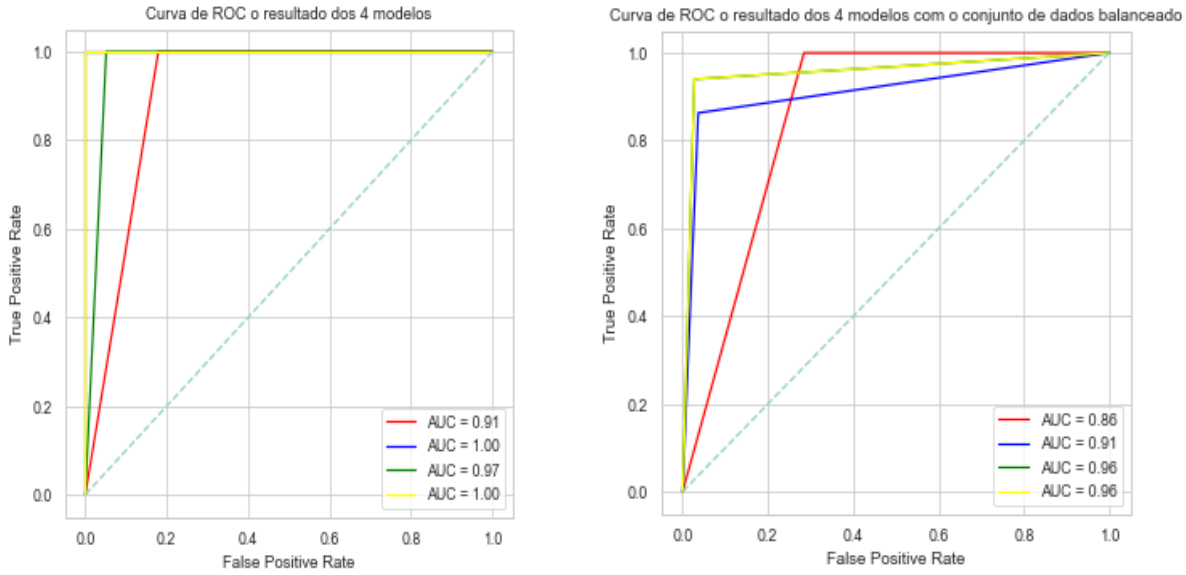

Figura 29 - Curvas de ROC dos 4 modelos VS Curvas de ROC dos 4 modelos com o conjunto de dados balanceado.

Por fim, os resultados obtidos foram comparados com outros estudos no qual o mesmo *dataset* foi utilizado. Essa comparação pode ser analisada através da Tabela 22.

Tabela 22 - Resultados de outros estudos que utilizaram o mesmo dataset.

Estudo de Referência	Modelo	Precisão
Nissar et al. (2019)	*SMOTE + Random Forest*	94.89%
Mohammadi et al. (2021)	*Logistic Regression + Voting*	97.22%
K. Saravanapriya (2017)	*Support Vector Machine* Linear	72.76%
Hoq, Uddin e Park (2021)	*Support Vector Machine*	86%

Comparando com os estudos já realizados (Tabela 22), o estudo do algoritmo *Random Forest* e *Logistic Regression* obteve um pior desempenho. No entanto, os restantes modelos obtiveram resultados superiores ao de outros estudos (Tabela 21).

6. CONCLUSÕES E TRABALHO FUTURO

6.1 Conclusões

Durante os últimos anos foram partilhados por B. E. Sakar et al. (2013), El Moudden, Ouzir e El Bernoussi (2017), vários conjuntos de dados com o objetivo de estudar a doença de Parkinson, que revelam informação acerca da marcha, caligrafia, imagem médica e registos da voz. Com eles, vários investigadores têm investido o seu tempo no estudo desta doença bem como na aplicação de algoritmos de *Machine Learning* na área da saúde, que apresenta inúmeros problemas e aplicações nos quais se pode implementar processos de extração de conhecimento com inúmeras vantagens, podendo-se ver uma aplicação efetiva de processo de previsão em muitos campos da saúde. Com o desenvolvimento deste trabalho, relativo à demonstração de técnicas de *Machine Learning* no processo de diagnóstico mais rápido da doença de Parkinson e identificação, de entre os algoritmos aplicados quais apresentam um melhor desempenho, permitiu verificar que os objetivos traçados foram alcançados. Deste modo, conclui-se em primeiro lugar que os algoritmos *Random Forest*, *Logistic Regression* e *Support Vector Machine* são bons candidatos à classificação da doença de Parkinson, pois apresentam valores de precisão acima de 80%.

Na etapa de otimização dos parâmetros dos modelos onde é aplicada a técnica *GridSearch*, a qual seleciona dentro da combinação dos vários parâmetros quais os melhores valores para cada um. Verifica-se que tanto na primeira abordagem que se utiliza o conjunto de dados desequilibrado como na segunda abordagem que se utiliza do conjunto de dados equilibrado o valor selecionado para o parâmetro C no algoritmo *Support Vector Machine* com o *kernel* linear o valor atribuído é 1. Demonstrando que quando se treina e testa o modelo *Support Vector Machine* com o *kernel* linear o parâmetro C manter-se-á estável.

Conclui-se ainda que, na primeira abordagem o algoritmo *Logistic Regression* e o algoritmo *Support Vector Machine* com o *kernel* polinomial, apresentam um valor de precisão de 100% o que possivelmente se deve ao facto de o conjunto de dados ser desequilibrado, mas também devido à seleção de 150 componentes principais na etapa de seleção de características. Isto é, talvez seja necessário aplicar mais do que uma técnica de seleção de características para restringir melhor o número de características de forma que os resultados não sejam enviesados, no sentido de classificarem corretamente os indivíduos que tem a doença de Parkinson e os indivíduos que não tem a doença de Parkinson.

Verifica-se ainda que a aplicação da técnica SMOTE que gera o número equivalente de registos para a classe minoritária, com base no número de registos da classe maioritária dando a possibilidade de se partir de um conjunto de dados equilibrado. Demonstra que os resultados obtidos em todos os modelos são mais consistentes, pois a matriz de confusão de todos os algoritmos na segunda abordagem mostra que o comportamento ao nível de da classificação dos indivíduos que apresentam a doença de Parkinson e dos indivíduos que não apresentam a doença de Parkinson já não é feita uma distinção tão "perfeita" como na primeira abordagem pois as matrizes de confusão devolvidas na segunda abordagem apresentam casos de indivíduos classificados incorretamente.

Por fim, chega-se à conclusão de que a técnica de *SMOTE* apresenta a vantagem de se trabalhar com um conjunto de dados equilibrado e que está relacionada com uns níveis de precisão mais congruentes obtidos na segunda abordagem e ainda se verifica que o algoritmo que apresenta melhores resultados nas duas abordagens é o algoritmo *Support Vector Machine*.

A principal limitação deste estudo prende-se a dois tópicos. O primeiro tópico é o tamanho do conjunto de dados que é relativamente pequeno (754 registos) para aplicação de algoritmos de *Machine Learning*, isto é, seria útil ter um conjunto dados com as mesmas condições, mas com um maior número de registos para podermos comparar os resultados a precisão de cada modelo. O segundo tópico advém da compreensão do conjunto de dados selecionado, pois como é constituído pelas características vocais exige um conhecimento prévio dessas características, mas também exige conhecimento da disciplina de processamento digital de sinal por causa da forma que as características são extraídas de cada indivíduo.

6.2 Trabalho futuro

Uma possível sugestão para trabalho futuro, seria a realização de um estudo complementar, com os algoritmos utilizados nesta dissertação, no qual estudaríamos a segmentação das classes tem a doença Parkinson e não tem a doença Parkinson pelo género dos indivíduos no conjunto de dados e os casos seriam classificados pelo género feminino e masculino com o objetivo de estudar qual o género mais propenso a desenvolver esta doença.

Além disso, seria útil ter uma aplicação móvel ou um website, no qual os indivíduos gravariam a sua voz e através desse registo a aplicação disponibilizasse abertamente os dados para serem trabalhados em projetos de investigação, funcionando como um repositório *Open Soure*. Também, seria interessante conceber e implementar uma aplicação, para plataformas móveis, tipo smartphone, que disponibilizasse

meios para realizar a gravação da voz e, posteriormente, enviasse uma mensagem indicando se a pessoa que fez a gravação apresentava ou não a doença.

Neste trabalho, na etapa de pré-processamento apenas foi utilizada a técnica *Principal Component Analysis* para selecionar as características mais importantes, sendo que foi escolhida por causa da sua fácil redução de dimensionalidade das características e do seu reduzido tempo de processamento. Deste modo, seria interessante estudar outras técnicas de seleção de características, como por exemplo a técnica *SelectKBest* que permite utilizar métodos estatísticos como o chi-quadrado e a correlação entre as características, mas também a aplicação da técnica *Recursive Feature Selection* que permite a fazer conjuntos de combinações das várias características na construção do modelo de *Machine Learning* e depois escolhe-se a combinação de características que devolvem um modelo com melhor performance. Quando o conjunto de dados contêm várias características pode ser vantajoso combinar várias técnicas de seleção de características, porque apenas aplicando uma o filtro pode ser insuficiente na tarefa de selecionar quais são as melhores características que devolvem a melhor performance do modelo.

Por fim, como o aumento do número de indivíduos com a doença de Parkinson é uma preocupação a nível Mundial, seria uma mais valia estudar algoritmos de regressão aplicados por exemplo à previsão do número de casos de Parkinson por País no qual fosse construído um Índice para os próximos 5 anos. E o mesmo fosse mostrado através de *dashboards* a toda a população para que houvesse um incentivo a um rastreio precoce da doença de Parkinson.

REFERÊNCIAS BIBLIOGRÁFICAS

Amato, F., Borzì, L., Olmo, G., & Orozco-Arroyave, J. R. (2021). An algorithm for Parkinson's disease speech classification based on isolated words analysis. *Health Information Science and Systems*, *9*(1), 1–15. https://doi.org/10.1007/s13755-021-00162-8

Benabid, A. L. (2003). Deep brain stimulation for Parkinson's disease. *Current Opinion in Neurobiology*, *13*(6), 696–706. https://doi.org/10.1016/j.conb.2003.11.001

Bharadwa, A. (2021). Practical Example of Dimensionality Reduction. Retrieved October 9, 2022, from https://towardsdatascience.com/practical-example-of-dimensionality-reduction-d0525632c355

Bhattacharya, A. (2014). Curse of Dimensionality. *Fundamentals of Database Indexing and Searching*, 141–148. https://doi.org/10.1201/b17767-13

BOERSMA, & P. (2011). Praat : doing phonetics by computer [Computer program]. *Http://Www.Praat.Org/*. Retrieved from http://ci.nii.ac.jp/naid/20001461274/en/

Bot, B. M., Suver, C., Neto, E. C., Kellen, M., Klein, A., Bare, C., … Dorsey, E. (2016). The mPower study, Parkinson disease mobile data collected using ResearchKit. *Scientific Data*, *3*(1), 1–9.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Cattell, R. B. (1966). The scree test for the number of factors. *Multivariate Behavioral Research*, *1*(2), 245–276.

Celebi, M. E., & Aydin, K. (2016). *Unsupervised learning algorithms*. Springer.

Celik, E., & Ilhan Omurca, S. (2019). *Improving Parkinson's Disease Diagnosis with Machine Learning Methods; Improving Parkinson's Disease Diagnosis with Machine Learning Methods*.

Cerri, S., Mus, L., & Blandini, F. (2019). *Parkinson ' s Disease in Women and Men : What ' s the Difference ? 9*, 501–515. https://doi.org/10.3233/JPD-191683

Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). *SMOTE : Synthetic Minority Over-sampling Technique*. *16*, 321–357.

Cunningham, P., & Delany, S. J. (2020). *k-Nearest Neighbour Classifiers: 2nd Edition (with Python examples)*. (1), 1–22. https://doi.org/10.1145/3459665

El-Turabi, A., & Bachmann, M. F. (2018). Noninfectious Disease Vaccines. *Plotkin's Vaccines*, 689-697.e4. https://doi.org/10.1016/B978-0-323-35761-6.00040-7

El Bouchefry, K., & de Souza, R. S. (2020). Learning in Big Data: Introduction to Machine Learning. *Knowledge Discovery in Big Data from Astronomy and Earth Observation: Astrogeoinformatics*, 225–249. https://doi.org/10.1016/B978-0-12-819154-5.00023-0

El Moudden, I., Ouzir, M., & ElBernoussi, S. (2017). Feature selection and extraction for class prediction in dysphonia measures analysis:A case study on Parkinson's disease speech rehabilitation. *Technology and Health Care*, *25*(4), 693–708. https://doi.org/10.3233/THC-170824

Fröhlich, F. (2016). Parkinson's Disease. *Network Neuroscience*, 291–296. https://doi.org/10.1016/B978-0-12-801560-5.00023-9

Gayed, I., Joseph, U., Fanous, M., Wan, D., Schiess, M., Ondo, W., & Won, K.-S. (2015). The impact of DaTscan in the diagnosis of Parkinson disease. *Clinical Nuclear Medicine*, *40*(5), 390–393.

Géron, A. (2019). Hands-on Machine Learning whith Scikit-Learing, Keras and Tensorfow. In *O'Reilly Media, Inc*.

Goetz, C. G., Fahn, S., Martinez-Martin, P., Poewe, W., Sampaio, C., Stebbins, G. T., … Dubois, B. (2007). Movement Disorder Society-sponsored revision of the Unified Parkinson's Disease Rating Scale (MDS-UPDRS): process, format, and clinimetric testing plan. *Movement Disorders*, *22*(1), 41–47.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA data mining software: An update. *ACM SIGKDD Explorations Newsletter*, *11*(1), 10–18.

Han, J., Kamber, M., & Pei, J. (2012). Data Preprocessing. *Data Mining*, 83–124. https://doi.org/10.1016/B978-0-12-381479-1.00003-4

Han, J., Pei, J., & Tong, H. (2022). *Data mining: concepts and techniques*. Morgan kaufmann.

Hoq, M., Uddin, M. N., & Park, S. B. (2021). Vocal feature extraction-based artificial intelligent model for parkinson's disease detection. *Diagnostics*, *11*(6), 1–22. https://doi.org/10.3390/diagnostics11061076

Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. *Journal of Educational Psychology*, *24*(6), 417.

İstanbul Medipol Üniversitesi. (2018). Medipol UNV-İstanbul. Retrieved November 6, 2022, from https://www.medipol.edu.tr/

Jin, Z., Shang, J., Zhu, Q., Ling, C., Xie, W., & Qiang, B. (2020). RFRSF: Employee Turnover Prediction Based on Random Forests and Survival Analysis. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *12343 LNCS*, 503–515. https://doi.org/10.1007/978-3-030-62008-0_35

Jollife, I. T., & Cadima, J. (2016). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065). https://doi.org/10.1098/rsta.2015.0202

K. Saravanapriya. (2017). Performance Analysis of Classification Algorithms on Diabetes Dataset. *International Journal of Computer Sciences and Engineering*, *5*(9), 15–20. https://doi.org/10.26438/ijcse/v5i9.1520

Kaiser, H. F. (1960). The application of electronic computers to factor analysis. *Educational and Psychological Measurement*, *20*(1), 141–151.

Kalia, L. V., & Lang, A. E. (2015). Parkinson's disease. *The Lancet*, *386*(9996), 896–912. https://doi.org/10.1016/S0140-6736(14)61393-3

Kohavi, R., & John, G. H. (1997). Wrappers for feature subset selection. *Artificial Intelligence*, *97*(1–2), 273–324. https://doi.org/10.1016/s0004-3702(97)00043-x

Langley, P., & Carbonell, J. G. (1984). Approaches to machine learning. In *Journal of the American Society for Information Science* (Vol. 35). https://doi.org/10.1002/asi.4630350509

Loane, C., & Politis, M. (2011). Positron emission tomography neuroimaging in Parkinson's disease. *American Journal of Translational Research*, *3*(4), 323.

Max, A. (2009). Little, Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's Disease. *IEEE Transactions on Biomedical Engineering*, *56*(4).

Mitchell, T. M. (1997). *Machine learning* (Vol. 1). McGraw-hill New York.

Mohammadi, A. G., Mehralian, P., Naseri, A., & Sajedi, H. (2021). Parkinson's disease diagnosis: The effect of autoencoders on extracting features from vocal characteristics. *Array*, *11*, 100079. https://doi.org/10.1016/J.ARRAY.2021.100079

Moreira, A. C. (2007). Comparação da Análise de Componentes Principais e da CATPCA na Avaliação da Satisfação do Passageiro de uma Transportadora Aérea. *Investigação Operacional*, *27*, 165–178. Retrieved from http://www.scielo.oces.mctes.pt/scielo.php?pid=S0874-51612007000200005&script=sci_arttext

Neilson, L., Zande, J., & Abboud, H. (2020). Deep brain stimulation surgery in Parkinson's disease. In *Diagnosis and Management in Parkinson's Disease* (pp. 577–596). https://doi.org/10.1016/b978-0-12-815946-0.00034-x

Nissar, I., Rizvi, D. R., Masood, S., & Mir, A. N. (2019). 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). https://doi.org/10.4108/eai.13-7-2018.162806

Ntetsika, T., Papathoma, P. E., & Markaki, I. (2021). Novel targeted therapies for Parkinson's disease. *Molecular Medicine*, *27*(1). https://doi.org/10.1186/s10020-021-00279-2

Park, H. A. (2013). An introduction to logistic regression: From basic concepts to interpretation with particular attention to nursing domain. *Journal of Korean Academy of Nursing*, *43*(2), 154–164. https://doi.org/10.4040/jkan.2013.43.2.154

Pearson, K. (1901). LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *2*(11), 559–572.

Poewe, W. (2012). Global scales to stage disability in PD: the Hoehn and Yahr scale. *Rating Scales Parkinsons Dis*, 115–122.

Portugal, I., Alencar, P., & Cowan, D. (2018). The use of machine learning algorithms in recommender systems: A systematic review. *Expert Systems with Applications*, *97*, 205–227. https://doi.org/10.1016/j.eswa.2017.12.020

PubMed. (2022). National Library of Medicine. Retrieved October 8, 2022, from https://pubmed.ncbi.nlm.nih.gov/

Pyatigorskaya, N., Gallea, C., Garcia-Lorenzo, D., Vidailhet, M., & Lehericy, S. (2014). A review of the use of magnetic resonance imaging in Parkinson's disease. *Therapeutic Advances in Neurological Disorders*, *7*(4), 206–220.

Qasim, H. M., Ata, O., Ansari, M. A., Alomary, M. N., Alghamdi, S., & Almehmadi, M. (2021). Hybrid feature selection framework for the parkinson imbalanced dataset prediction problem. *Medicina (Lithuania)*, *57*(11). https://doi.org/10.3390/medicina57111217

Sakar, B. E., Isenkul, M. E., Sakar, C. O., Sertbas, A., Gurgen, F., Delil, S., … Kursun, O. (2013). Collection and analysis of a Parkinson speech dataset with multiple types of sound recordings. *IEEE Journal of Biomedical and Health Informatics*, *17*(4), 828–834. https://doi.org/10.1109/JBHI.2013.2245674

Sakar, C. O., Serbes, G., Gunduz, A., Tunc, H. C., Nizam, H., Sakar, B. E., … Apaydin, H. (2019). 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 Journal*, *74*, 255–263. https://doi.org/10.1016/j.asoc.2018.10.022

Schröer, C., Kruse, F., & Gómez, J. M. (2021). A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, *181*(2019), 526–534. https://doi.org/10.1016/j.procs.2021.01.199

Scikit-Learn biblioteca. (2022). Scikit-Learn Machine Learning in Python. Retrieved October 29, 2022, from https://scikit-learn.org/stable/

Seufert, E. B. (2014). Freemium Economics: Leveraging Analytics and User Segmentation to Drive Revenue. In *Freemium Economics: Leveraging Analytics and User Segmentation to Drive Revenue*. https://doi.org/10.1016/C2013-0-00599-3

Sharanyaa, S., Renjith, P. N., & Ramesh, K. (2020). Classification of parkinson's disease using speech attributes with parametric and nonparametric machine learning techniques. *Proceedings of the 3rd International Conference on Intelligent Sustainable Systems, ICISS 2020*, 437–442. https://doi.org/10.1109/ICISS49785.2020.9316078

Siuly, S., & Zhang, Y. (2016). Medical Big Data: Neurological Diseases Diagnosis Through Medical Data Analysis. *Data Science and Engineering*, *1*(2), 54–64. https://doi.org/10.1007/s41019-016-0011-3

Solana-Lavalle, G., & Rosas-Romero, R. (2021). Analysis of voice as an assisting tool for detection of Parkinson's disease and its subsequent clinical interpretation. *Biomedical Signal Processing and*

Control, *66*, 102415. https://doi.org/10.1016/j.bspc.2021.102415

Stoker, T. B., & Greenland, J. C. (2018). Preface. In *Parkinson's Disease: Pathogenesis and Clinical Aspects*. https://doi.org/10.15586/codonpublications.parkinsonsdisease.2018.pr

Tai, Y. C., Bryan, P. G., Loayza, F., & Peláez, E. (2021). A voice analysis approach for recognizing Parkinson's disease patterns. *IFAC-PapersOnLine*, *54*(15), 382–387.

Tambasco, N., Romoli, M., & Calabresi, P. (2018). Levodopa in Parkinson's disease: current status and future developments. *Current Neuropharmacology*, *16*(8), 1239–1252.

Tantithamthavorn, C., & Hassan, A. E. (2020). *The Impact of Class Rebalancing Techniques on the Performance and Interpretation of Defect Prediction Models*. *46*(11), 1200–1219.

UCI Machine Learning Repository. (2018). Parkinson's Disease Classification Data Set. Retrieved from https://archive.ics.uci.edu/ml/datasets/Parkinson%27s+Disease+Classification

Vapnik, V. (1998). Statistical learning theory new york. *NY: Wiley*, *1*(2), 3.

World Health Organization. (2022). Parkinson Disease. Retrieved August 10, 2022, from https://www.who.int/news-room/fact-sheets/detail/parkinson-disease

Yang, F. J. (2018). An implementation of naive bayes classifier. *Proceedings - 2018 International Conference on Computational Science and Computational Intelligence, CSCI 2018*, 301–306. https://doi.org/10.1109/CSCI46756.2018.00065