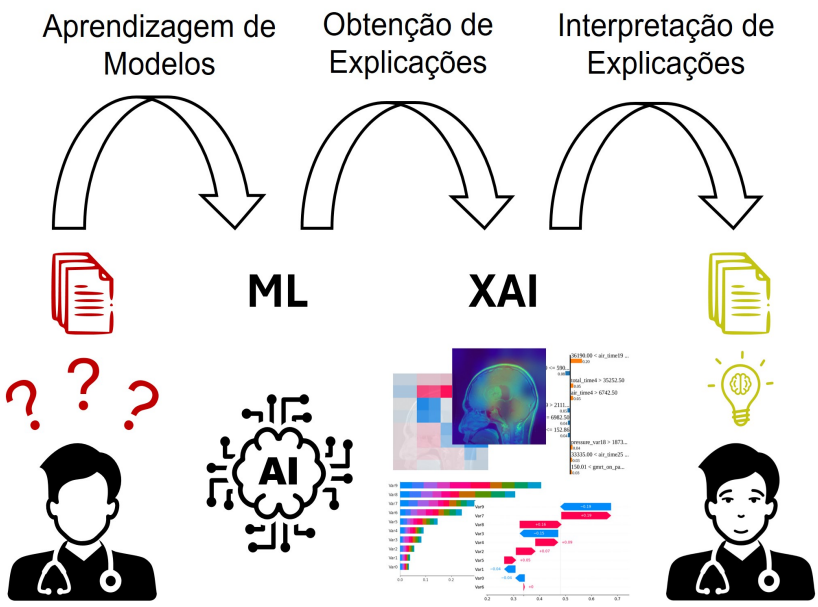
ISEL
INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA



Explicabilidade de Modelos de Classificação no Domínio Médico

ALEXANDRE VILELA REIS DE MELO MOREIRA

(Licenciado)

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e Multimédia

Orientadores: Doutor Artur Ferreira
Doutor Nuno Leite

Júri:

Presidente: Doutor Rui Jesus
Vogais: Doutor Pedro Jorge
Doutor Artur Ferreira

Dezembro 2024

Explicabilidade de Modelos de Classificação no Domínio Médico

ALEXANDRE VILELA REIS DE MELO MOREIRA

(Licenciado)

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática e Multimédia

Orientadores: Doutor Artur Ferreira, DEETC/ISEL
Doutor Nuno Leite, DEETC/ISEL

Júri:

Presidente: Doutor Rui Jesus, DEETC/ISEL
Vogais: Doutor Pedro Jorge, DEETC/ISEL
Doutor Artur Ferreira, DEETC/ISEL

Dezembro 2024

Agradecimentos

Parte da investigação realizada neste trabalho, foi financiada pelo Instituto Politécnico de Lisboa (IPL) no âmbito do Projeto IPL/IDI&CA2024/ML4EP_ISEL.

Declaração de integridade

Declaro que esta dissertação é o resultado da minha investigação pessoal e independente. O seu conteúdo é original e todas as fontes listadas nas referências bibliográficas foram consultadas e estão devidamente mencionadas no texto. Mais declaro que todas as referências científicas e técnicas relevantes para o desenvolvimento do trabalho estão devidamente citadas e constam das referências bibliográficas.

O autor

Assinado por: **Alexandre Vilela Reis de Melo Moreira**
Num. de Identificação: 30613345
Data: 2024.12.18 11:42:27+00'00'

Lisboa, 18 de Dezembro de 2024

Resumo

A Inteligência Artificial (IA) é uma área que tem vindo a ganhar destaque recentemente, particularmente a subárea de Aprendizagem Automática (AA). Nesta subárea, predominam algoritmos de aprendizagem caixa-preta, isto é, modelos cujos parâmetros internos não são diretamente observáveis pelo operador. As redes neuronais são um exemplo típico deste caso, dado que possuem um número de parâmetros de tal ordem de grandeza que a sua interpretação pelo ser humano torna-se impossível. Para além da elevada dimensionalidade, a própria estrutura deste tipo de modelos não pode ser interpretada, na medida em que dificilmente se consegue dar sentido aos parâmetros. Um contexto onde a precisão dos algoritmos é crucial é no domínio médico, onde a decisão de um algoritmo impactará a saúde de um indivíduo. Neste contexto, revela-se até perigoso não estabelecer nexos de causalidade entre os dados de entrada e a resposta de um modelo. Sem acesso ao "raciocínio" do modelo este não se pode contestar e portanto ter-se-ia de o aceitar ou descartar de forma dogmática, já que não existem justificações. Para tentar mitigar este efeito surge a *eXplainable Artificial Intelligence* (XAI). Esta área tem o objetivo de extrair explicações sobre as tomadas de decisão de modelos. Dois métodos de destaque desta área são o *Local Interpretable Model-agnostic Explanations* (LIME) e o *SHapley Additive exPlanations* (SHAP). Estas duas técnicas tiram partido apenas dos dados de entrada e saída do modelo sem aceder aos componentes internos do mesmo. Quando os dados estão num formato tabular, as explicações extraídas por estas técnicas indicam quais as características mais relevantes para uma determinada resposta do modelo. Neste trabalho, exploram-se várias técnicas de explicabilidade, através da sua aplicação em *datasets* sintéticos gerados controladamente e *datasets* do domínio médico e subsequente análise dos resultados obtidos. Foram analisados dois *datasets* reais. Primeiramente, foi analisado um *dataset* de domínio tabular designado por *Diagnosis AlzheimeR WIth haNdwriting* (DARWIN), que visa a deteção de Alzheimer através de tarefas de escrita. Neste conjunto de dados atingiu-se uma taxa de acerto média de 91% por parte dos classificadores *Random Forest* (RF) e *Explainable Boosting Machine* (EBM). Em segundo lugar foi analisado um *dataset* de imagens de ressonância magnética que visa distinguir três tipos de tumor. Neste *dataset* destacou-se a *Convolutional Neural Network* (CNN) ResNet50 atingiu uma taxa de acerto média de 90%.

Palavras-chave: Aprendizagem Automática, Domínio Médico, Explicabilidade, Inteligência Artificial, Inteligência Artificial Explicável, Interpretabilidade

Abstract

Artificial Intelligence (AI) is an area that has been gaining prominence recently, particularly the sub-area of *Machine Learning* (ML). In this sub-area, black-box learning algorithms predominate, i.e. models whose internal parameters are not directly observable by the operator. Neural networks are a typical example of this, since they have such a large number of parameters that they are impossible for humans to interpret. In addition to the high dimensionality, the very structure of this type of model cannot be interpreted, as it is difficult to make sense of the parameters. One context where the accuracy of algorithms is crucial is in the medical field, where an algorithm's decision will impact on an individual's health. In this context, it is even dangerous not to establish causal links between the input data and the response of a model. Without access to the model's "reasoning", it cannot be challenged and would therefore have to be dogmatically accepted or discarded, since there is no justification. To try to mitigate this effect, *eXplainable Artificial Intelligence* (XAI) emerges. This area aims to extract explanations about decision-making from models. Two prominent methods in this area are *Local Interpretable Model-agnostic Explanations* (LIME) and *SHapley Additive exPlanations* (SHAP). These two techniques only take advantage of the model's input and output data without accessing its internal components. When the data is in a tabular format, the explanations extracted by these techniques indicate which characteristics are most relevant to a given model response. In this work, various explainability techniques are explored by applying them to controllably generated synthetic datasets and datasets from the medical domain and subsequently analyzing the results obtained. Two real *datasets* were analyzed. Firstly, a tabular domain dataset called *Diagnosis AlzheimeR WIth haNdwriting* (DARWIN) was analyzed, which aims to detect Alzheimer's through writing tasks. On this dataset, the *Random Forest* (RF) and *Explainable Boosting Machine* (EBM) classifiers achieved an average hit rate of 91%. Secondly, a dataset of magnetic resonance images was analyzed in order to distinguish three types of tumor. In this dataset, *Convolutional Neural Network* (CNN) stood out. ResNet50 achieved an average hit rate of 90%.

Keywords: Artificial Intelligence, Explainability, Explainable Artificial Intelligence, Interpretability, Machine Learning, Medicine

Índice

Índice de Figuras

Índice de Tabelas

Índice de Listagens

Siglas

AA	Aprendizagem Automática ix, 1, 3, 5, 6, 8, 89
AI	*Artificial Intelligence* xi
AUC	*Area Under the Curve* 23, 38
AWS	*Amazon Web Services* 87
CNN	*Convolutional Neural Network* ix, xi, 41, 79, 80, 81, 82, 83, 84, 85, 86, 90
DARWIN	*Diagnosis AlzheimeR WIth haNdwriting* ix, xi, 32, 42, 69
DLBCL	*Diffuse large B cell lymphoma* 21
EBM	*Explainable Boosting Machine* ix, xi, 25, 26, 27, 41, 42, 49, 51, 52, 55, 56, 57, 58, 61, 62, 63, 64, 66, 68, 70, 74, 75, 76, 77, 78, 90
FIS	*Fuzzy Inference System* 27, 28, 29
GAM	*Generalized Additive Model* 25, 26
GBM	*Gradient Boosting Machine* 17
GCP	*Google Cloud Platform* 87
GLM	*Generalized Linear Models* 22
Grad-CAM	*Gradient-weighted Class Activation Map* 15, 23, 24, 42, 84
HVAC	*Heating, Ventilation, and Air Conditioning* 22
IA	Inteligência Artificial ix, 1, 3, 5, 6, 9, 22, 31, 89
ICE	*Individual Conditional Expectation* 18
Interpretable AI	*Interpretable Artificial Intelligence* 9
KFFS	*K-Fold Feature Selection* 21
LIME	*Local Interpretable Model-agnostic Explanations* ix, xi, 15, 16, 18, 19, 21, 22, 23, 42, 45, 46, 53, 55, 64, 72, 74, 75, 84, 87, 90, 91
LR	*Logistic Regression* 40, 41, 46, 49, 50, 51, 52, 53, 55, 56, 57, 58, 59, 61, 62, 63, 64, 66, 67, 68, 70, 71, 74, 76, 77, 78, 90, 91

LRP	*Layer-wise Relevance Propagation* 16, 25, 91
ML	*Machine Learning* xi, 1, 6
MLP	*Multi-Layer Perceptron* 22, 42, 80
PDP	*Partial Dependence Plot* 17, 18
PR	*Precision-Recall* 38, 52, 71
ReLU	*Rectified Linear Unit* 81
Responsible AI	*Responsible Artificial Intelligence* 9
RF	*Random Forest* ix, xi, 16, 22, 26, 41, 46, 49, 50, 51, 52, 53, 55, 56, 57, 58, 61, 62, 63, 64, 66, 68, 70, 71, 72, 74, 75, 76, 77, 78, 90
ROC	*Receiver Operating Characteristic* 38, 52, 71
SA	*Sensitivity Analysis* 17, 91
SHAP	*SHapley Additive exPlanations* ix, xi, 16, 18, 23, 24, 42, 45, 46, 55, 56, 61, 64, 74, 75, 78, 84, 90, 91
SVM	*Support Vector Machines* 21, 22, 41, 45, 46, 49, 50, 51, 52, 53, 55, 56, 57, 58, 60, 61, 62, 63, 64, 66, 67, 68, 70, 71, 72, 74, 75, 76, 77, 78, 79, 90
TL	*Transfer Learning* 79
XAI	*eXplainable Artificial Intelligence* ix, xi, 1, 3, 5, 9, 11, 18, 31, 40, 89
XGBoost	*eXtreme Gradient Boosting* 22, 23, 27

1 Introdução

1.1 Contexto e Motivação

A Inteligência Artificial (IA) [43] é uma área que tem vindo a ganhar destaque nos últimos anos e tende a aumentar devido às suas imensas aplicações e à automatização que a mesma pode fornecer. De entre os variados domínios de aplicação alguns são: económico, militar, médico, automobilístico e entretenimento. Dentro desta vasta área, existe a subárea de Aprendizagem Automática (AA), do inglês *Machine Learning* [3], que apresenta uma porção considerável de todo o protagonismo da IA, sendo uma das mais relevantes subáreas em termos de quantidade de pesquisa e desenvolvimento. A AA pode ser organizada em três ramos: aprendizagem por reforço, aprendizagem não supervisionada e aprendizagem supervisionada. Em cada um destes ramos, o objetivo é a resolução de um problema concreto onde se pretende criar um sistema que consiga desenvolver um programa sem intervenção humana capaz de, através de dados de entrada, estabelecer um modelo que realize aprendizagem sobre estes e produza dados de saída úteis para esta resolução.

A aprendizagem e subsequente geração de dados de saída envolve tomadas de decisão por parte do algoritmo de modo a apresentar uma solução. Os fundamentos desta tomada de decisão são, muitas vezes, opacos ao operador, pelo que este não consegue estabelecer nexos de causalidade do género:"*O modelo X tomou a decisão Y por causa de Z*". Isto revela-se um problema grave em certos domínios (e.g., médico, militar, económico) onde as consequências de um erro podem ser muito prejudiciais, levando a que haja insegurança no que toca à decisão de utilizar ou não algoritmos que ajudem a operação nestes domínios. Para além disso, saber a causa da tomada de decisão pode ajudar a própria pesquisa na medida em que se poderá analisar com mais profundidade as razões que fundamentam determinada solução.

Como tentativa de esclarecer a tomada de decisão de algoritmos surge o conceito de *eXplainable Artificial Intelligence* (XAI). Esta área visa a apresentação de algum tipo de explicação para a resposta de um algoritmo, podendo esta tomar várias formas como por exemplo visuais (no caso de *heatmaps*), numéricas, textuais ou baseadas em regras. Estas explicações podem apresentar um escopo global, onde o comportamento de um modelo é descrito como um todo ou um escopo local, onde são apresentadas explicações parciais

do comportamento do modelo válidas apenas para algumas ou até mesmo uma única decisão. A extração de explicações de modelos pode ser independente ou dependente do modelo. Quando a extração é independente diz-se que o método que a realiza é agnóstico ao modelo. Quando é dependente, isto é, quando são necessárias informações particulares de um modelo, considera-se que o método que realiza a extração é específico ao modelo.

Este trabalho irá incidir sobre dados do domínio médico, no qual a necessidade de explicações sobre os modelos é mais imperativa. É simples imaginar alguns contextos que ganhariam com a presença de explicação, tais como: identificação de doenças no geral, prescrição de medicamentos, deteção de cancro e criação de plano de tratamento. Em qualquer um destes casos, o médico (operador) e o paciente beneficiam da presença de explicações já que estas poderão desvendar correlações entre os dados que não eram previstas, levando à aquisição de conhecimento.

1.2 Problema e Resumo da Solução

Neste trabalho realiza-se o estudo de técnicas de explicabilidade para extrair conhecimento, sob a forma de explicações, sobre as decisões de modelos em dados do domínio médico. Estes dados assumem um formato tabular ou de imagem. Apesar de existirem outros tipos de extração de explicações, como por exemplo as explicações contra-factuais ou através de regras, este trabalho foca-se na seleção de características. No caso de um formato tabular, este processo consiste em selecionar as colunas/atributos que mais tiveram impacto sobre um certo resultado, no caso de explicações locais, ou que têm mais impacto mais no geral, no caso de explicações globais. Para dados na forma de imagem, a extração de explicações assume sobretudo a forma de seleção da zona mais relevante da imagem. As características originais de uma imagem são os seus *pixels*, portanto selecionando uma região de uma imagem (conjunto de *pixels*) selecionam-se, na verdade, as suas características.

Para ser possível a extração de explicações, têm de se realizar diversos passos, apresentados na figura 1.1.

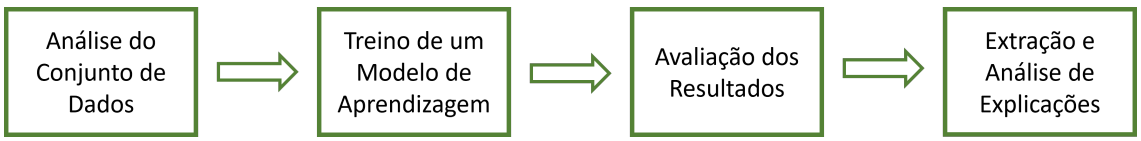


Figura 1.1: *Resumo do modo de operação adotado*

Em primeiro lugar, é necessária a aquisição e análise de um conjunto de dados (*dataset*) e, caso necessário, realizar pré-processamento sobre este. De seguida, terá de se treinar um modelo de classificação que irá realizar aprendizagem supervisionada sobre os dados. Posteriormente, proceder-se-á à avaliação dos resultados obtidos pelo classificador. Por fim, serão extraídas e analisadas explicações sobre o comportamento e aprendizagem do modelo treinado.

1.3 Contribuições da Tese

Do trabalho desenvolvido nesta tese, resultaram dois artigos científicos.

O primeiro artigo foi apresentado na conferência nacional de Simpósio em Informática, INForum2024:

Alexandre Moreira, Artur Ferreira, Nuno Leite, "Prediction of Alzheimer disease from handwriting tasks with explainability techniques", Simpósio em Informática (INForum), setembro de 2024, Lisboa, Portugal.

A hiperligação para este artigo é https://www.inforum.pt/static/files/papers/INForum_2024_paper_20.pdf

O segundo artigo foi publicado na conferência internacional EXPLAINS2024:

Alexandre Moreira, Artur Ferreira, Nuno Leite, "Prediction of Alzheimer disease on the DARWIN dataset with dimensionality reduction and explainability techniques", In Proceedings of the 1st International Conference on Explainable AI for Neural and Symbolic Methods, EXPLAINS2024, novembro de 2024, ISBN 978-989-758-720-7, pages 38-49. Aceite para publicação.

A hiperligação para este artigo é https://www.scitepress.org/Papers/2024/130174/130174.pdf

O código desenvolvido no âmbito desta tese, escrito na linguagem python (.py) sob a forma de jupyter notebooks (.ipynb), está disponível em https://github.com/Al3x4ndr3-M0r31r4/PRJ.git

1.4 Organização do Documento

O restante documento encontra-se organizado da forma que se resume em seguida.

No capítulo 2 é feito o enquadramento, adotando uma abordagem que expõe as matérias do geral para o particular. Começa-se por mencionar a área de IA como um todo, posteriormente expondo a AA em traços gerais para que depois possa ser introduzido o problema da opacidade dos seus modelos e subsequentemente enunciada a área de XAI, com uma taxonomia da mesma. Ainda neste capítulo, encontra-se o trabalho relacionado que aborda alguns métodos para extrair explicações de modelos e alguns dos seus domínios de aplicação.

No capítulo 3 é inicialmente realçada a importância da XAI para dados de domínio médico, que é a área deste trabalho, e são introduzidos os conjuntos de dados que serão utilizados para a realização de testes ao nível das suas características principais. Na secção 3.2, são apresentadas as métricas escolhidas para avaliar o desempenho dos modelos utilizados. Na

secção 3.3, é apresentada a sequência de ações a ser tomada para cada conjunto de dados, desde a sua análise até à avaliação dos resultados, e apresentados os modelos escolhidos para a classificação destes. Na secção 3.4, sendo esta a última, é abordado o modo de operação adotado para a utilização do sistema num contexto real.

No capítulo 4 são tratados 2 tipos de dados: sintéticos e reais. Os dados sintéticos têm a função de testar os métodos de classificação e de explicabilidade. Com este tipo de dados são feitas diversas alterações à forma de criar o conjunto de dados sintético para poder avaliar os modelos de diversas perspetivas. No contexto de dados reais, foram usados dois conjuntos de dados. Um é do domínio de dados tabular, com 174 exemplos, sendo vocacionado para a deteção de Alzheimer através da análise de escrita dos participantes. Com este conjunto de dados procuram-se quais os atributos de escrita que contribuem mais para detetar a doença. O outro conjunto de dados inclui imagens de ressonância magnética para a deteção de 3 tipos de cancro no cérebro. Aqui as explicações são mostradas através da saliência da região da imagem mais relevante para a sua classificação, eventualmente correspondendo à região do tumor.

O capítulo 5 conclui o documento com a indicação das principais conclusões deste estudo. Apresentam-se também as direções de trabalho futuro.

2 Enquadramento e Estado da Arte

Neste capítulo é feita a contextualização do problema através de uma abordagem do geral para o particular. Aqui são tratadas as áreas de carácter mais geral numa primeira instância, sendo feita uma progressiva especialização até ao detalhe do problema.

A secção 2.1 apresenta a área onde este trabalho se insere, a Inteligência Artificial (IA) realçando a subárea de Aprendizagem Automática (AA). A área de AA organiza-se em três subcategorias principais: aprendizagem por reforço, aprendizagem não supervisionada e aprendizagem supervisionada.

A secção 2.2 trata o problema da caixa preta, que refere a ausência de transparência de alguns algoritmos em relação às suas decisões. Quando um algoritmo oferece uma resposta, muitas vezes os fundamentos da mesma não são interpretáveis por seres humanos.

A secção 2.3 introduz a *eXplainable Artificial Intelligence* (XAI), aplicada às caixas pretas mencionadas na secção anterior, através da extração de explicações dos algoritmos para a tomada de decisão. Nesta secção, são introduzidos os conceitos inerentes a esta área e feita uma taxonomia dos métodos que pretendem fornecer algum tipo de explicações.

A secção 2.4 trata do Trabalho Relacionado, ou seja, propostas com alguma semelhança com o trabalho desenvolvido, a nível de abordagem, mencionando técnicas específicas, bem como a nível de domínio.

2.1 Inteligência Artificial e Aprendizagem Automática

A emergente área de IA é de grande heterogeneidade e vastidão, contendo muitas subáreas com objetivos díspares. Esta área insere-se no ramo das ciências da computação e dedica-se ao estudo e conceção de "máquinas inteligentes". A definição de IA está dependente do significado atribuído ao termo "inteligência" relativo a uma máquina. Em geral, considera-se que uma máquina é "inteligente" caso consiga reproduzir o raciocínio ou o comportamento humano considerado inteligente em alguma medida [50]. Neste contexto, a IA pode ser analisada em dois domínios: IA Geral e IA Restrita.

A IA Geral refere-se a um agente capaz de aprender e realizar qualquer tarefa que um ser

humano consiga. A IA Restrita destina-se à conceção de sistemas orientados à resolução de um problema específico bem definido. Este último tipo de IA é o que predomina num contexto prático e é onde está o foco do desenvolvimento e da pesquisa.

Uma das subáreas da IA com maior sucesso é a Aprendizagem Automática (AA) ou *Machine Learning*, sendo este termo introduzido por Arthur Samuel em 1959 [45]. Esta subárea dedica-se ao estudo e desenvolvimento de algoritmos que consigam realizar tarefas sem que lhes sejam fornecidas as instruções específicas para tal. A estes algoritmos é dado um conjunto de dados para encontrarem padrões, generalizarem e fazerem previsões sobre os mesmos. Este tipo de aprendizagem é baseada no raciocínio indutivo.

As várias abordagens presentes em AA, no geral, podem ser colocadas na subárea ou categoria de aprendizagem por reforço, aprendizagem não supervisionada ou aprendizagem supervisionada, tal como está representado na figura 2.1.

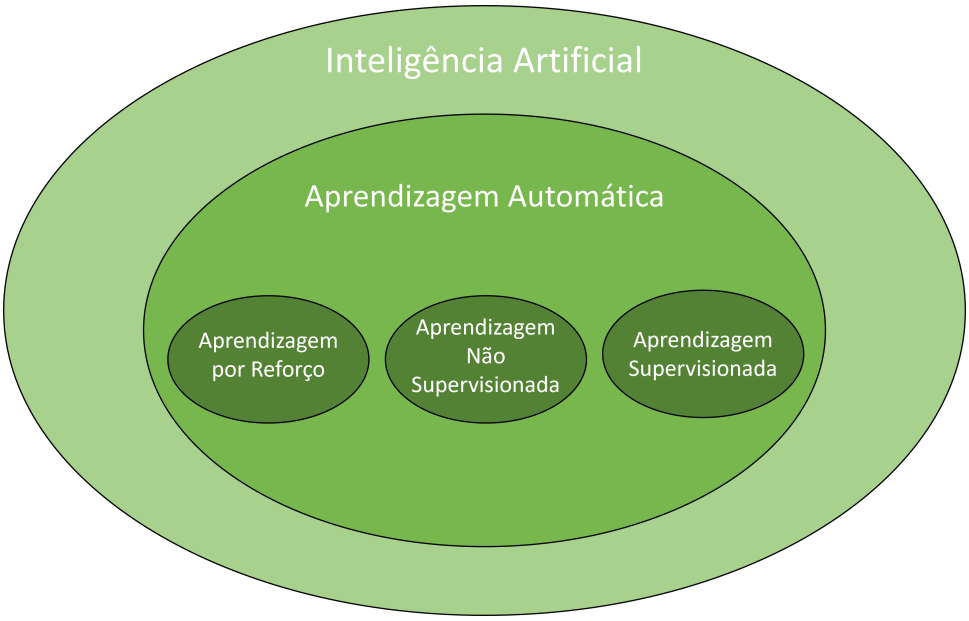


Figura 2.1: *AA como subárea da IA e subáreas da AA*

Na aprendizagem por reforço existem cinco conceitos principais: agente, ambiente, estado, ação e recompensa. O objetivo é o treino de um agente, através de recompensas, para realizar uma determinada tarefa interagindo com o ambiente através de ações. O agente obtém informação do ambiente em que se encontra, sendo essa informação interpretada para dar significado à observação, designado por estado. Com base na sua observação, o agente, segundo um determinado critério, realiza uma ação. A essa ação será dada uma recompensa positiva, negativa ou neutra, dependendo da consequência que determinada ação gerar, sendo que a maior recompensa será quando o mesmo atinge o objetivo. Quando atinge o objetivo, o agente iniciará outro episódio onde terá o mesmo objetivo. Neste novo episódio, o agente registou as recompensas recebidas e por isso realizou uma certa aprendizagem, isto é, associou um tipo de observação a uma recompensa. O objetivo do agente é maximizar o valor cumulativo das recompensas em cada episódio. À medida que o agente realiza ações e lhes associa recompensas, este fará uma escolha mais informada. A figura 2.2 apresenta o processo descrito de forma sucinta.

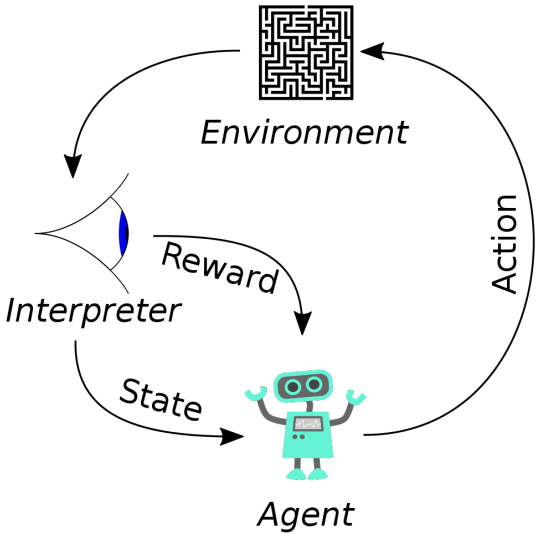


Figura 2.2: *Forma de funcionamento da aprendizagem por reforço. Adaptado de [37]*

A aprendizagem não supervisionada usa tipicamente dados num formato tabular onde existem d colunas que correspondem às variáveis/características e n linhas que correspondem às instâncias/exemplos de um determinado problema. A tabela 2.1 apresenta um conjunto de dados no domínio numérico.

Tabela 2.1: Conjunto de dados de entrada para aprendizagem não supervisionada

Var1	Var2	Var3
12	32	46
89	42	10
5	97	60
47	78	91
40	54	16

Algumas técnicas de este tipo de aprendizagem são o agrupamento (*clustering*), redução de dimensionalidade e modelos generativos. No caso do *clustering*, o objetivo é agrupar linhas/exemplos/instâncias que apresentem semelhanças entre si. A figura 2.3 ilustra este conceito, sendo que cada instância/exemplo do conjunto de dados corresponde a um ponto no espaço de características.

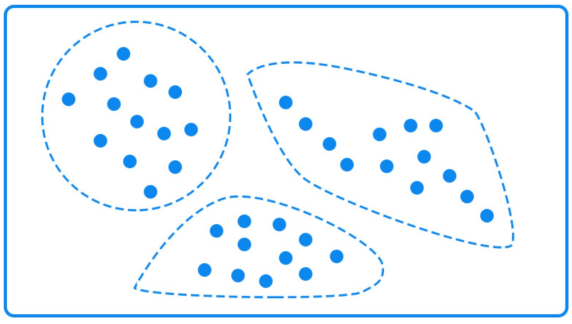


Figura 2.3: *Agrupamento de instâncias próximas. Adaptado de [21]*

Dois tipos de aprendizagem supervisionada são a classificação e a regressão. Na aprendizagem supervisionada, os dados de entrada são tipicamente em formato tabular, no entanto, apresenta uma coluna adicional que indica a etiqueta associada a cada instância. Por vezes, é considerado que a tarefa de agrupamento precede a tarefa de classificação, uma vez que a formação de agrupamentos pode ser interpretada como a atribuição de uma etiqueta a um conjunto de pontos para posteriormente ser usada na tarefa de classificação, tal como é mostrado na tabela 2.2.

Tabela 2.2: Conjunto de dados de entrada para aprendizagem supervisionada

Var1	Var2	Var3	Classe
12	32	46	2
89	42	10	0
5	97	60	1
47	78	91	1
40	54	16	0

Em primeiro lugar, para realizar a aprendizagem de modelos, o que se deve fazer é dividir todo o conjunto de dados (conjunto de dados) num conjunto de treino e num conjunto de teste. O que é pretendido é que o modelo a treinar encontre, no conjunto de treino, regras e formas de discriminação das classes, para depois no conjunto de teste, com base nos padrões aprendidos, tentar prever, para cada instância, a respetiva classe. No conjunto de treino, o modelo usa a etiqueta da classe, enquanto que no conjunto de teste a classe é ocultada para ser possível avaliar o seu desempenho. Numa tarefa de classificação, o modelo procurará estabelecer divisões entre exemplos de classes diferentes, tal como é apresentado na figura 2.4, considerando que cada forma geométrica corresponde a uma classe.

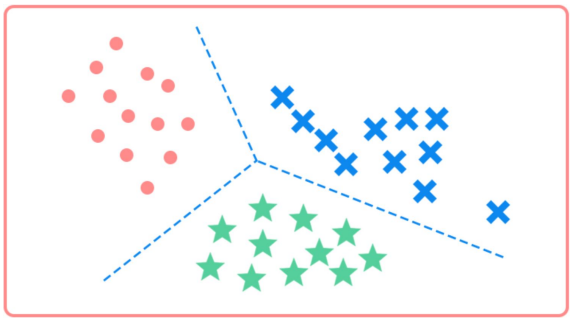


Figura 2.4: *Separação de diferentes classes na aprendizagem supervisionada. Adaptado de [21]*

2.2 O problema da caixa preta

No âmbito da AA, a tarefa dos algoritmos é determinar um modelo ou programa que, através dos dados de entrada, realize transformações aos mesmos, extraindo-lhes significado para produzir dados de saída. O processo de aprendizagem onde, através dos dados de entrada, são feitas decisões, muitas vezes, não pode ser percebido por humanos. Quando as

decisões feitas por um modelo não podem ser interpretadas por humanos, este é designado por caixa preta (*black box*).

Este fenómeno revela-se um problema com gravidade na medida em que existem alguns domínios (e.g., médico, militar, económico) que exigem um grande rigor no seu tratamento, uma vez que um erro poderá ter consequências penosas. Por essa razão, não basta criar modelos com elevado desempenho. Colocar um algoritmo a realizar tarefas em algum destes domínios sem que se perceba a sua base para realizar escolhas e decisões, contribui para uma grande desconfiança e pode causar até o abandono de qualquer mecanismo deste tipo. Por forma a gerar confiança no uso destas técnicas que possibilitam a automatização e, por vezes, o aumento de eficácia terá de se ter conhecimento, em algum nível, da sua forma de operar. Para além disso, saber os fundamentos para a tomada de decisão destes algoritmos apresenta diversas vantagens.

2.3 *eXplainable Artificial Intelligence* (XAI)

Para resolver ou pelo menos mitigar o problema da caixa preta abordado na secção 2.2, surge o conceito de *eXplainable Artificial Intelligence*. Outro termo utilizado neste âmbito, mas que apresenta um decaimento de uso segundo [5] é *Interpretable Artificial Intelligence* (Interpretable AI). Esta subárea da IA foca-se no esclarecimento do conteúdo das caixas pretas, tentando atingir o nível da transparência. Nesta secção, serão abordados os vários aspetos deste tópico, nomeadamente a sua utilidade, conceitos envolvidos e taxonomia.

2.3.1 Conceitos Fundamentais

Nas várias referências consultadas é muitas vezes mencionada a falta de unanimidade e de formalidade dos seus conceitos, especialmente em [31], pelo que os termos utilizados apresentam definições heterogéneas entre autores. Foram feitas diversas tentativas de formalização e normalização, tais como [1, 5, 12, 23, 25, 31, 36, 44, 46, 49, 51], com taxonomias ou novas definições.

No texto que se segue, serão apresentadas as principais definições de termos usados no âmbito de XAI. Em [1], XAI é vista como uma tentativa de implementação de *Responsible Artificial Intelligence* (Responsible AI) que refere o uso de IA tendo em consideração valores sociais, morais e éticos. Segundo estes autores, existem três definições fundamentais nesta área:

- Responsabilização – Necessidade de uma pessoa se explicar e justificar às pessoas que interagem com o sistema desenvolvido.
- Responsabilidade – O papel das próprias pessoas e a capacidade dos sistemas de IA de responder pela decisão de alguém e identificar erros ou resultados inesperados.
- Transparência – Necessidade de descrever, inspecionar e reproduzir os mecanismos com que a IA faz decisões e aprende a adaptar-se ao seu ambiente e à informação que lhe é fornecida.

Em [5], são propostas as seguintes definições:

- Inteligibilidade – Denota se a função de um modelo é clara, sem que se saibam detalhes do seu funcionamento interno.
- Compreensibilidade – Capacidade de um algoritmo de aprendizagem representar os conhecimentos aprendidos de uma forma compreensível para os humanos. Esta noção de compreensibilidade do modelo deriva dos postulados de Michalski [34], que afirmou que "*os resultados da indução por computador devem ser descrições simbólicas de determinadas entidades, semântica e estruturalmente semelhantes àquelas que um especialista humano poderia produzir observando as mesmas entidades. Os componentes dessas descrições devem ser compreensíveis como "pedaços" únicos de informação, diretamente interpretáveis em linguagem natural, e devem relacionar conceitos quantitativos e qualitativos de uma forma integrada*". Dada a sua difícil quantificação, a compreensibilidade está normalmente relacionada com a avaliação da complexidade do modelo.
- Interpretabilidade – Capacidade de explicar ou fornecer o significado em termos compreensíveis para um ser humano.
- Explicabilidade – Noção de explicação como uma interface entre os seres humanos e um tomador de decisão, ou seja, simultaneamente uma representação precisa do tomador de decisão e compreensível para os humanos.
- Transparência – Um modelo é considerado transparente se por si só é compreensível. Como um modelo pode apresentar diferentes graus de compreensibilidade, os modelos transparentes são organizados em três categorias: simuláveis, divisíveis e algoritmicamente transparentes.

No mesmo artigo [5], é referido que a mais importante definição é a inteligibilidade e que todos os outros conceitos são derivados deste. As definições consideradas mais importantes foram anteriormente enunciadas, no entanto, em [46, 51] analisam-se diversas contribuições para esta área e apresentam-se definições para além das mencionadas.

2.3.2 Utilidade da XAI

É legítimo questionar sobre a utilidade desta área como um todo, já que se podem colocar as seguintes questões: "*Quais são os benefícios de saber os fundamentos para o comportamento de um modelo? Não será mais importante o seu desempenho? No caso de exclusão mútua, devemos preferir as explicações ao desempenho?*". As respostas para estas perguntas dependem severamente do contexto/domínio de onde surgem. Tal como sugerido na secção 2.2, existem casos onde a presença de explicações é mais imperativa do que em outros. Por exemplo, em sistemas de recomendação, o erro é menos grave do que num sistema de diagnóstico médico, portanto, é mais importante ter conhecimento sobre o sistema que realiza diagnósticos, dado que este poderá influenciar o tratamento do paciente.

A utilidade da XAI é questionada em [5], já que responde diretamente às perguntas: "*O quê? Porquê? Como? Para quê?*". A pergunta "*O quê?*" refere-se ao que é na verdade uma explicação e a XAI como um todo. A pergunta "*Porquê?*" refere-se à necessidade da existência desta área. A pergunta "*Como?*" refere-se à maneira de atingir a explicabilidade. A pergunta "*Para quê?*" foca nos benefícios que a proliferação desta área poderá trazer, designadamente:

- Confiança – Ao saber as premissas que o modelo adotou para chegar a uma certa conclusão torna-o mais concreto aos olhos dos utilizadores. Tal pode ser visto como um aumento da clareza do modelo opaco, conduzindo a maior adoção.
- Causalidade – Saber as razões de ser dos dados de saída também pode gerar nexos de causalidade, na medida que se sabe o que provocou determinada conclusão.
- Justiça – Por vezes são usados algoritmos em situações relacionadas com Direito ou Finanças, por exemplo um pedido de empréstimo. Se o algoritmo rejeitar um pedido de empréstimo, é necessário conhecer a sua fundamentação para prestar contas sobre essa rejeição, nomeadamente será útil para a análise da instituição e para a elaboração de um documento justificativo a apresentar ao requerente.

Outra perspetiva é apresentada em [42], a qual salienta sobretudo a capacidade que esta área tem para proporcionar novo conhecimento científico. Isto verifica-se uma vez que métodos que fornecem explicações, no fundo, esclarecem os padrões existentes nos dados fornecidos a um certo algoritmo. Tal poderá gerar uma pesquisa informada, isto é, o conhecimento das correlações dos dados pode alavancar a pesquisa levando a outro tipo de descobertas.

Em [35], a área de XAI é analisada numa perspetiva social sem o aprofundamento de questões técnicas, tais como taxonomias ou detalhes de implementação de métodos que fornecem explicações. Neste artigo, são feitas reflexões sobre as características de uma boa explicação e mencionadas algumas teorias sobre as pessoas quererem explicações.

Também no escopo da pergunta "*No caso de exclusão mútua, devemos preferir as explicações ao desempenho?*" existe uma relação entre o desempenho que um modelo proporciona e o seu grau de interpretabilidade. Quanto maior for o desempenho, menor é a sua transparência e, portanto, menor é a sua interpretabilidade. O gráfico da figura 2.5, adaptado de [5], relaciona a interpretabilidade com o desempenho do modelo.

Como é possível constatar através da figura 2.5, os métodos de aprendizagem mais opacos tendem a ser também os que melhor desempenho oferecem. Tal gera um problema para os operadores no momento de decisão do método mais adequado. Apesar disso, existem técnicas que mantêm a taxa de acerto e, simultaneamente, oferecem explicações, tal como mencionadas nas próximas secções.

2.3.3 Transparência

Segundo [5, 31, 51], os métodos de aprendizagem podem ser opacos ou apresentar diversos níveis de transparência, nomeadamente:

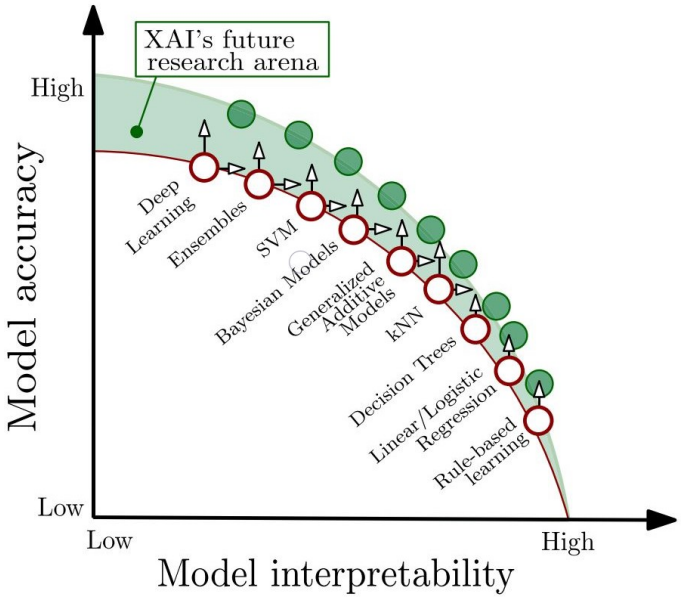


Figura 2.5: *Compromisso de Interpretabilidade. Adaptado de [5]*

1. Simuláveis – Modelos capazes de serem simulados ou pensados estritamente por humanos, sendo que a complexidade dos modelos afeta esta sua capacidade. Uma rede neuronal com camadas densas não é simulável, ao contrário de uma rede com um único nó. Por exemplo, um modelo *Bayesiano* é simulável se houver um número reduzido de variáveis e poder ser diretamente interpretado pelo utilizador.

2. Divisíveis – Modelos que podem ser separados em componentes simuláveis, mas que não podem ser humanamente analisados como um todo. Por exemplo, um modelo *Bayesiano* que envolva um número elevado de variáveis que, para ser interpretado, tenha que ser dividido em diversas partes para possibilitar a análise.

3. Algoritmicamente Transparentes – Modelos que, apesar de serem divisíveis e em teoria ser possível reproduzir o seu comportamento, as suas divisões não são possíveis de interpretar devido à sua complexidade. Por exemplo, um modelo *Bayesiano* que, devido à sua dimensão, não é interpretável sem recurso a métodos de análise estatística.

Em [5], são categorizados como transparentes os seguintes modelos: *Linear/Logistic Regression*, Árvores de Decisão, *K-Nearest Neighbors*, *Rule-Based Learners*, *General Additive Models*, Modelos *Bayesianos*. As redes neuronais convolucionais não são consideradas transparentes, dado que são um tipo de redes neuronais profundas. Não é possível perceber, por exemplo, a importância de cada característica ou a contribuição que a mesma teve para o resultado sem uma análise muito detalhada. Uma abordagem para tal seria o mapeamento dos dados de saída das camadas convolucionais nos dados de entrada para que se obtenha a dimensionalidade original e se possa tentar descobrir a relevância de cada característica.

Caso o modelo em questão não seja transparente, terá de se proceder à extração de explicações após o término do seu treino, ou seja, obter explicações *post-hoc*. Em [46], referem-se mais dois tipos de modelo: Misto e Auto-Explicativo. Para tal, terá de se proceder a algo

que consiga fazer tal extração, através da entidade Explicador. Na secção 2.3.5 ir-se-á abordar os vários tipos de Explicadores de acordo com a sua portabilidade.

2.3.4 Formas de Apresentação de Explicações

Em [51], são apresentadas várias formas que as explicações podem assumir: numéricas, baseadas em regras, textuais, visuais ou mistas. Outra categorização é a apresentada em [1], que divide os tipos de explicações em: visualização, extração de conhecimento, através de métodos de influência e baseadas em exemplos. A seguir, dão-se exemplos de algumas destas formas.

2.3.4.1 Numéricas

Em [2, 51], explicações em forma de gráficos ou probabilidades são consideradas explicações numéricas. Em [2], é proposto um método que analisa a importância de cada característica através da oclusão progressiva das mesmas. Primeiro é necessário o treino de um modelo para realizar a classificação de instâncias. Tendo o modelo treinado, é feita a progressiva oclusão de cada característica e evidenciado qual o impacto que essa oclusão tem na taxa de acerto do modelo. Um dos exemplos mostrados neste artigo é ilustrado na figura 2.6, aplicado a três modelos diferentes. Nesta figura, o eixo dos x representa a oclusão de uma característica, isto é, a redução, em percentagem, da sua importância. O eixo dos y representa o impacto que tal oclusão teve no desempenho do modelo.

2.3.4.2 Baseadas em Regras

As explicações baseadas em regras destinam-se a enunciar um conjunto de regras que são, no fundo, premissas na forma de implicação "*se X, então Y*" onde X seria condição de certa característica e Y a conclusão a que um modelo chegou. Um exemplo possível é o seguinte "*se idade* $>$ *65 então probabilidade de reumático* $\geq$ *0,50*".

Um exemplo concreto mostrado em [41] designa-se de *Anchors*, um método para extração de explicações baseado em regras. O exemplo deste artigo está representado na tabela 2.3.

Tabela 2.3: Sistema de explicação baseado em regras Anchors [41]

English	Portuguese
This is the **question** we must address	**Esta** é a questão que temos que enfrentar
This is the **problem** we must address	**Este** é o problema que temos que enfrentar
This is what we must address	É **isso** que temos de enfrentar

Neste exemplo é pretendida a tradução de Inglês para Português. As palavras a negrito representam os dados de entrada relevantes para se ter escolhido a palavra a negrito. Em forma de regras temos: "*Se existirem as palavras {This, is, question} então a tradução envolverá a palavra {Esta}*".

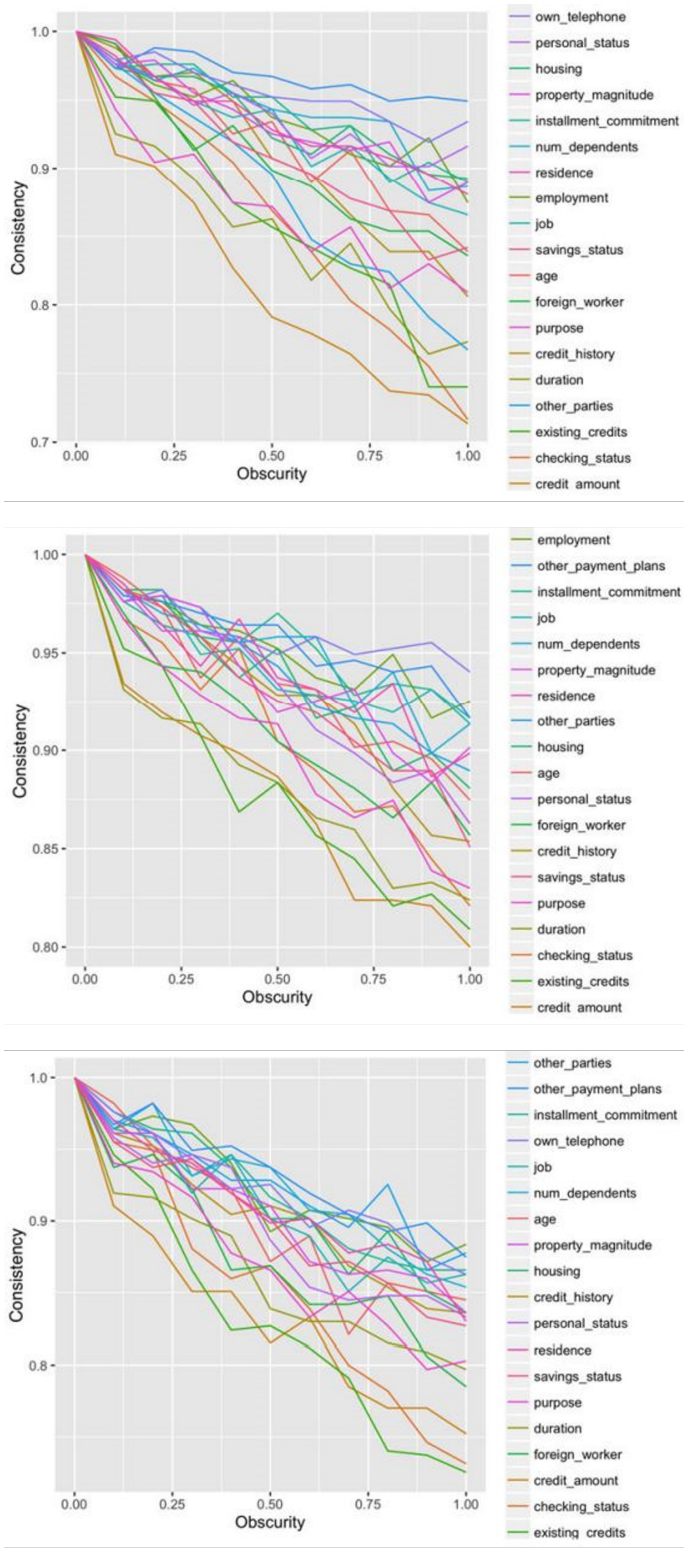
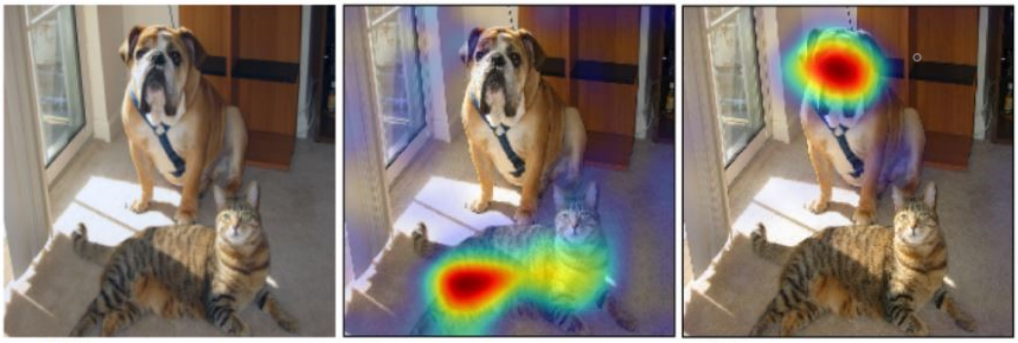


Figura 2.6: *Extração de explicações através de um gráfico. Adaptado de [2]*

2.3.4.3 Visuais

As explicações visuais são provavelmente as mais comuns e, muitas vezes, tomam a forma de *heatmaps* ou outro tipo de representação que indique as características mais relevantes para o resultado, principalmente quando o domínio dos dados de entrada é imagem. Um exemplo de extração de explicações através deste método é o *Gradient-weighted Class Activation Map* (Grad-CAM) [47]. Deste modo, o Grad-CAM é treinado com o conjunto de dados pretendido e, ao mesmo tempo, extrai as explicações do seu modo de operação. Um exemplo de aplicação pode ser visto na figura 2.7.

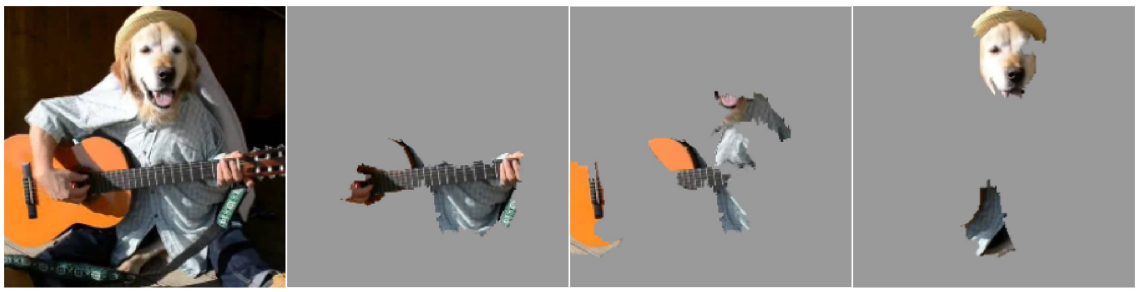


(a) *Imagem Original* (b) *'Cat'* (c) *'Dog'*

Figura 2.7: *Extração de explicações através de heatmapping (quanto mais vermelho mais relevante). Adaptado de [47]*

Nesta figura temos o destaque através de cor da zona da imagem que mais contribuiu para uma dada classificação. Quando a etiqueta a ser analisada é "cat" a zona da imagem mais relevante é a que contém um gato (figura 2.7 (b)). Quando a etiqueta a ser analisada é "dog" a componente mais relevante da imagem é a que corresponde ao cão (figura 2.7 (c)).

Outro método que oferece explicações em formato visual é o *Local Interpretable Model-agnostic Explanations* (LIME) [40]. Este método também indica que zonas da imagem contribuíram para a decisão de uma classe, tal como apresentado na figura 2.8.



(a) *Imagem Original* (b) *'Electric Guitar'* (c) *'Acoustic Guitar'* (d) *'Labrador'*

Figura 2.8: *Extração de explicações através de saliência de pixel. Adaptado de [40]*

2.3.5 Portabilidade dos Explicadores

Em diversas fontes consultadas, tais como [1, 5, 12, 46, 51], os Explicadores podem ser Agnósticos ou Específicos sobre o modelo.

2.3.5.1 Explicadores Agnósticos ao Modelo

Os explicadores agnósticos ao modelo apenas necessitam de acesso aos dados de entrada e de saída do modelo, não requerendo conhecimento sobre a estrutura ou sobre os parâmetros do mesmo. Um dos métodos que se encaixa nesta categoria é o LIME [40]. Para extrair as explicações, o mesmo realiza perturbações em instâncias selecionadas aleatoriamente e averigua de que forma é que essas perturbações afetam a classe da mesma. Neste método, para cada instância selecionada e suas perturbações, é treinado um modelo naturalmente interpretável, por exemplo, uma árvore de decisão. Desta forma, é possível saber as características mais importantes para esse modelo a nível local, já que no caso das árvores de decisão o primeiro nó (raiz da árvore) é o que contém a característica que mais discrimina e, portanto, a mais importante.

Outro método que não necessita de obter informação dos parâmetros internos do modelo opaco sobre o qual pretende extrair explicações é o *SHapley Additive exPlanations* (SHAP) [33]. Este modelo usa conceitos da teoria dos jogos para calcular a importância de cada característica. A ideia base deste método é a criação de um subconjunto de características onde $|S|$ é o número de elementos que este subconjunto apresenta e F é o número total de características. Para todas as permutações possíveis de S, é treinado o classificador com este conjunto S duas vezes por cada permutação, uma vez com a característica que se pretende averiguar a importância e outra sem essa mesma característica. Para cada treino, são comparadas as diferentes predições do conjunto que tinha característica e que não tinha. Se a diferença for significativa, então a característica terá uma maior importância, caso contrário, a característica é irrelevante. Esta é apenas a ideia central do método, já que na prática teremos aproximações para não ter que treinar o modelo para cada subconjunto, o que resultaria num processo computacionalmente bastante complexo e, portanto, inviável.

2.3.5.2 Explicadores Específicos ao Modelo

Os explicadores específicos ao modelo necessitam de ter acesso à estrutura interna do mesmo para conseguirem extrair algum tipo de explicações. Este tipo de explicadores podem não estar acoplados diretamente a um único método de aprendizagem específico e sim a um conjunto de métodos que tenham um modo de operação semelhante. No entanto, se o método de aprendizagem for bastante diferente, por exemplo comparando o *Random Forest* (RF) [9] a uma rede neuronal, terá de se fazer uma adaptação do explicador para este novo tipo de modelos.

Um método desta categoria é o *Layer-wise Relevance Propagation* (LRP) [6]. Este método gera explicações visuais através da saliência de pixel (*heatmapping*), isto é, determinar a relevância de cada pixel para a previsão feita. Tal como o próprio nome indica, tira partido das camadas que uma rede neuronal apresenta para calcular a relevância das várias dimensões dos dados de entrada, que em imagens correspondem aos pixels. A relevância é calculada a partir dos nós finais da rede onde a relevância é o próprio valor do nó. A relevância é calculada da camada de saída pelas camadas intermédias até chegar aos dados de entrada. Imaginando que um nó que está na camada de saída apresenta o valor 10, esta

será a sua relevância. Imagine-se também que este apresenta 3 conexões com 3 neurónios da camada anterior, onde a cada conexão está associado um coeficiente de ativação. O valor 10 será distribuído pelas três conexões de forma proporcional, isto é, quanto maior for o coeficiente de ativação maior será a relevância passada, portanto a soma das relevâncias passadas para estas três conexões será 10 distribuída percentualmente pelas três conexões. Um nó que não está na camada de saída obtém a sua relevância através da soma das relevâncias que lhe são passadas por cada conexão. No final, o valor destas relevâncias chegará aos dados de entrada que serão interpretadas como relevâncias de cada pixel, onde se poderá, posteriormente, fazer um *heatmap* para averiguar quais é que contribuíram mais para a escolha de determinada classe.

Outro método que acede aos parâmetros internos do modelo é o *Sensitivity Analysis* (SA) proposto em [7]. Este método indica as características mais relevantes de uma certa instância. Esta informação é adquirida através do cálculo da derivada local para cada característica de uma instância pretendida. Ao realizar este cálculo, é possível saber se, num certo espaço de pesos, a mudança do valor de uma característica afetaria em grande medida a previsão dada pelo modelo. Caso a derivada tenha um grande valor absoluto, então significa que a característica a ser analisada apresenta um papel relevante para a previsão, dado que a sua alteração conduz a resultados diferentes. Este tipo de explicação pode ser representado por vetores, sendo calculadas as derivadas em todas as dimensões dos dados. Para ser possível a visualização adequada, em [7] são mostrados estes vetores para todas as combinações dois a dois das dimensões dos dados, o que, para casos de elevada dimensionalidade, não seria comportável. Este método foca-se no fornecimento de explicações a nível local, de cada instância.

2.3.6 Localidade dos Explicadores

Também na literatura [1, 5, 12, 31, 44, 46, 51] é apresentada a distinção entre explicações locais e globais, sendo que os explicadores podem ter a capacidade de oferecer um ou ambos os tipos.

2.3.6.1 Explicadores Globais

Um explicador global oferece explicações para o comportamento do método de aprendizagem como um todo, a nível geral. Este avaliará o comportamento sistemático de um modelo sem que se foque em nuances locais.

Em [22], foi proposta a *Gradient Boosting Machine* (GBM) e como acessório foi também proposto o *Partial Dependence Plot* (PDP). A função do PDP é avaliar a correlação entre variáveis de um conjunto de dados, sendo que uma delas é a classe. Este método considera-se global, pois analisa o comportamento das características como um todo, por oposição à oferta de explicações para cada instância. Em primeiro lugar, é necessário escolher uma ou mais variáveis para analisar, sendo que para ser possível apresentar graficamente (*plot*) o resultado, o ideal é considerar apenas duas variáveis onde é possível ver claramente o resultado a duas dimensões. Se forem utilizadas três variáveis, o gráfico de dependência já poderá ter limitações. Depois de escolhidas, por suposição, duas variáveis, ir-se-á calcular a

influência de uma (variável de teste) sobre outra (variável alvo). Para isto, todas as outras variáveis do conjunto de dados são mantidas sem alterações. De seguida, itera-se a variável de teste sobre a sua gama de valores e, para cada valor diferente, faz-se a previsão da variável alvo. Por fim, para cada valor da variável de teste é calculada a média da variável alvo, dando origem a um gráfico que apresenta uma linha. Quando a variável é numérica, um eixo é a variável de teste e outro é a classe, sendo possível visualizar o comportamento da correlação.

2.3.6.2 Explicadores Locais

Um explicador local oferece explicações específicas para cada instância separadamente ou para um conjunto de instâncias. Este tipo de explicadores foca-se em encontrar nexos de causalidade para áreas específicas do modelo, podendo estas não ser válidas para o modelo como um todo. Os explicadores LIME [40] e SHAP [33], mencionados na secção 2.3.5.1, enquadram-se neste tipo.

O método *Individual Conditional Expectation* (ICE) foi proposto em [24] e tem por base o método PDP. O ICE consiste na avaliação, instância a instância, da influência de uma característica sobre a classe. O modo de operação deste método passa por, para cada linha do conjunto de dados, "congelar" todas as variáveis exceto a que se pretende analisar (variável de teste) e a classe (variável alvo). De seguida, faz-se variar a variável de teste pelos valores da sua gama e é feita uma nova previsão para cada valor considerado. Deste modo, são obtidos N pontos para cada instância onde este número representa a quantidade de valores experimentados e respetiva previsão. O resultado deste método é um gráfico que possui tantas linhas quanto instâncias do conjunto de dados, onde um eixo é a variável de teste e o outro é a classe.

Outro método que oferece explicações a nível local é o proposto em [20]. É adotada uma abordagem por sensibilidade, isto é, as explicações são extraídas através da observação do comportamento dos dados de saída, em função da manipulação dos dados de entrada. Para o caso de imagens, as perturbações feitas podem passar pela adição de ruído, da substituição de uma zona da imagem por valores de pixel constantes ou pela aplicação de um filtro de desfoque. Caso, depois de aplicadas as perturbações, o classificador atribua uma classe diferente da original ou tenha uma percentagem de certeza muito mais baixa, então significa que as zonas da imagem mais adulteradas apresentam um impacto significativo para o reconhecimento da classe. Em [20], são feitos testes sobretudo com imagens, mas este raciocínio pode ser aplicado a vários outros domínios.

2.3.7 Métricas de Avaliação

Em várias referências [1, 5, 12, 17, 23, 36, 44, 46, 51] é realçada a importância da aplicação de métricas para a XAI obtendo uma comparação sistemática entre as explicações e os métodos que as fornecem. As métricas são organizadas em três categorias: funcionais, baseadas em utilizadores e aplicacionais.

2.3.7.1 Métricas Funcionais

As métricas funcionais adotam uma abordagem objetiva, sendo caracterizadas pela ausência de intervenção de utilizadores. Por esta razão, estas métricas têm baixo custo, dado que podem ser automatizadas por um algoritmo, com as seguintes propriedades:

- Fidelidade – Avalia o quão certo é o explicador em relação ao modelo caixa-preta. Por exemplo, no caso do LIME [40], esta métrica corresponderia à taxa de acerto do modelo transparente usado para determinar a característica mais importante localmente quando comparado ao modelo caixa-preta.
- Complexidade Algorítmica – Avalia o quão complexo a nível computacional é o método que fornece as explicações. Caso este requeira elevada capacidade computacional esta métrica terá um valor baixo, caso contrário terá um valor alto.
- Consistência – Avalia a consistência do explicador na medida em que, para os mesmos dados de entrada, devem ser extraídas explicações semelhantes de modelos caixa-preta com dados de saída semelhantes. Caso as explicações variem entre modelos, mesmo que estes apresentem um resultado semelhante, revelará inconsistência. Se providenciarem explicações semelhantes, demonstrará consistência.

2.3.7.2 Métricas Baseadas em Utilizadores

As métricas baseadas em utilizadores requerem intervenção humana sobre as explicações e tentam avaliar alguns aspetos fundamentais das mesmas. Dado que o objetivo de uma explicação é facilitar a compreensão humana do modelo em questão, este tipo de avaliação é importante. Este tipo de métrica tem uma componente subjetiva, pelo que deverá ser realizada a avaliação com um número significativo de utilizadores. Alguns exemplos deste tipo de métrica são:

- Interpretabilidade – Avalia o quão fidedignas são as interpretações dos utilizadores às explicações que o modelo pretendia na verdade extrair, isto é, o quão capazes são os utilizadores de interpretar as explicações fornecidas.
- Eficiência – Avalia quanto tempo é que o utilizador requer para conseguir estabelecer um modelo mental do explicador, isto é, para interpretar o comportamento do modelo.
- Quantidade de informação – Avalia quanta informação é que uma explicação consegue expressar para um utilizador. Apesar desta métrica ter aparentemente um carácter objetivo, será uma mais valia a participação de utilizadores que tenham a capacidade de interagir com as mesmas, para ter uma perspetiva mais fidedigna com a realidade.

2.3.7.3 Métricas Aplicacionais

As métricas aplicacionais avaliam a experiência de utilizador como um todo, não só ao nível das explicações em si, mas também o contexto em que elas estão inseridas, nomeadamente a interface na qual são apresentadas. Alguns exemplos deste tipo de métrica são:

- Satisfação – Avalia o quão positiva foi a experiência do utilizador em interação com o sistema.

- Impacto de desempenho que o sistema de explicação forneceu – Avalia o quão importante foi a presença do sistema de explicação. Por exemplo, o quão o sistema ajudou os médicos a realizar diagnósticos.

- Capacidade de automatização – Avalia o quão possível ou até que ponto é que o trabalho manual dos utilizadores é necessário. É importante sobretudo quando a quantidade de explicações é elevada.

2.3.8 Taxonomia Adotada

A taxonomia adotada neste documento foi resultado da junção de várias referências que, quer realizando diretamente a sua taxonomia, quer salientando aspetos específicos da mesma, deram a sua contribuição. Apesar disso, esta não contempla o completo detalhe de outras fontes consultadas, já que este apresenta uma granularidade demasiado elevada para o intuito deste documento. Desta forma, a taxonomia aqui proposta é apenas uma versão resumida do detalhe que se pode obter consultando a literatura. A figura 2.9 apresenta o resumo desta mesma taxonomia.

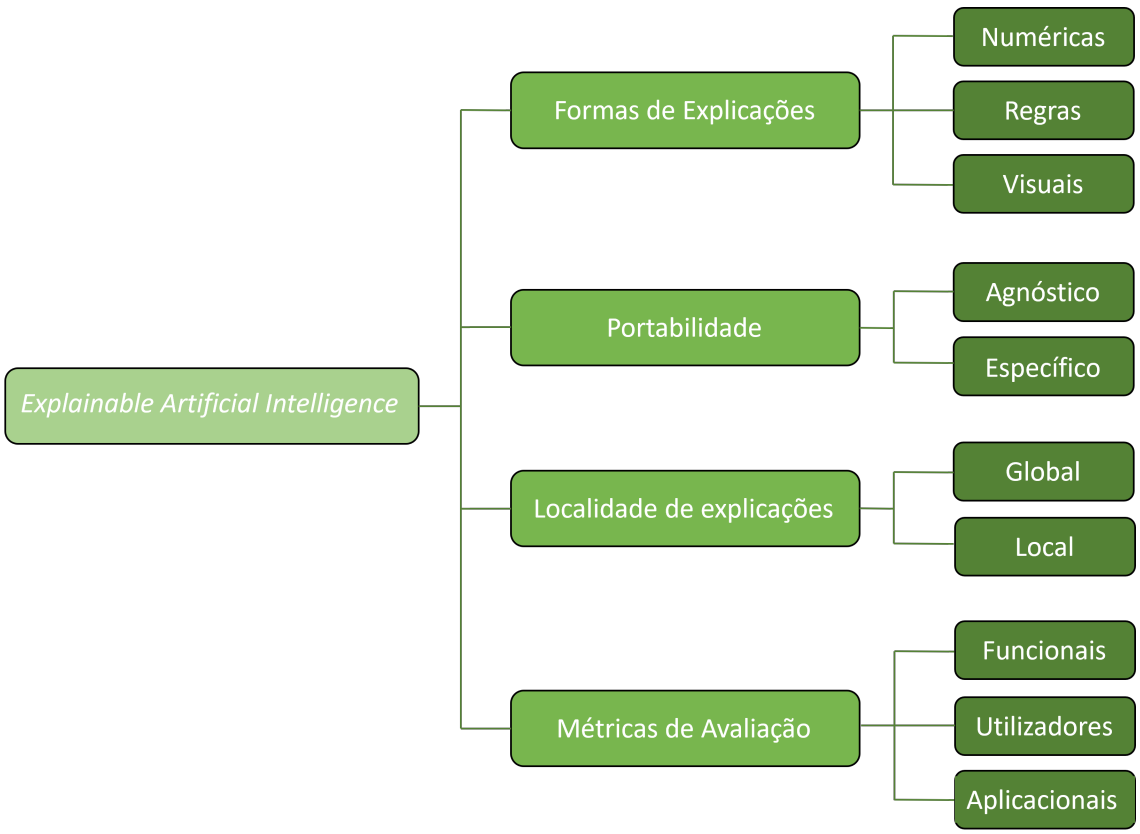


Figura 2.9: *Taxonomia de XAI considerada neste trabalho*

2.4 Trabalho Relacionado

Nesta secção apresentam-se métodos de obtenção de explicações de modelos e apresentadas algumas aplicações. A organização desta secção está de acordo com a taxonomia proposta anteriormente.

2.4.1 Métodos Agnósticos

Uma das formas mais simples de extração de explicações é através da *seleção de características* [27]. Este tipo de métodos, supervisionados ou não, realizam a escolha das características mais importantes segundo uma métrica. Estes métodos são agnósticos ao modelo, pois, mesmo os supervisionados, apenas necessitam da capacidade preditiva do modelo, isto é, não acedem a aspetos internos deste. O escopo do seu tipo de explicação é global, dado que, ao escolher as características mais importantes, não se estão a considerar especificidades de instâncias individuais, mas sim as mais importantes como um todo.

Em [19], é proposto um método de seleção de características que conduz à explicabilidade, o *K-Fold Feature Selection* (KFFS). O KFFS usa a técnica *K-Fold*, onde em cada *fold* efetua seleção de características utilizando um filtro genérico e regista as características selecionadas. Depois de feito este processo em todos os *folds*, selecionam-se as características que foram escolhidas acima de uma certa percentagem, dada como parâmetro de entrada do método. Foram realizados vários testes, sobretudo no domínio médico, como por exemplo na deteção de cancro da próstata, onde se pôde observar que de um total de 10 509 características cerca de 10 200 não foram selecionadas em nenhum *fold*, o que significa que o seu impacto é reduzido ou nulo para a deteção de cancro. Utilizando o classificador *Support Vector Machines* (SVM) [15] para o conjunto de dados *Diffuse large B cell lymphoma* (DLBCL), observou-se a menor taxa de erro com cerca de 25 características, o que corresponde a 0,24% da dimensionalidade total.

Na secção 2.3.5.1 foi introduzido o LIME proposto em [40]. Uma das métricas avaliada neste artigo é a fidelidade que os explicadores têm para com o classificador. Para isto foram selecionados dois conjuntos de dados de análise de sentimentos onde foram treinados os classificadores *Logistic Regression* e uma árvore de decisão. Estes modelos foram parametrizados por forma a utilizarem apenas 10 características para a atribuição de uma classe. De seguida, para cada instância do conjunto de teste, os explicadores testados declararam quais as características mais importantes e se estas incluíssem as 10 escolhidas pelo modelo, então a explicação seria fiel. O LIME foi o que obteve maior destaque nos dois conjuntos de dados, com uma taxa de fidelidade acima dos 90% em ambos, sendo que o segundo melhor tem sempre uma diferença de, pelo menos, 15%.

Em [8], é feito um estudo sobre a adoção do LIME por parte de médicos e comparadas as suas explicações com as dadas por profissionais. O conjunto de dados utilizado trata a deteção de sépsis (reação do corpo de forma exagerada a uma infeção) em pacientes. Foram feitas diversas avaliações sobre a satisfação dos profissionais, sendo que as notas variam de 1 a 5, tal como resumido na tabela 2.4.

Os autores concluíram que, para a maioria dos profissionais, a presença de explicações é

Tabela 2.4: Resumo da satisfação dos profissionais de saúde. Adaptado de [8]

Descrição	**Média**
Pontuação geral de satisfação em relação às explicações fornecidas pelo LIME	3,9
Pontuação de satisfação em relação às explicações fornecidas pelo LIME quando as variáveis escolhidas não se sobrepõem às três principais do médico	3,0
Pontuação de satisfação em relação às explicações fornecidas pelo LIME quando as variáveis escolhidas se sobrepõem às três principais do médico	4,3
Pontuação de confiança	3,4
Satisfação com a representação visual das explicações do LIME	3,8
Satisfação com a representação textual das explicações do LIME	2,6

bastante relevante e até coincidente com as suas visões, promovendo assim a confiança na utilização de técnicas de IA que ajudem de alguma forma ou até complementem o trabalho dos profissionais de saúde.

Em [18], é utilizado o LIME num contexto de eficiência energética de edifícios. O conjunto de dados utilizado foi extraído a partir de dados de um *Heating, Ventilation, and Air Conditioning* (HVAC) à base de água, onde a classe é o coeficiente de desempenho e as características são de três tipos, sendo eles:

- Exteriores – Por exemplo, a temperatura ambiente e grau de humidade;
- Operacionais – Tais como quociente de vazão e a temperatura de fornecida e devolvida;
- Temporais – Designa o mês, hora e tipo do dia.

Foram testados cinco modelos diferentes, sendo eles *Generalized Linear Models* (GLM), *Multi-Layer Perceptron* (MLP), SVM, RF e *eXtreme Gradient Boosting* (XGBoost), onde o que obteve maior taxa de acerto foi o RF com 95,4%, estando muito perto do XGBoost com 95,3%. Para conseguir avaliar o desempenho do LIME, foi proposta uma medida de confiança que se baseia nos coeficientes do modelo auxiliar usado para extrair as explicações, sendo que quanto mais coeficientes positivos houver e quanto maior o seu valor, mais fiável é a explicação. Por fim, o LIME é aplicado aos vários modelos testados e mostrada a explicação mais e a menos fiável dos modelos GLM, MLP e RF.

Em [28], foram testados conjuntos de dados de domínio médico onde se utilizaram vários métodos para extrair explicações dos classificadores utilizados, entre eles o LIME. O processo proposto neste artigo é dividido em quatro passos:

1. Construção do modelo;
2. Extração de explicações globais;
3. Extração de explicações locais;
4. Comprometimento de desempenho em função de interpretabilidade.

Na avaliação das explicações são consideradas duas métricas principais: fidelidade e monotonicidade. A primeira indica se o método avalia a importância das características de forma correta, isto é, se retirando uma característica considerada importante se verifica um impacto negativo no desempenho do modelo. A segunda avalia se ao adicionar cada característica, por ordem decrescente de importância, resulta num crescimento monótono do desempenho do modelo. Foram utilizados diversos classificadores onde, para cada conjunto de dados, foi avaliado o desempenho dos explicadores, nas métricas referidas e constatou-se que o LIME está sempre entre os métodos que têm melhor desempenho, sendo que, entre os seis conjuntos de dados considerados, apresentou desempenho de monotonicidade superior aos demais em três conjuntos.

Em [33], onde é proposto o SHAP, um dos testes realizados utilizou um conjunto de dados do domínio médico que se destina à deteção de doença. Para este problema, é mostrado que o SHAP apresenta um comportamento semelhante ao humano tendo considerado que as características ”febre“ e ”tosse“ contribuem para a presença de doença, ao contrário do LIME que considerou estas características como relevantes para a ausência de doença. Outro teste foi realizado no conjunto *MNIST* (dígitos manuscritos de 0 a 9), sugere que o SHAP tem a capacidade de selecionar os *pixels* relevantes para o reconhecimento dos dígitos (em particular do dígito ”8“).

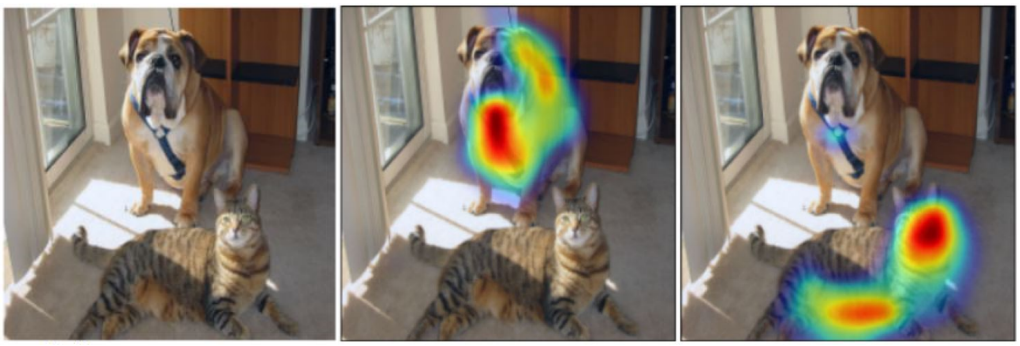
Em [38], foi utilizado o SHAP para prever os pacientes de alto risco a ter infarto cerebral (morte de tecido cerebral devido a mau fornecimento de sangue). Neste artigo, é utilizada informação real de um hospital e extraídas características de cinco fontes perfazendo 1534 instâncias e 1714 características. Para a realização de predições foi utilizado o XGBoost que atingiu *Area Under the Curve* (AUC) de 0,788 que foi considerado um valor satisfatório. Foi feita uma comparação do SHAP com o outro método que atribui importância a características cujo nome não foi mencionado e verificou-se que a sua avaliação das 20 características mais importantes coincidia em grande parte. Também se verificou que as características mais importantes estavam de acordo com a expectativa dos profissionais de saúde, exceto o “*A/G ratio*”. Depois de uma análise mais profunda, confirmou-se que esta variável de facto tinha um impacto significativo para a predição o que causou a adição de conhecimento para os profissionais de saúde que podem começar a fazer testes e a considerar esta variável futuramente.

2.4.2 Métodos Específicos

Em [47], é proposto o método Grad-CAM, o qual obtém explicações em forma de um *heatmap* apenas de redes neuronais convolucionais, sendo por isso, específico. O modo de operação do Grad-CAM passa por calcular os gradientes da camada de saída para a classe pretendida e posteriormente para as outras camadas. De seguida, para cada camada convolucional é feito o *global average pooling*, ou seja, é feita a média dos valores de cada mapa de características e posteriormente somados, sendo que, quanto mais alto for o seu valor mais importante é o mapa para a decisão da classe. Depois, é aplicada a função de ativação *ReLU* com o objetivo de eliminar os valores negativos, já que apenas queremos os valores que contribuem positivamente para a classe a ser analisada. Já que o resultado

do passo anterior não terá a mesma dimensão que a imagem de entrada terá de se fazer o *upsampling*, isto é, o aumento da resolução espacial para igualar à da imagem. O resultado é um *heatmap* onde é possível ter informação de que zonas da imagem mais contribuíram para a decisão feita pelo modelo.

Em [47], foram feitos diversos testes sendo utilizado mais que um conjunto de dados e com diferentes objetivos. Um dos objetivos selecionados foi a extração de explicações contrafactuais (*counterfactual*), isto é, indicar a porção de uma imagem que contribui para uma classe diferente da selecionada, tal como se observa na figura 2.10.



(a) *Imagem Original* (b) *Contrafactual 'Cat'* (c) *Contrafactual 'Dog'*

Figura 2.10: *Explicações contrafactuais utilizando Grad-CAM. Adaptado de [47]*

Uma outra forma comum deste tipo de explicações, ao invés de identificar as componentes que contribuem para uma classe diferente da selecionada, é alterar os dados de entrada de forma mínima para que outra classe seja identificada.

Em [52], é utilizado o Grad-CAM para a previsão do consumo de energia e são feitas comparações com o SHAP a nível de extração de explicações. O conjunto de dados utilizado comporta dados do consumo de hora em hora, em três cidades do Panamá, apresentando 43200 instâncias e 65 características. Sendo que o objetivo é prever o consumo de energia, este é um problema de regressão. As explicações locais extraídas ganham a forma de *heatmaps* no gráfico de variação entre duas características, como é possível observar na figura 2.11.

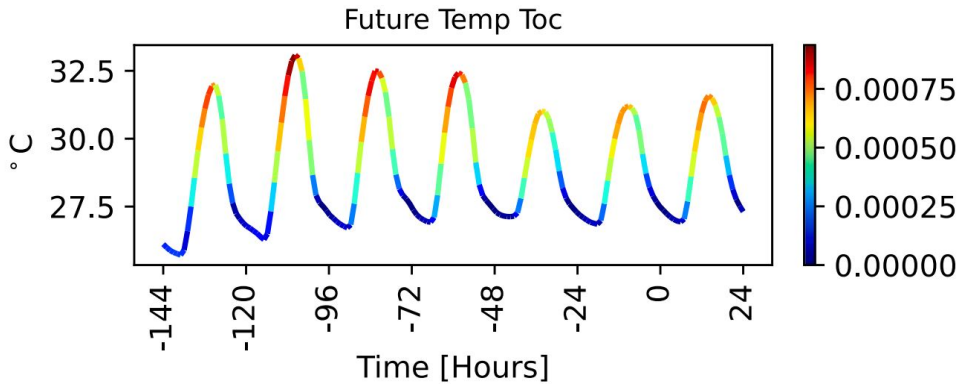


Figura 2.11: *Explicações locais através de heatmapping utilizando Grad-CAM. Adaptado de [52]*

Foi possível constatar que, quanto maior a temperatura ambiente, maior é a sua importância

para a demanda energética. Muitas indústrias, como por exemplo a alimentar, necessitam de manter os seus produtos a baixas temperaturas, logo esta é uma característica que, em princípio, fará aumentar a demanda energética. Para a extração de explicações globais, ambos os métodos indicaram que as características "Demand", "Diff demand" e "Future Temp Toc" apresentam elevada importância a nível geral.

Na secção 2.3.5.2 foi abordado o LRP proposto em [6]. Neste artigo são feitos diversos testes principalmente quando os dados de entrada são imagens. Um dos exemplos está ilustrado na figura 2.12.

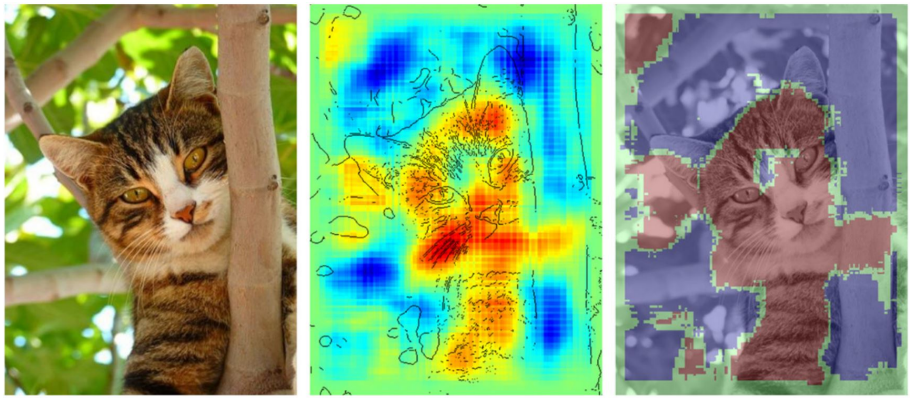


Figura 2.12: *Explicações locais através de heatmapping utilizando LRP. Adaptado de [6]*

Na figura 2.12, a imagem à esquerda é a original, a do meio é a resultante da aplicação do LRP sobreposto à imagem de contornos e a da direita é a imagem produzida após a realização de binarização pixel a pixel do *heatmap* sobreposto à imagem original. Também foram feitos testes sobre o conjunto de dígitos manuscritos MNIST.

Em [10] foi utilizado o LRP para a deteção de *Alzheimer* através de imagens médicas. O conjunto de dados utilizado é da *Alzheimer's Disease Neuroimaging Initiative* e apresenta 962 exames de 193 pacientes, dos quais 475 são casos positivos e 494 são casos negativos. Em primeiro lugar, foi treinada uma rede neuronal convolucional com quatro camadas convolucionais seguidas de duas camadas totalmente conectadas. O *kernel* utilizado apresentava dimensões $3 \times 3 \times 3$ e as duas últimas camadas apresentavam 128 e duas unidades, respetivamente. Verificou-se no conjunto de teste uma taxa de acerto de 87,96%. Numa fase seguinte utilizou-se o LRP para a extração de explicações através de *heatmaps* indicando em que zona da imagem é que se identificou anomalia, tal como se observa na figura 2.13.

Também foi feita uma análise de relevância de características utilizando LRP e chegou-se à conclusão que as mais importantes são as seguintes: "Cerebral white matter", "Cerebellum" e "Mid. frontal gyr.".

2.4.3 Métodos Transparentes

Em [32], foi proposto o *Explainable Boosting Machine* (EBM), que providencia elevada interpretabilidade e taxa de acerto equiparável a métodos mais complexos os quais tipicamente, como referido na secção 2.3.2, apresentam uma diminuição de transparência. Este método basea-se em *Generalized Additive Model* (GAM), um modelo que cria funções para

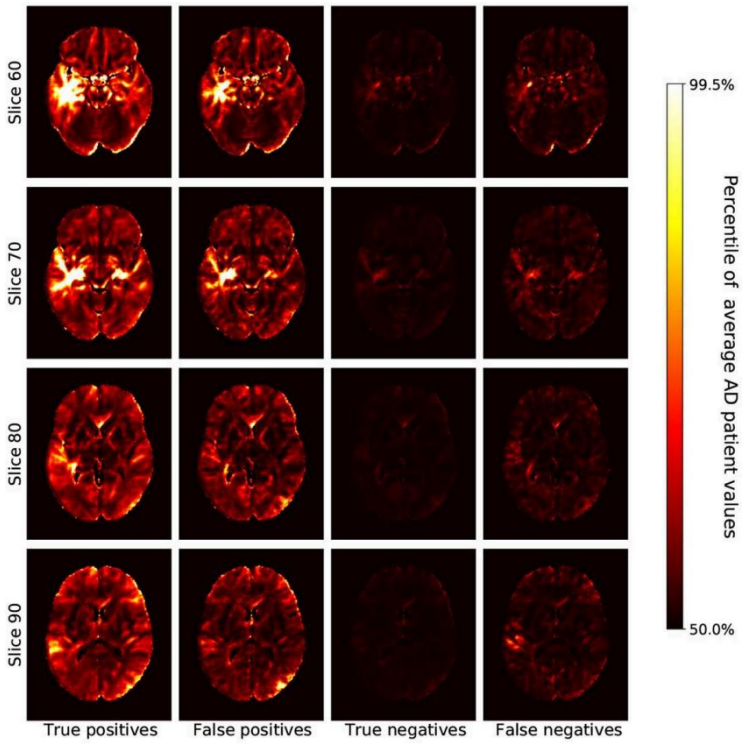


Figura 2.13: *Explicações de Alzheimer através de heatmapping utilizando LRP. Adaptado de [10]*

cada característica na forma de

$$g(E(Y)) = \beta_0 + f_1(x_1) + f_2(x_2) + ... + f_m(x_m), \tag{2.1}$$

onde $g(E(Y))$ é a previsão e $f_m(x_m)$ é uma função que descreve a característica m. Em adição ao GAM, o EBM considera a possibilidade de existência de correlações entre características, sendo expressas como

$$g(E(Y)) = \sum f_i(x_i) + \sum f_{ij}(x_i, x_j), \tag{2.2}$$

onde $f_{ij}(x_i, x_j)$ é a função que descreve a correlação entre a característica x_i e x_j. Existem diversas maneiras de implementar o EBM, sendo uma delas através da utilização de árvores de decisão. Este processo passa pela criação de uma árvore de decisão de baixa profundidade para cada característica de forma iterativa até que algum critério de paragem seja atingido, como por exemplo o máximo número de iterações ou um nível de desempenho aceitável. Após o seu treino, é possível a extração de explicações globais na forma de importância global de cada característica ou explicações locais onde é apresentada de que forma é que o valor de cada característica de uma instância contribuiu para a previsão. No artigo mostra-se que para os conjuntos de dados testados, para tarefas de regressão ou classificação o desempenho é muito semelhante ao do classificador RF.

Em [26], aborda-se a deteção de *angina pectoris* em mulheres, uma doença que é caracterizada pelo desconforto peitoral devido a problemas cardíacos. Neste artigo são testados

seis modelos: EBM, *CatBoost*, *AdaBoost*, XGBoost, *LightGBM* e *Logistic Regression*. O EBM foi o que obteve a melhor taxa de acerto de 95,5%, sendo que o segundo melhor foi o *CatBoost* com 94,5% e o que obteve pior desempenho foi o *AdaBoost* com 85% de taxa de acerto. Também através do EBM descobriu-se que a característica mais relevante era o histórico de ataques cardíacos na família seguida da idade.

Outro estudo foi feito em [11], prevendo a presença de *Alzheimer* utilizando EBM. O conjunto de dados utilizado comporta dados desde 2006 a 2022 no qual se consideram três tipos de características: escalas de cognição, medições através de ressonância magnética e dados sócio-demográfico-clínicos. Este conjunto de dados está organizado pelos anos de acompanhamento, sendo estes de 1 a 5, e a quantidade de pacientes aumenta consoante o tempo. A taxa de acerto ao longo do tempo é apresentada na figura 2.14.

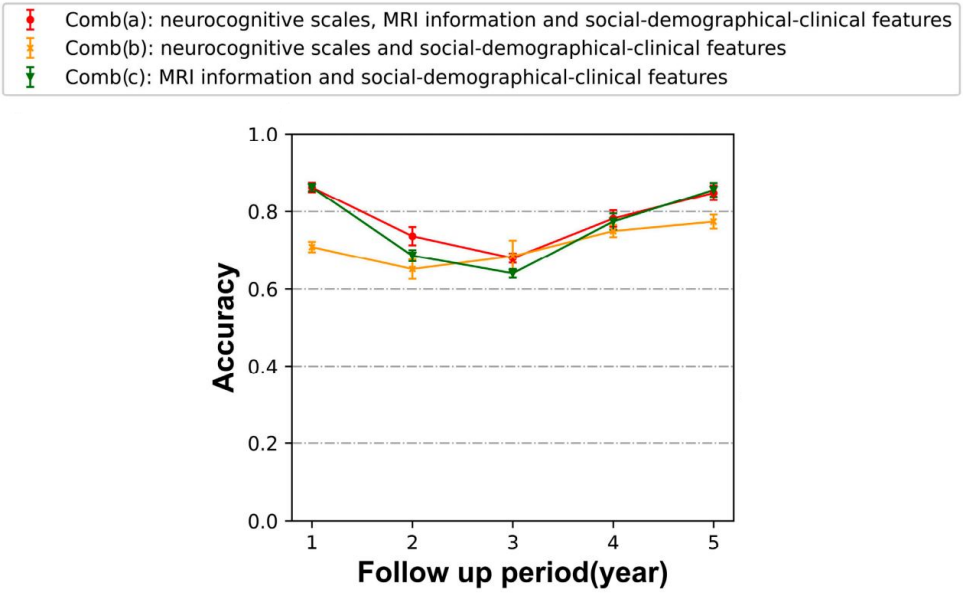


Figura 2.14: *Taxa de acerto ao longo do tempo. Adaptado de [11]*

Também foi feita uma análise de importância de características onde se pode concluir que a curto e longo prazo as características mais importantes estão relacionadas com medições obtidas através de ressonância magnética, enquanto que a médio prazo estão relacionadas com a escala de cognição.

Em [14], é pretendida a explicação da expressividade de genes na deteção do cancro do ovário utilizando algoritmos genéticos e regras difusas. O conjunto de dados utilizado contém 21 instâncias, 6 classes e cerca de 45000 características onde os valores de cada coluna são de domínio numérico e representam a expressividade de um gene. Aplicando um filtro onde são apenas considerados os genes que apresentam expressividade acima de 50, o número desce para cerca de 9000 características. De seguida, é feita uma análise diferencial de expressividade dos genes para poder perceber, estatisticamente, que mudanças podem ser consideradas significativas. Numa fase seguinte utilizaram-se regras difusas para realizar a classificação através de *Fuzzy Inference System* (FIS). Aqui são considerados três níveis de expressividade para cada gene: baixa, média e alta. A expressividade média é o valor médio do gene. As regras são definidas por curvas gaussianas equidistantes, onde um extremo é

a expressividade baixa, o outro é a expressividade alta e no centro está a expressividade média, tal como está ilustrado na figura 2.15.

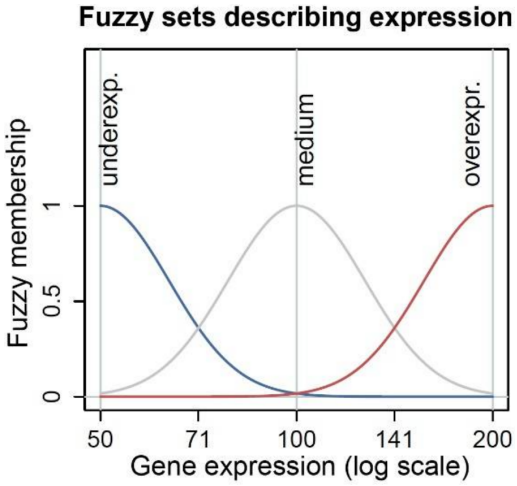


Figura 2.15: *Regras de lógica difusa para a expressividade de genes. Adaptado de [14]*

O classificador FIS fornece interpretabilidade, dado que cada instância poderá ser representada sob a forma de um conjunto de regras tal como apresentado na tabela 2.5.

Tabela 2.5: Modo de operação do FIS. Adaptado de [14]

Premissa (Se)	**Consequência (então)**
gene1 é médio e gene2 é alto e ... geneN é baixo	classe é KE

Devido à elevada dimensionalidade dos dados decidiu-se filtrar algumas das características utilizando algoritmos genéticos, onde este possui os seguintes atributos:

- Indivíduo – Lista de dimensão fixa que representa um conjunto de características.
- População inicial – Gerada aleatoriamente.
- Cruzamento – Um novo indivíduo é gerado escolhendo aleatoriamente características dos pais.
- Seleção – Cada indivíduo tem uma probabilidade de ser selecionado para cruzamento proporcional à sua aptidão.
- Mutação – Tem 50% de probabilidade de ocorrer e é realizada somando uma unidade ao gene escolhido.
- Elitismo – Em cada geração, um conjunto de indivíduos é reintroduzido.
- Imigração – Em cada geração é gerado um conjunto novo de indivíduos.
- Função de Aptidão:

$$\text{Aptidão} = \text{Taxa de acerto} \times 0{,}5 + \text{Simpl} \times 0{,}3 + \text{Inter} \times 0{,}2. \tag{2.3}$$

Onde taxa de acerto representa a percentagem de acertos do classificador, *Simpl* é um valor de 0 a 1 inversamente proporcional ao número de regras e *Inter* é um valor de 0 a 1 que

avalia quantos genes são relevantes para a tarefa de classificação. São premiados aqueles indivíduos que apresentam genes que se conhecidos com influência. O critério de paragem é composto pelas condições: o número máximo predefinido de gerações ser atingido ou a população de elite ter menos de 3 indivíduos diferentes.

A estrutura proposta em [14] proporciona a interpretabilidade, dado que o classificador FIS é naturalmente interpretável devido à utilização de regras e ao seu domínio de valores reduzido neste contexto (baixo, médio e alto). Para além disso, a utilização de algoritmos genéticos para a seleção de características, é uma forma de obtenção de explicações globais indicando quais as características mais importantes para o problema como um todo, tal como a tarefa de seleção de características.

Após realizados testes, chegou-se a uma população final de 72 indivíduos com uma semelhança de 78% entre si e que obtiveram uma taxa de acerto de 100% onde dos 10 genes mais frequentes sabe-se *a priori* que 7 deles têm influência significativa. No final, foram selecionados 10 conjuntos de regras utilizando 10 genes que não apresentaram qualquer erro para o conjunto de dados utilizado. É mencionado que isto pode ser causado pela baixa quantidade de instâncias, reconhecendo que devido ao seu custo de aquisição, significativo em muitos contextos do domínio médico, torna-se difícil de obter um conjunto de dados com maior representatividade.

3 Solução Proposta

Este capítulo descreve os conjuntos de dados e as métricas de avaliação no âmbito da abordagem seguida neste projeto e respetivo contexto. Na secção 3.1 abordam-se os dados do domínio médico, o foco deste trabalho, referindo alguns problemas comuns desta área. De seguida, na secção 3.2 são apresentadas as formas de avaliação adotadas para poder aferir o desempenho global do modelo. Na secção 3.3 é descrita a abordagem adotada e a sequência de tarefas necessárias desde a escolha dos dados até à extração de explicações do modelo. Na secção 3.4, é referido como implantar o sistema desenvolvido num contexto real, onde os modelos já estão treinados e se pretende o diagnóstico de novos pacientes.

3.1 Dados do Domínio Médico

A *eXplainable Artificial Intelligence* (XAI) pode ser aplicada nos mais variados domínios, sendo que a sua utilização poderá ter um carácter mais ou menos imperativo dependendo do contexto. Um dos casos onde o seu uso se torna imperativo é a extrema necessidade de estabelecer nexos de causalidade entre os dados de entrada e os dados de saída. Muitas vezes, esta necessidade está relacionada com a responsabilização da IA, ou seja, para atribuir a algum algoritmo a responsabilidade de realizar escolhas num certo domínio, este também deverá ter a capacidade de as justificar. Isto acontece devido às consequências que uma escolha errada poderá ter, existindo contextos nos quais um erro pode não apresentar grande custo e outros onde o custo é elevado. Um dos mais característicos exemplos de custo elevado na presença de erros é o domínio médico. Aqui, o custo pode não assumir meramente uma forma monetária, mas também biológica, isto é, o objeto deste domínio são seres vivos, pelo que no limite pode pôr em causa a vida de um indivíduo.

No domínio médico, os dados podem assumir principalmente dois tipos: imagem ou tabular. No caso das imagens, estas podem ser: radiografias, imagens de ressonância magnética, ultra-sons, entre outros. Quando aplicada a XAI neste domínio, esta assume muitas vezes a forma de saliência, indicando que zonas da imagem é que levaram a determinada decisão. Já quando o tipo de domínio é tabular, predomina a extração de explicações através da atribuição de importância a cada característica ou da seleção das mais importantes. Esta extração também pode ser vista como uma forma de saliência, mas neste caso os atributos

não são *pixels*, podendo apresentar um domínio fora deste escopo, por exemplo, o dos números reais ou não numérico.

Para a realização de experiências, foram selecionados dois conjuntos de dados provenientes de diferentes contextos, que serão agora descritos.

3.1.1 Deteção de Alzheimer

Foram consideradas diversas fontes para obter dados que envolvam a deteção da doença de Alzheimer. Um deles encontra-se em [30]. Este conjunto de dados contém imagens de ressonância magnética pré-processadas. O seu objetivo é a deteção de 4 níveis de demência, sendo eles: "Non Demented", "Very Mild Demented", "Mild Demented" e "Moderate Demented". Dados sobre esta doença em formatos tais como imagens do cérebro e dados biométricos genéticos são disponibilizados no repositório [4]. Neste repositório estão disponíveis diversos conjuntos de dados que visam o estudo e melhoramento de técnicas para a deteção de Alzheimer. Outro repositório que assenta sobre a deteção de Alzheimer é [39]. Em [29], é usado um conjunto de dados deste repositório que utiliza dados biométricos volumétricos com o intuito de melhorar a deteção de Alzheimer. Outro repositório que apresenta conjuntos de dados para a deteção desta doença em diversos formatos é [48]. O conjunto de dados escolhido foi o de deteção de Alzheimer através de escrita introduzido em [13], que se denominou por *Diagnosis AlzheimeR WIth haNdwriting* (DARWIN). Foi escolhido este devido a diversos fatores, um deles sendo a novidade do mesmo, dado que o tipo de dados mais comum baseia-se em imagens de ressonância magnética ou em expressão de genes, pelo que foi considerado de interesse explorar este tipo de dados. Um outro motivo foi o da sua fácil exploração, uma vez que dados de imagens ou expressão de genes apresentam uma dimensionalidade muito maior e poderão oferecer maior dificuldade no seu tratamento. Para além da dimensionalidade, o tamanho de outros conjuntos de dados pode chegar a 1,5 GB brutos ou mais, como é o caso de *OASIS-1* presente em [39], o que requer bastante poder computacional para realizar análises dos resultados sem ficar fortemente limitado pelo tempo. Alguns dos repositórios mencionados necessitam que seja submetido um pedido de acesso aos dados, tal não se verifica para o conjunto de dados escolhido, sendo a sua facilidade de aquisição outro fator para a sua escolha.

Uma vez escolhido o conjunto de dados a usar, este foi analisado com maior profundidade, sendo que os detalhes do conjunto de dados DARWIN são os seguintes:

- Número de instâncias – 174;
- Número de características – 451;
- Número de classes – 2;
- Valores omissos – Não.

Para a construção deste conjunto de dados, 174 indivíduos realizaram 25 tarefas, cuja descrição é apresentada na tabela 3.1.

Tabela 3.1: Descrição das tarefas requisitadas no conjunto de dados DARWIN

#	**Descrição**	**Categoria**
1	Escrever a assinatura	M
2	Ligar dois pontos horizontalmente, continuamente por 4 vezes	G
3	Ligar dois pontos verticalmente, continuamente por 4 vezes	G
4	Refazer um círculo (6cm de diâmetro), continuamente por 4 vezes	G
5	Refazer um círculo (3cm de diâmetro), continuamente por 4 vezes	G
6	Copiar as letras 'l', 'm' e 'p'	C
7	Copiar as letras nas linhas adjacentes	C
8	Escrever cursivamente uma sequência de 4 letras minúsculas 'l', num único movimento suave	C
9	Escrever cursivamente uma sequência de 4 bigramas minúsculos 'le', num único movimento suave	C
10	Copiar a palavra 'foglio'	C
11	Copiar a palavra 'foglio' em cima de uma linha	C
12	Copiar a palavra 'mamma'	C
13	Copiar a palavra 'mamma' em cima de uma linha	C
14	Memorizar as palavras 'telefono', 'cane' e 'negozio' e reescrevê-las	M
15	Copiar o inverso da palavra 'bottiglia'	C
16	Copiar o inverso da palavra 'casa'	C
17	Copiar seis palavras (regulares, não regulares, palavras inexistentes) nas caixas apropriadas	C
18	Escrever o nome do objeto na imagem (uma cadeira)	M
19	Copiar os campos de um vale postal	C
20	Escrever uma frase simples por meio de um ditado	M
21	Refazer uma forma complexa	G
22	Copiar um número de telefone	C
23	Escrever um número de telefone por meio de um ditado	M
24	Desenhar um relógio, com todas as horas e pôr os ponteiros às 11:05	G
25	Copiar um parágrafo	C

As letras da coluna "Categoria" indicam o tipo de capacidade a ser testada, tal como está referido a seguir:

- (M) – Memória e ditado;
- (G) – Gráfico;
- (C) – Cópia.

Cada tarefa apresenta individualmente 18 características. Estas são as mesmas para todas as tarefas, mas cada tarefa possui valores diferentes específicos. Estas características incluem o tempo da caneta no ar, a velocidade média da caneta no papel, número de vezes que a caneta está em contacto com o papel, o tempo total, entre outros. Temos 450 características que dizem respeito ao teste em si ($18 \times 25 = 450$) e uma característica designada por 'ID' para identificar cada indivíduo.

O significado das características extraídas apresenta-se na tabela 3.2.

Tabela 3.2: Descrição das características extraídas por cada tarefa no conjunto de dados DARWIN

#	**Designação**	**Descrição**
1	*Total Time*	Tempo total gasto para completar a tarefa
2	*Air Time*	Tempo gasto em movimentos com a caneta fora do papel
3	*Paper Time*	Tempo gasto em movimentos com a caneta no papel
4	*Mean Speed on-paper*	Rapidez média dos movimentos no papel
5	*Mean Speed in-air*	Rapidez média dos movimentos fora do papel
6	*Mean Acceleration on-paper*	Aceleração média dos movimentos no papel
7	*Mean Acceleration in-air*	Aceleração média dos movimentos fora do papel
8	*Mean Jerk on-paper*	Média do arrasto em movimentos no papel. Arrasto é a variação da aceleração em função do tempo
9	*Mean Jerk in-air*	Média do arrasto em movimentos fora do papel
10	*Pressure Mean*	Média dos níveis de pressão exercidos pela ponta da caneta
11	*Pressure Var*	Variância dos níveis de pressão exercidos pela ponta da caneta
12	*GMRT on-paper*	Generalização do tremor relativo médio
13	*GMRT in-air*	Generalização do tremor médio calculado em movimentos fora do papel
14	*Mean GMRT*	Média das duas métricas anteriores
15	*Pendowns Number*	Número total de vezes que a caneta pousou no papel durante a execução de uma tarefa
16	*Max X Extension*	Máxima extensão gravada ao longo do eixo X
17	*Max Y Extension*	Máxima extensão gravada ao longo do eixo Y
18	*Dispersion Index*	O índice de dispersão avalia o quão disperso é o traço, qual a porção de papel ocupada pelo traço em relação a um pedaço de papel

O conjunto de dados apresenta um problema de classificação binária e a distribuição de classes encontra-se ilustrada na figura 3.1.

Figura 3.1: *Distribuição das classes para o conjunto de dados DARWIN para deteção de Alzheimer*

Como é possível observar na figura 3.1, existe equilíbrio entre a classe dos negativos 'H' (Healthy) e a classe dos positivos 'P' (Pacient), apresentando praticamente o mesmo número

de exemplos.

3.1.2 Deteção de Cancro

Outro conjunto de dados considerado trata a deteção de cancro no cérebro através de imagens. O objetivo deste conjunto de dados é detetar que tipo de doença é que está presente numa dada imagem.

As características gerais deste conjunto de dados são:

- Número de instâncias – 3064;
- Resolução espacial da imagem – $512 \times 512 \times 1$ (Imagem em tons de cinzento);
- Número de bits por canal – 12;
- Número de classes – 3.

As classes existentes neste conjunto são as seguintes:

- 1 – Meningioma;
- 2 – Glioma;
- 3 – Pituitary tumor (Tumor Hipofisário).

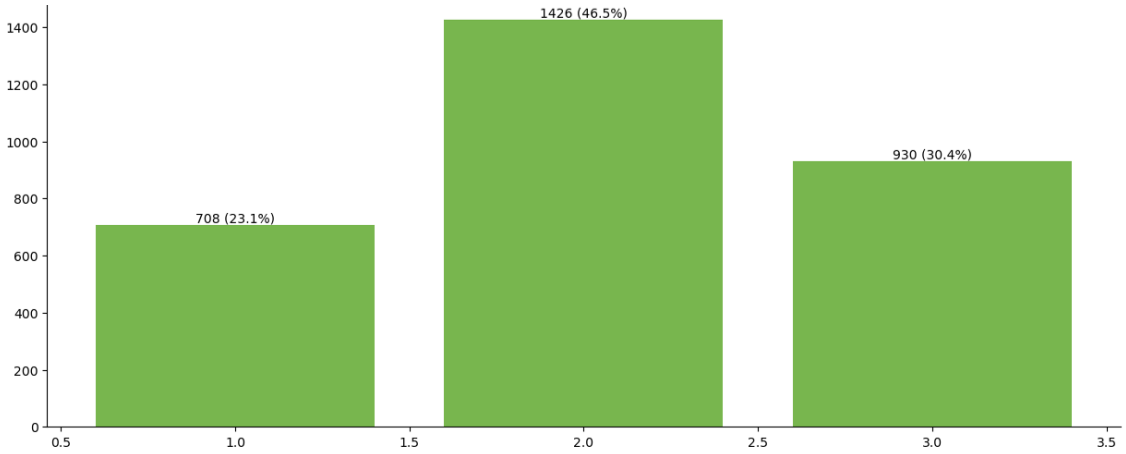


Figura 3.2: *Distribuição das classes para o conjunto de dados para a deteção de cancro no cérebro*

Na figura 3.2, observamos uma grande discrepância entre as 3 classes presentes, nomeadamente a classe "2" (46,8%) apresenta uma frequência de ocorrência muito maior que a 1 (23,2%) e que a 3 (30%).

3.2 Métricas de Avaliação

Nesta secção descrevem-se as formas de avaliar os modelos em relação ao seu desempenho. Esta avaliação fará uso de diversas métricas que serão agora enunciadas.

Quando um modelo de classificação realiza aprendizagem sobre um conjunto de dados, muitas das vezes o seu resultado não é perfeito, isto é, não obtém 100% de taxa de acerto. Quando o desempenho do modelo não é perfeito, é possível a criação de diversas métricas que não incluem meramente a taxa de acerto que poderão fornecer informação sobre os tipos de erro do classificador.

Dentro da tarefa de classificação é possível dividir a mesma em duas categorias: binária, quando o conjunto de dados possui duas classes, e multi-classe, quando o conjunto de dados possui mais que duas classes. Caso o número de classes não seja muito elevado, por exemplo, menor que 30, uma boa forma de ter noção geral do desempenho do modelo é através da matriz de confusão, sendo que mesmo quando se atingem elevados números de classes é possível recorrer a uma matriz à base de cor por forma a tornar mais intuitiva a sua interpretação. A matriz de confusão apresenta uma dimensão $C \times C$, onde C é o número de classes existentes, sendo indicado nas linhas a classe verdadeira e nas colunas a classe predita. Na tabela 3.3 está representada a meta-matriz para $C = 2$ (classificação binária).

Tabela 3.3: Matriz de confusão para classificação binária

		Classe Predita	
		Positive	Negative
Classe Verdadeira	Positive	TP	FN
	Negative	FP	TN

Muitas vezes na classificação binária, dado que apenas existem duas classes, diz-se que uma é o caso negativo e outra é o caso positivo. Nesta matriz é possível constatar a presença de 4 símbolos, sendo eles:

- True Positive (TP) – Número de casos positivos classificados como positivos
- False Positive (FP) – Número de casos negativos classificados como positivos
- False Negative (FN) – Número de casos positivos classificados como negativos
- True Negative (TN) – Número de casos negativos classificados como negativos

Quanto maiores forem os números fora da diagonal principal da matriz, maior será o erro. Para o caso de três classes, a matriz de confusão está representada pela tabela 3.4, onde a classe verdadeira é identificada depois do carácter "-". Por exemplo, um caso que é da classe C1, mas que foi classificado como sendo da classe C3 é representado por: C3-1. Esta análise é generalizável para casos com mais de três classes.

Tabela 3.4: Matriz de confusão para classificação multi-classe

		Classe Predita		
		Classe 1	Classe 2	Classe 3
Classe Verdadeira	Classe 1	C1	C2-1	C3-1
	Classe 2	C1-2	C2	C3-2
	Classe 3	C1-3	C2-3	C3

A taxa de acerto pode ser calculada a partir da matriz de confusão somando os números da diagonal principal da matriz e dividindo pela soma de todos os elementos da matriz. Para o caso da classificação binária pode ser descrita da seguinte forma:

$$\text{Taxa de Acerto} = \frac{TP + TN}{TP + FP + FN + TN}. \tag{3.1}$$

A classificação binária dispõe mais métricas, tais como: precisão (*precision*), cobertura (*recall*), taxa de falsos positivos, taxa de falsos negativos e *F1-Score*. Em alguns casos, estas métricas também são aplicadas num contexto multi-classe, no entanto, para isso ser possível implica sempre algum tipo de binarização, como por exemplo *one versus the rest*, onde é analisada uma classe em separado e todas as outras são aglutinadas e vistas como uma única classe.

A precisão pretende responder à pergunta "*Que percentagem dos casos classificados como positivos é que são, de facto, positivos?*" e pode ser descrita pela seguinte equação:

$$\text{Precisão} = \frac{TP}{TP + FP}. \tag{3.2}$$

A cobertura pretende responder à pergunta "*Que percentagem dos casos que são de facto positivos, é que foram classificados como positivos?*" e pode ser descrita pela seguinte equação:

$$\text{Cobertura} = \frac{TP}{TP + FN}. \tag{3.3}$$

A taxa de falsos positivos, cujo complementar é a *especificidade* que pode ser calculada por (1 - taxa de falsos positivos), pretende responder à pergunta "*Que percentagem dos casos negativos foram classificados como positivos?*" e pode ser descrita pela seguinte equação:

$$\text{Taxa de Falsos Positivos} = \frac{FP}{FP + TN}. \tag{3.4}$$

A taxa de falsos negativos, cujo complementar é a *sensibilidade* que pode ser calculada por (1 - taxa de falsos negativos), pretende responder à pergunta "Que percentagem dos casos positivos é que foram classificados como negativos?" e pode ser descrita pela seguinte equação:

$$\text{Taxa de Falsos Negativos} = \frac{FN}{FN + TP}. \tag{3.5}$$

O *F1-Score* é a média harmónica da precisão e da cobertura e pode ser descrito pela seguinte equação:

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Cobertura}}{\text{Precisão} + \text{Cobertura}}. \tag{3.6}$$

Outras duas métricas utilizadas na classificação binária são as curvas Precisão-Cobertura, também chamadas de *Precision-Recall* (PR) e *Receiver Operating Characteristic* (ROC).

As curvas de Precisão-Cobertura, tal como o próprio nome indica, relacionam a precisão e a cobertura onde um valor alto de precisão está relacionado a uma baixa taxa de falsos positivos e um valor alto de cobertura está relacionado com baixa taxa de falsos negativos. O desempenho ideal seria uma curva que alcançasse o canto superior direito, indicando uma alta precisão e uma alta cobertura. Um exemplo desta curva está na figura 3.3.

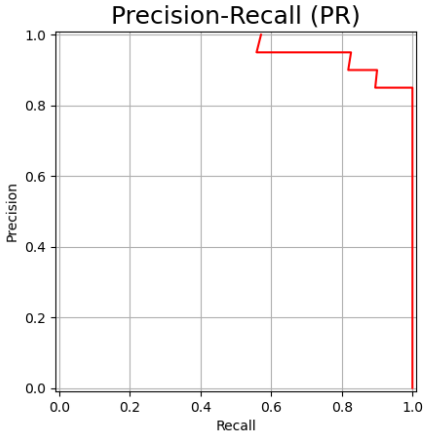


Figura 3.3: *Exemplo de uma curva de Precision-Recall*

As curvas ROC relacionam a taxa de verdadeiros positivos (*sensivity*/sensibilidade) com a taxa de falsos positivos (*specificity*/especificidade). Também é possível utilizá-las para casos multi-classe, escolhendo duas quaisquer classes e declarando uma como a positiva e outra como a negativa. O desempenho ideal seria uma curva que alcançasse o canto superior esquerdo, indicando alta *sensivity* e baixa *specificity*. Um exemplo desta curva pode ser encontrado na figura 3.4.

Sobre as curvas PR e ROC está associada uma outra métrica designada por *Area Under the Curve* (AUC), que designa a área por baixo da curva. Em ambos os casos, quanto maior for o valor desta métrica, melhor é o desempenho do método de aprendizagem.

3.3 Abordagem Proposta

Nesta secção será apresentada a sequência de ações tomadas para atingir o resultado final e detalhada cada componente. Para além disso, serão apresentados os modos de operação de alguns modelos usados para a concretização da arquitetura.

Em primeiro lugar, pensou-se nas componentes principais que o sistema a desenvolver teria de possuir para ser possível todo o processo pretendido. Chegou-se à conclusão que

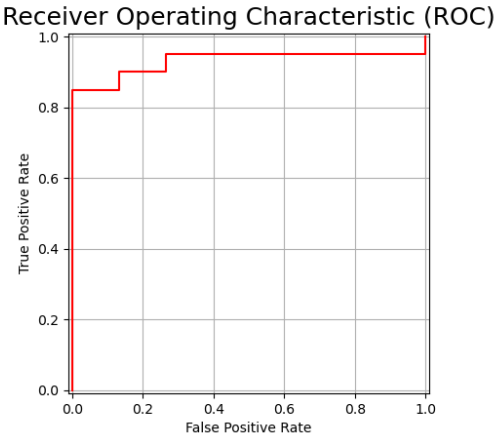


Figura 3.4: *Exemplo de uma curva de ROC*

existem duas arquiteturas diferentes que podem ser utilizadas: a que prevê a utilização de um explicador agnóstico ou específico ao modelo e a que prevê um modelo transparente. Para a representação visual da arquitetura, foram elaborados dois diagramas de blocos. O diagrama presente na figura 3.5 representa a arquitetura para explicadores agnósticos e específicos e o diagrama ilustrado na figura 3.6 usado no caso de explicadores transparentes.

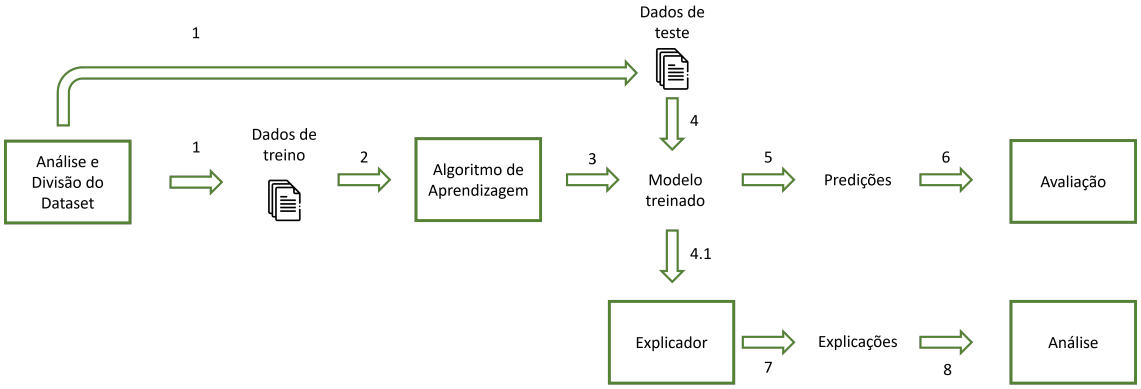


Figura 3.5: *Diagrama de blocos do sistema para explicadores agnósticos e específicos*

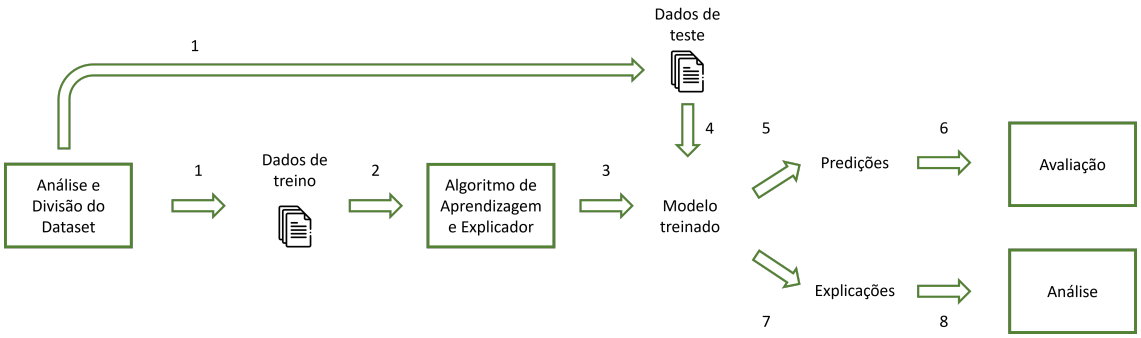


Figura 3.6: *Diagrama de blocos do sistema para explicadores transparentes*

A seguir estão explicados, passo a passo, os processos envolvidos nas arquiteturas apresentadas na figura 3.5 e na figura 3.6.

1. Em ambos os casos, começa-se com a aquisição de um conjunto de dados, neste caso, de domínio médico. Este conjunto de dados é analisado de modo a perceber o seu contexto e o significado de cada característica quando possível. Também é feita uma análise estatística dos seus atributos essenciais, tais como a sua quantidade de instâncias e características e o balanceamento das classes. Também neste passo é feita a divisão do conjunto de dados em treino e teste.

2. De seguida, os dados de treino são injetados no algoritmo de aprendizagem para ser possível realizar o treino do mesmo.

3. Segue-se o treino do algoritmo. Este processo é o mais complexo computacionalmente e esta complexidade cresce com a dimensionalidade e tamanho do conjunto de dados. Dependendo do método, este poderá adaptar-se melhor ou pior ao escalamento.

4. São utilizados os dados de teste sobre o modelo treinado de modo a ser possível fazer as predições sobre estes.

4.1. Este passo é apenas pertencente aos explicadores agnósticos e específicos ao modelo. Para este tipo de Explicadores é necessário que lhes seja injetado o método de aprendizagem ou algum atributo deste que lhe permita extrair explicações. Muitas vezes, esta injeção assume a forma do limiar de decisão do método de aprendizagem e não o método como um todo.

5. Realização de predições sobre o conjunto de teste tendo por base o modelo previamente treinado.

6. Uma vez feitas as predições sobre o conjunto de teste, é possível proceder à sua avaliação, tendo por base as métricas já mencionadas.

7. O foco orienta-se para a XAI e é feita a extração de explicações usando um Explicador agnóstico e, portanto, separado do modelo no caso da figura 3.5 ou usando o próprio método de aprendizagem que também é capaz da extração.

8. Por fim, tendo extraído as explicações procede-se à sua análise mais aprofundada para poder tirar conclusões sobre as mesmas e adquirir mais conhecimento sobre o método de aprendizagem.

Para a realização de aprendizagem da forma que está ilustrada pela figura 3.5, é necessária a escolha de um algoritmo de aprendizagem desacoplado do Explicador, pelo que, para a realização de testes, foram escolhidos alguns modelos.

Um dos mais simples modelos escolhidos foi o *Logistic Regression* (LR) [16]. Este é um algoritmo desenhado especialmente para classificação binária, embora também possa ser generalizado para multi-classe adotando uma estratégia *One-vs-Rest*, onde é feita a discriminação de cada classe face às restantes. Este método mapeia a relação das características com as classes através da função logística, também denominada por sigmóide que tem uma gama de valores [0, 1]. O treino é feito otimizando os coeficientes da seguinte equação:

$$P(y = 1|X) = \frac{1}{1 + e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\beta_3 x_3+...+\beta_n x_n)}}, \tag{3.7}$$

onde e é o número de Euler, y é a classe, β são os coeficientes e $X = [x_1, x_2, ..., x_n]$ é uma instância do conjunto de dados. Para a predição de novos dados, substitui-se o expoente de e pelos dados da instância que se pretende prever e obtém-se uma probabilidade; caso a probabilidade seja menor que o limiar definido (normalmente colocado a 0,5) é classificado como sendo da classe negativa (0); caso contrário é classificado como sendo da classe positiva (1). No caso de multi-classe os limiares são calculados separadamente para cada classe.

Outro método utilizado foi o *Random Forest* (RF). Este método pode ser utilizado para classificação ou regressão e tem por base a criação de diversas árvores de decisão para realizar as predições. O treino de cada árvore é feito com uma parcela de todos os dados de treino, onde cada árvore usa conjuntos diferentes para introduzir variabilidade nos resultados. A cada nó de cada árvore são consideradas características aleatórias, o que possibilita que cada árvore contribua diferentemente para o resultado final. Por fim, cada árvore é treinada tendo por métrica o máximo ganho de informação para definir as partições da mesma. O resultado da predição é a moda de todas as árvores geradas, isto é, a classe que apareceu mais vezes em todas as árvores é a escolhida.

Outro algoritmo de aprendizagem testado foi o *Support Vector Machines* (SVM) cuja ideia fundamental é criar um hiperplano capaz de discriminar as classes do problema em mãos. Um hiperplano não é mais do que um plano com N dimensões que, neste contexto, funciona como limiar para a separação das classes. Quando as classes não são linearmente separáveis, este classificador realiza uma projeção dos dados para um espaço de maior dimensionalidade onde estes já sejam separáveis. O seu treino assenta na otimização deste hiperplano, o que corresponde a encontrar o que maximiza a distância entre as classes. Esta maximização utiliza os vetores de suporte, que são os pontos (dados) que se encontram mais próximos do hiperplano, para realizar esta otimização, o que torna este processo eficiente em termos de memória, já que não são usados todos os dados de treino indiscriminadamente. Para a predição de novos dados é determinado em qual dos lados do hiperplano é que se encontram, onde cada lado corresponde a uma classe. Este tipo de algoritmo adota uma abordagem binária onde são apenas vistas duas classes, tal como no caso do LR. Também, tal como o LR, para suportar a resolução de problemas multi-classe, pode ser adotada uma abordagem *One-vs-Rest* ou *One-vs-One*, em que esta última trata de separar cada par de classes existente, havendo um total de $\frac{C(C-1)}{2}$ separações onde C é o número de classes.

Também foi utilizado o *Explainable Boosting Machine* (EBM) cujo funcionamento é apresentado em detalhe na secção 2.4.3.

Dado que um dos conjuntos de dados utilizados, nomeadamente, o de deteção de cancro do cérebro, consiste num conjunto de imagens, foi considerada a utilização de *Convolutional Neural Network* (CNN). Este é um tipo de redes neuronais muito utilizadas no domínio de imagem devido ao seu desempenho. Este tipo de redes é composto inicialmente por um conjunto de camadas convolucionais. Por sua vez, cada camada consiste num ou mais

filtros (*kernels*), onde cada um é uma grelha, muitas das vezes, quadrada. Uma convolução consiste em percorrer este *kernel* pela imagem de entrada, horizontal e verticalmente, e fazer o produto escalar da grelha pela porção da imagem correspondente, seguido da aplicação de uma função de ativação (e.g., $ReLU$). Os valores da grelha fazem parte dos parâmetros de aprendizagem, sendo que inicialmente podem ser gerados aleatoriamente. Passando por todas as camadas convolucionais, é criado o vetor resultante da aplicação de todas as convoluções. A próxima etapa consiste em passar esse vetor como entrada de uma rede MLP convencional. Para isto ter-se-á de "achatar" o vetor resultante, isto é, reduzi-lo a apenas um eixo, como se faz normalmente para o domínio de imagem. A saída da rede MLP apresentará tantos neurónios quanto o número de classes do problema. Nesta rede, o processo de aprendizagem é feito através de retro-propagação do erro, que é usado quer para atualizar os pesos dos neurónios da rede MLP e os pesos dos filtros. Para a previsão de novos dados, são aplicadas todas as convoluções aprendidas nas camadas convolucionais e o resultado destas é passado para a rede MLP, onde se obterá no final a classe predita. Este processo é idêntico à aprendizagem, apenas não existe a atualização de parâmetros do modelo.

Para proceder à extração de explicações, também foi necessário escolher os componentes que as permitem extrair. Para dados em formato tabular, no caso do conjunto de dados sintético e DARWIN, foram usados os seguintes métodos:

- LIME – Funcionamento descrito na secção 2.3.5.1
- SHAP – Funcionamento descrito na secção 2.3.5.1
- EBM – Funcionamento descrito na secção 2.4.3

Já para dados em formato de imagem também foram utilizados o LIME e o SHAP, no entanto o EBM foi trocado pelo método Grad-CAM, cujo funcionamento está descrito na secção 2.4.2.

3.4 Implantação (*Deployment*)

Tendo realizado diversas avaliações experimentais que permitiram avaliar o desempenho dos modelos, coloca-se a questão de como os utilizar num contexto real. O modo de operação adotado encontra-se na figura 3.7.

Para cada conjunto de dados utilizado, são treinados todos os modelos com todos os dados disponíveis, sem realizar a partição de treino e teste. Tendo todos os modelos treinados, estes são guardados para ser possível a sua utilização para novos dados. Quando se quiser prever um novo exemplo, isto é, quando se dá a chegada de um novo paciente e se pretende o seu diagnóstico, deve ser fornecido o exemplo no formato adequado ao formato com que o modelo foi treinado. Nesta fase, os modelos treinados são carregados para memória com o intuito de os utilizar para realizar o diagnóstico e extrair as explicações do mesmo. As explicações extraídas serão apenas locais, uma vez que se quer saber as explicações para

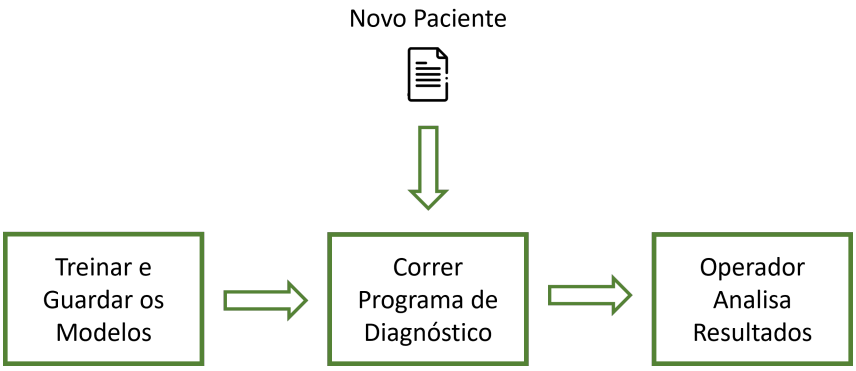


Figura 3.7: *Diagrama de blocos do sistema para novo diagnóstico*

cada diagnóstico realizado, e não sobre o desempenho geral dos modelos. Na fase seguinte, o operador (neste caso o médico) analisa os resultados obtidos e suas explicações para, posteriormente, tirar conclusões sobre o tratamento adequado para o paciente.

4 Avaliação Experimental

Neste capítulo são feitas experiências com dados sintéticos e com dados reais. O capítulo está organizado em três secções: a secção 4.1, a qual apresenta aspetos relevantes da implementação e do contexto onde o código foi corrido; a secção 4.2, que apresenta resultados obtidos a partir de conjuntos de dados gerados sinteticamente; e a secção 4.3, que apresenta resultados obtidos a partir de conjuntos de dados reais.

4.1 Aspetos de Implementação

Antes de proceder para a avaliação experimental, é importante declarar o contexto onde o código foi corrido, bem como esclarecer alguns pontos que influenciaram os resultados e a interpretação das explicações.

O código criado foi corrido numa máquina com as seguintes características:

1. Processador – AMD Ryzen 7 1700 Eight-Core Processor 3.00 GHz;
2. Memória RAM – 16 GB;
3. Placa Gráfica – Radeon RX 580;
4. Sistema Operativo – Windows 10 Home;
5. Linguagem – Python/Jupyter Notebook;
6. Ambiente – Anaconda Environment.

Em relação à implementação, um ponto relevante é o parâmetro *probability* do classificador SVM. Este parâmetro pode assumir o valor *True* ou *False*. Caso seja o primeiro, então é utilizado um método para calcular as probabilidades de cada instância para cada classe, que pode ser obtido através do *predict_proba()*. É referido na documentação que a maneira de calcular estas probabilidades, para além de apresentar imperfeições, é computacionalmente exigente e esta exigência aumenta com o número de instâncias. No entanto, dado que o LIME e o SHAP necessitam desta informação, este passo não pode ser descartado, levando a que, por vezes, o SVM se torne o modelo com maior tempo de execução.

Outro ponto relevante destas experiências é a extração de explicações, onde são utilizados os métodos LIME e SHAP.

Tal como referido em [36], o LIME apresenta alguns pontos negativos no seu desempenho em dados num formato tabular. Dado que este explicador usa um modelo linear simples para extrair as explicações, pode dar-se o caso deste não conseguir descrever relações mais complexas que sejam indispensáveis para o bom entendimento do modelo. Isto pode conduzir à falta de fidelidade para com o modelo original e mais complexo ao qual se pretende extrair explicações, isto é, pode dar-se o caso que este forneça explicações que indicam resultados diferentes do modelo original. Outra desvantagem é a falta de consistência. Se aplicado o LIME diversas vezes sobre a mesma instância ou sobre uma semelhante, este pode oferecer explicações díspares. Em parte, isto deve-se ao inerente carácter aleatório deste método que realiza perturbações sobre a instância a ser explicada para criar novos exemplos com o intuito de treinar um modelo mais simples. Apesar de nos conjuntos de dados usados não se terem verificado alterações, outro fator que pode contribuir para esta inconsistência é o valor de vizinhança indicado pelo parâmetro *kernel_width*. Este parâmetro indica a localidade das explicações. Caso este valor seja pequeno, apenas os pontos aleatórios gerados que estão muito próximos à instância a ser explicada terão impacto na explicação. Caso o valor seja grande, pontos aleatórios mais distantes também terão impacto na extração de explicações. Para possibilitar a reprodutibilidade, foi declarada uma semente (*seed*) de modo a garantir que a extração de explicações não varia caso os parâmetros não variem.

Para a utilização do SHAP foram usadas duas classes da biblioteca adotada para a extração de explicações, sendo elas o *KernelExplainer* e o *TreeExplainer*. O *KernelExplainer* permite a extração de explicações de qualquer tipo de modelo, mas apresenta um custo computacional elevado e foi necessário realizar amostragem dos dados de treino para poder ser usado em tempo útil. Os modelos que tiraram partido deste método foram o LR e o SVM. O *TreeExplainer* é um método específico para modelos baseados em árvores de decisão e está otimizado para este tipo de modelos, permitindo assim descartar o processo de amostragem, já que o seu custo computacional é menor. O método que usufruiu desta classe foi o RF. Em [36], é referido que uma desvantagem do *KernelExplainer* é desconsiderar a dependência entre características. Com o uso do *TreeExplainer*, esta dependência pode ser capturada através da alteração da fórmula usada para calcular a importância de cada característica. No entanto, apresenta a desvantagem de poder atribuir importância diferente de 0 a características que não contribuem para a predição.

4.2 Dados Sintéticos

Nesta secção são apresentados testes com dados sintéticos, isto é, dados que são gerados através de um algoritmo e que não apresentam nenhuma conexão com a realidade. Estes testes têm como objetivo aferir o potencial dos métodos para extração de explicações, sendo que os conjuntos de dados gerados estão sob o controlo parcial do operador.

Para a geração de dados sintéticos, foi elaborado um algoritmo que recebe os seguintes

parâmetros:

- Valor mínimo;
- Valor máximo;
- Número de linhas;
- Número de colunas;
- Número de classes;
- Coeficientes de aleatoriedade.

Os coeficientes de aleatoriedade referem-se ao grau de aleatoriedade que cada característica/coluna apresenta para cada classe. Estes coeficientes devem ser uma matriz com dimensões (#características, #classes), onde cada célula é um número entre 0 e 1 que indica o grau de aleatoriedade de uma característica para cada classe. Um exemplo de uma matriz de coeficientes de aleatoriedade para um conjunto de dados com 3 características e 3 classes está apresentado na tabela 4.1.

Tabela 4.1: Matriz de coeficientes de aleatoriedade

	Coef. C1	Coef. C2	Coef. C3
Característica1	0.1	0.4	0.85
Característica2	0.65	0.9	0.2
Característica3	0.7	0.15	0.1

Pegando no exemplo da *Característica1*, esta apresenta um coeficiente de aleatoriedade de 0,1 para a classe $C1$, de 0,4 para a classe $C2$ e de 0,85 para a classe $C3$, o que significa que apresenta aleatoriedade baixa para $C1$, aleatoriedade média para $C2$ e aleatoriedade elevada para $C3$. Quanto menor for a aleatoriedade, mais informação dará a característica para a previsão da classe e, por conseguinte, mais relevante será para o classificador.

O primeiro passo para criar o conjunto de dados é gerar valores de domínio real aleatoriamente no intervalo [*min_value*, *max_value*], onde a classe é gerada de forma incremental, sendo que as linhas estão organizadas por ordem crescente da classe (e.g., 0, 0, ..., 1, 1, ..., 2, 2,...). Por isto, é aconselhado que, aquando da partição do conjunto de dados num conjunto de treino e teste se mude a ordem dos dados através de métodos de permutação (*shuffle*). De seguida, é gerada uma matriz com dimensões ($N_{linhas} \times N_{colunas}$) com valores aleatórios entre 0 e 1. Se, para uma determinada linha e para uma determinada coluna, esse número não superar o coeficiente de aleatoriedade, então o valor aleatório gerado inicialmente é mantido. Caso contrário, significa que a característica terá informação sobre a classe.

Para embutir informação da classe numa característica, foi adotado um processo semelhante ao da quantização uniforme de um sinal contínuo, onde são criados intervalos para discretizar o sinal. Em primeiro lugar, é calculada a diferença entre o menor valor possível e o maior, isto será denotado *feat_interval*. Todas as características terão o mesmo domínio de valores,

que é dado como parâmetro deste método. De seguida, este intervalo será dividido pelo número de classes existentes menos 1, obtendo-se através do cálculo $\frac{feat\_interval}{N_{Classes}-1}$. Este valor é denotado por *class_interval*. A cada ponto desta divisão corresponderá o valor da característica para uma certa classe, relacionados através da fórmula:

$$Valor_{j,i} = min\_val + (class\_val \times class\_interval), \tag{4.1}$$

onde j é a linha, i é o índice de uma dada característica e $class\_val$ é o valor da classe para a linha em questão. De modo a facilitar a compreensão, veja-se um exemplo com 3 classes onde os valores variam no intervalo [-15, 15]. Em primeiro lugar, calcula-se a diferença do intervalo (*feat_interval*): 15 - (-15) = 30. De seguida, calcula-se o tamanho do intervalo para cada classe (*class_interval*): 30 / (3 - 1) = 15. Aplicando a fórmula para class_val=1, tem-se: $-15 + (1 \times 15) = 0$. Sendo assim, chega-se à divisão presente na figura 4.1.

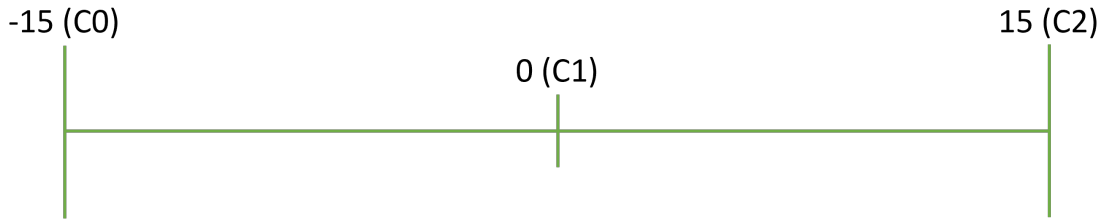


Figura 4.1: *Divisão do conjunto de dados sintético*

Também foi implementado outro tipo de geração de dados sintéticos. Este tipo é à base de curvas gaussianas e para este tipo de geração a matriz de coeficientes de aleatoriedade não é necessária, mas são necessários outros parâmetros, nomeadamente:

1. std_dev – Desvio padrão das curvas de domínio [0, $+\infty$];
2. class_sep – Separação das curvas de domínio [0, 1].

Estes dois novos parâmetros apresentam tantos elementos quanto o número de características, ao contrário da matriz de coeficientes de aleatoriedade que apresenta ($N_{Caracteristicas} \times N_{Classes}$). Isto implicará que todas as curvas gaussianas da mesma característica apresentem o mesmo desvio padrão. Este foi o processo adotado para reduzir a parametrização necessária, caso contrário ter-se-ia de declarar duas matrizes com as mesmas dimensões da dos coeficientes de aleatoriedade. A consequência desta simplificação é que, para o caso multi-classe, a mesma característica terá impacto semelhante para todas as classes. O parâmetro *class_sep* indica a separação das classes, se o seu valor for reduzido, o ponto central de cada curva gaussiana, que representa uma classe, estará mais próximo das restantes curvas, o que provocará sobreposição das mesmas gerando um pior desempenho. Caso seja elevado, o ponto central de cada curva estará mais afastado das restantes, melhorando o desempenho do modelo de classificação.

A fórmula usada para calcular o desvio padrão é a seguinte:

$$desvio\_padrao = std\_dev \times class\_interval \div 3. \tag{4.2}$$

Caso o parâmetro *std_ dev* seja 1, isto corresponderá a um desvio padrão de *class_ interval* / 3. Caso a *class_ sep* seja de 1, então o ponto médio de cada curva estará centrado no intervalo da sua respetiva classe, tal como demonstrado na figura 4.2.

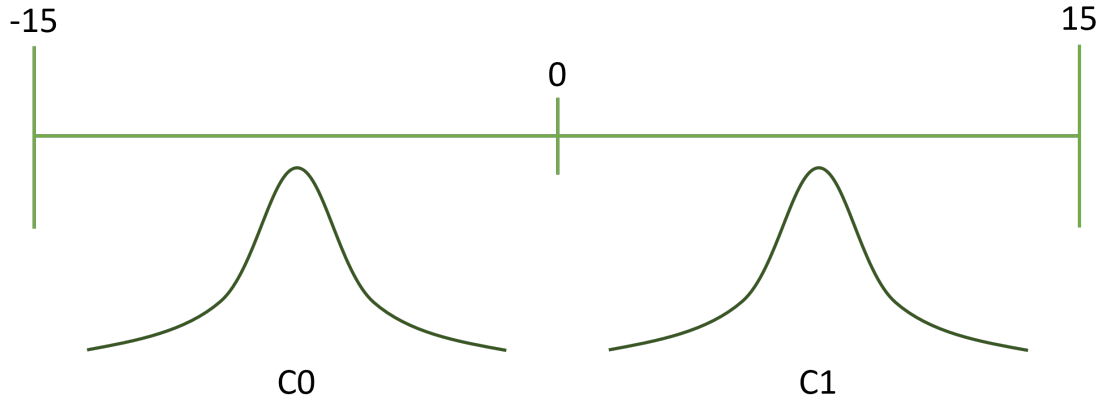


Figura 4.2: *Divisão do conjunto de dados sintético gerado com curvas gaussianas*

Caso o desvio padrão seja aumentado, então as curvas terão um maior "achatamento". Por outro lado, caso a separação das classes seja reduzida, os pontos centrais das gaussianas ficarão mais próximos do ponto médio do domínio de valores, fazendo com que se sobreponham.

Tendo apresentado em detalhe o modo de geração do conjunto de dados sintético, a fase seguinte é a de apresentação de resultados.

Foram realizados diversos testes com os modelos que serão utilizados ao longo de todas as experiências e verifica-se que os modelos baseados em árvore apresentam bastante mais facilidade em atingir resultados satisfatórios, nomeadamente o RF e o EBM. O LR e o SVM podem apresentar algumas dificuldades. Apesar disso, verifica-se que nestes dois últimos, se for retirada a aleatoriedade, estes apresentam taxas de acerto de 100% e à medida que se acrescenta aleatoriedade, vão perdendo desempenho, o que indica coerência lógica entre os dados e o comportamento dos classificadores. Também no caso da distribuição normal, verificou-se que o aumento do desvio padrão piora o desempenho, assim como a diminuição da separação das classes, já que o seu fator discriminante é reduzido.

4.2.1 Classificação Binária

O primeiro teste realizado foi o de classificação binária, cujos parâmetros para criação do conjunto de dados são:

- Valor Mínimo – -100;
- Valor Máximo – 100;
- Número de linhas – 1000;

- Número de colunas – 10;
- Número de classes – 2;
- Coeficientes de aleatoriedade –

$$\begin{bmatrix} 0.95 & 0.95 \\ 0.9 & 0.9 \\ 0.85 & 0.85 \\ 0.8 & 0.8 \\ 0.75 & 0.75 \\ 0.7 & 0.7 \\ 0.65 & 0.65 \\ 0.6 & 0.6 \\ 0.55 & 0.55 \\ 0.5 & 0.5 \end{bmatrix}$$

.

Como existem duas classes, a matriz de coeficientes terá de possuir duas colunas (uma por classe). No entanto, foi considerado que não faria sentido, ao nível da classificação binária, uma variável apresentar coeficientes diferentes para as duas classes. Isto porque se uma característica apresentar coeficiente de aleatoriedade baixo com uma classe, dependendo do classificador, este conseguirá descobrir a outra adotando o raciocínio "se apresentar linearidade é uma se não é outra". Apesar disso, se por alguma razão o operador pretender este efeito, poderá colocar coeficientes diferentes para cada classe, mesmo tratando-se de classificação binária.

Em primeiro lugar, foi feita a procura pelos melhores parâmetros de cada método de aprendizagem. Isto foi feito através do *GridSearchCV*, que recebe um conjunto de parâmetros e seus valores para testar todas as combinações e concluir qual destas obtém melhor desempenho. Para cada conjunto de valores é usado o *StratifiedKFold* com cinco partições (*folds*). Os conjuntos de parâmetros testados foram os seguintes:

- *Logistic Regression*

  max_iter – [100, 500, 1000, 2000]

  solver – ["lbfgs", "liblinear", "newton-cholesky"]

  C – [0.01, 0.1, 1, 10, 100]

- *Random Forest*

  n_estimators – [100, 250, 500, 1000, 2000]

  bootstrap – [True, False]

  max_features – ["sqrt", "log2", None]

- *Support Vector Machines*

kernel – ["rbf", "linear"]

C – [0.05, 0.1, 1, 10, 100, 500]

- *Explainable Boosting Machine*

  smoothing_rounds – [500, 1000]

  cyclic_progress – [0, 0.5]

  max_bins – [1024, 2048]

Os melhores parâmetros foram os seguintes:

- *Logistic Regression* – [max_iter=100, solver="liblinear", C=0.1];
- *Random Forest* – [n_estimators=1000, boostrap=False, max_features="sqrt"];
- *Support Vector Machines* – [kernel="rbf", C=0.1];
- *Explainable Boosting Machine* – [smoothing_round=1000, cyclic_progress=0, max_bins=1024].

As matrizes de confusão obtidas para cada classificador encontram-se na figura 4.3.

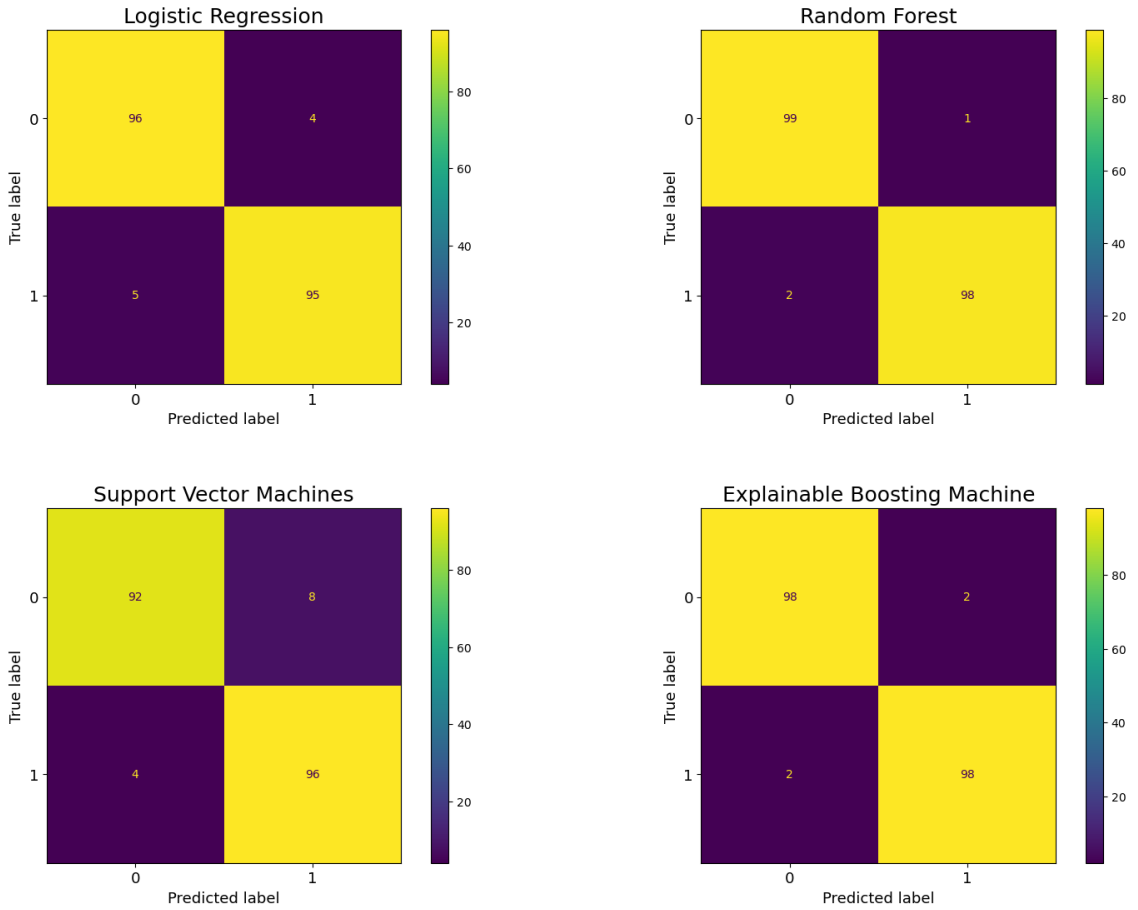


Figura 4.3: *Matrizes de confusão para classificação binária do conjunto de dados sintético*

Tabela 4.2: Resultado das métricas para a classificação binária do conjunto de dados sintético

Métrica	LR	RF	SVM	EBM
Taxa de acerto	0.95	**0.98**	0.94	**0.98**
Taxa de falsos positivos	0.04	**0.01**	0.08	0.02
Taxa de falsos negativos	0.05	**0.02**	0.04	**0.02**
Precisão	0.96	**0.99**	0.92	0.98
Cobertura	0.95	**0.98**	0.96	**0.98**
F-Score	0.95	**0.98**	0.94	**0.98**

Através desta matrizes foram obtidas as métricas presentes na tabela 4.2.

Para este problema, é possível averiguar através da tabela 4.2, que o modelo com melhor desempenho foi o RF, sendo que o EBM apresenta mais um erro do que este. Abaixo destes está o LR, com uma taxa de acerto de 95% e o que obteve pior desempenho geral foi o SVM com taxa de acerto de 94%. Apesar disso, obteve uma melhor taxa de falsos negativos, que num contexto médico é por vezes uma métrica mais importante do que a taxa de acertos.

De seguida, criaram-se 10 partições diferentes de treino e teste por forma a obter resultados das métricas mais fidedignos. Estes resultados encontram-se na tabela 4.3.

Tabela 4.3: Resultado das métricas em 10 partições diferentes de treino e teste

Métrica	LR	RF	SVM	EBM
Taxa de acerto	0.93 ±0.01	**0.98 ±0.01**	0.93 ±0.01	0.97 ±0.01
Taxa de falsos positivos	0.08 ±0.02	**0.03 ±0.02**	0.08 ±0.02	**0.03 ±0.02**
Taxa de falsos negativos	0.05 ±0.02	**0.01 ±0.01**	0.05 ±0.02	**0.01 ±0.01**
Precisão	0.92 ±0.02	**0.97 ±0.02**	0.92 ±0.01	**0.97 ±0.02**
Cobertura	0.95 ±0.02	**0.99 ±0.01**	0.95 ±0.02	**0.99 ±0.01**
F-Score	0.93 ±0.01	**0.98 ±0.01**	0.94 ±0.01	**0.98 ±0.01**

Através da tabela 4.3, verifica-se que não existe discrepância relevante entre diferentes partições, pelo que se prosseguirá com a análise dos resultados obtidos na primeira execução.

Uma outra forma de visualizar o desempenho dos classificadores é através das curvas PR e ROC, que se apresentam na figura 4.4.

Através da figura 4.4 é possível fazer uma separação em dois grupos: o RF e EBM com melhor desempenho e o LR e SVM com um desempenho inferior. Também é possível constatar o local onde o SVM se encontra em relação ao LR, apesar de terem as curvas bastante semelhantes, as coordenadas do SVM (indicadas pelo ponto preto), privilegiam um pouco mais a taxa de falsos negativos, resultando mesmo na degradação geral do desempenho. Caso não fosse pretendido este efeito seria possível, calibrando a função de decisão do SVM chegar à mesma taxa de acerto do LR.

Para a extração de explicações locais foi usada a instância presente na tabela 4.4 que pertence à classe 0.

Para o caso em que não existe aleatoriedade, as variáveis que compõem a instância todos

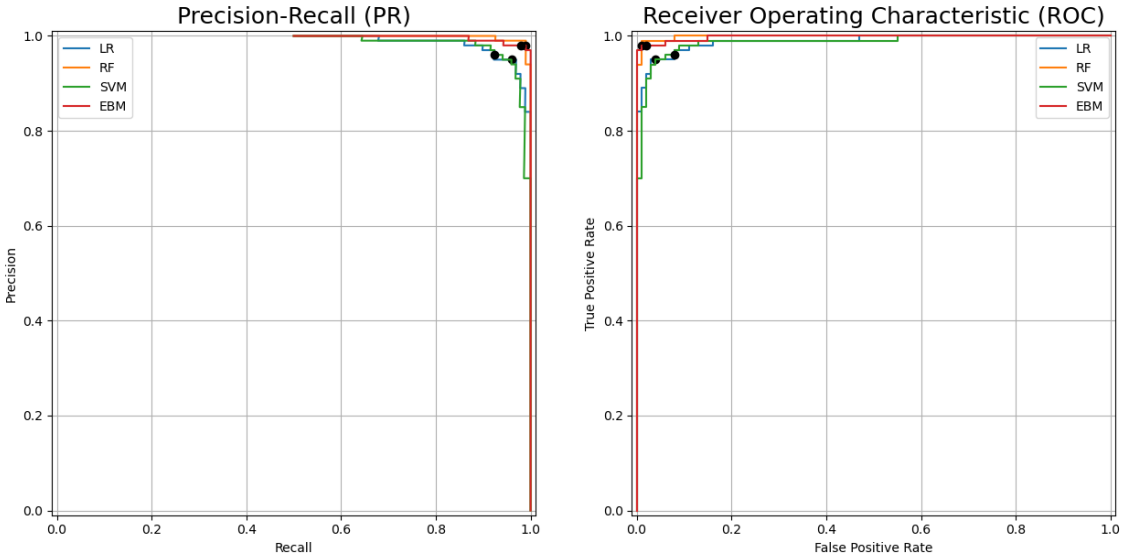


Figura 4.4: *Curvas PR e ROC para classificação binária do conjunto de dados sintético*

Tabela 4.4: Instância usada para extrair explicações locais

Var0	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8	Var9
-98.537	-44.395	40.606	-100	96.361	24.071	-4.498	52.286	80.665	-100

teriam valor -100, já que a instância pertence à classe 0. Caso a instância pertencesse à classe 1, todas as variáveis assumiriam o valor 100. Através da tabela 4.4 é possível constatar que as únicas variáveis que não possuem valores aleatórios são Var3 e a Var9, sendo que a Var0 apresenta um valor próximo do seu ideal. Todos os classificadores utilizados classificaram corretamente a instância da tabela 4.4 exceto o SVM.

A figura 4.5 ilustra as explicações locais obtidas para a instância previamente mencionada usando o método LIME.

Através da figura 4.5 é possível constatar que, sobretudo para os casos do LR e SVM, as características que apresentam um valor mais próximo de 100, como é o caso da Var7 e Var8, contribuem para a classe dos positivos. A Var4, que é a que apresenta valor mais perto de 100, também contribui significativamente para a classe dos positivos, o que não se verifica para o RF, que é a que apresenta menor relevância para esta instância. Este facto poderá ser relevante para o superior desempenho do modelo, já que este "não se deixa enganar" por este fator. As características Var3 e Var9, que apresentam relação com a classe, contribuem em todos os classificadores para a classe correta, o que revela coerência. Apesar disso, em nenhum modelo são as que contribuem mais para a tomada de decisão. Outro facto que é importante destacar é que o LR e o SVM têm as mesmas probabilidades, no entanto, apresentam classificações diferentes, já que o SVM errou na classificação da instância usada para extrair explicações locais. A justificação mais provável, como já referido anteriormente e como mencionado na documentação do SVM, é o facto do cálculo destas probabilidades, para este modelo, ser imperfeito podendo dar origem a estes cenários onde existe incoerência entre a classificação e as probabilidades. Um fator

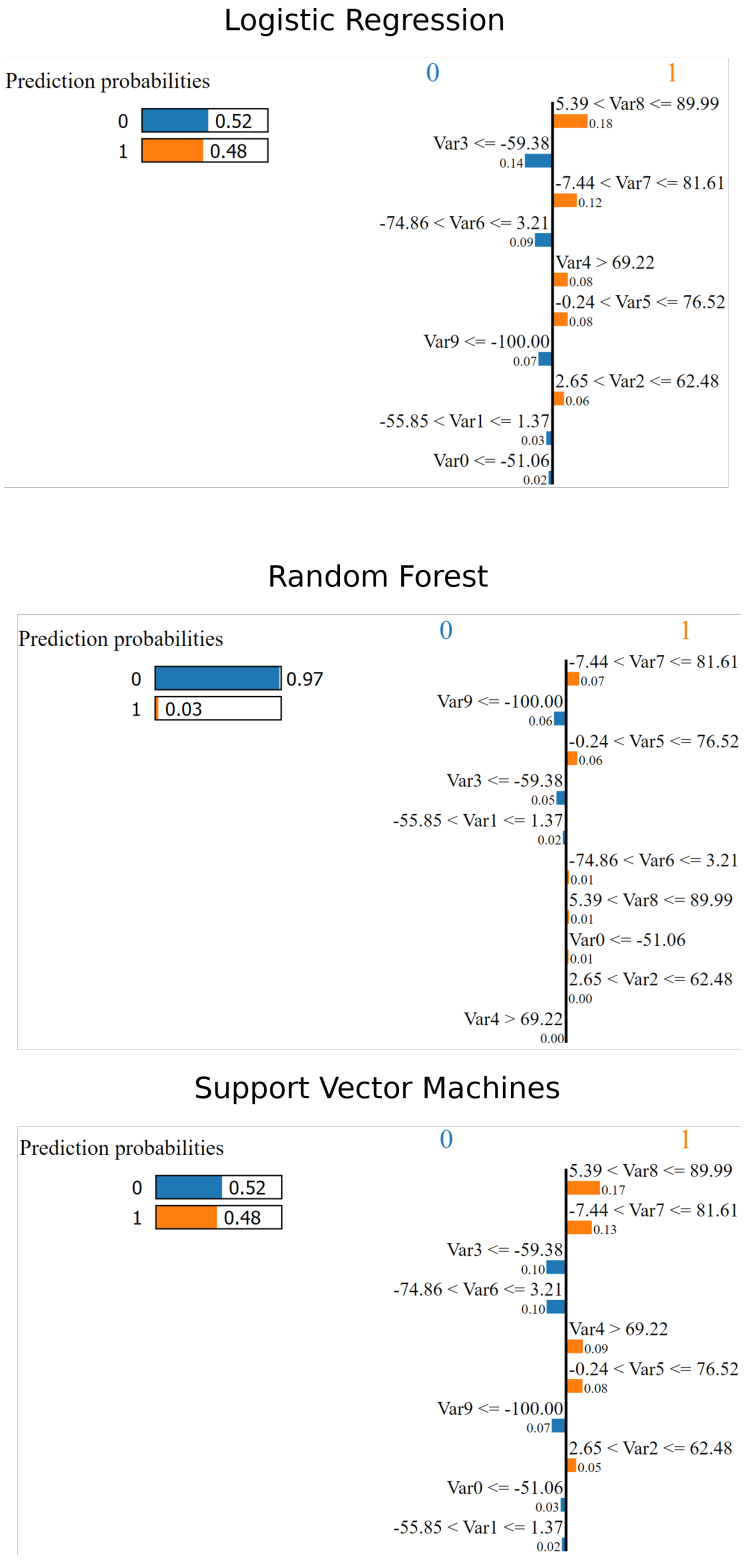


Figura 4.5: *Explicações locais para classificação binária do conjunto de dados sintético usando LIME*

que também pode contribuir para a inconsistência entre a predição e as probabilidades é o facto destas últimas estarem bastante próximas.

Na figura 4.6 encontram-se a extração de explicações locais para a instância previamente mencionada usando o método SHAP e EBM.

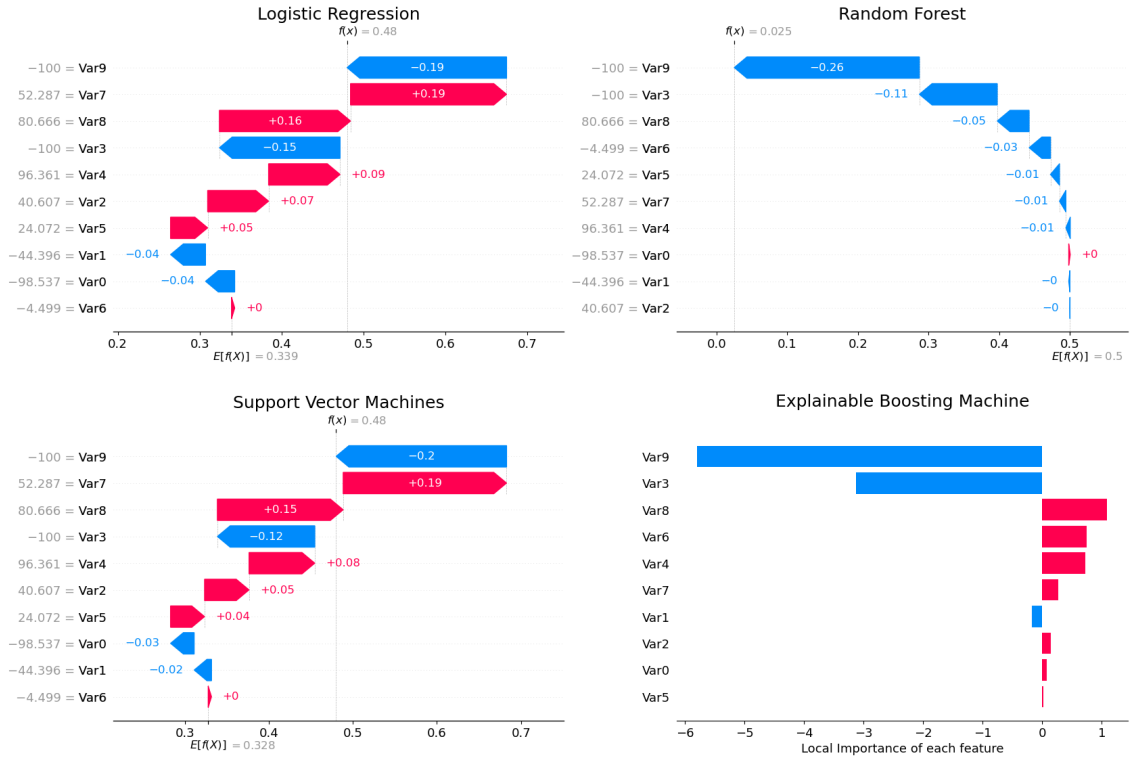


Figura 4.6: *Explicações locais para classificação binária do conjunto de dados sintético usando SHAP e EBM*

Verificam-se algumas diferenças com as explicações extraídas do LIME. A primeira diferença é que todos os classificadores consideraram a Var9 como sendo a que mais contribui para a classe negativa. Em relação ao LR e ao SVM, as variáveis Var7 e Var8 continuam com grande relevância para a classe dos positivos, superando a contribuição que a Var3 tem para a classe correta. Para o caso do RF, existem diferenças bastante significativas em relação ao LIME, sendo que as duas características mais importantes são as que não são aleatórias e, portanto, existe uma maior coerência. Outro facto relevante é que todas as características, com exceção da Var0, contribuem para a classe negativa, sendo que a relevância das últimas três características é residual. O EBM também reconhece as duas variáveis não aleatórias como as mais importantes e, por uma boa margem, as que mais contribuem para a predição. Em semelhança com o LR e o SVM, a Var8 também contribui para a classe errada, tal como as variáveis Var6 e Var4.

Na figura 4.7 estão apresentadas as explicações globais extraídas.

Através da figura 4.7 pode-se observar que, tanto o RF como o EBM acertaram na ordem de relevância de características, tendo em conta a sua aleatoriedade. Para o caso do LR, existe uma troca das características Var3, Var2 e Var4, sendo que a ordem deveria ser Var4, Var3 e Var2. Para o caso do SVM, apenas existe uma troca entre a Var3 e Var4. No geral,

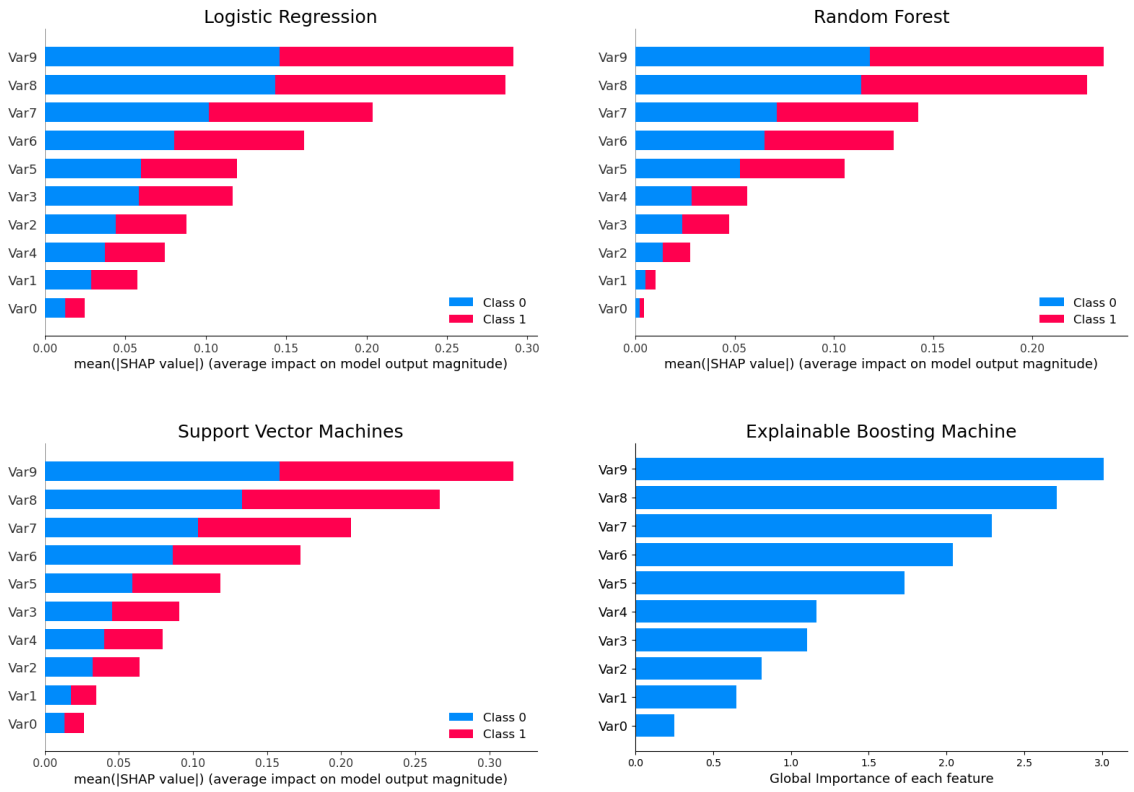


Figura 4.7: *Explicações globais para classificação binária do conjunto de dados sintético*

as explicações globais coincidem com a ordem de aleatoriedade estabelecida, revelando coerência entre os dados e os explicadores.

Também foram registados os tempos de execução dos modelos para as fases de procura dos melhores parâmetros e extração de explicações usando o SHAP. Os resultados são apresentados na tabela 4.5.

Tabela 4.5: Tempo de execução em segundos para a fase de procura de parâmetros e SHAP

Tarefa	LR	RF	SVM	EBM
Treino (GS)	3.1	47.1	10816.3	123.3
SHAP	25.7	2.4	217.8	–

Verifica-se que o SVM foi o único que demorou um tempo considerável, cerca de 3 horas. Foram feitos testes separadamente e verificou-se que o *kernel* linear é o que requer mais tempo de cálculo, o que *a priori* não se esperava uma vez que este é mais simples que o RBF. Isto pode dever-se ao facto do problema imposto não ser linearmente separável ao nível espacial, o que complicaria o cálculo dos vetores de suporte. O segundo classificador que teve um maior tempo de execução foi o EBM, tendo demorado perto de 2 minutos. Em relação ao SHAP, o SVM também foi o que demorou mais, cerca de 3 minutos e 38 segundos, o LR demorou cerca de 26 segundos. Dado que o RF apresenta a otimização previamente mencionada, este apenas demorou 2,4 segundos.

Um passo seguinte, foi a realização de alterações ao conjunto de dados para estudar de que forma é que estas alterariam os resultados. Nestes testes, os parâmetros obtidos foram

mantidos. No entanto, dado que se muda o conjunto, também pode haver novos valores para os parâmetros ótimos, mas devido ao elevado poder computacional que requeria realizar a procura pelos parâmetros ótimos e também pelo facto desse não ser o objetivo principal da análise, os parâmetros não foram alterados.

Um primeiro estudo foi o de mudar a quantidade de linhas do conjunto de dados. Foram feitos testes com os seguintes valores de linhas: 2000, 4000 e 10000 e verificou-se que, para os três testes, não houve mudanças significativas ao nível das métricas. Também ao nível de explicações globais não foram registadas mudanças significativas.

Outro estudo feito foi o de valores do domínio, mantendo 10 características e 1000 linhas. Foram experimentados os seguintes intervalos: [-250, 250], [-50, 50], [-25, 25], [-10, 10], [-1, 1], [0, 1]. Para os testes realizados, verificou-se que a mudança do domínio de valores não afeta ou apresenta um efeito reduzido sobre os resultados, registando-se que os valores de todas as métricas não apresentam mudanças relevantes.

O seguinte estudo foi o da alteração do número de características. Foram testados os seguintes valores: 5, 25 e 50. O cálculo dos coeficientes de aleatoriedade foi dado por $0.95 - 1/num\_features \times 2 \times idx$, onde idx é o índice da variável (e.g., o índice de Var9 é 9). Isto faz com que a relevância de uma característica aumente com o índice, dado que terá um menor coeficiente de aleatoriedade. Com cinco características os coeficientes de aleatoriedade são: 0,95, 0,85, 0,75, 0,65 e 0,55.

Com 5 características foram obtidas as métricas presentes na tabela 4.6.

Tabela 4.6: Resultado das métricas para a classificação binária do conjunto de dados sintético com 5 características

Métrica	LR	RF	SVM	EBM
Taxa de acerto	0.82 ±0.01	**0.9 ±0.01**	0.82 ±0.01	**0.9 ±0.01**
Taxa de falsos positivos	0.2 ±0.04	0.11 ±0.03	0.21 ±0.04	**0.1 ±0.02**
Taxa de falsos negativos	0.15 ±0.03	**0.09 ±0.02**	0.15 ±0.03	**0.09 ±0.03**
Precisão	0.81 ±0.03	0.89 ±0.02	0.81 ±0.02	**0.9 ±0.01**
Cobertura	0.85 ±0.03	**0.91 ±0.02**	0.86 ±0.03	**0.91 ±0.03**
F-Score	0.83 ±0.01	**0.9 ±0.01**	0.83 ±0.01	**0.9 ±0.01**

Através da tabela 4.6, é possível averiguar que houve uma degradação de desempenho generalizada, mas a diferença entre classificadores mantém-se, sendo que o RF e o EBM continuam a ser os superiores. Apesar disso, a nível de explicações globais existe uma menor confusão na ordem de características, sendo que em quase todos os testes, todos os modelos acertaram na ordem de relevância das mesmas. Uma possível causa para este efeito é o de haver um menor número de características, sendo assim, a sua aleatoriedade é mais díspar.

Para o caso de 25 características, todos os classificadores obtiveram métricas perfeitas ou perto disso, sendo que o RF e o EBM obtiveram taxa de acerto de 100% com desvio padrão de 0 e o LR e o SVM obtiveram taxa de acerto de 99% com desvio padrão de 1% no caso do LR e 0% no caso do SVM. Uma possível razão deste aumento é o de haver menor probabilidade de todas as características serem aleatórias, dada uma instância.

Possivelmente basta que uma característica ou um conjunto reduzido de características apresentem correlação com a classe para ser possível predizer a mesma. Dado que o nível de aleatoriedade foi mantido em termos de ordem de escala comparativamente ao caso de 10 características, adicionar mais características aumentará um número de características com correlação com a classe. Consequentemente, isto leva a um aumento de desempenho ao nível de todas as métricas. Apesar disso, verificou-se existência de dificuldade em colocar as variáveis na ordem de relevância correta. Na figura 4.8 encontra-se a extração de explicações globais num dos testes realizados.

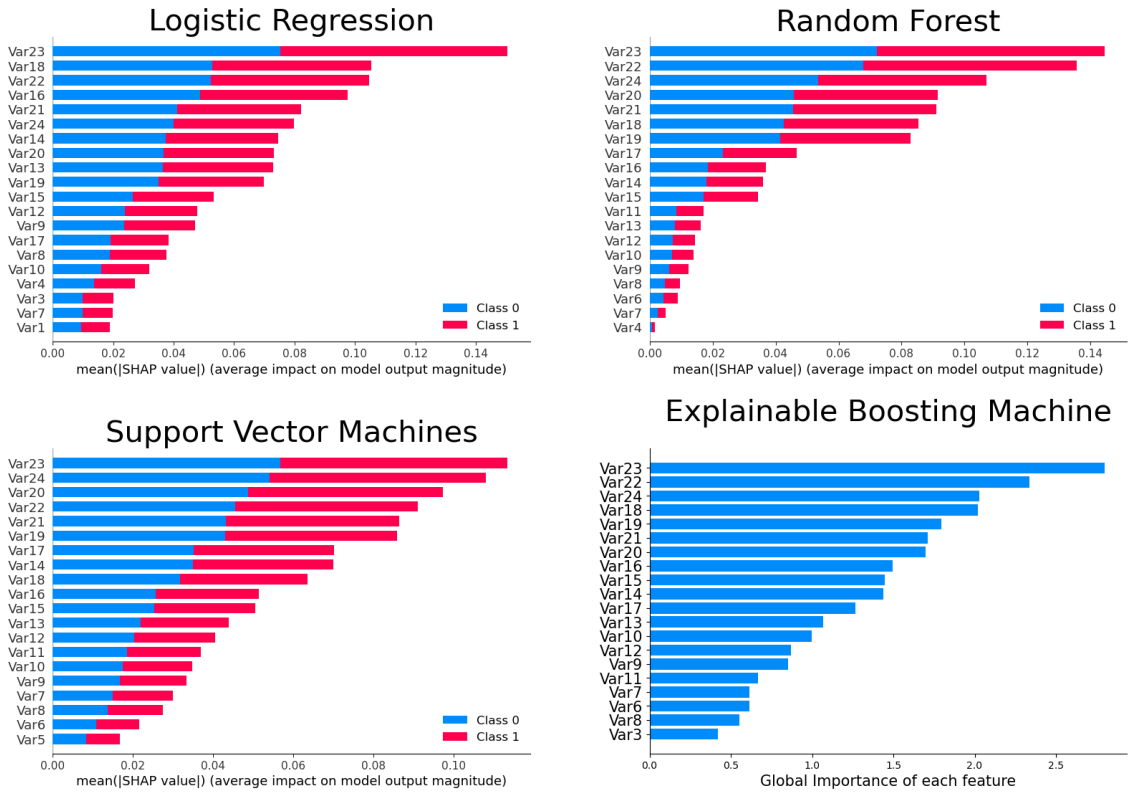


Figura 4.8: *Explicações globais para classificação binária do conjunto de dados sintético com 50 características*

Sobretudo para o caso do LR, existe uma grande diferença entre a ordem real de importância real das características e a extraída. Tal pode ser causado pelo número elevado de características, as quais não apresentam aleatoriedade muito diferente das suas adjacentes, podendo haver trocas. Os restantes modelos conseguiram determinar a ordem de melhor forma. Apesar de haver disparidades, muitas vezes são entre características adjacentes, que apresentam aleatoriedade semelhante.

Para 25 características, o RF e o EBM obtiveram taxa de acerto perfeita. O LR obteve cerca de 54% e o SVM cerca de 97%. Quer o RF, quer o EBM, ao nível de explicações globais, conseguiram acertar a ordem das primeiras seis características, mas depois começam a haver algumas trocas, mas no geral, conseguiram distinguir a importância das características. Para o LR a ordem de importância das características é distante da real, o que seria de esperar tendo em conta o seu desempenho. O SVM apenas considerou como importante a Var24, a característica mais importante.

Com 50 características, todos os classificadores atingiram a perfeição em todas as métricas. No entanto, novamente verificou-se dificuldade em encontrar a ordem de importância real das características. Desta vez, não só para o LR, mas em todos os modelos. Isto pode dever-se ao facto de que, usando a fórmula mencionada para gerar os coeficientes de aleatoriedade, com 50 características estas apenas têm diferença de 1% de aleatoriedade entre as suas adjacentes, o que dificulta a sua distinção em termos de relevância. Uma outra hipótese é a de que quanto maior for o número de características, maior dificuldade os métodos de explicabilidade têm para identificar a diferença de relevância entre as mesmas, no entanto, a primeira hipótese parece ser a mais provável.

O último teste realizado foi o de considerar uma distribuição gaussiana para criar o conjunto de dados, como mencionado anteriormente, tendo por parâmetros os seguintes valores:

1. Desvio Padrão (std_dev) – [1.5, 1.4, 1.3, 1.2, 1.1, 1.0, 0.9, 0.8, 0.7, 0.6];
2. Separação de classes ($class\_sep$) – [0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5];
3. Valor Mínimo – -100;
4. Valor Máximo – 100;
5. Número de linhas – 1000;
6. Número de colunas – 10;
7. Número de classes – 2.

Como é possível averiguar através dos parâmetros, a separação das classes é a mesma para todas as características. No entanto, o desvio padrão diminui com o aumento do índice da característica. Na figura 4.9 está apresentada a visualização das distribuições das várias características.

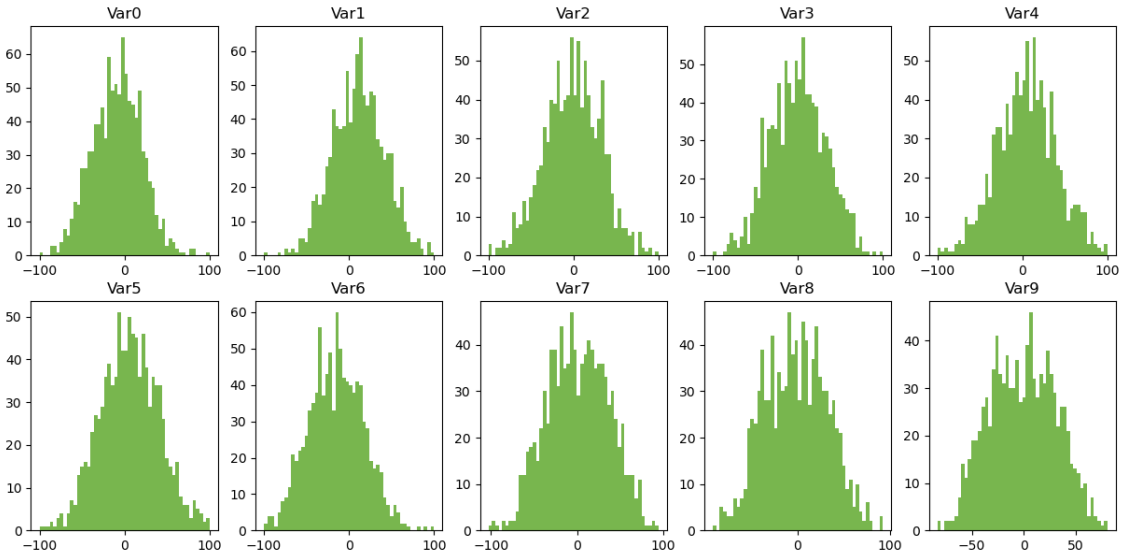


Figura 4.9: *Histograma das variáveis do conjunto de dados gerado com distribuição normal*

Para grande parte das variáveis não é possível visualizar as duas curvas normais (uma de cada classe), como mostrado anteriormente na figura 4.2, sendo que as mesmas parecem estar sobrepostas. Isto deve-se ao elevado desvio padrão das diversas características. Apesar da discriminação "a olho nu" parecer impossível, os classificadores utilizados obtêm elevadas taxas de acerto. Todos os classificadores atingiram taxas de acerto próximas de 99% exceto o SVM, que atingiu 98%. A figura 4.10 mostra as explicações globais extraídas de uma das execuções.

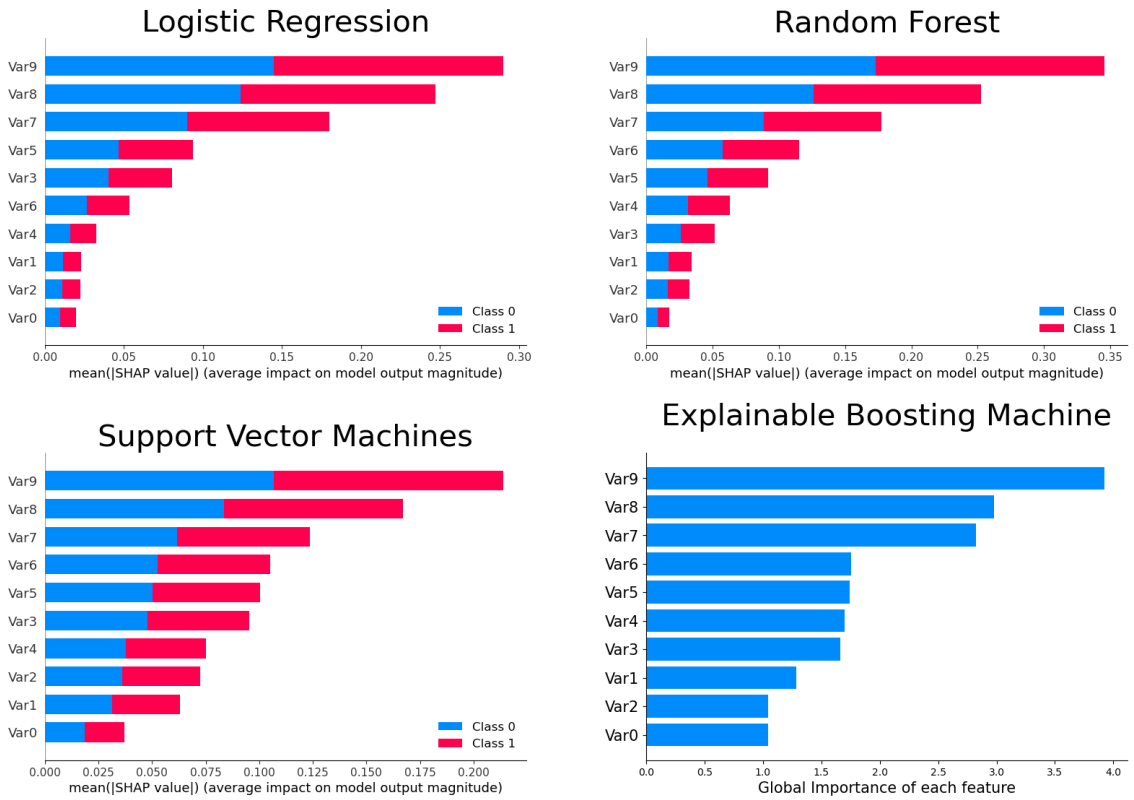


Figura 4.10: *Explicações globais para o conjunto de dados gerado com distribuição normal*

Apesar do elevado desempenho, parece existir uma maior confusão em comparação ao primeiro teste onde os dados seguiam uma distribuição gaussiana. Isto pode dever-se ao facto de os desvios padrão de cada curva serem próximos entre si, indicando que a diferença entre os desvios padrão de cada curva teria de ser maior para possibilitar que os métodos de explicabilidade conseguissem identificar a ordem de relevância correta de cada característica. Num contexto semelhante com dados reais, isto indica que características que apresentam um grau elevado de correlação poderão ser trocadas na sua ordem de importância, dado que apresentam um comportamento semelhante e, portanto, também relevância semelhante.

4.2.2 Classificação Multi-Classe

De seguida, analisou-se o contexto de multi-classe, onde foi criado um conjunto de dados com as seguintes características:

- Valor Mínimo – -100;
- Valor Máximo – 100;

- Número de linhas – 2500;
- Número de colunas – 10;
- Número de classes – 10;
- Coeficientes de aleatoriedade –

$$\begin{bmatrix} 0.05 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.05 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.05 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.05 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.05 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.05 & 0.9 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.05 & 0.9 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.05 & 0.9 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.05 & 0.9 \\ 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.9 & 0.05 \end{bmatrix}$$

.

Dado que existem 10 características e 10 classes, cada característica apresenta uma baixa aleatoriedade para uma das classes e elevada aleatoriedade para as restantes. Esta é uma forma de averiguar se os modelos e o SHAP conseguem descobrir esta correlação.

Neste contexto também foi efetuada a procura pelos melhores parâmetros, podendo os mesmos variar na seguinte gama:

- *Logistic Regression*

  max_iter – [100, 500, 1000, 2000]

  solver – [“lbfgs”, “newton-cg”, “sag”, “saga”]

  C – [0.01, 0.1, 1, 10, 100]

- *Random Forest*

  n_estimators – [100, 250, 500, 1000, 2000]

  bootstrap – [True, False]

  max_features – [“sqrt”, “log2”, None]

- *Support Vector Machines*

  “kernel”: [“rbf”, “sigmoid”],

  “C”: [0.05, 0.1, 0.5, 1, 10, 100],

  “decision_function_shape”: [“ovo”, “ovr”],

  “gamma”: [“auto”, “scale”]

- *Explainable Boosting Machine*

smoothing_rounds – [500, 1000]

cyclic_progress – [0, 0.5]

max_bins – [1024, 2048]

Para o caso do SVM, não foi considerado o *kernel="linear"* devido ao tempo de execução elevado. Para este conjunto de dados, em contrapartida foi considerado um outro, "sigmoid". Os parâmetros considerados como mais adequados foram os seguintes:

- *Logistic Regression* – [max_iter=1000, solver="lbfgs", C=0.01]
- *Random Forest* – [n_estimators=1000, boostrap=False, max_features="sqrt"]
- *Support Vector Machines* – [kernel="rbf", C=1, decision_shape_function="ovo", gamma="scale"]
- *Explainable Boosting Machine* – [smoothing_round=500,cyclic_progress=0, max_bins=1024]

A figura 4.11 apresenta as matrizes de confusão obtidas para esta execução.

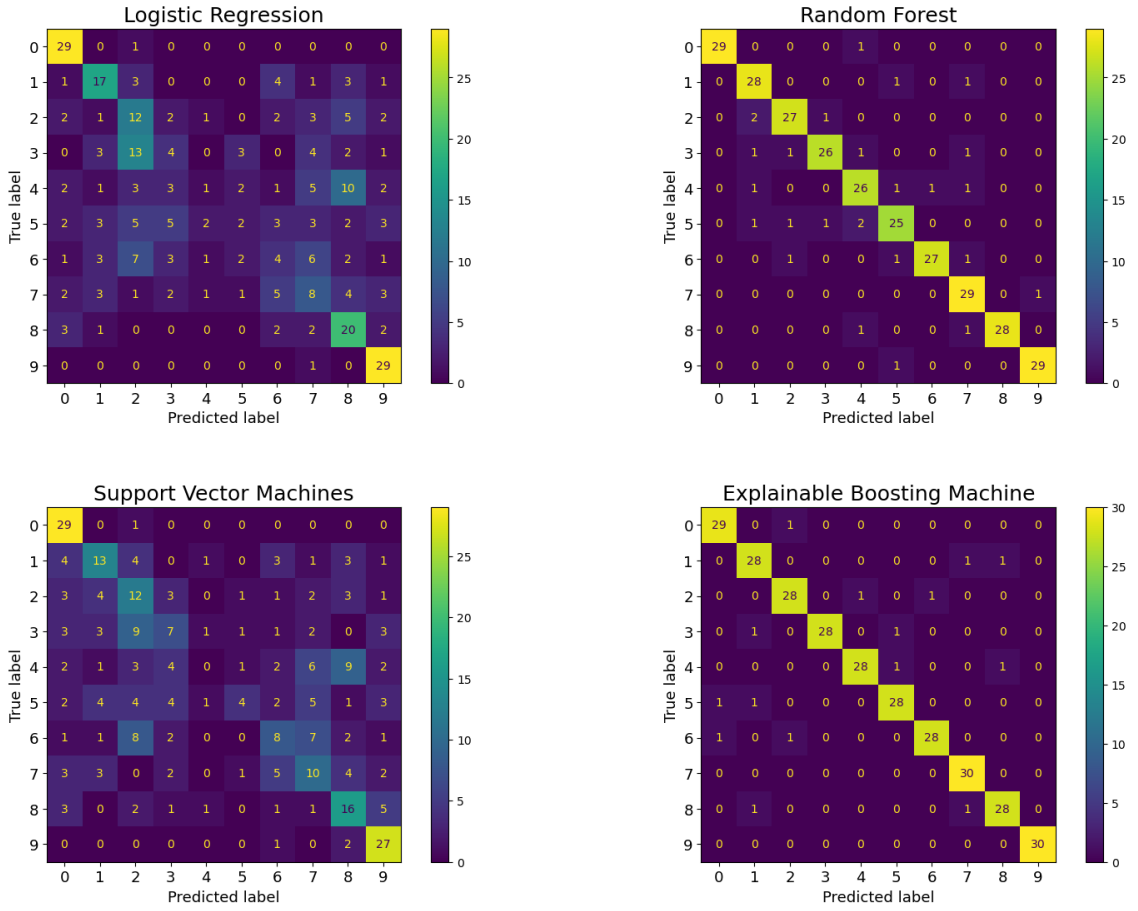


Figura 4.11: *Matrizes de confusão para o conjunto de dados multi-classe*

Das matrizes de confusão foram extraídas as taxas de acerto que cada classificador obteve em cada classe, ilustradas na figura 4.12.

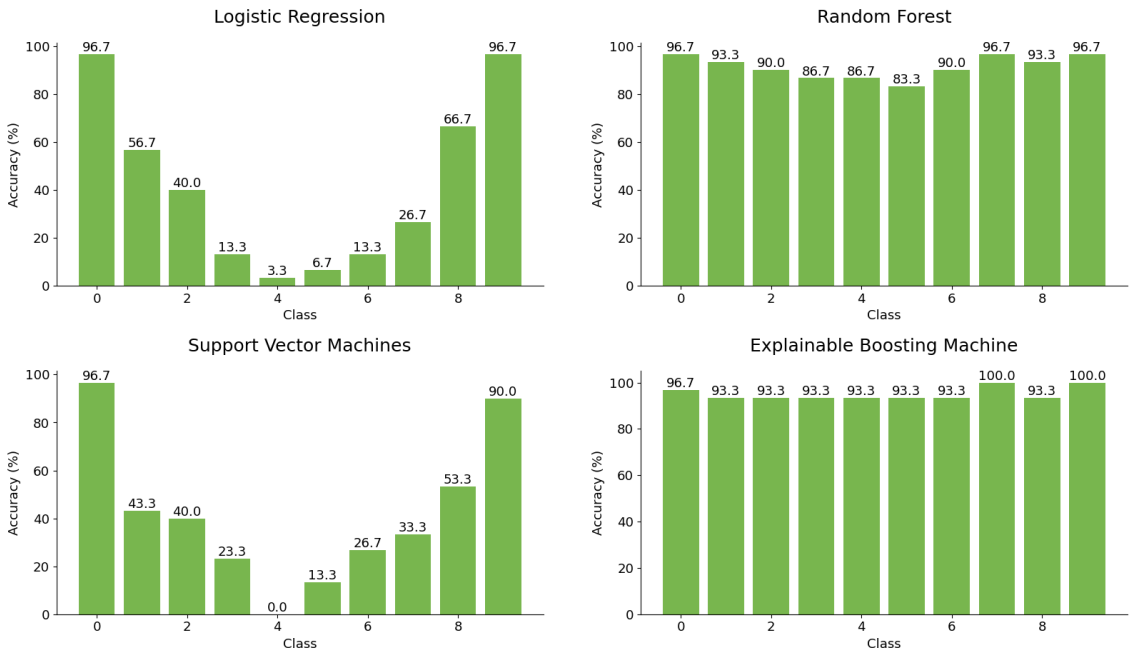


Figura 4.12: *Taxas de acerto para o conjunto de dados multi-classe*

Através da figura 4.11 e figura 4.12 é possível averiguar que o LR (42%) e o SVM (42%) obtiveram taxas de acerto extremamente inferiores ao RF (91%) e EBM (95%). Outro fator relevante é que as classes "do meio" (e.g., 4, 5 e 6) para os casos do LR e SVM apresentam taxas de acerto muito inferiores às das pontas (e.g., 0 e 9). Com um elevado grau de certeza, este efeito deve-se ao facto das classes do meio apresentarem valores mais próximos do centro de intervalo de valores [-100, 100] do que as classes dos extremos. Sendo assim haveria maior possibilidade de confundir estas com valores aleatórios do que se estiverem no limite do intervalo de valores. A figura 4.13 mostra a distribuição de valores para cada característica.

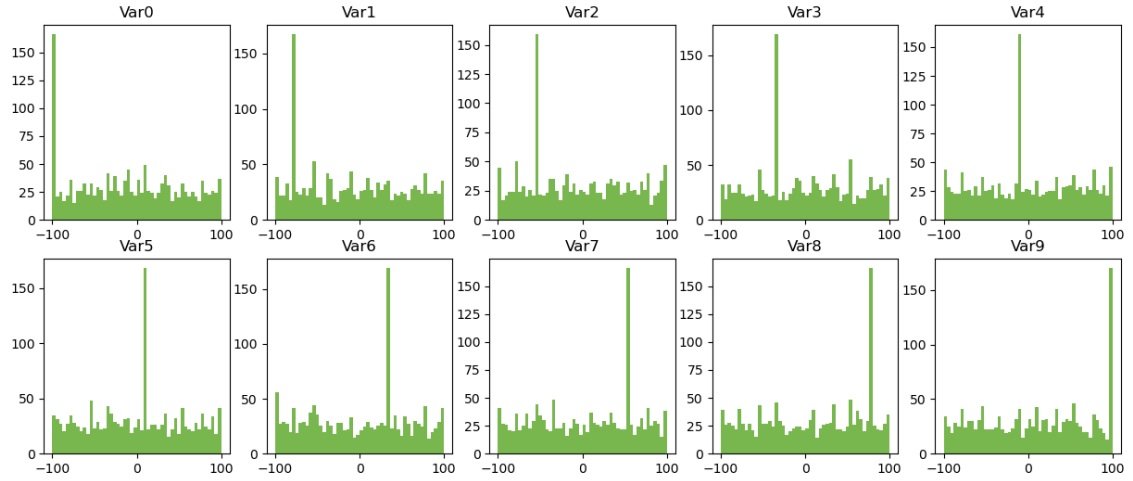


Figura 4.13: *Distribuição de valores de cada característica*

As classes "do meio" apresentam valores mais próximos da média dos valores devido à forma como é gerado o conjunto de dados, sendo separado o intervalo de valores por cada classe. Sendo que cada variável contribui em grande parte para a sua respetiva classe (e.g.,

Var4 -> Classe 4, Var5 -> Classe 5), classes do meio estão mais próximas da média dos dados o que faz com que estes possam ser confundidos com os valores aleatórios devido à forma como estão concebidos os algoritmos de aprendizagem.É razoável assumir que, dado que os valores seguem uma distribuição uniforme, não haveria razão para que valores mais próximos da média contribuam para a "confusão" do algoritmo. No entanto, algoritmos como LR e SVM podem ser bastante afetados por ruído e sendo que o modo de gerar o conjunto de dados é adicionando ruído sob forma de valores aleatórios, estes modelos sofrem para distinguir os pontos de classes próximos à média do ruído.

Foram feitas outras amostragens de treino e teste para comprovar a consistência dos resultados obtidos e verificou-se que estes são consistentes, tendo sido obtido resultados semelhantes nas 10 amostragens feitas, com baixo desvio padrão.

A instância usada para extrair explicações locais encontra-se na tabela 4.7, sendo a mesma pertencente à classe 6.

Tabela 4.7: Instância usada para extrair explicações locais em contexto multi-classe

Var0	Var1	Var2	Var3	Var4	Var5	Var6	Var7	Var8	Var9
33,3	17,9	10,5	-52,5	35,6	-25,1	33,3	63,2	89,6	34,5

A figura 4.14 apresenta as explicações locais obtidas usando o LIME.

Através da figura 4.14, temos que o único classificador que acertou na classe da instância foi o RF, sendo que os restantes classificaram a instância como sendo da classe 7. Apesar disso, em todos os casos a Var6 contribui para a classificação correta e para o RF e SVM é a de maior relevância para indicar se a instância é ou não da classe 6. Este resultado é coerente com a realidade. O LR considerou a Var3 como mais relevante para fazer esta decisão.

Na figura 4.15 encontram-se as explicações locais extraídas usando o SHAP e o EBM.

O SHAP, para casos multi-classe, segue uma abordagem semelhante ao LIME em termos de visualização sendo necessário escolher uma classe para poder visualizar as contribuições de cada variável para a mesma. O EBM adiciona informação da contribuição de cada variável para cada classe. Em todos os casos a variável de maior importância é a Var6, contribuindo positivamente para a classe 6. A Var0, que possui o mesmo valor da Var6, contribuindo positivamente para a classe 6, detém uma posição de algum destaque, estando em 2.º no LR, em 3.º no SVM e EBM e em 4.º no RF. No caso do EBM, é o facto da Var6 também contribuir negativamente em grande medida para a negação das outras classes. Este comportamento tem sentido na medida que, se a Var6 apresenta um valor que indica correlação com a classe, isto significa a exclusão de todas as outras.

De seguida, foram extraídas explicações globais, estando as mesmas ilustradas na figura 4.16.

Excluindo o EBM, os restantes classificadores consideram que as variáveis são tanto mais importantes quanto mais afastadas estão da média do domínio dos valores. Para o caso do

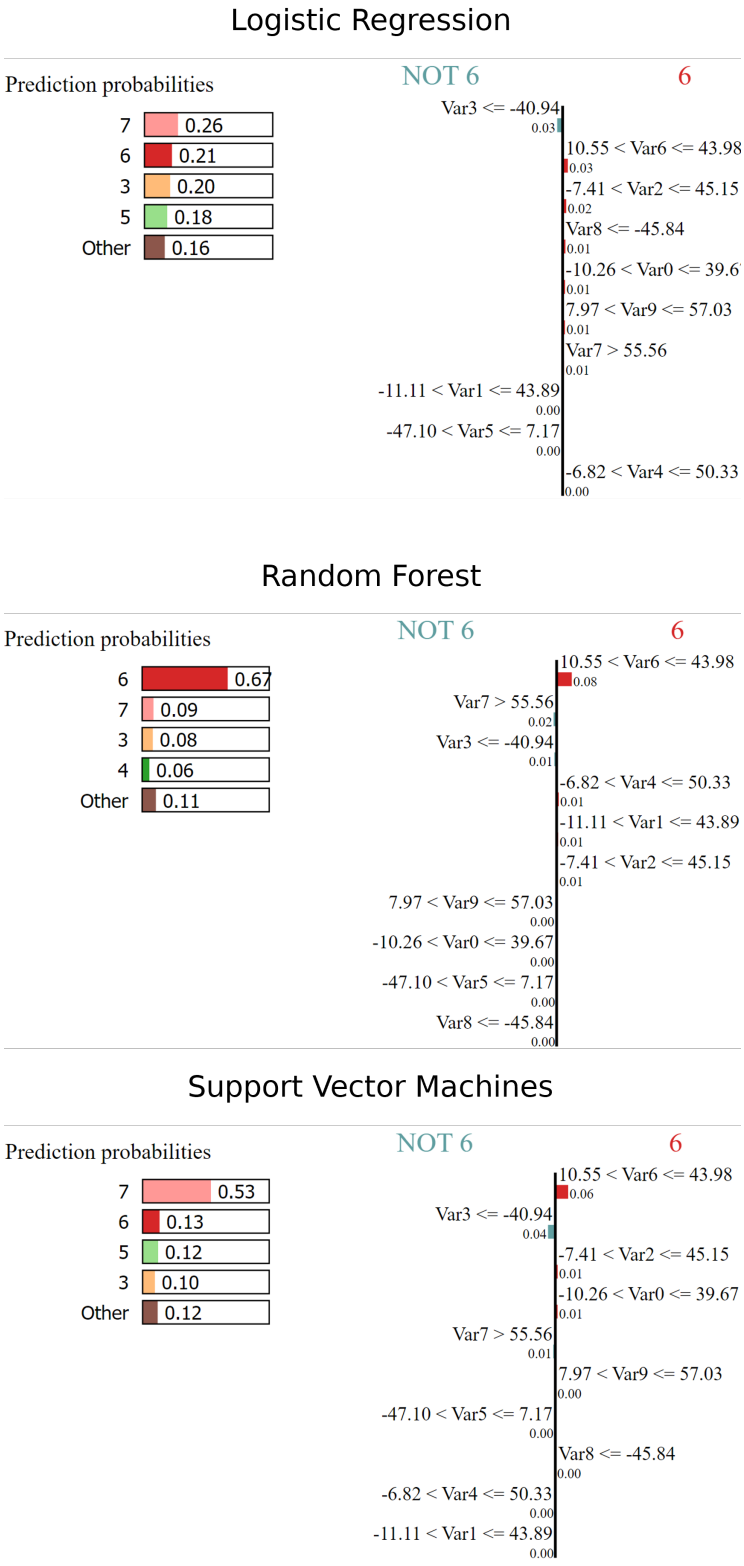


Figura 4.14: *Explicações locais para classificação multi-classe usando o LIME*

Figura 4.15: *Explicações locais para classificação multi-classe usando o SHAP e o EBM*

RF, em primeiro lugar estão as variáveis dos extremos, Var9 e Var0. De seguida, posicionam-se as interiores a essas, Var8 e Var1, sendo as Var4 e Var5 as menos relevantes. Para o LR e SVM é visível um comportamento semelhante, mas menos linear. Também para estes três classificadores é possível ver que cada variável contribui mais para a sua respetiva classe, tal como esperado. O EBM apresenta um comportamento menos óbvio, apesar de colocar as variáveis dos extremos como mais relevantes, as duas seguintes são as do centro, que na teoria seriam as menos relevantes. Isto poderá dever-se ao facto deste modelo não ser tão sensível ao ruído como os restantes, ou não da mesma forma. Um fator que contribui para a confirmação deste efeito é o facto de, no geral, a importância das variáveis é mais balanceada no EBM. Dado que cada variável apresenta a mesma importância para a sua respetiva classe, na teoria, todas as variáveis deveriam ter a mesma relevância, só que para classes diferentes, neste aspeto o EBM capta melhor este detalhe.

O EBM também permite a análise de contribuição global de cada característica, de forma mais detalhada, apresentada na figura 4.17 para a Var1.

Na zona onde a Var1 apresenta correlação com a classe 1 (-77,7), existe um pico de importância para a afirmação desta classe e negação das restantes. Excluindo este pico, esta variável apresenta pouca relevância, já que o restante se trata de valores pseudo-aleatórios.

Testou-se a utilização do conjunto de dados gerado através de distribuição normal com os seguintes parâmetros:

1. Desvio Padrão (std_dev) – [1.75 1.65 1.55 1.45 1.35 1.25 1.15 1.05 0.95 0.85];
2. Separação de classes – ($class\_sep$): [0.65 0.65 0.65 0.65 0.65 0.65 0.65 0.65 0.65 0.65];

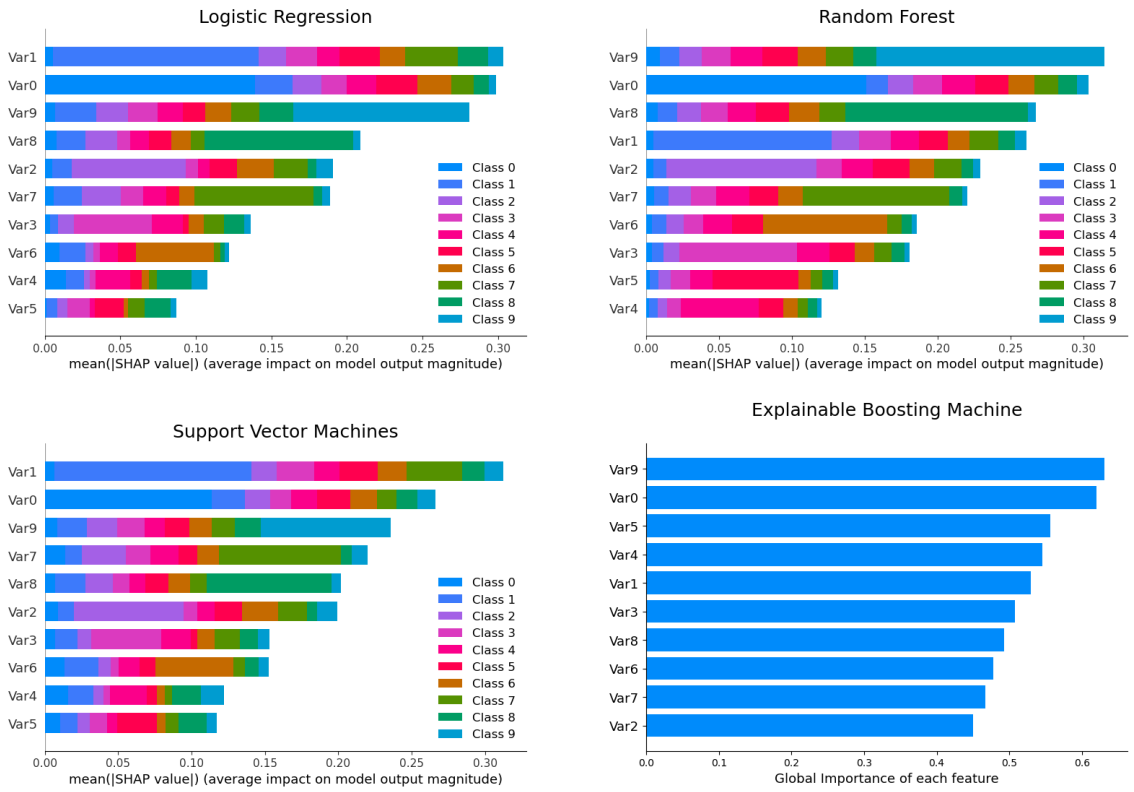


Figura 4.16: *Explicações globais para classificação multi-classe usando o SHAP e EBM*

Figura 4.17: *Explicação do comportamento da Var1 usando o EBM*

3. Valor Mínimo – -100;
4. Valor Máximo – 100;
5. Número de linhas – 1500;
6. Número de colunas – 10;
7. Número de classes – 10.

As taxas de acerto obtidas, para cada classe, encontram-se na figura 4.18.

Através da figura 4.18, temos que as taxas de acerto do LR e do SVM melhoraram substancialmente. Esta é uma prova sólida de que o método de criação do conjunto de dados através

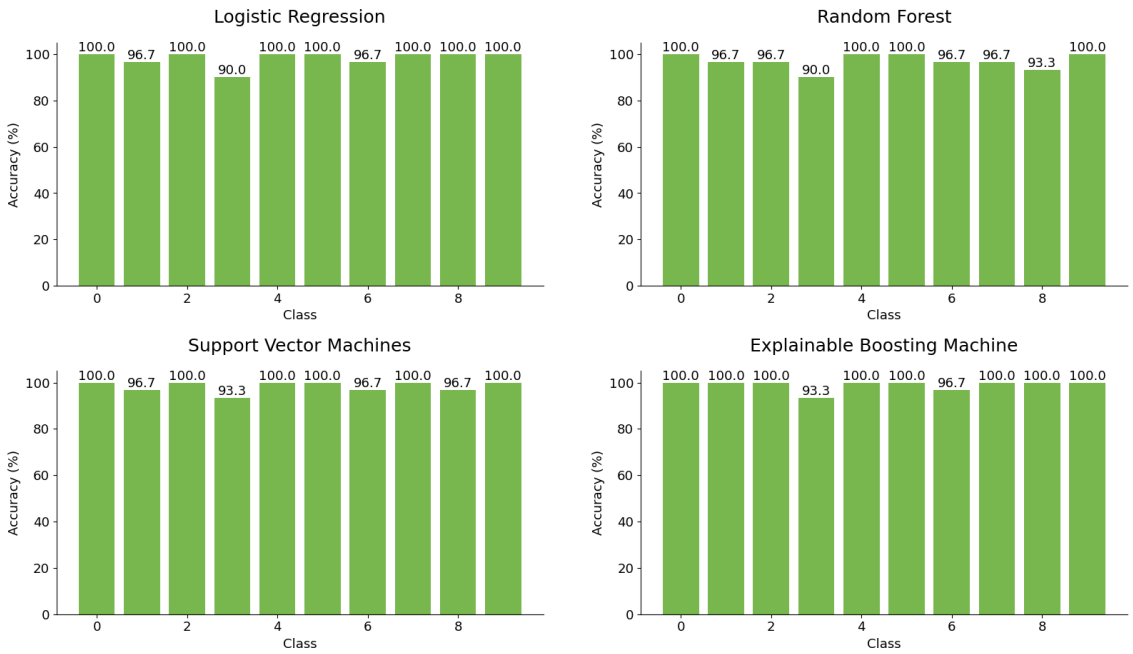


Figura 4.18: *Taxas de acerto para as várias classes usando o conjunto de dados de distribuição normal*

da adição de ruído é o causador do decréscimo do desempenho dos dois classificadores. O contexto do conjunto de dados criado através de distribuições normais possibilita que estes dois classificadores mantenham uma postura competitiva face ao RF e ao EBM. As taxas de acerto foram as seguintes: LR – 98%; RF – 97%; SVM – 98%; EBM – 99%. O melhor classificador foi o EBM, com margem mínima em relação aos restantes.

As explicações globais extraídas para este conjunto de dados encontram-se na figura 4.19.

Excetuando o caso do RF, que ordenou as variáveis corretamente, os restantes classificadores apresentam alguma dificuldade em fazê-lo. O LR, SVM e o EBM apresentam dificuldades sobretudo nas características menos relevantes. Isto pode dever-se ao facto do desvio padrão, à medida que se aumenta o índice da variável, ser diminuído por 0.1. Quanto menor for o desvio padrão, maior será o impacto deste decréscimo, pelo que as variáveis de maior índice (com menor desvio padrão) são mais distinguíveis do que as que têm um menor índice (com maior desvio padrão).

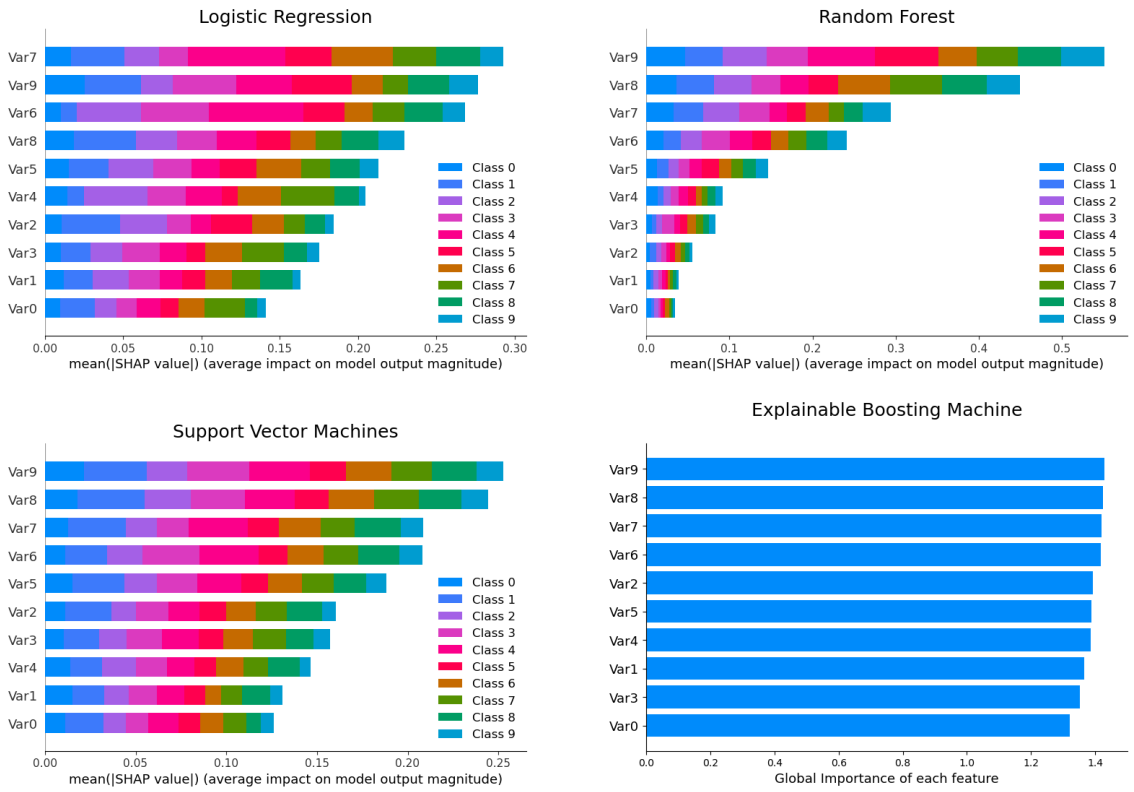


Figura 4.19: *Explicações globais do conjunto de dados multi-classe com distribuição normal usando SHAP e EBM*

4.3 Dados Reais

4.3.1 Deteção de Alzheimer Através de Escrita

Nesta secção ir-se-á abordar o conjunto de dados DARWIN enunciado anteriormente na secção 3.1.1.

O primeiro passo considerado, foi o de analisar o domínio de valores de cada coluna para identificar se a mesma seria categórica ou numérica. Muitas vezes, no caso de características categóricas, é necessário transformá-la num domínio numérico. No caso do conjunto de dados DARWIN, apenas existia uma única coluna com valores categóricos, a coluna "ID". Esta apenas tem a função de identificar o indivíduo que estava a ser testado, sendo que cada instância tem um "ID" diferente. Resolveu-se descartar esta coluna, perfazendo um total de 450 características.

Outro pré-processamento aplicado foi converter o domínio de valores das classes de categórico para numérico da seguinte forma:

- H (Healthy) – 0;
- P (Pacient) – 1.

Tendo já feito os pré-processamentos necessários ao conjunto de dados, é possível proceder à partição do mesmo num conjunto de treino e num de teste. A distribuição seguida foi de 80% para o de treino e 20% para o de teste.

Segue-se a seleção da instância que será utilizada para a extração de explicações locais. A instância selecionada foi a número 3 do conjunto de teste, que não será representada aqui devido à elevada dimensionalidade do conjunto de dados, da classe "0".

Os modelos utilizados apresentam os seguintes hiperparâmetros:

- *Logistic Regression* – [solver="lbfgs", max_iter=500, C=1]
- *Random Forest* – [n_estimators=100, boostrap=True, max_features="sqrt"]
- *Support Vector Machines* – [kernel="linear", C=1, decision_function_shape="ovr", gamma="scale"]
- *Explainable Boosting Machine* – [smoothing_round=200, max_bins=1024, cyclic_progress=1]

Muitos dos parâmetros aqui especificados já recebem estes valores por omissão. Devido à presença de alterações no caso do LR onde as *max_iter* são 100 por omissão, e no caso do SVM onde o *kernel* é "rbf" por omissão, decidiu-se apresentar aqui o valor dos parâmetros considerados como principais. No caso do LR, esta alteração teve origem num aviso que apareceu, pois o número de iterações era insuficiente para o algoritmo convergir. No caso do SVM, deveu-se a resultados muito piores usando o *kernel* por omissão.

Depois de treinados e testados, os modelos obtiveram os resultados presentes na tabela 4.8, obtidos através das matrizes de confusão ilustradas na figura 4.20.

Tabela 4.8: Resultado das métricas para a deteção de Alzheimer

Métrica	LR	RF	SVM	EBM
Taxa de acerto	**0.89**	**0.89**	**0.86**	**0.89**
Taxa de falsos positivos	**0.07**	**0.07**	0.13	**0.07**
Taxa de falsos negativos	**0.15**	**0.15**	**0.15**	**0.15**
Precisão	**0.94**	**0.94**	0.89	**0.94**
Cobertura	**0.85**	**0.85**	**0.85**	**0.85**
F-Score	**0.89**	**0.89**	0.87	**0.89**

Na figura 4.20 temos que todos os classificadores obtiveram matrizes de confusão iguais, exceto o SVM, o que se poderá verificar no valor das métricas. Através da tabela 4.8 é possível averiguar que, ao nível de taxa de acerto, os classificadores tiveram um desempenho bastante semelhante, sendo que o único que ficou abaixo de 89% de acerto foi o SVM. Em relação a falsos positivos, o SVM apresenta o pior desempenho, o que resultou também na pior taxa de acerto. Uma métrica, talvez mais importante que a taxa de acerto no domínio médico, é a dos falsos negativos, para a qual todos os modelos apresentam o mesmo desempenho. A precisão, dado que decresce com o aumento dos falsos positivos, também foi afetada negativamente no caso do SVM. Em relação à cobertura, todos os classificadores obtiveram o mesmo desempenho. O *f-score*, foi menor no SVM, dado que é uma média harmónica da cobertura e precisão.

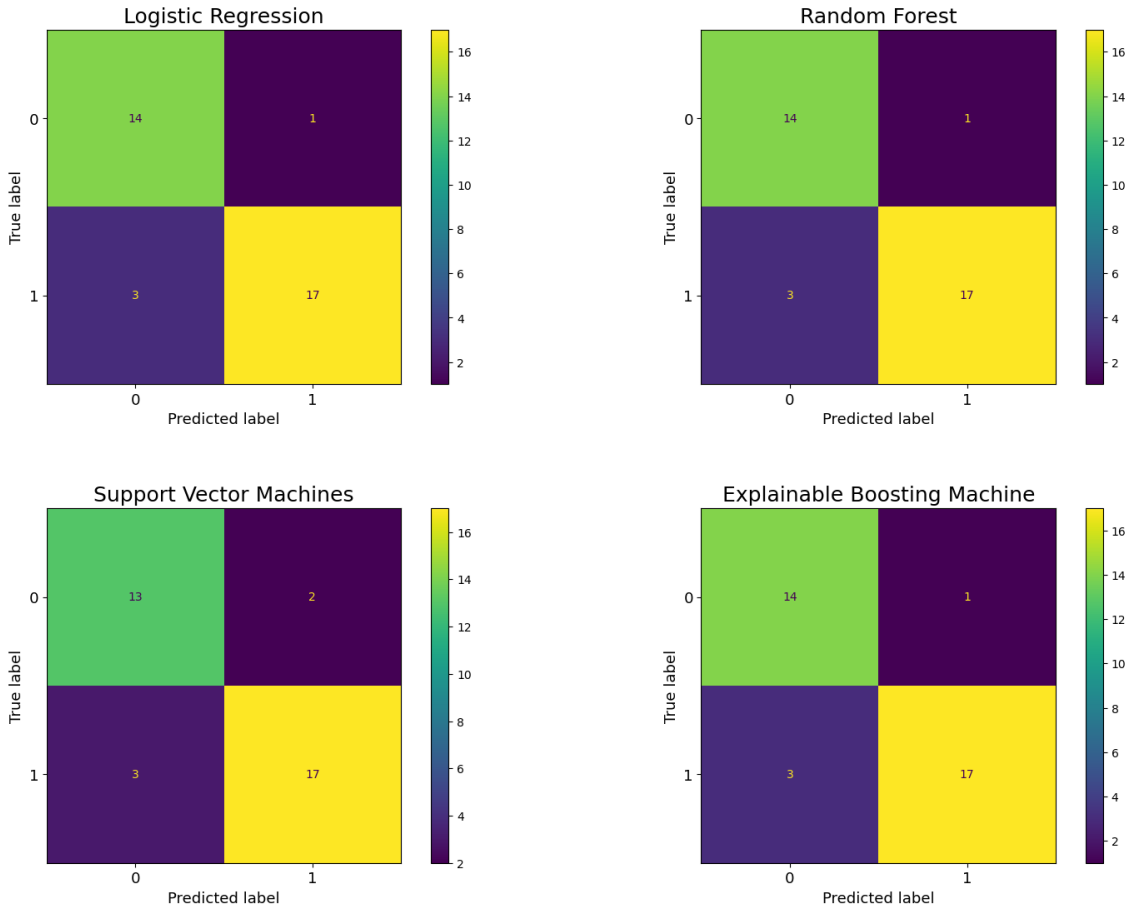


Figura 4.20: *Matrizes de confusão para a deteção de Alzheimer*

Em [13], também foram feitos testes usando diversos classificadores. Para além de considerar como tarefa de classificação todas as características disponíveis, foi também feita a classificação de cada tarefa em separado. Verificou-se que, em quase todos os casos, as tarefas analisadas em separado obtêm menor taxa de acerto do que considerando todas as tarefas. Isto pode provar a pertinência de realizar diversas tarefas, dado que estas testam diferentes aspetos relevantes para a deteção de Alzheimer. Os resultados apresentados neste artigo e os obtidos não divergem em grande medida, sendo que, foram obtidas as seguintes taxas de acerto para os classificadores LR, RF e SVM, respetivamente: 88,29 (±4,90), 81,86 (±7,20) e 79,00 (±7,55). A melhoria mais significativa em relação aos resultados obtidos foi no RF, sendo os restantes semelhantes. Em relação à sensibilidade, foram apresentados os seguintes valores: 84,17, 90,28 e 77,50. Para este caso, foram obtidos resultados semelhantes para o LR, inferiores para o RF e superiores para o SVM.

Na figura 4.21 temos as curvas PR e ROC para os diversos classificadores utilizados. Os pontos indicados em cada um dos gráficos representa as coordenadas (x, y) dos respetivos classificadores. Na curva PR as coordenadas são representadas por (precisão, cobertura) e na curva ROC são representadas por (taxa de falsos positivos, taxa de verdadeiros positivos). Nestas curvas, está saliente a diferença do SVM para os restantes, tendo em ambas o ponto mais distante do ideal.

Para a extração de explicações locais, teve de ser escolhida uma instância do conjunto

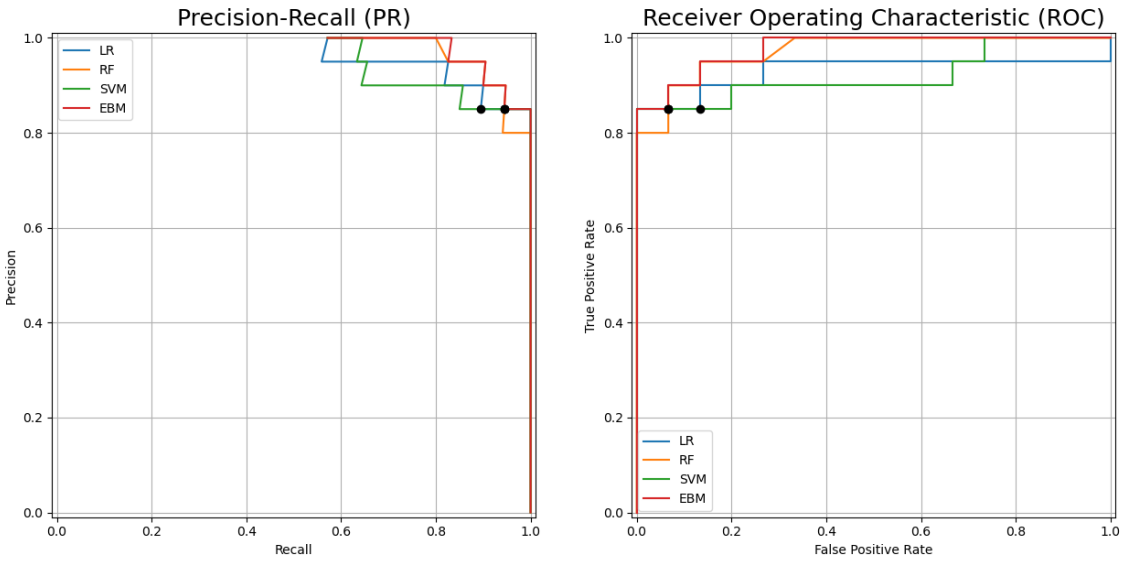


Figura 4.21: *Curvas PR e ROC para a deteção de Alzheimer*

de teste, já que o conjunto de treino é usado para guiar os explicadores na extração de explicações. Devido à elevada dimensionalidade deste conjunto de dados, não seria comportável revelar o valor de todas as características da instância escolhida, no entanto, é importante indicar que a mesma pertence à classe "0" e é necessário ter este fator em conta para analisar as explicações extraídas.

Na figura 4.22 é possível constatar a extração de explicações locais para três classificadores usando o LIME. Cada extração de explicações pode ser dividida em três componentes da esquerda para a direita:

- Probabilidades do próprio classificador;
- Importância das características mais relevantes;
- Características selecionadas e o seu valor.

A característica considerada mais importante com forte influência para a classe "1" é o "airtime_19", indicando o tempo da caneta fora do papel para a tarefa 19. Em geral, parece que um "air_time" alto, mesmo para outras tarefas, contribui para a classe positiva, o que parece fazer sentido, dado que o indivíduo passaria mais tempo sem realizar a tarefa propriamente dita, o que indica dificuldade. Outro grupo de características considerado importante foi o "pressure_var", com 2 ocorrências entre as 10 características mais importantes. O SVM apresenta resultados semelhantes a estes, a maior diferença é que a segunda característica mais relevante é "total_time19" indicando que um menor tempo para realizar a tarefa contribui para a classe negativa. Esta classificação parece fazer sentido, uma vez que quanto menos tempo for necessário para realizar uma tarefa maior será a destreza e, portanto, deverá tratar-se de um indivíduo mais saudável. Para o caso do RF, a extração de explicações mudou consideravelmente a nível de valores de importância. No entanto, apresenta semelhanças na medida em que um menor "air_time" indica uma

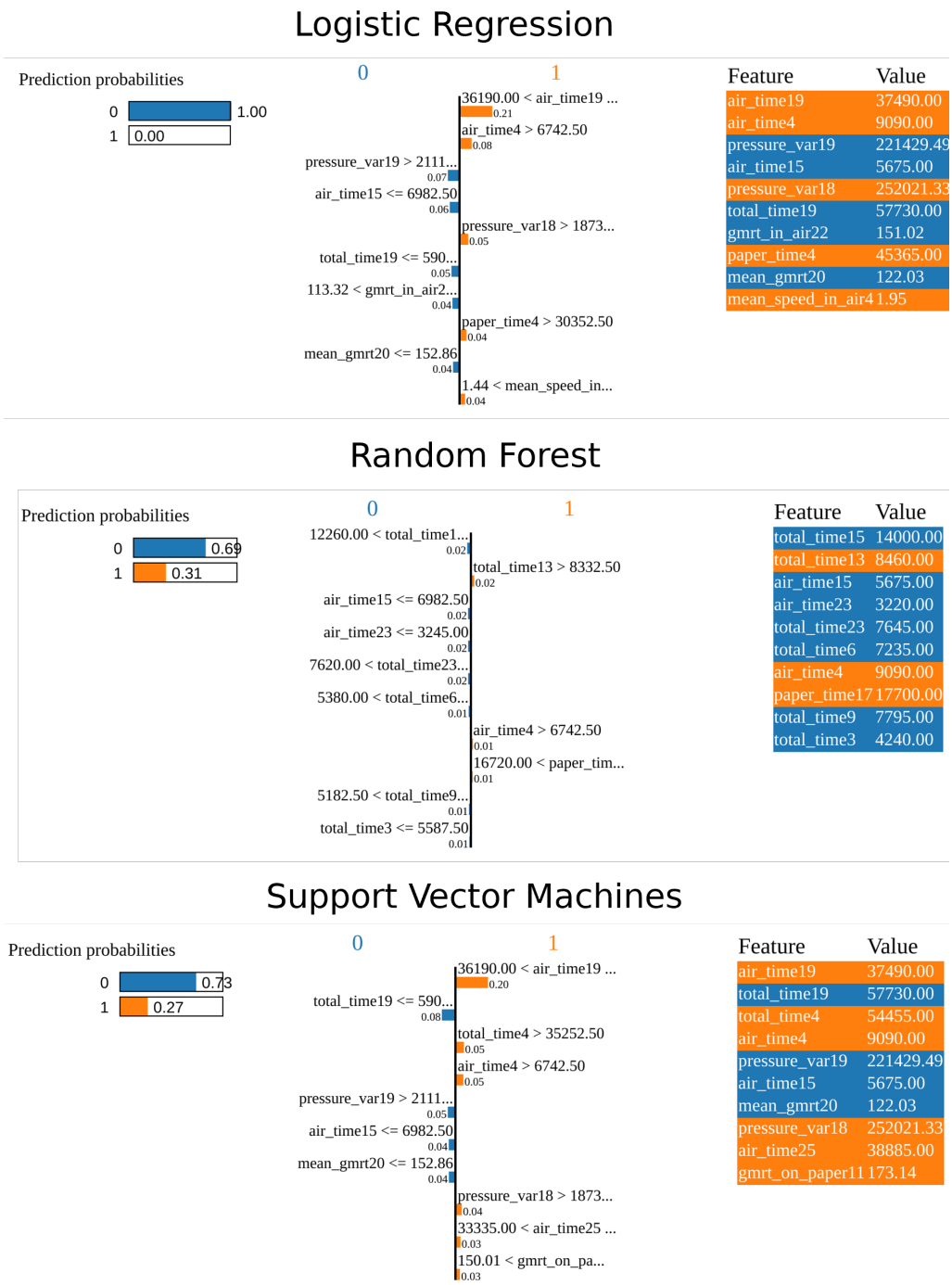


Figura 4.22: *Explicações locais do LIME para a deteção de Alzheimer*

maior inclinação para a classe negativa. Os grupos de características mais frequentes nas 10 características mais importantes foram "air_time" e "total_time". Um consenso entre os três classificadores é o grupo "air_time" assumir um papel de grande importância. Em todos os casos, parece que existe uma discrepância entre as probabilidades do modelo e as explicações, já que em todos a probabilidade pende com uma considerável certeza para a classe dos negativos. No entanto, isto não se verifica nas explicações. Esta aparente falta de fidelidade do LIME, neste caso, pode dever-se ao facto de todas as restantes 440 características estarem ocultas, apesar de poderem ter um peso significativo para a escolha do modelo. Uma outra hipótese é a de o LIME não conseguir capturar relações complexas entre os dados, apesar destas existirem, já que depende apenas da sua inerente linearidade.

Para os mesmos três modelos analisados anteriormente e o EBM foram extraídas explicações locais da mesma instância. Devido ao custo computacional que é exigido por este método, foram feitas amostragens dos dados para os métodos LR e SVM, sendo apenas utilizadas 20 instâncias para o seu treino. Para o RF não foi necessário amostragem. Sendo o EBM um explicador específico ao modelo, este é capaz de, simultaneamente, realizar a tarefa de classificação e extrair explicações desta, pelo que apenas se encontra junto dos restantes para ser possível a sua comparação. Também para tornar mais suave a sua comparação, foram feitas alterações à visualização deste método para se assemelhar à do SHAP e apesar da sua ligeira diferença, apresentam o mesmo tipo de informação. Os resultados estão apresentados na figura 4.23.

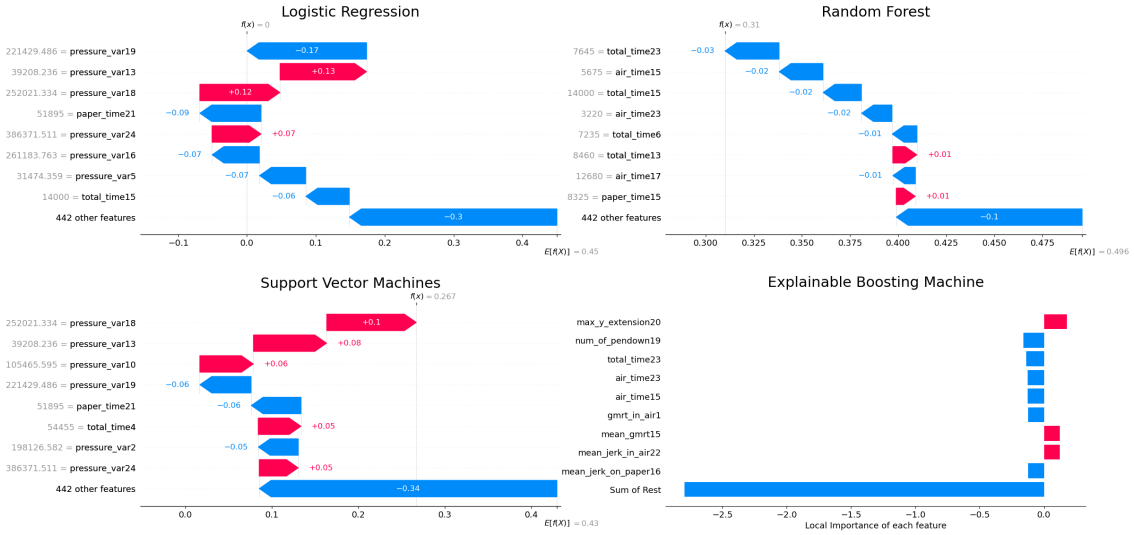


Figura 4.23: *Explicações locais do SHAP e EBM para a deteção de Alzheimer*

Analisando a figura 4.23, é possível notar a diferença de fidelidade em relação ao LIME, dado que o valor da função de decisão, para este caso, é sempre a mesma em relação ao modelo original, como se pode averiguar diretamente abaixo dos títulos. Mais uma vez, é possível notar uma grande semelhança entre as explicações extraídas do LR e do SVM. Para o caso do LR, entre as 9 características mais importantes, 6 delas são do grupo "pressure_var", que indica a variação de pressão, o que faz sentido para uma doença como o Alzheimer, já que à medida que progride afeta cada vez mais as capacidades motoras do paciente. Para o caso do SVM, também se verifica uma predominância do grupo de características

"pressure_var" sendo que apresentam em semelhança 5 características dentro das 9 mais importantes. Outro aspeto interessante é que as 3 características mais importantes para o caso do SVM, pendem para a classe positiva, o que pode contribuir para o aumento de incerteza por parte do modelo, uma vez que a classe verdadeira é a negativa. No caso do RF, parece haver ênfase sobre o tempo total necessário para concluir a tarefa, expresso pelo grupo de características "total_time", com 4 ocorrências, e sobre o tempo com a caneta fora do papel, expresso pelo grupo de características "air_time", com 3 ocorrências. Para o RF, no geral, o SHAP extraiu explicações semelhantes às do LIME. O EBM considerou que a característica mais importante é a extensão em y da tarefa 20, contribuindo para a classe positiva. Outro grupo de características que nunca tinha sido considerado até agora é o "num_of_pendown". Este também identifica a importância no tempo total para realizar uma tarefa e no tempo que a caneta está fora do papel. Em geral, a soma da importância das características que não são tidas como mais importantes têm maior impacto do que as características mais importantes de forma individual, sendo que não são apenas as características mais relevantes que influenciam as explicações, mas sim o conjunto de todas.

Agora será feita uma análise da importância global das características tendo por base a aplicação do SHAP aos modelos utilizados e o EBM. Os gráficos que servirão de base para esta análise estão ilustrados na figura 4.24.

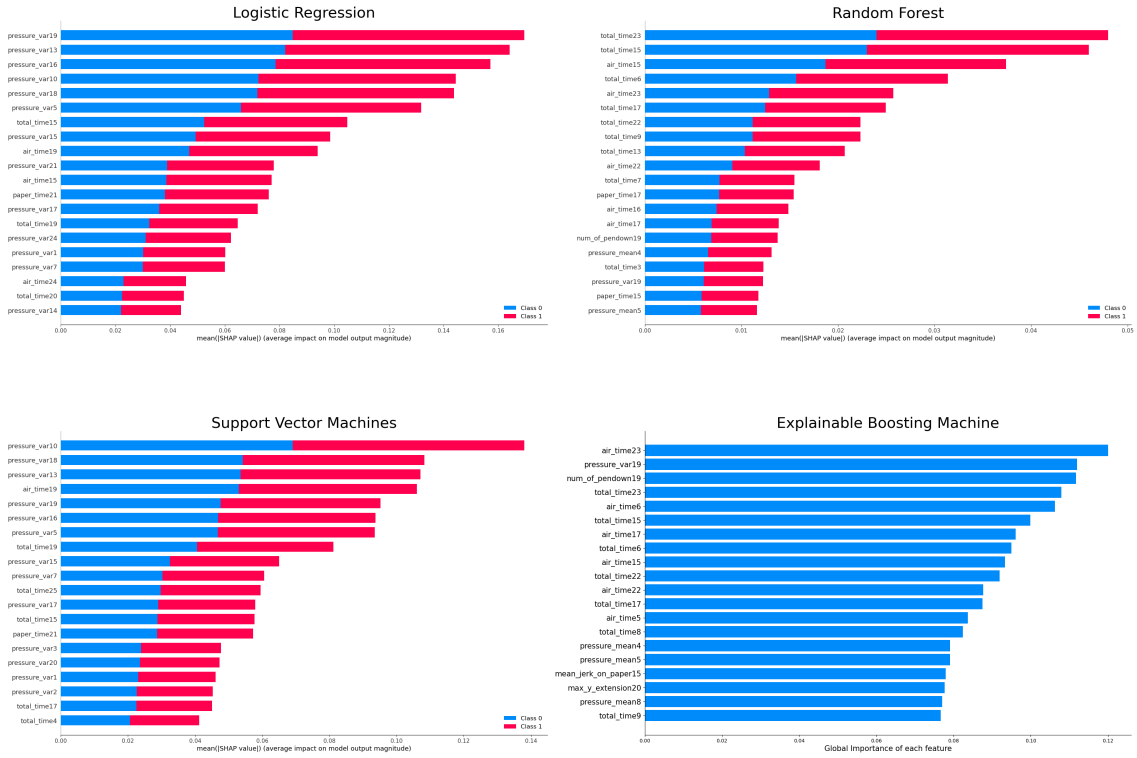


Figura 4.24: *Explicações globais do SHAP e EBM para a deteção de Alzheimer*

Através da figura 4.24 é possível averiguar que os modelos que usaram o SHAP para extrair explicações apresentam barras de duas cores simbolizando cada classe. Este aspeto é mais relevante para quando é aplicada a classificação multi-classe, uma vez que na classificação binária, muitas vezes, as barras das duas classes apresentam a mesma dimensão, ou seja, as características contribuem com igual intensidade para as duas classes.

Em relação ao LR e ao SVM é possível averiguar que as características mais relevantes a nível global assemelham-se com as mais relevantes a nível local. O grupo de características "pressure_var" predomina como sendo o mais relevante com 13 ocorrências em ambos. Em segundo lugar, a nível de número de ocorrências ficaram os grupos "air_time" e "total_time" ambos com 3. No SVM, também predominou o grupo "pressure_var" com 13 ocorrências, sendo que o grupo "total_time" aparece 5 vezes, ficando em segundo lugar a nível de frequência. No RF, o grupo que predominou foi "total_time" com 9 ocorrências, e o segundo grupo com 5 ocorrências foi o "air_time". O grupo predominante para o EBM também foi o "total_time" com 7 ocorrências, apesar da característica mais importante ter sido "air_time23" que pertence ao segundo grupo mais frequente com 6 ocorrências. Algo que todos os modelos parecem concordar é que o grupo "total_time" apresenta uma importância significativa a nível global, o que é intuitivo dado que uma pessoa que apresente capacidade motora debilitada, que é um dos sintomas de Alzheimer, demorará mais tempo para completar as tarefas. Para além disso, as tarefas 19 e 17 aparecem em todas as explicações globais o que pode indicar que apresentam uma maior capacidade para detetar a presença desta doença.

Depois desta primeira análise, foi usado o *GridSearchCV* do *scikit-learn*, com o intuito de chegar a hiperparâmetros que resultassem num melhor desempenho por parte dos modelos. Este método recebe o intervalo de valores a ser testado para cada hiperparâmetro pretendido e realiza validação cruzada sobre cada conjunto destes. A validação cruzada adotada apresentou 7 partições (*folds*) e garante que a distribuição das classes é a mais uniforme possível entre estes através de estratificação (*stratified*). Os hiperparâmetros testados foram os seguintes:

- *Logistic Regression*

  max_iter – [100, 500, 1000, 2000]

  solver – ["lbfgs", "liblinear"]

  C – [0.01, 0.1, 1, 10, 100]

- *Random Forest*

  n_estimators – [100, 250, 500, 1000, 2000]

  bootstrap – [True, False]

  max_features – ["sqrt", "log2", None]

- *Support Vector Machines*

  kernel – ["rbf", "linear"]

  decision_function_shape – ["ovr", "ovo"]

  gamma – ["scale", "auto"]

  C – [0.1, 1, 10, 100, 500]

- *Explainable Boosting Machine*

smoothing_rounds – [500, 1000]

cyclic_progress – [0, 0.5]

max_bins – [1024, 2048]

As métricas que definem o desempenho do modelo foram a taxa de acerto e a cobertura, uma vez que esta última aumenta com a diminuição dos falsos negativos.

Para os valores de hiperparâmetros testados, os melhores encontrados foram os seguintes:

- *Logistic Regression* – [max_iter=500, solver="lbfgs", C=0,01]
- *Random Forest* – [n_estimators=100, bootstrap=False, max_features="log2"]
- *Support Vector Machines* – [kernel="rbf", decision_function_shape="ovr", gamma="scale", C=500]
- *Explainable Boosting Machine* – [smoothing_rounds=500, cyclic_progress=0, max_bins=1024]

Os resultados das várias métricas para os hiperparâmetros selecionados encontram-se expressos na tabela 4.9.

Tabela 4.9: Resultado das métricas para a deteção de Alzheimer depois do GridSearchCV

Métrica	LR	RF	SVM	EBM
Taxa de acerto	**0.91**	0.89	**0.91**	0.89
Taxa de falsos positivos	**0.0**	0.07	0.07	0.07
Taxa de falsos negativos	0.15	0.15	**0.1**	0.15
Precisão	**1.0**	0.94	0.95	0.94
Cobertura	0.85	0.85	**0.9**	0.85
F-Score	**0.92**	0.89	**0.92**	0.89

Através da tabela 4.9 é possível averiguar que apenas os modelos LR e SVM obtiveram melhorias nas métricas, sendo agora os de melhor desempenho. Apesar de terem a mesma taxa de acerto, nem todas as métricas destes dois modelos são iguais. Enquanto o LR privilegiou a redução de falsos positivos, o SVM privilegiou a redução de falsos negativos. Dado que se trata de um contexto médico, uma reduzida taxa de falsos negativos sobrepõe-se a uma reduzida taxa de falsos positivos. O SVM, que numa primeira instância obteve piores resultados, agora torna-se o preferível. Depois da procura pelos melhores parâmetros, o SVM conseguiu melhores resultados tendo por base os obtidos em [13], onde foi registado uma cobertura inferior. O LR também obteve resultados superiores em relação à taxa de acerto e taxa de falsos positivos.

Outro teste realizado consistiu em aplicar os vários métodos de aprendizagem a diferentes partições de treino e teste. Este teste utilizou os parâmetros obtidos através do *GridSearchCV* e tem por objetivo aumentar a fiabilidade dos resultados, uma vez que, com conjuntos diferentes é possível reduzir o enviesamento associado a uma única partição. Neste sentido,

foram criadas 10 partições diferentes e testados os modelos. A tabela 4.10 apresenta os resultados obtidos nesta experiência.

Tabela 4.10: Métricas obtidas para 10 partições de treino e teste (média e desvio padrão)

Métrica	**LR**	**RF**	**SVM**	**EBM**
Taxa de acerto	0.81 ± 0.08	**0.91 ± 0.04**	0.82 ± 0.09	**0.91 ± 0.04**
Taxa de falsos positivos	0.17 ± 0.09	0.08 ± 0.09	0.13 ± 0.11	**0.07 ± 0.07**
Taxa de falsos negativos	0.22 ± 0.10	**0.10 ± 0.06**	0.23 ± 0.09	0.11 ± 0.07
Precisão	0.83 ± 0.09	**0.93 ± 0.07**	0.87 ± 0.11	**0.93 ± 0.06**
Cobertura	0.78 ± 0.10	**0.90 ± 0.06**	0.77 ± 0.09	0.89 ± 0.07
F-score	0.80 ± 0.09	**0.91 ± 0.04**	0.81 ± 0.09	**0.91 ± 0.04**

O LR e o SVM diminuíram consideravelmente o seu desempenho, cerca de 10%, tendo uma taxa de acerto de 81% e 82%, respetivamente. Estes resultados levam a concluir que a partição usada na primeira fase e na fase de *GridSearch* contribuiu para que os valores destes classificadores fossem inflacionados. Com estes testes, viu-se que os classificadores que apresentam melhor desempenho são o RF e o EBM, os dois com 91% de taxa de acerto. O valor das métricas entre estes dois classificadores não é suficientemente díspar para concluir se um é melhor que o outro. O mesmo se aplica à relação entre o LR e SVM.

Também foi realizada uma análise do tempo de execução entre as duas fases, apresentada na tabela 4.11.

Tabela 4.11: Comparação do tempo de execução em segundos para as duas fases

Tarefa	LR	RF	SVM	EBM
Treino	0.1	0.3	0.0	55.9
Treino (GS)	4.8	109.6	0.8	7601.4
SHAP	91.7	0.4	99.4	-
SHAP (GS)	89.3	0.6	129.8	-

Através da tabela 4.11 é possível constatar que na primeira fase, sem aplicação do *GridSearchCV*, apenas o EBM demorou um tempo significativo, cerca de 55 segundos. Com um grau elevado de certeza, isto deve-se ao facto de, no treino, realizar os cálculos necessários para a extração de explicações locais e globais. Para comparar justamente o tempo de execução dos modelos ter-se-ia de somar o tempo de extração de explicações ao tempo de treino do modelo. No que toca à duração do SHAP, verifica-se que, mesmo realizando a amostragem dos dados, este requer elevado poder computacional, sendo que este aplicado ao LR demorou cerca de 91 segundos e aplicado ao SVM cerca de 100 segundos. No caso do RF, foi praticamente instantâneo, mesmo sem amostragem, devido à otimização disponível para classificadores baseados em árvores de decisão, referida anteriormente.

Em relação à fase de *fine-tuning*, é possível averiguar que o RF e o EBM foram os que demoraram um tempo significativo, sendo que o RF demorou perto 2 minutos e o EBM demorou cerca de 2 horas e 8 minutos, sendo por uma grande margem, o que mais poder computacional requer de todos os métodos. No que toca ao tempo requerido para a extração de explicações por parte do SHAP, verifica-se que este foi semelhante ao da primeira fase

em todos os modelos, exceto no SVM, onde houve um aumento de aproximadamente 30 segundos. Tal leva à conclusão de que o tempo de execução depende dos parâmetros do modelo ao qual está a ser aplicado o explicador.

4.3.2 Deteção de tumor do cérebro através de imagem

Outro conjunto de dados utilizado foi o de deteção de cancro do cérebro através de imagem, cujas características foram enunciadas na secção 3.1. Apesar destas imagens, inicialmente, serem compostas por 12 bits por píxel em tons de cinzento, estas tiveram de ser convertidas para imagens RGB com 8 bits por canal, processo demonstrado pela figura 4.25.

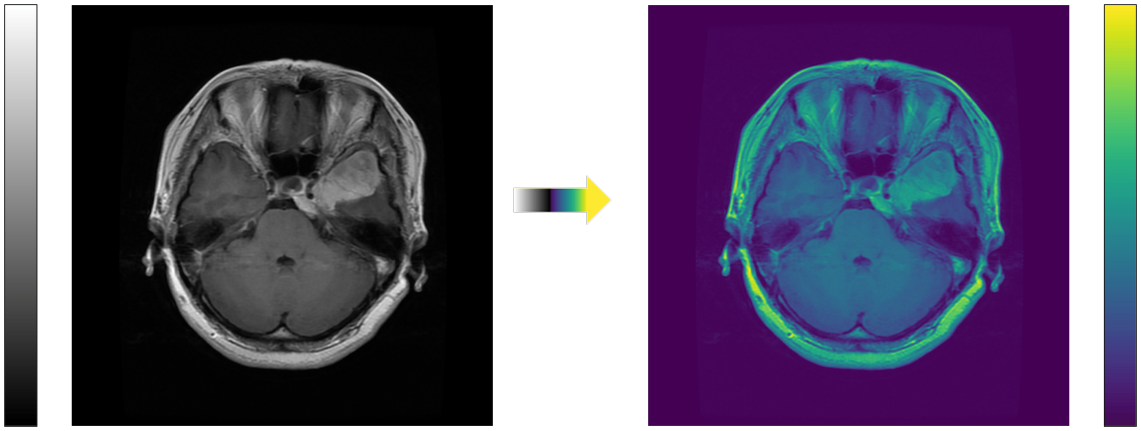


Figura 4.25: *Transformação de palete – tons de cinzento para RGB*

O mapa de cor (*colormap*) utilizado para realizar a transformação foi o *Viridis* disponibilizado pela biblioteca OpenCV e, devido à sua forma de funcionamento, este apenas aceita imagens com 8 bits por canal. A conversão foi realizada através de: img8bits = img12bits / 4095 * 255, seguida de um arredondamento para valores inteiros no intervalo [0, 255]. Também para diminuir o custo computacional da aprendizagem e dado que muitas vezes, as imagens não necessitam de elevada resolução espacial para obter bons resultados, esta foi escalada de (512×512) para (256×256), que é 4 vezes menos resolução do que a original. Sendo assim, as características das imagens usadas para o treino dos modelos são as seguintes:

- Número de instâncias – 3064;
- Tamanho da imagem – $256 \times 256 \times 3$ (Imagem RGB);
- Número de bits por canal – 8;
- Número de classes – 3 (enunciadas no capítulo 3, secção 3.1.2).

Depois de realizados os pré-processamentos mencionados, foram escolhidas 4 modelos CNN para realizar aprendizagem, 1 com arquitetura concebida de raiz e 3 pré-treinadas. A implementação destas redes foi feita usando as bibliotecas *tensorflow* e *keras*.

As redes pré-treinadas tiram partido do conceito de *Transfer Learning* (TL). Este tem por base o uso de uma rede neuronal, muitas vezes complexa, treinada num determinado

domínio que depois é usada num outro domínio. Neste contexto as redes pré-treinadas obtiveram os valores dos seus parâmetros através de aprendizagem sobre o conjunto de dados *ImageNet*. Este é um conjunto de dados que possui cerca de 14 milhões de imagens dos mais variados contextos, tais como animais, objetos ou veículos. Sendo que estas redes foram treinadas num conjunto de dados muito complexo, talvez o conhecimento extraído possa ser utilizado, em alguma medida, noutros contextos, neste caso o de deteção de cancro do cérebro. Esta ideia tem por base a de que existem certas características comuns a diferentes domínios, nomeadamente, características fundamentais, tais como contornos ou formas, que podem ser captadas pelas camadas convolucionais destas redes. Dado que é necessário um elevado poder computacional para treinar uma CNN desta magnitude de raiz, são utilizados os parâmetros aprendidos por esta num contexto muito mais complexo, assumindo que estes serão úteis no contexto pretendido, o que muitas vezes se verifica. A alteração que é necessária realizar é a da camada MLP que existe no final das CNN, uma vez que os pesos dos neurónios estão fortemente acoplados ao domínio com que foram treinados, ao contrário das camadas convolucionais. Para além disso muitas vezes o número de saídas (que corresponde ao número de classes) da rede pré-treinada, difere do número de saídas para o contexto pretendido.

A arquitetura de CNN não pré-treinada é a presente na listagem 4.1.

```
1  inputs = layers.Input(shape=(256, 256, 3))
2
3  x = layers.Conv2D(16, (3, 3), activation='relu')(inputs)
4  x = layers.Conv2D(16, (3, 3), activation='relu')(x)
5  x = layers.MaxPooling2D((3, 3))(x)
6
7  x = layers.Conv2D(32, (3, 3), activation='relu')(x)
8  x = layers.Conv2D(32, (3, 3), activation='relu')(x)
9  x = layers.MaxPooling2D((3, 3))(x)
10
11 x = layers.Conv2D(64, (3, 3), activation='relu')(x)
12 x = layers.Conv2D(64, (3, 3), activation='relu')(x)
13 x = layers.MaxPooling2D((2, 2))(x)
14
15 x = layers.Conv2D(128, (3, 3), activation='relu')(x)
16 x = layers.Conv2D(128, (3, 3), activation='relu')(x)
17 x = layers.MaxPooling2D((2, 2))(x)
18
19 x = layers.Flatten()(x)
20
21 x = layers.Dense(512, activation='relu')(x)
22
23 outputs = layers.Dense(3, activation='softmax')(x)
```

Listagem 4.1: Criação de uma CNN de raiz

Inicialmente, é necessário declarar a dimensão dos dados de entrada. Neste caso, imagens. Esta arquitetura tem por base o padrão de declarar duas (ou mais) camadas convolucionais seguidas de *pooling*. Um tamanho de filtro (*kernel*) utilizado é (3×3), que foi adotado em

todas as camadas convolucionais declaradas. Outros tamanhos comuns são (5×5) e (7×7). A função de ativação usada para todas as camadas convolucionais foi a *Rectified Linear Unit* (ReLU), que apresenta larga utilização neste contexto.

Outro detalhe que se encontra nesta arquitetura é o de o número de filtros dobrar a cada camada de *pooling*. A razão de ser desta abordagem é que o nível de profundidade aumenta a cada *pooling*. Isto é, as camadas convolucionais iniciais são mais gerais, tendo em vista captar aspetos mais genéricos da imagem, tais como contornos. À medida que se aumenta a profundidade, estas terão a função de captar aspetos cada vez mais específicos, para os quais poderá ser necessária uma maior quantidade de filtros.

Em relação às camadas de *pooling*, foram usados dois tipos de *pool_size*, nomeadamente (3×3) e (2×2). Em parte, esta abordagem deve-se à tentativa de controlar a quantidade de parâmetros treináveis, sendo que quanto menor for a *pool_size*, menor será a redução de dimensionalidade e, portanto, existirão mais parâmetros. Foi optado por não se reduzir de forma tão agressiva nas camadas mais profundas, uma vez que o detalhe aqui está mais condensado, pelo que poderá ser necessária uma maior precisão.

Depois das camadas convolucionais e de *pooling*, é criada uma camada densa com 512 neurónios (duas vezes a resolução inicial da imagem). Este número aparece também com a tentativa de controlar o número de parâmetros treináveis e foi considerado um número suficiente para a tarefa em mãos. Por fim, é declarada a camada de saída com 3 unidades, correspondendo às três classes presentes no conjunto de dados. O número de parâmetros treináveis e total desta CNN é 885395.

Para a utilização de CNN pré-treinadas são necessários dois passos. Em primeiro lugar, é necessário obter os pesos da rede pré-treinada, que para este caso foram escolhidos os pesos obtidos usando o conjunto de dados *ImageNet*. Em segundo lugar, uma vez que o problema a resolver é diferente do resolvido previamente, é descartar as camadas densas, uma vez que estas são específicas do problema, devido à quantidade de parâmetros da camada de saída. As 3 redes pré-treinadas utilizadas encontram-se descritas na tabela 4.12.

Tabela 4.12: CNN utilizadas

CNN	N.º original de params.	N.º de params. convolucionais
VGG16	∼138 M	∼14,7 M
MobileNetV2	∼3,4 M	∼2,2 M
ResNet50	∼25,6 M	∼23,6 M

Na listagem 4.2 está a construção da ResNet50 para o problema de deteção de cancro no cérebro através de imagem. As restantes CNN seguem o mesmo modo de operação.

```
1 base_model = ResNet50(weights='imagenet', include_top=False, input_shape
      =(256, 256, 3))
2
3 for layer in base_model.layers:
4     layer.trainable = False
5
6 x = base_model.output
```

```
x = layers.GlobalAveragePooling2D()(x)
x = layers.Dense(512, activation='relu')(x)
predictions = layers.Dense(3, activation='softmax')(x)

tl_rn = models.Model(inputs=base_model.input, outputs=predictions)
```

Listagem 4.2: Criação de uma CNN pré-treinada

Na listagem 4.2 é possível averiguar que, em primeiro lugar, cria-se a rede neuronal pretendida, sem a camada densa ($include\_top = False$) e com os pesos do *ImageNet*. De seguida, "congelam-se" os pesos das restantes camadas. Depois, realiza-se o *GlobalAveragePooling* de modo a reduzir consideravelmente a dimensionalidade, controlando o número de parâmetros treináveis. De seguida, é declarada uma camada densa com 512 neurónios e a camada de saída com 3 unidades. A quantidade de neurónios obtida para as CNN utilizadas está presente na tabela 4.13.

Tabela 4.13: Parâmetros das CNN utilizadas

CNN	Parâmetros Totais	Parâmetros Treináveis
VGG16	14 978 883	264 195
MobileNetV2	2 195 395	657 411
ResNet50	24 638 339	1 050 627

Depois de declaradas as arquiteturas a utilizar procedeu-se ao treino das mesmas, sendo que para cada uma foi declarado o otimizador *Nadam*, tal como está presente na listagem 4.3.

```
opt = keras.optimizers.Nadam(learning_rate=0.005, beta_1=0.9,
beta_2=0.99)

cnn_raw.compile(loss="sparse_categorical_crossentropy",
optimizer=opt, metrics=["accuracy"])

cnn_raw.fit(train_dataset, epochs=10)
```

Listagem 4.3: Declaração do otimizador

Através da listagem 4.3 é possível averiguar que a taxa de aprendizagem (*learning_rate*) apresenta o valor 0,005, quando o valor por omissão é de 0,001. Este valor foi alterado, pois a taxa de aprendizagem por omissão foi considerada mais baixa do que o necessário para obter resultados em tempo útil. Também, para todas as CNN foram consideradas 10 épocas de treino.

Em primeiro lugar, depois de treinadas, verificaram-se as matrizes de confusão presentes na figura 4.26 com taxas de acerto por classe presentes na tabela 4.14.

Na tabela 4.14 está indicada a média de taxa de acerto por classe e a taxa de acerto total. Estas podem diferir porque o número de exemplos por classe também difere. Em relação à taxa de acerto, é possível averiguar que a arquitetura com resultados menos bons foi a feita de raiz, com 78%. De seguida, vem a *MobileNetV2* com 87%, a *VGG16* com 91%

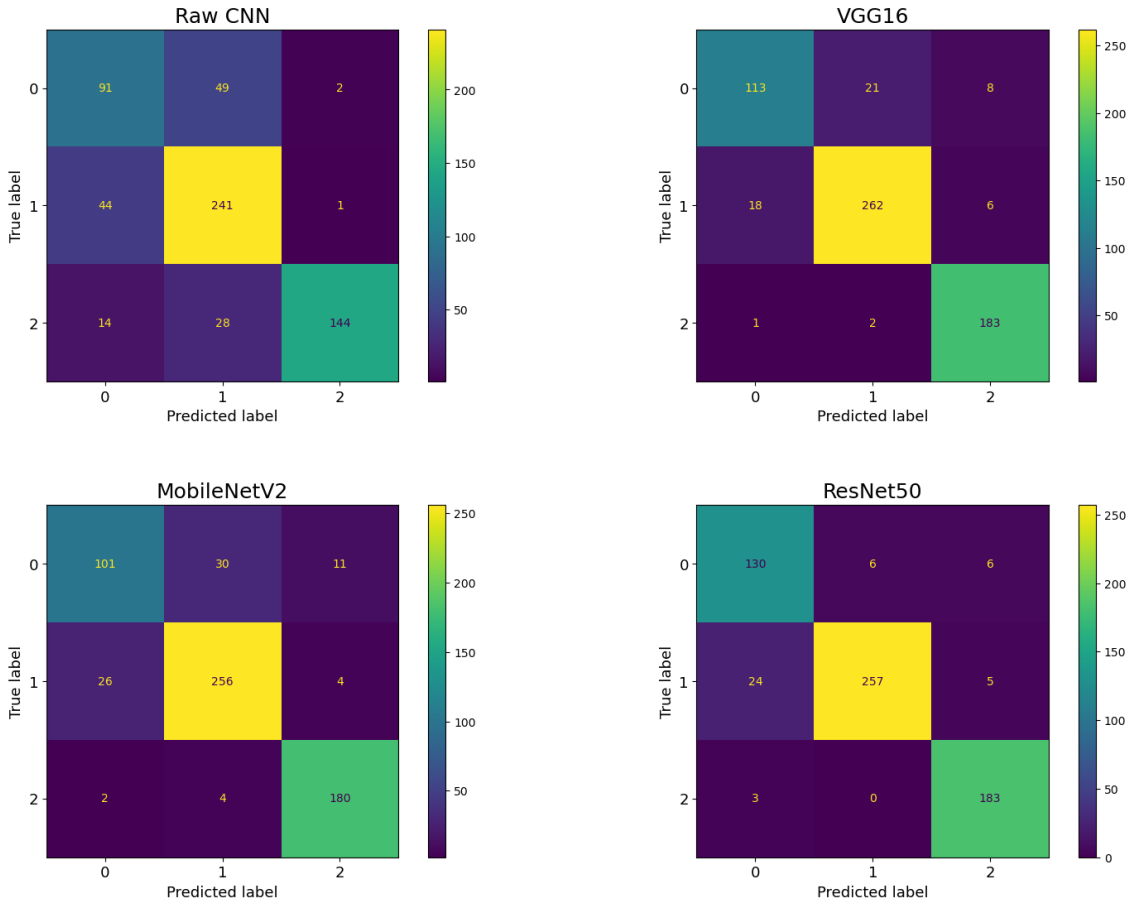


Figura 4.26: *Matrizes de confusão para a deteção de cancro no cérebro*

Tabela 4.14: Resultado das métricas para a deteção de tumor no cérebro

Classes	Raw CNN	VGG	MNet	ResNet	Média por classe
0	0.64	0.8	0.71	**0.92**	0.77
1	0.84	**0.92**	0.9	0.9	0.89
2	0.77	**0.98**	0.97	**0.98**	0.93
Média	0.75	0.9	0.86	**0.93**	-
Taxa de acerto	0.78	0.91	0.87	**0.93**	-

e maior taxa de acerto foi obtida pela *ResNet50* com 93%. Um detalhe visível é que a classe 0, exceto para a ResNet50, é a que obtém piores resultados e a classe 2, excetuando a CNN feita de raiz, obtém melhores resultados. Isto pode indicar que o tipo de tumor 0, "Meningioma", é, em média, mais difícil de perceber e pode ser confundido, como se pode ver através das matrizes de confusão com o tipo de tumor 1, "Glioma". O tipo de tumor 2, "Pituitary tumor", é mais facilmente discriminado.

O passo seguinte foi a extração de explicações, sendo que apenas foram extraídas explicações locais das mesmas, dado que ao nível de imagem, não são disponibilizadas explicações globais, contudo nalguns contextos, como é o caso deste, poder ser útil uma vez que se poderia analisar padrões gerais que contribuíssem para cada tipo de tumor. Foi escolhido um exemplo de cada classe para extrair explicações. As explicações para as classes 0, 1 e 2

encontram-se ilustradas, respetivamente, nas figuras 4.27, 4.28 e 4.29. Estas figuras estão organizadas da seguinte forma:

- 1ª coluna – Imagem original
- 2ª coluna – Explicação extraída pelo LIME
- 3ª coluna – Explicação extraída pelo SHAP
- 4ª coluna – Explicação extraída pelo Grad-CAM
- 5ª coluna – Máscara que representa a verdadeira zona do tumor

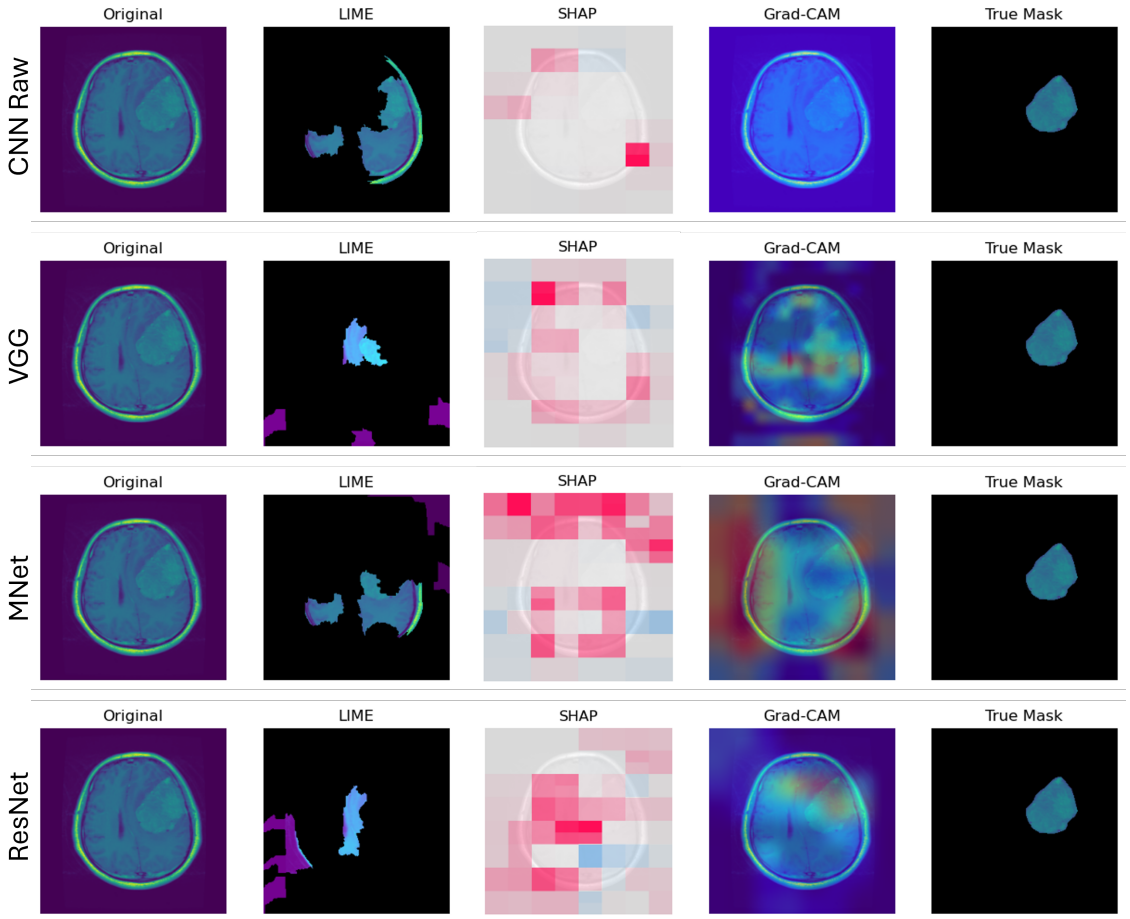


Figura 4.27: *Explicações extraídas para a deteção de cancro no cérebro da classe 0*

Observando as figuras 4.27, 4.28 e 4.29, é possível constatar que, para grande parte dos casos o LIME e o SHAP salientam zonas da imagem que não correspondem ao tumor em si. O LIME, muitas vezes, salienta zonas fora da zona de interesse, representada pela zona de tom escuro ao redor da cabeça do paciente. Algumas exceções a esta regra são o SHAP para a *ResNet*, na classe 0, que apesar de não ser considerada a zona mais relevante, a zona do tumor foi considerada relevante. No LIME da *raw* CNN para a classe 2, apesar de salientar algumas zonas consideradas irrelevantes, o tumor apresenta-se completamente incluído numa das suas saliências. Em adição ao LIME e ao SHAP, foi utilizado o Grad-CAM mencionado na secção 2.4.2 e 2.3.4.3. Para o caso da *raw* CNN, este método não

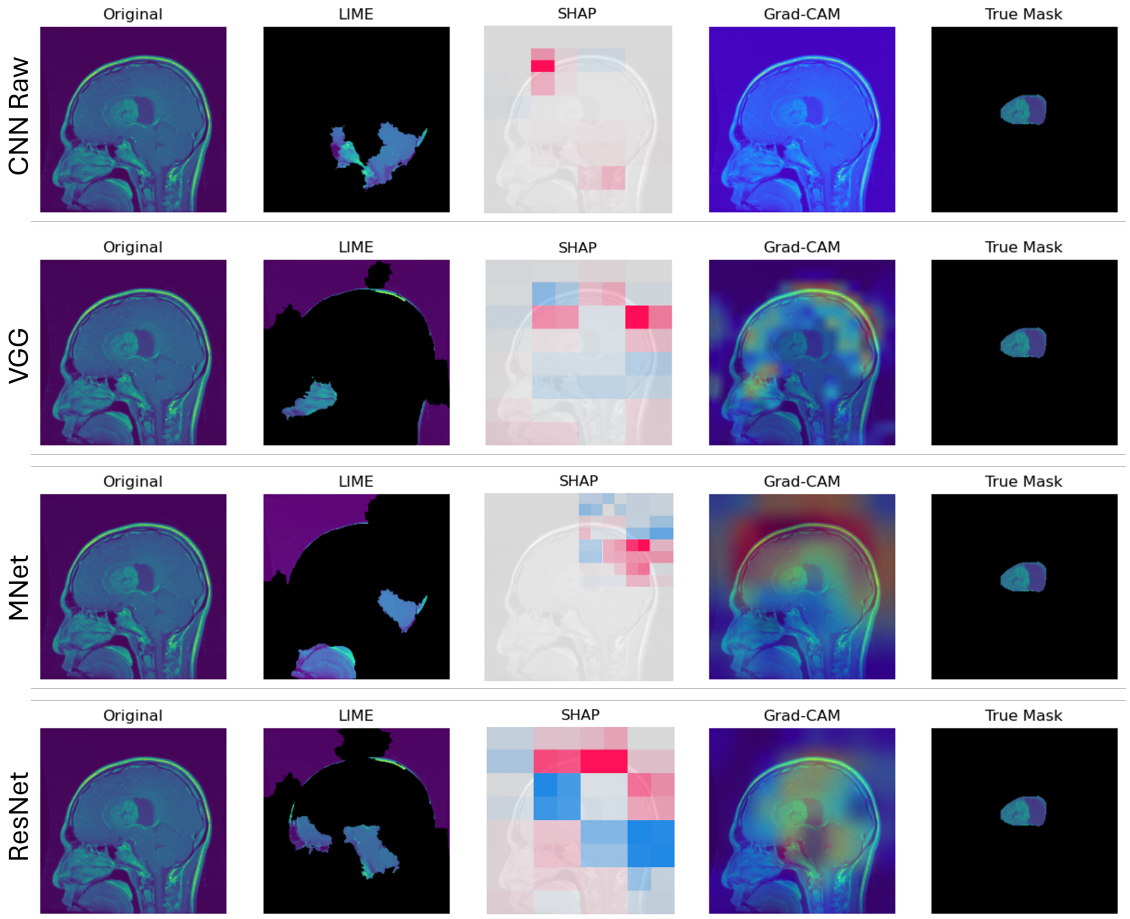


Figura 4.28: *Explicações extraídas para a deteção de cancro no cérebro da classe 1*

apresentou nenhum tipo de saliência, provavelmente devido à falta de complexidade da mesma, uma vez que nas restantes CNN, que apresentam complexidade maior a nível de parâmetros e número de convoluções aparece sempre algum tipo de saliência. Para o caso da *VGG*, este método não apresentou resultados muito satisfatórios, apenas na classe 0, uma porção relevante do tumor é salientada pelo Grad-CAM. Para o caso da *MobileNetV2*, os resultados não melhoram muito significativamente em relação à *VGG*. Apenas na classe 1 e 2 é dada alguma relevância a uma porção do tumor, mas nunca é dada a relevância máxima. A *ResNet* destacou-se das restantes arquiteturas, não só a nível de desempenho como também a nível de explicabilidade. Apesar de não apresentar resultados perfeitos, a zona de interesse marcada por esta arquitetura apresenta sempre uma porção considerável do tumor. Para o caso da classe 1, esta indica relevância próximo à zona do tumor e no interior da mesma. Para a classe 2, apesar da relevância máxima estar um pouco abaixo da zona do tumor, toda esta zona é considerada relevante. O caso da classe 3 é o que mais está condizente com a realidade, sendo que a zona do tumor situa-se no centro da zona indicada como relevante.

Para providenciar métricas mais consistentes com a realidade, reduzindo a influência da aleatoriedade, foram criados 10 conjuntos de treino/teste diferentes, todos eles albergando o conjunto de dados completo. Com estes 10 conjuntos extraíram-se apenas as métricas de desempenho sem considerar a explicabilidade, dado que o objetivo deste passo é meramente

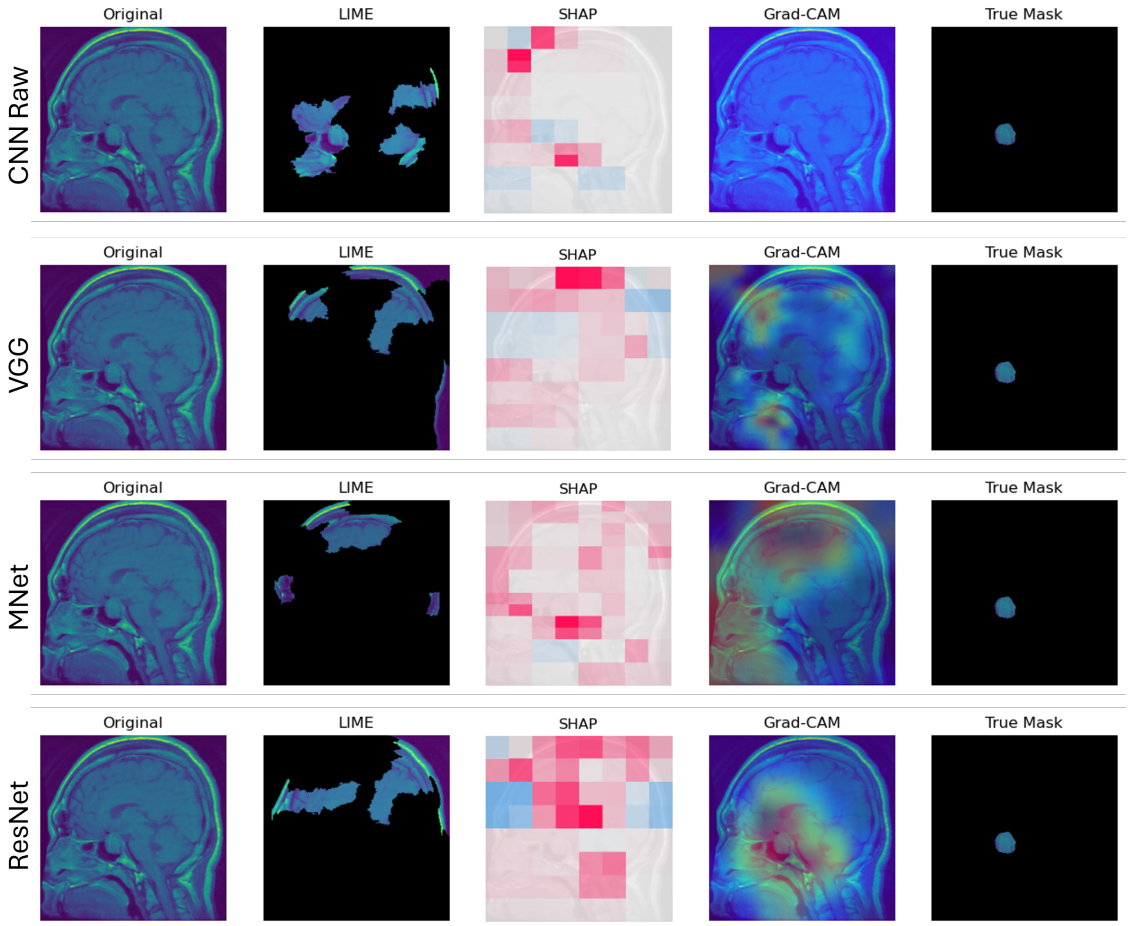


Figura 4.29: *Explicações extraídas para a deteção de cancro no cérebro da classe 2*

constatar se o desempenho obtido anteriormente está de acordo com o desempenho médio ou se é díspar. O resultado das métricas extraídas e respetivos desvios padrão encontra-se na tabela 4.15.

Tabela 4.15: Resultado das métricas e respetivos desvios padrão para a deteção de tumor no cérebro com 10 conjuntos de treino/teste

Classes	Raw CNN	VGG	MNet	ResNet
0	0.52 ± 0.38	0.69 ± 0.16	0.77 ± 0.12	**0.81 ± 0.09**
1	0.89 ± 0.1	**0.93 ± 0.07**	0.88 ± 0.06	**0.93 ± 0.06**
2	0.64 ± 0.45	**0.94 ± 0.04**	0.93 ± 0.05	**0.94 ± 0.03**
Média	0.68 ± 0.24	0.85 ± 0.04	0.86 ± 0.03	**0.89 ± 0.02**
Taxa de acerto	0.73 ± 0.18	0.88 ± 0.02	0.87 ± 0.02	**0.9 ± 0.02**

Na tabela 4.15 é possível averiguar que, novamente a arquitetura que apresentou melhor desempenho foi a ResNet, seguida da VGG, MNet e em último lugar, com uma distância considerável entre as restantes, a CNN feita de raiz. Um facto interessante de comentar é que a CNN feita de raiz apresenta elevados desvios padrão para todas as suas classes, mas em especial para a classe 0 e 2. Isto deve-se ao facto de, em três dos 10 testes, esta ter chegado a um ótimo local que corresponde a classificar todas as instâncias como a classe 1, visto que é a mais numerosa, atingindo uma taxa de acerto de aproximadamente 46,5%.

No geral também se pode constatar que a classe que menor taxa de acerto possui é a 0, a que possui menor número de exemplos. Este efeito não se verifica entre a classe 1 e 2, apesar da última ter significativamente menos exemplos.

A alternativa mais direta para resolver o problema dos ótimos locais, neste caso, seria a de aplicar técnicas de *sampling*, nomeadamente *undersampling* ou *oversampling*, por forma a tornar a quantidade de dados mais equilibrada. Dado que os dados se tratam de imagens o mais direto seria o de aplicar *undersampling*, removendo exemplos das classes maioritárias. No entanto, este processo apresenta a desvantagem de perda de informação, uma vez que dados úteis estariam a ser descartados. Para além disso a única arquitetura que parece ter graves problemas com a discrepância do número de exemplos de cada classe é a criada de raiz, nas restantes este efeito é consideravelmente menos visível. Isto pode levar à conclusão de que o verdadeira problema não é esta discrepância, mas sim a própria arquitetura desta rede, que por algum motivo parece ser suscetível a ótimos locais. Algumas formas de mitigar este problema seria o de alterar o otimizador e os respetivos parâmetros, tais como a *learning rate* ou o *momentum*. Uma outra opção é a de alterar a arquitetura da própria rede, criando um maior número de camadas convolucionais. Um outro aspeto que poderia ajudar seria o de adicionar camadas de *dropout*, estas têm a função de "ligar ou desligar" os neurónios em cada iteração, por exemplo um *dropout* de 40%, por cada passagem, ativa apenas 60% dos neurónios, o que ajuda a reduzir a probabilidade de ótimos locais e sobreaprendizagem (*overfitting*), mantendo a complexidade necessária.

Apesar da *ResNet50* apresentar resultados satisfatórios para o Grad-CAM, no geral a explicabilidade ao nível de imagem, para a deteção de cancro no cérebro deste conjunto de dados, poderia ser melhorada. Uma possível melhoria a aplicar seria realizar *fine-tuning* aos modelos e também aumentar o número de épocas de treino como uma medida de aumento de desempenho. Sendo que a extração de explicações é inerente aos modelos utilizados para as extrair, modelos com maior taxa de acerto, na teoria, também proporcionam explicabilidade mais fidedigna. Outro método que se poderia adotar seria o de treinar de raiz as arquiteturas usadas para que estas pudessem estar mais adaptadas ao problema específico de deteção de tumor no cérebro através de imagens de ressonância magnética. No entanto, para tal seria necessário poder computacional elevado, sendo que ter-se-ia, provavelmente, de recorrer a serviços de *cloud* que facilitem a sua execução, tais como *Google Cloud Platform* (GCP) e *Google Colab*, *Amazon Web Services* (AWS) ou *Microsoft Azure*. Outra abordagem que poderia melhorar os resultados seria o de testar mais tipos de pré-processamento, tais como testar diferentes resoluções de imagem, mapas de cor ou manter as imagens iniciais a tons de cinzento a 12 *bits*, mas descartar o uso do LIME, já que este requer imagens RGB.

5 Conclusões

Este trabalho incide sobre a área de Inteligência Artificial (IA), mais especificamente no subcampo de Aprendizagem Automática (AA). Esta área sofre de um problema de opacidade em grande parte dos seus modelos, no que toca ao processo de decisão, o que pode causar desconfiança para com os mesmos, levando ao seu abandono. Para desvendar as causas das decisões realizadas pelos modelos, surge a área de *eXplainable Artificial Intelligence* (XAI), que é o foco deste trabalho. Em primeiro lugar, foi consultada a literatura de modo a fornecer a contextualização e estado atual da pesquisa nesta área. Através desta pesquisa verificou-se que não existe unanimidade em grande parte dos termos inerentes à XAI, tais como "interpretabilidade", "explicabilidade" e "transparência". Esta heterogeneidade deve-se à subjetividade destes termos, sendo que uma boa explicação depende fortemente do que a observa, não existindo algo que consiga ser entendido/interpretado por todos. Apesar disso, tentou-se definir o significado para estes e outros termos utilizados neste contexto.

Nesta área pode ser feita a distinção de tipos de explicações: visual, numérica e à base de regras. Cada explicação também apresenta um escopo podendo ser local, quando é pretendido que apenas seja explicada uma única decisão ou um conjunto reduzido de decisões, ou global, quando se pretende a explicação de um modelo como um todo. Os componentes que extraem explicações designam-se por explicadores e apresentam diversas relações com o modelo. Caso o modelo seja transparente, então este também é considerado um explicador visto que fornece inerentemente, de alguma forma, justificativas para as suas decisões, por exemplo através de pesos para cada característica. Os que não são transparentes necessitam de componentes externos para providenciar esclarecimentos sobre a sua tomada de decisão. Estes explicadores podem ser agnósticos ao modelo, quando apenas dependem da capacidade preditiva do mesmo, ou específicos ao modelo, quando necessitam de acesso à estrutura interna do modelo, estando feitos apenas para um tipo de arquitetura.

Em adição, existem algumas métricas que se podem aplicar aos projetos que têm em vista a área de XAI. Estas podem ser: funcionais, baseadas em utilizadores ou aplicacionais. As métricas funcionais destinam-se à avaliação do explicador tentando reduzir ao máximo qualquer subjetividade. As métricas baseadas em utilizadores já requerem o envolvimento

de pessoas, aumentando a subjetividade inerente, possivelmente acrescentando informação mais útil do que o primeiro tipo de métricas. As métricas aplicacionais têm por objetivo avaliar o modo de mostrar as explicações, pelo que o seu domínio alberga mais do que conhecimentos de informática e pode englobar ciências de conhecimento direcionado ao humano.

Numa fase seguinte foram apresentados os métodos utilizados, nomeadamente o *Logistic Regression* (LR), *Random Forest* (RF), *Support Vector Machines* (SVM) e *Explainable Boosting Machine* (EBM). Para além disso, foram definidas as métricas adotadas para avaliar o seu desempenho, sendo elas: taxa de acerto, taxa de falsos positivos, taxa de falsos negativos, precisão, cobertura e f1-score. Ainda nesta fase foi apresentada a sequência de ações, através de um diagrama de blocos, necessárias para cumprir todos os objetivos desde a partição do conjunto de dados até à avaliação dos resultados obtidos e das explicações extraídas.

Tendo o modo de operação definido, passou-se à avaliação experimental. Nesta fase escolheram-se dois tipos de dados: sintéticos e reais. Nesta etapa, para além dos classificadores, foram escolhidos os métodos: o *Local Interpretable Model-agnostic Explanations* (LIME), capaz de extrair explicações locais, e o *SHapley Additive exPlanations* (SHAP), capaz de extrair tanto explicações locais como globais.

O objetivo do conjunto de dados sintéticos é averiguar o comportamento dos modelos escolhidos e, principalmente, dos explicadores. Foram criados dois métodos para gerar dados sintéticos, um que tem por base a adição de ruído e outro à base de curvas gaussianas. Nesta etapa foram dadas importâncias diferentes a cada característica e pretende-se constatar se a ordem de importância de características dada pelos explicadores condiz com a utilizada para gerar o conjunto de dados, comprovando a sua pertinência. Foram realizados diversos testes através da mudança de parâmetros que geram o conjunto de dados e verificou-se que o SVM e o LR podem ter problemas para classificação multi-classe quando o método de geração é através de ruído. No entanto, esta dificuldade não se verifica quando são utilizadas curvas gaussianas.

Para o conjunto de dados de deteção de Alzheimer, os resultados mais consistentes apontam para que o RF e o EBM tenham os desempenhos superiores ao nível de todas as métricas. Aqui também se extraíram explicações globais e locais. Através da análise das explicações globais concluiu-se que a variância da pressão, o tempo com a caneta no ar e o tempo total requerido para terminar a tarefa, são os três grupos que mais contribuem para indicar a presença ou ausência de enfermidade.

Para o conjunto de dados de imagens foram treinadas 4 *Convolutional Neural Network* (CNN), uma criada de raiz e as restantes pré-treinadas com os pesos do *ImageNet*, sendo elas: VGG16, MobileNetV2 e ResNet50. Para este caso viu-se que, por uma boa margem, a que apresentou pior desempenho foi a CNN criada de raiz, o que seria de esperar uma vez que apresenta menor complexidade a nível de convoluções e menor profundidade que as restantes. As CNN pré-treinadas apresentaram desempenhos semelhantes, com melhor desempenho para ResNet50. Em relação à extração de explicações, apenas foram extraídas do escopo

local usando o SHAP e o LIME, bem como um método específico, o Grad-CAM. Dado que, para este conjunto de dados, tem-se a localização verdadeira do tumor, é possível comparar o desempenho das várias técnicas de explicabilidade em relação ao seu desempenho. Tendo por base os métodos utilizados, os métodos em geral não apresentam grande acuidade para identificar a zona do tumor com exceção de um caso, nomeadamente o da ResNet50 aplicando o Grad-CAM. Para este cenário, os resultados são sempre informativos da zona certa do tumor, mesmo que não seja claro na totalidade.

5.1 Trabalho Futuro

Relativamente a perspetivas futuras, o trabalho realizado pode apresentar diversas ramificações:

- Experimentar outros métodos de explicabilidade que não tenham sido utilizados nos testes, como o caso do *Layer-wise Relevance Propagation* (LRP) ou o *Sensitivity Analysis* (SA) de modo a comparar as suas explicações com as previamente extraídas.
- Comparar as explicações fornecidas inerentemente de modelos transparentes, como uma árvore de decisão ou o LR, com as explicações extraídas desses mesmos modelos mas através de métodos específicos ou agnósticos ao modelo. Este estudo tem interesse para verificar se as explicações extraídas pelos vários métodos coincidem ou se são díspares. Caso sejam díspares, isso poderia ser uma indicação de falta de fidelidade, já que o modelo transparente apresenta as "respostas verdadeiras" para as justificações das suas decisões.
- Conceber um método que combine diversos métodos, por exemplo juntar os métodos LIME e SHAP, transformando os seus pesos, no caso do LIME e os valores de *Shapley*, no caso do SHAP, para o mesmo domínio tratando as explicações de ambos os métodos como apenas uma e avaliar o seu desempenho. Esta é uma proposta que está numa fase abstrata e teria de ser feita bastante pesquisa para saber se esta ideia é sequer passível de sair do ramo da teoria.
- Conceber um explicador de raiz. Para tornar isto possível haveria duas possibilidades: construir um classificador transparente que apresente bom desempenho ou construir algum tipo de explicador, específico ou agnóstico, que consiga, com fiabilidade, encontrar as justificativas para as decisões dos modelos. Em qualquer um dos casos muito provavelmente o conhecimento requerido teria de vir, não só de informática, mas sobretudo de matemática, dado que os modelos e os explicadores, normalmente baseiam-se em conceitos matemáticos que são depois aplicados a um algoritmo, com uma certa função, pelo que o conhecimento desta área teria de ser largamente expandido.

Bibliografia

[1] A. Adadi e M. Berrada. "Peeking Inside the Black-Box: A Survey on Explainable Artificial Intelligence (XAI)". Em: *IEEE Access* 6 (2018), pp. 52138–52160. DOI: 10.1109/ACCESS.2018.2870052 (ver pp. 9, 13, 15, 17, 18).

[2] P. Adler, C. Falk, S. A. Friedler, T. Nix, G. Rybeck, C. Scheidegger, B. Smith e S. Venkatasubramanian. "Auditing black-box models for indirect influence". Em: *Knowledge and Information Systems* 54.1 (out. de 2017), pp. 95–122. ISSN: 0219-3116. DOI: 10.1007/s10115-017-1116-3 (ver pp. 13, 14).

[3] E. Alpaydın. *Introduction to machine learning*. Fourth edition. Adaptive computation and machine learning. Description based on publisher supplied metadata and other sources. Cambridge, Massachusetts: The MIT Press, 2020. 1691 pp. ISBN: 9780262358064 (ver p. 1).

[4] *Alzheimer's Disease Neuroimaging Initiative (ADNI)*. https://adni.loni.usc.edu/data-samples/access-data/. Accessed: 2024-06-03 (ver p. 32).

[5] A. B. Arrieta, N. D. Rodríguez, J. D. Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila e F. Herrera. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". Em: *Inf. Fusion* 58 (2020), pp. 82–115. DOI: 10.1016/J.INFFUS.2019.12.012 (ver pp. 9–12, 15, 17, 18).

[6] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller e W. Samek. "On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation". Em: *PLOS ONE* 10.7 (jul. de 2015). Ed. por O. D. Suarez, e0130140. ISSN: 1932-6203. DOI: 10.1371/journal.pone.0130140 (ver pp. 16, 25).

[7] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen e K.-R. Müller. "How to Explain Individual Classification Decisions". Em: (ago. de 2010), 1803–1831 (ver p. 17).

[8] N. Barr Kumarakulasinghe, T. Blomberg, J. Liu, A. Saraiva Leao e P. Papapetrou. "Evaluating Local Interpretable Model-Agnostic Explanations on Clinical Machine Learning Classification Models". Em: *2020 IEEE 33rd International Symposium on Computer-Based Medical Systems (CBMS)*. IEEE, jul. de 2020. DOI: 10.1109/cbms49503.2020.00009 (ver pp. 21, 22).

[9] L. Breiman. Em: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 0885-6125. DOI: 10.1023/a:1010933404324 (ver p. 16).

[10] M. Böhle, F. Eitel, M. Weygandt e K. Ritter. "Layer-Wise Relevance Propagation for Explaining Deep Neural Network Decisions in MRI-Based Alzheimer's Disease Classification". Em: *Frontiers in Aging Neuroscience* 11 (jul. de 2019). ISSN: 1663-4365. DOI: 10.3389/fnagi.2019.00194 (ver pp. 25, 26).

[11] J. Cai, W. Hu, J. Ma, A. Si, S. Chen, L. Gong, Y. Zhang, H. Yan e F. Chen. "Explainable Machine Learning with Pairwise Interactions for Predicting Conversion from Mild Cognitive Impairment to Alzheimer's Disease Utilizing Multi-Modalities Data". Em: *Brain Sciences* 13.11 (out. de 2023), p. 1535. ISSN: 2076-3425. DOI: 10.3390/brainsci13111535 (ver p. 27).

[12] D. V. Carvalho, E. M. Pereira e J. S. Cardoso. "Machine Learning Interpretability: A Survey on Methods and Metrics". Em: *Electronics* 8.8 (jul. de 2019), p. 832. ISSN: 2079-9292. DOI: 10.3390/electronics8080832 (ver pp. 9, 15, 17, 18).

[13] N. D. Cilia, G. De Gregorio, C. De Stefano, F. Fontanella, A. Marcelli e A. Parziale. "Diagnosing Alzheimer's disease from on-line handwriting: A novel dataset and performance benchmarking". Em: *Engineering Applications of Artificial Intelligence* 111 (mai. de 2022), p. 104822. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2022.104822 (ver pp. 32, 71, 77).

[14] A. Consiglio, G. Casalino, G. Castellano, G. Grillo, E. Perlino, G. Vessio e F. Licciulli. "Explaining Ovarian Cancer Gene Expression Profiles with Fuzzy Rules and Genetic Algorithms". Em: *Electronics* 10.4 (fev. de 2021), p. 375. ISSN: 2079-9292. DOI: 10.3390/electronics10040375 (ver pp. 27–29).

[15] C. Cortes e V. Vapnik. "Support-vector networks". Em: *Machine Learning* 20.3 (set. de 1995), pp. 273–297. ISSN: 1573-0565. DOI: 10.1007/bf00994018 (ver p. 21).

[16] D. R. Cox. "The Regression Analysis of Binary Sequences". Em: *Journal of the Royal Statistical Society Series B: Statistical Methodology* 20.2 (jul. de 1958), pp. 215–232. ISSN: 1467-9868. DOI: 10.1111/j.2517-6161.1958.tb00292.x (ver p. 40).

[17] F. Doshi-Velez e B. Kim. *Towards A Rigorous Science of Interpretable Machine Learning*. 2017. DOI: 10.48550/ARXIV.1702.08608 (ver p. 18).

[18] C. Fan, F. Xiao, C. Yan, C. Liu, Z. Li e J. Wang. "A novel methodology to explain and evaluate data-driven building energy performance models based on interpretable machine learning". Em: *Applied Energy* 235 (fev. de 2019), pp. 1551–1560. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2018.11.081 (ver p. 22).

[19] A. J. Ferreira e M. A. T. Figueiredo. "Union k-Fold Feature Selection on Microarray Data". Em: *Proceedings of the 12th International Conference on Data Science, Technology and Applications, DATA 2023, Rome, Italy, July 11-13, 2023*. Ed. por O. Gusikhin, S. Hammoudi e A. Cuzzocrea. SCITEPRESS, 2023, pp. 540–547. DOI: 10.5220/0012135800003541 (ver p. 21).

[20] R. C. Fong e A. Vedaldi. "Interpretable Explanations of Black Boxes by Meaningful Perturbation". Em: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. IEEE Computer Society, 2017, pp. 3449–3457. DOI: 10.1109/ICCV.2017.371 (ver p. 18).

[21] J. Forjan. "Supervised Machine Learning, Unsupervised Machine Learning, and Deep Learning". Em: *AnalystPrep* (2021) (ver pp. 7, 8).

[22] J. H. Friedman. "Greedy function approximation: A gradient boosting machine." Em: *The Annals of Statistics* 29.5 (out. de 2001). ISSN: 0090-5364. DOI: 10.1214/aos/1013203451 (ver p. 17).

[23] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. A. Specter e L. Kagal. "Explaining Explanations: An Overview of Interpretability of Machine Learning". Em: *5th IEEE International Conference on Data Science and Advanced Analytics, DSAA 2018, Turin, Italy, October 1-3, 2018*. Ed. por F. Bonchi, F. J. Provost, T. Eliassi-Rad, W. Wang, C. Cattuto e R. Ghani. IEEE, 2018, pp. 80–89. DOI: 10.1109/DSAA.2018.00018 (ver pp. 9, 18).

[24] A. Goldstein, A. Kapelner, J. Bleich e E. Pitkin. *Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation*. 2013. DOI: 10.48550/ARXIV.1309.6392 (ver p. 18).

[25] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti e D. Pedreschi. "A Survey of Methods for Explaining Black Box Models". Em: *ACM Comput. Surv.* 51.5 (2019), 93:1–93:42. DOI: 10.1145/3236009 (ver p. 9).

[26] E. Guldogan, F. H. Yagin, A. Pinar, C. Colak, S. Kadry e J. Kim. "A proposed tree-based explainable artificial intelligence approach for the prediction of angina pectoris". Em: *Scientific Reports* 13.1 (dez. de 2023). ISSN: 2045-2322. DOI: 10.1038/s41598-023-49673-2 (ver p. 26).

[27] I. Guyon e A. Elisseeff. "An introduction to variable and feature selection". Em: (mar. de 2003), 1157–1182 (ver p. 21).

[28] H. Hakkoum, A. Idri e I. Abnane. "Global and local interpretability techniques of supervised machine learning black box models for numerical medical data". Em: *Engineering Applications of Artificial Intelligence* 131 (mai. de 2024), p. 107829. ISSN: 0952-1976. DOI: 10.1016/j.engappai.2023.107829 (ver p. 22).

[29] L. N. Koenig, G. S. Day, A. Salter, S. Keefe, L. M. Marple, J. Long, P. LaMontagne, P. Massoumzadeh, B. J. Snider, M. Kanthamneni, C. A. Raji, N. Ghoshal, B. A. Gordon, M. Miller-Thomas, J. C. Morris, J. S. Shimony e T. L. Benzinger. "Select Atrophied Regions in Alzheimer disease (SARA): An improved volumetric model for identifying Alzheimer disease dementia". Em: *NeuroImage: Clinical* 26 (2020), p. 102248. ISSN: 2213-1582. DOI: 10.1016/j.nicl.2020.102248 (ver p. 32).

[30] S. Kumar e S. Shastri. *Alzheimer MRI Preprocessed Dataset*. 2022. DOI: 10.34740/KAGGLE/DSV/3364939. URL: https://www.kaggle.com/dsv/3364939 (ver p. 32).

[31] Z. C. Lipton. "The mythos of model interpretability". Em: (set. de 2018), 36–43. DOI: 10.1145/3233231. URL: https://doi.org/10.1145/3233231 (ver pp. 9, 11, 17).

[32] Y. Lou, R. Caruana, J. Gehrke e G. Hooker. "Accurate intelligible models with pairwise interactions". Em: *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. KDD '13. Chicago, Illinois, USA: Association for Computing Machinery, ago. de 2013, 623–631. ISBN: 9781450321747. DOI: 10.1145/2487575.2487579. URL: https://doi.org/10.1145/2487575.2487579 (ver p. 25).

[33] S. M. Lundberg e S. Lee. "A Unified Approach to Interpreting Model Predictions". Em: *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*. Ed. por I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan e R. Garnett. 2017, pp. 4765–4774. URL: https://proceedings.neurips.cc/paper/2017/hash/8a20a8621978632d76c43dfd28b67767-Abstract.html (ver pp. 16, 18, 23).

[34] R. S. Michalski. "A Theory and Methodology of Inductive Learning". Em: *Artif. Intell.* 20.2 (1983), pp. 111–161. DOI: 10.1016/0004-3702(83)90016-4 (ver p. 10).

[35] T. Miller. "Explanation in artificial intelligence: Insights from the social sciences". Em: *Artif. Intell.* 267 (2019), pp. 1–38. DOI: 10.1016/J.ARTINT.2018.07.007 (ver p. 11).

[36] C. Molnar. *Interpretable machine learning. A guide for making black box models explainable*. Second edition. Literaturverzeichnis: Seiten 309-318. Munich, Germany: Christoph Molnar, 2022. 1318 pp. (ver pp. 9, 18, 46).

[37] D. Mwiti. "10 Real-Life Applications of Reinforcement Learning". Em: *neptune.ai* (2023) (ver p. 7).

[38] Y. Nohara, K. Matsumoto, H. Soejima e N. Nakashima. "Explanation of machine learning models using shapley additive explanation and application for real data in hospital". Em: *Computer Methods and Programs in Biomedicine* 214 (fev. de 2022), p. 106584. ISSN: 0169-2607. DOI: 10.1016/j.cmpb.2021.106584 (ver p. 23).

[39] *Open Access Series of Imaging Studies (OASIS)*. https://sites.wustl.edu/oasisbrains/. Accessed: 2024-06-03 (ver p. 32).

[40] M. T. Ribeiro, S. Singh e C. Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". Em: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '16. San Francisco, California, USA: Association for Computing Machinery, ago. de 2016, 1135–1144. ISBN: 9781450342322. DOI: 10.1145/2939672.2939778. URL: https://doi.org/10.1145/2939672.2939778 (ver pp. 15, 16, 18, 19, 21).

[41] M. T. Ribeiro, S. Singh e C. Guestrin. "Anchors: High-Precision Model-Agnostic Explanations". Em: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*. Ed. por S. A. McIlraith e K. Q. Weinberger. AAAI Press, 2018, pp. 1527–1535. DOI: 10.1609/AAAI.V32I1.11491 (ver p. 13).

[42] R. Roscher, B. Bohn, M. F. Duarte e J. Garcke. "Explainable Machine Learning for Scientific Insights and Discoveries". Em: *IEEE Access* 8 (2020), pp. 42200–42216. DOI: 10.1109/ACCESS.2020.2976199 (ver p. 11).

[43] S. Russell. *Artificial intelligence. A modern approach*. Ed. por P. Norvig. Third edition. Description based on print version record. Boston: Pearson, 2016. 11151 pp. ISBN: 9781292153971 (ver p. 1).

[44] W. Saeed e C. W. Omlin. "Explainable AI (XAI): A systematic meta-survey of current challenges and future opportunities". Em: *Knowl. Based Syst.* 263 (2023), p. 110273. DOI: 10.1016/J.KNOSYS.2023.110273 (ver pp. 9, 17, 18).

[45] A. L. Samuel. "Some studies in machine learning using the game of checkers". Em: *IBM Journal of Research and Development* 44.1.2 (jan. de 2000), pp. 206–226. ISSN: 0018-8646. DOI: 10.1147/rd.441.0206 (ver p. 6).

[46] G. Schwalbe e B. Finzel. "XAI Method Properties: A (Meta-)study". Em: *CoRR* abs/2105.07190 (2021). DOI: 10.48550/arxiv.2105.07190. arXiv: 2105.07190. URL: https://arxiv.org/abs/2105.07190 (ver pp. 9, 10, 12, 15, 17, 18).

[47] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh e D. Batra. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". Em: *Int. J. Comput. Vis.* 128.2 (2020), pp. 336–359. DOI: 10.1007/S11263-019-01228-7 (ver pp. 15, 23, 24).

[48] *The National Institute on Aging Genetics of Alzheimer's Disease Data Storage Site (NIAGADS)*. https://www.niagads.org/home. Accessed: 2024-06-03 (ver p. 32).

[49] E. Tjoa e C. Guan. "A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI". Em: *IEEE Trans. Neural Networks Learn. Syst.* 32.11 (2021), pp. 4793–4813. DOI: 10.1109/TNNLS.2020.3027314 (ver p. 9).

[50] A. M. TURING. "I.—COMPUTING MACHINERY AND INTELLIGENCE". Em: *Mind* LIX.236 (out. de 1950), pp. 433–460. ISSN: 0026-4423. DOI: 10.1093/mind/lix.236.433 (ver p. 5).

[51] G. Vilone e L. Longo. "Explainable Artificial Intelligence: a Systematic Review". Em: *CoRR* abs/2006.00093 (2020). DOI: 10.48550/arxiv.2006.00093. arXiv: 2006.00093. URL: https://arxiv.org/abs/2006.00093 (ver pp. 9–11, 13, 15, 17, 18).

[52] C. van Zyl, X. Ye e R. Naidoo. “Harnessing eXplainable artificial intelligence for feature selection in time series energy forecasting: A comparative analysis of Grad-CAM and SHAP”. Em: *Applied Energy* 353 (jan. de 2024), p. 122079. ISSN: 0306-2619. DOI: 10.1016/j.apenergy.2023.122079 (ver p. 24).