UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE ESTATÍSTICA E INVESTIGAÇÃO OPERACIONAL

Modelo de Previsão do Contacto Decisor no Contexto Empresarial

Margarida Pereira Norberto Dias

Mestrado em Estatística e Investigação Operacional
Especialização em Estatística

Trabalho de Projeto orientado por:
Prof. Doutor Filipe Roberto de Jesus Ramos
Susana Dias de Almeida Caçador

2023

Agradecimentos

Iniciando pela parte académica, ao meu orientador, Professor Filipe Ramos, agradeço todas as sugestões construtivas, a disponibilidade, apoio e incentivo que demonstrou ao longo destes meses. Gostava ainda de salientar a forma entusiasta com que me propôs cada desafio, que contribuiu muito para o meu desenvolvimento pessoal e académico. À Professora Teresa Alpuim e ao Professor João Gomes, afirmo que não será esquecida a sua disponibilidade e dedicação para com os alunos.

Relativamente, à equipa de Advanced Analytics que integrei durante o estágio curricular numa empresa de Telecomunicações Portuguesa, gostava de expressar a minha mais sincera gratidão pela forma com que fui recebida e integrada por todos os membros da equipa, pela partilha ativa de conhecimento e pelo apoio de todos durante este processo. Expresso especial agradecimento à minha orientadora externa, Susana Almeida, por me inserir de forma ativa nos processos de negócio, por ao longo destes meses me ter proposto desafios que contribuíram bastante para a minha evolução a nível pessoal e profissional e ainda, todo o apoio e motivação demonstrados durante esta etapa. À *Data Scientist Senior*, Catarina Freitas, que durante esta fase foi uma mentora, agradeço com muito carinho a partilha de conhecimento técnico fundamental para o trabalho que desenvolvi e por me ter ajudado a desenvolver o meu sentido de espírito crítico. Ao *Data Engineer* Sérgio Afonso, gostava de agradecer a bela equipa que formámos juntos, além de toda a ajuda e apoio demonstrados durante o projeto. Por último, aos meus colegas de estágio Rita Dias e Luís Santos, agradeço com muito carinho todo o apoio diário, a motivação, o companheirismo, a entreajuda e a bonita amizade que criámos durante os últimos meses, foram sem dúvida fundamentais para mim nesta jornada.

Na esfera pessoal, deixo um especial agradecimento aos meus pais por todas as oportunidades que me proporcionaram ao longo destes anos e por serem aqueles que mais acreditam em mim. Mãe, agradeço-te por seres a minha melhor amiga e por estares sempre comigo nos momentos complicados. À minha irmã Beatriz e ao meu cunhado Roman, expresso a minha gratidão por toda a força, por me acolherem sempre de braços abertos e pela minha sobrinha Amélia que foi a minha maior fonte de alegria durante este processo. À minha avó Eluzinda afirmo que recordarei sempre com muito carinho a sua preocupação e cuidado para comigo, em especial, durante esta etapa.

Expresso ainda um sentido agradecimento aos meus amigos por todo o apoio e pelos momentos de lazer que me ajudaram a espairecer. Em especial, aos meu colegas de casa, Bruna, Xico e Ricardo, que se tornaram na minha família em Lisboa e estiveram comigo nos melhores e nos piores momentos dos últimos anos. Finalmente, um agradecimento especial ao meu namorado Francisco por acreditar sempre em mim, por me incentivar durante toda esta etapa e por ser o meu melhor amigo.

Resumo

No setor B2B de uma empresa de telecomunicações, cada empresa/cliente tem associados vários contactos à sua carteira. Deste modo, a comunicação com um cliente específico traduz-se num grande desafio, dado que se torna difícil para a operadora identificar qual o número mais eficaz para o contactar. Note-se que o contacto com o cliente é um fator crucial para o sucesso de qualquer empresa, no que diz respeito ao aumento a satisfação do mesmo, prolongamento dos períodos de fidelização e à identificação de oportunidades comerciais.

Posto isto, o objetivo do presente projeto de trabalho consistiu em desenvolver um modelo preditivo, que estimasse a probabilidade de um determinado contacto ser o contacto decisor de um cliente. Por outras palavras, pretende-se que o modelo identifique o contacto da pessoa que decide a recusa ou aceitação de uma proposta, em nome de uma empresa.

Para esse efeito, foi criada uma variável resposta binária, sendo que, foram testadas quatro formulações distintas da mesma, a partir da informação dos resultados de negócio. Este projeto teve por base metodologias de manipulação e modelação de dados, nas áreas de Estatística e *Machine Learning*, sendo implementado na linguagem de programação Python. Neste sentido, para cada formulação da variável resposta, efetuou-se uma comparação entre dois tipos de modelos, o *Random Forest* e o *Gradient Boosting*, ambos com otimização Bayesiana de hiper-parâmetros.

Posteriormente, foi selecionado o modelo que apresentou melhores resultados e procedeu-se à validação dos respetivos *outputs*, em campanhas de *telemarketing*. Desta forma, verificou-se que que o modelo desenvolvido permitiu dar resposta ao desafio de identificar o contacto decisor, tendo-se revelado ser uma mais-valia para a empresa de telecomunicações em questão. Por conseguinte, o modelo garante ganhos de eficiência e taxa de decisão superior, em comparação com a metodologia anteriormente utilizada para contacto de clientes nos processos de negócio.

Palavras-chave: Telecomunicações, *Machine Learning*, *Random Forest*, *Gradient Boosting*, Otimização Bayesiana

Abstract

In the B2B sector of a telecommunications company, each enterprise/client has several contacts associated with its portfolio. Consequently, it arises the challenge of identifying the decision-maker contact, since it becomes difficult for the operator to identify the best number to contact. Noteworthy the contact with the customer is an important factor to increase customer satisfaction, extending loyalty periods and identifying business opportunities.

In this context, the objective of this project was to develop a predictive model that estimates the probability of a specific contact being the decision-maker for a customer. In other words, the model aims to identify the contact person, who decides the refusal or acceptance of a proposal, on behalf of a company.

For this purpose, a binary response variable was created, and four different formulations of it were tested, using business outcome data. The project relied on data manipulation and modeling methodologies in the areas of Statistics and Machine Learning, implemented in the Python programming language. In this sense, for each formulation of the response variable, a comparison was made between two types of models, Random Forest and Gradient Boosting, both with Bayesian hyperparameter optimization.

Subsequently, the model that yielded the best results was selected and its outputs were validated through telemarketing campaigns. It was found that the developed model successfully addressed the challenge of identifying the decision-making contact and proved to be an asset for the telecommunications company. Indeed, the model ensures efficiency gains and a higher decision rate compared to the previously used methodology for customer contact in business processes.

Keywords: Telecommunications, Machine Learning, Random Forest, Gradient Boosting, Bayesian Optimization

Índice

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

AA	*Advanced Analytics*
B2B	*Business to Business*
B2C	*Business to Client*
ML	*Machine Learning*
PME	Pequenas e Médias Empresas
GUC	Gestor único de Contacto
MIRA	*Maximal importance of the random attributes*
CART	*Classification and Regression Tree*
VP	Verdadeiros Positivos
FN	Falsos Negativos
FP	Falsos Positivos
VN	Verdeiros Negativos
ROC	*Receiver Operating Charateristic*
AUROC	*Area Under the ROC*
TPE	*Tree Parzen Estimator*
EI	*Expected Improvement*
SHAP	*Shapley Additive Explanations*
SAC	Serviço de Apoio ao Cliente
KPI	*Key Performance Indicators*

Capítulo 1 – Introdução

O presente trabalho foi elaborado no âmbito do projeto final de mestrado de Estatística e Investigação Operacional, através de um estágio curricular que ocorreu em contexto empresarial, na área de *Advanced Analytics* (AA) do segmento empresarial, de uma empresa de Telecomunicações Portuguesa[1].

1.1. Contextualização e Motivação

Ao longo das últimas décadas, temos presenciado uma evolução tecnológica notória e acelerada. Como consequência, o setor das Telecomunicações, a nível mundial, foi obrigado a acompanhar este progresso de forma a estar alinhado com esta nova realidade. Em Portugal, o mercado das Telecomunicações acompanhou de igual forma esta evolução, pelo que, também, nos últimos anos, tem crescido de forma exponencial mostrando-se ser, atualmente, bastante moderno e flexível no que concerne à enorme variedade de ofertas e pacotes de serviços que disponibiliza.

Segundo números divulgados pela ANACOM [1], relativamente ao 3º Trimestre de 2022, é possível verificar que, em Portugal, existem aproximadamente 4,5 milhões de subscritores de pacotes de serviços que originaram uma receita de cerca de 1393 milhões de euros, entre janeiro e setembro de 2022. É de salientar que 65% das receitas são relativas a pacotes 4P (televisão, internet fixa, telefone fixo e telemóvel) e 5P (televisão, internet fixa, telefone fixo e telemóvel e internet móvel), o que demonstra que os portugueses, na sua maioria, preferem optar por pacotes com maior número de serviços incluídos. Ainda, relativamente aos dados deste trimestre, verificou-se que existem 4 milhões de clientes com telefone em local fixo e o tráfego médio mensal deste tipo de serviço é de 48 minutos. No que diz respeito aos serviços móveis, existem 13,4 milhões de cartões ativos, com um tráfego médio mensal de 222 minutos. Em relação a serviços de distribuição de sinais de televisão por subscrição, verificaram-se 4,4 milhões de assinantes de televisão, sendo que, destes, cerca de 4 milhões são clientes residenciais e 485 mil são clientes não residenciais. Nos serviços de internet em local fixo, verificaram-se cerca de 4,4 milhões de acessos e um respetivo tráfego mensal de 252 GB por acesso.

Através do relatório da ANACOM [1], foi ainda possível verificar que, em Portugal, as principais empresas do setor das Telecomunicações são a MEO, a NOS, a NOWO e a Vodafone. Estas concorrem entre si na oferta de uma série de pacotes que agregam todos os serviços anteriormente referidos.

No setor das Telecomunicações, tal como em todos os setores que envolvam vendas ao público, pretende-se fazer uma gestão equilibrada do tempo e do trabalho gasto em produção e, posteriormente, da quantia investida na divulgação de determinado produto. Sendo assim, é fundamental para este tipo de negócio, definir qual o público-alvo a que se destina determinada oferta, com a finalidade de maximizar os seus lucros. A empresa onde foi realizado este estágio curricular, está organizada precisamente nesta lógica, pelo que é constituída por dois grandes segmentos de negócio, *Business to Client* (B2C) e *Business to Business* (B2B). O segmento B2C centra-se nas vendas residenciais (cliente individual), sendo por isso o seu foco o consumidor final. Por outro lado, o segmento B2B ocupa-se dos negócios não residenciais efetuados entre empresas.

[1] Por questões de confidencialidade o nome da empresa não será revelado.

No segmento B2B de uma empresa de Telecomunicações, por norma uma empresa (cliente) tem mais do que um contacto associado à sua carteira, podendo mesmo atingir valores na casa das centenas, de acordo com a sua dimensão e respetivo número de colaboradores. Atualmente, esta situação traduz-se num grande desafio no que toca à comunicação com um cliente específico (contacto da carteira), uma vez que se torna difícil para a operadora identificar qual o melhor número para o contactar. É de salientar que o contacto com o cliente é um fator decisivo e de enorme relevância para garantir o sucesso de uma empresa de Telecomunicações, precisamente no que concerne ao aumento da sua satisfação, ao prolongamento dos períodos de fidelização, à identificação de problemas e preocupações inerentes ao mesmo e/ou ainda oportunidades comerciais que permitam direcionar campanhas comerciais específicas.

Relativamente à empresa de Telecomunicações em questão, sabe-se que no segmento empresarial cerca de 96% dos clientes têm identificado um contacto principal e, em média, um cliente tem 5 contactos associados. De forma a ajudar os sistemas de apoio à decisão e a otimizar a comunicação entre esta operadora e o cliente, surgiu na área de AA um projeto denominado de *GUC* Atende. Neste sentido, foi desenvolvido um modelo preditivo com uma variável resposta binária, através algoritmos de Inteligência Artificial (em específico de *Machine Learning (*ML)), que têm como finalidade identificar qual(is) é(são) o(s) contacto(s) do cliente (empresa) com maior probabilidade de atender as chamadas efetuadas com o intuito de dar a conhecer campanhas comerciais.

O painel de variáveis utilizado para a construção do modelo analítico referido é constituído por informação relativa a três perfis de variáveis, sendo o primeiro a caraterização de Pequenas e Médias Empresas (PME), onde existem variáveis relativas ao número de serviços que um cliente tem, a quantia paga mensalmente, o total de dias que restam para terminar a sua fidelização, o número de funcionários da empresa, etc. O perfil de Tráfego que resulta de um histórico dos 2 meses anteriores, onde consta a informação relativa ao tráfego efetuado para todos os contactos e para contactos pertencentes ao mesmo NIF, bem como tráfego recebido de todos os contactos e por contactos pertencentes ao mesmo NIF. Por último, o perfil de Telefonia que resulta do histórico dos últimos 6 meses e as variáveis têm em conta o resultado das chamadas anteriores efetuadas para um determinado contacto e para as quais exista histórico. A variável resposta deste modelo foi construída a partir da informação constante neste último perfil, isto é, toma o valor 1- "Se o contacto atendeu a última chamada efetuada" e 0- "Caso contrário".

Após serem obtidos os resultados do modelo, ou, por outras palavras, após ter sido efetuada a previsão da probabilidade de cada contacto atender uma chamada, foi atribuído um *rank.* Para cada cliente, a primeira posição do *rank* corresponde ao contacto com maior probabilidade de atender chamadas, analogamente, a última posição corresponde ao contacto que tem menor probabilidade de atendimento.

Foi preparado um lote de campanha, que se encontrava dividido em três segmentos, com o objetivo de se medir o desempenho do modelo analítico desenvolvido nas ações do negócio. O 1º segmento consistia em avaliar o método atual, até aquele momento, onde se selecionaram três contactos do cliente, sendo que o primeiro correspondia ao contacto principal e os restantes eram selecionados de forma aleatória. No 2º segmento, pretendia-se avaliar o método analítico, deste modo, foram enviados para serem testados nas ações do negócio os contactos de um cliente que se situavam nas três melhores posições do *ranking* do GUC Atende. Por último no 3º segmento, optou-se por uma abordagem híbrida onde, dos três contactos selecionados, o primeiro correspondia ao contacto principal do cliente e os restantes provinham do *ranking* obtido no GUC Atende. Após a avaliação dos resultados da campanha efetuada, com o lote de contactos acima descrito, verificou-se que o método, exclusivamente analítico,

provou aumentar a taxa de alcance do cliente, através de menos tentativas de contacto, em média, por cliente.

Dado que o modelo desenvolvido provou trazer um incremento positivo nos resultados das campanhas comerciais, surgiu a ideia de um novo projeto, o qual foi intitulado de GUC Decide, com a finalidade de dar continuidade a este processo. Pretende-se, assim, evoluir a inteligência desenvolvida até ao momento, para a identificação do contacto de uma empresa que tem maior probabilidade de ser o "contacto do decisor". Por outras palavras, considerando o universo de contactos que atendem as chamadas em campanhas comerciais, pretende-se identificar o contacto da pessoa que decide em nome de uma empresa a recusa ou aceitação de uma proposta com ou sem benefício.

Posto isto, a grande motivação deste projeto de trabalho é, precisamente, efetuar uma investigação detalhada com o intuito de desenvolver um modelo preditivo, através de ferramentas estatísticas e algoritmos de ML que permita dar resposta ao desafio de identificar o contacto da pessoa que toma as decisões em nome de determinada empresa (cliente). É importante salientar que o presente projeto de trabalho, tal como foi referido anteriormente, foi realizado no setor empresarial (B2B), assim sendo, o domínio do estudo é centrado apenas em PME.

1.2. Objetivo

Após ter sido efetuada a contextualização e descrita a motivação deste estudo, é necessário clarificar os objetivos que se pretendem atingir com a realização deste projeto de trabalho. Sendo assim, à partida, existem conceitos essenciais relacionados com as áreas do negócio e com a própria definição de um projeto desta dimensão que tiveram de ser devidamente compreendidos e estudados, de modo a possuir um entendimento ideal do trabalho a desenvolver. Deste modo, os principais objetivos inerentes a este projeto consistiram em:

1. Desenvolver competências de aprendizagem computacional, ML e Inteligência Artificial em *Python*, que possam ser utilizados no apoio à decisão do negócio;
2. Prever a identificação do melhor contacto de um cliente, através de modelos preditivos ou sugerir procedimentos de escolha do mesmo segundo modelos prescritivos, tendo em conta as características do cliente, perfis do contacto e os resultados de campanhas anteriores;
3. Testar e avaliar os resultados dos modelos desenvolvidos, através de metodologias estatísticas e métodos utilizados atualmente pela equipa de AA do setor empresarial, tais como *SHAP*, *LIFT* e Matriz de Confusão;
4. Testar os *outputs* dos modelos em ações do negócio, para avaliar a performance e o potencial do produto desenvolvido. Isto é, averiguar se o modelo construído tem algum impacto positivo em campanhas de *telemarketing*, bem como no prolongamento da fidelização do cliente.

Para além dos pontos supramencionados, paralelamente, traçam-se alguns objetivos complementares, tais como, adquirir conhecimento relativamente às áreas de negócio das Telecomunicações, desenvolver capacidade de resolução de problemas a partir de bases de dados reais e complexas, enriquecer competências interpessoais, através da integração na equipa de AA do setor empresarial e utilização dos métodos de trabalho da empresa.

1.3. Estrutura do documento

O documento está organizado em 5 capítulos, devidamente complementados pelo capítulo da conclusão. Deste modo, após o Capítulo 1- **Introdução** o trabalho apresenta-se estruturado da seguinte forma:

- Capítulo 2 - **Revisão de Literatura** – Este capítulo é dedicado à investigação e levantamento de alguns trabalhos existentes na literatura, em especial, relativos a projetos de ML desenvolvidos no mesmo âmbito e sobre o que já é feito, atualmente, nesta área.
- Capítulo 3 - **Enquadramento Teórico** – Capítulo dedicado à investigação e explanação teórica de conceitos, métodos e procedimentos de modelação utilizados durante o desenvolvimento e implementação do projeto.
- Capítulo 4 - **Metodologia** – Neste capítulo é descrito o painel de variáveis utilizado na fase de modelação. Para além disso, são descritas de forma pormenorizada, as diferentes etapas do projeto e do processo analítico seguido. Por fim, é feita uma referência aos procedimentos em *Python*, ferramenta de implementação utilizada no decorrer do projeto, bem como as principais bibliotecas utilizadas.
- Capítulo 5 - **Resultados e Avaliação do Modelo** – Este capítulo destina-se à exposição da análise exploratória de dados, efetuada ao painel de variáveis, tais como relações entre variáveis e distribuição da variável resposta. De seguida, é apresentada a descrição das variáveis e hiper-parâmetros utilizados nos modelos, bem como a análise comparativa dos resultados, obtidos nos dois tipos de modelos testados. Apresenta-se, ainda, uma secção onde podem encontrar-se os resultados da performance do modelo nas ações de negócio.
- Capítulo 6 - **Conclusão** – Capítulo dedicado à descrição das contribuições e conclusões obtidas no desenvolvimento do projeto, bem como à identificação das limitações do mesmo e do trabalho a desenvolver futuramente.

Capítulo 2 – Estado de Arte

Graças aos avanços tecnológicos a que assistimos, nas últimas décadas, o setor das Telecomunicações sofreu um aumento acentuado no número de subscritores e na procura de serviços tecnológicos de alta qualidade. Com efeito, nota-se que os serviços oferecidos pelas operadoras são cada vez mais inteligentes e fáceis de utilizar, de modo a estarem alinhados com as diferentes necessidades dos utilizadores [2]. Como consequência da evolução tecnológica, este setor enfrenta uma grande competitividade de mercados, dado que as empresas concorrem entre si na oferta de melhores preços e serviços, dificultando a retenção de clientes.

Neste sentido, surge a necessidade de investimento em recursos de automatização e otimização dos serviços operacionais das empresas de Telecomunicações [3], nomeadamente, em tecnologias de ML, que revelam ser cruciais para a extração de perceções e tendências, a partir dos dados, relevantes para os processos de tomada de decisão. Efetivamente, estas metodologias são uma mais-valia para qualquer setor, dado que através do conhecimento das necessidades de cada cliente é possível melhorar a experiência do mesmo, através de um serviço totalmente personalizado.

Desta forma, seguem-se alguns exemplos das diferentes aplicações das tecnologias da ML, no setor das Telecomunicações, tais como: segmentação de clientes, prevenção de *churn*, previsão das melhores ofertas para um determinado cliente, identificação de fraude, previsão do valor dos utilizadores, otimização de preços e o desenvolvimento de produtos [3].

De acordo com [4], os decisores e analistas comerciais defendem que angariar novos clientes se revela mais dispendioso, do que manter os atuais. Neste sentido, foi desenvolvido um estudo para a implementação de um modelo de previsão de *churn*, no setor das Telecomunicações, que utilizou técnicas de classificação e de criação de *Clusters* para a identificação dos clientes que abandonam a subscrição de determinado serviço e os fatores subjacentes ao abandono. O modelo proposto começa por classificar os clientes através do algoritmo de classificação *Random Forest*, avaliado com recurso a métricas de avaliação como a exatidão, precisão, sensibilidade, *F1-score* e a curva de ROC. Após ser efetuada a classificação, o modelo segmenta os clientes com propensão de *churn* em grupos, com os mesmos padrões comportamentais, de modo a direcionar propostas comerciais de retenção. De salientar, que o modelo proposto apresentou resultados positivos, visto que permitiu alcançar uma taxa de *churn* mais reduzida.

Por sua vez, em [5] é feita referência ao desenvolvimento de um modelo de propensão ao *churn*, no setor em questão, onde é efetuada uma comparação do desempenho de vários algoritmos de modelação de dados de ML, tais como: a Regressão Logística, *Support Vector Machines,* o *Random Forest* e o *Gradient Boosting*. Para esse efeito, a metodologia seguida compreendeu as seguintes etapas: pré-processamento de dados, seleção de variáveis relevantes, divisão dos dados em conjuntos de treino e teste, aplicação dos algoritmos de modelação e avaliação dos mesmos. Através deste estudo, concluiu-se que o *Gradient Boosting* foi o modelo que apresentou melhores métricas de desempenho (nomeadamente, de AUROC), em contrapartida, o modelo de *Support Vector Machines* revelou ser inferior aos restantes.

Os sistemas de deteção de anomalias na rede de tráfego são cruciais para o reconhecimento de atividades suspeitas, tais como ataques invisíveis e desconhecidos [6]. Neste sentido, em [6] é apresentado um estudo de deteção de anomalias, para o qual foi utilizado um conjunto de dados que reflete a rede de tráfego atual. Por conseguinte, foi utilizado o algoritmo *LightGBM*, com recurso *a*

10-fold-cross-validation, para efetuar a classificação binária do conjunto de dados referido. O principal objetivo deste artigo consistiu na comparação do modelo proposto com estudos anteriormente realizados, a partir do mesmo conjunto de dados. Desta forma, foi possível concluir que o modelo desenvolvido garantiu melhor desempenho na deteção de anomalias na rede de tráfego, em comparação com os restantes estudos.

No que se refere ao setor das Telecomunicações, ainda existem poucas referências na literatura acerca de projetos de ML de classificação binária. No entanto, as metodologias de ML abordadas nos artigos anteriormente referidos têm bastante aplicabilidade e revelam bons resultados em vários contextos e setores de negócio.

Neste sentido, no setor da saúde, foi desenvolvido um estudo que propôs uma abordagem de classificação de eletrocardiogramas para deteção de doenças cardíacas através de metodologias de ML [7], nomeadamente os algoritmos *Random Forest* e *Gradient Boosting.* Por sua vez, no setor da banca dos EUA foram identificadas variáveis-chave para antecipar e prevenir o incumprimento bancário das instituições financeiras [8]. Para esse efeito, foi implementado um modelo capaz de prever a probabilidade de falência de 156 bancos dos EUA, com recurso ao algoritmo *Gradient Boosting*. Numa outra perspetiva, no setor aeronáutico, foi desenvolvido um estudo que incidiu nos factores que afetam o atraso de voos e propõe um modelo baseado no algoritmo *Gradient Boosting* para a previsão generalizada de atrasos de voos [9].

Capítulo 3 – Enquadramento Teórico

Este capítulo é dedicado à descrição de todos os procedimentos e metodologias (numa perspetiva mais teórica) utilizadas durante o processo de modelação dos dados. Desta forma, o objetivo incide na apresentação geral de procedimentos metodológicos em ML, desde práticas de pré-processamento de dados, à descrição alguns tipos de modelos, métricas de avaliação do seu desempenho, métodos de otimização de hiper-parâmetros e interpretação da explicabilidade dos modelos.

3.1. Tratamento e Pré-processamento de Dados

Quando trabalhamos com bases de dados reais e complexas, deparamo-nos, frequentemente, com problemas na sua organização e estrutura. Por norma, estes problemas estão relacionados com observações e/ou variáveis que apresentam valores em falta, escalas diferentes, *outliers* e ainda configurações inapropriadas. Por conseguinte, é necessário realizar uma série de procedimentos e transformações aos dados, com o intuito de os preparar e estruturar de forma correta, para a etapa da modelação.

O tratamento de valores em falta é uma das metodologias de preparação dos dados mais importante, na fase de pré-processamento. É importante salientar que a perda de informação pode originar o enviesamento dos dados em estudo, pelo que se torna importante perceber o motivo pelo qual os valores de uma variável e/ou observação se encontram em falta. A ausência de determinados valores nos dados, deve-se, muitas vezes, à inexistência da informação pretendida para uma dada amostra (conjunto de observações), ou pela incapacidade de obter determinada informação até ao momento da construção do modelo [10]. Neste sentido, existem inúmeras abordagens utilizadas para lidar com este tipo de problemas inerentes aos dados, salientando-se, de seguida, algumas das práticas habituais:

- Eliminar variáveis – nas situações em que a percentagem de valores em falta de determinada variável seja elevada, justifica-se proceder à sua eliminação;
- Eliminar observações – quando os valores omissos se concentram num conjunto de observações específico, pode-se proceder à eliminação dessa subamostra. Contudo, é necessário ser cauteloso com esta prática, uma vez que, eliminar observações, pode não ser o método mais indicado quando o painel de variáveis tem dimensão reduzida;
- Preenchimento dos valores em falta – neste caso, podemos recorrer à informação que consta nos dados para calcular o valor de algumas estatísticas relativas a uma determinada variável. Consequentemente, este valor é utilizado para preencher as observações com valores em falta dessa mesma variável. Algumas das metodologias mais comuns, para proceder ao seu preenchimento, são o cálculo da média, da mediana, da moda ou apenas uma constante pertinente.

A utilização de cada uma das técnicas acima descritas fica muitas vezes ao critério do investigador. Contudo, a escolha deve ser adaptada à realidade e às necessidades da estrutura de dados de cada projeto, visto que que não existe uma forma única e correta de as aplicar.

A codificação de variáveis categóricas representa mais uma prática fundamental na fase de pré-processamento dos dados. Quando uma base de dados apresenta variáveis categóricas [11],

deparamo-nos com um entrave, dado que a maioria dos modelos de ML aceita apenas informação numérica. Nestes casos, é necessário proceder à codificação das variáveis categóricas.

Uma possível solução, para resolver estas situações, consiste na criação de variáveis *dummy*, através de uma metodologia designada de *One-Hot Encoding* (apenas um atributo igual a 1 (quente), os restantes iguais a zero (frio)). Esta prática traduz-se na criação de colunas binárias (variável *dummy*). Por outras palavras, adiciona-se à base de dados uma nova coluna, por cada categoria de uma variável, que toma o valor 1 quando se verifica determinada condição, e o valor 0 caso contrário. No final do processo da criação das variáveis *dummy*, o número de novas colunas adicionadas à base de dados, nos algoritmos de ML, usualmente corresponde ao número de categorias, k, da variável que se pretendia transformar.

A decisão de incluir todas as variáveis *dummy*, na fase de modelação, depende do tipo de algoritmo a utilizar. Por exemplo, em modelos de regressão linear, que contam com termos de interceção, não se justifica utilizar todas as variáveis *dummy* criadas, não sendo necessário utilizar uma delas. Por outro lado, se o modelo escolhido não for sensível a esta questão, como é o exemplo das árvores de decisão, incluir todas as k colunas resultantes da codificação de variáveis, pode aumentar a interpretabilidade do mesmo.

Quando uma variável categórica conta com muitas categorias, este método de codificação de variáveis pode não ser o mais apropriado, dado que pode vir a aumentar, significativamente, a dimensão da base de dados. Consequentemente, o tempo de treino do modelo pode vir a ser mais lento e o seu desempenho pior, o que não é uma situação desejável. Assim sendo, nestes casos, é possível seguir uma outra abordagem, onde se procede à substituição de cada categoria, por números relacionados com as mesmas.

Por norma, as bases de dados reais são constituídas por um elevado número de variáveis, no entanto, verifica-se, frequentemente, que algumas delas são redundantes e, por isso, não são relevantes para o modelo [12]. Como foi referido anteriormente, existem várias desvantagens associadas ao facto de lidar com bases de dados de elevada dimensão, tais como: tornar o tempo de execução dos algoritmos mais lento, consumir muitos recursos e em determinados algoritmos de ML, pode mesmo vir a prejudicar a classificação e sua interpretabilidade.

Inicialmente, a importância de cada variável não é conhecida, pelo que o ideal é recorrer a metodologias de seleção de variáveis, a fim de selecionar um conjunto mais reduzido e relevante, que permita obter melhores resultados. Neste sentido, apresenta-se, de seguida, o Boruta que constitui um algoritmo de seleção de variáveis, tendo por base o algoritmo de classificação *Random Forest*.

No que se refere ao *Random Forest*, esta temática irá ser abordada, de forma detalhada, na secção 3.2.4. Todavia, é de salientar que o *Random Forest* é um algoritmo que defende que adicionar aleatoriedade a um determinado sistema e recolher os resultados das amostras aleatórias pode reduzir o impacto enganoso das permutações pelos diferentes objetos e correlações.

O *Random Forest* é um algoritmo onde a classificação dos dados é realizada por votação de vários classificadores fracos e imparciais, as árvores de decisão, que são desenvolvidos de forma independente, em diferentes amostras de treino. Para além disso, a sua execução é relativamente rápida, pode ser aplicado sem otimização de hiper-parâmetros e fornece uma estimativa numérica da importância das variáveis.

Desta forma, no Boruta, primeiramente, procede-se ao aumento da dimensão da base de dados, através do acréscimo de réplicas aleatórias de cada uma das variáveis. Por outros termos, para cada variável do conjunto de dados original cria-se uma variável "sombra" correspondente, cujos valores são obtidos através da permutação aleatória dos valores da variável original. Posteriormente, através do modelo *Random Forest*, realiza-se a classificação do painel de dados aumentado (com todas as variáveis originais e as suas réplicas) e calcula-se a importância das variáveis.

Deste modo, o conjunto de variáveis "sombra" é utilizado como referência para decidir quais são as variáveis realmente importantes para a classificação. A avaliação da importância de uma variável da base de dados original é efetuada através da comparação com a variável "sombra", correspondente [13]. Assim, apenas as variáveis cuja importância é maior do que a da réplica que lhe corresponde são consideradas importantes.

É essencial referir que, na secção 3.2.4 se procede à explicação de como é efetuado o cálculo da medida de importância de uma variável, no algoritmo do *Random Forest*. Contudo, de seguida, importa perceber quais são e em que consistem os diferentes passos, deste algoritmo de seleção de variáveis. Posto isto, segue-se a descrição das diferentes etapas do Boruta [12]:

1. Aumentar a base de dados, através da criação de réplicas (variáveis "sombra") de cada uma das variáveis existentes. De seguida, os valores das variáveis replicadas são permutados pelos diferentes objetos. Consequentemente, todas as correlações existentes entre as variáveis "sombra" e a variável resposta são aleatórias;
2. Realizam-se diversas execuções do *Random Forest*. De salientar que as variáveis "sombra" são baralhadas antes de cada iteração do algoritmo, pelo que a parte aleatória do sistema é sempre diferente;
3. Em cada iteração é calculada a importância de cada variável do sistema aumentado;
4. Para uma determinada iteração do algoritmo, uma variável é considerada importante, se o valor da sua importância for superior à importância máxima de todas as variáveis aleatórias;
5. Seguidamente, é realizado um teste de hipóteses[2] para todas as variáveis. A hipótese nula é que a importância de uma variável é igual à *maximal importance of the random attributes* (MIRA) (importância máxima das variáveis aleatórias). Este é um teste de hipóteses bilateral, pelo que a hipótese nula pode ser rejeitada quando a importância de uma variável é, significativamente, maior ou menor, do que MIRA.
   Para cada variável, é efetuada uma contagem do número de vezes que o valor da sua importância foi superior à MIRA (contagem do número de êxitos, por cada variável). O número de êxitos esperado, para N iterações do algoritmo, é $\mathrm{E} = 0{,}5\mathrm{N}$, com desvio-padrão $S = \sqrt{0{,}25N}$ (distribuição Binomial[3] com $p = q = 0{,}5$). Desta forma, uma variável é considerada importante (selecionada), quando o número de êxitos é, significativamente, superior ao valor esperado. Por outro lado, uma variável não é considerada relevante (rejeitada), quando o número de êxitos é, significativamente, inferior ao valor esperado.
   Neste sentido, para um determinado número de iterações do algoritmo, os limites de seleção e rejeição de variáveis são obtidos para um determinado nível de confiança. Ou seja, as áreas de seleção e rejeição de variáveis correspondem às partes extremas da distribuição Binomial (caudas

[2] Como o próprio nome diz, são testes estatísticos que permitem rejeitar ou não hipóteses testadas, com determinado grau de confiança, baseados em valores amostrais.
[3] Distribuição de probabilidade discreta, que descreve o número de sucessos, que ocorrem em n repetições independentes de uma experiência aleatória, com probabilidade de sucesso p.

da distribuição), tal como se pode verificar na Figura 3.1 (no exemplo apresentado o algoritmo foi executado para 20 iterações e as caudas representam 5% da distribuição);

6. As variáveis que, na etapa anterior, não foram consideradas importantes são removidas do sistema, juntamente com as variáveis aleatórias replicadas que lhes correspondem;
7. Este processo é realizado até ser obtida uma conclusão final, relativamente à importância de todas as variáveis.
   Em alternativa, o algoritmo pode ser executado durante um determinado número de iterações, previamente definido. Neste caso, podem sobrar algumas variáveis sobre as quais o algoritmo revela alguma incerteza e, por isso, não foram nem rejeitadas nem selecionadas. Sendo assim, estas variáveis são consideradas indeterminadas e devem também ser selecionadas.

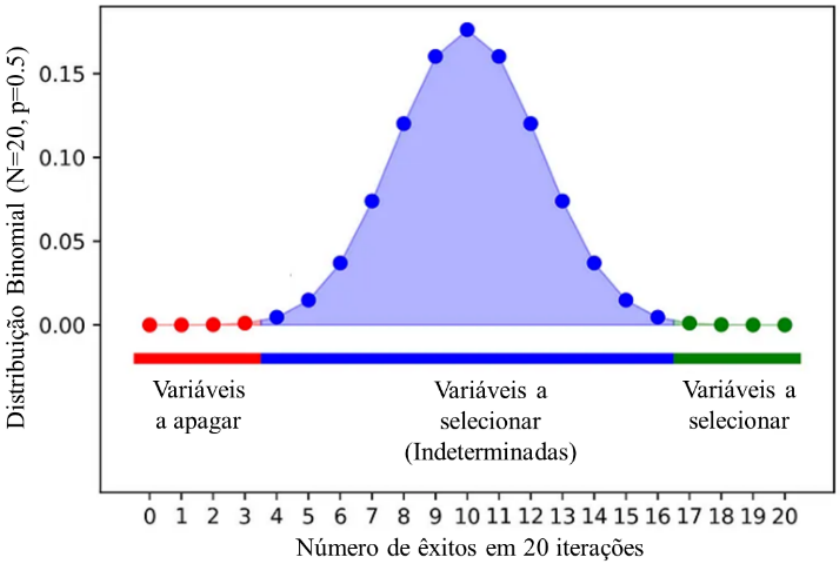

Figura 3.1 - Exemplo ilustrativo de seleção de variáveis com o *Boruta*.

Fonte: Figura adaptada de [14].

3.2. Machine Learning

Machine Learning traduz-se na ciência de programar computadores através da implementação de algoritmos e sistemas que são capazes de melhorar a sua performance, através da experiência que adquirem a partir dos dados que lhes são fornecidos. As metodologias de ML constituem um ramo da Inteligência Artificial e surgiram, precisamente, da necessidade crescente de processar grandes quantidades de dados e obter conhecimento a partir dos mesmos. Desta forma, através da utilização das fontes de dados apropriadas, é possível implementar/construir os modelos certos que proporcionam a realização das tarefas pretendidas [11].

No nosso quotidiano lidamos, frequentemente, com problemas para os quais as soluções existentes requerem bastantes ajustes manuais ou longas listas de regras. Por outro lado, existem problemas que, simplesmente, se revelam muito complexos para a aplicação das metodologias tradicionais. Nestes casos, os modelos de ML revelam ser uma abordagem bastante vantajosa e eficaz, porquanto, permitem simplificar código, melhorar o comportamento dos modelos existentes, automatizar tarefas que apresentam uma dificuldade elevada para a sua execução manual e, por fim, solucionar problemas para os quais ainda não se conhece uma solução.

Para além destes benefícios, na presença de problemas complexos ou estruturas de dados de elevada dimensão, os algoritmos de ML conduzem a uma melhor compreensão do próprio problema em estudo, uma vez que, a partir destes, o ser humano consegue adquirir perceções relevantes acerca dos dados. De facto, através da execução de um algoritmo de ML é possível encontrar determinadas tendências nos dados, identificar as variáveis mais importantes para explicar o caso em estudo, detetar correlações inesperadas entre variáveis e descobrir fórmulas matemáticas que expliquem as relações entre os dados.

Os algoritmos de ML são ainda caraterizados pela sua facilidade de adaptação a novas bases de dados, o que é uma grande vantagem, relativamente aos algoritmos tradicionais. Por fim, é de destacar que ML se baseia em metodologias de Matemática e Estatística que, aliadas à aprendizagem computacional, permitem solucionar problemas de elevada complexidade.

Na presente secção são descritos vários procedimentos metodológicos de ML, sendo possível consultar nos livros [10], [11] e [15] informação mais detalhada acerca de cada um deles.

3.2.1. Tipos de Aprendizagem

Os sistemas de ML podem ser classificados de acordo com a quantidade e tipo de supervisão que recebem durante o treino dos algoritmos [11]. Assim, distinguem-se, de seguida, as quatro principais categorias: Aprendizagem Supervisionada, Aprendizagem Não Supervisionada, Aprendizagem Semi-Supervisionada e Aprendizagem por Reforço[4].

Na Aprendizagem Supervisionada, para uma determinada amostra de dados utilizada para o treino do algoritmo, é conhecido o verdadeiro valor da variável resposta, designado de rótulo. Por outras palavras, significa que este tipo de aprendizagem requer dados de treino onde os valores da variável dependente são, necessariamente, conhecidos. A título de exemplo, salientam-se algoritmos de classificação e regressão. Em ambas as abordagens, a partir do treino de dados históricos, dos quais se conhece o verdadeiro valor da variável em estudo, é possível obter as respetivas previsões para novos dados, dos quais não se conhece o rótulo. A este processo de previsão da variável em estudo para novos conjuntos de dados, dá-se o nome de inferência.

A principal diferença entre algoritmos de classificação e regressão reside no tipo de problemas a que se direcionam. A aplicação de algoritmos de classificação ocorre quando a variável dependente é categórica, sendo assim, pretendem-se obter as previsões da classe a que pertencem determinadas observações. Por outro lado, em algoritmos de regressão, a variável resposta é numérica, pelo que se pretendem obter as previsões de um valor numérico. É de assinalar que em classificação também é possível efetuar a previsão de um valor numérico, porém, esse valor está sempre associado a uma determinada categoria.

Em Aprendizagem Não Supervisionada, os dados utilizados para o treino dos algoritmos não possuem rótulos, isto é, não é conhecido o verdadeiro valor a variável resposta. Assim, o principal objetivo deste tipo de algoritmos é inferir acerca das relações existentes entre um determinado conjunto de observações. Um algoritmo, frequentemente associado a este tipo de aprendizagem, é a criação de *Clusters*, que pretende avaliar a similaridade das observações. Desta forma, neste algoritmo, colocam-se as observações semelhantes num mesmo grupo (*cluster*) e observações dissemelhantes em grupos

[4] Alguns autores consideram apenas três abordagens para os tipos de aprendizagem [16] .

diferentes. A título ilustrativo, veja-se a Figura 3.2, a qual esquematiza os resultados decorrentes dos dois paradigmas, distinguindo-se a Aprendizagem Supervisionada da Aprendizagem Não Supervisionada.

No que toca, a Aprendizagem Semi-Supervisionada, os dados utilizados para o treino dos algoritmos, estão parcialmente rotulados, ou seja, para determinadas observações é conhecida a verdadeira variável resposta, enquanto para outras não é conhecida. Neste sentido, a maior parte dos algoritmos de Aprendizagem Semi-Supervisionada, consistem na combinação de algoritmos de Aprendizagem Supervisionada e Não Supervisionada.

Por fim, temos a Aprendizagem por Reforço, onde o sistema de aprendizagem é designado de agente e tem a capacidade de selecionar e executar determinada ação, com o intuito de obter recompensas. O objetivo deste tipo de sistema é aprender por si próprio qual a melhor estratégia (política) a seguir, com o objetivo de obter o máximo de recompensa. Assim, uma política define qual a ação que o agente deve executar em determinada situação.

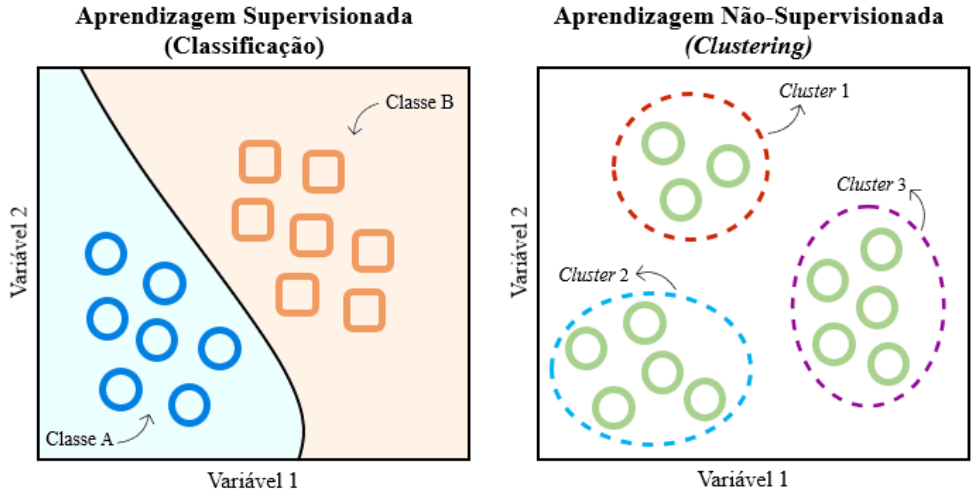

Figura 3.2 - Aprendizagem Supervisionada vs. Aprendizagem Não Supervisionada

3.2.2. Árvores de Decisão

As árvores de decisão, são algoritmos de ML versáteis e capazes de lidar com estruturas de dados bastante complexas. Este tipo de algoritmo aplica-se tanto a problemas de classificação como de regressão, contudo, o foco do estudo centra-se apenas em árvores de decisão de classificação. Neste sentido, procede-se à explicação do algoritmo das árvores de decisão, uma vez que este constitui a base dos algoritmos *Random Forest* e *Gradient Boosting*, que irão ser aprofundados nas secções 3.2.4 e 3.2.5, respetivamente.

Árvores de decisão são algoritmos que particionam, recursivamente, a população em estudo, em grupos denominados por nodos, de dimensões mais reduzidas e mais homogéneos, relativamente à variável resposta. A homogeneidade em problemas de classificação significa que os grupos resultantes da partição dos dados são mais "puros", ou seja, que após ser efetuada a divisão dos dados, cada nodo contém uma maior proporção de observações, de uma determinada classe da variável dependente [10].

Fazendo uma analogia de árvore, nas árvores de decisão existe sempre um nodo de origem para a partição dos dados, o qual se denomina de nodo raiz. Por sua vez, os segmentos que ao longo do desenvolvimento da árvore conectam cada um dos nodos, designam-se de ramos. Neste sentido, constata-se, ainda, que os nodos que resultam de uma determinada partição e que, posteriormente, dão origem a uma nova, se intitulam de nodos internos. Desta forma, os nodos internos estão sempre associados a uma determinada condição de partição dos dados e, por esse motivo, são considerados nodos de decisão. Finalmente, os nodos que não possuem ramificações são denominados de nodos folha ou nodos terminais.

Um dos algoritmos mais utilizados para treinar árvores de decisão denomina-se *Classification And Regression Tree* (CART). O conceito deste é muito simples, primeiramente, a partir do nodo-raíz é encontrado o valor de corte de uma determinada variável numérica independente, segundo o qual a divisão dos dados leva a subconjuntos mais homogéneos, relativamente às classes da variável resposta. De seguida, o algoritmo efetua a divisão dos dados em dois subconjuntos segundo essa mesma condição, originando, assim, dois nodos. Isto significa que cada observação é classificada de acordo com o valor que toma na variável escolhida como condição para a partição. Para um determinado nodo, que resulta de uma ramificação, verifica-se que a probabilidade de uma observação pertencer a uma determinada classe da variável dependente, corresponde à proporção de observações dessa classe, nesse mesmo nodo.

Após ser efetuada a divisão do conjunto dos dados, o algoritmo volta a dividir os subconjuntos resultantes da divisão anterior em dois novos subconjuntos e, assim, sucessivamente. Este processo é realizado enquanto o algoritmo encontrar uma partição que conduza a nodos mais "puros", relativamente à variável resposta. Assim, no final da classificação, as predições de cada observação correspondem à classe que tem maior probabilidade de ocorrer no nodo folha a que se aplicam.

A pureza de um nodo, em problemas de classificação, pode ser definida através da maximização da exatidão ou, de forma equivalente, minimizando o erro da classificação. No entanto, esta medida pode ser pouco robusta, uma vez que a partição dos dados é efetuada de modo a minimizar os erros da classificação, ao invés do foco ser a partição dos dados, de acordo com a classe a que pertencem. Posto isto, existem medidas alternativas para a partição dos dados, sendo uma delas o índice de *Gini*. É possível consultar outras medidas de partição dos dados, no seguinte livro [10]. Para um problema de duas classes, o índice de *Gini* (G) para um determinado nodo é definido por

$$G = p_1(1 - p_1) + p_2(1 - p_2) \tag{3.1}$$

onde p_1 e p_2 representam as probabilidades de as observações pertenceram à classe 1 e 2 da variável dependente, respetivamente. Como é considerado um problema de duas classes, $p_1 + p_2 = 1$, desta forma a equação 3.1 pode ser escrita como $2p_1p_2$. Assim, é fácil de perceber que o índice *Gini* é mínimo, quando a probabilidade de uma classe é próxima de zero, significando que um nodo é puro relativamente à outra classe.

Efetivamente, um nodo da árvore de decisão é considerado "puro" (índice de *Gini* = 0), se todas as observações às quais se aplica pertencem à mesma classe. Em contrapartida, para $p_1 = p_2$ esta medida atinge o seu máximo, indicando que um nodo não é "puro". Desta forma, na presença de uma variável independente numérica e uma variável resposta binária, para cada ponto de partição resulta uma tabela de contingência $2x2$, tal como a Tabela 3.1.

Tabela 3.1: Tabela de contingência 2x2, obtida em cada ponto de partição de uma árvore de decisão.

	Classe 1	**Classe 2**	
$Partição\ >$	n_{11}	n_{12}	n_{+1}
$Partição \leq$	n_{21}	n_{22}	n_{+2}
	n_{1+}	n_{2+}	n

Antes de ser efetuada qualquer partição nos dados em estudo, o índice de Gini é dado pela seguinte expressão:

$$G\ (antes\ da\ partição) = 2\left(\frac{n_{1+}}{n}\right)\left(\frac{n_{2+}}{n}\right) \tag{3.2}$$

Após ser efetuada uma partição dos dados, da qual resultam dois nodos, o índice de Gini para a partição de "maior que" é dado por $2\left(\frac{n_{11}}{n_{+1}}\right)\left(\frac{n_{12}}{n_{+1}}\right)$ e para a partição "menor ou igual que" por $2\left(\frac{n_{21}}{n_{+2}}\right)\left(\frac{n_{22}}{n_{+2}}\right)$. Estes valores são, então, combinados, com a proporção de observações em cada um dos nodos obtidos, $\left(\frac{n_{+1}}{n}\right)$ e $\left(\frac{n_{+2}}{n}\right)$, que representam os respetivos pesos para a partição "maior que" e "menor ou igual que ". Assim, depois de algumas simplificações, a função custo que o algoritmo CART pretende minimizar para avaliar uma partição é dada por:

$$G(depois\ da\ partição) = \tag{3.3}$$

$$= \left(\frac{n_{+1}}{n}\right) G\ (\text{Partição} >) + \left(\frac{n_{+2}}{n}\right) G(\text{Partição} \leq)$$

$$= 2\left[\left(\frac{n_{11}}{n}\right)\left(\frac{n_{12}}{n_{+1}}\right) + \left(\frac{n_{21}}{n}\right)\left(\frac{n_{22}}{n_{+2}}\right)\right]$$

As árvores de decisão são algoritmos poderosos, versáteis e que apresentam relativa facilidade de interpretação. Em contrapartida, são algoritmos suscetíveis a *overfitting*[5] *e* bastante sensíveis a pequenas variações nos dados pelo que se podem tornar pouco robustas. Com efeito, uma pequena variação nos dados pode originar uma grande mudança no resultado, ou seja, nas predições finais obtidas pelo modelo. A instabilidade das árvores de decisão pode ser limitada através das metodologias de *Model Ensembles*, tal como irá ser estudado de seguida.

3.2.3. Model Ensembles

Quando trabalhamos com problemas de elevada complexidade, pode tornar-se vantajoso efetuar uma combinação simultânea de vários algoritmos de ML para encontrar uma solução. Por norma, estas combinações de algoritmos denominam-se de *Model Ensembles*. Esta é uma metodologia de Aprendizagem Supervisionada que, atualmente, é considerada uma das mais poderosas e eficazes. Contudo, promove o aumento significativo da complexidade associada aos modelos, bem como dos recursos necessários para a sua execução [15].

[5] Ocorre quando um modelo se ajusta de forma perfeita, aos dados com os quais foi efetuado o seu treino. Desta forma, quando é aplicado a dados que desconhece, revela ter fraco desempenho.

A principal motivação, para a criação deste tipo de métodos, surge da necessidade de aliar a aprendizagem computacional à estatística. As intuições estatísticas defendem que para uma determinada experiência efetuar a média de várias medições conduz a estimativas mais estáveis e viáveis, do que quando é considerada uma única medição. Dado que, através da média das medições é possível reduzir a variância (aleatoriedade) associada a cada uma das medições independentes. Assim, a partir dos mesmos dados, a construção de um modelo que resulta da combinação de vários algoritmos permite reduzir a variância das flutuações aleatórias dos algoritmos singulares.

Deste modo, em determinados tipos de *Model Ensembles*, realiza-se o treino de um conjunto de classificadores distintos e independentes, em paralelo, e são obtidas as previsões resultantes de cada um. De seguida, dependendo do tipo de problema em estudo, as previsões obtidas são combinadas através da realização de uma votação, ou do cálculo da média ponderada.

Em problemas de classificação, agregam-se as predições resultantes de cada classificador independente e efetua-se uma votação das classes. Desta forma, a classe que angariar um maior número de votos, corresponde à previsão final do conjunto de classificadores, tal como se pode observar na Figura 3.3. Em problemas de regressão, o processo é análogo ao que foi descrito anteriormente, todavia, a previsão final é obtida através do cálculo da média das previsões obtidas em cada um dos classificadores independentes.

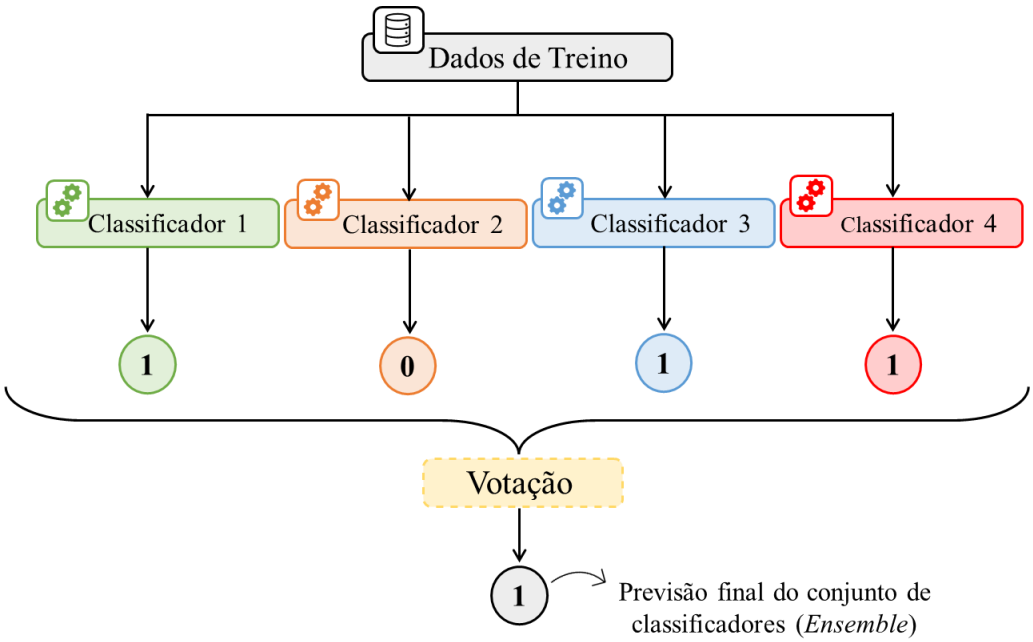

Figura 3.3- Previsão por votação de um conjunto de classificadores distintos (*Ensemble*)

É possível que num conjunto de classificadores exista um que apresente melhores resultados de exatidão do que os restantes, ainda assim, é importante referir que o conjunto de todos os classificadores consegue obter melhor desempenho. De facto, ainda que cada classificador independente seja fraco, o conjunto dos classificadores (*Ensemble*) pode ser forte, desde que haja um número suficiente de classificadores fracos e diversos para se efetuar o treino do modelo [11].

Uma forma de obter um conjunto de classificadores diversificados é através da utilização de vários algoritmos distintos para treino, tal como foi referido anteriormente. Outra abordagem, consiste na utilização do mesmo algoritmo em cada classificador, contudo, o treino é efetuado em diferentes

subconjuntos aleatórios da população de dados em estudo. Posteriormente, são obtidas as predições de cada classificador e procede-se à sua agregação por votação. A este processo dá-se o nome de *Bagging*, abreviatura de "*bootstrap aggregating"* [15].

Bagging é considerado um método simples de construção de *Model Ensembles*, embora altamente eficaz, que consiste na criação de diversos modelos, em diferentes amostras de observações aleatórias, do conjunto de dados original. As amostras são escolhidas, aleatoriamente, com reposição a partir dos dados em estudo, às quais se dá o nome de amostras de *bootstrap*. Uma vez que as amostras de *bootstrap* são escolhidas com reposição, no geral contém observações duplicadas. Por outro lado, algumas observações do conjunto de dados inicial ficam em falta, ainda que a dimensão da amostra aleatória coincida com a dimensão dos dados iniciais.

Esta situação é, precisamente, aquilo que se pretende, dado que são as diferenças entre as amostras de *bootstrap* que originam a diversidade entre cada um dos classificadores deste tipo de *Model Ensemble.* De facto, tal como é possível visualizar na Figura 3.4, as amostras de *bootstrap*, pretendem aumentar a diversidade dos subconjuntos de dados, onde é treinado cada classificador, de forma a obter classificadores pouco correlacionados entre si.

Note-se que, efetuar o treino de cada classificador independente em subconjuntos de dados aleatórios, origina um viés superior do que se os mesmos fossem treinados no conjunto de dados original. Não obstante, através da agregação por votação ou cálculo da média (classificação e regressão, respetivamente) das predições de todos os classificadores independentes, é reduzida a variância e o enviesamento. Com efeito, o objetivo é que o conjunto de classificadores (*Ensemble*), apresente um viés semelhante e uma variância menor, do que um único classificador que seja treinado no conjunto de dados original [11].

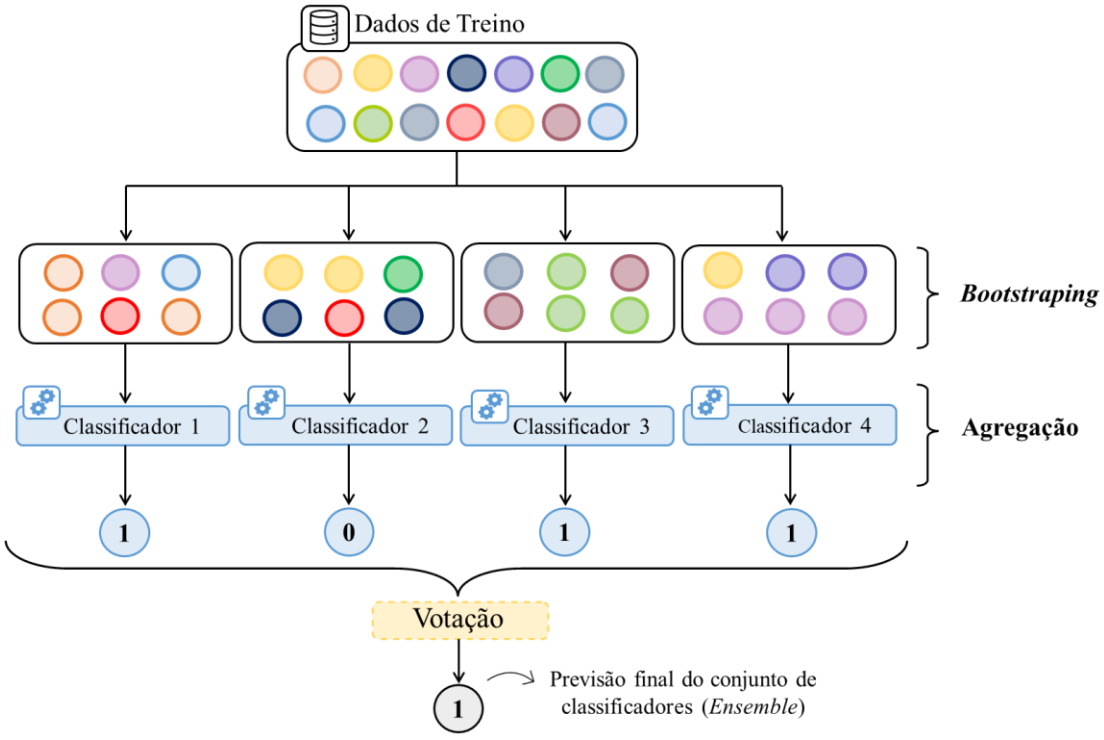

Figura 3.4 - *Bagging* ("*bootstrap aggregating*").

A construção de *Model Ensembles*, segundo a metodologia de *Bagging*, apresenta a limitação de analisar todas as variáveis do conjunto de dados original para o treino de cada classificador. Desta forma, os diferentes classificadores não são totalmente independentes, pelo que se considera que existem correlações entre si. Para além disso, uma vez que todas as variáveis são analisadas, é necessário um maior número de recursos para o treino do modelo.

3.2.4. Random Forest

O *Random Forest*, é um tipo de *Model Ensemble*, onde a classificação é realizada por votação de várias árvores de decisão, que são desenvolvidos de forma independente e em paralelo, em diferentes amostras aleatórias do conjunto de dados original. Assim, as árvores de decisão são consideradas as componentes fundamentais deste algoritmo.

Tal como foi ilustrado anteriormente, a construção de árvores de decisão, segundo a metodologia de *bootstrap aggregating*, permite obter previsões com um maior grau de confiança, do que quando é considerada apenas uma árvore. No entanto, verifica-se que as árvores de decisão resultantes deste processo não são completamente independentes entre si, dado que todas as variáveis do conjunto de dados original são analisadas em cada partição [10].

Com efeito, as árvores de decisão desenvolvidas, apresentam estruturas semelhantes, especialmente no topo das árvores, devido às correlações das variáveis independentes, com a variável resposta. Consequentemente, as árvores de decisão obtidas são bastante correlacionadas entre si.

O algoritmo *Random Forest* providencia uma forma de mitigar as correlações entre as árvores de decisão e, deste modo, reduzir a variância das previsões obtidas. Assim como o *Bagging*, o *Random Forest*, é um algoritmo onde a partir das amostras de *bootstrap* são desenvolvidas em paralelo várias árvores de decisão. Porém, para uma determinada árvore de decisão, em cada partição, é escolhida uma amostra aleatória de k variáveis a partir da totalidade de variáveis p dos dados, como candidatas a ser analisadas.

Isto significa, que para cada partição de uma determinada árvore, é analisado um conjunto de variáveis de menor dimensão, do que o conjunto de variáveis original. Uma vez que as variáveis são selecionadas de forma aleatória, no início de cada partição, os subconjuntos de variáveis analisados são distintos, pelo que variam de partição para partição [10].

No final do algoritmo, são obtidas as predições de cada árvore de decisão, que, de seguida, são agregadas por votação ou cálculo da média (problemas de classificação ou regressão, respetivamente), a fim de obter a previsão final pretendida [17]. Neste sentido, o pseudocódigo relativo ao algoritmo do *Random Forest*, para um problema de classificação, encontra-se apresentado na Tabela 3.2.

Tabela 3.2 – Algoritmo relativo ao *Random Forest*

Algoritmo 1: Random Forest

Selecionar o número de árvores de decisão a construir, $\boldsymbol{m}$

Para $i = 1$ até m **fazer**

Gerar uma amostra de *bootstrap* a partir dos dados originais

Treinar um modelo de árvore de decisão nessa amostra

Para cada partição da árvore de decisão fazer

i. Selecionar de forma aleatória um conjunto de variáveis, a partir do conjunto de variáveis original

ii. A partir do conjunto de variáveis escolhidas, selecionar uma para proceder à partição dos dados

iii. Efetuar a divisão dos dados em dois subconjuntos, segundo o critério de partição

Fim

Utilizar um critério de paragem, que determine quando a construção da árvore está completa

Fim

Saída do conjunto de árvores de decisão *(Ensemble)*

Para obter as previsões de um determinado ponto $\boldsymbol{x}$: Seja $\hat{C}_i(x)$ a previsão da classe da i-ésima árvore de decisão do *Random Forest.* Então $\hat{C}_{RF}(x) = voto\ maioritário\ \{\hat{C}_i(x)\}_1^m$

Dado que, em cada partição de uma árvore de decisão, o algoritmo seleciona, aleatoriamente, um conjunto de variáveis para analisar, as correlações entre as árvores são, necessariamente, reduzidas. Também, devido a este fator, em comparação com o *Bagging*, o *Random Forest* revela ser mais eficiente em termos computacionais. Assim, do modelo *Random Forest* resultam árvores de decisão diversificadas o que, de forma geral, conduz a modelos com melhor desempenho.

Um benefício que o *Random Forest* apresenta, consiste em fornecer uma estimativa numérica da importância de cada variável, isto é, permite quantificar o impacto de cada variável para o conjunto de classificadores (*Ensemble*). Para um problema de classificação, a medida de importância de uma variável é dada pela melhoria da pureza dos nodos das árvores de decisão, ou seja, pelo índice de Gini.

Neste sentido, tem-se que, para todos os nodos, nos quais uma determinada variável X foi considerada para a partição dos dados, existe associado um índice de Gini. Desta forma, a medida de importância da variável X consiste na média pesada dos índices de Gini de todos os nodos onde a variável foi utilizada. Mais precisamente, é uma média ponderada, onde o peso de cada nodo, corresponde à proporção de observações ao qual se aplica [11].

Esta medida de importância, é calculada para cada uma das variáveis que foram consideradas para o treino do algoritmo. De seguida, os valores de importância das variáveis são standardizados, o que significa que a soma da importância de todas as variáveis tem de ser igual a 1 [11].

3.2.5. Gradient Boosting

Boosting refere-se a um método de *Ensemble*, que combina vários modelos fracos num modelo forte. Ao contrário do *Bagging*, onde se efetua o treino de modelos independentes em paralelo, o foco das metodologias de *Boosting* é treinar modelos, sequencialmente, com o objetivo de que cada modelo tente corrigir o modelo precedente [11]. Assim, neste tipo de método, pretende-se que os erros resultantes de cada modelo diminuam à medida que o número de iterações de determinado algoritmo aumenta.

Se considerarmos, uma vez mais, a árvore de decisão como o elemento base de aprendizagem, verificamos que neste tipo de *Ensemble* se procede ao treino sequencial de árvores de decisão. Tal como foi referido nas secções anteriores, ajustar um modelo de árvore de decisão aos dados em estudo, gera, normalmente *overfitting* e, consequentemente, predições pouco robustas. Em contrapartida, através da metodologia de *Boosting*, é possível obter predições mais viáveis e confiantes, dado que a aprendizagem é feita de forma lenta e tem como objetivo, corrigir de forma progressiva, os erros de aprendizagem resultantes das árvores de decisão anteriores.

De facto, existem vários métodos de *Boosting*, porém, de seguida, destaca-se apenas um dos principais, o *Gradient Boosting*. Este é um algoritmo de ML, que se aplica tanto a problemas de regressão como de classificação, sendo que o foco nesta secção será apenas problemas de classificação.

O princípio fundamental do *Gradient Boosting* consiste em a partir de uma determinada função de perda (gradiente estocástico) e um modelo de aprendizagem fraco (árvores de decisão), encontrar um modelo aditivo que minimize essa mesma função [10]. Assim, em cada iteração do algoritmo, em vez de cada modelo ser aplicado à variável resposta dos dados em estudo, é aplicado aos erros residuais que resultaram do modelo anterior.

O algoritmo do *Gradient Boosting*, por norma, é inicializado com as predições inicias dos dados e, consequentemente, com o cálculo dos resíduos. De seguida, aplica-se uma árvore de decisão aos resíduos que foram calculados, com o intuito de minimizar a função de perda. Por outras palavras, é executada uma árvore de decisão, onde os erros residuais calculados anteriormente são utilizados como variável resposta [11].

A árvore de decisão construída é adicionada à função que se pretende ajustar e os valores dos resíduos devem, então, ser atualizados. Posteriormente, os resíduos atualizados, correspondem à nova variável resposta, pelo que se efetua o treino de uma nova árvore de decisão. A árvore que foi treinada é, novamente, adicionada à função de perda e os resíduos atualizados, tal como se pode verificar na Figura 3.5. Este processo continua durante um determinado número de iterações pré-definido.

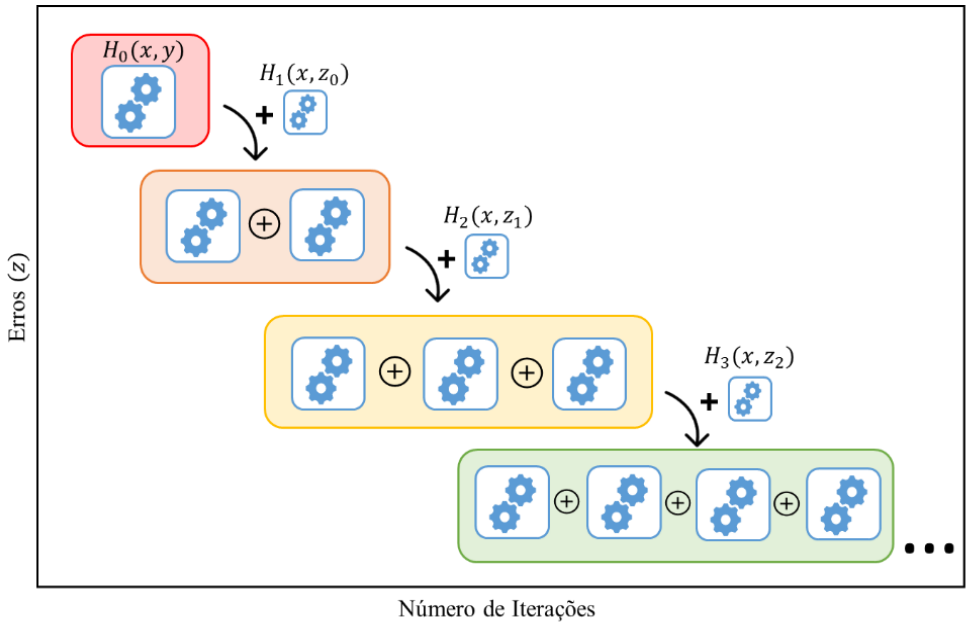

Figura 3.5 - *Gradient Boosting*, treino sequencial de classificadores, a partir dos resíduos obtidos na iteração anterior.

Neste sentido, torna-se importante perceber como são calculadas as previsões de cada observação dos dados em estudo, necessárias para inicializar o algoritmo e durante o processo. A formulação da função gradiente estocástica consiste em modelar um evento de probabilidade $\hat{p}_i$ para cada observação x_i dos dados, dado por:

$$\hat{p}_i = \frac{1}{1 + e^{[-f(x_i)]}} \tag{3.4}$$

onde $f(x_i)$ corresponde a uma previsão do modelo, no intervalo $[-\infty, +\infty]$, para uma determinada observação [10]. Neste algoritmo, as previsões iniciais dos dados são dadas através do cálculo do logaritmo das *odds* da variável resposta. Seja $\hat{p}$ a proporção de eventos de uma determinada classe da variável resposta, então as *odds* de uma observação pertencer a essa mesma classe correspondem a $\frac{\hat{p}}{1-\hat{p}}$. Assim, conclui-se que as previsões iniciais para cada observação são dadas por:

$$f(x_i)^{(0)} = \log\left(\frac{\hat{p}}{1 - \hat{p}}\right) \tag{3.5}$$

Após serem obtidas as previsões iniciais do algoritmo, procede-se á sua transformação em probabilidades $\hat{p}_i$ e são calculados os resíduos para cada uma das observações dos dados. O cálculo dos resíduos é efetuado através da diferença entre o verdadeiro valor da variável resposta y_i (valor observado) e o valor previsto $\hat{p}_i$, pelo que vem:

$$z_i = y_i - \hat{p}_i \tag{3.6}$$

De seguida, é efetuado o treino de uma árvore de decisão $H(x, z)$, onde os resíduos calculados correspondem à variável resposta. Importa, agora, perceber como são calculadas as previsões dos resíduos que resultam da árvore construída. Uma vez que as previsões iniciais foram dadas em termos de logaritmo das *odds*, as previsões que resultam da árvore de decisão têm de sofrer uma transformação, de modo a obtê-las também em termos do logaritmo das *odds*. Assim, para um determinado nodo terminal R_j, tem-se que a previsão da árvore de decisão construída resulta da expressão:

$$r_j = \frac{\sum_{x_i \in R_j}(y_i - \hat{p}_i)}{\sum_{x_i \in R_j} \hat{p}_i\,(1-\hat{p}_i)} \tag{3.7}$$

Com efeito, as previsões de cada nodo são obtidas através da divisão do somatório dos resíduos de cada uma das observações às quais esse nodo se aplica, pelo somatório do produto das probabilidades anteriormente calculadas pelo seu complementar.

Deste modo, as previsões finais pretendidas para cada observação são dadas pela soma da previsão anterior com a previsão resultante da árvore de decisão, multiplicada pelo *learning rate* $\lambda \in [0,1]$. O *learning rate* é um parâmetro responsável por dimensionar a contribuição de cada árvore de decisão do *Ensemble* [10]. Quanto mais reduzido for o seu valor, maior é número de árvores necessárias para ajustar aos dados de treino e vice-versa. Sendo assim, quando este parâmetro toma um valor elevado, origina um número de árvores de decisão insuficiente para aplicar aos dados. Em contrapartida, quando toma um valor de reduzido, origina muitas árvores, pelo que pode ocorrer *overfitting*. Geralmente, esta metodologia de regularização denomina-se de *shrinkage* e as previsões finais obtém-se a partir de:

$$f(x)^{(m)} = f(x)^{(m-1)} + \lambda r \tag{3.8}$$

Neste sentido, o pseudocódigo relativo ao algoritmo do *Gradient Boosting*, para um problema de classificação, encontra-se apresentado na Tabela 3.3 .

Tabela 3.3 – Algoritmo relativo ao *Gradient Boosting*

Algoritmo 2: *Gradient Boosting*

Inicializar as predições das observações dos dados: $f(x_i)^{(0)} = \log\left(\frac{\hat{p}}{1-\hat{p}}\right)$

Para $m = 1$ até M **fazer**

i. Converter as predições de cada observação em probabilidades: $\hat{p}_i = \frac{1}{1+e^{[-f(x_i)]}}$, para $i = 1, \dots, n$

ii. Calcular os resíduos de cada observação: $z_i = y_i - \hat{p}_i$, para $i = 1, \dots, n$

iii. Gerar uma amostra aleatória a partir dos dados de treino

iv. Treinar uma árvore de decisão $H(x, z)$ na amostra gerada e utilizar os resíduos como variável resposta

v. Calcular as estimativas dos resíduos de cada nodo terminal R_j da árvore de decisão:

$r_j = \frac{\sum_{x_i \in R_j}(y_i - \hat{p}_i)}{\sum_{x_i \in R_j} \hat{p}_i\,(1-\hat{p}_i)}$, com $j = 1, \dots, J$

vi. Atualizar o modelo atual $f(x)^{(m)} = f(x)^{(m-1)} + \lambda r$

Fim

O *Gradient Boosting* é um algoritmo que também providencia uma estimativa numérica do impacto de cada variável em estudo para o conjunto de modelos desenvolvidos. Desta forma, em todos os nodos, cuja variável *X* foi usada para ramificar as árvores do *Ensemble*, existe associado um valor da métrica de partição utilizada. Assim sendo, procede-se à agregação do valor dessa métrica, resultante de cada

uma das árvores de decisão que a utilizou essa variável para a partição dos dados. Finalmente, a medida de importância da variável *X* é dada através cálculo da média desses valores de importância, no conjunto de modelos do *Ensemble* [10].

É de salientar as metodologias que, tal como o *Gradient Boostimg*, utilizam como base de aprendizagem estatística progressiva, tendem, geralmente, a originar bons resultados. Resultante do facto de, através da aplicação de modelos aos erros residuais, ser possível melhorar, de forma progressiva, determinadas áreas, onde cada um dos modelos singulares não se consegue ajustar de forma correta aos dados.

3.2.6. Comparação de Modelos

De modo a finalizar a secção 3.2, procede-se a uma breve análise comparativa e sistemática dos modelos apresentados, destacando os pontos fortes e os pontos fracos de cada um, tal como é possível observar na Tabela 3.4.

Tabela 3.4- Análise comparativa de modelos de ML

Modelo	Pontos Fortes	Pontos Fracos
Árvores de Decisão	• Algoritmos versáteis; • Capazes de lidar com variáveis de qualquer natureza; • Fáceis de interpretar;	• Suscetíveis a *overfitting;* • Sensíveis a pequenas variações nos dados; • Pouco robustas;
Random Forest	• Utiliza amostras de *bootstrap* para o treino do modelo; • Robusto a dados com ruído e capaz de lidar com grandes volumes de dados; • Fornece uma estimativa numérica da importância de cada variável;	• Dificuldade de interpretação; • Requer ajustes de hiper-parâmetros; • Requer maior esforço computacional para a sua execução (tempo e memória);
Gradient Boosting	• Utiliza metodologias de b*oosting* para treinar o modelo; • Gera modelos precisos, isto é, apresenta bom desempenho; • Robusto a dados com ruído; • Providencia uma estimativa numérica da importância das variáveis;	• Difículdade de interpretação; • Requer ajustes de hiper-parâmetros; • Requer maior esforço computacional para a sua execução; • Propenso a *overfitting* com bases de dados reduzidas;

Face a estas considerações, quando estamos na presença de grandes volumes de dados que apresentam algum ruído, isto é, valores omissos, *outliers* ou variáveis de natureza diferente, os modelos *Random Forest* e *Gradient Boosting* parecem ser mais adequados. Com efeito, apesar de serem reconhecidas algumas desvantagens, nomeadamente a nível de recursos computacionais e interpretabilidade, nota-se que ambos os modelos são mais robustos e apresentam melhor desempenho em comparação com uma única árvore de decisão.

3.3. Métricas de Avaliação dos Modelos

Após ser ajustado um modelo aos dados de uma população em estudo, um passo importante do processo, consiste em avaliar o desempenho do mesmo. Isto significa que, tendo em conta as previsões obtidas, é crucial perceber em que pontos o modelo que foi ajustado comete erros de previsão e, por outro lado, onde é que as previsões são corretas. Em problemas de classificação binários, a avaliação da performance de um modelo está diretamente relacionada com o facto de o modelo ajustado conseguir identificar de forma correta as classes a que as observações dos dados pertencem.

Deste modo, o desempenho de um modelo de classificação pode ser resumido através de uma matriz de confusão, tal como a que se pode observar na Figura 3.6. O objetivo de uma matriz de confusão, é contar o número de observações que foram classificadas de forma correta e incorreta. Para esse efeito, é necessário recorrer à comparação das verdadeiras classes a que pertencem as observações dos dados com as classes previstas pelo modelo [11]. Neste sentido, cada linha de uma matriz de confusão representa a classe atual dos dados, enquanto cada coluna representa a classe prevista pelo modelo.

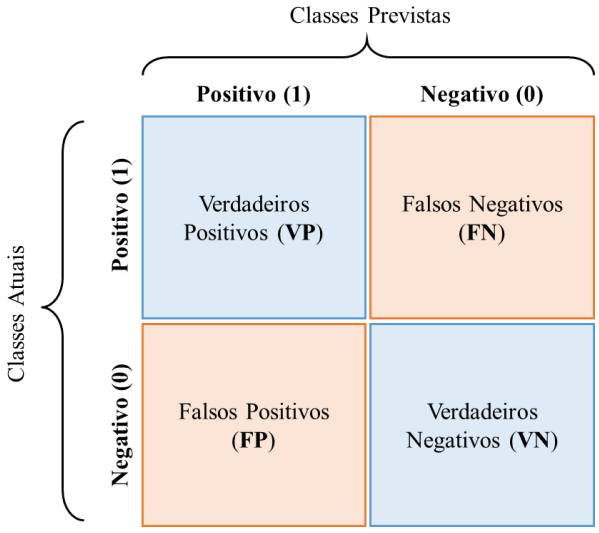

Figura 3.6 - Matriz de confusão de um problema de classificação binário

A primeira linha da matriz considera as observações em que a classe verdadeira é positiva (1), sendo que, dessas observações, algumas são, corretamente, classificadas como positivas, pelo que se denominam verdadeiros positivos (VP). Em contrapartida, nesta linha existem observações em que a sua verdadeira classe é positiva, contudo, são classificadas como negativas (0), a estas chamamos de falsos negativos (FN).

A segunda linha da matriz de confusão, considera as observações em que a classe atual é negativa, sendo que também aqui, existem observações onde a classe foi, incorretamente, prevista como positiva, às quais se dá o nome de falsos positivos (FP). Por outro lado, as observações da segunda linha, que tal como a sua verdadeira classe foram classificadas como negativas, designam-se de verdadeiros negativos (VN).

Com efeito, a matriz de confusão fornece informação pertinente para a avaliar o desempenho de um modelo. No entanto, por vezes, é necessário ter em conta determinadas métricas específicas que se mostram relevantes para o estudo da performance de um modelo. Uma métrica de avaliação interessante de se ter em conta é a exatidão das previsões que o modelo origina, também designada de acurácia, a mesma é definida pela proporção de observações que este consegue classificar de forma correta. Assim, a exatidão/acurácia de um modelo é dada pela divisão do número de previsões corretamente classificadas pelo número total de previsões, pelo que resulta:

$$acurácia = \frac{VP + VN}{VP + FN + VN + FP} \tag{3.9}$$

Alternativamente, a percentagem de erro do modelo é definida pela proporção de observações classificadas de forma incorreta. Por conseguinte, é fácil perceber que a acurácia e a percentagem de erro são complementares, pelo que a sua soma tem de ser igual a 1.

A precisão de um classificador é mais uma métrica utilizada para a avaliação do desempenho, que é considerada importante. Esta é definida como a precisão das previsões positivas, sendo dada pela proporção de observações que foram classificadas como positivas, que são previstas de forma correta, logo surge:

$$precisão = \frac{VP}{VP + FP} \tag{3.10}$$

Tipicamente, a precisão é usada juntamente com outra métrica de avaliação, denominada de sensibilidade ou taxa de verdadeiros positivos. A sensibilidade é definida pela proporção de observações com classe atual positiva, que são detetadas corretamente como positivas pelo classificador, assim tem-se que:

$$sensibilidade = \frac{VP}{VP + FN} \tag{3.11}$$

Outra métrica utilizada para a avaliação é a especificidade ou taxa de verdadeiros negativos, que avalia a capacidade de o modelo conseguir identificar de forma correta observações da classe negativa. Assim, a especificidade é definida pela proporção de observações com classe atual negativa, que são devidamente detetadas como negativas pelo classificador, logo vem que:

$$especificidade = \frac{VN}{VN + FP} \tag{3.12}$$

Quando se pretende efetuar uma comparação entre dois classificadores, é frequente proceder-se à combinação de duas métricas, anteriormente mencionadas (precisão e sensibilidade), numa métrica que se designa de F_1 *score*. Esta métrica é definida pela média harmónica entre a precisão e a sensibilidade, vindo:

$$F_1 = 2 \times \frac{precisão \times sensibilidade}{precisão + sensibilidade} = \frac{VP}{VP + \frac{FN + FP}{2}} \quad (3.13)$$

O F_1 score, tem tendência a favorecer classificadores que têm precisão e sensibilidade semelhante, contudo, esta situação de similaridade nem sempre ocorre. Com efeito, em determinadas ocasiões torna-se relevante dar mais atenção à precisão e noutras à sensibilidade. Isto significa, que dependendo do contexto do problema em estudo, se pode pretender rejeitar muitos verdadeiros positivos e portanto, obter um elevado número de FN, que originam uma sensibilidade do modelo reduzida. Nestes casos, pretendem-se manter as observações que com certeza são positivas, pelo que a intenção é que existam poucos FP e, consequentemente, obter uma precisão elevada [11].

Por outro lado, existem situações onde é crucial identificar como positivas todas as observações com classe atual positiva, sem que nenhuma seja rejeitada. Isto quer dizer que se pretende reduzir ao máximo o número de FN (maior sensibilidade). Nesta situação, em particular, pode existir um maior número de FP, pelo que se obtém um modelo com menor precisão. Assim, é simples perceber que quando aumenta a precisão, reduz a sensibilidade e vice-versa. A esta situação dá-se o nome de *tradeoff* entre a precisão e a sensibilidade.

Tendo em conta as previsões obtidas pelo classificador, pode ser selecionado um *threshold*, segundo o qual as observações são consideradas da classe positiva ou da classe negativa. Neste sentido, as observações com *score* de previsão superior ao *threshold*, são consideradas da classe positiva, em contrapartida, as que têm um *score* inferior a esse valor são da classe negativa.

Salienta-se que, se selecionarmos diferentes valores de *threshold*, obtemos valores de precisão e sensibilidade diferentes. Com efeito, se o *threshold* selecionado tomar um valor reduzido, a sensibilidade aumenta e a precisão diminui, se tomar um valor elevado ocorre o contrário, tal como se pode verificar na Figura 3.*7*.

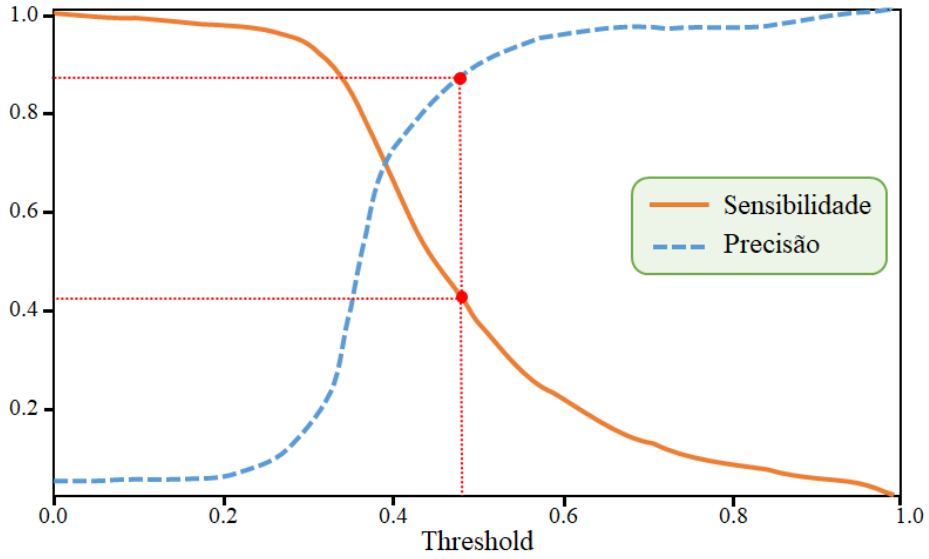

Figura 3.7 - *E*xemplo gráfico do *tradeoff* entre a precisão e a sensibilidade, para diferentes *thresholds.*

Fonte: Figura adaptada de [11].

Em suma, quanto mais reduzido for o valor do *threshold* selecionado, maior é o número de observações dos dados que são consideradas da classe positiva. Neste caso, o número de FP aumenta, dado que muitas das observações cuja classe atual é negativa são consideradas como positivas, pelo que se obtém uma precisão reduzida. Por outro lado, quando o *threshold* selecionado é elevado, a maior parte das observações dos dados, são consideradas como sendo da classe negativa. Nesta situação, obtém-se um valor de sensibilidade reduzido, dado que o número de FN aumenta, devido a muitas das observações que são consideradas, como sendo da classe negativa, serem na realidade da classe positiva.

É importante referir que não existe uma forma única de seleção do *threshold* para otimizar um problema. De facto, tal como foi acima mencionado, o próprio problema em estudo tem influência na métrica de avaliação a que se pretende conceder maior destaque e, consequentemente, no *threshold* a selecionar.

A curva *receiver operating charateristic* (ROC) constitui outra metodologia de avaliação da performance de um modelo de classificação binária. A curva de ROC representa a taxa de verdadeiros positivos contra a taxa de falsos positivos para vários *thresholds*. A taxa de falsos positivos corresponde à proporção de observações com classe atual negativa, que são classificadas como positivas. Note-se que este valor é complementar à especificidade ou taxa de verdadeiros negativos, anteriormente definida. Assim sendo, a curva de ROC, representa a sensibilidade versus $1 - especificidade$ [11].

Também aqui existe um *tradeoff* entre a sensibilidade e a taxa de falsos positivos, dado que quanto maior é a sensibilidade (taxa de verdadeiros positivos), maior é o número de FP que o classificador produz, tal como se pode verificar na Figura 3.8. A linha a tracejado, apresentada na mesma figura, representa a curva de ROC de um classificador completamente aleatório. Assim, considera-se que a curva de ROC de um bom classificador se afasta o máximo possível dessa linha, na direção do canto superior esquerdo.

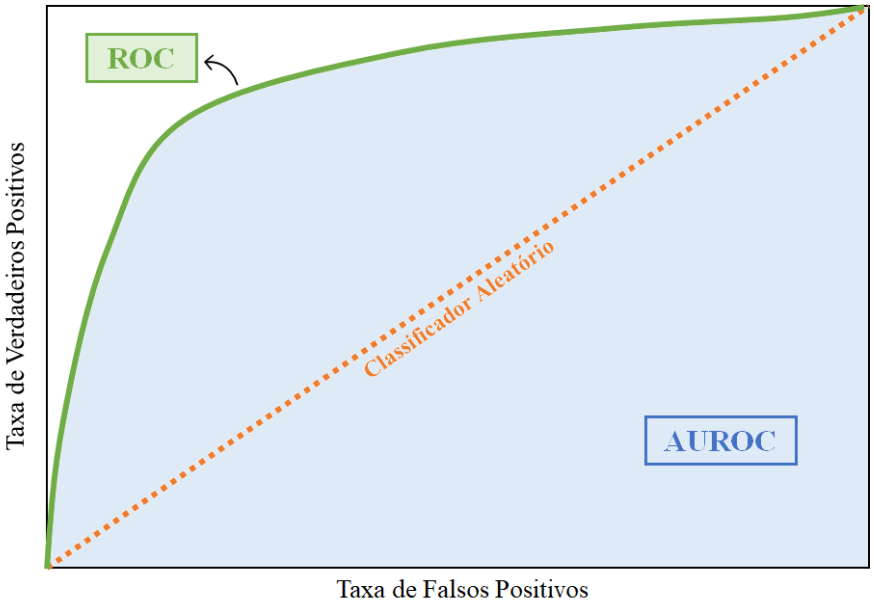

Figura 3.8 - Exemplo gráfico de uma curva de ROC e AUROC de um problema de classificação binário

Uma forma eficiente de efetuar uma comparação entre classificadores, consiste na medição da *area under the ROC curve* (AUROC). A AUROC representa a medida de separabilidade de classes, ou seja, traduz o quão acertada é a capacidade de um classificador distinguir as duas classes. Quanto maior for

a AUROC, melhor é a capacidade de o modelo prever, corretamente, as classes negativas e classes positivas. Desta forma, quanto mais próxima a curva ROC estiver do canto superior esquerdo, maior é a AUROC e melhor é o classificador.

Neste sentido, conclui-se que um classificador é considerado perfeito quando possui um valor de AUROC igual a 1 e, por isso, distingue de forma perfeita as classes. Um classificador é considerado completamente aleatório, se o valor de AUROC for igual a 0.5, o que significa que este não tem capacidade para efetuar distinção entre classes. Finalmente, quando o valor de AUROC é, aproximadamente 0, o modelo identifica as classes de forma inversa, isto é, prevê a classe negativa como positiva e vice-versa.

3.4. Calibração de Probabilidades

Quando lidamos com problemas de classificação, pretende-se que as probabilidades de classe ou *scores*, estimados pelo modelo ajustado, traduzam o verdadeiro valor da probabilidade subjacente à amostra de dados em estudo. Na realidade, o modelo ajustado aos dados pode-se revelar demasiado otimista ou pessimista, nas suas previsões. Assim, as probabilidades previstas (*scores*) necessitam de ser calibradas, de modo a refletirem a verdadeira probabilidade do evento em estudo [10]. Para esse efeito, utiliza-se, geralmente, um subconjunto do conjunto de dados em estudo, o qual se designa de conjunto de validação.

A título de exemplo, se para uma determinada observação, o modelo prevê uma probabilidade de 20% de esta pertencer à classe positiva, então, este valor está bem calibrado, se em média 1 de 5 observações com caraterísticas semelhantes, também são da classe positiva.

Uma forma de avaliar a qualidade das probabilidades obtidas por um classificador consiste na construção de um gráfico de calibração. Este gráfico permite obter uma visualização da probabilidade observada de o evento em estudo ocorrer, versus a probabilidade de classe prevista pelo modelo. Para esse efeito, após ser ajustado um modelo de classificação a um determinado conjunto de dados, é necessário agrupar os *scores* obtidos. Deste modo, os dados são agrupados de acordo com as probabilidades previstas para cada observação, como, por exemplo, em conjuntos como os que se seguem: $[0{,}10\%], [10\%, 20\%], \dots, [\,90\%, 100\%]$ [10].

Após os dados serem agrupados, determina-se a proporção de observações com classe positiva em cada um dos conjuntos formados. No seguimento do exemplo anterior, suponhamos que no grupo de probabilidades inferiores a 10% ($[0{,}10\%]$) se encontram 50 observações e uma única observação com classe observada positiva. Nestas circunstâncias, o ponto médio do conjunto $[0{,}10\%]$, isto é, a probabilidade média prevista corresponde a 5% e a taxa observações positivas observadas a 2% $(1/50\,)$ [10].

Neste sentido, no gráfico de calibração representa-se o ponto médio de cada um dos grupos formados, no eixo do x e a respetiva taxa de eventos positivos observados nesse grupo, no eixo do y. É de salientar que, quando os pontos do gráfico se situam sob a reta $x = y$, significa que o modelo originou probabilidades bem calibradas.

Quando se verifica que as probabilidades previstas pelo modelo, não se encontram calibradas, é necessário recorrer a métodos que possibilitam o ajuste do seu padrão. Um dos métodos mais utilizados

denomina-se de Platt Scalling e recorre a um modelo de regressão logística para efetuar a transformação dos *scores* em probabilidades calibradas. A regressão logística é um modelo de ML, utilizado para problemas de classificação, que tem como finalidade estimar a probabilidade de um determinado evento ocorrer.

Tal como foi referido na secção 3.2.5, se $\hat{p}$ representa a probabilidade de o evento ocorrer, as *odds* desse evento correspondem a $\frac{\hat{p}}{1-\hat{p}}$. Assim, a regressão logística modela o logaritmo das *odds* de um evento, como uma função linear, pelo que surge:

$$log\left(\frac{\hat{p}}{1-\hat{p}}\right) = \beta_0 + \beta_1 x_1 + \cdots + \beta_P x_P \tag{3.14}$$

onde P representa o número de variáveis independentes presentes nos dados. Por norma, o lado direito da equação acima apresentada, denomina-se de componente linear. Uma vez que o logaritmo das *odds* pode variar de -∞ a +∞, não há preocupação com o intervalo de valores que os termos da componente linear tomam. Assim, depois de algumas simplificações, obtém-se:

$$\hat{p} = \frac{1}{1 + e^{[-(\beta_0+\beta_1 x_1+\cdots+\beta_P x_P)]}} \tag{3.15}$$

Esta função não linear é uma função sigmoidal dos termos do modelo, que restringe as estimativas de probabilidades a um intervalo entre 0 e 1.

Neste sentido, o método de Platt Scalling ajusta um modelo de regressão logística às probabilidades previstas pelo classificador binário utilizado, de modo a obter as respetivas probabilidades calibradas [10]. Isto significa que as previsões obtidas pelo modelo são utilizadas para processar as estimativas de probabilidades a partir da expressão:

$$\hat{p}^* = \frac{1}{1 + e^{[-(\beta_0+\beta_1\hat{p})]}} \tag{3.16}$$

onde os parâmeros β são estimados através da previsão das verdadeiras classes, em função das probabilidades de classe não calibradas $\hat{p}$, obtidas pelo classificador. Posto isto, após ser efetuada a calibração de probabilidades, a amostra de dados deve ser, novamente, classificada, de modo a garantir a consistência entre as novas probabilidades e as classes previstas.

3.5. Otimização de Hiper-parâmetros

Por norma, durante o processo de modelação, procede-se à divisão dos dados em estudo em dois subconjuntos de dimensões mais reduzidas, um conjunto para treino e um conjunto para teste de um algoritmo. Neste sentido, no primeiro conjunto de dados, tal como o nome indica, efetua-se o treino de um algoritmo. Por sua vez, dado que o segundo conjunto de dados é desconhecido para o algoritmo, é utilizado para obter as respetivas previsões e avaliar sua performance [10].

Encontrar a combinação de hiper-parâmetros, que otimiza determinado modelo, constitui uma etapa crucial do processo de modelação de um conjunto de dados. No entanto, o desempenho de um modelo de ML é bastante sensível aos hiper-parâmetros selecionados, pelo que proceder à sua otimização, tendo em conta apenas um conjunto de dados para treino e teste, não é, geralmente, recomendado.

Com efeito, treinar e validar um modelo, apenas num conjunto de dados, pode originar um modelo com *overfitting* ou até reduzir a sua capacidade de generalização. Assim, uma boa prática na fase de modelação, consiste em treinar e validar um modelo em vários subconjuntos, do conjunto de dados de treino, através de um método designado de *k-fold cross-validation.*

De forma sucinta, a *cross-validation* é uma metodologia de ML, composta por dois princípios fundamentais, a saber: efetuar a partição dos dados de treino em k, conjuntos aleatórios com a mesma dimensão, denominados de *folds* e exercer rotatividade entre os conjuntos utilizados para treino e validação de um modelo. Neste sentido, quando se procede à divisão dos dados designados para treino, são obtidos k conjuntos, sendo que destes $k - 1$ são, efetivamente, utilizados para o treino do modelo, e 1 é utilizado como conjunto de validação, onde se avalia o desempenho do mesmo.

Este processo repete-se k vezes, de modo a que cada uma das k *folds* resultantes das partições dos dados seja utilizada tanto para o treino como para a validação do algoritmo, tal como é possível observar na Figura 3.9. Desta forma, numa primeira iteração, a primeira *fold* é utilizada para validação e as restantes $k - 1$ para treino. Na segunda iteração, a primeira *fold* passa a ser utilizada para treino, a segunda para validação e as restantes para treino. Este processo ocorre de forma sucessiva, até todos os conjuntos (*folds*) formados terem sido rodados entre treino e validação.

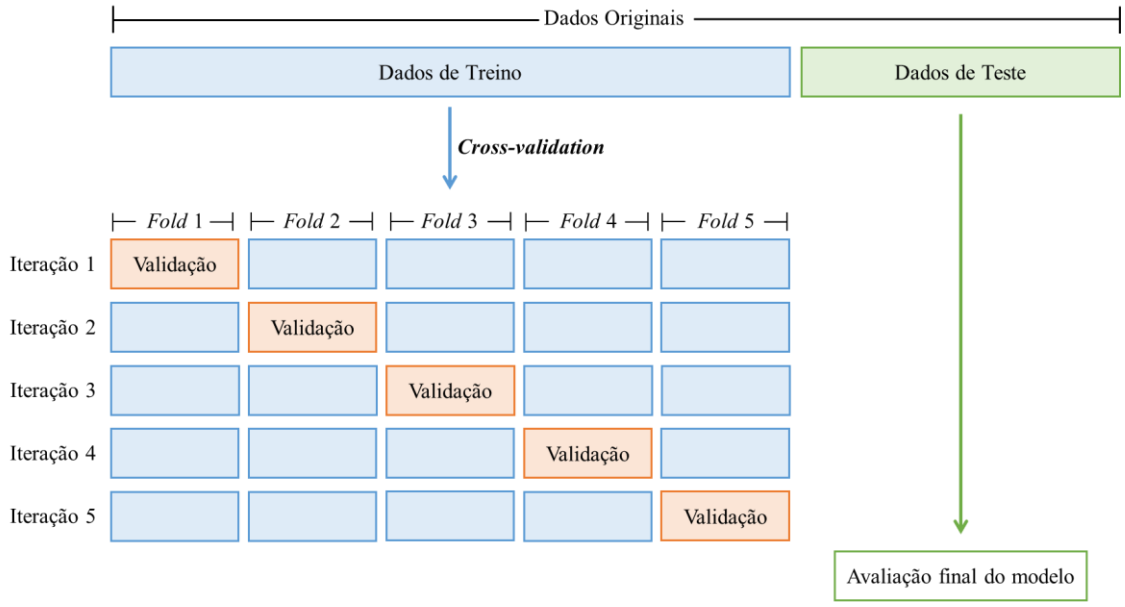

Figura 3.9 - Esquema ilustrativo da partição dos dados em treino e teste, com *5-fold Cross-validation*

Tal como foi acima mencionado, pretende-se que em cada um dos conjuntos de validação, seja medida a performance do algoritmo treinado. Para esse efeito, é necessário ter em conta uma determinada métrica de avaliação para algoritmos de classificação, tais como as que foram referidas na secção 3.2.6 (AUROC, exatidão, precisão, …).

Sendo assim, em cada uma das k iterações da *cross-validation* é registado o valor da métrica escolhida para a validação da qualidade do modelo. Deste modo, no final do processo, o valor da métrica que permite avaliar o modelo treinado é dada pela média dos valores obtidos dessa mesma métrica, em cada um dos conjuntos de validação. Por fim, após serem encontrados os hiper-parâmetros, que otimizam um determinado modelo, procede-se à sua avaliação no conjunto de dados para teste, de modo a verificar a qualidade da sua performance.

Uma metodologia eficiente para encontrar os melhores parâmetros de um determinado modelo de ML é a otimização Bayesiana que tem como finalidade encontrar os extremos de uma determinada função objetivo [18], [19], [20]. Este método de otimização é, particularmente, útil em situações onde otimizar uma determinada função objetivo se revela uma tarefa dispendiosa, dado que não se possui qualquer conhecimento acerca das suas derivadas e o problema em estudo não ser convexo. Com efeito, este procedimento de otimização aplica-se em situações onde a função objetivo que se pretende otimizar não tem uma expressão específica que seja conhecida, contudo, é possível obter observações dessa mesma função a partir de certos espaços de valores amostrados.

Esta prática, denomina-se de Bayesiana, uma vez que a sua aplicação tem como base o teorema de Bayes. Este teorema afirma que a probabilidade a posteriori de um modelo (ou hipótese) M, tendo por base uma evidência E, é proporcional à probabilidade de E com base em M, multiplicada pela probabilidade a priori de M:

$$P(M|E) \propto P(E|M)P(M) \tag{3.17}$$

No que toca à estatística Bayesiana, a nomenclatura a priori representa uma certa crença sobre o espaço de possíveis funções objetivo. Embora a função objetivo, que se pretende otimizar, seja desconhecida, é razoável assumir que existe um conhecimento prévio sobre algumas das suas propriedades, o que torna certas funções mais plausíveis do que outras.

Deste modo, o princípio chave da otimização Bayesiana reside, precisamente, na equação 3.17, uma vez que ao incorporar perceções prévias sobre o problema, consegue direcionar o estudo e proporcionar um equilíbrio entre a procura e a exploração de um determinado espaço de hiper-parâmetros. Desta forma, se existir uma determinada crença prévia de que a função objetivo não tem ruído, os dados com elevada variância ou oscilações devem ser considerados menos prováveis, do que aqueles que apenas se desviam da média.

Posto isto, este é um método de otimização de hiper-parâmetros, onde a função objetivo avalia várias combinações de possíveis hiper-parâmetros de um modelo. Inicialmente, este método recebe um espaço de hiper-parâmetros, previamente definido e, posteriormente, são avaliadas várias combinações desses parâmetros, segundo a métrica de avaliação obtida no conjunto de validação. Deste modo, num problema de classificação, o foco é maximizar a métrica de avaliação escolhida (qualquer uma das mencionadas na secção 3.2.6), pelo que se pretende maximizar o valor da função objetivo:

$$x_{opt} = arg\ max_{x \in \mathcal{A}}\ f(x) \tag{3.18}$$

onde o conjunto de soluções viáveis e a função objetivo têm, normalmente, as seguintes propriedades:

- O *input* x está em $\mathbb{R}^d$, para um valor de d que não seja demasiado elevado;
- O conjunto $\mathcal{A}$ é delimitado (é originado pelo produto de domínios univariados ligados e limitados);
- A função objetivo f é contínua;
- Avaliar a função objetivo f é uma tarefa dispendiosa, dado que o número de avaliações que é possível executar é limitado. Esta limitação deve-se ao facto de cada iteração demorar uma quantidade substancial de tempo para ser executada e ser necessária uma quantidade elevada de recursos computacionais;
- A expressão e as propriedades (concavidade ou a linearidade) de f não são conhecidas;
- O foco deste tipo de otimização é encontrar um ótimo global, em vez que um ótimo local.

Idealmente, pretende-se ativar a função objetivo, quando estivermos razoavelmente seguros de que temos um conjunto de hiper-parâmetros, que conduz ao melhor valor da métrica de avaliação no conjunto de validação. Por este motivo, são necessários dois mecanismos de apoio que permitam identificar quais os conjuntos de hiper-parâmetros avaliados até ao momento e utilizar essa informação para sugerir novos conjuntos de hiper-parâmetros a ser avaliados.

Neste sentido, o processo é inicializado por um modelo probabilístico de regressão $\mathcal{M}$, a partir de um conjunto de amostras do domínio $\mathcal{A}$. De outro modo, significa que se pretende atualizar a distribuição a posteriori de f, a partir dos dados disponíveis. É de salientar que a probabilidade a posteriori capta as nossas crenças atualizadas sobre a função objetivo desconhecida, pelo que este passo pode ser interpretado como uma estimativa da função objetivo a partir de uma função de substituição.

A otimização Bayesiana utiliza ainda uma função de aquisição S que permite determinar qual é o melhor ponto local a ser analisado na próxima iteração. Deste modo, a partir do modelo probabilístico atual, são selecionados, sequencialmente, vários pontos do domínio através da otimização da função de aquisição S. Com efeito, a função de substituição e a função de aquisição funcionam em conjunto, de modo a propor combinações de parâmetros, que conduzem ao melhor valor da métrica de avaliação na função objetivo.

No entanto, nesta etapa do processo, deparamo-nos com um dilema entre a pesquisa e a exploração de pontos locais. De facto, pretendem-se encontrar vários valores de x distintos, a fim de aumentar o conhecimento acerca do espaço de valores em estudo que sirva de base à decisão em causa. Por outro lado, procura-se explorar valores de x, onde se espera que a função objetivo seja elevada. Para esse efeito, é necessário considerar as informações das iterações anteriores, de modo a ser possível escolher os valores de x, que conduzem a melhorias significativas na função objetivo.

Posto isto, um dos grandes benefícios desta metodologia de otimização de hiper-parâmetros é proporcionar um equilíbrio entre a pesquisa e a exploração de valores de x, uma vez que tanto se pretende possuir conhecimento acerca do espaço de valores em estudo, como se pretende escolher valores que conduzam a recompensas na função objetivo. Para além disso, a partir da função de aquisição, é possível reduzir de forma significativa o número de avaliações necessárias da função objetivo.

Sendo assim, defina-se x_i como a i-ésima amostra e $y_i = f(x_i)$ como a observação da função objetivo em x_i. É de realçar que a observação obtida pode ser aleatória, quer porque f é aleatória, quer porque o processo de observação está sujeito a ruído. Posteriormente, o resultado obtido é adicionado a um conjunto de histórico $\mathcal{D} = \{(x_1, y_1), \ldots, (x_i, y_i)\}$, que é utilizado para atualizar o modelo de regressão $\mathcal{M}$ e gerar a próxima sugestão a ser avaliada [20].

Na prática, existe uma quota de tempo total ou recursos disponíveis para a execução do algoritmo em questão, o que impõe um limite T ao número total de iterações a realizar durante o processo [20]. Assim, o processo de otimização descrito encontra-se apresentado na Tabela 3.5.

Tabela 3.5 – Algoritmo relativo à otimização Bayesiana

Algoritmo 3: Otimização Bayesiana

Input: $f, \mathcal{A}, S, \mathcal{M}$

$\mathcal{D} \leftarrow \text{INITSAMPLES}(f, \mathcal{A})$

Para $i = 1$ até T **fazer**

Atualizar a distribuição a posteriori de f a partir dos dados disponíveis:

$p(y|x, \mathcal{D}) \leftarrow FITMODEL\ (\mathcal{M}, \mathcal{D})$

Seja x_i um maximizante da função de aquisição sobre x, onde a função de aquisição é calculada através da distribuição a posteriori atual, $x_i \leftarrow\ arg\ max_{x \in \mathcal{A}}\ S(x, p(y|x, \mathcal{D}))$

Observar $y_i = f(x_i)$

Aumentar os dados $\mathcal{D}_i = \{\mathcal{D}_{i-1}, (x_i, y_i)\}$

Fim

O *Tree Parzen Estimator* (TPE) é um modelo probabilístico $\mathcal{M}$, frequentemente utilizado neste tipo de metodologias de otimização. O mesmo tem como objetivo modelar a probabilidade de um conjunto de parâmetros, dado um determinado valor da métrica de avaliação $p(x|y, \mathcal{D})$, substituindo as distribuições a priori, por dois processos hierárquicos $\ell(x)$ e $g(x)$, que atuam como modelos generativos para todas as variáveis do domínio [21]. Assim, estes processos modelam as variáveis do domínio, quando a função objetivo se situa em valores superiores ou inferiores de um determinado quantil especificado:

$$p(x|y, \mathcal{D}) = \begin{cases} \ell(x), se\ y < y^* \\ g(x), se\ y \geq y^* \end{cases} \tag{3.19}$$

Neste contexto, $\ell(x)$ representa a densidade originada pela utilização das observações x_i, quando o valor da métrica de avaliação é inferior a y^* (por exemplo, o valor da métrica de avaliação mais elevado até ao momento) e $g(x)$ corresponde à densidade das restantes observações para valores da métrica superiores a y^*. Os estimadores de Parzen são organizados em estrutura de árvore, preservando qualquer dependência condicional especificada, o que origina um ajuste por variável para cada processo $\ell(x)$ e $g(x)$ [21].

Por fim, tal como foi referido anteriormente, os parâmetros propostos a ser avaliados pela função objetivo, são selecionados através da aplicação de um determinado critério ao modelo probabilístico $\mathcal{M}$. Esse critério é definido pela função de aquisição S, sendo que uma possível abordagem, para esse efeito, é o *Expected Improvement* (EI). Assim, a parametrização de $p(x, y)$ como $p(y)p(x|y)$ no algoritmo TPE, facilita a otimização do EI [21]:

$$\begin{aligned} EI_{y^*}(x) &= \int_{-\infty}^{y^*} (y^* - y) p(y|x, \mathcal{D})\ dy \\ &= \int_{-\infty}^{y^*} (y^* - y) \frac{p(x|y, \mathcal{D}) p(y)}{p(x)}\ dy \end{aligned} \tag{3.20}$$

Por construção, $\gamma = p(y < y^*)$ e $p(x) = \int_{\mathbb{R}} p(x|y, \mathcal{D})p(y)\, dy = \gamma\ell(x) + (1-\gamma)g(x)$. Assim, depois de algumas simplificações obtém-se:

$$EI_{y^*}(x) = \frac{\gamma y^* \ell(x) - \ell(x) \int_{-\infty}^{y^*} yp(y)dy}{\gamma\ell(x) + (1-\gamma)g(x)} \propto \left(\gamma + \frac{g(x)}{\ell(x)}(1-\gamma)\right)^{-1} \quad (3.21)$$

A expressão 3.21 mostra que para maximizar o EI é preferível possuir pontos de x com maior probabilidade em $\ell(x)$ e baixa probabilidade em $g(x)$. Verifica-se, ainda, que a estrutura em árvore de ℓ *e* g facilita a seleção de vários candidatos e a respetiva avaliação de acordo com $g(x)/\ell(x)$. Deste modo, infere-se que em cada iteração é devolvido precisamente o candidato x_i com maior EI, para ser avaliado pela função objetivo [21].

3.6. Explicabilidade dos Modelos

Tal como foi referido nas secções 3.2.4 e 3.2.5, após ser efetuado o treino de um determinado modelo de ML, é possível obter uma estimativa da importância de cada uma das variáveis consideradas para o estudo. Porém, a classificação da importância de cada variável por si só não é suficiente para explicar uma determinada previsão individual do modelo.

Com efeito, nesta etapa, pretende-se explorar a explicabilidade do modelo estimado, isto é, interpretar com clareza os resultados obtidos, que servem de apoio à decisão. Em particular, no que concerne a um problema de classificação binário, pretende-se entender o motivo pelo qual determinada observação foi prevista como sendo da classe positiva ou negativa e em que medida cada variável em estudo contribuiu para a previsão final.

Neste sentido, introduz-se o modelo *Shapley Additive Explanations* (SHAP), que tem como finalidade explicar uma predição $f(x)$ de uma observação x através do cálculo da contribuição relativa de cada variável para o resultado [22]. Desta forma, é necessário perceber como são calculados os valores da contribuição de cada variável, também designados de valores de SHAP, para as predições obtidas pelo modelo treinado.

Assim sendo, o valor da contribuição marginal de uma determinada variável, é dado pela diferença entre o resultado da previsão, quando a variável está presente e quando essa mesma variável está ausente. Todavia, este valor de contribuição tem de ser calculado para cada uma das combinações (subgrupos) de variáveis possíveis, onde a variável, em questão, está presente. Assim, o valor de SHAP (contribuição marginal) de uma variável, corresponde à média do valor da contribuição marginal obtido em cada uma das combinações possíveis.

A título de exemplo, suponhamos que estamos na presença de 4 variáveis (X_1, X_2, X_3, X_4), que são utilizadas para prever um determinado resultado P. A fim de calcular a contribuição marginal (valor de SHAP) da variável X_1, é necessário construir todas as possíveis combinações de variáveis, onde esta variável aparece, representadas na Figura 3.10. Para cada combinação, a contribuição marginal é calculada como a diferença do resultado P, quando X_1 está presente, e o resultado P, quando X_1 não está presente. Desta forma, a contribuição marginal da variável X_1 corresponde à média dos valores de contribuição, calculados em cada combinação.

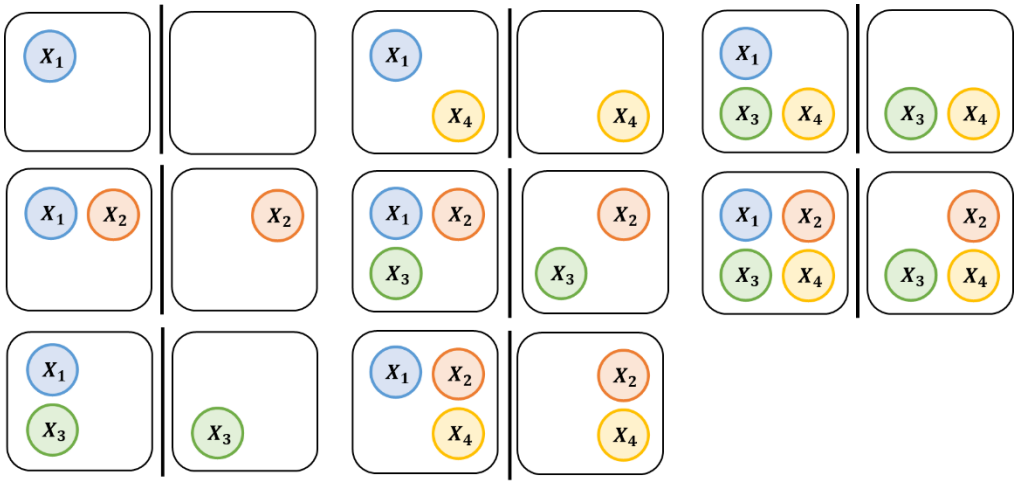

Figura 3.10 – Ilustração de todas as combinações de variáveis possíveis, relativamente à variável X_1 do exemplo dado

Posto isto, o SHAP calcula o impacto de cada variável nas predições obtidas pelo modelo estimado. Para tal, dada uma observação $x = [x_1, \ldots, x_N]$ e um modelo preditivo f treinado, tem-se que o SHAP aproxima f a partir de um modelo $g(.)$, capaz de explicar, facilmente, a contribuição de cada variável para o modelo [23].

O modelo $g(.)$ é caraterizado por apresentar apenas uma única solução, com três propriedades desejáveis: precisão local, ausência e consistência. A precisão local indica que o resultado obtido pelo modelo g se encontra em conformidade com a predição dada pelo modelo f, quando aplicados à mesma observação. A ausência, garante que para as variáveis independentes que não constam no modelo g, a sua contribuição é nula. E, por fim, a consistência estabelece que se o modelo f se alterar, de tal forma que a contribuição de uma variável aumenta ou permanece a mesma, a sua importância no modelo g não é afetada no sentido oposto [24].

Desta forma, a função de explicativa $g(.)$ recebe um vetor binário $z' \subset \{0,1\}^N$, sendo N o número de variáveis presentes em x. O vetor z' representa a presença ou ausência de uma variável: entrada igual a 1, significa que a variável correspondente contribui para a explicação, isto é, foi utilizada para a previsão, enquanto entrada igual a 0 significa que a variável não tem contribuição. Para além disso, $g(.)$ recebe ainda, $\phi_i \in \mathbb{R}$ que representa a contribuição (valores de SHAP) de cada variável independente x_i, para o modelo ajustado [22]. Assim, a função $g(z')$ pode ser formulada da seguinte forma:

$$g(z') = \phi_0 + \sum_{i=1}^{N} \phi_i z'_i, \phi_i \in \mathbb{R} \tag{3.22}$$

Intuitivamente, o modelo g pode ser utilizado para interpretar tanto uma única previsão, como o modelo completo, com base na contribuição média de cada variável em todas as observações dos dados. Por conseguinte, dado um modelo preditivo f e uma observação x, os valores de SHAP podem ser calculados através da expressão:

$$\phi_i(f, x) = \frac{1}{N!} \sum_{S \subseteq P \setminus \{x_i\}} [|S|!\,(N - |S| - 1)!][f(S \cup \{x_i\}) - f(S)] \tag{3.23}$$

onde P representa o conjunto dos valores de todas as variáveis independentes, S o subconjunto de variáveis utilizadas, que não inclui o valor da variável x_i e, por fim, |S|, o número de variáveis independentes não nulas em S.

A distribuição dos valores de SHAP, para cada variável independente, pode ser visualizada através de um diagrama em forma de violino. Assim, no eixo horizontal, (abcissas) representam-se os valores de SHAP e no eixo vertical (ordenadas), as variáveis explicativas, por ordem decrescente dos valores de SHAP. Desta forma, quanto maior for o valor de SHAP no sentido positivo, maior é a contribuição dessa variável na direção positiva, e vice-versa. Neste diagrama, pode, ainda, visualizar-se a ordem de grandeza do valor de uma variável através de cores. Sendo assim, a cor torna-se vermelha à medida que o valor de uma variável aumenta, e azul quando o valor da variável diminui [23]. Deste modo, a partir do diagrama referido, é possível obter uma perceção, de como a ordem de grandeza dos valores de uma variável influenciam a sua contribuição no sentido positivo ou negativo da classificação.

Dado que os modelos de ML se revelam bastantes complexos, é fundamental analisar a sua explicabilidade e interpretar os seus resultados. Com efeito, um dos principais requisitos que um modelo deve ter é a transparência, que se considera crucial para garantir a confiança por parte do ser humano, no produto desenvolvido.

Capítulo 4 – Metodologia e Dados

Neste capítulo, além de serem descritos os dados utilizados para o estudo e as situações de variável resposta que se testaram, procurar-se-á explicar as macro etapas que fizeram parte do projeto, assim como o processo analítico adotado durante a modelação dos dados e, ainda, fornecer algumas considerações relacionadas com a implementação computacional.

4.1.Dados

Como foi referido anteriormente, o objetivo deste estudo é encontrar o contacto decisor de um cliente. No entanto, recorde-se que o âmbito deste projeto ocorre no setor empresarial de uma empresa de telecomunicações, pelo que todos os clientes se referem a empresas/instituições. Deste modo, importa perceber a lógica segundo a qual o painel de variáveis foi construído.

O painel de variáveis, utilizado para a construção do modelo preditivo em questão, é constituído por informação relativa a quatro perfis de variáveis e a variável resposta. Além disso, o intervalo temporal definido para o estudo foi de quatro meses, a saber, setembro, outubro, novembro e dezembro de 2022.

O primeiro perfil conta com variáveis relativas à caraterização do cliente/empresa. Desta forma, neste perfil encontram-se variáveis, como o tipo de empresa do cliente, o respetivo número de colaboradores, o número de contactos associados, a tipificação dos contactos (móvel ou fixo) e se o cliente tem um contacto principal assinalado. Para além disso, existem, ainda, variáveis que indicam a quantidade de serviços que um determinado cliente possui, bem como a sua tipificação, isto é, quantos são serviços de televisão, voz fixa, internet fixa, voz móvel e internet móvel[6].

O painel de variáveis conta, também, com um perfil de variáveis históricas relativas ao tráfego associado a cada contacto de um cliente, nos 2 meses anteriores ao mês de referência. Neste sentido, neste perfil, encontram-se variáveis de tráfego efetuado e recebido para todos os contactos, bem como para contactos pertencentes ao mesmo NIF, nos 2 meses antecedentes.

O terceiro perfil de variáveis é relativo ao histórico de telefonia de cada contacto, nos 6 meses anteriores. Desta forma, as variáveis de telefonia para um determinado contacto, têm em conta o resultado das chamadas anteriores, efetuadas para esse contacto. Alguns exemplos de variáveis que este perfil contém, são: o número de tentativas necessárias para obter uma resposta por parte de um contacto num ciclo (período em que ocorrem as campanhas de *telemarketing*, sendo que um ano, é constituído por 13 ciclos de igual dimensão), rácio de atendimento, e resultado de telefonia do contacto, isto é, se este atendeu ou não uma chamada.

Por fim, o último perfil de variáveis utilizado refere-se à interatividade de cada contacto, com o Serviço ao Cliente (SAC), nos 6 meses antecedentes ao mês de referência. Deste modo, este perfil apresenta variáveis, como o número total de chamadas que um contacto efetua ou recebe do SAC, bem como o número total de tópicos abertos para um contacto, nos 6 meses anteriores. É importante referir que o termo tópico neste contexto, se refere a qualquer assunto ou problema que um cliente necessite

[6] No Anexo A é apresentada a descrição detalhada de cada uma das variáveis utilizadas neste estudo.

de comunicar à empresa, ou vice-versa (assuntos de faturação, problemas técnicos, mudança de morada, agendamentos, etc).

A variável resposta do estudo foi construída a partir da informação dos *Outcomes*, isto é, dos resultados de negócio de cada chamada efetuada para um contacto durante as campanhas de *telemarketing*. Dado que o objetivo do estudo, é encontrar o contacto que toma decisões em nome de uma determinada empresa, entende-se como decisão tanto a recusa, como a aceitação de uma proposta por parte de um contacto.

No que toca aos resultados do negócio, estes são representados numa árvore complexa de *Outcomes*, onde cada resultado se ramifica em inúmeras situações. Por este motivo, para a definição da variável resposta em questão, foi considerado apenas o primeiro e segundo níveis desta árvore.

Neste sentido, no primeiro nível da árvore de *Outcomes*, os possíveis resultados de negócio de uma determinada chamada são os seguintes: sucesso, insucesso com oferta apresentada, insucesso sem oferta apresentada, não contactado, cancelado, *callback* com oferta apresentada (chamada agendada para outro horário) e *outcome* não comercial. Por sua vez, no segundo nível da árvore, teve-se em consideração apenas informação relativa a duas categorias: agendado decisor e *callback* decisor.

Desta forma, a partir dos vários tipos de resultados de negócio referidos, foi criada a variável resposta binária que corresponde à agregação dos diferentes resultados referidos. Sendo assim, consideraram-se quatro agregações diferentes para a formulação da variável resposta:

- **Situação 1:** Neste caso, considera-se que um contacto é decisor, e, por isso, a variável resposta toma o valor 1 (classe positiva), quando os resultados de negócio são: sucesso, insucesso com oferta apresentada e insucesso sem oferta apresentada. Por outro lado, considera-se que um contacto não é decisor, pelo que a variável dependente toma o valor 0 (classe negativa), quando o resultado de negócio corresponde a não contactado, cancelado, *callback* com oferta apresentada e *outcome* não comercial;
- **Situação 1.1:** Este caso é idêntico ao da situação anterior, diferindo apenas na inserção das duas categorias do segundo nível da árvore, para serem consideradas na decisão. Desta forma, uma observação pertence à classe positiva da variável resposta, quando os resultados de negócio são: sucesso, insucesso com oferta apresentada, insucesso sem oferta apresentada, agendado decisor e *callback* decisor. Por sua vez, tal como na situação 1, uma observação pertence à classe negativa da variável resposta, quando o resultado de negócio corresponde a qualquer uma das restantes categorias;
- **Situação 2:** Neste caso, considera-se que um contacto toma decisões, e, por isso, a variável resposta toma o valor 1, quando os resultados de negócio são: sucesso e insucesso com oferta apresentada. Por outro lado, considera-se que um contacto não é decisor, pelo que a variável dependente toma o valor 0, quando o resultado de negócio corresponde a insucesso sem oferta apresentada, não contactado, cancelado, *callback* com oferta apresentada e *outcome* não comercial;
- **Situação 2.1:** Este caso é idêntico ao da situação 2, diferindo apenas na inserção de duas novas categorias para serem consideradas na decisão. Deste modo, a variável reposta toma o valor 1, quando os resultados de negócio são: sucesso, insucesso com oferta apresentada, agendado decisor e *callback* decisor. Por outro lado, tal como na situação 2, a variável resposta toma o valor 0, quando o resultado de negócio, corresponde a qualquer uma das restantes categorias.

Constata-se que a grande diferença entre a situação 1 e 2 é que esta última corresponde a uma formulação mais restrita da variável resposta. Isto significa que apenas se consideram como decisores os contactos a quem, anteriormente, foram apresentadas as propostas de negócio e a partir daí foi tomada uma decisão (recusa ou aceitação).

Desta forma, cada observação do painel de dados refere-se a uma chamada de telefonia efetuada para um contacto que tenha sido atendida, num determinado mês. Com efeito, o estudo centra-se num universo de chamadas atendidas, uma vez que um indivíduo só tem oportunidade para tomar decisões quando atende uma chamada.

Neste sentido, é expectável que num mês existam várias chamadas efetuadas para um determinado contacto. A título de exemplo, se no mês de setembro foram efetuadas quatro chamadas para um contacto, no painel de dados aparecem 4 observações para esse contacto nesse mês, sendo que o que difere é o resultado do negócio. De facto, o evento em estudo é precisamente o resultado de negócio das chamadas anteriores.

Posto isto, o painel de variáveis é constituído por cerca de 82500 observações e 166 colunas, sendo que destas, 165 correspondem a variáveis independentes e uma à variável resposta (tendo em conta a situação que se pretende testar). No que toca à tipologia das variáveis explicativas, na sua maioria são variáveis quantitativas, sendo que existem 12 variáveis binárias e apenas duas variáveis qualitativas nominais. Das variáveis quantitativas, 132 são contínuas e 19 são discretas.

As variáveis categóricas são o tipo de empresa de cada cliente (B_TIPOLOGIA) e a fonte da qual provém os contactos (B_TABELA_FONTE). No que diz respeito ao tipo de empresa de um cliente, esta variável apresenta 3 categorias, por sua vez, a fonte da qual provém os dados, é constituída por duas classes, tal como se pode observar na Figura 4.1.

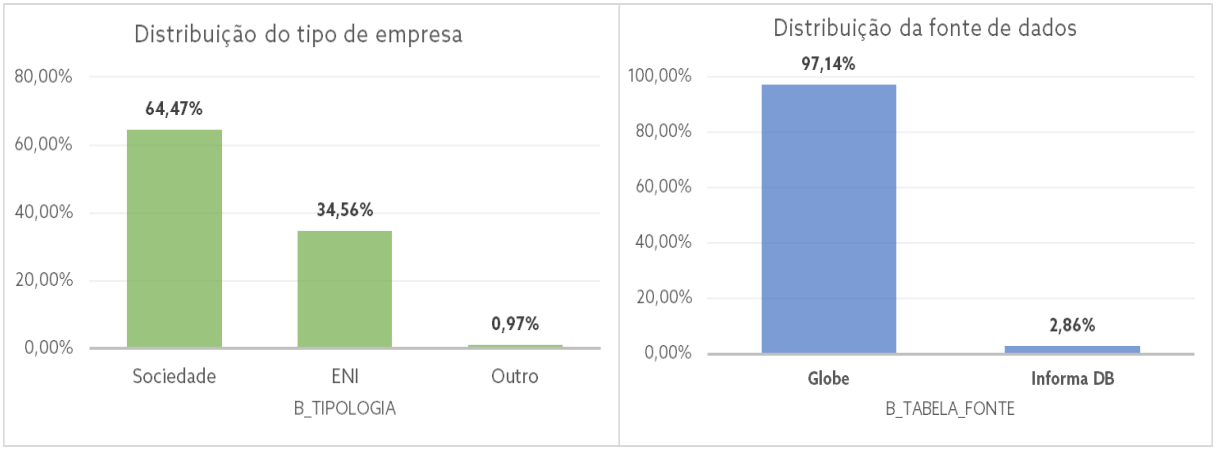

Figura 4.1 - Gráficos da distribuição das classes, de cada uma das variáveis categóricas, presentes nos dados

Assim, verifica-se que, do conjunto de empresas utilizadas para este estudo, cerca de 64,47% são sociedades, 34,56% empresários em nome individual e 0,97% Outros. Relativamente à fonte da qual provém os dados, cerca de 97,14% dos dados resultam de GLOBE e os restantes da Informa DB.

É importante realçar que o painel não contém observações duplicadas e conta com 137 variáveis onde existem valores omissos. Destas variáveis, a sua maioria possui menos de 15% de valores em falta, ainda assim, a maior percentagem registada é de 56%, que se verifica em algumas variáveis de tráfego e telefonia.

No que se refere aos resultados de negócio, das chamadas que constituem o painel de variáveis, verifica-se que aquele que predomina é o *callback* com oferta apresentada (cerca de 56,34% das observações), tal como se pode observar na Figura 4.2. Por sua vez, o insucesso com oferta apresentada representa 17,55% das observações e o insucesso sem oferta apresentada cerca de 12,56%. Por fim, os resultados de negócio que apresentam menor representatividade no painel são o cancelado, o sucesso e o *outcome* não comercial (1,61%, 1,11% e 0,13% das observações, respetivamente).

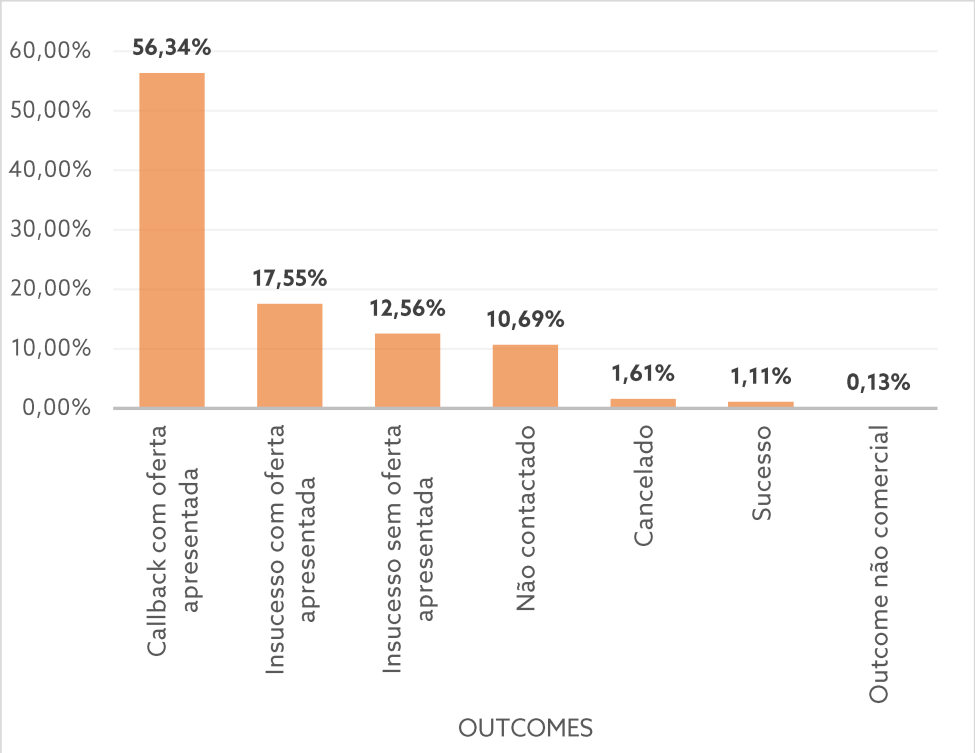

Figura 4.2 - Gráfico que apresenta a distribuição dos resultados de negócio, no painel de variáveis

4.2.Procedimentos Metodológicos

Num projeto desta dimensão, é de extrema importância, que o produto analítico que se pretende desenvolver esteja alinhado com as necessidades dos processos de negócio/operação. Neste sentido, a primeira etapa do projeto consistiu na definição do problema, ao qual se pretendia dar resposta. Assim sendo, foi necessário identificar e compreender o desafio, definir as abordagens a seguir, ou seja, o que se esperava obter e como implementar os *outputs* do modelo na operação e, ainda, definir os KPI (*Key Performance Indicators*) que se pretendiam atingir (neste caso, aumentar a taxa de decisão).

Ainda com o intuito de consolidar a definição do problema, foi necessário identificar as fontes de dados que deveriam ser utilizadas para o estudo. Assim, foi nesta etapa que se procedeu à exploração dos dados, nomeadamente informação de *Outcomes*, de modo a ser possível adquirir conhecimento detalhado sobre a informação existente. Com efeito, a exploração de dados foi uma etapa crucial, no que toca à definição da variável resposta que se pretendia estudar (tal como foi referido na secção 4.1, foram testadas quatro formulações distintas da variável resposta).

Após ter sido definido o âmbito de todo o projeto, era evidente que se pretendia desenvolver um modelo preditivo de classificação, que recomenda o conjunto de contactos mais prováveis de serem os decisores. De facto, é de salientar que a definição do problema e a exploração de dados são etapas complementares e de extrema importância para o sucesso do projeto.

Posteriormente, o foco incidiu na identificação do intervalo temporal do estudo e na preparação dos dados para a construção do painel de variáveis. Para esse efeito, foi efetuado o tratamento de informação, identificação e cálculo de variáveis, que caraterizam os contactos como decisores (caraterização de empresas, tráfego, telefonia e interatividades com o SAC).

Uma vez terminado o painel de variáveis, reuniram-se as condições necessárias para iniciar a modelação dos dados. Nesta fase, procedeu-se à identificação, desenvolvimento e experimentação de metodologias de ML, com a finalidade de desenvolver o modelo preditivo pretendido. Assim, tal como foi referido anteriormente, foram testadas quatro formulações distintas da variável resposta, sendo que se avançou com aquela que apresentou melhores resultados.

Desta forma, após ser validado o modelo a testar nas ações de negócio, realizou-se o processo de inferência. Por outras palavras, atribuiu-se o *output* do modelo escolhido a uma base de clientes, da qual se desconhece a variável resposta. É também, nesta fase, que, de acordo com a probabilidades obtidas, é atribuído um *ranking* a cada contacto de um cliente para, posteriormente, serem testados.

Na fase de experimentação, pretende-se avaliar o comportamento do modelo em campanhas de *telemarketing*. Sendo assim, é uma fase de extrema relevância, uma vez que se irá analisar o possível incremento do modelo na operação. Neste sentido, é necessário planear de forma pormenorizada o piloto que se pretende testar, nomeadamente, definir o segmento de controlo e os restantes, a dimensão do lote de contactos que se pretende enviar, quantos contactos de um cliente se pretendem enviar para a operação por segmento e definir os KPI a ser avaliados, durante o ciclo de campanha (por exemplo, taxa de atendimento e de decisões, total de chamadas efetuadas, etc).

Relativamente à avaliação de resultados, esta deve ocorrer diariamente, durante o ciclo em que a campanha está em vigor. Desta forma, é possível ter a perceção, em tempo real, de qual foi a evolução dos resultados de cada segmento do piloto, durante o ciclo. Aliás, como estamos a lidar, diretamente, com vendas e receitas, é importante monitorizar a eficiência do lote de contactos enviado, de forma a detetar quaisquer problemas que possam surgir (por vezes é necessário suspender a campanha para não prejudicar as vendas).

Após terminar o ciclo de campanha, é necessário interpretar os resultados e perceber se o modelo correspondeu aos KPI que se pretendiam medir. Caso os resultados não correspondam ao expectável, pode ser necessário repensar o planeamento do piloto. Por sua vez, se os resultados obtidos forem muito abaixo do esperado, pode ser necessário redefinir todo o âmbito do problema.

Em contrapartida, se os resultados obtidos na experimentação forem positivos, segue-se a recomendação. Nesta fase, de acordo com os resultados da operação e o *output* do modelo, recomenda-se o número de contactos por cliente, assim como se encontra o ponto de corte a partir do qual se justifica enviar os contactos para campanha. Além do mais, pode ser necessário recomendar as regras que ditam a ordem e tentativas de marcação dos contactos na operação, de modo a estarem alinhadas com o modelo analítico desenvolvido.

Este processo termina com a implementação do produto analítico como modelo de recomendação de contactos para campanhas de *telemarketing*. De forma análoga, significa que o modelo entra em

produtização na empresa. Assim, as principais etapas que fizeram parte do projeto encontram-se resumidas no esquema da Figura 4.3.

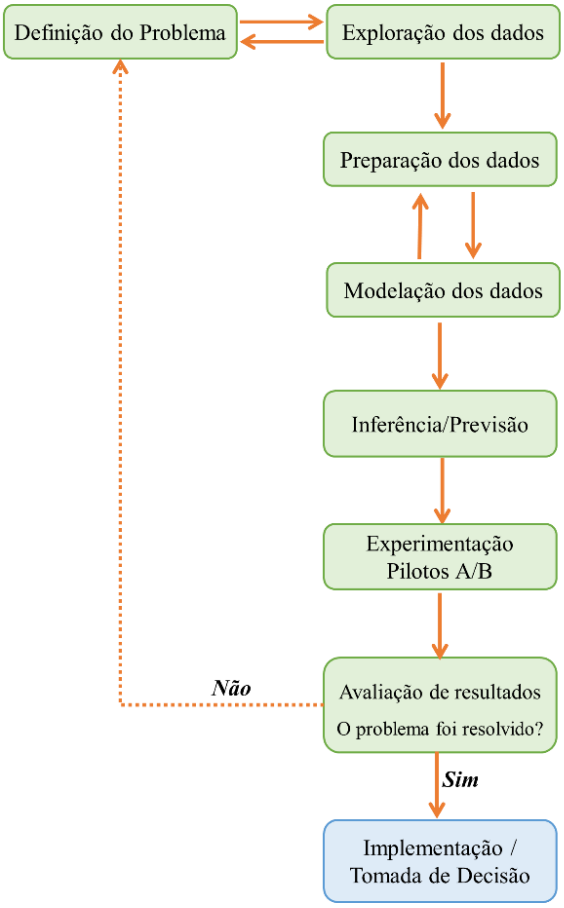

Figura 4.3 - Esquema das principais etapas do projeto

Todavia, é ainda fundamental proceder à descrição do processo analítico seguido durante a modelação dos dados. Com efeito, foi nesta fase que se aplicaram as metodologias referidas no Capítulo 3 – Enquadramento Teórico, como é possível observar no esquema da Figura 4.4.

Neste sentido, a primeira etapa do processo analítico consistiu na análise exploratória do painel de variáveis. Assim, o objetivo consistiu em analisar as principais caraterísticas dos dados, tais como a dimensão do painel, tipos de variáveis, existência de valores omissos e duplicados. Além disso, a intenção era também estudar as tendências ou padrões existentes nos dados, como relações entre variáveis independentes e entre estas e a variável resposta, distribuição da variável resposta e algumas estatísticas básicas.

Uma vez conhecidos os dados, separaram-se as variáveis independentes da variável resposta a estudar[7] e, posteriormente, dividiu-se o painel em três segmentos: treino, validação e teste. Tal como foi descrito na secção 3.5, a divisão dos dados em diferentes segmentos representa uma boa prática, no que diz respeito à validação da performance de um modelo.

[7] É de salientar que a metodologia seguida na fase de modelação dos dados foi análoga para cada uma das situações de variável resposta e para cada algoritmo de modelação que se testou.

Posteriormente, foi necessário efetuar o pré-processamento de dados, visto que o painel apresentava variáveis com valores omissos, variáveis categóricas e ainda variáveis com configurações inadequadas. Neste sentido, foram aplicadas uma série de transformações aos dados, de modo a tornar todas as variáveis e observações legíveis, para o modelo que se pretendia implementar.

Os procedimentos seguidos consistiram em tratar as configurações inapropriadas de algumas variáveis, nomeadamente, substituir vírgulas por pontos e remover colunas que não se pretendiam que entrassem para o estudo (por exemplo, identificação e descrição de campanha, ciclo, datas, etc). Para além disso, procedeu-se à transformação das variáveis categóricas em numéricas, através da criação de variáveis *dummy*. Tal como foi referido na secção 4.1, existiam apenas duas variáveis categóricas, uma com 3 e a outra com 2 classes, desta forma, a codificação de variáveis deu origem a 5 novas variáveis binárias no painel.

Ainda na etapa de pré-processamento de dados, procedeu-se ao preenchimento dos valores omissos pela mediana da respetiva variável. No entanto, nas variáveis de tráfego, os valores em falta foram preenchidos por 0, uma vez que quando não se possui registo de tráfego de um contacto, considera-se que este não teve atividade, no intervalo temporal considerado. Por fim, dado que o painel era composto por um elevado número de variáveis independentes, recorreu-se a um algoritmo de seleção de variáveis, neste caso, o Boruta mencionado na secção 3.1, executado para 100 iterações.

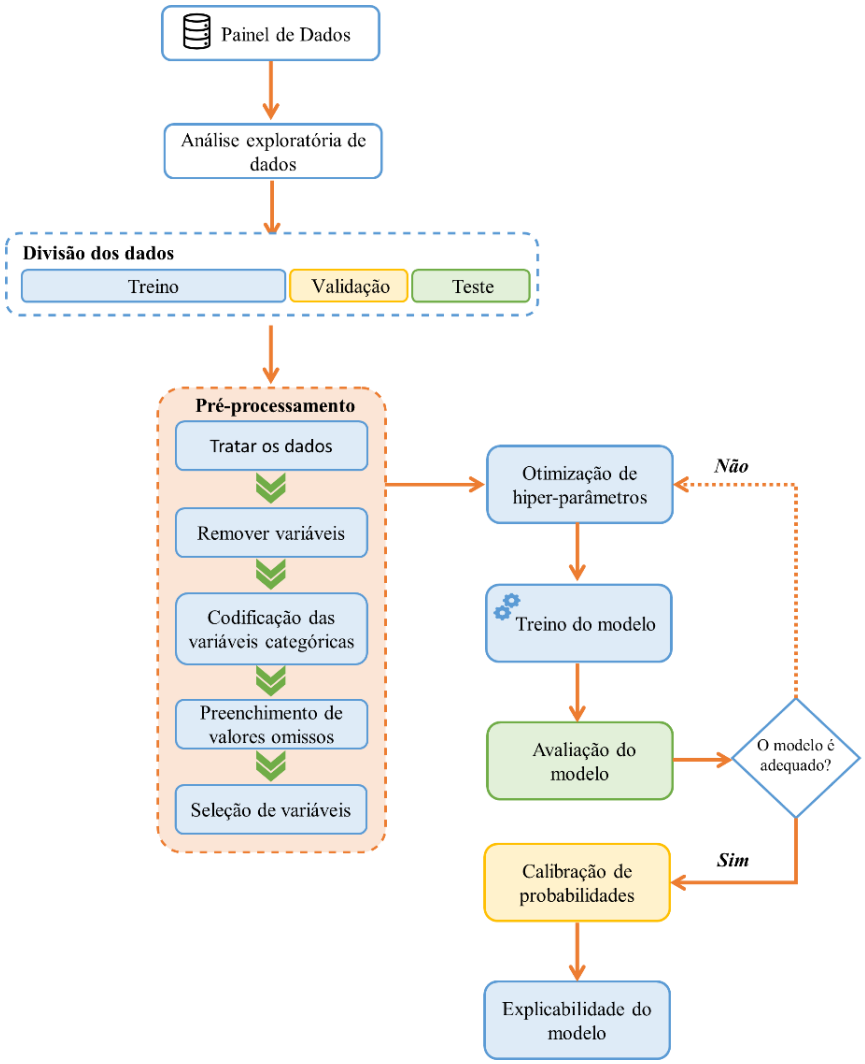

Figura 4.4 - Esquema das etapas do processo analítico seguido modelação dos dados

A fim de encontrar o melhor conjunto de hiper-parâmetros, para o modelo que se pretendia testar, recorreu-se a uma metodologia de otimização Bayesiana para 100 repetições, com recurso a *5-fold cross-validation*, onde a métrica que se pretendia maximizar era a AUROC. Sendo assim, o passo seguinte consistiu no treino de um algoritmo de modelação de dados, neste contexto, o *Random Forest* ou o *Gradient Boosting*, com o respetivo conjunto de hiper-parâmetros, encontrado na etapa anterior.

Após ser efetuado o treino do modelo, seguiu-se a avaliação da sua performance, com recurso à curva de ROC, matriz de confusão e a outras métricas de avaliação, anteriormente mencionadas. Note-se que é nesta etapa que é possível identificar se o modelo é suscetível a *overfitting* (efetuando uma comparação entre as métricas de avaliação obtidas para o conjunto de treino e teste) e se as métricas de avaliação obtidas correspondem às necessidades.

Deste modo, se o modelo não for adequado, é necessário regressar ao passo de otimização de hiper-parâmetros. Esta é uma situação bastante frequente, pelo que, para a solucionar, é essencial, experimentar diferentes espaços de valores ou novos tipos de hiper-parâmetros, de modo a melhorar a sua otimização. Por outro lado, se o modelo treinado for considerado apropriado, deve prosseguir-se para a verificação da calibração de probabilidades. Isto é, verificar se as probabilidades devolvidas pelo modelo se encontram calibradas, de forma correta.

Por fim, o último passo do processo analítico, seguido na fase de modelação de dados, consistiu na avaliação da explicabilidade do modelo, com recurso aos valores de SHAP, referidos na secção 3.6. Sendo assim, avaliaram-se as variáveis que foram consideradas mais importantes pelo modelo, bem como a sua contribuição para a classificação das observações.

4.3. Implementação Computacional

O modelo de previsão do contacto decisor, foi implementado com recurso à ferramenta de programação Python. Esta linguagem surgiu no início dos anos 90, como sucessora de uma linguagem designada de ABC e foi criada por Guido van Rossun, no Stichting Mathematisch Centrum, situado nos Países Baixos. Em 2001, foi formada a Python Software Foundation, uma organização sem fins lucrativos concebida, especificamente, para deter a propriedade intelectual associada a esta linguagem [25].

O Python, é caraterizado por possuir estruturas de dados de alto nível, bastante eficientes e uma abordagem eficaz no que diz respeito à programação orientada por objetos. Aliás, esta linguagem destaca-se de outras, por apresentar facilidade de aprendizagem e compreensão e, ainda, por oferecer múltiplas possibilidades de desenvolvimento. Com efeito, dispõe de uma sintaxe simples e clara e disponibiliza um conjunto de bibliotecas devidamente estruturadas. É ainda de salientar, a facilidade que apresenta para a criação de scripts e versatilidade de desenvolvimento em diversas áreas e plataformas.

Neste sentido, durante as etapas de exploração, análise e modelação de dados do projeto, algumas das principais bibliotecas de Python utilizadas foram as seguintes:

- **Pandas** – Ferramenta bastante flexível, no que toca à análise e manipulação de dados;
- **Matplotlib e Seaborn** – Bibliotecas dedicadas a metodologias de visualização de dados em Python;

- **Scikit-learn** – Fornece uma variedade de ferramentas simples e eficientes, de análise preditiva de dados;
- **LightGBM** – Biblioteca que fornece estruturas para treino de *Gradient Boosting*, através de algoritmos de aprendizagem, baseados em árvores de decisão;
- **Optuna** – Ferramenta concebida, especificamente, para ML, que tem como finalidade a otimização automática de hiper-parâmetros;
- **Shap** – Disponibiliza diversas possibilidades de visualização e interpretação para explicar os resultados de qualquer modelo de ML;

Capítulo 5 – Resultados e Avaliação dos Modelos

Este capítulo diz respeito à análise e explicação dos resultados obtidos durante o projeto. Sendo assim, primeiramente será apresentada a análise exploratória efetuada aos dados, onde é possível encontrar algumas das tendências detetadas nos mesmos. De seguida, encontram-se os conjuntos de hiper-parâmetros utilizados para o treino dos algoritmos de ML testados, neste caso, o *Random Forest* e o *Gradient Boosting*, bem como a análise comparativa dos resultados obtidos em ambos. Por último, são apresentados os resultados atingidos na fase de experimentação do modelo, nas ações do negócio e a respetiva recomendação final.

É importante referir, que, neste capítulo, o foco da análise diz respeito apenas à variável resposta binária da situação 2, descrita na secção 4.1, uma vez que foi a escolhida para prosseguir no estudo durante a modelação dos dados.

5.1. Análise Exploratória de Dados

De acordo com os dados do painel de variáveis deste estudo, em média um cliente/empresa tem associados 4 contactos, sendo que o valor mais elevado que se regista é de cerca de 105 contactos. Desses, a maioria corresponde a contactos móveis (83,7%), isto é, iniciados pelo algarismo 9 e cerca de 16% a contactos fixos (iniciados por 2). As restantes observações correspondem a minorias de contactos iniciados por outros algarismos. Nota-se, ainda, que cerca de 62% dos clientes têm associado um contacto principal na ficha de cliente.

Relativamente aos serviços que os clientes em estudo possuem, tal como é possível observar na Figura 5.1, é percetível que o serviço com maior adesão é a Voz Móvel (32,12% dos serviços). Por sua vez, 21,98% dos serviços dos clientes são de Voz Fixa, 19,56% de Internet Fixa e 17,09% de Televisão. Por último, o serviço que apresenta menor adesão é a Internet Móvel (9,25% dos serviços). Ainda, que as empresas que fazem parte do estudo possuem, em média, 5 serviços de telecomunicações.

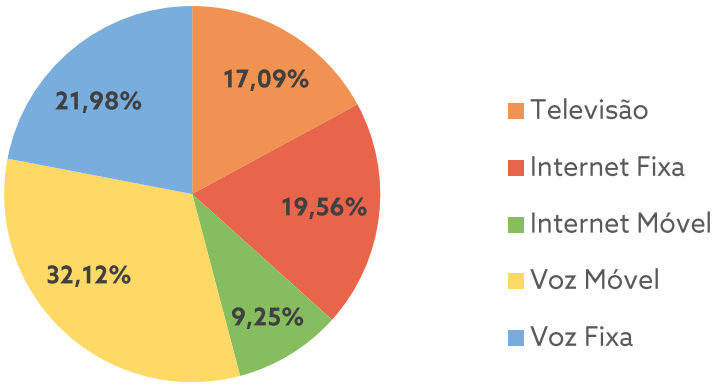

Figura 5.1- Gráfico que representa a adesão dos clientes do estudo, a cada tipo de serviço

Segundo as variáveis que dizem respeito à telefonia, verifica-se que, num histórico de 6 meses, a taxa média de atendimento dos contactos deste estudo é de 88%. Isto significa que, nesse espaço de tempo, em média, os contactos atendem 88% das chamadas que a empresa de telecomunicações em

questão lhes direciona. Observa-se, ainda, que em média são necessárias 2 tentativas de contacto, por ciclo, para se obter uma resposta de telefonia por parte de um contacto.

Por sua vez, num intervalo temporal de 6 meses, verifica-se que um contacto efetua e recebe, em média, 13 chamadas do SAC, sendo que estas, na sua maioria, não ficam com tópicos associados (uma chamada para o SAC, pode ocorrer apenas para pedir ou transmitir informações).

No respeitante ao tráfego dos contactos do painel, para um intervalo temporal de 2 meses, em média, um contacto efetua e recebe chamadas em cerca de 21 dias. Ainda, nesse espaço de tempo, um contacto tem uma atividade média, de cerca de 416,77 minutos. Para além disso, observa-se que o tráfego dos contactos ocorre, principalmente, entre as 12h e as 19h, durante os dias úteis.

Tendo ainda em conta a informação relativa ao tráfego dos contactos, nos 2 meses de histórico considerados, apresentam-se na Figura 5.2 as correlações de Spearman entre algumas delas.

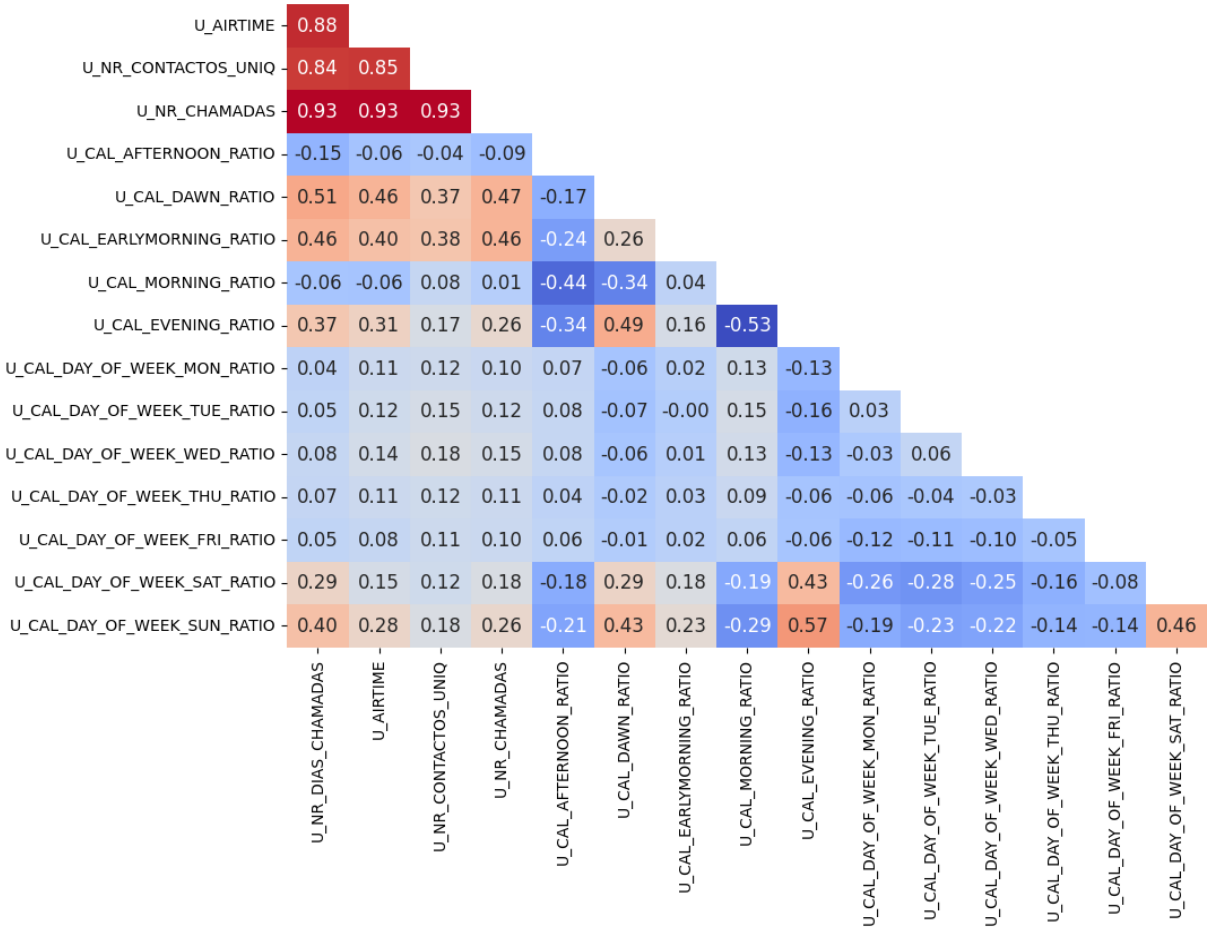

Figura 5.2 –Correlações de Spearman entre algumas variáveis de tráfego presentes no painel

Neste sentido, verifica-se que a correlação mais elevada é de 0,93 e diz respeito ao número total de chamadas efetuadas e recebidas, com as seguintes variáveis: o número de dias em que os contactos registam atividade; a soma do tráfego recebido e efetuado; e o número distinto de contactos para os quais um contacto apresenta chamadas, nos 2 meses de histórico considerados. A correlação existente entre estas variáveis é considerada forte, pelo que se conclui que os contactos que apresentam maior número de chamadas, efetuadas e recebidas (U_NR_CHAMADAS), estão associados a contactos com

maiores valores das restantes três variáveis referidas (U_NR_DIAS_CHAMADAS, U_AIRTIME, U_NR_CONTACTOS_UNIQ).

Observa-se, ainda, que existe uma correlação elevada entre a soma (em minutos) do tráfego recebido e efetuado pelos contactos e o número de dias em que estes efetuam ou recebem chamadas, nos 2 meses de histórico considerados (U_AIRTIME e U_NR_DIAS_CHAMADAS, respetivamente). A correlação de 0,88 entre ambas expressa que o aumento da soma do tráfego, recebido e efetuado, incrementa o número de dias em que os contactos registam atividade de tráfego.

Por fim, observa-se uma correlação moderada negativa (-0,53) entre o rácio de tráfego ocorrido no início da manhã e o rácio de tráfego ocorrido ao final do dia (U_CAL_MORNING RATIO e U_CAL_EVENING_RATIO, respetivamente). Isto significa que os contactos que apresentam maior atividade no início da manhã (entre as 8 e as 12 horas) estão associados a contactos que apresentam menor atividade ao fim do dia (entre as 19 horas e as 23 horas).

No que se refere à variável resposta, como se encontra representado na Figura 5.*3*, distribui-se da seguinte forma: em 81,34% dos eventos não se verifica decisão (0- insucesso sem oferta apresentada, não contactado, cancelado, *callback* com oferta apresentada ou *outcome* não comercial), por outro lado, em 18,66% ocorre decisão (1- sucesso ou insucesso com oferta apresentada).

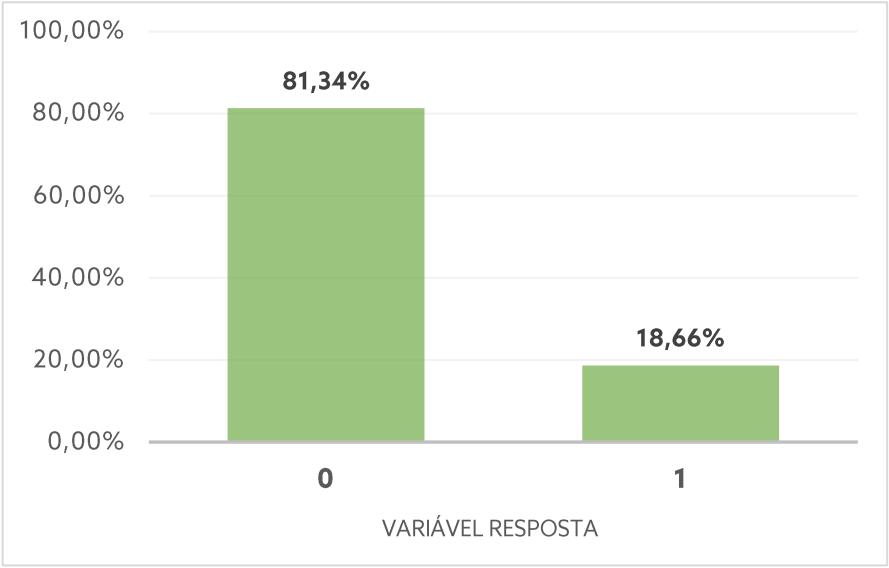

Figura 5.3- Distribuição da variável dependente, com a qual se avançou no estudo

Neste sentido, verifica-se, ainda, que, relativamente ao intervalo temporal a que o estudo se refere, durante um ciclo de campanha, a taxa de decisão por cliente, em média, é cerca de 30%. Por sua vez, a taxa de decisão por chamada é de cerca de 18%. Por outras palavras, significa que durante um ciclo, em cerca de 18% das chamadas se chega à fala com os contactos decisores das empresas.

Uma vez estudadas as principais tendências e padrões existentes nos dados, reuniram-se as condições necessárias para iniciar a aplicação de algoritmos de modelação aos mesmos.

5.2.Modelação

Tendo por base a metodologia descrita na secção 4.2, seguida durante o processo analítico, verificou-se que após a execução do algoritmo de seleção de variáveis (Boruta) para a variável resposta considerada neste capítulo, foram selecionadas 51 variáveis explicativas[8]. Com efeito, reduziu-se a dimensão do painel de 168 variáveis independentes (dimensão do painel com a inclusão das variáveis *dummy*) para 51, sendo que foi nas variáveis de tráfego onde se notou a maior redução. Note-se que, esta era uma situação expectável, visto que, tal como foi analisado anteriormente, algumas das variáveis de tráfego eram bastante correlacionadas entre si. Desta forma, estas variáveis conseguem explicar-se umas às outras, pelo que não é necessário serem utilizadas, na sua totalidade, para o treino do algoritmo.

De seguida, apresentam-se os espaços de valores que originaram o conjunto ótimo de hiper-parâmetros, a ser utilizado para o treino de cada modelo testado. Sendo assim, relativamente ao algoritmo *Random Forest*, o intervalo de valores definido para a otimização Bayesiana encontra-se a seguir descrito:

- Número de árvores de decisão a ajustar – entre 50 e 300;
- Profundidade máxima de cada árvore de decisão – entre 2 e 25;
- Número de variáveis independentes analisadas para encontrar a melhor partição – entre 1 e 51;
- Número mínimo de observações necessárias para a partição de um nodo interno– entre 2 e 80;
- Número mínimo de observações num nodo-folha – entre 1 e 60;

Uma vez efetuada a otimização dos hiper-aparâmetros, constatou-se que o conjunto que otimizou a AUROC do algoritmo *Random Forest* foi o seguinte: 269 árvores de decisão a ajustar; profundidade máxima de 19 nodos em cada árvore de decisão; 28 variáveis explicativas a analisar, a fim de encontrar a melhor partição, e, por último, são necessárias, no mínimo, 42 observações para a partição de cada nodo interno e 38 observações em cada nodo-folha. É ainda importante referir que o critério de avaliação utilizado para a partição de cada nodo foi o índice de Gini e que se recorreu ao método de *bootsrap* para a amostragem de observações.

No que se refere ao algoritmo do *Gradient Boosting*, os intervalos de valores definidos para a procura do conjunto de hiper-parâmetros ótimo são descritos de seguida:

- Número de árvores de decisão a ajustar – entre 50 e 300;
- Número máximo de nodos folha numa árvore de decisão – entre 2 e 50;
- Profundidade máxima de cada árvore de decisão – entre 2 e 30;
- Número mínimo de observações num nodo-folha – entre 50 e 300;
- Fração de variáveis a selecionar antes de cada iteração para analisar – entre 0,5 e 1;
- Parâmetro responsável por dimensionar a contribuição de cada árvore de decisão – entre 0,001 e 0,1;

Neste sentido, verificou-se que o conjunto de hiper-parâmetros que maximizou a AUROC do algoritmo *Gradient Boosting* foi o seguinte: 248 árvores de decisão a ajustar; um máximo de 49 nodos-folha em cada árvore de decisão; profundidade máxima de 26 nodos em cada árvore; 59 observações

[8] É possível consultar no Anexo B a totalidade das variáveis selecionadas pelo Boruta, para o treino dos modelos

no mínimo em cada nodo-folha; seleção cerca de 88% das variáveis explicativas antes do treino do modelo e, por fim, a contribuição de cada árvore de decisão para o *ensemble* é de 0,033.

5.3. Análise Comparativa de Modelos

De seguida, procede-se à apresentação dos resultados das métricas de avaliação[9] obtidos para os dados de treino e teste para cada um dos algoritmos de modelação testados, neste caso, o *Random Forest* e o *Gradient Boosting.*

Os valores das métricas de avaliação relativas ao *Random Forest* são apresentados na Tabela 5.1. Assim sendo, ao observar os valores que constam na tabela referida, verifica-se que, no geral, os valores das métricas obtidas para o conjunto de dados de treino são superiores às dos dados de teste. Esta situação é justificada pelo facto do algoritmo ter aprendido a classificar as observações de acordo com as variáveis existentes nos dados de treino. No entanto, os valores obtidos nos dois conjuntos de dados não apresentam uma discrepância significativa entre si, logo, não se considera que tenha ocorrido *overfit* nos dados.

No que diz respeito à exatidão das previsões que o modelo originou, constata-se que este é capaz de classificar corretamente cerca de 82% das observações dos dados de teste. Dado que a percentagem de erro de um modelo é complementar à sua exatidão, verifica-se que o modelo classifica de forma incorreta cerca de 18% das observações dos dados de teste. Observa-se, ainda, que, nos dados de teste, o modelo prevê de forma correta, cerca de 59% das observações que foram classificadas como positivas. Contudo, relativamente às observações que na realidade pertencem à classe positiva da variável resposta, o modelo apenas identifica cerca de 13% corretamente.

Com efeito, observa-se que a precisão das previsões positivas do modelo é superior ao da sensibilidade, pelo que o modelo origina uma maior quantidade de FN, do que FP nas suas previsões. Isto indica que, de forma geral, o modelo tem maior facilidade em identificar uma observação da classe positiva da variável dependente como sendo da classe negativa, do que identificar uma observação da classe negativa como sendo da positiva.

Tabela 5.1- Valores das métricas de avaliação obtidos no *Random Forest,* nos dados de treino e teste

Métricas \ Dados	**Treino**	**Teste**
Acurácia	0,84	0,82
Precisão	0,78	0,59
Sensibilidade	0,18	0,13
F1 Score	0,29	0,21
AUROC	0,86	0,76

Assim como foi referido na secção 3.2.6, a AUROC traduz o quão acertada é a capacidade de um modelo distinguir as duas classes da variável resposta, ou seja, representa a medida de separabilidade das classes, pelo que, quanto mais próxima for da unidade melhor. Na Figura 5.4, encontram-se

[9] Nos Anexos C, D e E é apresentada, de forma detalhada, os resultados obtidos para as restantes formulações de variável resposta testadas.

representadas as curvas de ROC obtidas para os dados de treino e teste, bem como as respetivas AUROC. Assim, tal como é possível observar na figura mencionada, não é percetível uma discrepância significativa em ambas as curvas e entre os valores de AUROC obtidos em ambos os conjuntos de dados. Desta forma, o valor de 0,76 de AUROC obtido para o conjunto de dados de teste, mostra que o modelo tem capacidade para distinguir, de forma correta, as duas classes que constituem a variável resposta.

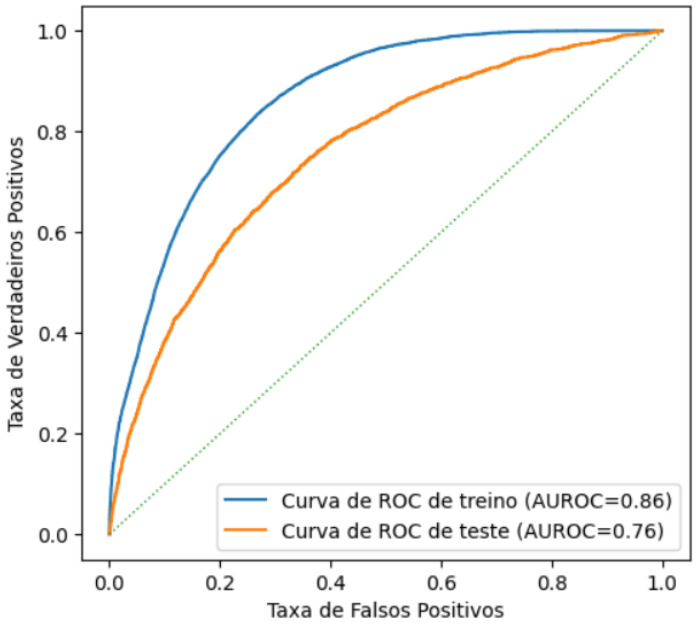

Figura 5.4 - Curva de ROC e respetiva AUROC do algoritmo *Random Forest*

No que se refere ao *Gradient Boosting*, os valores das métricas de avaliação são apresentados na Tabela 5.2. Ao observá-la, verifica-se, uma vez mais, que os valores obtidos são superiores para os dados de treino, pelo mesmo motivo, anteriormente mencionado. Analogamente ao que se observou no *Random Forest*, no *Gradient Boosting* também não se verifica uma discrepância significativa entre as métricas obtidas para os dois conjuntos de dados, pelo que não se considera que tenha ocorrido *overfitting*.

Neste caso, relativamente à exatidão das previsões do modelo, verifica-se que este classifica corretamente cerca de 82% das observações presentes nos dados de teste, pelo que apresenta uma percentagem de erro de classificação de 18%. Relativamente à precisão do modelo, nota-se que, das observações previstas como positivas, cerca 59% foram classificadas de forma assertiva. Porém, nas observações dos dados de teste, que são verdadeiramente da classe positiva da variável dependente, verifica-se que o modelo apenas acerta em 16%.

Também nesta situação é percetível que o modelo tem maior tendência em classificar uma observação que é decisor, como não sendo, do que o contrário. Com efeito, constata-se que o valor da precisão das previsões positivas é superior ao da sensibilidade, pelo que se conclui que o modelo origina uma quantidade de FN superior à de FP, o que explica o facto de o valor da sensibilidade ser mais reduzido.

Tabela 5.2-Valores das métricas de avaliação obtidos no *Gradient Boosting*, nos dados de treino e teste

Métricas \ Dados	**Treino**	**Teste**
Acurácia	0,84	0,82
Precisão	0,74	0,59
Sensibilidade	0,21	0,16
F1 Score	0,33	0,25
AUROC	0,86	0,77

Na Figura 5.5, encontram-se representadas as curvas de ROC obtidas para os dados de treino e teste, bem como as respetivas AUROC. Assim, neste caso, também é possível verificar que não existe uma discrepância significativa entre ambas as curvas e entre os valores de AUROC obtidos. Assim, o valor de 0,77 de AUROC obtido para o conjunto de dados de teste demonstra que o modelo consegue distinguir, de forma correta, as duas classes que constituem a variável dependente.

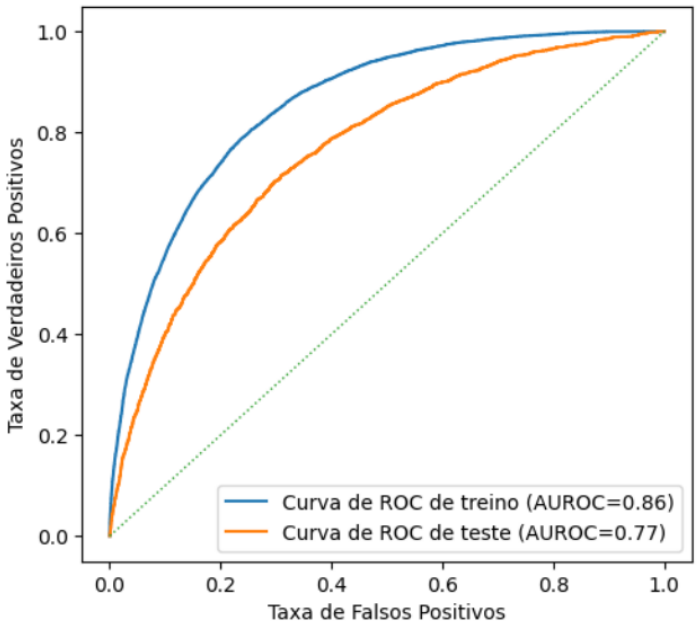

Figura 5.5- Curva de ROC e respetiva AUROC do algoritmo *Gradient Boosting*

É importante salientar que, como foi referido na secção 3.4, após ser ajustado um modelo aos dados e, posteriormente, ser efetuada a avaliação do desempenho do mesmo, se deve proceder à verificação da qualidade das probabilidades previstas. Assim, para ambas as metodologias testadas, observou-se que as probabilidades previstas se encontravam devidamente calibradas, não sendo necessário nenhum ajustamento.

Assim sendo, efetua-se, de seguida, uma análise comparativa entre os resultados obtidos em ambos os modelos. Neste sentido, através da observação da Tabela 5.1 e da Tabela 5.2, é possível verificar que, na sua generalidade, os valores das métricas de avaliação referentes ao *Gradient Boosting* são iguais ou superiores aos que se referem ao *Random Forest*. Nomeadamente o $F_1 score$ que, tal como foi referido na secção 3.2.6, é uma métrica de avaliação utilizada para a comparação de classificadores. Com efeito, verificou-se que tanto para a formulação da variável resposta tida em consideração neste capítulo, como para as restantes formulações testadas durante este estudo (os resultados encontram-se

detalhados nos Anexos C, D e E), o algoritmo *Gradient Boosting* originou sempre um valor de $F_1 score$ superior ao do *Random Forest*.

No que diz respeito ao valor da AUROC obtido para ambos os algoritmos, observa-se que, apesar da diferença entre ambos ser muito reduzida, aquele que apresentou um valor desta métrica mais próximo da unidade foi o *Gradient Boosting*. Assim, conclui-se que este modelo apresenta uma ligeira melhoria no que toca à distinção das classes da variável dependente.

Além de ser desejável construir um modelo com bom desempenho, ou seja, resultados satisfatórios nas métricas de avaliação, é de extrema importância desenvolver um produto analítico eficiente. Desta forma, para análise comparativa de modelos foram consideradas algumas medidas de utilização de recursos computacionais. Assim sendo, na Figura. 5.6 encontram-se representadas as medidas de utilização de recursos computacionais consideradas para este estudo, sendo que para cada algoritmo essas medidas correspondem à média de 5 execuções.

Neste sentido, é percetível que o *Random Forest* seja o algoritmo que necessita um maior intervalo de tempo para ser executado. De facto, nota-se que este demora cerca de 8 vezes mais do que o *Gradient Boosting*. É importante salientar que na figura apenas se encontra apresentado o tempo de execução de cada modelo, no entanto, quando é necessário efetuar otimização de hiper-parâmetros, este processo requer horas para a sua execução, mantendo a mesma proporção.

No que se refere à alocação de memória durante execução de cada uma as metodologias testadas, verifica-se que o *Random Forest* é aquele que apresenta maior valor. Efetivamente, constata-se que ambos os algoritmos apresentam uma diferença bastante significativa no que diz respeito à alocação de memória, necessária para a sua execução. Por fim, no que concerne à capacidade de processamento que cada modelo necessita para o seu funcionamento, verifica-se que o *Gradient Boosting* se destaca, no entanto, ambos apresentam valores semelhantes desta medida.

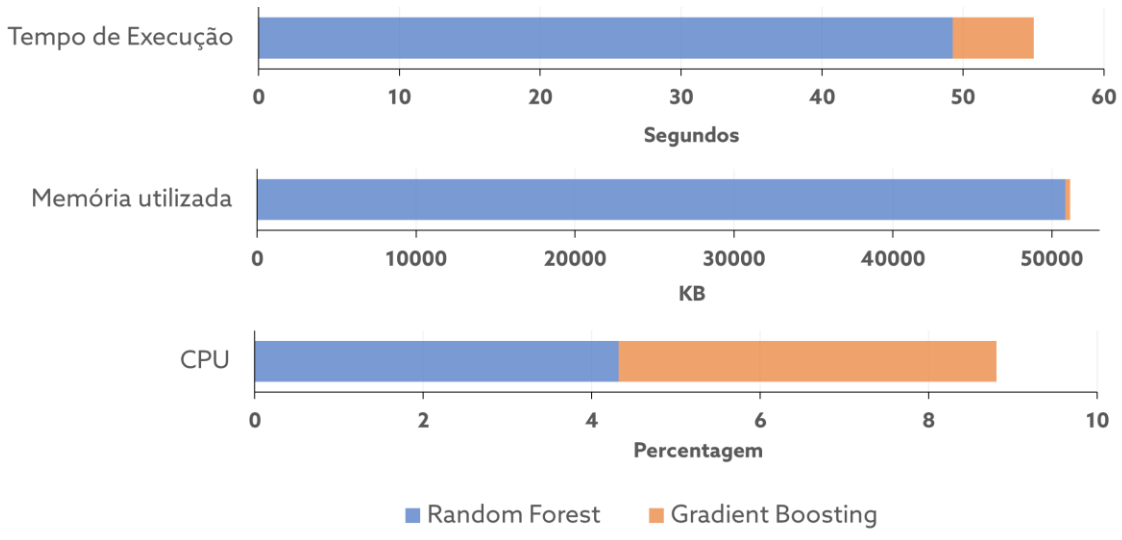

Figura. 5.6- Gráfico que apresenta a utilização de recursos computacionais de cada algoritmo testado

Tendo em conta os resultados das métricas de avaliação de cada algoritmo e as medidas de utilização de recursos computacionais mencionadas, conclui-se que o modelo que apresentou melhor desempenho foi o *Gradient Boosting.* Efetivamente, observa-se que este modelo exibiu maior valor de AUROC e que revelou ser mais eficiente em termos computacionais. Por estes motivos, o *Gradient Boosting*, com

a variável resposta considerada neste capítulo, foi o algoritmo eleito para prosseguir no projeto. De seguida apresentam-se os resultados da explicabilidade do modelo selecionado.

De acordo com Figura 5.7, as 15 variáveis consideradas mais importantes para classificar os contactos de um cliente, numa das duas classes da variável resposta, são: o total de dias que restam para terminar a fidelização de um cliente; total de chamadas que um contacto efetua ou recebe do SAC, nos 6 meses anteriores; tempo (em minutos), que passou desde a última vez que um contacto foi contactado numa campanha de *telemarketing;* total de chamadas efetuadas para um determinado contacto, nos 6 meses de histórico considerados; rácio entre o tráfego total, para contactos ao domingo, e o tráfego total; rácio entre o tráfego ao final do dia e o tráfego total. As restantes variáveis apresentadas na figura dizem respeito, na sua totalidade, a variáveis de tráfego dos contactos, nos 2 meses de histórico considerados.

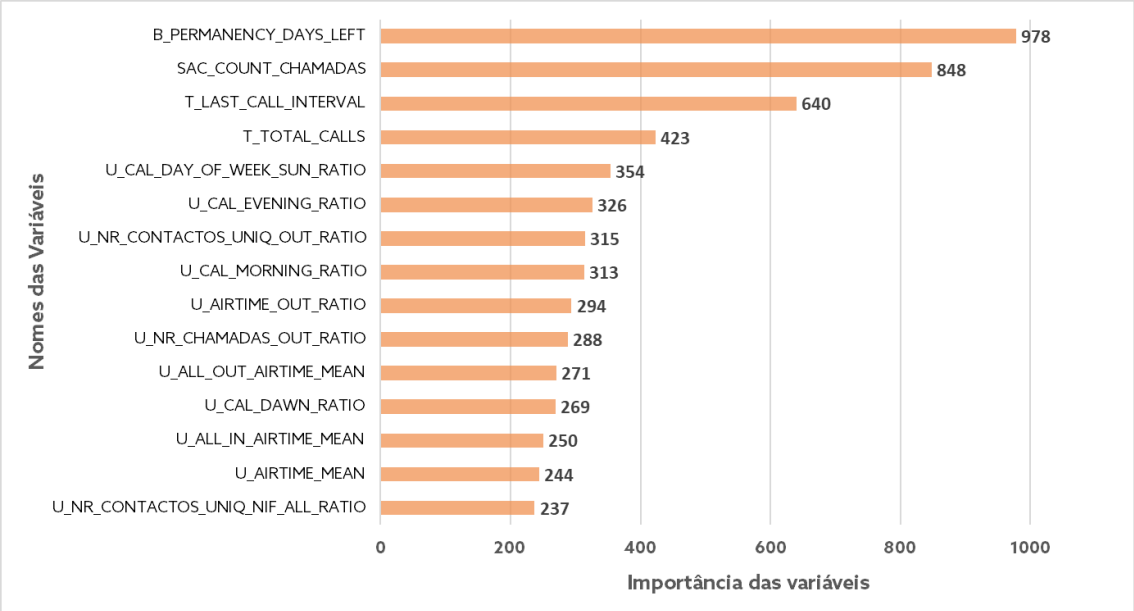

Figura 5.7- Conjunto das variáveis independentes consideradas mais importantes para estimar a variável resposta

Além de identificar as variáveis mais importante para estimar a variável dependente, pretende-se ainda perceber em que medida cada caraterística contribuiu para a previsão final de cada observação. Deste modo, na Figura 5.8 encontram-se representadas as 15 variáveis que apresentaram os valores de SHAP mais elevados, assim como a respetiva contribuição de cada uma delas, no sentido positivo ou negativo da classificação.

Através da observação da Figura 5.8, verifica-se que um contacto assinalado como contacto principal de um cliente e que apresenta menor histórico de chamadas, nos 6 meses de histórico, tem maior propensão a ser decisor. Para além disso, observa-se que um contacto que revele um número de atendimentos reduzido e um elevado número de tentativas de contacto, por ciclo, em campanhas de *telemarketing*, nos 6 meses anteriores, é caraterizado por ser decisor.

Por sua vez, observa-se que um contacto que tenha atividade reduzida com o SAC, é propenso a ser decisor. Em contrapartida, os contactos que apresentam chamadas para o SAC, com um tópico associado, são tendencialmente decisores. Neste sentido, verifica-se, ainda que os contactos decisores são caraterizados por possuírem um elevado número de tópicos associados.

Verifica-se, ainda, que o modelo identifica os contactos móveis como decisores e que os contactos associados a clientes, cujo número de dias para terminar a sua fidelização é elevado, são propensos a serem classificados como decisores. Por fim, nota-se que o modelo identifica com mais facilidade o contacto decisor, quando a empresa de um cliente é mais reduzida (menor número de funcionários e menor número de contactos por NIF).

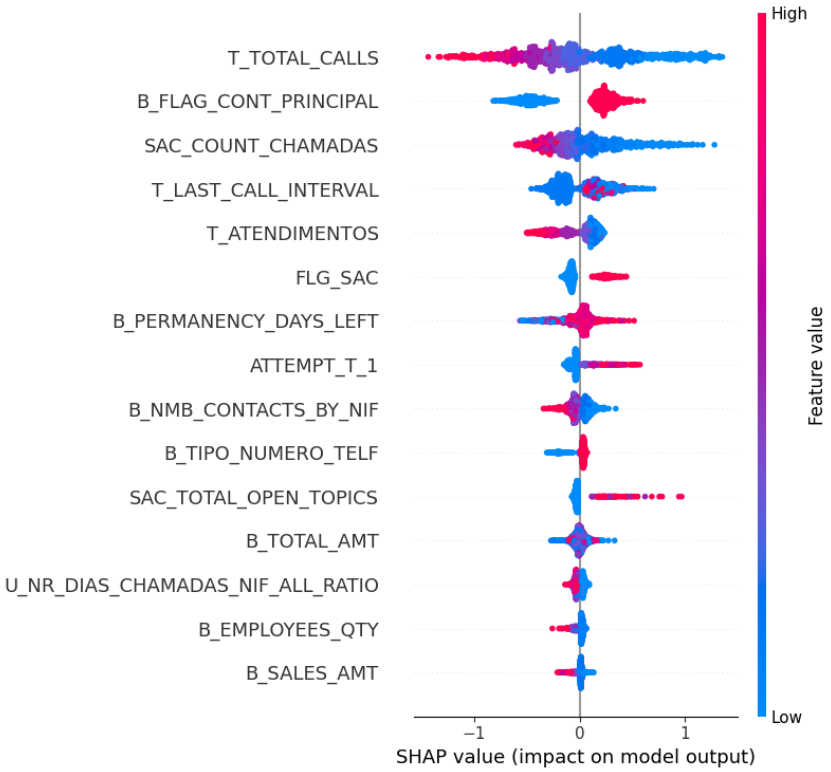

Figura 5.8- Gráfico de SHAP que representa a contribuição das variáveis mais importantes para a classificação

5.4.Tomada de Decisão: Pilotos A/B

Uma vez efetuada a seleção do melhor modelo, realizou-se o processo de inferência para a totalidade da base de dados de clientes do mês de fevereiro, do setor B2B da empresa de telecomunicações. Assim sendo, o passo seguinte consistiu em testar/validar os *outputs* do modelo nos processos de negócio, ou seja, nas campanhas de *telemarketing.*

Neste sentido, organizou-se um piloto com 6926 clientes do setor B2B, dividido em 3 segmentos aleatórios com dimensão aproximadamente igual (2310, 2308, 2308 clientes), para ser testado no ciclo 4 (27 de março até 24 de abril de 2023). Assim, descrevem-se, de seguida, os 3 segmentos de clientes utilizados durante esta campanha:

- **Ficha de cliente**: O primeiro segmento que se considerou ser o grupo de controlo deste piloto, dizia respeito à ficha de cliente. Por outras palavras, utilizaram-se os contactos de um cliente por ordem aleatória, sendo que o contacto principal tinha prioridade. Assim sendo, neste segmento enviaram-se para campanha todos os contactos associados a um cliente.
- **GUC Atende**: O segundo segmento estava associado ao *output* do modelo GUC Atende, descrito na Capítulo **1 – Introdução**. Isto significa que os contactos de um cliente foram classificados de

acordo com o modelo de atendimento de chamadas e, posteriormente, esses contactos foram utilizados na campanha, pela ordem do *ranking* que lhes foi atribuída. Não obstante, neste segmento enviou-se um máximo de 5 contactos por cliente.
É importante frisar que, nestas circunstâncias, apenas se enviam para campanhas contactos com previsão de probabilidade de atendimento acima de 0,13. Assim, para clientes que não tinham nenhum contacto recomendado pelo modelo, utilizou-se o respetivo contacto principal associado.

- **GUC Decide**: O terceiro segmento do piloto dizia respeito aos *outputs* do modelo descrito neste estudo, pelo que os contactos foram enviados para campanha, de acordo com o *ranking* de decisão que lhes foi atribuído. Porém, neste segmento enviaram-se apenas 2 contactos por cliente.

Definiram-se, ainda, alguns KPI, a serem medidos para os 3 segmentos mencionados, durante o decorrer do piloto, com a finalidade de provar que o modelo analítico GUC Decide trazia ganhos de eficiência para o funcionamento das campanhas de *telemarketing*. Através deste piloto pretendia-se provar que, apesar de se disponibilizar um menor número de contactos para o segmento do GUC Decide, se iria conseguir alcançar uma taxa de decisão superior ou semelhante à dos outros segmentos, com menor esforço e utilização de recursos.

Desta forma, os KPI considerados para este piloto foram a taxa de atendimento, total de chamadas efetuadas em cada segmento, total de tentativas de contacto até se obter uma resposta, taxa de decisão e, por fim, total de sucessos comerciais (aceitação de uma proposta). Neste sentido, procede-se de seguida à exposição dos resultados do ciclo 4, relativos ao piloto descrito.

Através da observação da Figura 5.9, é possível compreender a variação da taxa de atendimento por segmento do piloto, durante o decorrer do ciclo 4. Assim, verifica-se que, durante todo o ciclo de campanha, os segmentos associados aos modelos analíticos (GUC Atende e GUC Decide) apresentaram uma taxa de atendimento diária superior à do segmento da ficha de cliente.

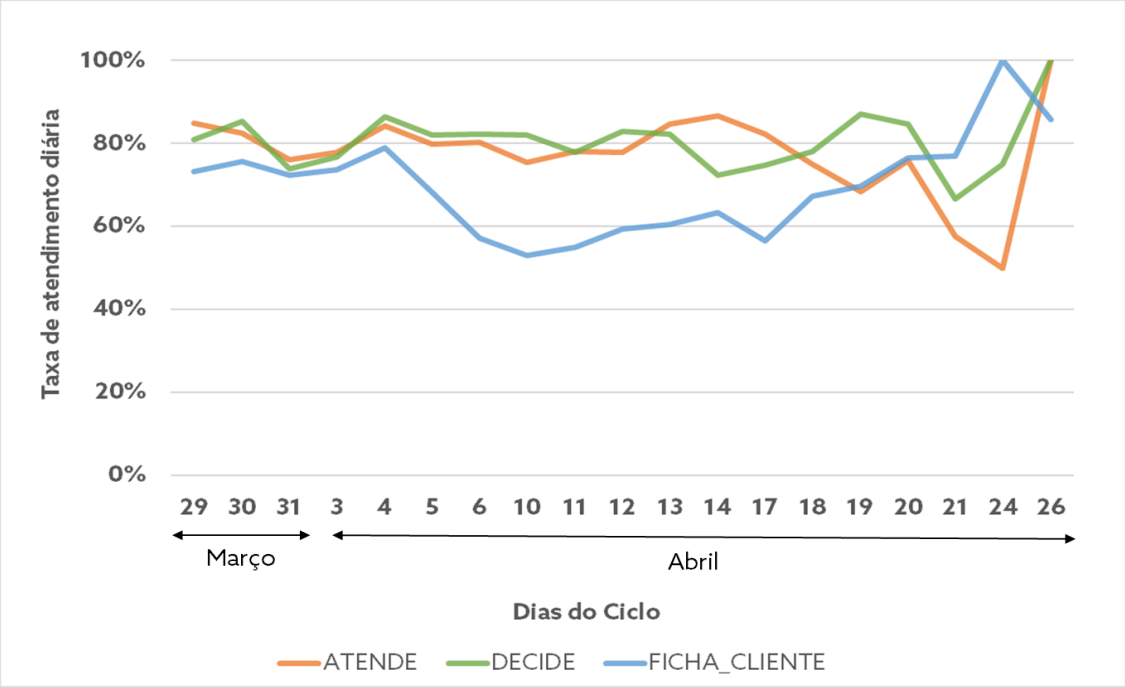

Figura 5.9 – Gráfico que apresenta a taxa de atendimento diária, por segmento, durante o ciclo 4

De facto, é percetível que a taxa de atendimento dos modelos analíticos manteve um comportamento relativamente homogéneo ao longo do ciclo. Em contrapartida, no segmento da ficha de cliente, nota-se que existiu maior variação desta medida no decorrer da campanha, sendo que a diferença é significativamente notória em relação aos restantes. É importante destacar que, entre os dias 20 e 26 de abril, os contactos de cada segmento já tinham sido, na sua maioria, percorridos, sendo que nestes dias se efetuaram apenas alguns *callbacks*, o que influenciou a taxa de atendimento.

Neste sentido, observou-se que a taxa média de atendimento para o segmento da ficha de cliente, no ciclo 4, foi de cerca de 64%, por sua vez, para o GUC Atende foi de 79% e para o GUC Decide de 80%. Deste modo, conclui-se que as recomendações de contactos dos modelos analíticos são significativamente melhores que a ficha de cliente, no que toca ao atendimento de chamadas.

Na Figura 5.10, encontra-se representado o número médio de tentativas de chamada por cliente, durante o ciclo 4. Assim sendo, observa-se que o segmento de contactos da ficha de cliente se destaca, relativamente aos restantes. Por outro lado, percebe-se que para o segmento do GUC Decide são efetuadas menos tentativas de chamada para conversar com o cliente.

Na realidade, observa-se que para o segmento da ficha de cliente foram efetuadas mais tentativas de chamadas por cliente, o que é explicado pelo facto dos contactos serem utilizados de forma aleatória, isto é, sem nenhuma recomendação. De outro modo, significa que são efetuadas chamadas para contactos de um cliente que revelam não ser os mais indicados, o que justifica o valor desta medida ser superior neste segmento do piloto. Por sua vez, no caso dos modelos analíticos verifica-se que o número médio de tentativas para falar com o cliente é mais reduzido, o que indica que a ordem de recomendação dos contactos é benéfica para os processos do negócio.

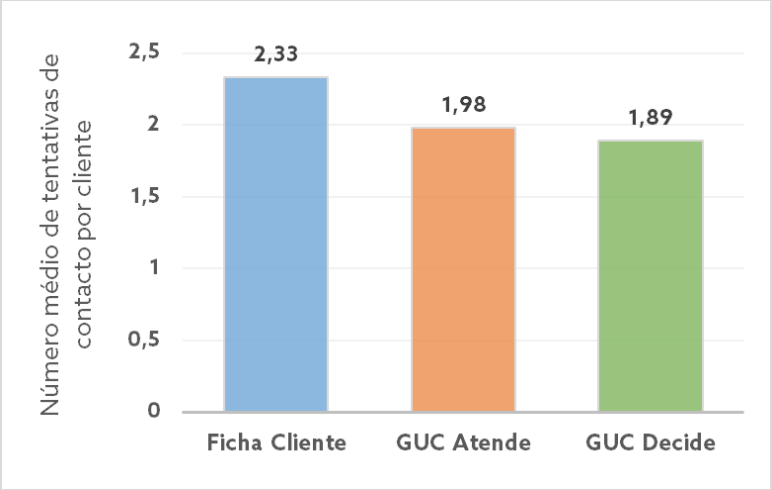

Figura 5.10 – Gráfico que apresenta o número média de tentativas de chamada por cliente, para cada segmento do piloto

No que se refere ao total de chamadas efetuadas em cada um dos segmentos do piloto, durante a campanha no ciclo 4, tal como se encontra representado na Figura 5.11, uma vez mais a ficha de cliente é aquele que apresenta maior valor. Por sua vez, o segmento do GUC Decide é aquele que apresenta menor número de chamadas. Assinala-se que a ficha de cliente apresenta um aumento de cerca de 28% de chamadas, relativamente ao segmento do GUC Decide. Uma vez mais, este é um indicador de que os modelos analíticos, principalmente o GUC Decide, garantem ganhos eficiência na operação.

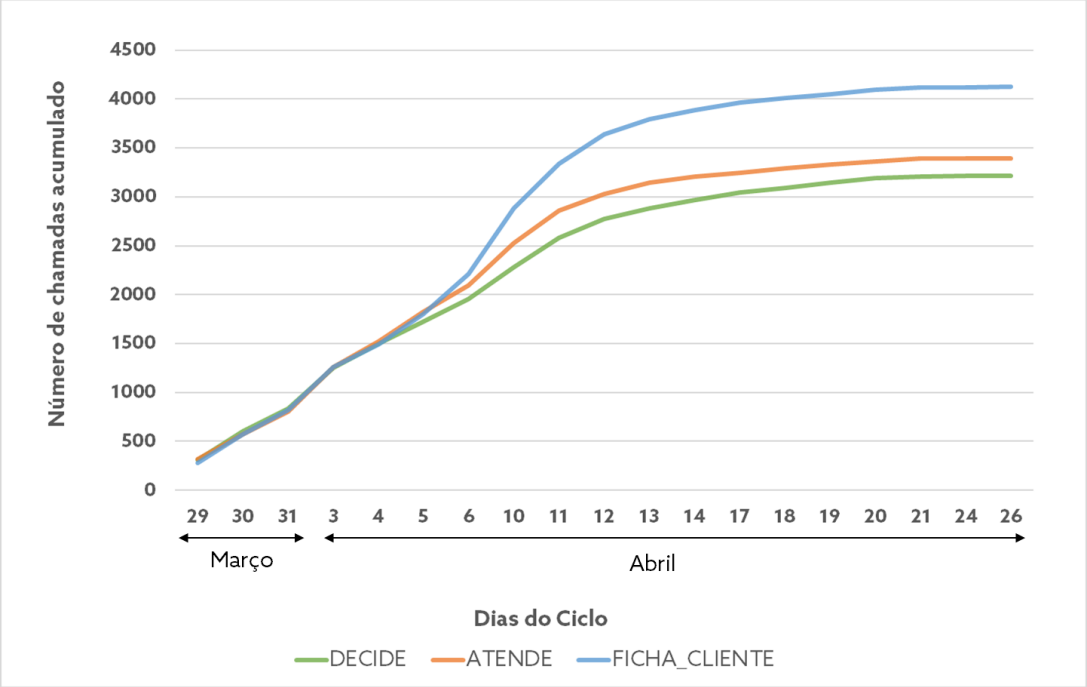

Figura 5.11- Número de chamadas acumulado para cada segmento do piloto, no ciclo 4

No entanto, resta ainda perceber como ocorreu a decisão em cada um dos segmentos do piloto do ciclo 4. Assim sendo, tendo em conta a formulação de decisão com a qual se avançou no estudo (Situação 2, descrita na secção 4.1), observa-se, na Figura 5.12 que o segmento do GUC Decide foi aquele que se destacou nesta medida. Em contrapartida, o segmento da ficha de cliente foi aquele que apresentou menor taxa de decisão.

Desta forma, verifica-se que o segmento do modelo GUC Decide apresenta um aumento de cerca de 23% (variação percentual) na taxa de decisão, relativamente ao da Ficha de Cliente. Da mesma forma, observa-se que o modelo GUC Decide revela um aumento de cerca de 17% nesta medida, em comparação com o modelo do GUC Atende. Com efeito, este é um resultado extremamente relevante, dado que com este estudo se pretendia aumentar a decisão dos clientes nas campanhas de *telemarketing*.

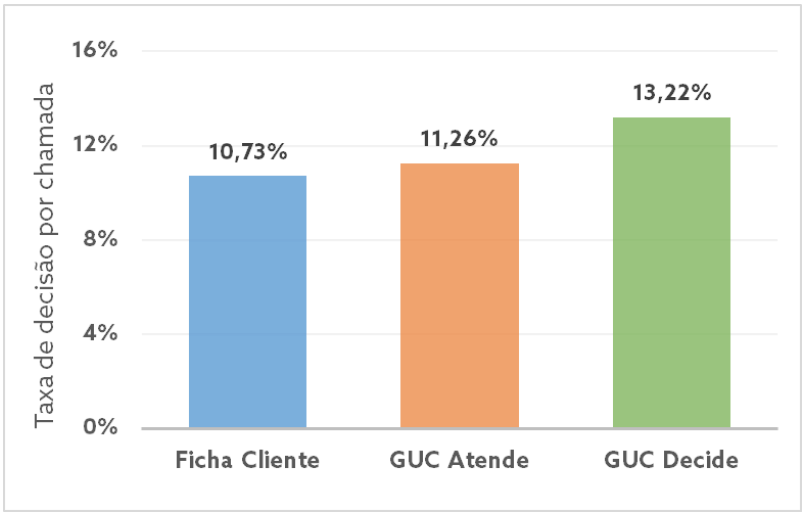

Figura 5.12- Gráfico que apresenta a taxa de decisão obtida para cada um dos segmentos do piloto, no ciclo 4

Tendo em conta os resultados relativos ao piloto com o qual se avançou, retiram-se as seguintes conclusões:

- Os modelos analíticos revelam garantir uma taxa de atendimento superior no decorrer da campanha;
- No segmento do GUC Decide são necessárias menos tentativas de chamada para falar com um decisor, em comparação com os restantes segmentos;
- Nos segmentos da ficha de cliente e do GUC Atende efetuou-se um maior número de chamadas durante o ciclo, e obteve-se uma taxa de decisão inferior à do GUC Decide. De facto, nestes segmentos, verificou-se um esforço por parte do negócio, significativamente superior, para taxas de decisão inferiores.

Salienta-se que as conclusões retiradas revelam que nos segmentos da ficha de cliente e do modelo GUC Atende, os operadores têm maior dificuldade em falar com o decisor de uma empresa. Com efeito, durante o ciclo 4, tal como se pretendia provar, além de o GUC Decide ter apresentado uma taxa de decisão superior, revelou garantir ganhos de eficiência para os processos de negócio. Neste sentido, para obter uma taxa de decisão superior à dos restantes segmentos, foram necessárias menos tentativas de chamada por cliente para falar com o decisor, logo efetuou-se um número de chamadas significativamente inferior, o que garantiu uma melhor gestão dos recursos da operação.

Posto isto, com o intuito de possuir um maior grau de confiança no comportamento dos *outputs* do modelo GUC Decide na operação, realizou-se um novo piloto. Para além disso, pretendia-se, ainda, perceber qual o impacto do número de contactos, que se enviam por cliente, em cada segmento, nos resultados das campanhas comerciais. Nessa perspetiva, avançou-se com um piloto para decorrer no ciclo 5 (24 de abril a 22 de maio), com cerca de 13146 clientes, uma vez mais dividido nos 3 segmentos (4362, 4360, 4424 clientes, respetivamente) anteriormente referidos. No entanto, nesta experimentação, enviou-se um limite de 5 contactos por cliente, em todos os segmentos, de modo a que os três estivessem nas mesmas condições.

Note-se que os KPI que se tencionavam medir neste novo piloto foram análogos aos que tinham sido definidos para o piloto do ciclo 4. No que se refere aos resultados do piloto com que se avançou no ciclo 5 verificou-se que estes vieram reforçar as conclusões retiradas, a partir dos resultados do ciclo 4. Assim, observou-se, de novo, que durante o ciclo 5, nos segmentos da ficha de cliente e do GUC Atende, se efetuou um maior número de chamadas, assim como mais tentativas de chamada por cliente para taxas de decisão inferiores.

Conquanto, no ciclo 5, a taxa de decisão do GUC Atende tenha sido mais equilibrada relativamente à do GUC Decide, constata-se que, uma vez mais, este último garante ganhos de eficiência durante as campanhas. Efetivamente, comprovou-se, novamente, que o segmento do GUC Decide apresentou menos esforço por parte dos colaboradores da operação, para falar com os decisores das empresas.

Capítulo 6 – Conclusão

6.1. Contribuições

O principal objetivo deste projeto de trabalho consistiu na implementação de um modelo de preditivo, no setor B2B, de uma empresa de telecomunicações, capaz de estimar a probabilidade de um determinado contacto pertencer à pessoa que toma decisões em nome de uma empresa/cliente. De modo a dar resposta a este desafio, foi implementada uma abordagem de um problema de classificação, com uma variável dependente binária (1- verifica-se decisão, 0- não se verifica decisão). Neste sentido, foram testadas quatro formulações distintas da variável resposta, a partir dos resultados do negócio, com recurso a dois tipos de algoritmos de modelação de *Machine Learning*, o *Random Forest* e o *Gradient Boosting*, ambos com otimização Bayesiana de hiper-parâmetros.

Para dar resposta a este problema, foi utilizada informação relativa à caraterização das empresas dos clientes (sendo que, o estudo centrou-se apenas em PME) e, ainda, histórico relativo ao tráfego, associado a cada contacto de um cliente, telefonia e interatividades com o SAC da empresa de telecomunicações. Cada observação do painel de dados corresponde a uma chamada de telefonia efetuada pela empresa de telecomunicações para um contacto de um cliente, num determinado mês.

Deste modo, pretendia-se estudar, precisamente, o resultado de negócio de cada chamada efetuada para um contacto, isto é, se se verificou decisão ou não. Assim, o estudo incidiu sobre as chamadas atendidas, uma vez que um indivíduo só tem oportunidade para decidir a recusa ou aceitação de uma proposta quando atende uma chamada.

No que se refere aos resultados obtidos, verificou-se que a variável resposta que revelou ser mais assertiva foi a que se encontra descrita na situação 2, da secção 4.1, em ambas as metodologias utilizadas. Recorde-se que nesta situação a variável resposta toma o valor 1, quando os resultados de negócio correspondem a sucesso e insucesso com oferta apresentada e o valor 0, quando o resultado corresponde a insucesso sem oferta apresentada, não contactado, cancelado, *callback* com oferta apresentada e *outcome* não comercial. Para esta formulação da variável resposta verificou-se que em cerca de 81,34% dos eventos, os contactos não apresentaram uma decisão, por sua vez, em 18,66% ocorreu decisão.

Assim sendo, através da análise comparativa entre os dois tipos de modelos utilizados retiram-se as seguintes conclusões:

- Para a situação referida da variável dependente obteve-se uma AUROC de 0,76 e 0,77, respetivamente, para o modelo *Random Forest* e *Gradient Boosting*. Ainda que os valores sejam bastante semelhantes, aquele que se encontra mais próximo da unidade é o que corresponde à solução obtida pelo *Gradient Boosting.* Com efeito, observou-se que este modelo consegue distinguir, de forma correta, as duas classes que constituem a variável resposta;
- Verificou-se, em ambas os modelos, maior tendência em classificar uma observação que é decisor, como não sendo, do que o contrário. Assim, conclui-se que ambos originam uma quantidade de FN superior à de FP, ou seja, valor de precisão das previsões positivas superior ao da sensibilidade;

- Em todas as situações da variável resposta que foram testadas, o modelo *Gradient Boosting* apresentou sempre métricas de avaliação iguais ou superiores às do *Random Forest*. Nomeadamente, de $F_1 score$, métrica indicada para a comparação de modelos;
- Pretendia-se que o modelo desenvolvido revelasse ter bom desempenho, contudo era essencial desenvolver um produto analítico eficiente a nível computacional. Neste sentido, também o *Gradient Boosting* foi o algoritmo que apresentou melhores métricas de eficiência computacional, em termos de tempo de execução, memória utilizada e CPU.

Posto isto, o *Gradient Boosting* com a variável resposta da situação 2 foi a abordagem eleita para prosseguir no estudo. Assim, realizaram-se 2 pilotos com o intuito de validar os *outputs* do modelo selecionado, nos processos de negócio, isto é, nas campanhas de *telemarketing*, da empresa de telecomunicações.

Desta forma, verificou-se que o modelo construído requer menos tentativas de chamada para obter uma decisão por parte de um contacto, o que origina um volume de chamadas significativamente inferior, em comparação com a metodologia anteriormente utilizada pela empresa. Além disso, observou-se ainda que o modelo apresentou taxas de decisão e atendimento superiores. Assim sendo, conclui-se que o modelo implementado revelou ser uma mais-valia para a empresa de telecomunicações, uma vez que, além de originar taxas de decisão mais satisfatórias, garante ganhos de eficiência na gestão de recursos por parte da operação nas campanhas.

6.2. Limitações e Trabalho Futuro

De facto, foi possível responder com sucesso ao desafio de identificar o contacto da pessoa que decide a recusa ou aceitação de uma determinada proposta, em nome de uma empresa. Certos do contributo do projeto desenvolvido para a empresa de telecomunicações em questão, identificam-se, algumas limitações que podem dar origem a trabalho futuro.

Tendo em conta o limite temporal disponibilizado para o desenvolvimento do projeto, reconhece-se que no momento dedicado à construção do painel de variáveis, faltou a inclusão de alguma informação relevante para o modelo. Nomeadamente informação histórica relativa às decisões dos contactos, associados aos clientes. A título de exemplo, salienta-se a criação de variáveis de contagem do número de vezes que o contacto apresentou um resultado de "sucesso" ou "insucesso com oferta apresentada" em chamadas anteriores, ou, ainda, a criação de uma variável binária que indique se o contacto apresentou decisões nos meses anteriores. Com efeito, considero que a inclusão desta informação no painel de dados e, posteriormente, a realização do treino do modelo, irá trazer perceções positivas.

Quanto à utilização do modelo GUC Decide nos processos de negócio, recomenda-se que o número de contactos que se enviam para as campanhas não seja limitado, e que seja encontrado o ponto de corte das probabilidades previstas a partir do qual é relevante enviar os contactos. Mais precisamente, recomenda-se que, juntamente com a operação, seja definido o *threshold* a partir do qual uma observação deve pertencer à classe positiva da variável dependente, de forma a garantir a obtenção dos valores de precisão e sensibilidade do modelo mais alinhados com as necessidades. Tal como foi acima mencionado, a precisão e a sensibilidade são duas métricas de avaliação com comportamento inverso, pelo que é importante decidir a qual se pretende dar maior destaque. No Anexo F, apresenta-se o gráfico

que mostra a variação das duas métricas referidas para diferentes pontos de corte das probabilidades do modelo.

Note-se que esta decisão deve ser aliada aos resultados obtidos nos processos de negócio, isto é, verificar a partir de que probabilidade os contactos que se enviaram para campanha não apresentaram o comportamento desejado. Neste sentido, seria benéfico analisar esta situação e, posteriormente, validar a decisão através de testes piloto.

Cientes das limitações existentes e perspetivados possíveis caminhos futuros, reconhece-se que o projeto desenvolvido tem forte aplicabilidade no setor B2B da empresa onde foi realizado este estágio curricular, oferecendo oportunidades de otimização e desenvolvimento.

Referências Bibliográficas

[1] ANACOM, “Factos & Números -3º Trimestre 2022,” 21 Dezembro 2022. [Online]. Available: https://www.anacom.pt/streaming/infografia_3T22.pdf?contentId=1735392&field=ATTACHED_FILE. [Acedido em Fevereiro 2023].

[2] M. E. a. A. E. F. Alhaqui, “Machine learning for telecoms: From churn prediction to customer relationship management,” em *IEEE International Conference on Machine Learning and Applied Network Technologies (ICMLANT)*, Soyapango, El Salvador, 2022.

[3] H. H. H. Mahmoud1 e T. Ismail, “A Review of Machine learning Use-Cases in Telecommunication Industry in the 5G Era,” em *16th International Computer Engineering Conference, ICENCO,* 2020.

[4] I. ULLAH, B. RAZA, A. K. MALIK, M. IMRAN, S. U. ISLAM e S. W. KIM, “A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector,” *IEEE Access*, vol. 7, pp. 60134-60149, 2019.

[5] A. Gaur e R. Dubey, “Predicting Customer Churn Prediction In Telecom Sector Using Various Machine Learning Techniques,” em *International Conference on Advanced Computation and Telecommunication (ICACAT)*, Bhopal, India, 2018.

[6] M. K. Islam, P. Hridi, M. S. Hossain and H. S. Narman, "Network Anomaly Detection Using LightGBM: A Gradient Boosting Classifier," in *30th International Telecommunication Networks and Applications Conference (ITNAC)*, VIC, Australia, 2020.

[7] F. I. Alarsan e d. M. Younes, “Analysis and classifcation of heart diseases,” *Journal of Big Data*, vol. 6, nº 81, 2019.

[8] P. Carmona, F. Climent e A. Momparler, “Predicting failure in the U.S. banking sector: An extreme gradient boosting approach,” *International Review of Economics & Finance*, vol. 61, pp. 304-323, 2019.

[9] F. Liu, J. Sun, M. Liu, J. Yang e G. Gui, “Generalized Flight Delay Prediction Method Using Gradient Boosting Decision Tree,” em *IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*, Antwerp, Belgium, 2020.

[10] M. Kuhn e K. Johnson, Applied Predictive Modeling, Nova Iorque: Springer, 2013.

[11] A. Geron, Hands On Machine Learning with Scikit Learn Keras and Tensorflow, Estados Unidos da América: O'REILLY Media, Inc., 2019.

[12] M. B. Kursa, A. Jankowski e W. R. Rudnicki, “Boruta - A system for feature selection,” *Fundamenta Informaticae*, vol. 101, nº 4, 2010.

[13] M. B. Kursa e W. R. Rudnicki, “Feature Selection with the Boruta Package,” *Journal of Statistical Software,* vol. 36, nº 11, 2010.

[14] J. Arcanjo, "Como selecionar as melhores variáveis parao seu modelo com Boruta," [Online]. Available: https://medium.com/data-hackers/como-selecionar-melhores-vari%C3%A1veis-para-o-seu-modelo-com-boruta-ef7cbfb3fc35. [Accessed Março 2023].

[15] P. FLACH, MACHINE LEARNING-The Art and Science of Algorithms that Make Sense of Data, Nova Iorque: Cambridge University Press, 2012.

[16] F. R. d. J. Ramos, Data Science na modelação e previsão de séries económico-financeiras: das metodologias clássicas ao Deep Learning, Repositório do Iscte. http://hdl.handle.net/10071/22964, 2021.

[17] T. Hastie, R. Tibshirani e J. Friedman, The Elements of Statistical Learning: Data Mining, Inference, And Prediction, Springer, 2009.

[18] E. Brochu, V. M. Cora e N. d. Freitas, “A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning,” 14 Dezembro 2010.

[19] P. I. Frazier, “A Tutorial on Bayesian Optimization,” 10 Julho 2018.

[20] I. Dewancker, M. McCourt e S. Clark, “Bayesian Optimization Primer,” *SIGOPT.*

[21] J. Bergstra, R. Bardenet, Y. Bengio e B. Kégl, “Algorithms for Hyper-Parameter Optimization,” 12 Dezembro 2011.

[22] M. L. Baptista, K. Goebel e E. M. Henriques, “Relation between prognostics predictor evaluation metrics and local interpretability SHAP values,” *Artificial Intelligence,* vol. 306, 2022.

[23] K. Futagami, Y. Fukazawa, N. Kapoor e T. Kito, “Pairwise acquisition prediction with SHAP value interpretation,” *Journal of Finance and Data Science,* vol. 7, 2021.

[24] M. J. Ariza-Garzon, J. Arroyo, A. Caparrini e M. J. Segovia-Vargas, “Explainability of a Machine Learning Granting Scoring Model in Peer-to-Peer Lending,” *IEEE Access,* vol. 8, 2020.

[25] G. v. Rossum e F. L. Drake, Python Tutorial, Python Software Foundation, 2012.

Anexo A

Glossário de variáveis

Nome	Descrição
NIF	Identificação do cliente;
NUMERO_TELF	Contacto do Cliente;
YEAR	Ano a que se referem os dados;
MONTH	Mês a que se referem os dados;
B_TIPOLOGIA	Tipo de empresa. Pode tomar 3 valores possíveis: • ENI-NIF começado por 1, 2 ou 3. É um empresário em nome individual; • SOCIEDADE → NIF começado por 5; • OUTRO → NIF começado por outro algarismo.
B_ NUMB_CONTACTS_BY_NIF	Número de contactos associados a um NIF (cliente/empresa);
B_TABELA_FONTE	Fonte da qual provém os contactos: GLOBE; INFORMADB;
B_IS_IN_GLOBE	*Flag* que indica se o contacto está em GLOBE;
B_IS_IN_INFORMADB	*Flag* que indica se contacto está na InformaDB;
B_IS_NEW_CONTACT	*Flag* que indica se o contacto foi encontrado na telefonia;
B_TIPO_NUMERO_TELF	Toma o valor 2 se é um contacto fixo e 9 se é móvel, mas podem existir contactos começados por outro algarismo;
B_ FLAG_PORTFOLIO	Flag que indica se contacto está em portfolio;
B_FLAG_CONT_PRINCIPAL	Flag que indica se contacto é principal;
B_ RGU_QTY	Número de serviços de um cliente;
B_ TV_QTY	Número de serviços de um cliente que são de televisão;
B_ VF_QTY	Número de serviços de um cliente que são de Voz Fixa;
B_ IF_QTY	Número de serviços de um cliente que são de Internet Fixa;
B_ VM_QTY	Número de serviço de um cliente que são de Voz Móvel;
B_ IM_QTY	Número de serviço de um cliente que são de Internet Móvel;
B_TOTAL_AMT	Quantia paga pelo cliente mensalmente (Preço da fatura);
B_PERMANENCY_DAYS_LEFT	Total de dias que restam para terminar a fidelização do cliente (pode assumir um número negativo, correspondendo ao nº de dias desde que terminou a fidelização);
B_ SALES_YEAR_DAT	Último ano de registo de vendas na InformaDB;
B_ SALES_AMT	Variável de caraterização da empresa;

B_EMPLOYEES_QTY	Número de colaboradores de uma empresa;
B_ QUALITY_SCORES_VAL	Valor que varia entre 1 e 5, quanto mais alto melhor a empresa;
U_NR_DIAS_CHAMADAS	Número de dias que um contacto efetuou ou recebeu chamadas, nos 2 meses anteriores;
U_AIRTIME_MEAN	Média (em minutos) do tráfego recebido e efetuado, nos 2 meses anteriores;
U_AIRTIME	Soma (em minutos) do tráfego recebido e efetuado, nos 2 meses anteriores;
U_ALL_OUT_NR_CONTACTOS_UNIQ	Número distinto de contactos para os quais um contacto efetuou chamadas, nos 2 meses anteriores;
U_ALL_OUT_NR_CHAMADAS	Número total de chamadas efetuadas por um contacto, nos 2 meses anteriores;
U_ALL_OUT_NR_DIAS_CHAMADAS	Número total de dias em que um contacto efetuou chamadas, nos 2 meses anteriores;
U_ALL_OUT_AIRTIME_MEAN	Média (em minutos) do tráfego efetuado por um contacto, nos 2 meses anteriores;
U_ALL_OUT_AIRTIME_SUM	Soma (em minutos) do tráfego efetuado por um contacto, nos 2 meses anteriores;
U_ALL_OUT_CALL_SUM_DAWN	Soma (em minutos) do tráfego efetuado entre as 23H e as 5H por um contacto, nos 2 meses anteriores;
U_ALL_OUT_CALL_SUM_EARLYMORNING	Soma (em minutos) do tráfego efetuado entre as 5H e as 8H por um contacto, nos 2 meses anteriores;
U_ALL_OUT_CALL_SUM_MORNING	Soma (em minutos) do tráfego efetuado entre as 8H e as 12H por um contacto, nos 2 meses anteriores;
U_ALL_OUT_CALL_SUM_AFTERNOON	Soma (em minutos) do tráfego efetuado entre as 12H e as 19H por um contacto, nos 2 meses anteriores;
U_ALL_OUT_CALL_SUM_EVENING	Soma (em minutos) do tráfego efetuado entre as 19H e as 23H por um contacto, nos 2 meses anteriores;
U_ALL_OUT_CONTACTO_OTHER_TYPE_2_SUM	Soma (em minutos) do tráfego efetuado para números de telefone começados pelo algarismo 2, nos 2 meses anteriores;
U_ALL_OUT_CONTACTO_OTHER_TYPE_3_SUM	Soma (em minutos) do tráfego efetuado para números de telefone começados pelo algarismo 3, nos 2 meses anteriores;
U_ALL_OUT_CONTACTO_OTHER_TYPE_7_SUM	Soma (em minutos) do tráfego efetuado para números de telefone começados pelo algarismo 7, nos 2 meses anteriores;

U_ALL_OUT_CONTACTO_OTHER_TYPE_9_SUM	Soma (em minutos) do tráfego efetuado para números de telefone começados pelo algarismo 9, nos 2 meses anteriores;
U_ALL_OUT_CAL_DAY_OF_WEEK_MON_SUM	Soma (em minutos) do tráfego efetuado nos 2 meses anteriores, numa segunda-feira;
U_ALL_OUT_CAL_DAY_OF_WEEK_TUE_SUM	Soma (em minutos) do tráfego efetuado nos 2 meses anteriores, numa terça-feira;
U_ALL_OUT_CAL_DAY_OF_WEEK_WED_SUM	Soma (em minutos) do tráfego efetuado nos 2 meses anteriores, numa quinta-feira;
U_ALL_OUT_CAL_DAY_OF_WEEK_FRI_SUM	Soma (em minutos) do tráfego efetuado nos 2 meses anteriores, numa sexta-feira;
U_ALL_OUT_CAL_DAY_OF_WEEK_SAT_SUM	Soma (em minutos) do tráfego efetuado nos 2 meses anteriores, num sábado;
U_ALL_OUT_CAL_DAY_OF_WEEK_SUN_SUM	Soma (em minutos) do tráfego efetuado nos 2 anteriores, num domingo;
U_NIF_OUT_NR_CONTACTOS_UNIQ	Número distinto de contactos que pertencem ao mesmo NIF, para os quais um contacto efetuou chamadas, nos 2 meses anteriores;
U_NIF_OUT_NR_CHAMADAS	Número de chamadas efetuadas para contactos que pertencem ao mesmo NIF, nos 2 meses anteriores;
U_NIF_OUT_NR_DIAS_CHAMADAS	Número de dias em que se efetuaram chamadas para contactos que pertencem ao mesmo NIF, nos 2 meses anteriores;
U_NIF_OUT_AIRTIME_MEAN	Média (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, nos 2 meses anteriores;
U_NIF_OUT_AIRTIME_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertençam ao mesmo NIF, nos 2 meses anteriores;
U_NIF_OUT_CALL_SUM_DAWN	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, entre as 23H e as 5H, nos 2 meses anteriores;
U_NIF_OUT_CALL_SUM_EARLYMORNING	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, entre as 5H e as 8H, nos 2 meses anteriores;
U_NIF_OUT_CALL_SUM_MORNING	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, entre as 8H e as 12H, nos 2 meses anteriores;
U_NIF_OUT_CALL_SUM_AFTERNOON	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, entre as 12H e as 19H, nos 2 meses anteriores;

U_NIF_OUT_CALL_SUM_EVENING	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, entre as 19H e as 23H, nos 2 meses anteriores;
U_NIF_OUT_CONTACTO_OTHER_TYPE_2_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, para números de telefone começados pelo algarismo 2, nos 2 meses anteriores;
U_NIF_OUT_CONTACTO_OTHER_TYPE_3_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, para números de telefone começados pelo algarismo 3, nos 2 meses anteriores;
U_NIF_OUT_CONTACTO_OTHER_TYPE_7_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, para números de telefone começados pelo algarismo 7, nos 2 meses anteriores;
U_NIF_OUT_CONTACTO_OTHER_TYPE_8_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, para números de telefone começados pelo algarismo 8, nos 2 meses anteriores;
U_NIF_OUT_CONTACTO_OTHER_TYPE_9_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, para números de telefone começados pelo algarismo 9, nos 2 meses anteriores;
U_NIF_OUT_CAL_DAY_OF_WEEK_MON_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertencem ao mesmo NIF, nos 2 meses anteriores, numa segunda-feira;
U_NIF_OUT_CAL_DAY_OF_WEEK_TUE_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertençam ao mesmo NIF, nos 2 meses anteriores, numa terça-feira;
U_NIF_OUT_CAL_DAY_OF_WEEK_WED_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertençam ao mesmo NIF, nos 2 meses anteriores, numa quarta-feira;
U_NIF_OUT_CAL_DAY_OF_WEEK_THU_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertençam ao mesmo NIF, nos 2 meses anteriores, numa quinta-feira;
U_NIF_OUT_CAL_DAY_OF_WEEK_FRI_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertençam ao mesmo NIF, nos 2 meses anteriores, numa sexta-feira;
U_NIF_OUT_CAL_DAY_OF_WEEK_SAT_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertençam ao mesmo NIF, nos 2 meses anteriores, num sábado;

U_NIF_OUT_CAL_DAY_OF_WEEK_SUN_SUM	Soma (em minutos) do tráfego efetuado para contactos que pertençam ao mesmo NIF, nos 2 meses anteriores, num domingo;
U_ALL_IN_NR_CONTACTOS_UNIQ	Número distinto de contactos, dos quais um contacto recebeu chamadas nos 2 meses anteriores;
U_ALL_IN_NR_CHAMADAS	Número total de chamadas recebidas por um contacto, nos meses 2 anteriores;
U_ALL_IN_NR_DIAS_CHAMADAS	Número total de dias em que um contacto recebeu chamadas, nos 2 meses anteriores;
U_ALL_IN_AIRTIME_MEAN	Média (em minutos) do tráfego recebido nos 2 meses anteriores;
U_ALL_IN_AIRTIME_SUM	Soma (em minutos) do tráfego recebido, nos 2 meses anteriores;
U_ALL_IN_CALL_SUM_DAWN	Soma (em minutos) do tráfego recebido, entre as 23H e as 5H, por um contacto, nos 2 meses anteriores;
U_ALL_IN_CALL_SUM_EARLYMORNING	Soma (em minutos) do tráfego recebido, entre as 5H e as 8H, por um contacto, nos 2 meses anteriores;
U_ALL_IN_CALL_SUM_MORNING	Soma (em minutos) do tráfego recebido, entre as 8H e as 12H por um contacto, nos 2 meses anteriores;
U_ALL_IN_CALL_SUM_AFTERNOON	Soma (em minutos) do tráfego recebido, entre as 12H e as 19H por um contacto, nos 2 meses anteriores;
U_ALL_IN_CALL_SUM_EVENING	Soma (em minutos) do tráfego recebido, entre as 19H e as 23H por um contacto, nos 2 meses anteriores;
U_ALL_IN_CONTACTO_OTHER_TYPE_2_SUM	Soma (em minutos) do tráfego recebido de números de telefone começados pelo algarismo 2, nos 2 meses anteriores;
U_ALL_IN_CONTACTO_OTHER_TYPE_3_SUM	Soma (em minutos) do tráfego recebido de números de telefone começados pelo algarismo 3, nos 2 meses anteriores;
U_ALL_IN_CONTACTO_OTHER_TYPE_7_SUM	Soma (em minutos) do tráfego recebido de números de telefone começados pelo algarismo 7, nos 2 meses anteriores;
U_ALL_IN_CONTACTO_OTHER_TYPE_8_SUM	Soma (em minutos) do tráfego recebido de números de telefone começados pelo algarismo 8, nos 2 meses anteriores;
U_ALL_IN_CONTACTO_OTHER_TYPE_9_SUM	Soma (em minutos) do tráfego recebido para números de telefone começados pelo algarismo 9, nos 2 meses anteriores;
U_ALL_IN_CAL_DAY_OF_WEEK_MON_SUM	Soma (em minutos) do tráfego recebido, nos 2 meses anteriores, numa segunda-feira;

U_ALL_IN_CAL_DAY_OF_WEEK_TUE_SUM	Soma (em minutos) do tráfego recebido, nos 2 meses anteriores, numa terça-feira;
U_ALL_IN_CAL_DAY_OF_WEEK_WED_SUM	Soma (em minutos) do tráfego recebido, nos 2 meses anteriores, numa quarta-feira;
U_ALL_IN_CAL_DAY_OF_WEEK_THU_SUM	Soma (em minutos) do tráfego recebido, nos 2 meses anteriores, numa quinta-feira;
U_ALL_IN_CAL_DAY_OF_WEEK_FRI_SUM	Soma (em minutos) do tráfego recebido, nos 2 meses anteriores, numa sexta-feira;
U_ALL_IN_CAL_DAY_OF_WEEK_SAT_SUM	Soma (em minutos) do tráfego recebido, nos 2 meses anteriores, num sábado;
U_ALL_IN_CAL_DAY_OF_WEEK_SUN_SUM	Soma (em minutos) do tráfego recebido nos 2 meses anteriores, num domingo;
U_NIF_IN_NR_CONTACTOS_UNIQ	Número distinto de contactos que pertencem ao mesmo NIF, dos quais um contacto recebeu chamadas, nos 2 meses anteriores;
U_NIF_IN_NR_CHAMADAS	Número de chamadas recebidas de contactos que pertencem ao mesmo NIF, nos 2 meses anteriores;
U_NIF_IN_NR_DIAS_CHAMADAS	Número de dias em que se receberam chamadas de contactos que pertencem ao mesmo NIF, nos 2 meses anteriores;
U_NIF_IN_AIRTIME_MEAN	Média (mensal, em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, nos 2 meses anteriores;
U_NIF_IN_AIRTIME_SUM	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, nos 2 meses anteriores;
U_NIF_IN_CALL_SUM_DAWN	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, entre as 23H e as 5H, nos 2 meses anteriores;
U_NIF_IN_CALL_SUM_EARLYMORNING	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, entre as 5H e as 8H, nos 2 meses anteriores;
U_NIF_IN_CALL_SUM_MORNING	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, entre as 8H e as 12H, nos 2 meses anteriores;
U_NIF_IN_CALL_SUM_AFTERNOON	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, entre as 12H e as 19H, nos 2 meses anteriores;
U_NIF_IN_CALL_SUM_EVENING	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, entre as 19H e as 23H, nos 2 meses anteriores;

U_NIF_IN_CONTACTO_OTHER_TYPE_2_SUM	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, para números de telefone começados pelo algarismo 2;
U_NIF_IN_CONTACTO_OTHER_TYPE_3_SUM	Soma (em minutos) do tráfego recebido por contactos que pertençam ao mesmo NIF, para números de telefone começados pelo algarismo 3;
U_NIF_IN_CONTACTO_OTHER_TYPE_7_SUM	Soma (em minutos) do tráfego recebido por contactos que pertençam ao mesmo NIF, para números de telefone começados pelo algarismo 7;
U_NIF_IN_CONTACTO_OTHER_TYPE_8_SUM	Soma (em minutos) do tráfego recebido por contactos que pertençam ao mesmo NIF, para números de telefone começados pelo algarismo 8;
U_NIF_IN_CONTACTO_OTHER_TYPE_9_SUM	Soma (em minutos) do tráfego recebido por contactos que pertençam ao mesmo NIF, para números de telefone começados pelo algarismo 9;
U_NIF_IN_CAL_DAY_OF_WEEK_MON_SUM	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, nos 2 meses anteriores, numa segunda-feira;
U_NIF_IN_CAL_DAY_OF_WEEK_TUE_SUM	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, nos 2 meses anteriores, numa terça-feira;
U_NIF_IN_CAL_DAY_OF_WEEK_WED_SUM	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, nos 2 meses anteriores, numa quarta-feira;
U_NIF_IN_CAL_DAY_OF_WEEK_THU_SUM	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, nos 2 meses anteriores, numa quinta-feira;
U_NIF_IN_CAL_DAY_OF_WEEK_FRI_SUM	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, nos 2 meses anteriores numa sexta-feira;
U_NIF_IN_CAL_DAY_OF_WEEK_SAT_SUM	Soma (em minutos) do tráfego recebido por contactos que pertencem ao mesmo NIF, nos 2 meses anteriores, num sábado;
U_NIF_IN_CAL_DAY_OF_WEEK_SUN_SUM	Soma (em minutos) do tráfego recebido por contactos que pertençam ao mesmo NIF, nos 2 meses anteriores, num domingo;
U_NR_CONTACTOS_UNIQ	Número distinto de contactos para os quais um contacto efetuou ou recebeu chamadas, nos 2 meses anteriores;
U_NR_CONTACTOS_UNIQ_OUT_RATIO	Rácio entre o número distinto de contactos para os quais um contacto efetuou chamadas nos 2 meses anteriores e o número distinto de contactos para os

	quais um contacto efetuou ou recebeu chamadas. Dá a perspetiva de *outgoing* total;
U_NR_CHAMADAS	Número total de chamadas efetuadas e recebidas, nos 2 meses anteriores;
U_NR_CHAMADAS_NIF	Rácio entre número total de chamadas efetuadas e recebidas para contactos pertencentes ao mesmo NIF e número total de chamadas. Dá-nos a perspetiva do tráfego para contactos da mesma rede, em comparação com o total de contactos;
U_NR_CHAMADAS_NIF_ALL_RATIO	Rácio entre número total de chamadas efetuadas e recebidas, para contactos pertencentes ao mesmo NIF e o número total de chamadas;
U_NR_CHAMADAS_OUT_RATIO	Rácio entre o total de chamadas efetuadas e total de chamadas, nos 2 meses anteriores;
U_NR_CHAMADAS_NIF_OUT_RATIO	Rácio entre número de chamadas efetuadas para contactos pertencentes ao mesmo NIF e o total chamadas para contactos pertencentes ao mesmo NIF;
U_CAL_DAWN_RATIO	Rácio entre o tráfego efetuado e recebido, entre as 23H e as 5H, e o tráfego efetuado e recebido;
U_CAL_EARLYMORNING_RATIO	Rácio entre o tráfego efetuado e recebido, entre as 5H e as 8H, e o tráfego efetuado e recebido;
U_CAL_MORNING_RATIO	Rácio entre o tráfego efetuado e recebido, entre as 12H e as 19H, e o tráfego efetuado e recebido;
U_CAL_EVENING_RATIO	Rácio entre o tráfego efetuado e recebido, entre as 19H e as 23H, e o tráfego efetuado e recebido;
U_NR_DIAS_CHAMADAS_OUT_RATIO	Rácio entre o número de dias de tráfego efetuado e o número de dias de tráfego efetuado e recebido;
U_AIRTIME_OUT_RATIO	Rácio entre o tráfego efetuado e o tráfego recebido e efetuado;
U_OTHER_TYPE_2_SUM_RATIO	Rácio entre o tráfego efetuado e recebido, para contactos cujo número de telefone começa com o algarismo 2 e o tráfego efetuado e recebido;
U_OTHER_TYPE_3_SUM_RATIO	Rácio entre o tráfego efetuado e recebido para contactos cujo número de telefone começa com o algarismo 3 e o tráfego efetuado e recebido;
U_OTHER_TYPE_7_SUM_RATIO	Rácio entre o tráfego efetuado e recebido para contactos cujo número de telefone começa com o algarismo 7 e o tráfego efetuado e recebido;
U_OTHER_TYPE_8_SUM_RATIO	Rácio entre o tráfego efetuado e recebido para contactos cujo número de telefone começa com o algarismo 8 e o tráfego efetuado e recebido;

U_OTHER_TYPE_9_SUM_RATIO	Rácio entre o tráfego efetuado e recebido para contactos cujo número de telefone começa com o algarismo 9 e o tráfego efetuado e recebido;
U_CAL_DAY_OF_WEEK_MON_RATIO	Rácio entre o tráfego efetuado e recebido para contactos à segunda-feira e o tráfego efetuado e recebido;
U_CAL_DAY_OF_WEEK_TUE_RATIO	Rácio entre o tráfego efetuado e recebido para contactos à terça-feira e o tráfego efetuado e recebido;
U_CAL_DAY_OF_WEEK_WED_RATIO	Rácio entre o tráfego efetuado e recebido para contactos à quarta-feira e o tráfego efetuado e recebido;
U_CAL_DAY_OF_WEEK_THU_RATIO	Rácio entre o tráfego efetuado e recebido para contactos à quinta-feira e o tráfego efetuado e recebido;
U_CAL_DAY_OF_WEEK_FRI_RATIO	Rácio entre o tráfego efetuado e recebido para contactos à sexta-feira e o tráfego efetuado e recebido;
U_CAL_DAY_OF_WEEK_SAT_RATIO	Rácio entre o tráfego efetuado e recebido para contactos ao sábado e o tráfego efetuado e recebido;
U_CAL_DAY_OF_WEEK_SUN_RATIO	Rácio entre o tráfego efetuado e recebido para contactos ao domingo e o tráfego efetuado e recebido;
U_NR_CONTACTOS_UNIQ_NIF	Número distinto de contactos que pertencem ao mesmo NIF para os quais se efetuam ou recebem chamadas;
U_NR_CONTACTOS_UNIQ_NIF_ALL_RATIO	Rácio entre número distinto de contactos para os quais se efetuam ou recebem chamadas que pertençam ao mesmo NIF e todos os contactos para os quais se efetuam ou recebem chamadas;
U_NR_CONTACTOS_UNIQ_NIF_OUT_RATIO	Rácio entre o número distinto de contactos para os quais se efetuam chamadas e o número distinto de contactos para os quais se efetuam ou recebem chamadas;
U_NR_DIAS_CHAMADAS_NIF	Número de dias em que foram efetuadas ou recebidas chamadas pertencentes ao mesmo NIF;
U_NR_DIAS_CHAMADAS_NIF_ALL_RATIO	Rácio entre o número dias de chamadas para contactos pertencentes ao mesmo NIF para os quais se efetuam ou recebem chamadas e o número de dias de chamadas para contactos (pertencentes ao mesmo NIF ou não) para os quais se efetuam ou recebem chamadas;
U_NR_DIAS_CHAMADAS_NIF_OUT_RATIO	Rácio entre o número dias em que se efetuam chamadas para contactos pertencentes ao mesmo NIF e o número de dias em que se efetuam ou recebem

	chamadas para contactos (pertencentes ao mesmo NIF ou não);
U_NR_CONTACTOS_UNIQ_DIFF	Subtração entre o número único de contactos para os quais se efetuam ou recebem chamadas do mês anterior e de há 2 meses (seja m o mês atual, então (chamadas m-1) - (chamadas m-2));
U_NR_CHAMADAS_DIFF	Subtração entre o número de chamadas do mês anterior e de há 2 meses;
U_NR_CONTACTOS_UNIQ_NIF_DIFF	Subtração entre o número único de contactos para os quais se efetuam ou recebem chamadas do mês anterior e de há 2 meses, para contactos pertencentes ao mesmo NIF;
U_NR_CHAMADAS_NIF_DIFF	Subtração entre o número de chamadas do mês anterior e de há 2 meses, para contactos pertencentes ao mesmo NIF;
U_HAS_USAGE	*Flag* que indica se um contacto tem variáveis de tráfego preenchidas para esse mês ou não;
CAMPAIGN_ID	Número de identificação da campanha de telemarketing;
CAMPAIGN_DSC	Descrição da campanha de *telemarketing*;
SUBCAMPAIGN_ID	Número de identificação da subcampanha de *telemarketing*;
SUBCAMPAIGN_DSC	Descrição da subcampanha de *telemarketing*;
T_CALL_ID	Número de identificação de uma chamada;
SALE_CYCLE_ID	Ciclo da campanha de *telemarketing*;
DATA	Data em que foi efetuada a chamada;
WRONG_NUMBER	*Flag* a indicar se o contacto alguma vez disse que não pretendia ser contactado;
T_ATENDIMENTOS	Número de chamadas atendidas por parte de um contacto, nos 6 meses anteriores;
T_TOTAL_CALLS	Número total de chamadas para um determinado contacto, nos 6 meses anteriores, ou seja, tendo em conta o histórico já existente;
RATIO_ATENDIMENTO	Rácio de atendimento por parte de um contacto, nos 6 meses anteriores;
CALL_RESULT_T-1	Resultado da chamada anterior, ou seja, no instante t-1, sendo que a variável resposta representa a chamada no instante t. (1 se for atendimento, 0 caso contrário);
CHAIN_N_T-1	*Chain number* da ocorrência anterior de um contacto em telefonia, ou seja, no instante t-1, sendo que a

	target é a chamada no instante t. *Chain number* indica a prioridade de um contacto, sendo que 1 é contacto principal e daí em diante vai perdendo importância;
CHAIN_N_CHANGED	*Flag* que indica se o *chain number* alguma vez mudou;
ATTEMPT_T-1	Número de tentativas necessárias, por ciclo, para ter uma resposta de telefonia por parte do contacto, seja o resultado atendimento ou ir para o voice-mail. Novamente, no instante t-1, por exemplo: • Tentativa 1: Ligamos para o cliente e este não atende; (ATTEMPT_T-1 =1) • Tentativa 2: Ligamos novamente para o cliente e não atende; (ATTEMPT_T-1 =2) • Tentativa 3: Ligamos novamente e atende. (ATTEMPT_T-1 =3)
T_LAST_CALL_INTERVAL	Tempo (em minutos) que passou desde a última vez que este contacto foi contactado numa campanha de telemarketing;
T_HISTORY	*Flag* que indica se um contacto tem ou não histórico de telefonia;
CALL_RESULT	Resultado da chamada atual, de um determinado contacto, ou seja, se um contacto atendeu a chamada ou não;
SAC_COUNT_CHAMADAS	Número total de chamadas que um contacto efetua para o SAC, nos 6 meses anteriores;
SAC_COUNT_TOPIC_TECNICO	Número de tópicos técnicos abertos para um contacto, nos 6 meses anteriores;
SAC_COUNT_TOPIC_NAO_TECNICO	Número de tópicos não técnicos abertos para um contacto, nos 6 meses anteriores;
SAC_TOTAL_OPEN_TOPICS	Número total de tópicos abertos para um Contacto, nos 6 meses anteriores;
FLG_SAC	*Flag* que indica se um contacto efetuou chamadas para o SAC e tem tópicos abertos, nos 6 meses anteriores.

Anexo B

Variáveis selecionadas pelo Boruta

No que se refere à variável resposta da situação 2, descrita na secção 4.1, durante a fase de pré-processamento de dados, o conjunto de variáveis independentes selecionado para prosseguir no estudo é, de seguida, identificado. De destacar que o algoritmo utilizado para a seleção de variáveis foi o Boruta, sendo assim, como mencionado na secção 3.1, as variáveis selecionadas correspondem às que o algoritmo revelou ter certeza da sua importância para o estudo e às consideradas indeterminadas.

Desta forma, as variáveis independentes que o Boruta considerou que, indubitavelmente, deviam ser selecionadas foram:

- B_NMB_CONTACTS_BY_NIF;
- B_TIPO_NUMERO_TELF;
- B_FLAG_PORTFOLIO;
- B_FLAG_CONT_PRINCIPAL;
- B_RGU_QTY;
- B_VM_QTY;
- B_TOTAL_AMT;
- B_PERMANENCY_DAYS_LEFT;
- B_SALES_AMT;
- B_EMPLOYEES_QTY;
- U_AIRTIME_MEAN;
- U_AIRTIME;
- U_ALL_OUT_NR_CONTACTOS_UNIQ;
- U_ALL_OUT_AIRTIME_MEAN;
- U_ALL_OUT_AIRTIME_SUM;
- U_ALL_OUT_CONTACTO_OTHER_TYPE_2_SUM;
- U_ALL_IN_NR_CONTACTOS_UNIQ;
- U_ALL_IN_AIRTIME_MEAN;
- U_ALL_IN_AIRTIME_SUM;
- U_ALL_IN_CALL_SUM_MORNING;
- U_ALL_IN_CALL_SUM_EVENING;
- U_ALL_IN_CONTACTO_OTHER_TYPE_2_SUM;
- U_NR_CONTACTOS_UNIQ;
- U_NR_CONTACTOS_UNIQ_OUT_RATIO;
- U_NR_CHAMADAS_NIF;
- U_NR_CHAMADAS_NIF_ALL_RATIO;
- U_NR_CHAMADAS_OUT_RATIO;
- U_CAL_DAWN_RATIO;
- U_CAL_MORNING_RATIO;
- U_CAL_EVENING_RATIO;
- U_NR_DIAS_CHAMADAS_OUT_RATIO;
- U_AIRTIME_OUT_RATIO;

- U_CAL_DAY_OF_WEEK_SUN_RATIO;
- U_NR_CONTACTOS_UNIQ_NIF_ALL_RATIO;
- U_NR_DIAS_CHAMADAS_NIF_ALL_RATIO;
- RATIO_ATENDIMENTO;
- ATTEMPT_T_1;
- T_ATENDIMENTOS;
- T_TOTAL_CALLS;
- T_LAST_CALL_INTERVAL;
- T_HISTORY;
- SAC_COUNT_CHAMADAS;
- SAC_COUNT_TOPIC_TECNICO;
- SAC_COUNT_TOPIC_NAO_TECNICO;
- SAC_TOTAL_OPEN_TOPICS;
- FLG_SAC;

Por sua vez, as variáveis explicativas consideradas como indeterminadas, isto é, em que o algoritmo do Boruta revelou alguma incerteza, foram as seguintes:

- U_NR_DIAS_CHAMADAS;
- U_ALL_IN_NR_CHAMADAS;
- U_ALL_IN_CALL_SUM_AFTERNOON;
- U_NR_CHAMADAS;
- CHAIN_N_T_1;

Anexo C

Variável resposta da situação 1- Resultados dos modelos

Neste anexo, apresentam-se os resultados das métricas de avaliação obtidas para os dados de treino e teste para a variável resposta da situação 1, descrita da secção 4.1, para cada um dos algoritmos de modelação testados.

Neste sentido, a variável dependente da situação 1 distribui-se da seguinte forma: em 68,77% dos eventos não se verifica decisão (0 - não contactado, cancelado, *callback* com oferta apresentada e *outcome* não comercial), por outro lado, em 31,23% ocorre decisão (1 - sucesso, insucesso com oferta apresentada e insucesso sem oferta apresentada).

Os valores das métricas de avaliação, relativas ao *Random Forest* são apresentados na Tabela C.1. Neste caso, verifica-se que o modelo é capaz de classificar, corretamente, cerca de 74% das observações dos dados de teste. Desta forma, a percentagem de erro deste modelo é de 26%, pelo que classifica de forma incorreta cerca de 26% das observações dos dados de teste. Por sua vez, observa-se, que nos dados de teste, o modelo prevê, de forma correta, cerca de 67% das observações que foram classificadas como positivas. No entanto, relativamente às observações que na realidade pertencem à classe positiva da variável resposta, o modelo identifica cerca de 31% corretamente.

Efetivamente, observa-se que a precisão das previsões positivas do modelo é superior à sensibilidade do modelo. Isto mostra que o modelo revela ser mais propenso a identificar uma observação da classe positiva da variável dependente como sendo da classe negativa (FN), do que identificar uma observação da classe negativa como sendo da positiva (FP).

Tabela C.1- Valores das métricas de avaliação obtidos nos dados de treino e teste, para o *Random Forest*

Métricas \ Dados	**Treino**	**Teste**
Acurácia	0,76	0,74
Precisão	0,78	0,67
Sensibilidade	0,35	0,31
F1 Score	0,49	0,42
AUROC	0,86	0,75

Ao observar ainda a tabela mencionada, verifica-se que na sua generalidade, os valores das métricas obtidas para o conjunto de dados de treino são superiores às dos dados de teste. Tal como foi referido, anteriormente, esta situação é justificada pelo facto, de o algoritmo ter aprendido a classificar as observações, de acordo com as caraterísticas dos dados de treino. Ainda assim, a diferença entre valores não é significativa, pelo que não se considera que tenha ocorrido *overfitting.*

Dado que, neste estudo, o valor da AUROC foi a medida escolhida para avaliar a qualidade de um modelo e o valor obtido nesta situação é inferior aos valores mencionados na secção 5.3, não se considerou esta hipótese para continuar no estudo.

Na Tabela C.*2*, apresentam-se os valores das métricas de avaliação obtidos para o *Gradient Boosting*. Relativamente à exatidão das previsões do modelo, verifica-se que este prevê, de forma correta, cerca de 75% das observações presentes nos dados de teste, pelo que apresenta uma

percentagem de erro de classificação de 25%. No que diz respeito à precisão do modelo, nota-se que das observações previstas como positivas, cerca 65% foram classificadas de forma assertiva. Por outro lado, para as observações que na realidade pertencem á classe positiva da variável resposta, o modelo identifica 39% corretamente.

Também neste caso, se verifica que a precisão das previsões é superior à sensibilidade do modelo. Assim, constata-se que, tal como o caso anterior, este modelo tem maior tendência a identificar uma observação que é decisor como não sendo (FN), do que o contrário (FP).

Tabela C.2- Valores das métricas de avaliação obtidos nos dados de treino e teste, para o *Gradient Boosting*

Métricas \ Dados	**Treino**	**Teste**
Acurácia	0,77	0,75
Precisão	0,72	0,65
Sensibilidade	0,42	0,39
F1 Score	0,53	0,49
AUROC	0,82	0,77

Tal como é possível verificar através da observação da tabela acima apresentada, o valor da AUROC para esta situação em particular é semelhante aos valores desta métrica apresentados na secção 5.3. Assim, a razão pela qual esta hipótese não foi escolhida para prosseguir no estudo, deve-se ao facto de se ter considerado que, para resultados semelhantes, era preferível considerar uma formulação de variável resposta mais específica, no que concerne às decisões dos clientes. Com efeito, a situação 2 da variável reposta constitui uma formulação mais assertiva da definição de decisão, ou seja, garante que de facto o cliente teve oportunidade para aceitar ou recusar uma proposta.

Anexo D

Variável resposta da situação 1.1- Resultados dos modelos

Neste anexo, apresentam-se os resultados das métricas de avaliação obtidas para os dados de treino e teste para a variável resposta da situação 1.1, descrita da secção 4.1, para cada um dos algoritmos de modelação testados. Neste sentido, a variável dependente da situação 1.1 distribui-se da seguinte forma: em 53,63% dos eventos não se verifica decisão (0 - não contactado, cancelado, *callback* com oferta apresentada e *outcome* não comercial), por outro lado, em 46,37% ocorre decisão (1 - sucesso, insucesso com oferta apresentada e insucesso sem oferta apresentada, *callback* decisor e agendado decisor).

Na Tabela D.1, são apresentados os valores das métricas de avaliação relativas ao *Random Forest.* No que se refere à exatidão (acurácia) das previsões que o modelo originou, verifica-se que este é capaz de classificar corretamente cerca de 65% das observações dos dados de teste. Assim sendo, a percentagem de erro deste modelo é de 35%. Observa-se, ainda, que nos dados de teste, o modelo prevê, de forma correta, cerca de 64% das observações, que foram classificadas como positivas. Relativamente às observações que na realidade são decisores, o modelo identifica cerca de 54% corretamente.

Neste caso, observa-se que a precisão das previsões positivas do modelo está relativamente equilibrada com a sensibilidade do modelo. Isto indica que o modelo identifica uma observação da classe positiva da variável dependente como sendo da classe negativa (FN), na mesma proporção que identifica uma observação da classe negativa como sendo da positiva (FP).

Tabela D.1- Valores das métricas de avaliação obtidos nos dados de treino e teste, para o *Random Forest*

Métricas \ Dados	**Treino**	**Teste**
Acurácia	0,90	0,65
Precisão	0,95	0,64
Sensibilidade	0,84	0,54
F1 Score	0,89	0,58
AUROC	0,97	0,70

Na Figura D.1, encontram-se representadas as curvas de ROC obtidas para os dados de treino e teste, bem como as respetivas AUROC. Ao observar a figura mencionada, é percetível que existe uma discrepância significativa entre as curvas de ROC e, consequentemente, na AUROC dos dois conjuntos de dados. De facto, verifica-se que, na sua generalidade, os valores das métricas obtidas para o conjunto de dados de treino são significativamente superiores às dos dados de teste. Tal como foi referido anteriormente, esta situação é justificada pelo facto de o algoritmo ter aprendido a classificar as observações de acordo com as caraterísticas dos dados de treino.

No entanto, nesta situação em particular, nota-se que, os valores obtidos nos dois conjuntos de dados apresentam uma discrepância significativa entre si, pelo que se considera que tenha ocorrido *overfitting.* Efetivamente, o modelo apresentava um desempenho quase perfeito nos dados de treino, sendo que, quando aplicado aos dados de teste, este piorou significativamente. Dado ser um acontecimento indesejável, não se considerou esta situação para prosseguir no estudo.

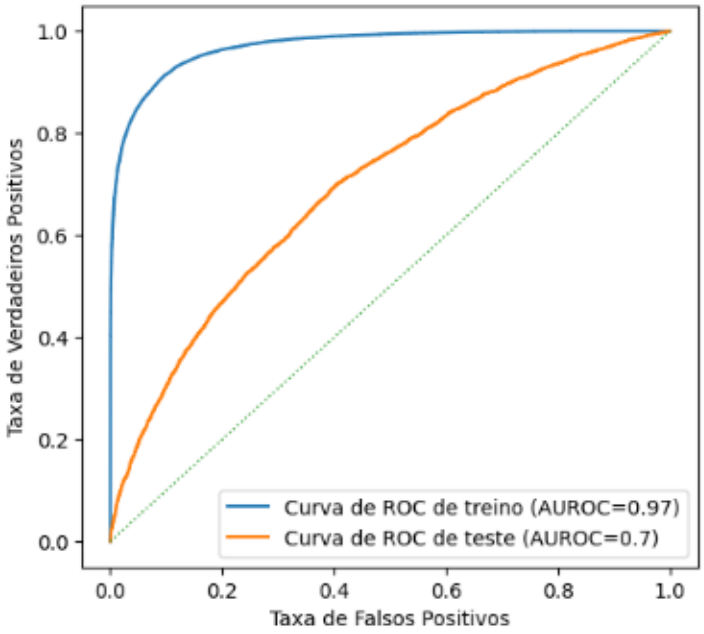

Figura D.1- Curva de ROC e respetiva AUROC do algoritmo *Random Forest*, para a variável resposta da situação 1.1

Por sua vez, no *Gradient Boosting*, os valores das métricas de avaliação são apresentados na Tabela D.2. Relativamente à exatidão das previsões do modelo, verifica-se que este prevê, de forma correta, cerca de 67% das observações presentes nos dados de teste, pelo que apresenta uma percentagem de erro de classificação de 33%. No que diz respeito à precisão do modelo, nota-se que das observações previstas como decisores, cerca 65% foram classificadas de forma assertiva. Por outro lado, para as observações que na realidade pertencem à classe positiva da variável resposta, o modelo identifica 57% corretamente.

Neste caso, observa-se também que o valor da precisão das previsões positivas do modelo está próximo do valor da sensibilidade do modelo. Assim, conclui-se que o modelo identifica uma observação da classe positiva da variável dependente como sendo da classe negativa (FN), na mesma medida que identifica uma observação da classe negativa como sendo da positiva (FP).

Tabela D.2- Valores das métricas de avaliação obtidos nos dados de treino e teste, para o *Gradient Boosting*

Métricas \ Dados	**Treino**	**Teste**
Acurácia	0,74	0,67
Precisão	0,75	0,65
Sensibilidade	0,66	0,57
F1 Score	0,70	0,61
AUROC	0,81	0,71

Ao observar ainda a tabela acima apresentada, nota-se que não existe uma discrepância significativa entre os valores da AUROC dos dois conjuntos de dados. Neste caso, em específico foi obtida uma AUROC de 0,71 no conjunto de dados de teste. Uma vez que a AUROC foi a métrica escolhida para avaliar a qualidade de um modelo e, nas situações descritas na secção 5.3, se obtiveram melhores valores desta métrica, não se considerou esta situação para prosseguimento do estudo.

Anexo E

Variável resposta da situação 2.1- Resultados dos modelos

De seguida, procede-se à apresentação dos resultados das métricas de avaliação obtidas para os dados de treino e teste para a variável resposta da situação 2.1, descrita da secção 4.1, para cada um dos algoritmos de modelação testados.

Neste caso, a variável dependente da situação 1 distribui-se da seguinte forma: em 66,19% dos eventos não se verifica decisão (0 - insucesso sem oferta apresentada, não contactado, cancelado, *callback* com oferta apresentada e *outcome* não comercial), em contrapartida, em 33,81% ocorre decisão (1 - sucesso, insucesso com oferta apresentada, *callback* decisor, agendado decisor).

Os valores das métricas de avaliação, relativas ao *Random Forest* são apresentados na Tabela E.1. No que se refere à exatidão (acurácia) das previsões que o modelo originou, verifica-se que este é capaz de classificar corretamente cerca de 68% das observações dos dados de teste. Desta forma, a percentagem de erro deste modelo é de 32%, pelo que o modelo classifica de forma incorreta cerca de 32% das observações dos dados de teste. Observa-se, ainda que, nos dados de teste, o modelo prevê, de forma correta, cerca de 61% das observações, que foram classificadas como positivas. No entanto, relativamente às observações que na realidade pertencem á classe positiva da variável resposta, o modelo apenas identifica cerca de 18% corretamente.

De facto, observa-se que a precisão das previsões positivas do modelo é superior ao da sensibilidade, pelo que o modelo origina uma maior quantidade de FN, do que FP nas suas previsões. Isto indica que, de forma geral, o modelo tem maior facilidade em identificar uma observação que é decisor como não sendo, do que o contrário.

Tabela E.1- Valores das métricas de avaliação obtidos nos dados de treino e teste, para o *Random Forest*

Métricas \ Dados	**Treino**	**Teste**
Acurácia	0,75	0,68
Precisão	0,85	0,61
Sensibilidade	0,3	0,18
F1 Score	0,44	0,28
AUROC	0,86	0,67

Na Figura E.1, encontram-se representadas as curvas de ROC obtidas para os dados de treino e teste, bem como as respetivas AUROC. Neste caso, em particular, nota-se existe uma diferença significativa entre as curvas de ROC e os valores da AUROC obtidos para os dados de teste e para os dados de treino. Desta forma, o valor de 0,67 de AUROC obtido para o conjunto de dados de teste demonstra que este modelo tem maior dificuldade em distinguir as classes da variável resposta, do que os restantes modelos apresentados nos anexos anteriores e na secção 5.3.

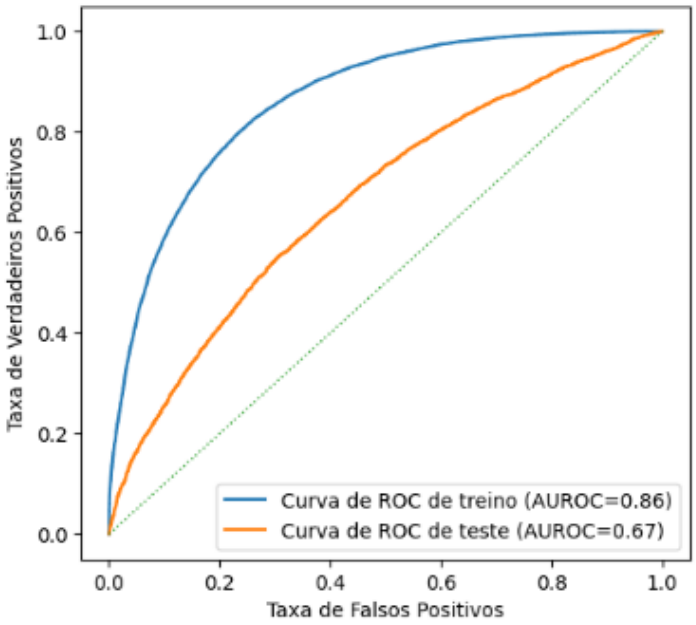

Figura E.1- Curva de ROC e respetiva AUROC do algoritmo *Random Forest*, para a variável resposta da situação 2.1

Quanto ao *Gradient Boosting*, os valores das métricas de avaliação são apresentados na Tabela E.2. Neste caso, relativamente à exatidão das previsões do modelo, verifica-se que este classifica corretamente cerca de 69% das observações presentes nos dados de teste, assim apresenta uma percentagem de erro de classificação de 31%. No que se refere à precisão do modelo, nota-se que das observações previstas como positivas, cerca 58% foram classificadas de forma assertiva. Porém, para as observações que na realidade pertencem à classe positiva da variável resposta, o modelo apenas identifica cerca de 18% corretamente.

Uma vez mais, nesta situação é percetível que o modelo tem maior tendência em classificar uma observação da classe positiva, como sendo da classe negativa da variável resposta. De facto, nota-se que o valor da precisão das previsões positivas é superior ao da sensibilidade, pelo que se conclui que o modelo origina uma quantidade de FN superior à de FP.

Tabela E.2- Valores das métricas de avaliação obtidos nos dados de treino e teste, para o *Gradient Boosting*

Métricas \ Dados	**Treino**	**Teste**
Acurácia	0,77	0,69
Precisão	0,84	0,58
Sensibilidade	0,39	0,24
F1 Score	0,53	0,34
AUROC	0,84	0,68

Na Figura E.2, encontram-se representadas as curvas de ROC obtidas para os dados de treino e teste, bem como as respetivas AUROC. Neste caso, em específico, constata-se que existe uma discrepância superior entre os valores da AUROC obtidos para os dados de teste e para os dados de treino. Desta forma, o valor de 0,68 de AUROC obtido para o conjunto de dados de teste demonstra que este modelo tem maior dificuldade em distinguir as classes da variável resposta, do que os restantes modelos apresentados nos anexos anteriores e na secção 5.3.

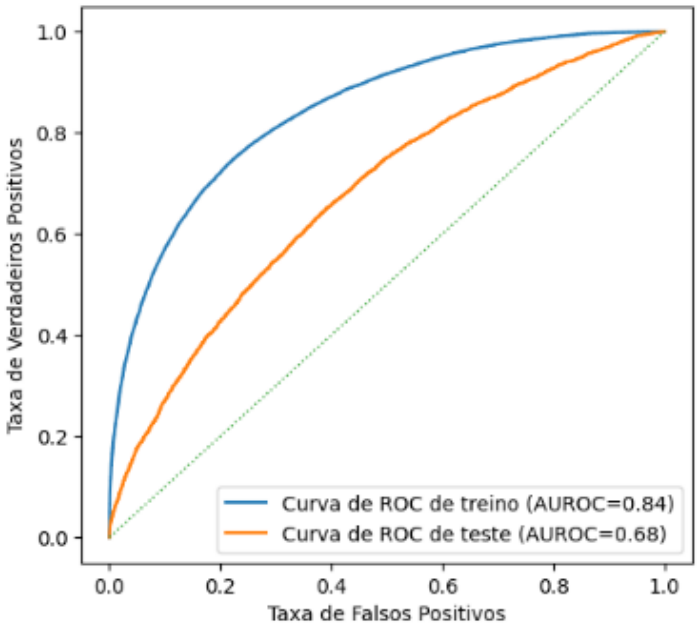

Figura E.2- Curva de ROC e respetiva AUROC do algoritmo *Gradient Boosting*, para a variável resposta da situação 2.1

Em suma, ao observar os valores que constam nas tabelas deste anexo, verifica-se que, no geral, os valores das métricas obtidas para o conjunto de dados de treino e de teste apresentam uma diferença significativa entre si, o que não é uma situação favorável. Para além disso, observa-se que os valores de AUROC para o conjunto de teste, em ambas as metodologias, foram inferiores aos das situações descritas anteriormente. Tendo em conta os motivos apresentados, não se considerou que esta formulação de variável resposta fosse a mais indicada para prosseguir no estudo.

Anexo F

Proposta para otimização do ponto de corte das probabilidades do modelo

Na Figura F.1, encontra-se representada a variação do comportamento das métricas de avaliação de precisão e sensibilidade para diferentes pontos de corte das probabilidades previstas pelo modelo. Tal como foi referido na secção 3.2.6, as duas métricas referidas têm um comportamento inverso, pelo que quando o valor de uma aumenta, o da outra diminui. Deste modo, verifica-se que, se for selecionado um *threshold* com valor reduzido, a sensibilidade é superior à precisão. Por sua vez, quanto mais próximo da unidade for este valor, maior será a precisão do modelo, em comparação com a sensibilidade.

Com efeito, existem situações onde se pretendem identificar como positivas todas as observações com classe atual positiva, sem que nenhuma seja rejeitada. Isto significa que é crucial reduzir ao máximo o número de FN (maior sensibilidade) podendo, assim, originar um elevado número de FP (menor precisão). Por outro lado, pode ser desejável obter, precisamente, a situação contrária. Desta forma, recomenda-se que a partir do gráfico abaixo representado e das conclusões retiradas dos pilotos, esta decisão seja tomada juntamente com a operação.

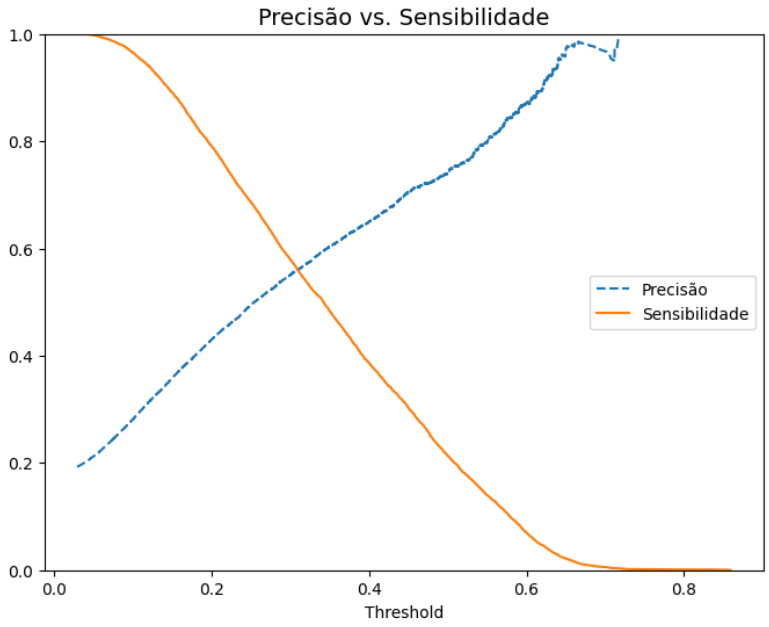

Figura F.1- Comportamento da precisão e sensibilidade do modelo para diferentes pontos de corte