Catarina Gonçalves Simões Nicolau Vicente

Licenciada em Engenharia Informática

Machine Learning Algorithms – Application on Big Data to Predict Retention Actions Needs

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática

Orientador:	Nuno Tiago Marujo da Silva Santos Pereira, Licenciatura em Engenharia Eletrotécnica e Computadores, worldIT - Consulting Services
Co-orientador:	Joaquim Francisco Ferreira da Silva, Professor Auxiliar, Universidade Nova de Lisboa

Júri

Presidente:	Doutor Jorg Matthias Knorr
Vogais:	Doutor João Carlos Gomes Moura Pires
	Licenciado Nuno Tiago Marujo da Silva Santos Pereira

FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Agosto, 2020

Machine Learning Algorithms – Application on Big Data to Predict Retention Actions Needs

Este documento foi gerado utilizando o processador (pdf)LaTeX, com base no template "novathesis" [1] desenvolvido no Dep. Informática da FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Agradecimentos

Desde já, manifesto o meu agradecimento a todos que, de algum modo, me ajudaram na realização desta dissertação.

Em primeiro lugar, quero expressar o meu agradecimento ao Orientador Nuno Tiago Pereira que, para além de me ter dado a oportunidade de integrar este projeto de investigação, me prestou todo o apoio ao longo da dissertação e me transmitiu, pacientemente, todos os conhecimentos teóricos e científicos sobre esta temática.

Agradeço, de igual forma, ao Professor Joaquim Silva toda a ajuda e disponibilidade fornecida e por me ter transmitido também o conhecimento necessário para entender a problemática desta dissertação.

Quero agradecer também ao Automóvel Club de Portugal (ACP) por me ter dado todas as condições para poder realizar este trabalho.

Por último, mas não menos importante, deixo um enorme agradecimento à minha família, especialmente aos meus pais, por todo o apoio, compreensão e paciência que tiveram para comigo ao longo do meu percurso académico.

Resumo

A utilização de técnicas de *Machine Learning* é cada vez mais comum em múltiplas aplicações práticas. Hoje em dia, os resultados da aplicação destas técnicas influenciam já de forma rotineira a nossa vida e tarefas quotidianas. Vídeos que nos são sugeridos para visualizar; qual o percurso que tomamos até ao nosso destino; o reconhecimento facial em sistemas biométricos e de segurança; todos são exemplos práticos dos avanços efetuados nesta área.

Muitos modelos de *Machine Learning* são *black box*, dada a complexidade dos problemas abordados e da sua natureza algorítmica e, por vezes, não oferecem uma perceção dos seus processos de decisão ou não são diretamente interpretáveis no que toca às razões que originam as suas previsões e resultados.

A utilização de Métodos Explicativos evidencia padrões nos dados, permitindo uma interpretação de resultados mais assertiva. Assim, esta dissertação pretende desenvolver um protótipo que combine técnicas de *Machine Learning* com os Métodos Explicativos de forma a melhorar a avaliação e validação de indicadores, tornando mais consistente e assertivo o processo de obtenção de resultados pelo algoritmo e de como ele é afetado. Do ponto de vista comercial, com base nos resultados obtidos pelos modelos, espera-se que a consequente definição ou reengenharia de estratégias possa obter melhores resultados operacionais e a melhoria contínua de indicadores.

Com este protótipo pretendo demonstrar que, do ponto de vista prático, a obtenção de indicadores representativos da permanência/fidelidade de clientes numa organização, aplicando técnicas de *Machine Learning* sobre dados reais, e usando métodos explicativos, uma vez interpretada a influência e peso das características dos dados sobre o/os modelo/os, é possível a redefinição e afinação de estratégias operacionais.

Especificamente, como caso prático desta dissertação, é esperado que sistemas corporativos como sejam sistemas de *Customer Relationship Management* possam beneficiar dos resultados desta dissertação através da aplicação das técnicas de *Machine Learning* e da interpretação dos Métodos Explicativos.

Palavras-chave: *Machine Learning*, *Customer Relationship Management*, Métodos Explicativos, Indicadores Probabilísticos

Abstract

The use of Machine Learning techniques is increasingly commonplace in multiple practical applications. Nowadays, the results of the application of these techniques are already routinely influencing our life and day-to-day tasks. Suggestions of videos to visualize; which route to take to a destination; facial recognition in biometric and security systems; all are practical examples of the advances made in this area.

Many Machine Learning models are black box, given the complexity of the problems addressed and their algorithmic nature and, sometimes, do not offer a perception of their decision-making processes or are not directly interpretable when it comes to the reasons that originate their forecasts and results.

The use of Explanatory Methods highlights patterns in the data, allowing a more assertive interpretation of results. Thus, this dissertation intends to develop a prototype that combines Machine Learning techniques with Explanatory Methods in order to improve the evaluation and validation of indicators, making the process of obtaining results by the algorithm and how it is affected more consistent and assertive. From a commercial point of view, based on the results of the models applied to the data, the consequent definition or reengineering of strategies obtains better operational results and the continuous improvement of indicators.

With this prototype I intend to demonstrate that, from a practical point of view, obtaining representative indicators of customer permanence/loyalty in an organization, applying Machine Learning techniques on real data, and using explanatory methods, once the influence and weight of the characteristics are interpreted from the data on the model/s, it will be possible to redefine and fine-tune operational strategies.

Specifically, as a practical case of this dissertation, it is expected that corporate systems such as Customer Relationship Management systems can benefit from the results of this dissertation through the application of Machine Learning techniques and the interpretation of Explanatory Methods.

Keywords: Machine Learning, Customer Relationship Management, Explanation methodology, Probabilistic indicators

Índice

Lista de Figuras

Lista de Tabelas

Siglas

ACP	Automóvel Club de Portugal
AUC	Area Under The Curve
B2B	Business to Business
CA	Correspondence analysis
CRM	Customer Relationship Management
CTI	Computer Telephony Integration
CV	Cross-Validation
DT	Árvores de Decisão
FAMD	Factor Analysis of Mixed data
FN	False Negative
FP	False Positive
FPR	False Positive Rate
INE	Instituto Nacional de Estatística
LLM	Logit Leaf Model
LMT	Logistic Model Tree
LR	Regressão Logística
MAE	Mean Absolute Error
MCA	Multiple Correspondence Analysis
MFA	Multiple Factor Analysis
ML	Machine Learning
MLP	Multi-Layer Perceptron
MSE	Mean Squared Error

NB	Naïve Bayes
NBTree	Naïve Bayes Tree Algorithm
PCA	Principal Component Analysis
RF	Random Forests
RGPD	Regulamento Geral de Proteção de Dados
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristics
SQL	Structured Query Language
SSMS	SQL Server Management Studio
SVC	Support Vector Classification
SVD	Singular Value Decomposition
SVM	Support Vector Machine
SVM-RFE	Recursive Feature Elimination
SVR	Support Vector Regression
TN	True Negative
TP	True Positive
TPR	True Positive Rate

Capítulo 1

Introdução

Este capítulo apresenta brevemente o trabalho proposto e a ser desenvolvido por esta dissertação, descrevendo os problemas levantados e apresentação de uma motivação para a solução proposta.

1.1 Contexto e Descrição

Numa definição inevitavelmente incompleta, a inteligência artificial pode ser vista como o estudo de agentes que percebem o mundo ao seu redor, formam planos e tomam decisões para alcançar os seus objetivos. Um agente é visto como um sistema que entende o ambiente e define ações de forma a potenciar o sucesso. Os seus alicerces incluem matemática, lógica, filosofia, probabilidade, linguística, neurociência e teoria da decisão. Os rápidos avanços na disponibilização e acesso a dados e o aumento do poder de processamento computacional, aceleraram a adoção de sistemas de inteligência artificial.

Machine Learning (ML) é um subcampo da inteligência artificial, sendo o seu objetivo central permitir obter resultados de forma autónoma através do processamento de dados históricos. Os algoritmos de ML permitem identificar padrões em dados, criar modelos e fazer previsões sem a necessidade de ter regras e modelos explícitos pré-programados. ML é um conceito vasto e é utilizado em aplicações que vão, por exemplo, desde a classificação de imagens, até à sugestão de combinações potencialmente mais eficazes de moléculas para a produção de fármacos, com vista a atingir os melhores resultados na cura de uma patologia clínica. A aplicação de algoritmos ou técnicas de ML pode ajudar no processo de compreensão de problemas complexos. Nomeadamente, se for interpretável, qual o contributo das características dos dados de treino, para os resultados obtidos na aplicação dos algoritmos.

Esta dissertação pretende promover a utilização de técnicas de ML em dados reais,

por forma a obter indicadores, produzindo paralelamente explicações sobre a razão da obtenção dos mesmos. Desta forma, espera-se ser possível fazer evoluir esses indicadores ao longo do tempo, avaliando resultados e previsões e redefinindo estratégias corporativas de modo a influenciar a evolução no sentido esperado.

Como aplicação mais específica, pretendo utilizar o protótipo na previsão do tempo de permanência/fidelização de um cliente (ou potencial cliente que tenha um histórico de interação e dados de perfil), a partir de dados dos seus hábitos de consumo anteriores, do seu perfil e das suas interações com a organização, no contexto duma organização comercial fornecedora de serviços e produtos. De futuro, os modelos poderão ser estendidos para o cálculo da oferta dos produtos e serviços mais adequados a oferecer aos clientes, tendo em conta dados adicionais aos dados usados para a previsão dos tempos de permanência.

Umas das funções de um sistema de Customer Relationship Management (CRM) é manter atualizada informação sobre o cliente, os serviços e produtos subscritos, interações com a organização e historial de consumo. Este conhecimento permite que as organizações possam agir e interagir de forma rápida, eficiente e contextualizada com o perfil de cliente, permitindo providenciar simultaneamente um melhor serviço de apoio, satisfação e retenção e, paralelamente, potenciar o aumento das vendas (e permanência como cliente ou sócio).

No contexto da presente dissertação, serão usados como caso de estudo dados provenientes na sua maioria do sistema de CRM do ACP. É de referir, no entanto, que se pretende que os objetivos a atingir nesta dissertação sejam aplicáveis a outros cenários no contexto de dados de CRM e não exclusivamente ao ACP, sendo que haverá sempre especificidades distintas para diferentes organizações.

Pretendo ter indicadores que avaliem o perfil de cliente num determinado momento, nomeadamente a sua propensão para ficar/sair. Com esta informação, para casos com alta probabilidade de saída, a organização poderá agir precocemente, de modo a contribuir para aumentar a retenção e eventuais vendas. No caso particular do ACP, os indicadores a obter deverão refletir e contribuir para decisões e ações da:

- Execução de ações preventivas e pró-ativas para evitar desistências ou insatisfação do cliente.
- Avaliação de cenários, de forma precoce, de modo a que no momento do contacto com o cliente os resultados do serviço e/ou das vendas sejam mais eficazes.

1.2 Motivação

A aplicação de técnicas de ML pretende fazer previsões o mais assertivas possíveis, tendo como ideia principal que através do processamento de grandes quantidades de dados poderão ser obtidos resultados que, uma vez conhecidos, permitam agir, de forma mais

informada, na tomada de decisões e melhorar o desempenho de processos. Numa situação ótima o sistema deve ser capaz de generalizar com base no histórico de dados passados, ou seja, obter resultados de previsão com o menor desvio possível, com base em dados de entrada não conhecidos anteriormente.

De forma geral, embora o processamento de quantidades muito elevadas de dados possam ser um desafio, uma vez que é necessário poder computacional também proporcionalmente elevado, as técnicas de ML serão naturalmente mais úteis e fiáveis na medida em que a quantidade dos dados disponíveis para o treino dos algoritmos aumenta. Nomeadamente, na análise de problemas complexos. De ressalvar que, a quantidade de dados não é garantia de sucesso, a sua qualidade é crítica para obtenção de resultados eficazes.

Algumas vantagens da utilização de técnicas de ML em aplicações práticas são:

- **Aumento da produtividade** – As técnicas de ML permitem a criação de sistemas capazes de analisar grandes quantidades de dados e realizar cálculos complexos com exatidão e velocidade. Isso permite, entre outros aspetos, identificar padrões que passam despercebidos aos seres humanos. O aumento da velocidade com que os problemas são identificados e corrigidos é também uma das principais motivações pelo crescente interesse do uso de ML. Por exemplo, em muitas ocasiões é gasto mais tempo na identificação do problema do que na solução em si. A utilização de técnicas de ML pode acelerar a obtenção de resultados e economizar tempo na execução de processos. O aumento da produtividade tem como consequência a redução de custos.

- **Mais assertividade na tomada de decisões estratégicas** – A utilização de algoritmos de ML permite fundamentar o processo de tomada de decisões por meio da previsão de cenários a curto, médio e longo prazo.

- **Personalização de serviços** – É possível identificar a preferência, bem como ajustar os conteúdos e produtos mais indicados a um utilizador, analisando os dados e o histórico do mesmo. No entanto, há que ter em conta que a personalização de serviços pode igualmente ser encarada como uma desvantagem. Por exemplo, tendo em conta que o recomendado é semelhante ao que o utilizador consumiu, a longo prazo estas decisões podem levar ao desinteresse deste, pois deixa de existir diversidade dos conteúdos que lhe são apresentados.

1.3 Objetivos Finais da Dissertação

O ACP é uma organização de referência a nível nacional, sendo o maior e mais antigo Clube de Portugal, com um sistema de CRM com mais de 10 anos de registos de interações e informação dos seus sócios. O objetivo geral desta dissertação é, partindo de informação recolhida do sistema de CRM, providenciar indicadores de previsão que permitam

antecipar medidas para a retenção de sócios e melhoria de serviço e vendas. O conhecimento e fundamentação destes indicadores permitirão redefinir estratégias internas de ação no ACP. Serão assim testados e comparados diferentes algoritmos e, posteriormente, executado o de melhor resultado nas previsões/resultados para os indicadores a trabalhar. Neste processo é/foi crítico a salvaguarda das regras de proteção de dados (Regulamento Geral de Proteção de Dados (RGPD)) e a anonimização.

Em suma, a seguinte pergunta deve ser respondida:

- Como podemos obter e melhorar indicadores de previsão sobre dados reais e obter explicações sobre estes que nos permitam redefinir estratégias internas de ação?

1.4 Contribuições Chave

Pretende-se que o trabalho desenvolvido durante esta dissertação possa prever, de modo mais assertivo, os seguintes indicadores específicos e que fatores os influenciam:

- Qual será o ESTADO previsto de um sócio/*prospect* mediante a alteração de características (por exemplo: novo produto subscrito/retirado)?
- Durante quanto tempo um sócio permanecerá associado do ACP (previsão da antiguidade)?

1.5 Estrutura do Documento

A estrutura do documento é detalhada na seguinte lista:

- Capítulo 1 – Introdução: este capítulo;
- Capítulo 2 – *Background*: uma descrição dos conceitos e tecnologias pilares para esta dissertação;
- Capítulo 3 – Trabalho relacionado: uma análise e descrição de abordagens que visam atingir objetivos semelhantes a esta dissertação;
- Capítulo 4 – Metodologias para a Solução – uma apresentação/justificação das metodologias usadas com vista a alcançar os objetivos desta dissertação;
- Capítulo 5 – Resultados: uma apresentação dos resultados da solução desenvolvida.
- Capítulo 6 – Conclusões e Trabalho Futuro: uma apresentação das conclusões retiradas da solução desenvolvida e uma apresentação dos possíveis trabalhos futuros.

Capítulo 2

Background

Este capítulo estabelece o domínio desta dissertação, detalhando alguns conceitos chave no seu campo de estudo.

2.1 Tipos de *Machine Learning*

O conceito de ML reflete a capacidade de, artificialmente, se obter resultados de forma coerente com o que foi previamente aprendido em fase de treino. Esta fase de treino pode assumir diferentes tipos, conforme descrito de seguida.

2.1.1 *Supervised Learning*

Tal como o nome sugere, *Supervised Learning* necessita que os objetos (amostra) fornecidos para treino contenham também a identificação da classe/rótulo a que pertencem. É fornecido ao algoritmo uma quantidade de amostras variada e com várias características/atributos, para além, da já referida classe/rótulo.

Em termos gerais, *Supervised Learning* pode ser utilizado para resolver dois tipos de problemas:

- **Classificação** (quando o objetivo é prever a que classe, de um conjunto discreto, cada novo objeto pertence) – O objeto não utilizado no treino, é categorizado como pertencente a um dado grupo. Cada objeto é associado a um valor qualitativo que corresponde a uma classe. Por exemplo, estando num cenário em que é necessário identificar células benignas e malignas, estamos perante um problema de classificação. No entanto, a classificação não se limita ao caso binário; podem ter mais do que duas classes.

- **Regressão** (quando o objetivo é prever valores contínuos) – Neste caso, cada objeto não tem associada, como resultado, uma classe (valor categórico) mas sim um valor quantitativo. O objetivo do modelo é estimar o resultado correto, quando em presença de um novo objeto, dado um conjunto de características. Estes problemas pretendem encontrar uma linha ou uma curva que se generalize bem para a maioria dos pontos. Por exemplo, estando num cenário em que é necessário determinar a esperança média de vida de pacientes com cancro, estamos perante um problema de regressão.

Alguns problemas de regressão podem ser transformados num problema de classificação. Além disso, certos problemas podem pertencer a ambos os tipos. A Figura 2.1 mostra exemplos de classificação e regressão.

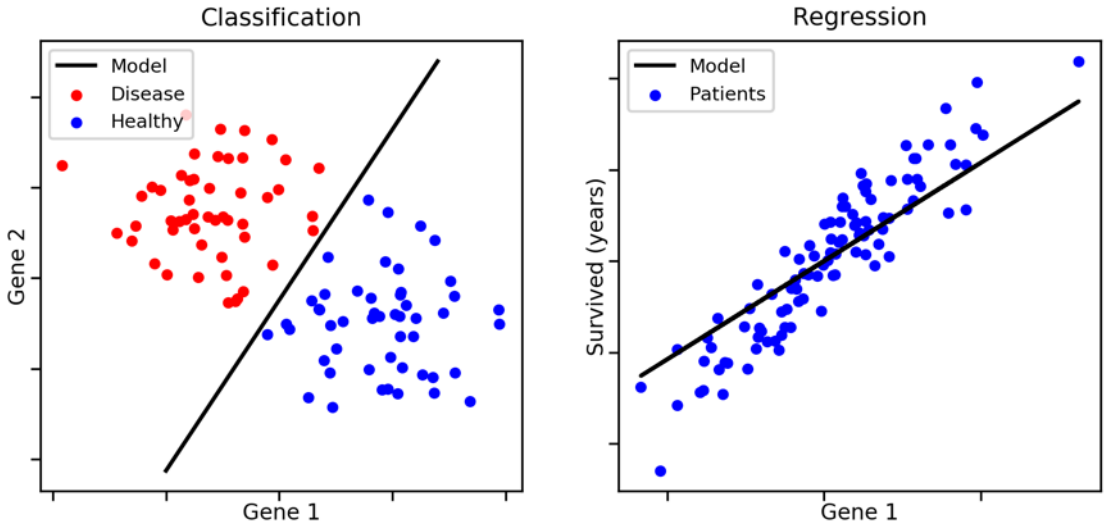


Figura 2.1: Exemplo dos problemas de classificação e regressão.

2.1.2 *Unsupervised Learning*

Unsupervised Learning infere padrões de um conjunto de dados sem referência a resultados conhecidos ou rotulados. Ao contrário do *Supervised Learning*, não pode ser diretamente aplicado a um problema de classificação ou regressão porque não são conhecidos os valores de classe/resultado dos dados. Os dados de treino não têm, pois, rótulo. O algoritmo identifica padrões de forma autónoma. Esta técnica mostra-se muito útil e necessária, sobretudo quando temos uma estrutura de dados tão complexa que é difícil detetar esses mesmos padrões de forma a identificar classes/grupos.

2.1.3 *Semisupervised Learning*

São fornecidos dados de treino parcialmente rotulados, ou seja, dados não rotulados e dados rotulados. A maioria dos algoritmos de *Semisupervised Learning* são combinações

de algoritmos de *Supervised* e *Unsupervised Learning*.

2.1.4 *Reinforcement Learning*

Este tipo de ML é bastante diferente dos últimos três tipos descritos. Neste sistema de aprendizagem, temos um agente que observa o ambiente, seleciona e executa ações com o objetivo de obter recompensas ou penalidades (recompensas negativas) em troca. O agente é treinado para aprender a escolher a melhor ação num determinado cenário, com base nas recompensas e *feedback*. Ou seja, o propósito é aprender qual a melhor estratégia para receber a maior recompensa ao longo do tempo.

2.2 Principais etapas de *Machine Learning*

Muitos dos problemas abordados com técnicas de ML, em particular recorrendo a *Supervised Learning* seguem, de forma geral, as seguintes etapas:

1. **Recolha de dados**

   Com base no objetivo final, esta consiste numa das primeiras etapas, que passa por medir informação e/ou recolher os dados. Os dados recolhidos serão utilizados como dados de treino. É uma etapa crítica, pois a quantidade e qualidade dos dados recolhidos está normal e diretamente relacionada com a qualidade posteriormente obtida no modelo preditivo. Este passo exige muitas vezes uma preparação e implementação cuidadas, pois em muitas circunstâncias é necessário implementar equipamento e infraestrutura de recolha, que podem ir desde sensores físicos, redes de comunicação, até sistemas aplicacionais e formação de recursos humanos.

2. **Análise e preparação dos dados**

   Após a recolha dos dados de treino, segue-se a análise e preparação dos dados. É quase sempre indispensável entender, processar e aplicar transformações. Note-se que uma parte muito significativa do tempo despendido na aplicação de técnicas de ML é gasto na análise e preparação dos dados.

   É muitas vezes necessário fazer uma redução de caraterísticas/atributos. Se for possível reduzir o número de características quando estas são muito correlacionadas, é possível obter melhores desempenhos na aplicação dos algoritmos, para a mesma quantidade de dados. Este passo deve ser cuidadosamente efetuado pois pode ter um efeito significativo sobre os resultados dos modelos.

   Além da verificação, transformação e garantia da qualidade dos dados, existe um passo muito importante que consiste na preparação destes para a sua aplicação nos modelos. O treino dos modelos com os dados exige a pré separação de conjuntos a partir do conjunto inicial, para que sejam posteriormente utilizados nas fases de teste e validação dos modelos, devendo a separação manter a sua significância.

Nomeadamente, os dados devem ser disponibilizados aos algoritmos de forma aleatória, uma vez que a ordem não deve afetar o modelo, e divididos em pelo menos dois conjuntos: um conjunto que contém os dados de treino e outro conjunto que contém os dados de validação.

3. **Escolha do modelo**

   Nesta etapa é escolhido um algoritmo e modelo, tendo em conta a natureza dos dados e o tipo de problema que se pretende resolver. Esta é uma etapa importante, no entanto a sua execução é rápida, a não ser que seja necessária a implementação do algoritmo de raiz.

4. **Treino dos dados**

   Neste passo, são usados os dados de treino. No treino supervisionado os dados estão catalogados sobre qual o resultado/classe associado a estes dados. Pretende-se assim que o sistema aprenda as características (o perfil) de cada grupo (classe) de modo a que consiga posteriormente classificar futuros objetos, com base no conhecimento adquirido.

5. **Avaliação**

   Após a fase de treino estar concluída, o modelo será avaliado de forma a qualificar a sua qualidade. Para tal, será usado o conjunto de dados de validação, já separado anteriormente. Assim, é possível testar o modelo com dados que nunca foram utilizados na fase de treino. Esta validação permite analisar o comportamento do modelo com dados que ainda não foram usados. Isto deve ser representativo do comportamento do modelo no mundo real.

6. **Ajuste de parâmetros**

   Depois da fase de avaliação, é possível que se queira melhorar o modelo. Para tal, os parâmetros podem ser ajustados e o modelo voltar a ser testado. Este pode muitas vezes ser considerado um processo experimental.

7. **Previsão e manutenção**

   ML com treino supervisionado é na sua essência a tentativa de, através da inferência de cenários passados e variados cujo resultado é conhecido, usar esse conhecimento para prever novos resultados face a cenários presentes. Assim, neste último ponto, o modelo treinado é utilizado para prever os resultados desejados, sendo importante manter o sistema ao longo do tempo e realimentar novos resultados de forma sistemática. Este passo pode igualmente consumir recursos significativos, tanto a nível material como humano, pois exige a adaptação/alteração de processos e sistemas.

2.3 Principais desafios de Machine Learning

Conforme já referido, a principal tarefa na aplicação de técnicas de ML é selecionar um algoritmo, modelá-lo e treiná-lo com dados catalogados. Neste contexto, alguns pontos típicos que podem correr mal são a seleção de um algoritmo não adequado, a má qualidade ou má preparação dos dados.

2.3.1 Quantidade insuficiente dos dados de treino

Para regressões lineares, algoritmos relativamente simples serão suficientes para fazer previsões de certas variáveis. As técnicas de ML são normalmente utilizadas em cenários de resultados não lineares e variados de complexidade elevada. Assim, por norma é necessária uma quantidade elevada de dados para que a maioria dos algoritmos de ML funcionem corretamente. Mesmo quando se trata de um problema mais simples, podem ser necessários muitos exemplos; para problemas mais complexos, como reconhecimento de voz ou imagem, é necessária uma quantidade muito superior de exemplos (com a exceção de quando se reutiliza partes de um modelo já existente).

2.3.2 Dados de treino não representativos

É de extrema importância que os dados de treino sejam representativos dos vários casos que se deseja generalizar. Ao utilizar um conjunto de dados não representativos, o modelo que está a ser treinado tem menos probabilidades de fazer previsões precisas. Assim, ter um conjunto de dados que seja representativo pode ser um grande desafio. Se a amostra de dados for pequena, podemos ter ruído de amostragem (dados não representativos como resultado do acaso), mas também amostras muito grandes podem não ser representativas se o método de amostragem não for perfeito ou contiver erros.

2.3.3 Qualidade insuficiente dos dados

Se os dados de treinos contiverem muitos erros, valores discrepantes e ruído, será mais difícil para o sistema detetar os padrões subjacentes, diminuindo assim a probabilidade de o sistema ter um bom desempenho. Assim, é importante garantir a quantidade e qualidade dos dados utilizados no treino dos modelos.

2.3.4 Características irrelevantes

Um sistema tenderá a ter uma aprendizagem de melhor qualidade se os dados de treino contiverem características relevantes suficientes e poucas irrelevantes. A parte crítica do sucesso de um projeto de ML envolve ter um bom conjunto de dados com características relevantes. Para chegar a esse ponto, geralmente, o processo envolve os seguintes passos: selecionar, de entre as características existentes, as mais úteis para treinar; combinar

características de forma a eliminar possíveis redundâncias; usar novas características reunindo novos dados de fontes adicionais.

2.3.5 *Overfitting*

O *overfitting* é um problema comum na aplicação de técnicas de ML: o modelo adquire conhecimento, tem um bom desempenho e explica perfeitamente os dados de treino que o modelo recebeu, no entanto, não generaliza bem para os dados de teste. Ocorre quando o modelo tem uma alta sensibilidade aos dados de treino, podendo mesmo acabar por obter resultados determinados por características que não são representativas dos padrões do mundo real.

2.3.6 *Underfitting*

O *underfitting* é o oposto do *overfitting*: o modelo não foi treinado com dados representativos ou com qualidade suficiente para poder capturar e generalizar a partir destes, mesmo se estes forem em grande quantidade. Como resultado, o modelo apresenta uma baixa precisão na previsão.

Capítulo 3

Trabalho relacionado

Neste capítulo, são apresentados/descritos conceitos e tecnologias e são mencionadas algumas abordagens que, de certa forma, estão relacionadas com esta dissertação, seja por partilharem objetivos semelhantes ou por empregarem ferramentas e técnicas que pretendemos explorar.

3.1 Conceitos/Tecnologias chave

Apresentação/descrição de conceitos e tecnologias que estão relacionados com os artigos/trabalhos apresentados na Secção 3.2.

3.1.1 *Support Vector Machine* (SVM)

O SVM é uma abordagem de ML muito versátil e eficaz, podendo ser usada para tarefas tanto de regressão (linear ou não linear), como de classificação, embora seja maioritariamente utilizada em tarefas de classificação. É um modelo particularmente adequado para a classificação de conjuntos de dados complexos, mas de pequena ou média dimensão. Produz previsões de bom nível com menos recursos de computação.

A principal desvantagem do SVM é que gera o modelo *black box*, ou seja, o conhecimento aprendido pelo SVM durante o treino não é diretamente interpretável por um humano. O processo de conversão desses modelos opacos num modelo transparente é frequentemente considerado como extração de regras.

No SVM, o procedimento de treino é baseado na teoria de aprendizagem estatística [1]. É bastante usado em diversas aplicações, como análise de genes [2], previsão de séries temporais financeiras [3], reconhecimento facial [4], entre outras.

Para problemas de classificação, o principal objetivo do SVM é encontrar um hiperplano num espaço N-dimensional, em que N representa o número de características, com

separação ideal e que, tanto quanto possível, classifique as amostras de dados e as separe em classes, minimizando o risco de classificar de forma errada as amostras de treino e as de teste. Os hiperplanos são vistos como limites de decisão que ajudam na classificação das amostras de dados. A dimensão do hiperplano depende do número de características.

Existem muitos hiperplanos possíveis que podem ser escolhidos para a separação das classes de pontos. Desta forma, o objetivo é encontrar um plano que melhor divida duas classes dadas, maximizando a distância entre a fronteira e os vetores de suporte. Por outras palavras, a distância máxima entre amostras de dados de ambas as classes. Maximizar a distância da margem permite que as futuras amostras possam ser classificadas com mais confiança. Os vetores de suporte são pontos que correspondem a mostras que estão mais próximas do hiperplano de separação das classes, permitindo a maximização da margem do classificador. Estes pontos influenciam a posição e orientação do hiperplano. A Figura 3.1 apresenta uma visão geral do processo.

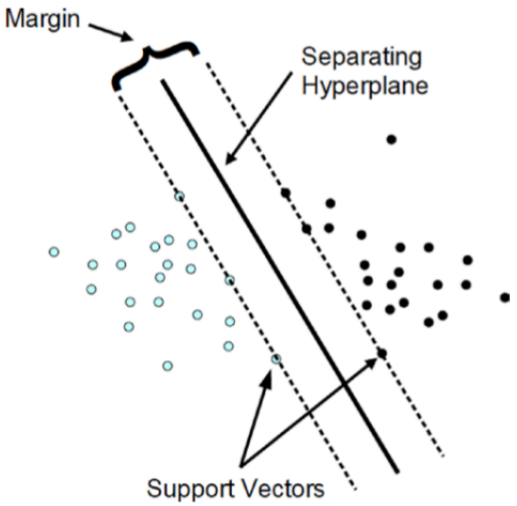


Figura 3.1: Visão geral da classificação SVM bidimensional.

De forma a lidar com problemas de conjuntos de dados que não são linearmente separáveis, pode-se utilizar com o SVM a adição de características dimensionais que são funções polinomiais de outras características, permitindo assim à abordagem encontrar uma margem, como está representado na Figura 3.2.

3.1.2 *Neural Network*

É uma estrutura de aprendizagem que utiliza uma rede de funções para entender e traduzir um *input* de dados num *output* desejado. O conceito de *Neural Network* foi inspirado na biologia humana e na maneira como os neurónios do cérebro humano funcionam juntos para entender os *inputs* dos sentidos humanos [5].

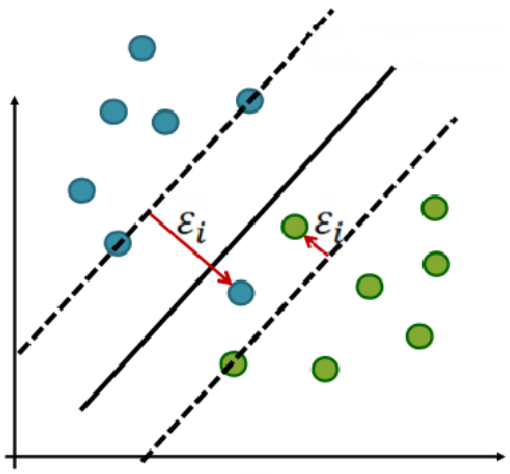


Figura 3.2: Conjunto de dados não linearmente separável.

3.1.2.1 *Multi-layer Perceptron (MLP)*

MLP é um algoritmo de aprendizagem supervisionado que aprende uma função através do treino de um conjunto de dados. Dado um conjunto de *features* (atributos) e um objetivo, o algoritmo pode aprender uma função aproximada não linear, tanto para classificação como para regressão. É diferente da regressão logística pois, entre a camada de *input* e a de *output*, pode haver uma ou mais camadas não lineares, chamadas as *hidden layers* (camadas ocultas). De forma genérica, as redes neuronais organizam-se de acordo com o modo apresentado na Figura 3.3.

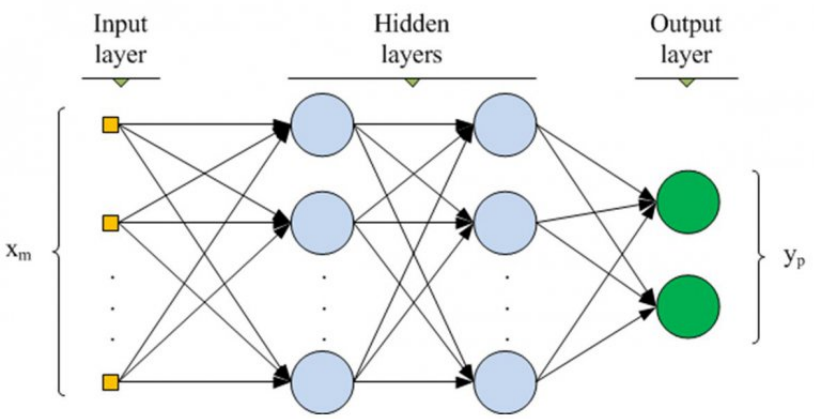


Figura 3.3: Diagrama geral de uma *Neural Network*.

A camada de *input*, *input layer*, consiste num conjunto de neurónios, que representam as *input features*, ou seja onde se introduzem os dados a serem processados. A camada *hidden layer*, que pode ter uma ou mais camadas de modo a processar os dados provenientes da camada de *input*. A camada de *output* recebe os valores da camada *hidden layer* e transforma-os em valores de output.

3.1.3 *Random Forest (RF)*

RF é um tipo de algoritmo de ML supervisionado e baseado na aprendizagem através de conjuntos [6]. Este é um tipo de aprendizagem onde se utiliza diferentes tipos de algoritmos ou o mesmo algoritmo várias vezes, de modo a formar um modelo de previsão mais robusto. O algoritmo RF é composto por várias árvores de decisão resultando numa floresta de árvores ,sendo que não calcula simplesmente a média das previsões de árvores. De um modo geral, o algoritmo é representado tal como mostra a Figura 3.4. Este algoritmo pode ser utilizado tanto para tarefas de regressão como de classificação.

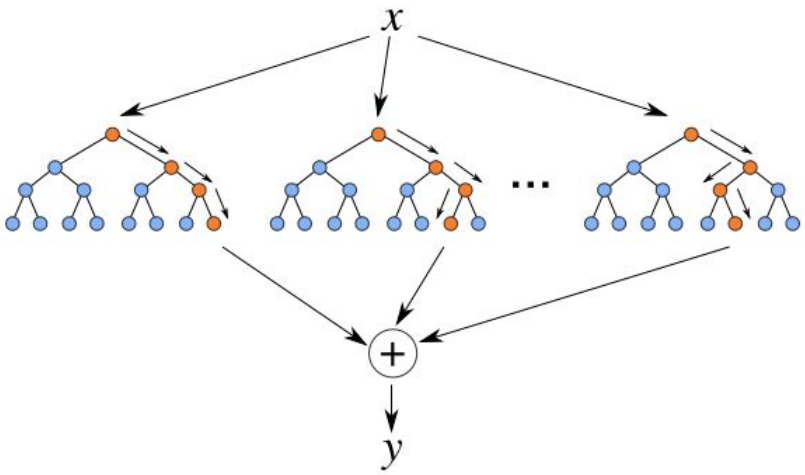


Figura 3.4: Diagrama geral de um *Random Forest*.

De modo a justificar o nome aleatório, este algoritmo usa dois conceitos principais:

1. Amostragem aleatória de dados de treino ao construir as árvores. Durante o treino, cada árvore numa floresta aleatória aprende com uma amostra aleatória dos dados. As amostras são desenhadas com substituição, fazendo com que algumas amostras sejam usadas várias vezes numa única árvore. Assim, ao treinar cada árvore com amostras diferentes, apesar de cada árvore ter hipótese de ter alta variação em relação a um conjunto específico de dados de treino, no geral, toda a floresta terá menor variação. No momento do teste, as previsões são feitas pela média das previsões de cada árvore de decisão.

2. Subconjuntos aleatórios de *features* considerados ao dividir nós. Apenas um subconjunto das *features* do conjunto de dados é considerado para dividir os nós nas árvores de decisão. Geralmente, o número de *features* a considerar é definido por *sqrt(n_features)*, o que significa que, por exemplo, se houver 16 *features* em cada nó de cada árvore, apenas 4 *features* aleatórias serão consideradas para dividir o nó.

Como em qualquer algoritmo, há vantagens e desvantagens na sua utilização. O algoritmo RF não é tendencioso, pois existem várias árvores e cada árvore é treinada com um

subconjunto de dados, o que faz com que a parcialidade geral do algoritmo seja reduzida. Quando é introduzido um novo ponto de dados no conjunto de dados, o algoritmo geral não será muito afetado, porque novos dados podem fazer impacto numa árvore, mas muito dificilmente irão impactar todas as árvores, o que torna o algoritmo muito estável. A grande desvantagem do algoritmo RF reside na sua complexidade, pois necessita de muitos recursos computacionais devido ao grande número de árvores de decisão geradas. Assim, este algoritmo requer muito mais tempo para treinar do que outros algoritmos comparáveis.

3.1.4 *Recursive Feature Elimination* (SVM-RFE)

O algoritmo Recursive Feature Elimination (SVM-RFE) [2] é utilizado com o objetivo de selecionar características para serem eliminadas, antes do treino do modelo. É reconhecido como um dos métodos mais eficazes de filtragem e, pode ser escolhido com o intuito de evitar *overfitting* quando existe um grande número de características.

Neste algoritmo, o conjunto de dados é usado para treino com a abordagem linear SVM para produzir um vetor de ponderação para cada característica, informação que é usada para eliminar algumas das características consideradas irrelevantes.

A seleção de características usando o algoritmo SVM-RFE reduz assim a dimensão dos dados, apresentando apenas as principais características, o que poderá acelerar posteriormente o processo de treino.

3.1.5 *Costumer Relationship Management* (CRM)

O CRM é um conceito que é aplicado operacionalmente nas organizações através de software e processos dedicados. É utilizado para registar a informação de perfil, interações, histórico de produtos adquiridos, situação e histórico de faturação, tanto de clientes como (se possível) de potenciais clientes. Tem como objetivo tanto garantir o melhor serviço, responder às necessidades e aprofundar o conhecimento sobre os clientes, como melhorar as vendas utilizando metodologias de marketing assentes no conhecimento dos sistemas de CRM [7].

A gestão efetiva da informação de cliente e agregação de múltipla informação da organização relacionada (produtos, recursos logísticos e humanos, concorrência, etc.) é crítica para o conceito de CRM para, por exemplo, adaptação de produtos e inovação de serviços, melhoria de tempos de resposta e operacionalização de campanhas eficientes.

O CRM pode ser focado em várias áreas: operacional, analítico, colaborativo e estratégico.

- **CRM operacional** - Sistemas operacionais que garantem o serviço de atendimento, a resolução de problemas e o acesso a dados por agentes de *front* e *back office*, entre outros.

- **CRM analítico** - Baseia-se na exploração da informação associada ao cliente. Através de *Data Mining* [1], é possível interrogar os dados e obter respostas a questões como: *Quem são os clientes mais valiosos?*; *Quais são os clientes que têm maior propensão a mudar para a concorrência?*; *Que clientes seriam mais propensos a aderir a determinada oferta?*.

- **CRM colaborativo** - Garante o correto fluxo de informação de oportunidades de negócio, recolhidas e registadas ao interagir com clientes potenciais, a membros de diferentes departamentos. Essa informação é compartilhada de modo a que a possam utilizar, de forma adequada, em ações de *marketing*, vendas, desenvolvimentos de produto ou contabilidade.

- **CRM estratégico** - É focado no desenvolvimento de uma cultura de negócios centrada no conhecimento obtido pelas áreas de CRM descritas acima. Operação, produto, atendimento e vendas podem ser consideradas como áreas de ação, para as orientações empresariais dependentes das decisões estratégicas.

3.1.6 *Predictive Churn Model*

As organizações de serviços devem ser pró-ativas na medição e compreensão dos níveis atuais de satisfação dos clientes. O abandono de clientes representa um problema real, pelo facto de estarmos numa era em que os mercados estão cada vez mais saturados por uma oferta a nível global, que intensifica a competição entre empresas. A utilização da informação histórica do cliente é uma das ferramentas mais importantes para combater a rotatividade de clientes [8].

Com vista a reduzir esta rotatividade, uma estratégia de retenção passa pela pesquisa e identificação de clientes que mostram uma alta propensão para abandonar a condição de cliente [8], [9]. A análise à rotatividade de clientes pode ser abordada de dois ângulos diferentes. Por um lado, o objetivo é melhorar os modelos de previsão de rotatividade, em que os modelos são desenvolvidos e propostos com o propósito de aumentar o desempenho preditivo [10], permitindo agir antes do abandono. Por outro lado, o objetivo é perceber o que leva à rotatividade e detetar fatores importantes de rotatividade, como a satisfação do cliente [11], [12].

Churn pode ser definido de forma ligeiramente diferente por cada organização ou produto. O tema de *Churn Prediction* engloba uma parte muito importante do CRM sendo conhecido que, por várias razões, é muito mais lucrativo manter e satisfazer clientes existentes do que atrair novos:

1. Empresas de sucesso têm clientes de longo prazo, permitindo assim que estas se concentrem nas necessidades dos seus clientes, em vez de procurar novos e potencialmente pouco lucrativos [13].

[1] Processo analítico projetado para explorar grandes quantidades de dados, com o objetivo de encontrar padrões entre variáveis

2. Clientes que saem da empresa podem influenciar outros clientes a terem a mesma ação [14].

3. Clientes de longo prazo têm efeitos benéficos. Tendem a comprar mais e a encaminhar as pessoas para a empresa através de um *feedback* positivo e são menos dispendiosos para servir, uma vez que a empresa já possui as suas informações e conhece as suas necessidades [8].

4. Ações de *marketing* competitivas têm menos efeito sobre clientes de longo prazo [15].

Estes efeitos fazem com que o custo de manter um cliente existente seja muito menor do que adquirir um novo [16].

Normalmente, os clientes que deixam de usar um serviço ou produto durante um determinado período de tempo são chamados de *churners*. De uma perspetiva da aplicação de técnicas de ML, o *churn* pode ser definido como um problema de classificação binária.

O problema de *Churn Prediction* é considerado como uma das aplicações mais importantes do CRM analítico e, por exemplo, usando as informações extraíveis a partir do SVM, os fornecedores de serviços podem obter intuições claras e eficientes sobre os seus clientes e podem construir melhores políticas para os reter. As Árvores de Decisão (DT) e a Regressão Logística (LR) são dois algoritmos muito populares para estimar uma probabilidade de rotatividade, porque combinam bom desempenho preditivo com boa compreensão [17]. Embora ambas as técnicas sejam úteis e tenham as suas vantagens, estas também apresentam desvantagens. As DT lidam muito bem com efeitos de interação entre variáveis, mas têm dificuldades para lidar com relações lineares entre variáveis. A LR lida muito bem com relações lineares entre variáveis, mas apresenta dificuldades com os efeitos de interação entre variáveis. O RF também tem sido usado em vários estudos de *Churn Prediction*, onde se comprovou que alcança um excelente desempenho preditivo [18].

3.1.7 *Naïve Bayes Tree Algorithm* (NBTree)

O algoritmo Naïve Bayes Tree Algorithm (NBTree) é um híbrido do algoritmo Naïve Bayes (NB) [19] e do algoritmo DT [20]. O conhecimento aprendido é representado na forma de árvore, sendo que esta é construída recursivamente. Os nós da folha são categorizadores de NB [21]. Para limitar a medida de entropia, um *threshold* é escolhido para atributos contínuos.

A utilidade de um nó é encontrada através do cálculo da discretização dos dados e pelo cálculo da estimativa da exatidão de *5-fold cross validation* usando NB no nó. A utilidade de uma divisão é a soma ponderada de utilidade dos nós. Para evitar divisões com pouco valor, uma divisão é considerada significativa se a redução relativa no erro for maior que 5% e houver um mínimo de 30 instâncias no nó [21].

Para atributos de valor discreto, o método NB apresenta um bom desempenho. Com o aumento no tamanho dos dados, o desempenho também melhora. O NBTree é útil para grandes quantidades de dados [21]. No entanto, no caso de atributos de valor contínuo, o método NB não leva em consideração as interações de atributos. Por outro lado, as DT não têm um bom desempenho quando a quantidade de dados é muito grande. Essas lacunas são superadas pelo algoritmo NBTree [22].

NBTree é utilizado com o objetivo de geração de regras [21].

3.1.8 Métodos Explicativos

Para explicar os modelos de classificação e as suas previsões foram introduzidos dois métodos gerais por [23], [24]. Ambos os métodos são baseados na ideia de que a importância de uma característica ou de um grupo de características num modelo específico pode ser estimada, simulando a falta de conhecimento sobre os valores da(s) característica(s). Portanto, os métodos contrastam o resultado de um modelo usando todas as características com o resultado obtido usando apenas um subconjunto delas. O *output* dos métodos é uma decomposição das previsões dos modelos de ML nas contribuições individuais das características. As explicações geradas seguem de perto o modelo aprendido e permitem a visualização da decisão de cada instância separadamente.

Enquanto que os modelos de previsão simples, como DT e regras de decisão, são autoexplicativos se forem relativamente pequenos, os modelos mais complexos exigem técnicas de explicação específicas do modelo ou métodos gerais de explicação.

Dois métodos de explicação são os métodos EXPLAIN [23] e IME [25] que podem ser aplicados a qualquer modelo de previsão, o que os torna uma ferramenta útil tanto para os interpretar como para comparar os seus diferentes tipos. A principal técnica que estes métodos utilizam reside na análise de sensibilidade: alterar o *input* do modelo e observar mudanças no *output*.

A componente chave dos métodos de explicação geral é a previsão condicional esperada: a previsão para a qual apenas um subconjunto das características de *input* é conhecido. Calcular a contribuição de uma característica de *input* usando a previsão condicional esperada é comum a todos os métodos gerais de explicação. No entanto, os métodos diferem em quantos e quais os subconjuntos de características que têm em conta e como combinam as previsões condicionais. O método EXPLAIN calcula a contribuição de uma característica de *input* omitindo apenas essa característica, e o método IME considera todos os subconjuntos de características. O método EXPLAIN não deteta as dependências entre características. Isto é resolvido pelo método IME, no entanto, para grandes conjuntos de dados, este método pode ser lento e ter que ser pré-computado para ser usado interativamente. As interações entre características são detetadas, mas não são expressas diretamente na visualização, pois o método não o permite. Desta forma, o utilizador precisa de descobrir manualmente o tipo de dependência com a análise interativa. As explicações seguem de perto o modelo de previsão, ou seja, se o modelo estiver errado

ou tiver um mau desempenho, isso será refletido pelas explicações.

3.1.9 *Data-driven decision-making*

Data-driven decision-making consiste na prática de fundamentar as decisões na análise de dados e não na intuição [26]. De acordo com os artigos [27], [28], com o uso de *data-driven decision-making*, as empresas no primeiro terço da sua existência são, em média, 5% mais produtivas e 6% mais lucrativas que os seus concorrentes que não usam esta metodologia.

3.1.10 Componentes principais de dados

Os métodos dos componentes principais são usados para a análise de inter-relações entre variáveis e para a explicação dessas variáveis em termos das suas dimensões inerentes (componentes). Têm como principal objetivo condensar a informação contida nas variáveis originais num conjunto de variáveis estatísticas (componentes) com a perda mínima de informação [29]. A Figura 3.5 ilustra o tipo de análise a ser executada, dependendo do tipo de variáveis contidas no conjunto de dados.

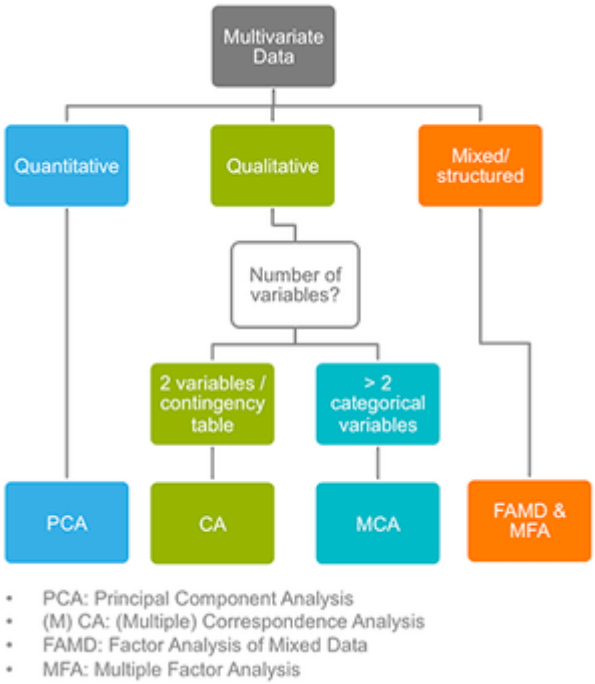


Figura 3.5: Utilização dos métodos de componentes principais.

3.1.10.1 *Principal Component analysis* (PCA)

Grandes conjuntos de dados são cada vez mais comuns e geralmente são difíceis de interpretar. Para interpretar esses conjuntos de dados, é necessário reduzir drasticamente a sua dimensionalidade de modo a torná-los interpretáveis, garantindo que a maioria das informações nos dados sejam preservadas [17].

Muitas técnicas foram desenvolvidas com este objetivo mas, o Principal Component Analysis (PCA) é uma das mais antigas e consequentemente mais usada. A ideia principal do PCA passa por reduzir a dimensionalidade de um conjunto de dados preservando, tanto quanto possível, a informação contida considerando as *features* originais. O PCA permite resumir e visualizar as informações num conjunto de dados contendo observações descritas por várias variáveis quantitativas inter-correlacionadas. Cada variável pode ser considerada como uma dimensão diferente.

O PCA é uma técnica estatística que consiste na substituição das antigas *features* por novas em que cada uma destas é uma combinação linear das anteriores, de forma a que as novas *features* não estejam correlacionadas entre si, podendo assim formar um espaço vetorial de eixos ortogonais. Esta técnica organiza as *features* por ordem decrescente de variância, de tal forma que, à primeira coluna da matriz output do PCA está associada a maior variância; à última coluna, a menor. Assim, existe uma concentração da variância total nas "primeiras" *features*, o que permite uma redução do seu número inicial.

A literatura mais antiga sobre PCA data de [30] e [31]. Desde então, o uso do PCA aumentou e um grande número de variantes foi desenvolvido em diversas disciplinas.

3.1.10.2 *Correspondence Analysis* (CA)

A Correspondence analysis (CA) é uma extensão do PCA adequada para explorar relações entre variáveis qualitativas (ou dados categóricos). Estas variáveis representam categorias. É usada para analisar frequências formadas por duas variáveis categóricas, uma tabela de dados conhecida como tabela de contingência.

3.1.10.3 *Multiple Correspondence Analysis* (MCA)

O Multiple Correspondence Analysis (MCA) permite analisar o padrão de relacionamento de várias variáveis dependentes categóricas e é uma extensão do CA para resumir e visualizar uma tabela de dados, que contenha mais do que duas variáveis categóricas. Também pode ser visto como uma generalização do PCA quando as variáveis a serem analisadas são categóricas em vez de contínuas [32].

3.1.10.4 *Multiple Factor Analysis* (MFA)

O Multiple Factor Analysis (MFA) [33] é um método de análise de dados multivariada para resumir e visualizar uma tabela de dados complexa, na qual as observações são descritas por vários conjuntos de variáveis (quantitativas e/ou qualitativas) estruturadas em grupos. Este método tem em consideração a contribuição de todos os grupos ativos de variáveis para definir a distância entre as observações. O número de variáveis em cada grupo pode diferir e a natureza das variáveis pode variar de um grupo para outro, mas as variáveis devem ter a mesma natureza num determinado grupo [32].

3.1.10.5 *Factor Analysis of Mixed Data* (FAMD)

Muitos conjuntos de dados atualmente contêm variáveis contínuas e categóricas. O FAMD é um método de componentes principais dedicado à descrição, resumo e visualização da matriz multidimensional com variáveis contínuas e categóricas [34]. Como qualquer método de componente principal, o seu objetivo é estudar as semelhanças entre observações, as relações entre variáveis (variáveis contínuas e categóricas) e vincular o estudo das observações com o das variáveis. Tal método, reduz a dimensionalidade dos dados e fornece um subespaço que melhor represente os dados. A redução da dimensionalidade é alcançada através do Singular Value Decomposition (SVD) de matrizes específicas, como referido em [35].

O princípio da FAMD é equilibrar a influência das variáveis contínuas e categóricas na análise. A lógica é ponderar as variáveis de forma a que cada variável de ambos os tipos contribuam, de maneira equivalente, para a construção das dimensões da variabilidade. O algoritmo FAMD pode ser visto aproximadamente como uma mistura entre análise de PCA e MCA. Por outras palavras, o algoritmo atua como PCA para variáveis contínuas e como MCA para variáveis categóricas. As variáveis quantitativas e qualitativas são normalizadas durante a análise para equilibrar a influência de cada conjunto de variáveis.

3.2 Artigos relacionados

Nesta Secção são referenciados e apresentados alguns artigos mais detalhadamente que, de forma geral, visam atingir objetivos semelhantes ou utilizam ferramentas e técnicas que pretendemos explorar no contexto desta dissertação.

Os conceitos principais que são propostos e utilizados nestes artigos foram previamente descritos com mais detalhe na Secção 3.1.

3.2.1 Extração de regras com base na utilização do SVM

No artigo [36] é proposta uma abordagem híbrida para extrair regras a partir da utilização do SVM para fins de gestão de relacionamento com o cliente CRM. A abordagem híbrida proposta consiste em três fases e é representada na Figura 3.6:

1. Seleção de características usando o SVM-RFE.
2. Extração de vetor de suporte usando SVM.
3. Geração de regras usando o NBTree.

O conjunto de dados analisados neste estudo é sobre *Churn Prediction* no cliente de cartão de crédito bancário (dados provenientes do Chile em 2004) que é altamente desequilibrado, com 93.24% de clientes fiéis e 6.76% de clientes *churned*.

Com a utilização de um conjunto de dados com características reduzidas, a abordagem proposta extrai regras de menor extensão, ou seja, regras simplificadas melhorando

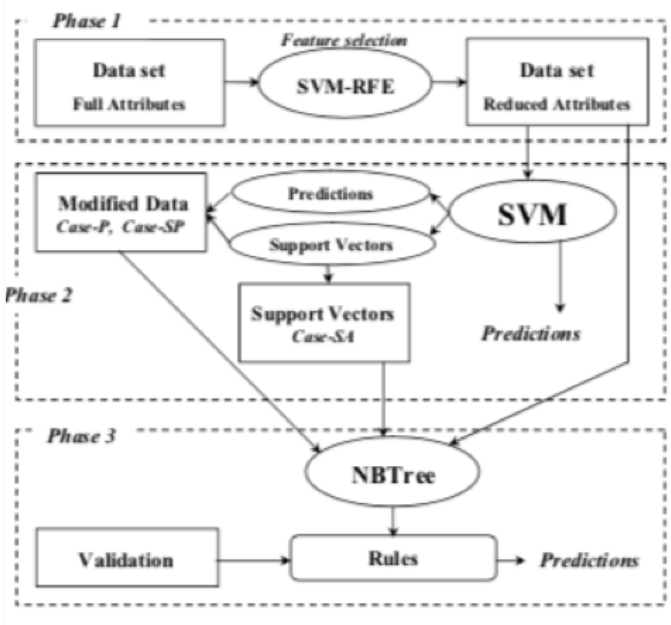


Figura 3.6: Extração de regras utilizando características selecionadas dos dados.

assim a compreensão dessas mesmas regras. As regras geradas atuam como um sistema especializado e preventivo de alerta antecipado, para a administração do banco.

Este artigo pode ser considerado como um estudo sobre a eficiência do SVM para lidar com dados desequilibrados tendo em conta a extração de regras, sendo também visto como um estudo analítico de CRM aplicado ao problema *churn*.

3.2.2 Aplicação de modelos de ML e obtenção de explicações

No artigo [37] é introduzida uma abordagem para a construção de um sistema inteligente que combina modelos de elevado desempenho, metodologia de explicação e consultores humanos, para ajudar a superar a resistência às mudanças. Este artigo centra-se em como abordar as necessidades dos utilizadores num ambiente de negócios complexo, explicar os modelos preditivos incompreensíveis de ML e as suas recomendações, com o objetivo de permitir que os utilizadores de negócios apliquem os modelos de ML com melhor desempenho à sua escolha e obtenham explicações interativas abrangentes, independentemente do modelo escolhido.

Os modelos *black box* de ML de alto desempenho obtêm um comportamento preditivo significativamente melhor do que modelos simples e interpretáveis. Esta é uma das razões para o baixo uso e aceitação de modelos de ML preditivos em áreas onde a transparência e a compreensão das decisões são necessárias. No artigo é explicado como os modelos *black box* de ML se podem tornar transparentes e ajudar os especialistas do domínio a avaliar e validar as suas convicções.

Para demonstrar o poder e a utilidade da metodologia de explicação para modelos de

ML, é apresentado um caso de previsão de vendas Business to Business (B2B) [2] no mundo real, demonstrando assim como resolver um problema de suporte à decisão.

É explicado o design do conjunto de dados com características descritivas que refletem o processo e o histórico de vendas. Foi feita a avaliação de vários modelos *black box* de ML e foi selecionado o modelo com melhor desempenho.

Um processo de negócios de alto nível alavancando o sistema de previsão inteligente proposto é apresentado na Figura 3.7, de acordo com os seguintes passos:

1. Recolha de casos de vendas B2B, com resultados conhecidos para apoiar numa tarefa de previsão para novas oportunidades de vendas.

2. Os dados recolhidos são processados por várias técnicas de ML, resultando num modelo de previsão validado estatisticamente.

3. A metodologia fornece explicações e permite uma avaliação cognitiva do modelo pelos utilizadores.

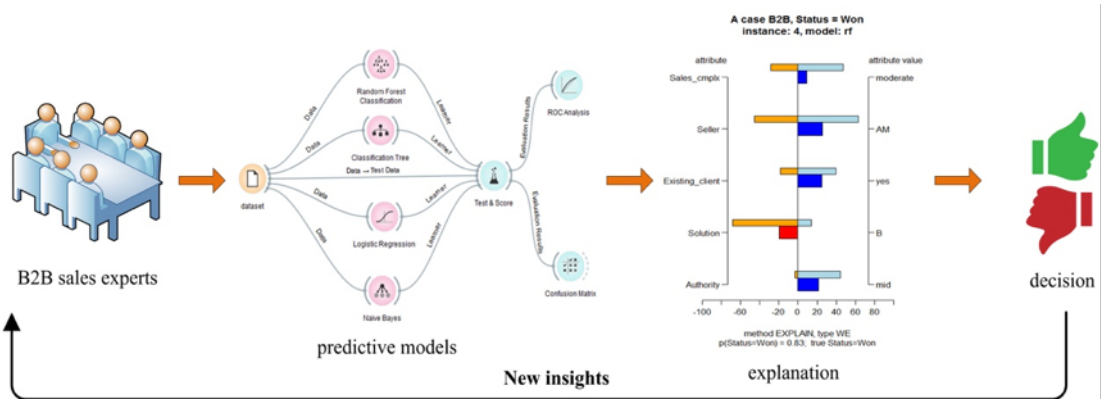


Figura 3.7: Visão geral de alto nível do sistema inteligente apresentado.

A título de exemplo, na Figura 3.8a, é visualizado o caso de quando os vendedores estão interessados em explicações de previsões para novos casos, para os quais o resultado é ainda desconhecido. A probabilidade prevista de uma venda bem-sucedida é de 0.72. A explicação revela uma fraca influência negativa do facto de a venda não ser discutida com um cliente existente. O atributo *Seller* com valor *AM* parece ter um impacto positivo marginal neste caso. Os atributos *Solution*, *Sales_complexity* e *Authority* contribuem para um resultado positivo. A probabilidade relativamente baixa prevista desencadeia uma discussão sobre as ações necessárias para aumentar a probabilidade de ganhar o contrato. Alguns atributos não podem ser mudados, no entanto, outros podem. A Figura 3.8b mostra a implicação da mudança no atributo *Seller*. A probabilidade de ganhar o negócio sobe para 0.92. Nesta Figura, as barras de explicação indicam os fortes impactos positivos de todos os valores dos atributos.

[2] Tipo de negócio feito de empresa para empresa, e não diretamente ao consumidor. Refere-se a duas empresas que fazem negócios como cliente e fornecedor.

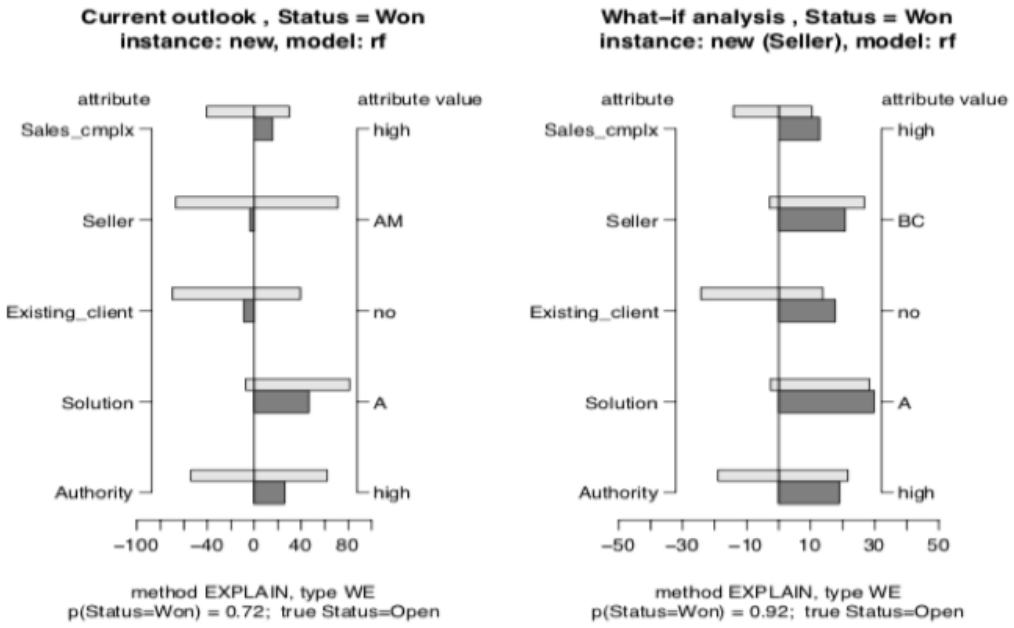
As explicações dos modelos de previsão e da análise *what-if* provaram ser um suporte efetivo para as previsões de vendas B2B. A metodologia apresentada aperfeiçoou a comunicação interna da equipa e melhorou a reflexão sobre o conhecimento. Uma vantagem significativa do método apresentado é a possibilidade de avaliar as ações do vendedor e delinear recomendações gerais na estratégia de vendas.

Em suma, as empresas/entidades que desejem aplicar a abordagem apresentada devem seguir os seguintes passos:

1. Identificar as características descritivas informativas e disponíveis, refletindo o contexto do processo de tomada de decisão.

2. Selecionar o modelo de previsão de ML com melhor desempenho.

3. Usar os métodos de explicação apresentados para expor a situação.



(a) (b)

Figura 3.8: Explicação para um novo caso (3.8a) e a sua variação (3.8b).

3.2.3 Desempenho preditivo e compreensibilidade

No artigo [38] é proposto um algoritmo híbrido, LLM, com o objetivo de melhor classificar os dados. Utiliza uma combinação de DT e LR e é desenvolvido para reduzir as desvantagens de ambos os modelos, mantendo as suas vantagens.

O LLM pode ser dividido em duas etapas: uma etapa de segmentação e uma etapa de previsão. Na primeira etapa, os segmentos dos clientes são identificados usando regras de decisão. Nesta fase é construída uma árvore de decisão para identificar os segmentos

de clientes homogéneos. Na segunda etapa, são aplicadas regressões logísticas para cada um dos segmentos.

Na Figura 3.9 é apresentada uma representação conceptual do LLM mostrando o fluxo de dados. Nesta representação, o conjunto inicial de clientes *S* foi dividido em três subconjuntos *S1*, *S2* e *S3* pela árvore de decisão. De seguida, uma regressão logística é ajustada para cada subconjunto, resultando nas probabilidades para cada instância nos subconjuntos.

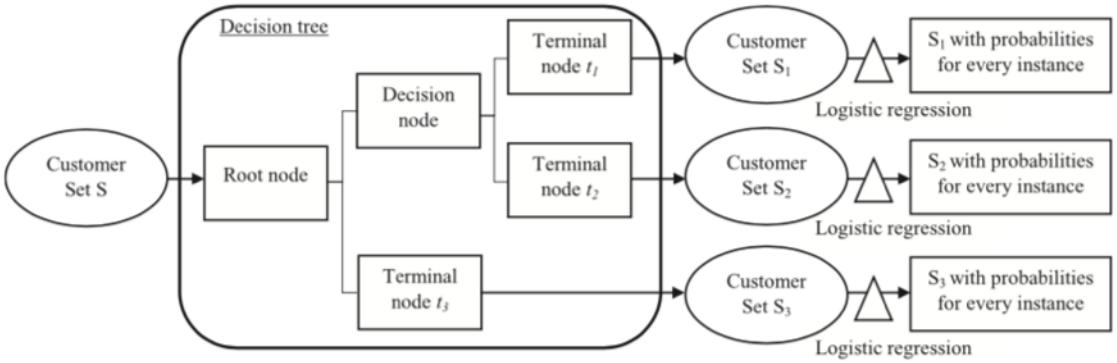


Figura 3.9: Apresentação conceptual do LLM.

Foram utilizados catorze conjuntos de dados de diferentes indústrias, nos quais o LLM é comparado com quatro algoritmos relacionados: DT, LR, RF e Logistic Model Tree (LMT) . É referido que o LLM é uma opção viável, tanto em termos de desempenho preditivo com em compreensibilidade. É um algoritmo que tem a capacidade de lidar com a heterogeneidade entre os clientes.

3.2.4 Métodos baseados na análise de componentes principais em estudos de Bioinformática

No artigo [39] é exposto o facto de que nos estudos de Bioinformática, são adotadas técnicas de perfil de alto rendimento, levando a medições de alta dimensão. Ou seja, tal como referido no artigo, na análise de dados de Bioinformática, o desafio criado surge da alta dimensionalidade das medições. Por exemplo, com um chip *Affymetrix* típico, pode criar um perfil de expressões de 40000 sondas. Num estudo de associação ampla do genoma (GWA), um milhão ou mais de polimorfismo de nucleotídeo único (SNPs) podem ser perfilados. O foco principal do artigo analisado está nas metodologias de análise, sendo que é fornecida uma análise aprofundada do PCA.

O PCA é uma abordagem clássica de redução de dimensão. Devido à sua simplicidade computacional, tem sido amplamente utilizado em estudos de Bioinformática e mostrou desempenho satisfatório. Apesar das vantagens do PCA, este também possui certas limitações. Existem algumas perguntas em aberto incluindo, por exemplo, a escolha do número adequado de componentes. O exame dos estudos publicados sugere que o desempenho

do PCA (e de facto de todas as abordagens de redução de dimensão e seleção de variáveis) depende dos dados. Devido ao seu custo computacional baixo, o PCA pode ser uma ferramenta preferida de redução de dimensão em muitos estudos.

O estudo exemplificativo dos artigos anteriores, fornece-me uma base de direção e conhecimento para o planeamento do protótipo de sistemas a construir, com vista a alcançar os objetivos gerais a que me proponho: obtenção de indicadores com base em dados de CRM e a explicação de regras associadas ao resultado.

Capítulo 4

Metodologias para a Solução

Este capítulo apresenta as metodologias aplicadas com vista a alcançar os objetivos propostos por esta dissertação e respetivas justificações para as opções tomadas.

4.1 Acesso e extração dos dados

A existência de elevadas quantidades de dados é essencial para o treino adequado de modelos de ML. No entanto, antes da disponibilização destes dados aos modelos de ML ou à maioria dos projetos de análise baseados em dados, é fundamental a estruturação dos mesmos e garantir que estes estejam limpos, consistentes e precisos.

Além da análise estatística que pode e deve ser efetuada aos dados, o conhecimento destes, a sua proveniência e contexto ao nível de negócio bem como a sua correlação com a realidade pode ser importante nas decisões e direções de modelação a tomar. Desta forma, no decorrer desta dissertação, estive em contacto regular com o director de *marketing* do ACP, que tem um conhecimento extenso dos dados que nos foram providenciados.

Assim, uma das primeiras etapas do trabalho e desenvolvimento desta dissertação foi a exploração dos dados. Esta fase permitiu-me alcançar um conhecimento mais aprofundado sobre estes dados, criando uma melhor perspetiva do seu conteúdo.

Do ponto de vista prático, o ACP disponibilizou os dados num banco de dados do *Structured Query Language (SQL) Server*. Como tal, foi utilizado o SQL Server Management Studio (SSMS) , que é um produto integrado da *Microsoft* para o desenvolvimento, configuração, análise e gestão de bancos de dados do *SQL Server*. De referir que não estamos a aceder a dados diretos do CRM. O ACP disponibilizou um *dataset* um pouco mais *flat* e com dados anonimizados por questões de RGPD, o que pode ter alguma influência no tipo de dados disponibilizados, sendo que este é um fator importante a ter em conta em projetos de ML.

Para criar um modelo de ML bem-sucedido é necessário treinar, testar e validar o modelo, antes de ser implementado em sistemas em produção. A preparação dos dados é usada para criar uma base limpa para os passos de criação do modelo de ML. A limpeza dos dados tem como objetivo a identificação de dados incorretos e/ou incompletos seguida da sua correção e remoção. Esta fase de preparação é fundamental e pode demorar um período significativo, no decorrer do processo de construção. Em sistemas de produção é essencial criar mecanismos (preferencialmente) automáticos de execução e controlo deste processo de angariação e limpeza de dados, uma vez que vão ser executados continuamente.

Uma vez que a implementação de algoritmos de ML pode ser complexa e requerer muito tempo, é essencial ter um ambiente bem estruturado e testado de forma a permitir que sejam apresentadas as melhores soluções de codificação. Foi utilizada a linguagem *Python* que possui um extenso conjunto de bibliotecas [1] para ML, que permitem a redução do tempo de desenvolvimento. Legibilidade, versatilidade e facilidade são algumas das principais razões para usar *Python*.

No caso específico desta dissertação, as *queries* que irão alimentar os modelos de ML, foram executadas através da conexão do *Python* ao *SQL Server*, através da biblioteca *pyodbc*. Depois de estabelecida a conexão entre o *Python* e o *SQL Server*, torna-se assim possível usar o SQL diretamente no *Python*. Foi utilizada a aplicação *web Jupyter Notebook* que permite editar e executar documentos (*notebooks*) que contêm código (por exemplo, *python*), equações, figuras e texto, entre outros.

4.2 Análise do *dataset*

O *dataset* utilizado no caso de estudo subjacente a esta dissertação foi disponibilizado pelo ACP e tem cerca de 600 000 registos que, em sumário, apresenta informação do conjunto de entidades contacto. Esta entidade representa um sócio com diversos estados possíveis associados, algumas informações pessoais (por exemplo idade, distrito, antiguidade no ACP, a razão pela qual entrou para o ACP, etc.), informação sobre o tipo de serviços que estão associados a um sócio (por exemplo, se tem assistência em viagem, se tem seguro, se tem cartão de saúde, etc.), bem como o número de interações que um sócio teve com o ACP. Os dados disponibilizados contêm um total de 33 *features*, sendo que 18 são *features* numéricas e as restantes são *features* categóricas, tal como se pode ver na Tabela A.1.

Numa associação como o ACP, um dos fatores cruciais é manter os sócios ativos durante o máximo período de tempo. Assim, a ANTIGUIDADE dos sócios, é uma *feature* que considerei ser importante analisar pelo facto de representar o tempo de fidelização de um sócio, sendo que o valor 0 significa que um "contacto" é sócio há menos de 1 ano. Analisando a distribuição da Antiguidade dos sócios com um intervalo de 10 anos, verificou-se

[1] Código pré-escrito usado para resolver tarefas de programação.

que grande parte dos sócios apresenta uma Antiguidade compreendida entre os 1 e 10 anos (Figura 4.1).

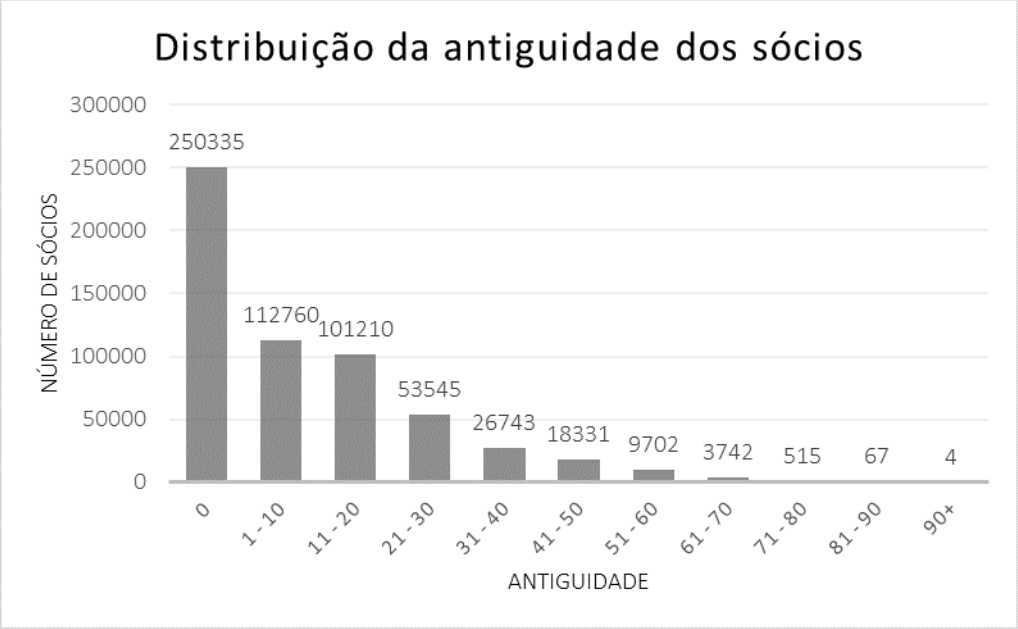


Figura 4.1: Distribuição da antiguidade dos sócios.

Foi tomada a decisão de fazer subconjuntos de dados para a criação de modelos de ML, pois em termos de negócio era importante para o ACP agrupar clientes com características semelhantes. Adicionalmente, quando iniciei o treino dos modelos, verifiquei que do ponto de vista prático, com o conjunto de dados existente, não tinha à disposição o poder computacional necessário para, em tempo útil, treinar, testar e validar os modelos de ML.

Assim, decidi fazer a seguinte divisão, em função da ANTIGUIDADE, que é uma *feature* presente no *dataset* disponibilizado:

- Antiguidade 0 - 10 anos.
- Antiguidade 11 - 40 anos.
- Antiguidade 40 - 94 anos.

Ainda assim, o número de linhas de dados para cada subconjunto de dados, apresentado na Tabela 4.1, mostrou-se elevado para *input* e processamento aos algoritmos de ML, tendo em conta o poder computacional que tinha à disposição (Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz 8.00 GB RAM). Como tal, optei por utilizar conjuntos de 50 000 registos aleatórios de cada subconjunto dos "escalões" de Antiguidade, sendo que para o subconjunto Antiguidade 41 - 94 anos foram utilizados todos os dados, pois apresenta um número inferior a 50 000.

Os sócios apresentam um estado de acordo com a sua situação. É de referir que mais de metade dos sócios presentes neste conjunto de dados já não se encontram a utilizar

Tabela 4.1: Sub-Conjuntos de dados.

Sub-Conjunto	Número
Antiguidade 0 - 10 anos	277 078
Antiguidade 11 - 40 anos	267 515
Antiguidade 41 - 94 anos	32 361

os serviços. Ou seja, apresentam um estado que já não é ativo. Todas as informações associadas aos estados são apresentadas na Tabela 4.2.

Tabela 4.2: Informações associadas aos estados possíveis de um sócio.

Estado	Tipificação	Descrição	Número
ACT	Ativo	O sócio encontra-se ativo.	256 401
DEMIT_AUTO	Demitido automaticamente	Após a suspensão, se não há regularização das quotas ao fim de um determinado período de tempo, o ACP demite o sócio automaticamente.	163 589
DEMIT	Demitido	O sócio demitiu-se.	156 074
SUSP	Suspenso	O ACP suspende o sócio temporariamente até regularização das quotas.	890

De realçar que a fonte de informação utilizada refere-se a sócios, ativos ou não, mas que o mesmo princípio de análise poderá ser utilizado para candidatos a sócios (*prospects*). Estes poderão ter tido interações com o ACP, sejam telefonicamente ou via sítio *Web*, e igualmente ter feito um registo por sua iniciativa com o seu perfil e produtos de interesse. Assim, estas interações permitiriam ao ACP aplicar os modelos para determinar a espectativa de antiguidade e as alterações de estado, modificando parâmetros de entrada (campanhas de redução de quotas, produtos de oferta, etc.), com vista a cativar estes *prospects*, que uma vez integrados passariam a ser novos sócios.

4.3 Tradução das *features* categóricas em *features* numéricas

Em projetos de ML, uma parte importante é o tratamento das *features*. É muito comum os *datasets* apresentarem *features* categóricas. Em particular, grande parte dos algoritmos de ML exigem que o seu *input* seja numérico e, portanto, torna-se assim necessário fazer alguma manipulação de dados - codificar as *features* categóricas em *features* numéricas.

O *dataset* utilizado nesta dissertação, apresenta tanto *features* categóricas como *features* numéricas. Como tal, foi necessário traduzir as *features* categóricas em *features* numéricas,

levando assim ao consequente aumento significativo do número de *features* (este aumento ocorre quando o número de *features* apresentado reflete toda a variância associada aos dados). Após a tradução, o *dataset* apresenta um número superior de *features* relativamente ao que apresentava inicialmente. Assim, é necessária uma redução deste número, ou seja, uma redução da dimensionalidade dos dados, devido à estratégia possível ser utilizada em termos de capacidade computacional.

A tradução das *features* categóricas em *features* numéricas foi feita utilizando o método de componente principal FAMD, referido na Secção 3.1.10.5, que pondera as novas *features* de forma a que a influência das *features* categóricas e numéricas seja equivalente (Figura 4.2). Na construção prática do projeto, optei por utilizar a biblioteca *Light_FAMD* que é uma uma biblioteca para análise fatorial de processamento de dados mistos. Esta biblioteca para além do método FAMD, inclui uma variedade de outros métodos, tais como o PCA, referido na Secção 3.1.10.1, e MCA, referido na Secção 3.1.10.3 e fornece uma implementação, segundo a documentação, eficiente e leve para cada algoritmo. Tal como mencionado, a tradução de *features* categóricas, faz com que o número de *features* (ou número de componentes) aumente. Assim, ao proceder à sua redução, é necessário garantir que a maioria das informações dos dados seja preservada.

	SEXO	PAIS	ANTIGUIDADE	TIPO_SOCIO	IDADE	DISTRITO	ESTADO	CANAL	RAZAO	CAMPANHA	...
0	M	PORTUGAL ...	8.0	B	33.0	PORTO ...	DEMIT_AUTO	SECÇÃO REGIONAL NORTE ...	SERVIÇOS DOCUMENTAÇÃO ...	MGM GERAL ...	...
1	M	PORTUGAL ...	6.0	O	52.0	SETÚBAL ...	DEMIT_AUTO	DELEGAÇÃO DE SETÚBAL ...	ACÇÃO PROMOCIONAL ...	ACP VIAVERDE S/ EXTRATO (OURO) ...	...
2	M	PORTUGAL ...	7.0	D	8.0	LISBOA ...	ACT	DELEGAÇÃO DO ESTORIL ...	JUNIOR ...	JUNIOR - ISENTO ...	...
3	F	PORTUGAL ...	1.0	C	61.0	LISBOA ...	ACT	CONTACT CENTER ...	SERVIÇOS DOCUMENTAÇÃO ...	CONTACT CENTER ...	...
4	M	PORTUGAL ...	8.0	O	40.0	LISBOA ...	DEMIT	CONTACT CENTER ...	OUTRAS ASSISTÊNCIAS - MÉDICA, ...	DB4 TMKT 2 ...	...

(a) Dados originais.

	0	1	2	3	4	5	6	7	8
0	497.653849	-334.959865	59.472424	309.897521	231.112952	53.083241	-53.157132	-55.772232	-243.641380
1	455.136123	-260.454501	63.073794	-192.104475	145.051259	-197.841827	21.839947	-108.650230	229.257219
2	614.104062	168.243674	445.116168	-26.658707	-236.344385	9.337438	-105.711736	10.488467	-19.866971
3	580.095791	211.219351	-431.952084	3.317715	-57.024869	232.294098	-184.847276	32.585943	-109.303756
4	603.156362	-197.797798	-77.520254	-389.463580	-123.039897	23.910132	-77.723491	-144.526197	-19.346811

(b) Dados após a aplicação do FAMD.

Figura 4.2: Comparação do aspeto visual dos dados antes e depois da aplicação do FAMD.

Levanta-se então a questão de: como escolher o número de componentes? A escolha deste número não deve ser feita manualmente. Em vez disso, devemos ter em conta a variância explicada por cada componente gerado. Tipicamente, é expectável que o número de componentes escolhido contenha entre 95% e 99% da variância total. Desta forma, espera-se poupar em número de componentes e consequentemente em complexidade

computacional.

Tendo em conta os subconjuntos definidos, para o subconjunto de dados Antiguidade 0 - 10 anos, para obter 95% da variância explicada, são necessários 37 componentes principais (Figura 4.3). Uma vez que este é um número de componentes que pode ser demasiado elevado para *input* de alguns algoritmos de ML (ex: *Model Based Cluster Analisys*), foi necessário definir, por prudência, outra abordagem. Assim, foi verificado para cada componente principal qual a sua percentagem de variância explicada. Todos os componentes que apresentaram uma percentagem de variância individual inferior a um certo limiar não foram considerados. Desta forma, escolhendo 0.01 para o limiar, para o subconjunto de dados Antiguidade 0 - 10 anos, o número de componentes foi reduzido para 17, que corresponde a 84% da variância explicada. Este valor de variância inferior ao expectável pode revelar perdas de informação dos dados.

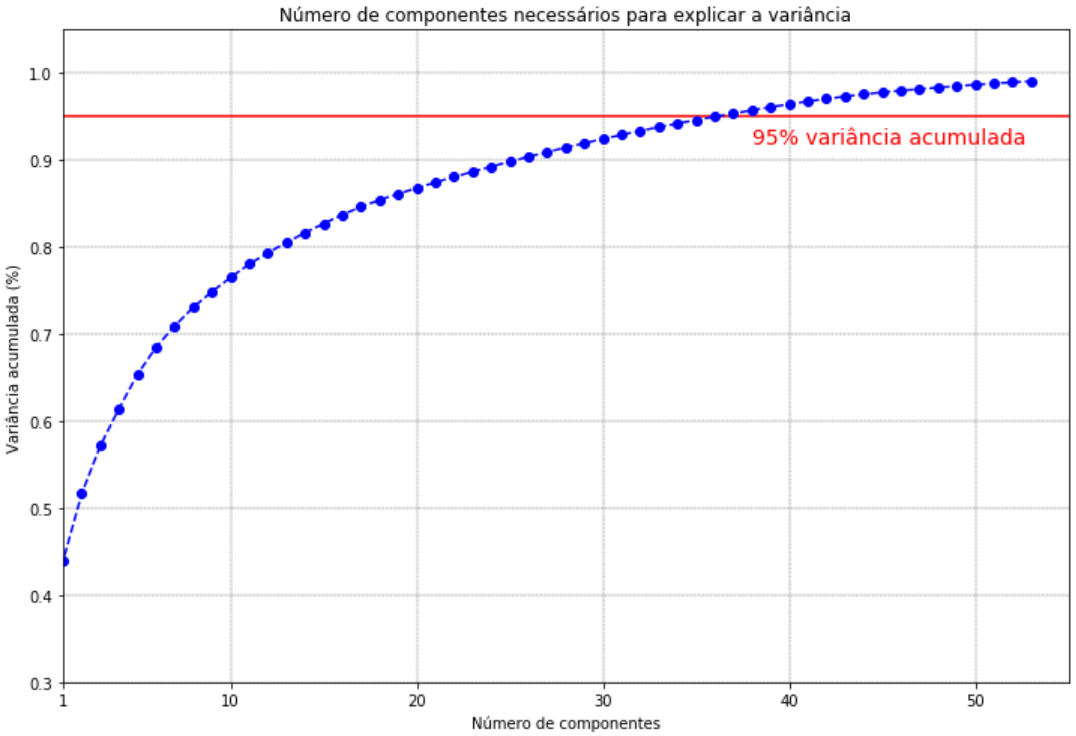


Figura 4.3: Número de componentes principais para explicar a variância no subconjunto de dados Antiguidade 0 - 10 anos.

Em relação ao subconjunto de dados Antiguidade 11 - 40 anos, são necessários 16 componentes principais para explicar 95% da variância (Figura 4.4).

Para o subconjunto de dados Antiguidade 41 - 94 anos, são necessários apenas 3 componentes principais para explicar 95% da variância (Figura 4.5).

Desta forma, nestes dois últimos casos não foi necessário considerar os componentes que apresentam uma percentagem individual de variância inferior ao limiar definido.

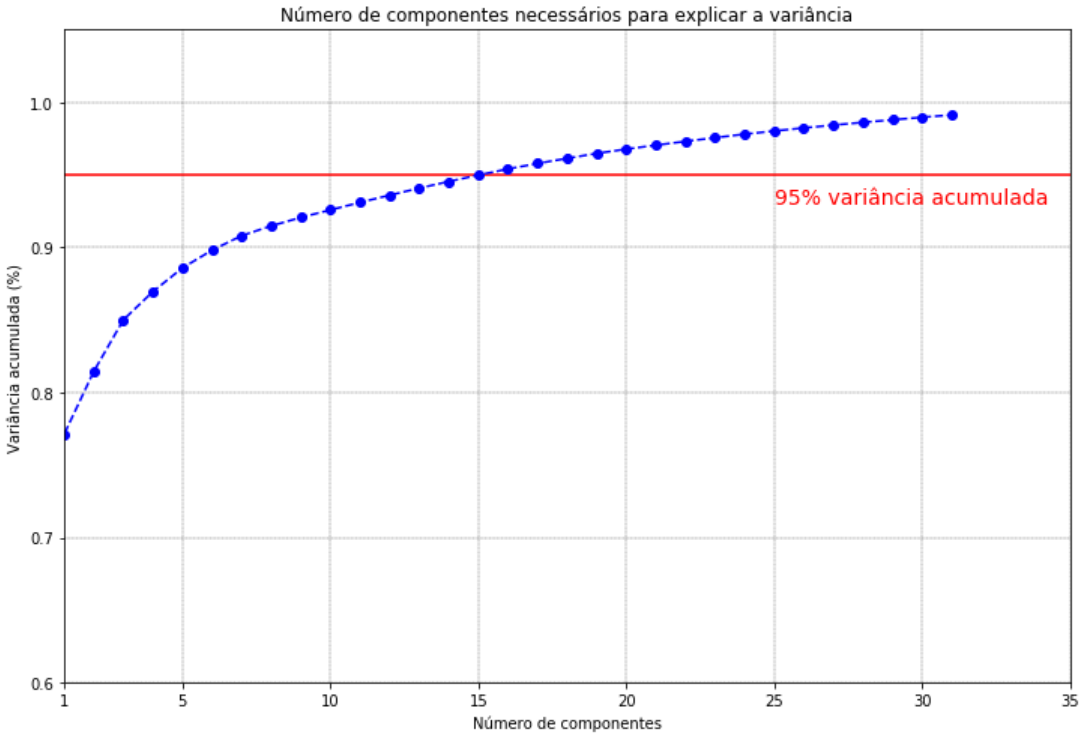


Figura 4.4: Número de componentes principais para explicar a variância no subconjunto de dados Antiguidade 11 - 40 anos.

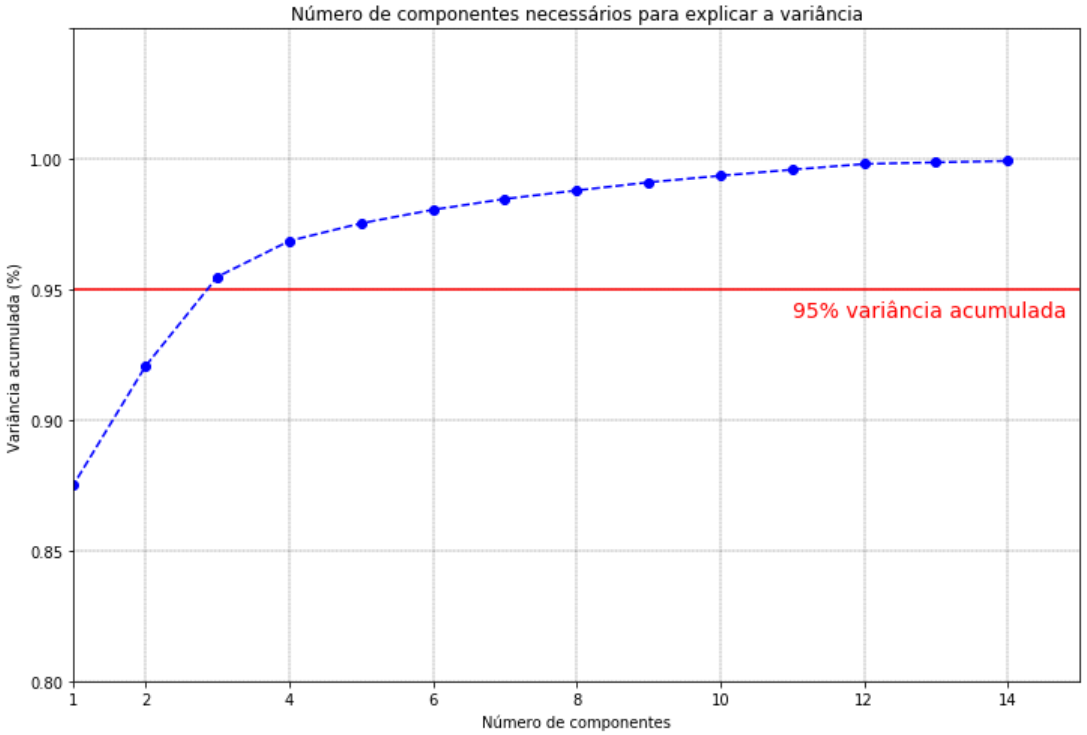


Figura 4.5: Número de componentes principais para explicar a variância no subconjunto de dados Antiguidade 41 - 94 anos.

4.4 Subconjuntos de treino e teste

Além das técnicas para avaliar quanto um modelo está a aprender com os dados, por exemplo, o seu grau de acerto, como veremos na Secção 4.6, uma prática bem estabelecida é dividir o *dataset* disponível em dois subconjuntos, um de treino e outro de teste. Desta forma será mais provável garantir que o modelo tem um melhor desempenho quando são apresentados dados previamente desconhecidos. O *dataset* é dividido aleatoriamente de modo a ter uma representação de todos os pontos de dados em ambos os subconjuntos.

O subconjunto de treino contém o *output* correto para que o modelo aprenda com esses dados de forma a generalizar para outros dados posteriormente, ou seja, é permitido ao modelo que "conheça" as respostas corretas.

O subconjunto de teste (conjunto de dados não conhecido pelo modelo onde não há acesso às respostas, apenas às *features*) tem como objetivo verificar se o modelo é consistente e testar a sua precisão.

Foi analisada também a abordagem que consiste em reservar parte dos dados de forma a criar também um subconjunto de validação: o treino/ajuste do modelo seria feito com o subconjunto de treino, de seguida, o subconjunto de validação seria usado para fornecer uma avaliação imparcial de um ajuste do modelo no subconjunto de dados de treino ao serem ajustados os hiperparâmetros e, por fim, o subconjunto de teste seria usado para fornecer uma avaliação imparcial de um ajuste final do modelo no subconjunto de dados de treino. Ainda que o subconjunto de validação seja predominantemente usado para descrever a avaliação do modelo ao ajustar hiperparâmetros, ao dividir os dados em três conjuntos, o número de amostras é reduzido drasticamente, ou seja, perdemos dados que poderiam ser usados para fazer aprender o modelo.

Para evitar esta perda de dados de treino, foi utilizado um procedimento chamado Cross-Validation (CV) . Assim, como referido anteriormente, o subconjunto de teste é necessário para a avaliação do ajuste final, mas o conjunto de validação não é necessário ao utilizar o CV. A abordagem utilizada foi então fazer a divisão em dois subconjuntos - treino e teste. Para tal, através da biblioteca *Sklearn*, sub-biblioteca *model_selection* foi importado o método *train_test_split*. O CV divide o subconjunto de treino em *k-fold* (conjuntos menores), Figura 4.6. O modelo é treinado usando *k-1 folds* como dados de treino; o modelo resultante é validado na parte restante dos dados (é usado como um conjunto de testes para calcular uma medida de desempenho). A medida de desempenho relatada pelo *k-fold* CV é então a média dos valores calculados no *loop*. Apesar do conjunto de dados a utilizar ter sido reduzido, esta continua a ser uma abordagem computacionalmente cara.

A divisão dos subconjuntos de treino e teste foi feita depois do *dataset* apenas conter variáveis numéricas (Secção 4.3), que é quando o conjunto de dados apresenta o seu formato final que irá ser dado a *input* dos algoritmos de ML.

Do *dataset* original foram retidos 80% dos dados para subconjunto de treino e os restantes 20% para o subconjunto de teste.

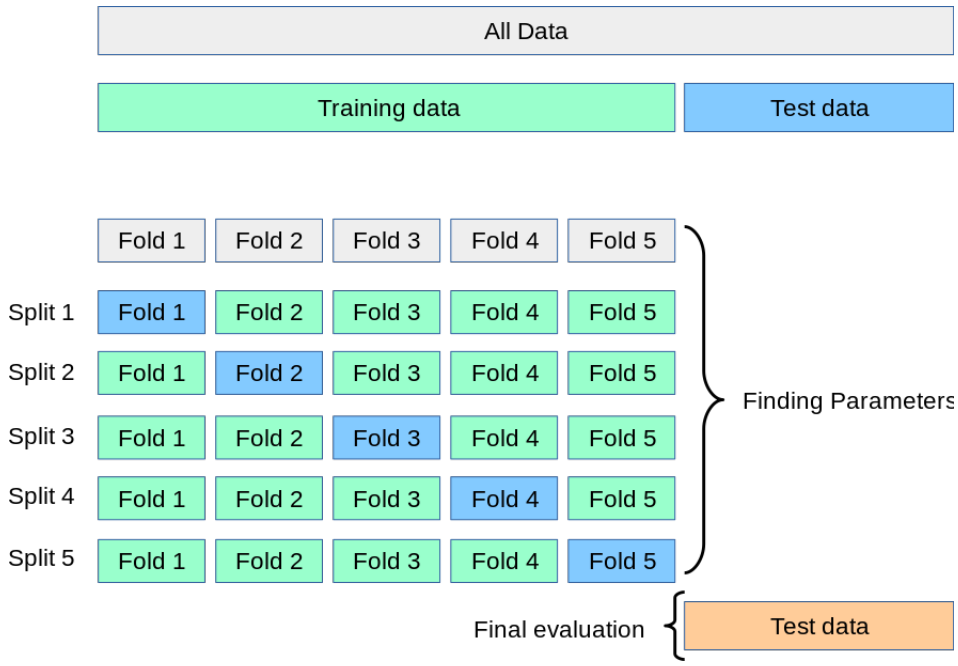


Figura 4.6: Cross-Validation no subconjunto de treino.

4.5 Algoritmos considerados

Um modelo de ML necessita que sejam definidos valores para os seus hiperparâmetros (características de um modelo que são externas ao modelo e cujo valor não pode ser estimado a partir dos dados; o valor dos hiperparâmetros devem ser definidos antes de treinar o modelo; por exemplo: no caso do algoritmo RF, um dos híperparâmetros será o número de árvores). Normalmente, os valores para os hiperparâmetros são configurados quase aleatoriamente e, de seguida, são analisados quais os valores que resultam num melhor desempenho. No entanto, selecionar aleatoriamente os hiperparâmetros para o algoritmo, para além de ser uma tarefa bastante exaustiva uma vez que é necessário comparar o desempenho dos algoritmos com os diferentes valores dos hiperparâmetros, consome também recursos, tempo de computação e de análise. Existe ainda o risco ao avaliar diferentes combinações para os hiperparâmetros dos modelos de *overfitting*, porque os hiperparâmetros podem ser ajustados até que o modelo esteja otimizado o máximo possível. Assim, em vez dos valores dos hiperparâmetros serem selecionados aleatoriamente, a abordagem adotada foi utilizar um algoritmo que encontra automaticamente os melhores valores para um modelo específico. Para responder a esta necessidade, foi utilizado o algoritmo *Grid Search*, que é usado para encontrar os hiperparâmetros ideais de um modelo que resultam nas previsões mais precisas.

Para utilizar o algoritmo *Grid Search*, foi importada a classe *GridSearchCV* da biblioteca *sklearn.model_selection*. A primeira etapa para a utilização do algoritmo *Grid Search* é criar um dicionário com todos os hiperparâmetros e os respetivos conjuntos de valores que pretendemos testar para obter o melhor desempenho. O algoritmo *Grid Search* testa

todas as combinações possíveis dos valores dos hiperparâmetros definidos no dicionário e retorna a combinação com a maior precisão. O algoritmo *Grid Search* pode tornar-se lento, devido ao número potencialmente grande de combinações a serem testadas. Aliada à utilização da classe *GridSearchCV* está associada a utilização do procedimento CV, referido na Secção 4.4. A sua utilização aumenta ainda mais o tempo de execução e a complexidade.

No contexto das linhas orientadoras que nortearam os desenvolvimentos a levar a cabo nesta dissertação, recordo que a aplicação de algoritmos de ML utilizados, têm como objetivo prever o estado de um sócio bem como a sua antiguidade, conforme foi referido na Secção 1.4. Para tal, foi utilizada a biblioteca *Scikit-Learn* do *Pyton* para importar os algoritmos SVM, RF e MLP.

Conforme especificado na Secção 1.4, trabalho sobre duas *features* em particular: ANTIGUIDADE e ESTADO. De seguida, apresento para cada uma das *features* os algoritmos utilizados. De acordo com o estudo preliminar, estes algoritmos são os que mais se adequam a este trabalho em concreto.

Para a previsão da ANTIGUIDADE foram utilizados os seguintes algoritmos de regressão e respetivas classes para comparação:

- Regressor RF.

  Classe *RandomForestRegressor* da biblioteca *sklearn.ensemble*.

- Regressor MLP.

  Classe *MLPRegressor* da biblioteca *sklearn.neural_network*.

- SVM (*Linear* Support Vector Regression (SVR)).

  Classe *svm* da biblioteca *sklearn*.

Para a previsão do ESTADO foram utilizados os seguintes algoritmos de classificação e respetivas classes para comparação:

- Classificador RF.

  Classe *RandomForestClassifier* da biblioteca *sklearn.ensemble*.

- Classificador MLP.

  Classe *MLPClassifier* da biblioteca *sklearn.neural_network*.

- SVM (Support Vector Classification (SVC)).

  Classe *svm* da biblioteca *sklearn*.

Em resumo e do ponto de vista operacional, pretendo que seja possível prever a antiguidade de um sócio face ao seu perfil. Em termos práticos, esta previsão pode ser útil para determinar a antiguidade de um sócio alterando as características do seu perfil,

sendo assim possível prever durante quanto tempo um sócio irá permanecer no ACP. Desta forma, surge uma oportunidade para o ACP de sugerir novos serviços a um sócio de modo a prolongar a fidelidade com o clube. Em suma, o ACP pode, por exemplo, avaliar mensalmente todos os seus sócios de forma a prever quais, no futuro, irão ter menor antiguidade promovendo assim campanhas de retenção para os mesmos. Relativamente ao estado, podem ser feitos testes com possíveis diferentes ações (por exemplo, oferta de assistências, oferta de serviços, etc.) de forma a inferir quais é que sofrem uma alteração no seu estado, e tal como para o fator antiguidade, o ACP pode desenvolver ações para contrariar estados não ativos.

4.6 Métricas de Avaliação

Após o treino dos modelos, medir o desempenho dos resultados destes é uma tarefa necessária no uso de ML, pois é indicador da qualidade das previsões/resultados de um modelo treinado.

4.6.1 Métricas de Avaliação para algoritmos de Classificação

4.6.1.1 Confusion Matrix

Na área de ML e mais especificamente no problema de classificação estatística, a *confusion matrix* (Figura 4.7) é uma tabela com um *layout* específico que é frequentemente usada para a medida de desempenho de um algoritmo de classificação, tipicamente um algoritmo de *supervised learning*. Cada linha da matriz representa as instâncias numa classe prevista, enquanto cada coluna representa as instâncias numa classe real.

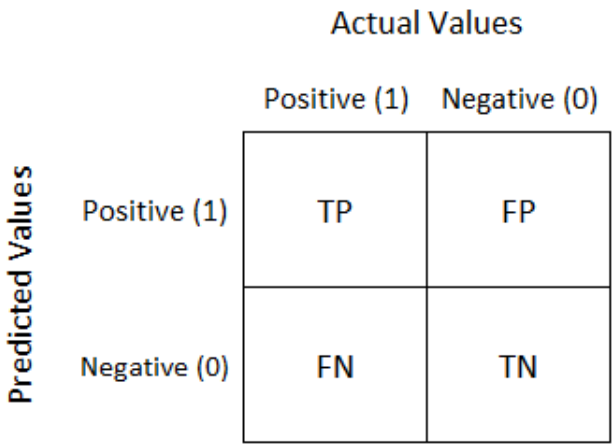


Figura 4.7: *Confusion Matrix*.

Na análise preditiva, uma *confusion matrix* é uma tabela que informa o número de True Positive (TP) , True Negative (TN) , False Positive (FP) e False Negative (FN) .

TP: é um resultado em que o modelo prediz corretamente a classe positiva.

TN: é um resultado em que o modelo prediz corretamente a classe negativa.

FP: é um resultado em que o modelo prediz incorretamente a classe positiva – erro do tipo 1.

FN: é um resultado e que o modelo prediz incorretamente a classe negativa – erro do tipo 2.

Os dados desta matriz ajudam no cálculo das métricas que são descritas nas próximas Secções.

Isto permite uma análise mais detalhada do que a mera proporção de classificações corretas (*Accuracy*, Secção 4.6.1.2). Do ponto de vista de execução do protótipo, de forma a apresentar esta tabela, foi necessário importar a classe *confusion_matrix* da biblioteca *sklearn.metrics*.

4.6.1.2 Accuracy

A *Accuracy* é a métrica mais comum para problemas de classificação, no entanto, não é uma métrica confiável para o desempenho real de um algoritmo, uma vez que pode produzir resultados enganadores se o conjunto de dados for desequilibrado (ou seja, quando o número de observações nas diferentes classes varia). Assim, é uma métrica realmente adequada apenas quando, idealmente, há um número igual de observações em cada classe e quando todas as previsões e erros de previsão são igualmente importantes.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

4.6.1.3 Classification Report

Um *Classification Report* é usado para medir a qualidade das previsões de um algoritmo. O relatório pode incluir as seguintes medidas:

- ***Precision*** - Indica, de entre todas as classes positivas previstas, quantas são realmente positivas. É intuitivamente a capacidade do classificador de não rotular como positiva uma amostra negativa.

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{TotalPredictedPositive} \tag{4.2}$$

- ***Recall*** – Indica, de entre todas as classes positivas, quantas foram previstas corretamente. Este indicador deve ser o mais alto possível. É intuitivamente a capacidade do classificador de encontrar todas as amostras positivas.

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{TotalActualPositive} \tag{4.3}$$

- ***F1 -score*** – É difícil comparar dois modelos com baixa precisão e alto *recall* ou vice-versa. Então, de forma a torna-los comparáveis, é usado o *F1-score*. Pode ser interpretado como uma média harmónica ponderada da precisão e *recall*.

$$F1 - score = 2 \times \frac{Precision * Recall}{Precision + Recall} \tag{4.4}$$

- ***Support*** – É o número de ocorrências de cada classe.
- ***Macro Average*** – Calcula a métrica independentemente para cada classe e, em seguida, obtém a média (tratando portanto todas as classes igualmente).
- ***Micro Average*** – Agrega as contribuições de todas as classes para calcular a métrica. É mostrada apenas quando temos mais do que duas classes porque, caso contrário, corresponde à precisão. Esta é uma métrica importante quando há desequilíbrios nas classes (ou seja, quando temos mais amostras de uma classe do que de outras).
- ***Weighted Average*** – Calcula a métrica com base no *support*.

De forma a apresentar o relatório, foi necessário importar a classe *classification_report* da biblioteca *sklearn.metrics*.

4.6.1.4 Curva Area Under The Curve (AUC)-Receiver Operating Characteristics (ROC)

A AUC-ROC é uma métrica que é representada por um gráfico de avaliação, para verificar ou visualizar o desempenho de qualquer modelo de classificação, independentemente do limiar de classificação.

A curva ROC é traçada em função do True Positive Rate (TPR) (eixo yy) e do False Positive Rate (FPR) (eixo xx), tal como está representado na Figura 4.8 onde TPR é o sinónimo de *Recall* e FPR é definido por:

$$FPR = \frac{FP}{FP + TN} \tag{4.5}$$

A AUC mede toda a área bidimensional abaixo da curva ROC, representa o grau ou medida de separabilidade. Diz o quanto o modelo é capaz de distinguir entre classes. Quanto maior a AUC, melhor é a capacidade do modelo de fazer previsões corretas. Um modelo excelente tem AUC perto de 1, o que significa que o modelo é bastante capaz de separar classes. Um modelo com mau desempenho tem AUC próximo de 0, o que significa que tem uma má medida de separabilidade. Por exemplo, o modelo prevê uma classe negativa como uma classe positiva e vice-versa. Quando o AUC é 0,5 significa que o modelo não tem capacidade de separação de classes.

Para gerar o gráfico, foi necessário importar as classes *roc_curve* e *auc* da biblioteca *sklearn.metrics*.

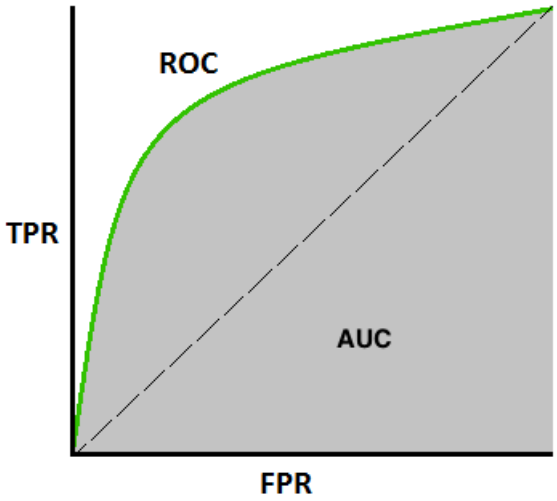


Figura 4.8: Curva AUC-ROC.

4.6.2 Métricas de Avaliação para algoritmos de Regressão

4.6.2.1 Mean Absolute Error (MAE)

A MAE é uma métrica de erro de regressão simples e intuitiva, pois observa a diferença absoluta entre a observação real e as previsões do modelo. Matematicamente, é calculado através de:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - \hat{y}_i\right| \tag{4.6}$$

É calculado o valor residual para cada ponto de dados, utilizando apenas o valor absoluto de cada um, para que os resíduos positivos e negativos não sejam cancelados. De seguida, calculamos a média de todos esses resíduos. Efetivamente, o MAE descreve a magnitude típica dos resíduos. Como usamos o valor absoluto do residual, o MAE não indica desempenho inferiorizado ou superiorizado do modelo. Cada resíduo contribui proporcionalmente à quantidade total de erros, o que significa que erros maiores contribuirão linearmente para o erro geral. Um valor pequeno de MAE sugere que o modelo é ótimo na previsão, enquanto um valor grande de MAE sugere que o modelo pode ter problemas em determinadas áreas. Um MAE com valor 0 significa que o modelo é perfeito. Foi utilizada a classe *mean_absolute_error* da biblioteca *sklearn.metrics*.

4.6.2.2 Mean Squared Error (MSE)

Talvez o MSE seja a métrica mais simples e comum para avaliação de algoritmos de regressão. É definida por:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{4.7}$$

Onde y_i é o output real e $\hat{y}_i$ é a previsão do modelo. O MSE mede basicamente o erro quadrático médio das previsões. Para cada ponto, calcula o quadrado da diferença entre as previsões e o *target* e, de seguida, calcula a média desses valores. Quanto maior esse valor, pior o modelo. O MSE nunca apresenta um valor negativo, pois os erros individuais de previsão são corrigidos antes de ser efetuada a soma. Num modelo perfeito o valor seria 0. É uma métrica útil quando temos valores inesperados com os quais nos devemos preocupar, já que o efeito quadrático sobrevaloriza os maiores desvios em relação à média. No entanto, se for feita uma única previsão desfavorável, o quadrado tornará o erro ainda pior e poderá distorcer a métrica de modo a superestimar a má avaliação do modelo. Este pode ser um comportamento particularmente problemático quando temos um conjunto de dados com ruído [2], até um modelo "perfeito" pode ter um MSE alto nesta situação, tornando assim difícil analisar o quão bem o modelo está a executar. Por outro lado, se todos os erros forem pequenos, ou seja, menores que 1, ocorre o efeito oposto: é subestimada a má avaliação do modelo. Enquanto cada resíduo no MAE contribui proporcionalmente para o erro total, o erro cresce quadraticamente no MSE. Foi utilizada a classe *mean_squared_error* da biblioteca *sklearn.metrics*.

4.6.2.3 Root Mean Squared Error (RMSE)

O RMSE é a raiz quadrada do MSE, ou seja, é a raiz quadrada da média das diferenças quadráticas entre previsão e observação real. A raiz quadrada é introduzida de forma a fazer com que a escala dos erros seja igual à escala dos *targets*.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} = \sqrt{MSE} \tag{4.8}$$

Assim, é importante entender em que sentido o RMSE é semelhante ao MSE e quais são as diferenças que apresenta. São semelhantes em termos dos seus minimizadores uma vez que todo o minimizador do MSE é também um minimizador para o RMSE e vice-versa, pois a raiz quadrada é uma função não decrescente. Por exemplo, se tivermos dois conjuntos de previsões, A e B, e o MSE de A for maior que o MSE de B, podemos ter certeza de que RMSE de A é maior RMSE de B. Funciona da mesma forma na direção oposta.

$$MSE_{(a)} > MSE_{(b)} <=> RMSE_{(a)} > RMSE_{(b)} \tag{4.9}$$

A raiz quadrada dos erros quadráticos médios tem algumas implicações interessantes para o RMSE. Como os erros são elevados ao quadrado antes da média, o RMSE atribui

[2] Dados que por algum motivo não são totalmente confiáveis.

um peso relativamente alto a erros grandes. Isso significa que o RMSE deve ser mais útil quando erros grandes são particularmente indesejáveis.

4.6.2.4 Comparações entre métricas

O MAE é mais robusto (menos sensível a valores discrepantes) que o MSE, no entanto isso não significa que seja sempre melhor usar o MAE.

Sobre a métrica MAE é importante referir que não penaliza os grandes erros de forma tão acentuada como o MSE. Portanto, não é tão sensível aos valores discrepantes como o MSE.

De forma sucinta, se os dados possuírem valores típicos ou valores atípicos que não queremos valorizar, deve ser usado o MAE; se os dados possuírem valores típicos cuja presença queremos detetar, deve ser usado o RMSE. O RMSE tem o benefício de penalizar grandes erros, pode por isso ser mais apropriado em alguns casos. Tanto o MAE quanto o RMSE expressam o erro médio de previsão do modelo em unidades da variável de interesse. Todas as métricas referidas podem variar de 0 a ∞ e são indiferentes à direção dos erros. São métricas negativamente orientadas, o que significa que valores mais baixos apontam para uma boa capacidade preditiva, enquanto valores elevados sugerem o contrário.

4.7 Métodos Explicativos

O desenvolvimento desta dissertação tem também como objetivo produzir explicações para cada característica e os seus respetivos valores, de forma a obter a sua importância e determinar se têm uma influência positiva ou negativa nos algoritmos.

As interpretações das metodologias de explicação permitirão a melhor compreensão dos mecanismos em causa e o aperfeiçoamento de indicadores, permitindo a redefinição de estratégias, levando a que sistemas de CRM beneficiem com este sistema.

Através do uso da metodologia de explicação geral EXPLAIN ou IME, como visto na Secção 3.1.8, a função *explainVis* gera explicações e respetivas visualizações para os dados sobre os quais queremos justificações. Os métodos EXPLAIN e IME são implementados na *package* R *ExplainPrediction*. As explicações suportam vários modelos implementados nos pacotes *CORElearn*. Uma explicação de uma previsão tanto pode ser dada para instâncias (por instâncias, entenda-se amostras) individuais como para a agregação de explicações de instâncias que irá fornecer uma explicação geral do modelo como um todo. Esta é uma função de explicação de *front-end*, que chama internamente outras funções.

Na explicação geral do modelo, todos os valores/intervalos numéricos dos atributos são visualizados, bem como um resumo ponderado sobre todos esses valores. Nas visualizações ao nível da instância, são apresentadas as contribuições de cada *feature*, bem como as contribuições médias das *features* no subconjunto de treino.

Tal como referido na Secção 3.1.8, o método EXPLAIN não tem a capacidade de detetar as dependências entre características, problema este resolvido pelo método IME. Ainda assim o método escolhido a ser usado nesta dissertação foi o método EXPLAIN, pois o método IME é lento quando comparado ao método EXPLAIN e quando aplicado a grandes conjuntos de dados. Esta decisão foi também tomada com base no facto de não ter capacidade computacional suficiente à disposição. Não obstante, o método EXPLAIN também se revelou demorado, sendo esta demora explicada pela insuficiente capacidade computacional.

Capítulo 5

Resultados

Este capítulo contém os resultados e respetiva análise de modo a permitir discutir o desempenho.

No presente capítulo são apresentados, analisados e comparados os resultados dos três algoritmos utilizados, conforme referidos na Secção 4.5.

Para comparação dos resultados, para os algoritmos de classificação, foram analisadas apenas no subconjunto Antiguidade 0 - 10 anos, as seguintes métricas: *Confusion Matrix*, *Classification Report* e curva ROC. Os resultados para os subconjuntos Antiguidade 11 - 40 anos e 41 - 94 anos encontram-se no Apêndice A.

Para os algoritmos de regressão, foram analisadas as métricas MAE, MSE e RMSE em todos os subconjuntos.

Em relação aos resultados dos Métodos Explicativos, estes foram gerados com o modelo obtido pelo algoritmo RF *Classifier* e utilizando o método EXPLAIN, apenas para o subconjunto Antiguidade 0 - 10 anos, atendendo ao facto de ser o subconjunto mais relevante e à morosidade da produção dos resultados. São apresentados e analisados parte dos atributos, sendo que os restantes se encontram no Apêndice B.

5.1 Resultados algoritmos de classificação

5.1.1 Resultados *Confusion Matrix*

Este sistema de classificação foi treinado para distinguir as classes ACT, DEMIT, DEMIT_AUTO e SUSP, cada classe representa um estado (tipificações referidas na Tabela 4.2). A *Confusion Matrix* irá resumir os resultados do modelo obtido do algoritmo, permitindo assim uma inspeção mais aprofundada.

Para avaliar os resultados preparados em ambientes de execução, foram executados, para comparação posterior, os algoritmos RF *Classifier*, MLP *Classifier* e SVM *Classifier*. Todas as *Confusion Matrix* geradas com os dados de treino, dos modelos obtidos pelos algoritmos referidos, foram obtidas usando uma amostra de 40 000 sócios, enquanto que as *Confusion Matrix* geradas com os dados de teste foram obtidas usando uma amostra de 10 000 sócios.

De forma a facilitar a análise da *Confusion Matrix* foram calculadas, em percentagem, as taxas de TP.

Algoritmo RF *Classifier*

Através da *Confusion Matrix* do modelo obtido pelo algoritmo RF *Classifier* nos dados de treino, Tabela 5.1a, podemos concluir que o sistema previu que a taxa de TP dos sócios ACT é de 95.6%. Relativamente aos sócios DEMIT a taxa de TP é de 94.3%, aos sócios DEMIT_AUTO é de 79.2% e aos sócios SUSP é de 100%. Analisando a *Confusion Matrix* nos dados de teste, Tabela 5.1b, sócios ACT apresentam uma taxa de TP de 82.7%, sócios DEMIT uma taxa de 42.1%, sócios DEMIT_AUTO uma taxa de 64.3% e sócios SUSP uma taxa de 0% (este valor explica-se pelo facto de nos dados de teste não existirem dados da classe SUSP).

Tabela 5.1: *Confusion Matrix* do modelo obtido pelo algoritmo RF *Classifier* nos dados de treino (5.1a) e nos dados de teste (5.1b) para o subconjunto Antiguidade 0 - 10 anos.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	13700	79	1297	0
	DEMIT	412	4853	2961	0
	DEMIT_AUTO	201	215	16253	0
	SUSP	15	2	8	4

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	2929	167	675	0
	DEMIT	292	451	1245	0
	DEMIT_AUTO	313	449	3465	0
	SUSP	6	3	5	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Em resumo, os resultados da *Confusion Matrix* do modelo obtido pelo algoritmo RF *Classifier* nos dados de treino mostram resultados bastante positivos. Relativamente aos dados de teste há uma diminuição da taxa de TP para todas as classes, no entanto, a classe que apresenta resultados menos satisfatórios é a classe DEMIT_AUTO. Quanto à classe

SUSP, como já referido anteriormente, esta não apresenta registos pelo que não é possível uma análise comparativa.

Algoritmo MLP *Classifier*

Na *Confusion Matrix* do modelo obtido pelo algoritmo MLP *Classifier* nos dados de treino, Tabela 5.2a, a taxa de TP dos sócios ACT é de 72%, dos sócios DEMIT é de 33.7% e dos sócios DEMIT_AUTO é de 55.4%. Nos dados de teste, Tabela 5.2b, as taxas são de 72.9% para os sócios ACT, 26.7% para os sócios DEMIT, 55.9% para os sócios DEMIT_AUTO. Tanto nos dados de treino como de teste não existem registos da classe SUSP.

Tabela 5.2: *Confusion Matrix* do modelo obtido pelo algoritmo MLP *Classifier* nos dados de treino (5.2a) e nos dados de teste (5.2b) para o subconjunto Antiguidade 0 - 10 anos.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	9277	71	5728	0
	DEMIT	1893	61	6272	0
	DEMIT_AUTO	1697	48	14924	0
	SUSP	17	1	11	0

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	2306	22	1443	0
	DEMIT	437	12	1539	0
	DEMIT_AUTO	413	11	3803	0
	SUSP	5	0	9	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Comparando os resultados da *Confusion Matrix* do modelo obtido pelo algoritmo MLP *Classifier* nos dados de treino com os resultados nos dados de teste, observamos que as taxas de TP se mantiveram constantes, com a exceção da classe DEMIT que sofreu um diminuição. Relativamente ao modelo obtido pelo algoritmo RF *Classifier*, o modelo obtido pelo algoritmo MLP *Classifier* apresenta um desempenho inferior.

Algoritmo SVM *Classifier*

Quanto à *Confusion Matrix* do modelo obtido pelo algoritmo SVM *Classifier* nos dados de treino, Tabela 5.3a, as taxas de TP são de 74.4% para sócios ACT, 65.1% para os sócios DEMIT e 58.9% para os sócios DEMIT_AUTO. Nos dados de teste, Tabela 5.3b, os sócios ACT apresentam um taxa de 71.6%, uma taxa de 38.4% para os sócios DEMIT e uma taxa de 56.8% para os sócios DEMIT_AUTO. Mais uma vez, tanto nos dados de treino como de teste não existem registos da classe SUSP.

Tabela 5.3: *Confusion Matrix* do modelo obtido pelo algoritmo SVM *Classifier* nos dados de treino (5.3a) e nos dados de teste (5.3b) para o subconjunto Antiguidade 0 - 10 anos.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	9962	275	4839	0
	DEMIT	1755	1045	5426	0
	DEMIT_AUTO	1646	282	14741	0
	SUSP	18	2	9	0

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	2339	93	1339	0
	DEMIT	442	122	1424	0
	DEMIT_AUTO	481	100	3646	0
	SUSP	4	2	8	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Comparando os resultados da *Confusion Matrix* do modelo obtido pelo algoritmo SVM *Classifier* e, tal como foi observado nos resultados da *Confusion Matrix* do modelo obtido pelo algoritmo MLP *Classifier*, nos dados de treino com os resultados nos dados de teste, observamos que as taxas de TP se mantiveram constantes, com a exceção da classe DEMIT que sofreu um diminuição.

De forma geral, o modelo obtido pelo algoritmo RF *Classifier* é o que apresenta os resultados mais satisfatórios.

5.1.2 Resultados *Classification Report*

Como referido na Secção 4.5, para a previsão do ESTADO foram utilizados três algoritmos para comparação. Dentro das classes que pretendemos prever, a classe SUSP não será considerada para análise pois apresenta um *support* extremamente baixo.

De acordo com o modelo obtido pelo algoritmo RF *Classifier* nos dados de treino (Tabela 5.4a), se este classificar que um elemento é ACT, então 96% das vezes está correto, se classificar que um elemento é DEMIT, então 94% das vezes está correto e se classificar que um elemento é DEMIT_AUTO então 79% das vezes está correto. Onde o modelo falha um pouco é no *recall* da classe DEMIT, o que significa que o modelo detetou apenas 59% dos DEMIT de todos os DEMIT disponíveis. Os resultados obtidos para os dados de teste no modelo obtido pelo algoritmo RF *Classifier* (Tabela 5.4b), mostram que se este classificar que um elemento é ACT, então 83% das vezes está correto, se classificar que um elemento é DEMIT, então 42% das vezes está correto e se classificar que um elemento

é DEMIT_AUTO então 64% das vezes está correto. Mais uma vez é refletido que onde o modelo apresenta algumas falhas é no *recall* da classe DEMIT, o que significa que o modelo detetou apenas 23% dos DEMIT de todos os DEMIT disponíveis. Em suma, estes são resultados satisfatórios tendo em conta a quantidade de dados que foi utilizada para treino e teste deste modelo.

Quanto aos outros dois algoritmos utilizados, o MLP *Classifier* (Tabela 5.5) e o SVM *Classifier* (Tabela 5.6), os respetivos modelos apresentam resultados consideravelmente menos satisfatórios.

O algoritmo que apresentou melhores resultados foi o RF *Classifier* para todos os subconjuntos definidos em função da ANTIGUIDADE.

Tabela 5.4: *Classification Report* do modelo obtido pelo algoritmo RF *Classifier* nos dados de treino (5.4a) e nos dados de teste (5.4b) para o subconjunto Antiguidade 0 - 10 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.96	0.91	0.93	15076
DEMIT	0.94	0.59	0.73	8226
DEMIT_AUTO	0.79	0.98	0.87	16669
SUSP	1.00	0.14	0.24	29
accuracy			0.87	40000
macro avg	0.92	0.65	0.69	40000
weighted avg	0.89	0.87	0.86	40000

(a) *Classification Report* do modelo nos dados de treino.

	precision	***recall***	***f1-score***	***support***
ACT	0.83	0.78	0.80	3771
DEMIT	0.42	0.23	0.29	1988
DEMIT_AUTO	0.64	0.82	0.72	4227
SUSP	0.00	0.00	0.00	14
accuracy			0.68	10000
macro avg	0.47	0.46	0.45	10000
weighted avg	0.67	0.68	0.67	10000

(b) *Classification Report* do modelo nos dados de teste.

Tabela 5.5: *Classification Report* do modelo obtido pelo algoritmo MLP *Classifier* nos dados de treino (5.5a) e nos dados de teste (5.5b) para o subconjunto Antiguidade 0 - 10 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.72	0.62	0.66	15076
DEMIT	0.34	0.01	0.01	8226
DEMIT_AUTO	0.79	0.98	0.68	16669
SUSP	0.00	0.00	0.00	29
accuracy			0.61	40000
macro avg	0.40	0.38	0.34	40000
weighted avg	0.57	0.61	0.54	40000

(a) *Classification Report* do modelo nos dados de treino.

	precision	***recall***	***f1-score***	***support***
ACT	0.73	0.61	0.67	3771
DEMIT	0.27	0.01	0.01	1988
DEMIT_AUTO	0.56	0.90	0.69	4227
SUSP	0.00	0.00	0.00	14
accuracy			0.61	10000
macro avg	0.39	0.38	0.34	10000
weighted avg	0.56	0.61	0.54	10000

(b) *Classification Report* do modelo nos dados de teste.

Tabela 5.6: *Classification Report* do modelo obtido pelo algoritmo SVM *Classifier* nos dados de treino (5.6a) e nos dados de teste (5.6b) para o subconjunto Antiguidade 0 - 10 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.74	0.66	0.70	15076
DEMIT	0.65	0.13	0.21	8226
DEMIT_AUTO	0.59	0.88	0.71	16669
SUSP	0.00	0.00	0.00	29
accuracy			0.64	40000
macro avg	0.50	0.42	0.41	40000
weighted avg	0.66	0.64	0.60	40000

(a) *Classification Report* do modelo nos dados de treino.

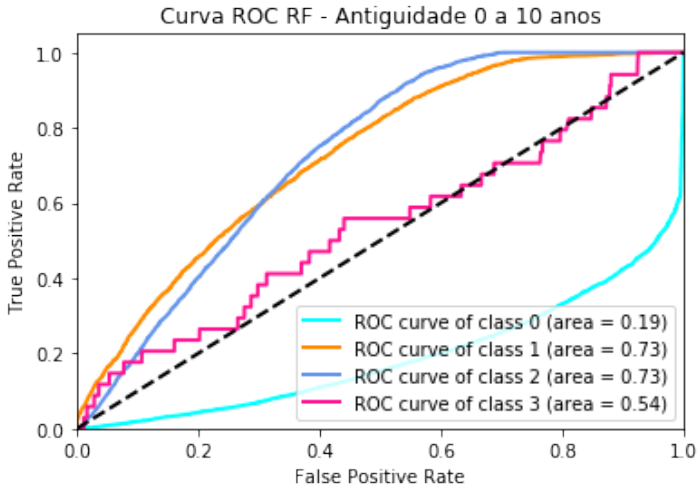
	precision	***recall***	***f1-score***	***support***
ACT	0.72	0.62	0.66	3771
DEMIT	0.38	0.06	0.11	1988
DEMIT_AUTO	0.57	0.86	0.69	4227
SUSP	0.00	0.00	0.00	14
accuracy			0.61	10000
macro avg	0.42	0.39	0.36	10000
weighted avg	0.59	0.61	0.56	10000

(b) *Classification Report* do modelo nos dados de teste.

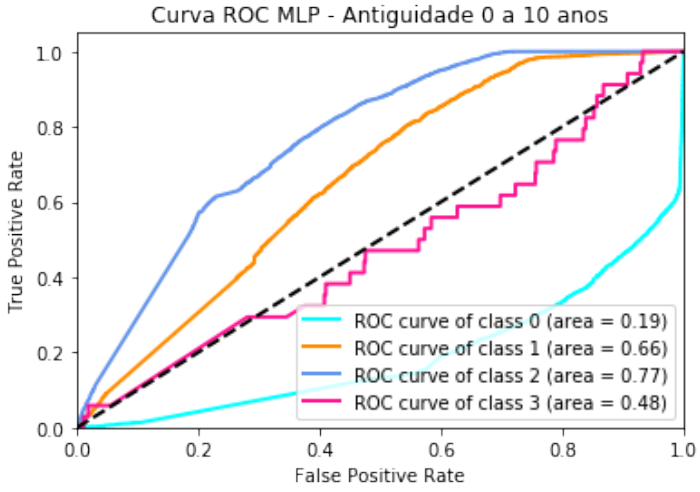
5.1.3 Resultados Curva ROC

De acordo com os gráficos resultantes que representam a curva ROC, para o modelo obtido pelo algoritmo RF *Classifier* (Figura 5.1a), este apresenta uma AUC de 0.73 tanto para a classe ACT como para a classe DEMIT, uma AUC de 0.54 para a classe DEMIT_AUTO e uma AUC de 0.19 para a classe SUSP. Tendo em conta que quanto maior a AUC, melhor a capacidade do modelo de fazer previsões corretas, podemos concluir que o modelo tem um desempenho razoável ao separar as classes ACT e DEMIT, que não tem a capacidade de separar a classe DEMIT_AUTO e que separa de forma errada a classe SUSP.

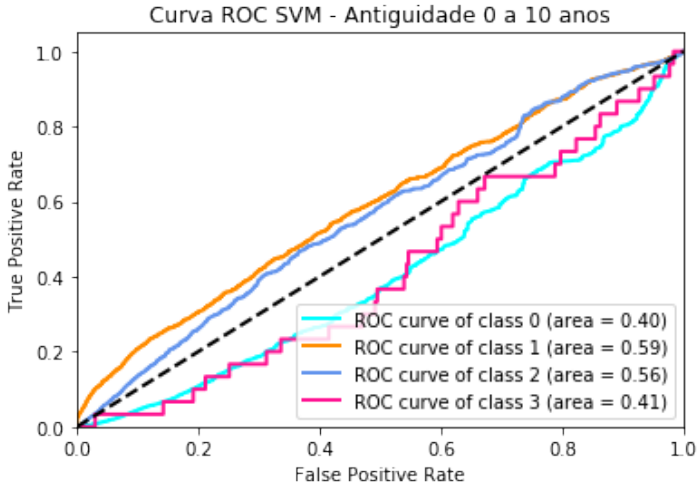
Relativamente aos outros dois algoritmos considerados, o modelo resultante do algoritmo MLP *Classifier* (Figura 5.1b) apresenta resultados semelhantes ao do modelo obtido pelo algoritmo RF *Classifier* e o modelo resultante do algoritmo SVM *Classifier* (Figura 5.1c) mostra que o modelo não tem a capacidade de separar qualquer uma das classes.



(a) Curva ROC RF.



(b) Curva ROC MLP.



(c) Curva ROC SVM.

Figura 5.1: Curvas ROC para os três algoritmos testados com incidência no subconjunto Antiguidade 0 - 10 anos.

5.2 Resultados algoritmos de regressão

Como referido na Secção 4.5, para a previsão da ANTIGUIDADE foram utilizados três algoritmos para comparação.

Para o subconjunto Antiguidade 0 - 10 anos o MAE é 0.54, o que é considerado um valor bastante satisfatório pois, para este caso, a Antiguidade varia entre 0 e 10 anos. Relativamente à análise da métrica MSE, é esperado que apresente um valor superior que o MAE devido à influência dos valores discrepantes. O MSE é 0.69, concluindo assim que de facto o MSE apresenta uma ordem de magnitude superior que o MAE. O RMSE correspondente é de 0.83. Todos os resultados das métricas de regressão para o subconjunto Antiguidade 0 - 10 anos para os três algoritmos, estão apresentados na Tabela 5.7.

Para o subconjunto Antiguidade 11 - 40 anos o MAE é 3.44, o que mais uma vez pode ser considerado um valor satisfatório pois, apesar de ter um valor superior relativamente ao subconjunto Antiguidade 0 - 10 anos, neste caso a Antiguidade varia entre 11 e 40, sendo este um intervalo superior. Quanto ao MSE, tem um valor bastante superior ao MAE, 21.90. Estes valores superiores podem ser justificados pelo facto de haver uma maior dispersão do valor da antiguidade nas amostras e para valores mais elevados de antiguidade existirem poucas amostras, levando assim a um erro maior. O RMSE correspondente é de 4.68. Todos os resultados das métricas de regressão para o subconjunto Antiguidade 11 - 40 anos para os três algoritmos estão apresentados na Tabela 5.8.

Para o subconjunto Antiguidade 41 - 94 anos o MAE é 5.31 e o MSE 45.14. Estes valores vão de encontro, mais uma vez, ao facto de a antiguidade apresentar uma grande variação. É ainda mais explicito neste subconjunto o facto de existirem poucas amostras de dados para os diferentes valores de antiguidade, o que justifica o elevado valor do MSE. O RMSE correspondente é de 6.71. Todos os resultados das métricas de regressão para o subconjunto Antiguidade 41 - 94 anos para os três algoritmos, estão apresentados na Tabela 5.9.

O algoritmo que apresentou melhores resultados foi o RF *Regressor* para todos os subconjuntos definidos em função da ANTIGUIDADE. Em suma, estes resultados são bastante afetados pelo número limitado de amostras de dados que foram utilizados. Um maior número de amostras iria levar certamente a uma melhor previsão, com menor erro.

Tabela 5.7: Resultados das métricas de avaliação dos algoritmos de regressão no subconjunto Antiguidade 0 - 10 anos.

Métrica	RF *Regressor*	MLP *Regressor*	*Linear* SVR
MAE	0.54	0.59	0.86
MSE	0.69	0.77	1.48
RMSE	0.83	0.88	1.21

Tabela 5.8: Resultados das métricas de avaliação dos algoritmos de regressão no subconjunto Antiguidade 11 - 40 anos.

Métrica	RF *Regressor*	MLP *Regressor*	*Linear* SVR
MAE	3.44	3.79	3.91
MSE	21.90	25.22	27.81
RMSE	4.68	5.02	5.27

Tabela 5.9: Resultados das métricas de avaliação dos algoritmos de regressão no subconjunto Antiguidade 41 - 94 anos.

Métrica	RF *Regressor*	MLP *Regressor*	*Linear* SVR
MAE	5.31	6.03	6.03
MSE	45.14	55.68	59.39
RMSE	6.71	7.46	7.70

5.3 Resultados Métodos Explicativos

O modelo gerado e treinado pelo algoritmo RF *Classifier* em função da ANTIGUIDADE, no subconjunto Antiguidade 0 - 10 anos foi usado como *input* para o método de explicação EXPLAIN.

Para entender o problema ao nível do modelo, todas as explicações para os dados de treino são combinadas. Foram produzidas visualizações de todos os atributos e respetivos valores, no subconjunto Antiguidade 0 - 10 anos, para os sócios com estado ACT. Apenas foram produzidas as visualizações para o subconjunto Antiguidade 0 - 10 anos pois, a nível de negócio é o que realmente importa analisar, não esquecendo também o elevado tempo de execução.

Uma visualização mostra o atributo e os seus valores. Os indicadores de impacto para os valores dos atributos espalham-se pelo eixo horizontal, o que indica que são possíveis tanto impactos positivos como negativos para os valores de cada atributo. Os valores de um atributo específico apresentam informações mais focadas que o atributo no geral. O impacto dos atributos no resultado é expresso como o peso da evidência (WE). A barra que representa o atributo é a média ponderada do impacto dos seus valores, representando assim a sua contribuição geral no modelo.

De seguida são apresentadas as explicações e respetivas interpretações para os atributos considerados mais relevantes para o estado ACT. Ou seja, de que forma é que os atributos contribuem para que o estado de um sócio seja ativo. Todas as restantes visualizações encontram-se no Apêndice B.

Observando o atributo ANTIGUIDADE (Figura 5.2) como um todo na explicação do modelo, vemos que os seus impactos positivo e negativo são aproximadamente os mesmos; portanto não se pode tirar uma conclusão geral sobre o seu impacto. Ao analisar os valores específicos, é possível formar um melhor entendimento. Todos os valores apresentam um

impacto misto. Os valores menores que 2.5 apresentam um impacto positivo significativo, enquanto que os restantes geralmente não contribuem positivamente para o resultado.

Quanto ao atributo IDADE (Figura 5.3) podemos constatar que tem um impacto mais positivo como um todo na explicação do modelo. Entre os valores do atributo, valores inferiores a 18.5 são indicadores de um impacto mais positivo, enquanto os restantes têm um maior impacto negativo.

O atributo SEXO (Figura 5.4) tem um ligeiro impacto mais positivo como um todo na explicação do modelo. O valor *F* apresenta um impacto positivo significativo enquanto que o valor *M* apresenta um impacto negativo.

Relativamente ao atributo TIPO_SOCIO (Figura 5.5), podemos concluir que tem um impacto significativo como um todo na explicação do modelo. Os valores *D*, *E*, *F* e *M* são indicadores de um impacto bastante positivo, enquanto que os restantes valores não têm uma influência tão forte, apresentando até na sua maioria um impacto negativo (o significado dos valores dos diferentes tipos de sócio encontram-se descritas na Tabela A.2).

No que diz respeito ao atributo TEM_ASSIST_VIAGEM (Figura 5.6) podemos constatar que apresenta um impacto positivo quase residual como um todo na explicação do modelo, visto que os seus impactos positivo e negativo são quase iguais. No entanto, ao analisar os valores de forma individual, ter assistência em viagem tem um impacto bastante positivo. Não ter assistência em viagem revela um impacto negativo.

À semelhança do atributo TEM_ASSIST_VIAGEM, o atributo TEM_SEGURO (Figura 5.7) apresenta também um impacto positivo quase residual como um todo na explicação do modelo. Mais especificamente, ter seguro tem consequências positivas enquanto que não ter apresenta consequências maioritariamente negativas.

O atributo TEM_CARTAO_SAUDE (Figura 5.8) segue o padrão do atributo ANTIGUIDADE, apresentando impactos positivos e negativos idênticos como um todo na explicação do modelo. No entanto, ter cartão de saúde impacta de forma positiva ao passo que não ter tem efeitos negativos.

5.3.1 Possíveis propostas para o ACP

Face aos resultados, seria benéfico para o ACP criar campanhas para sócios de promoções para Assistência em Viagem, Seguro e Cartão de Saúde visto que estas condicionantes têm impacto no estado de um sócio isto é, têm uma influência positiva no facto de um sócio permanecer com estado ativo. Fazer campanhas para jovens sócios também seria vantajoso uma vez que, tal como os resultados mostram, sócios mais jovens têm um grande impacto para que o estado seja ativo. É de salientar que, poderá ser também proveitoso promover novas ações para os tipos de sócio que não contribuem favoravelmente para que o estado seja ativo, revertendo assim a situação.

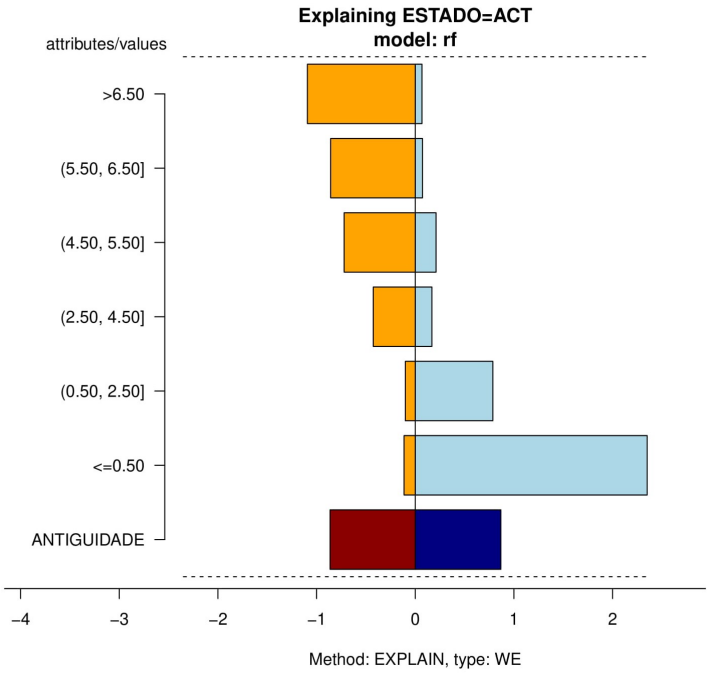


Figura 5.2: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo ANTIGUIDADE.

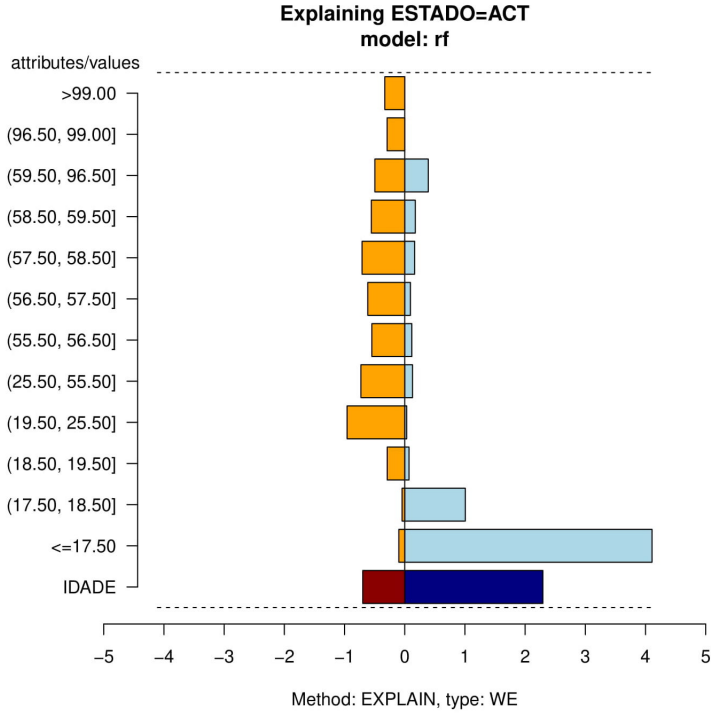


Figura 5.3: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo IDADE.

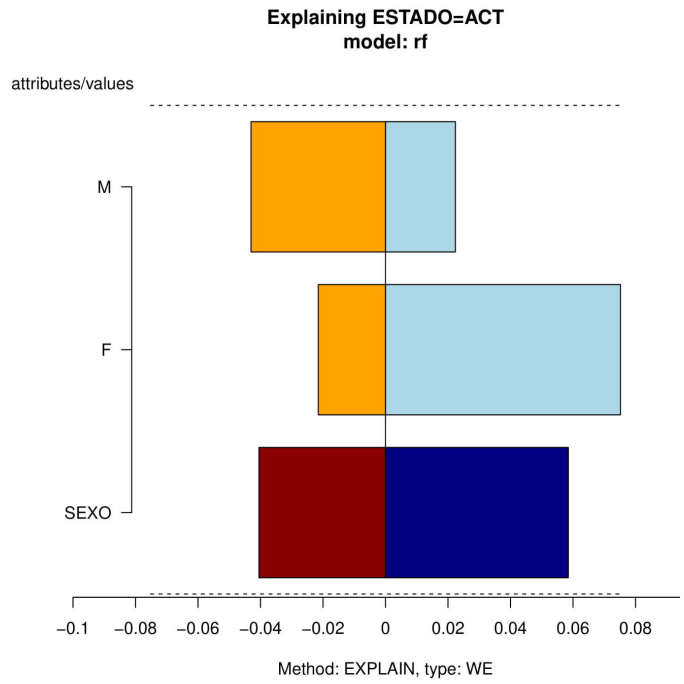


Figura 5.4: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo SEXO.

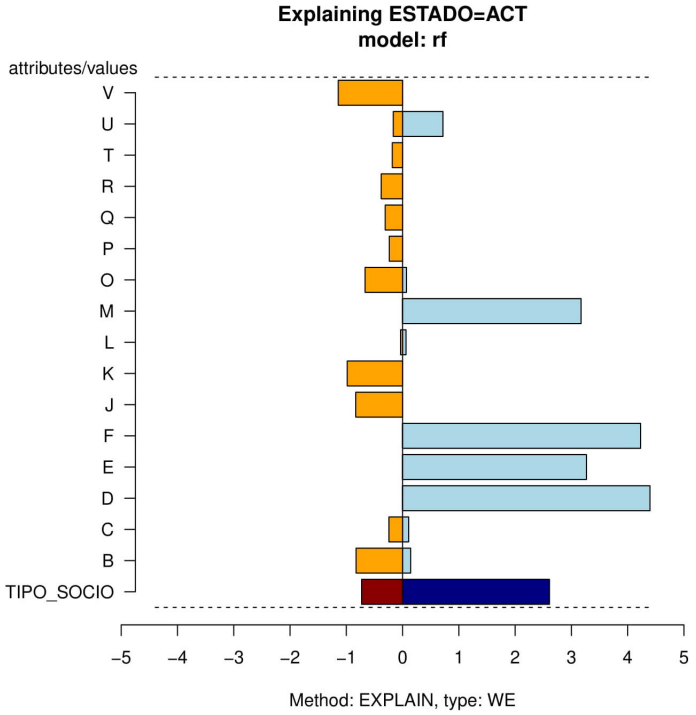


Figura 5.5: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TIPO_SOCIO.

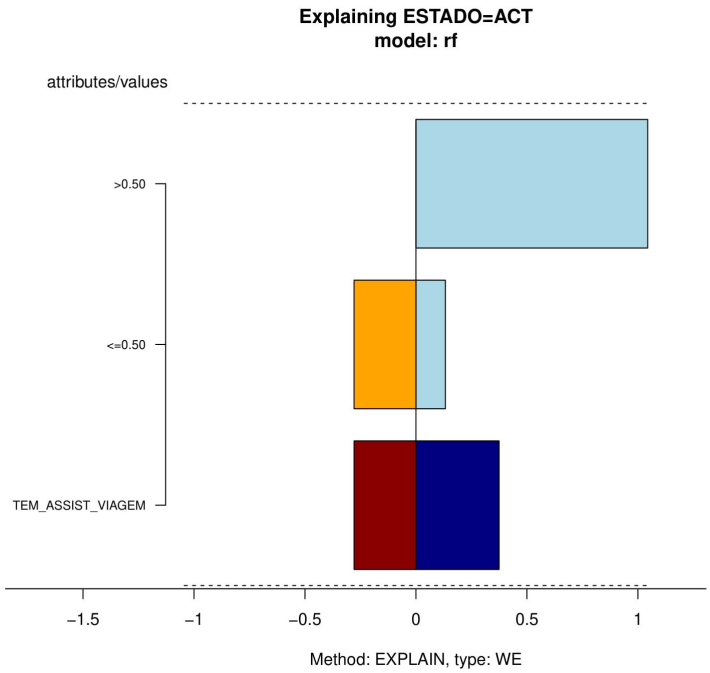


Figura 5.6: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_ASSIST_VIAGEM.

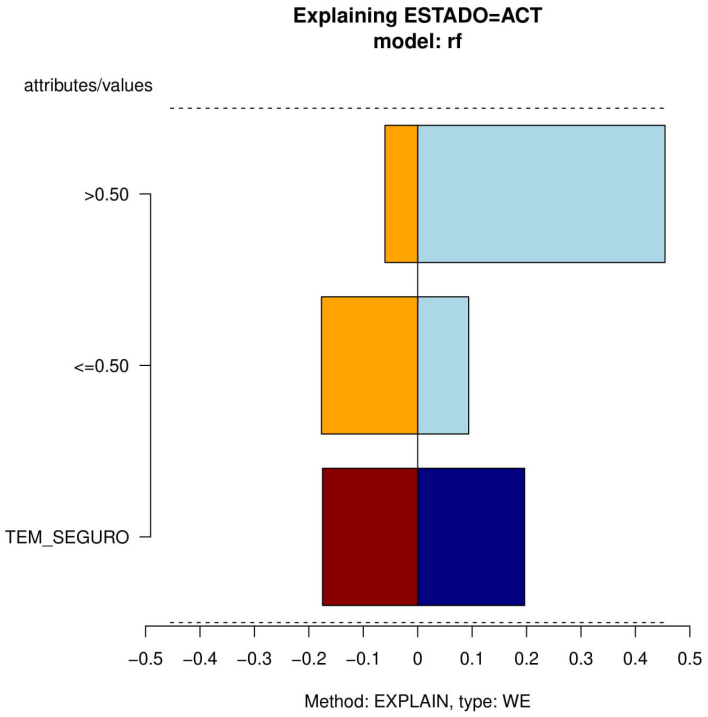


Figura 5.7: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_SEGURO.

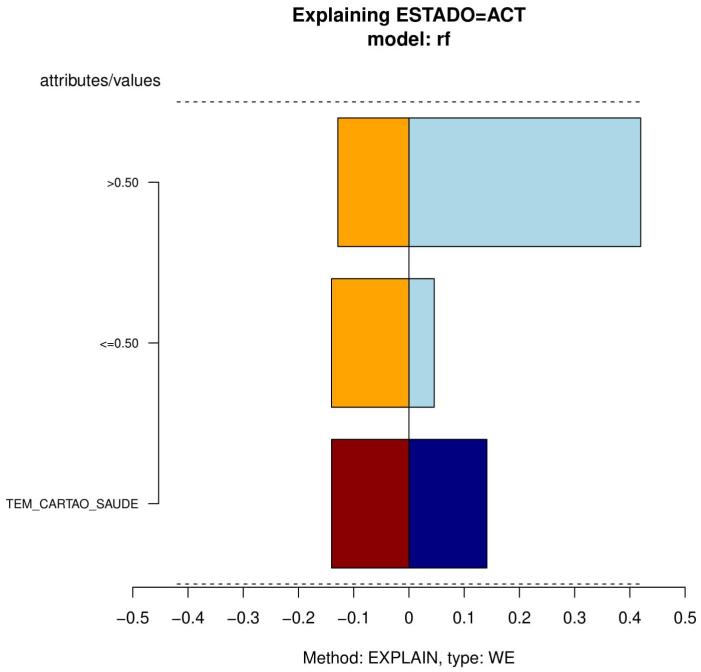


Figura 5.8: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_CARTAO_SAUDE.

Capítulo 6

Conclusões e Trabalho Futuro

Neste capítulo, são apresentadas as conclusões da presente dissertação e mencionados possíveis trabalhos futuros.

6.1 Conclusões

Perante o desafio de analisar as características que podem influenciar a permanência ou não de um sócio no ACP foram analisados vários algoritmos de classificação e de regressão aplicados aos dados fornecidos pelo ACP. Em particular, foi dada especial importância, por um lado, à capacidade de prever qual o estado futuro de um sócio perante a alteração do seu contexto no ACP (por exemplo: assistência em viagem, cartão de saúde, etc.). Por outro lado, foi possível prever a antiguidade do sócio perante um contexto. Em suma, prever o tempo de permanência do sócio no ACP, tendo em conta as diversas características que traduzem o contexto e que ao longo do tempo possam vir a ser alteradas. Concluí que, tanto no caso de classificação como de regressão, o algoritmo RF mostrou ser a melhor abordagem para o conjunto de dados utilizados para o efeito.

Foram também utilizados métodos explicativos que permitiram identificar, dentro das características do conjunto de dados, quais as que contribuem positiva ou negativamente para que um sócio esteja ativo. As interpretações das metodologias de explicação possibilitam uma melhor compreensão dos mecanismos em causa e o aperfeiçoamento de indicadores, permitindo a redefinição de estratégias, proporcionando que os sistemas de CRM beneficiem com este sistema. Estes factos levaram-me a concluir que, efetivamente, podem ser tomadas decisões ao nível do negócio para contribuir que um sócio permaneça ativo no ACP.

O trabalho desenvolvido nesta dissertação propiciou-me também uma maior tomada de consciência relativamente aos fatores que podem influenciar a fidelidade do sócio,

ajustando as suas características, obtidas a partir dos dados de consumo disponibilizados.

6.2 Trabalho Futuro

Este trabalho possui algumas limitações relacionadas com os dados disponíveis e com a capacidade computacional, o que leva à necessidade de futuras melhorias.

Primeiro que tudo, um dos fatores que tem impacto no desempenho e na precisão de um algoritmo são os dados utilizados no que diz respeito principalmente ao volume e à qualidade (correção e completude). Desta forma, o facto dos dados que foram disponibilizados estarem limitados por questões de RGPD pode ter influenciado os resultados. Segundo, e não menos importante, o facto de não ter capacidade computacional suficiente à disposição influenciou os resultados de forma significativa.

Ultrapassados estes fatores, será possível tornar os algoritmos mais precisos e obter a conclusão mais assertiva de que realmente os resultados estão corretos. Não pondo em causa os resultados obtidos, é importante referir que estes, por vezes, precisam do senso crítico humano para serem validados e garantir que fazem sentido serem aplicados no negócio.

Uma das propostas para trabalhos futuros consiste em aumentar a capacidade computacional para que os dados sejam utilizados na sua totalidade, de forma a produzir resultados mais confiáveis. Será também pertinente ter *features* mais informativas e conseguir testá-las. Desta forma, seria possível acompanhar a evolução do estado e antiguidade dos sócios à medida que se aplicam ações de retenção; conseguir prever quais os produtos/serviços a propor a um sócio tendo em conta os que o sócio já tem, tentando maximizar a sua antiguidade; propor produtos/serviços e promoções a *prospects* para que incentive a entrada de novos associados.

Do ponto de vista operacional, uma melhoria significativa que proponho será a integração do protótipo com os sistemas de canais de contacto dos *prospects* e sócios, de modo a que quando exista uma interação de entrada por parte de um destes *prospects*/sócios, o modelo seja imediatamente executado para apresentar as previsões e alertar em tempo real os agentes do ACP, ou numa evolução adicional, apresentar logo a lista de ações possíveis que o agente está autorizado a propor ao cliente/sócio.

Bibliografia

[1] V. N. Vapnik. *The Nature of Statistical Learning Theory*. New York, USA, 1995.

[2] I. Guyon, J. Weston, S. Barnhill e V. Vapnik. "Gene Selection for Cancer Classification using Support Vector Machines". Em: (2002), pp. 389–422. ISSN: 10970282. DOI: 10.1002/bip.360320308.

[3] K.-j. Kim. "Financial time series forecasting using support vector machines". Em: *Neurocomputing 55* (2003), pp. 307–319. ISSN: 09252312. DOI: 10.1109/CIS.2014.22.

[4] S. Chowdhury, J. K. Sing, D. K. Basu e M. Nasipuri. "Face recognition by generalized two-dimensional FLD method and multi-class support vector machines". Em: *Applied Soft Computing Journal* 11.7 (2011), pp. 4282–4292. ISSN: 15684946. DOI: 10.1016/j.asoc.2010.12.002.

[5] W. S. McCulloch e W. Pitts. "A logical calculus of the ideas immanent in nervous activity". Em: *The bulletin of mathematical biophysics 5.4* (1943), 115–133.

[6] L. E. O. Breiman. "Random Forests". Em: (2001), pp. 5–32.

[7] I. Dalla Pozza, O. Goetz e J. M. Sahut. "Implementation effects in the relationship between CRM and its performance". Em: *Journal of Business Research* 89.February (2018), pp. 391–403. ISSN: 01482963. DOI: 10.1016/j.jbusres.2018.02.004.

[8] J. Ganesh, M. J. Arnold e K. E. Reynolds. "Understanding the Customer Base of Service Providers: An Examination of the Differences between Switchers and Stayers". Em: *Journal of Marketing* 64.3 (2000), pp. 65–87. ISSN: 0022-2429. DOI: 10.1509/jmkg.64.3.65.18028.

[9] S. Keaveney. "Customer Switching Behavior in Service Industries: An Exploratory Study". Em: *Journal of Marketing* 59.2 (1995), pp. 71–82.

[10] W. Verbeke, D. Martens, C. Mues e B. Baesens. "Building comprehensible customer churn prediction models with advanced rule induction techniques". Em: *Expert Systems with Applications* 38.3 (2011), pp. 2354–2364. ISSN: 09574174. DOI: 10.1016/j.eswa.2010.08.023. URL: http://dx.doi.org/10.1016/j.eswa.2010.08.023.

[11] A. Gustafsson, M. D. Johnson e I. Roos. "The Effects of Customer Satisfaction, Relationship Commitment Dimensions, and Triggers on Customer Retention". Em: *Journal of Marketing* 69.4 (2005), pp. 210–218. ISSN: 0022-2429. DOI: 10.1509/jmkg.2005.69.4.210.

[12] H. Hansen, B. M. Samuelsen e J. E. Sallis. "The moderating effects of need for cognition on drivers of customer loyalty". Em: *European Journal of Marketing* 47.8 (2013), pp. 1157–1176. ISSN: 03090566. DOI: 10.1108/03090561311324264.

[13] W. J. Reinartz e V. Kumar. "The Impact of Customer Relationship Characteristics on Profitable Lifetime Duration". Em: *Journal of High Energy Physics* 67 (2003). ISSN: 10298479.

[14] I. Nitzan e B. Libai. "Social Effects on Customer Retention." Em: *Journal of Marketing* 75.6 (2011), pp. 24–38.

[15] M. Colgate, K. Stewart e R. Kinsella. "Costumer defection: A study of the student market in Ireland". Em: *International Journal of Bank Marketing* 14.3 (1996), pp. 23–29.

[16] G. Torkzadeh, J. C. J. Chang e G. W. Hansen. "Identifying issues in customer relationship management at Merck-Medco". Em: *Decision Support Systems* 42.2 (2006), pp. 1116–1130. ISSN: 01679236. DOI: 10.1016/j.dss.2005.10.003.

[17] W. Verbeke, K. Dejaeger, D. Martens, J. Hur e B. Baesens. "New insights into churn prediction in the telecommunication sector: A profit driven data mining approach". Em: *European Journal of Operational Research* 218.1 (2012), pp. 211–229. ISSN: 03772217. DOI: 10.1016/j.ejor.2011.09.031. URL: http://dx.doi.org/10.1016/j.ejor.2011.09.031.

[18] K. Coussement e D. Van den Poel. "Churn prediction in subscription services: An application of support vector machines while comparing two parameter-selection techniques". Em: *Expert Systems with Applications* 34.1 (2008), pp. 313–327. ISSN: 09574174. DOI: 10.1016/j.eswa.2006.09.038.

[19] P. Langley, W. Iba e K. Thompson. "An Analysis of Bayessian Classifiers". Em: *AAAI'92 Proceedings of the tenth national conference on Artificial intelligence* (1992), pp. 223 –228. ISSN: 0-262-51063-4.

[20] R. R. Quinlan. "C4.5: Programs for Machine Learning". Em: *Morgan Kaufmann Publishers Inc* (1993), pp. 235–240.

[21] R. Kohavi. "Scaling up the accuracy of Naive-Bayes Classifiers : a Decision-Tree Hybrid". Em: *In Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (1996), pp. 202–207. URL: www.aaai.org.

[22] S. V. Sabnani. "Computer Security : A Machine Learning Approach". Em: *Technical Report, MSc in Information Security at Royal Holloway, University of London.* (2008).

[23] M. Robnik-Šikonja e I. Kononenko. "Explaining classifications for individual instances". Em: *IEEE Transactions on Knowledge and Data Engineering* 20.5 (2008), pp. 589–600. ISSN: 10414347. DOI: 10.1109/TKDE.2007.190734.

[24] E. Strumbelj e I. Kononenko. "An Efficient Explanation of Individual Classifications using Game Theory". Em: *Journal of Machine Learning Research* 11 (2010), pp. 1–18. ISSN: 1532-4435. DOI: 10.1145/2858036.2858529.

[25] E. Štrumbelj, I. Kononenko e M. Robnik Šikonja. "Explaining instance classifications with interactions of subsets of feature values". Em: *Data and Knowledge Engineering* 68.10 (2009), pp. 886–904. ISSN: 0169023X. DOI: 10.1016/j.datak.2009.01.004.

[26] F. Provost e T. Fawcett. "Data Science and its Relationship to Big Data and Data-Driven Decision Making". Em: *Big Data* 1.1 (2013), pp. 51–59. ISSN: 2167-6461. DOI: 10.1089/big.2013.1508.

[27] A McAfee e E Brynjolfsson. "Big data Big Data". Em: *Harvard Busines Review* 90.10 (2012), pp. 59–68. ISSN: 00178012. DOI: 10.1007/s12599-013-0249-5.

[28] E. Brynjolfsson, L. M. Hitt e H. H. Kim. "Strength in Numbers: How Does Data-Driven Decisionmaking Affect Firm Performance?" Em: *Ssrn* (2011). ISSN: 1556-5068. DOI: 10.2139/ssrn.1819486.

[29] R Indhumathi e S. Sathiyabama. "Reducing and Clustering high Dimensional Data through Principal Component Analysis". Em: 11.8 (2010), pp. 1–4.

[30] K. Pearson. "On lines and planes of closest fit to systems of points in space". Em: *Phil. Mag. 2* (doi:10.1080/14786440109462720) (1901), pp. 559–572.

[31] H. Hotelling. "Analysis of a complex of statistical variables into principal components". Em: *J. Educ. Psychol. 24* (doi:10.1037/h0071325) (1933), pp. 417–441, 498–520.

[32] H. Lynne J. Williams Abdi. "Principal Component Analysis". Em: *WIREs Comp Stat* 2 (2010), 433–59. URL: http://staff.ustc.edu.cn/~zwp/teach/MVA/abdi-awPCA2010.pdf.

[33] J. Pagès. "Analyse Factorielle Multiple Appliquée Aux Variables Qualitatives et Aux Données Mixtes". Em: *Rev. Statistique Appliquée (4)* (2002), pp. 5–37.

[34] J. Pagès. "Analyse factorielle de données mixtes". Em: *Rev. Statistique Appliquée LII (4)* (2004), pp. 93–111.

[35] J. Benzécri. "Ournal de la société statistique de". Em: *Tome II: L'analyse des correspondances* 3 (1973).

[36] M. A. Farquad, V. Ravi e S. B. Raju. "Churn prediction using comprehensible support vector machine: An analytical CRM application". Em: *Applied Soft Computing Journal* 19 (2014), pp. 31–40. ISSN: 15684946. DOI: 10.1016/j.asoc.2014.01.031. URL: http://dx.doi.org/10.1016/j.asoc.2014.01.031.

[37] M. Bohanec, M. Robnik-Šikonja e M. Kljajić Borštnar. "Explaining Machine Learning Predictions". Em: *Expert Systems with Applications* (2017).

[38] A. De Caigny, K. Coussement e K. W. De Bock. "A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees". Em: *European Journal of Operational Research* 269.2 (2018), pp. 760–772. ISSN: 03772217. DOI: 10.1016/j.ejor.2018.02.009.

[39] S. Ma e Y. Dai. "Principal component analysis based methods in bioinformatics studies". Em: *Briefings in Bioinformatics* 12.6 (jan. de 2011), pp. 714–722. ISSN: 1467-5463. DOI: 10.1093/bib/bbq090. eprint: https://academic.oup.com/bib/article-pdf/12/6/714/590324/bbq090.pdf. URL: https://doi.org/10.1093/bib/bbq090.

Apêndice

Tabelas Auxiliares

Tabela A.1: Descrição das *features* do *dataset*.

Feature	**Descrição**	**Tipo**
SEXO	Sexo do sócio	Categórica
TIPO_SOCIO	Tipo do sócio	Categórica
PAIS	País do sócio	Categórica
DISTRITO	Distrito do sócio	Categórica
CONCELHO	Concelho do sócio	Categórica
CP4	Código postal do sócio	Categórica
ESTADO	Estado do sócio	Categórica
CANAL	Canal de entrada do sócio	Categórica
RAZAO	Razão de entrada do sócio	Categórica
CLASSICOS	Indica qual o plano associado aos carros clássicos	Categórica
GOLFE	Indica qual o plano associado ao golfe	Categórica
CAMPANHA	Campanha de entrada do sócio	Categórica
FREGUESIA	Freguesia do sócio	Categórica
DATA_ESTADO	Data desde a qual o sócio se encontra com o estado associado	Categórica
RAZAO_ESTADO	Razão do estado do sócio	Categórica
ANTIGUIDADE	Antiguidade do sócio	Numérica
IDADE	Idade do sócio	Numérica
TEM_SOCIO_CONJUGE	Indica se o sócio tem sócio cônjuge	Numérica
TEM_CARTA_COND	Indica se o sócio tem carta de condução	Numérica
TEM_EMAIL	Indica se o sócio tem email	Numérica
TEM_TELEMOVEL	Indica se o sócio tem telemóvel	Numérica
TEM_TELEFONE	Indica se o sócio tem telefone	Numérica
TEM_SEGURO	Indica se o sócio tem seguro	Numérica
TEM_ASSIST_VIAGEM	Indica se o sócio tem assistência em viagem	Numérica
TEM_OUTRAS_ASSIST	Indica se o sócio tem outras assistências	Numérica
TEM_COMPRAS_PROD	Indica se o sócio tem compras de produtos	Numérica
TEM_COMPRAS_SERV	Indica se o sócio tem compras de serviços	Numérica
TEM_CARTAO_SAUDE	Indica se o sócio tem cartão de saúde	Numérica
TEM_CONSUMOS_BP	Indica se o sócio tem consumos BP	Numérica
TEM_RENOV_CARTA	Indica se o sócio tem renovação da carta de condução	Numérica
TEM_GOLFE	Indica se o sócio tem golfe	Numérica
TEM_CLASSICOS	Indica se o sócio tem carros clássicos	Numérica
NUMERO_EVENTOS	Número de interações do sócio com o ACP	Numérica

Tabela A.2: Tipos de Sócios.

Valor	Tipo de Sócio
B	BRONZE
C	CÔNJUGE
D	ESTRELA 0-13 sem assistência médica
E	ESTRELA 0-13 com assistência médica
F	JÚNIOR 14-17 sem assistência
G	JÚNIOR 14-17 com assistência 2
H	HONORÁRIO
I	JÚNIOR 14-17 com assistência 1
J	JOVEM
K	BRONZE (estrangeiro)
L	JOVEM 26-30
M	JÚNIOR 14-17 com assistência médica
O	OURO
P	PRATA MASTER
Q	PRATA
R	PLATINA
S	PLATINA sem assistência
T	PRATA MASTER sem assistência
U	PRATA sem assistência
V	ACP MOOVE

Tabela A.3: *Confusion Matrix* do modelo obtido pelo algoritmo RF *Classifier* nos dados de treino (A.3a) e nos dados de teste (A.3b) para o subconjunto Antiguidade 11 - 40 anos.

		Valor Atual			
Valor Previsto		ACT	DEMIT	DEMIT_AUTO	SUSP
	ACT	11837	735	307	0
	DEMIT	686	14418	1448	0
	DEMIT_AUTO	568	4682	5253	0
	SUSP	42	14	8	2

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
Valor Previsto		ACT	DEMIT	DEMIT_AUTO	SUSP
	ACT	2901	271	136	0
	DEMIT	249	3186	622	0
	DEMIT_AUTO	195	1452	970	0
	SUSP	16	2	0	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Tabela A.4: *Confusion Matrix* do modelo obtido pelo algoritmo MLP *Classifier* nos dados de treino (A.4a) e nos dados de teste (A.4b) para o subconjunto Antiguidade 11 - 40 anos.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	5624	5503	1752	0
	DEMIT	5833	8861	1858	0
	DEMIT_AUTO	2894	4325	3284	0
	SUSP	26	34	6	0

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	1465	1383	460	0
	DEMIT	1475	2118	464	0
	DEMIT_AUTO	741	1079	797	0
	SUSP	4	13	1	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Tabela A.5: *Confusion Matrix* do modelo obtido pelo algoritmo SVM *Classifier* nos dados de treino (A.5a) e nos dados de teste (A.5b) para o subconjunto Antiguidade 11 - 40 anos.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	840	10394	1645	0
	DEMIT	338	14393	1821	0
	DEMIT_AUTO	225	6976	3302	0
	SUSP	9	51	6	0

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	216	2667	425	0
	DEMIT	92	3501	464	0
	DEMIT_AUTO	45	1765	807	0
	SUSP	2	15	1	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Tabela A.6: *Confusion Matrix* do modelo obtido pelo algoritmo RF *Classifier* nos dados de treino (A.6a) e nos dados de teste (A.6b) para o subconjunto Antiguidade 41 - 94 anos.

		Valor Atual			
Valor Previsto		ACT	DEMIT	DEMIT_AUTO	SUSP
	ACT	10252	405	49	0
	DEMIT	658	10738	118	0
	DEMIT_AUTO	422	2195	986	0
	SUSP	50	11	4	0

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
Valor Previsto		ACT	DEMIT	DEMIT_AUTO	SUSP
	ACT	2453	164	23	0
	DEMIT	256	2551	111	0
	DEMIT_AUTO	126	729	50	0
	SUSP	7	3	0	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Tabela A.7: *Confusion Matrix* do modelo obtido pelo algoritmo MLP *Classifier* nos dados de treino (A.7a) e nos dados de teste (A.7b) para o subconjunto Antiguidade 41 - 94 anos.

		Valor Atual			
Valor Previsto		ACT	DEMIT	DEMIT_AUTO	SUSP
	ACT	5652	5054	0	0
	DEMIT	4846	6668	0	0
	DEMIT_AUTO	1495	2108	0	0
	SUSP	46	19	0	0

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
Valor Previsto		ACT	DEMIT	DEMIT_AUTO	SUSP
	ACT	1364	1276	0	0
	DEMIT	1196	1722	0	0
	DEMIT_AUTO	419	486	0	0
	SUSP	6	4	0	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Tabela A.8: *Confusion Matrix* do modelo obtido pelo algoritmo SVM *Classifier* nos dados de treino (A.8a) e nos dados de teste (A.8b) para o subconjunto Antiguidade 41 - 94 anos.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	3809	6897	0	0
	DEMIT	2598	8916	0	0
	DEMIT_AUTO	941	2662	0	0
	SUSP	34	31	0	0

(a) *Confusion Matrix* do modelo nos dados de treino.

		Valor Atual			
		ACT	DEMIT	DEMIT_AUTO	SUSP
Valor Previsto	ACT	959	1681	0	0
	DEMIT	655	2263	0	0
	DEMIT_AUTO	244	661	0	0
	SUSP	2	8	0	0

(b) *Confusion Matrix* do modelo nos dados de teste.

Tabela A.9: *Classification Report* do modelo obtido pelo algoritmo RF *Classifier* nos dados de treino (A.9a) e nos dados de teste (A.9b) para o subconjunto Antiguidade 11 - 40 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.90	0.92	0.91	12879
DEMIT	0.73	0.87	0.79	16552
DEMIT_AUTO	0.75	0.50	0.60	10503
SUSP	1.00	0.03	0.06	66
accuracy			0.79	40000
macro avg	0.84	0.58	0.59	40000
weighted avg	0.79	0.79	0.78	40000

(a) *Classification Report* do modelo nos dados de treino.

	precision	***recall***	***f1-score***	***support***
ACT	0.86	0.88	0.87	3308
DEMIT	0.65	0.79	0.71	4057
DEMIT_AUTO	0.56	0.37	0.45	2617
SUSP	0.00	0.00	0.00	18
accuracy			0.71	10000
macro avg	0.52	0.51	0.51	10000
weighted avg	0.70	0.71	0.69	10000

(b) *Classification Report* do modelo nos dados de teste.

Tabela A.10: *Classification Report* do modelo obtido pelo algoritmo MLP *Classifier* nos dados de treino (A.10a) e nos dados de teste (A.10b) para o subconjunto Antiguidade 11 - 40 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.39	0.44	0.41	12879
DEMIT	0.47	0.54	0.50	16552
DEMIT_AUTO	0.48	0.31	0.38	10503
SUSP	0.00	0.00	0.00	66
accuracy			0.44	40000
macro avg	0.34	0.32	0.32	40000
weighted avg	0.45	0.44	0.44	40000

(a) *Classification Report* do modelo nos dados de treino.

	precision	***recall***	***f1-score***	***support***
ACT	0.40	0.44	0.42	3308
DEMIT	0.46	0.52	0.49	4057
DEMIT_AUTO	0.46	0.30	0.37	2617
SUSP	0.00	0.00	0.00	18
accuracy			0.44	10000
macro avg	0.33	0.32	0.32	10000
weighted avg	0.44	0.44	0.43	10000

(b) *Classification Report* do modelo nos dados de teste.

Tabela A.11: *Classification Report* do modelo obtido pelo algoritmo SVM *Classifier* nos dados de treino (A.11a) e nos dados de teste (A.11b) para o subconjunto Antiguidade 11 - 40 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.59	0.07	0.12	12879
DEMIT	0.45	0.87	0.60	16552
DEMIT_AUTO	0.49	0.31	0.38	10503
SUSP	0.00	0.00	0.00	66
accuracy			0.46	40000
macro avg	0.38	0.31	0.37	40000
weighted avg	0.51	0.46	0.38	40000

(a) *Classification Report* do modelo nos dados de treino.

	precision	***recall***	***f1-score***	***support***
ACT	0.61	0.07	0.12	3308
DEMIT	0.44	0.86	0.58	4057
DEMIT_AUTO	0.48	0.31	0.37	2617
SUSP	0.00	0.00	0.00	18
accuracy			0.45	10000
macro avg	0.38	0.31	0.27	10000
weighted avg	0.50	0.45	0.37	10000

(b) *Classification Report* do modelo nos dados de teste.

Tabela A.12: *Classification Report* do modelo obtido pelo algoritmo RF *Classifier* nos dados de treino (A.12a) e nos dados de teste (A.12b) para o subconjunto Antiguidade 41 - 94 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.90	0.96	0.93	10706
DEMIT	0.80	0.93	0.86	11514
DEMIT_AUTO	0.85	0.27	0.41	3603
SUSP	0.00	0.00	0.00	65
accuracy			0.78	25888
macro avg	0.64	0.54	0.55	25888
weighted avg	0.85	0.85	0.83	25888

(a) *Classification Report* do modelo nos dados de treino.

	precision	***recall***	***f1-score***	***support***
ACT	0.86	0.93	0.89	2640
DEMIT	0.74	0.87	0.80	2918
DEMIT_AUTO	0.27	0.06	0.09	905
SUSP	0.00	0.00	0.00	10
accuracy			0.78	6473
macro avg	0.47	0.46	0.45	6473
weighted avg	0.72	0.78	0.74	6473

(b) *Classification Report* do modelo nos dados de teste.

Tabela A.13: *Classification Report* do modelo obtido pelo algoritmo MLP *Classifier* nos dados de treino (A.13a) e nos dados de teste (A.13b) para o subconjunto Antiguidade 41 - 94 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.47	0.53	0.50	10706
DEMIT	0.48	0.58	0.53	11514
DEMIT_AUTO	0.00	0.00	0.00	3603
SUSP	0.00	0.00	0.00	65
accuracy			0.48	25888
macro avg	0.24	0.28	0.26	25888
weighted avg	0.41	0.48	0.44	25888

(a) *Classification Report* do modelo nos dados de treino.

	precision	***recall***	***f1-score***	***support***
ACT	0.46	0.52	0.48	2640
DEMIT	0.49	0.59	0.54	2918
DEMIT_AUTO	0.00	0.00	0.00	905
SUSP	0.00	0.00	0.00	10
accuracy			0.48	6473
macro avg	0.24	0.28	0.26	6473
weighted avg	0.41	0.48	0.44	6473

(b) *Classification Report* do modelo nos dados de teste.

Tabela A.14: *Classification Report* do modelo obtido pelo algoritmo SVM *Classifier* nos dados de treino (A.14a) e nos dados de teste (A.14b) para o subconjunto Antiguidade 41 - 94 anos.

	precision	***recall***	***f1-score***	***support***
ACT	0.52	0.36	0.42	10706
DEMIT	0.48	0.77	0.59	11514
DEMIT_AUTO	0.00	0.00	0.00	3603
SUSP	0.00	0.00	0.00	65
accuracy			0.49	25888
macro avg	0.25	0.28	0.25	25888
weighted avg	0.43	0.49	0.44	25888

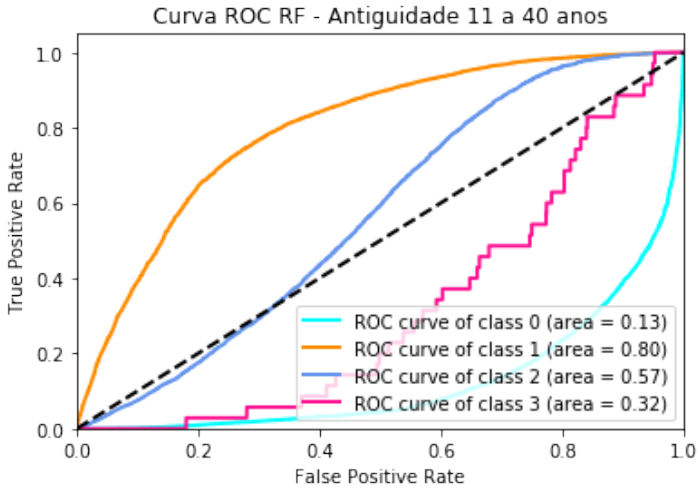
(a) *Classification Report* do modelo nos dados de treino.

	precision	***recall***	***f1-score***	***support***
ACT	0.52	0.36	0.43	2640
DEMIT	0.49	0.78	0.60	2918
DEMIT_AUTO	0.00	0.00	0.00	905
SUSP	0.00	0.00	0.00	10
accuracy			0.50	6473
macro avg	0.25	0.28	0.26	6473
weighted avg	0.43	0.50	0.44	6473

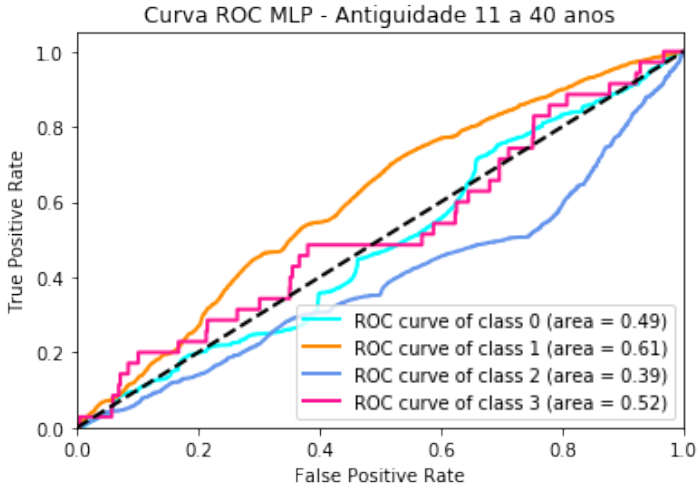
(b) *Classification Report* do modelo nos dados de teste.

Apêndice B

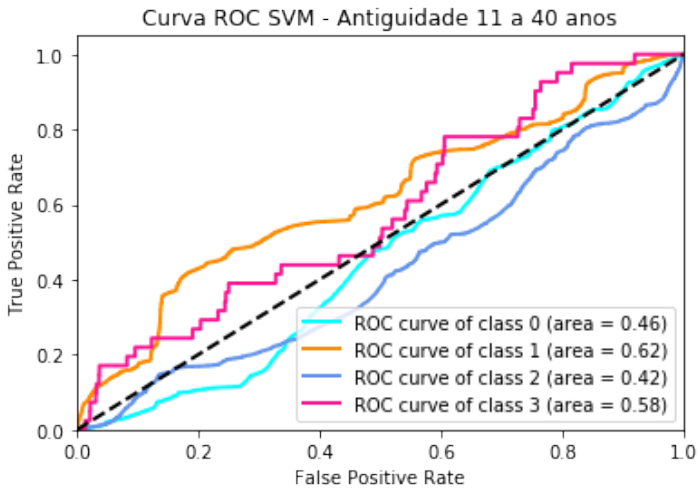
Figuras Auxiliares



(a) Curva ROC RF.

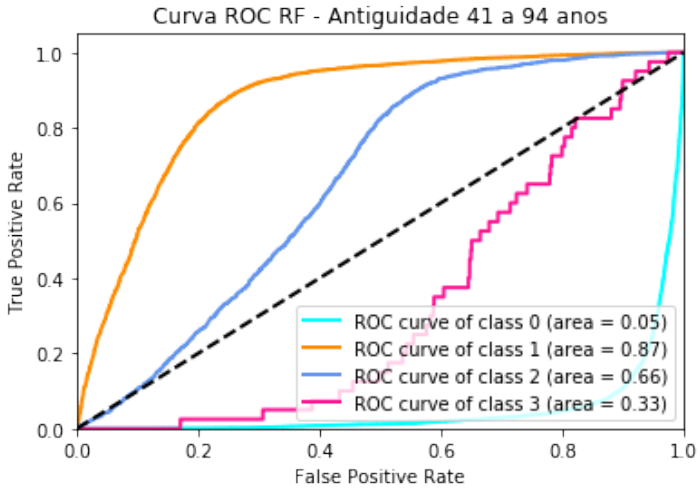


(b) Curva ROC MLP.

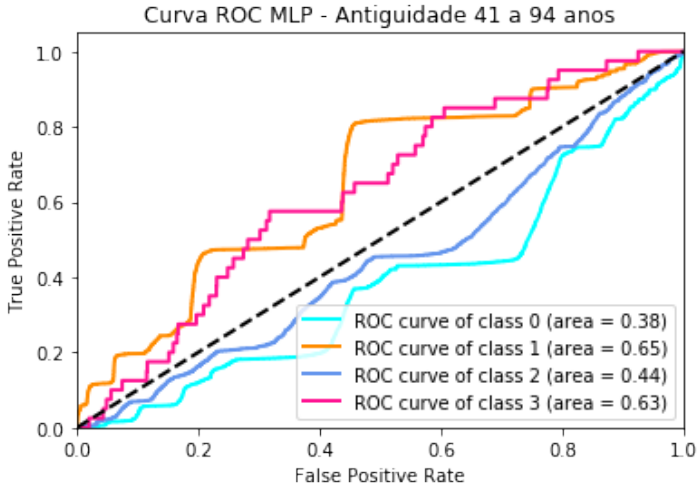


(c) Curva ROC SVM.

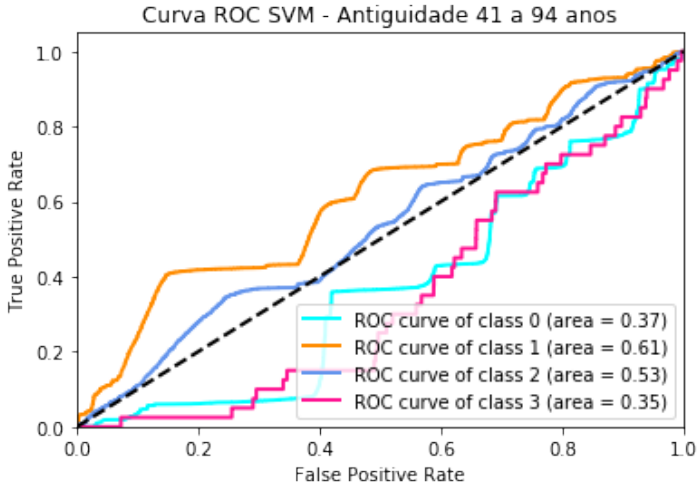
Figura B.1: Curvas ROC para os três algoritmos testados com incidência no subconjunto Antiguidade 11 - 40 anos



(a) Curva ROC RF.



(b) Curva ROC MLP.



(c) Curva ROC SVM.

Figura B.2: Curvas ROC para os três algoritmos testados com incidência no subconjunto Antiguidade 41 - 94 anos

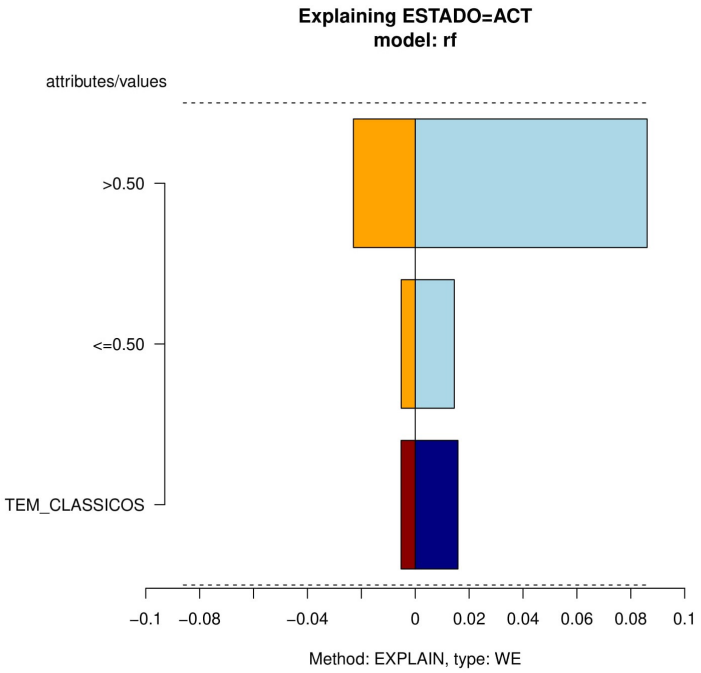


Figura B.3: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_CLASSICOS.

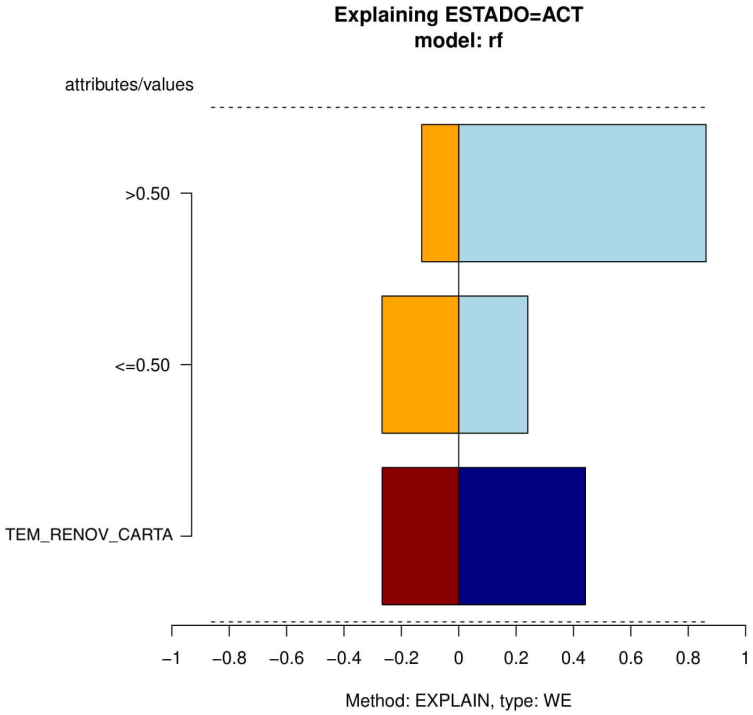


Figura B.4: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_RENOV_CARTA.

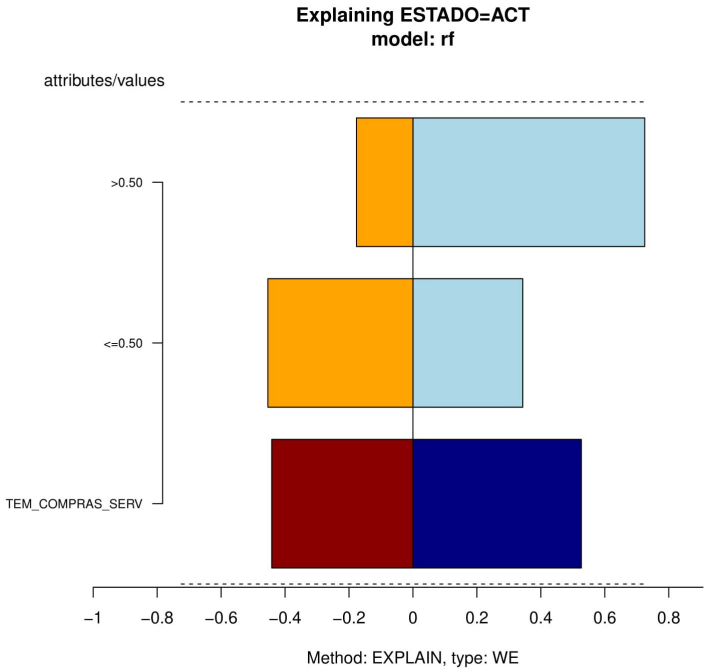


Figura B.5: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_COMPRAS_SERV.

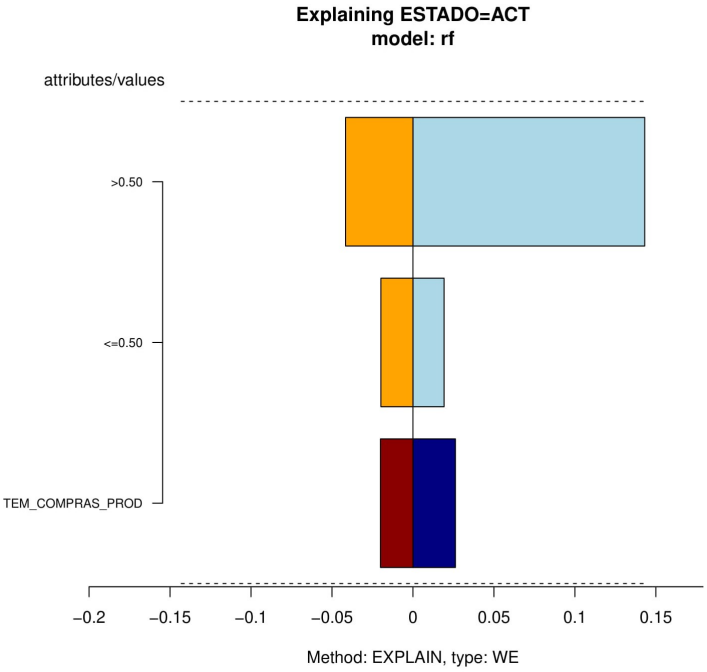


Figura B.6: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_COMPRAS_PROD.

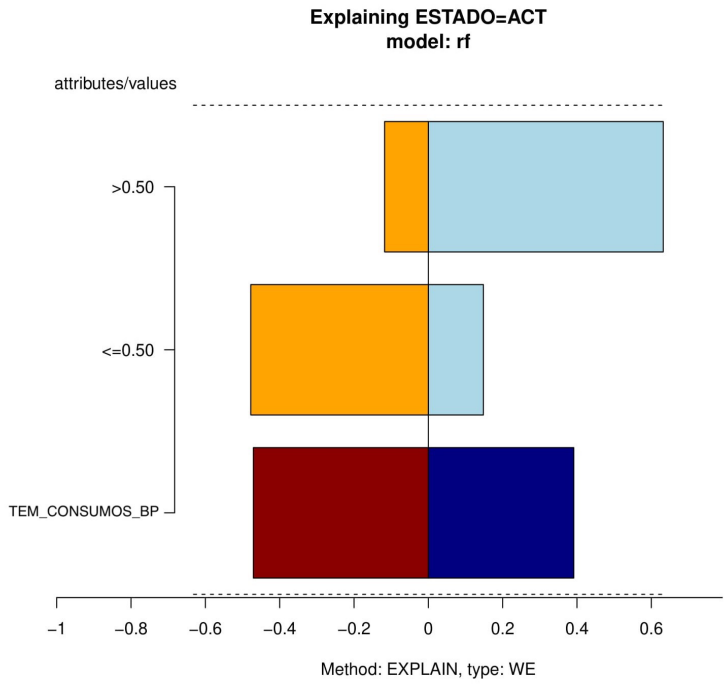


Figura B.7: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_CONSUMOS_BP.

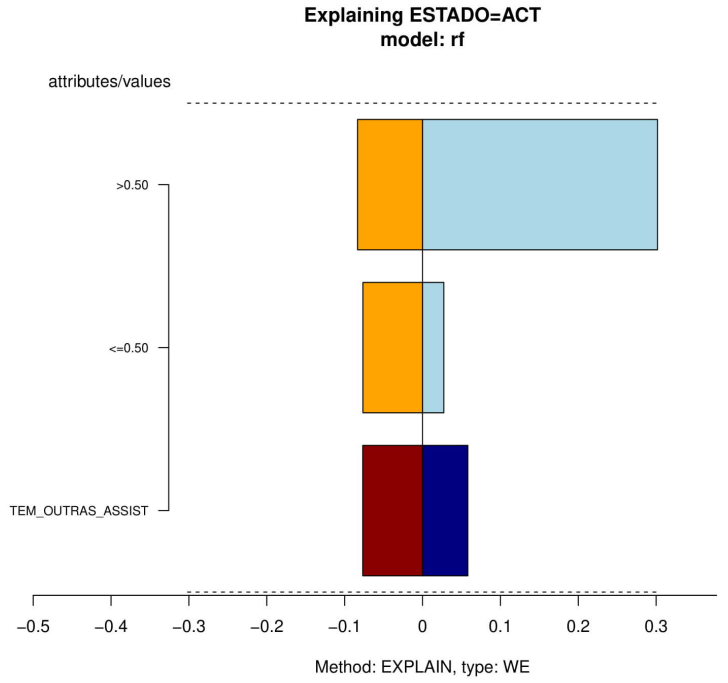


Figura B.8: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_OUTRAS_ASSIST.

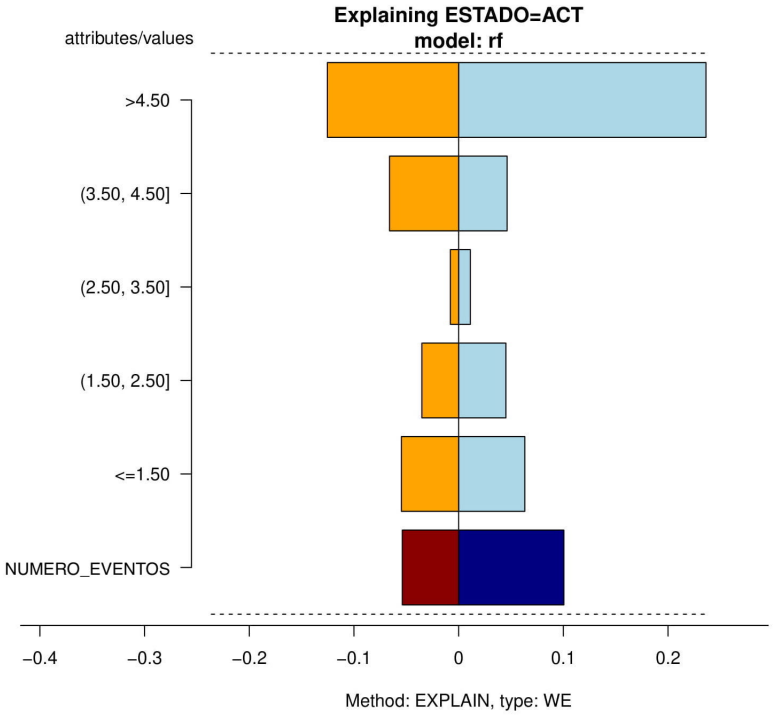


Figura B.9: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo NUMERO_EVENTOS.

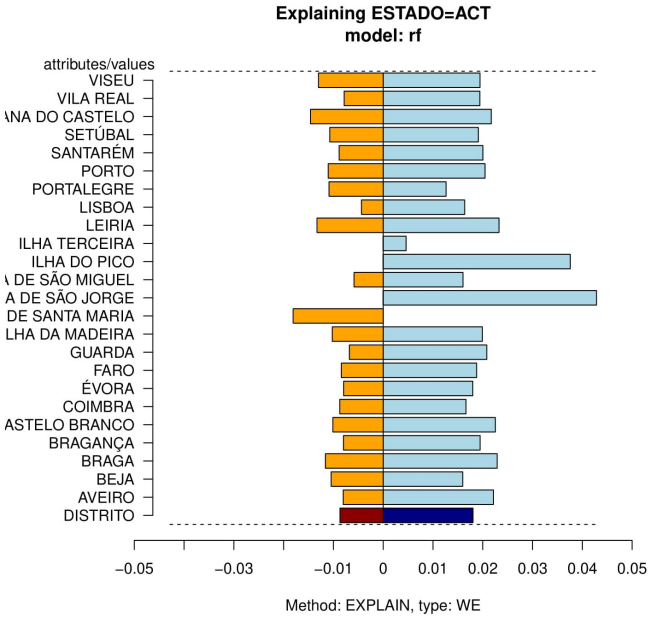


Figura B.10: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo DISTRITO.

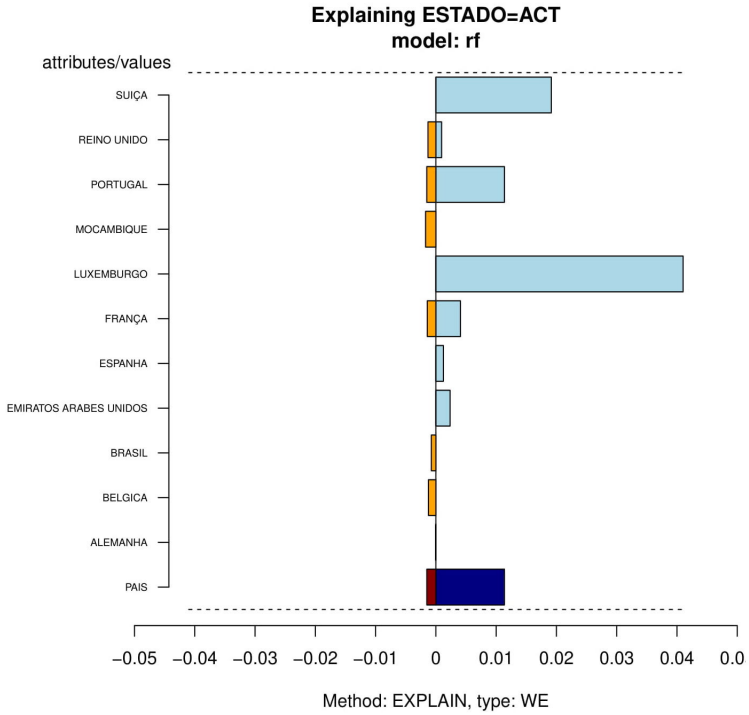


Figura B.11: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo PAIS.

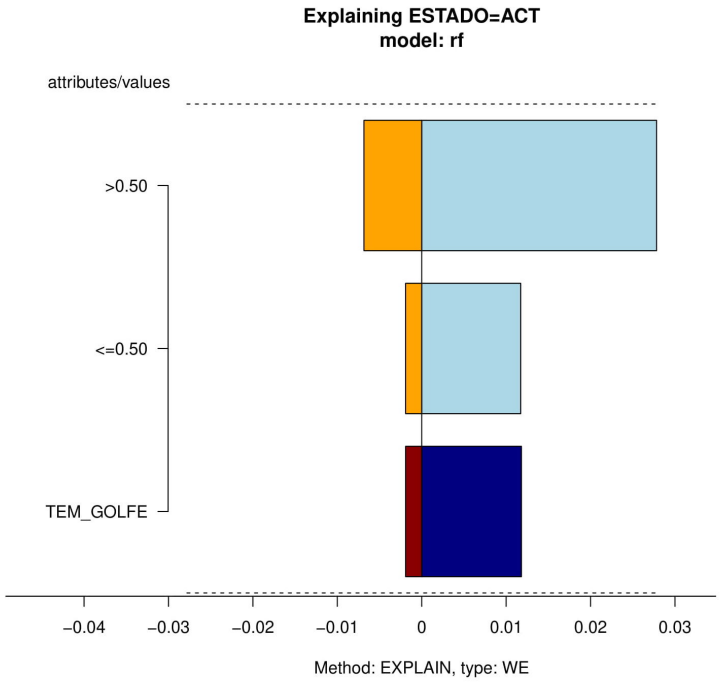


Figura B.12: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo TEM_GOLFE.

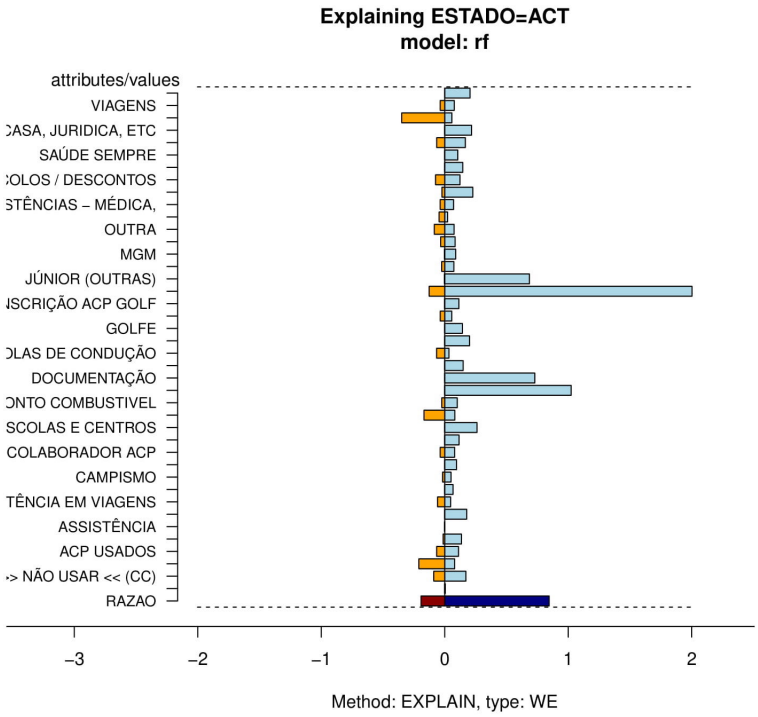


Figura B.13: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo RAZAO.

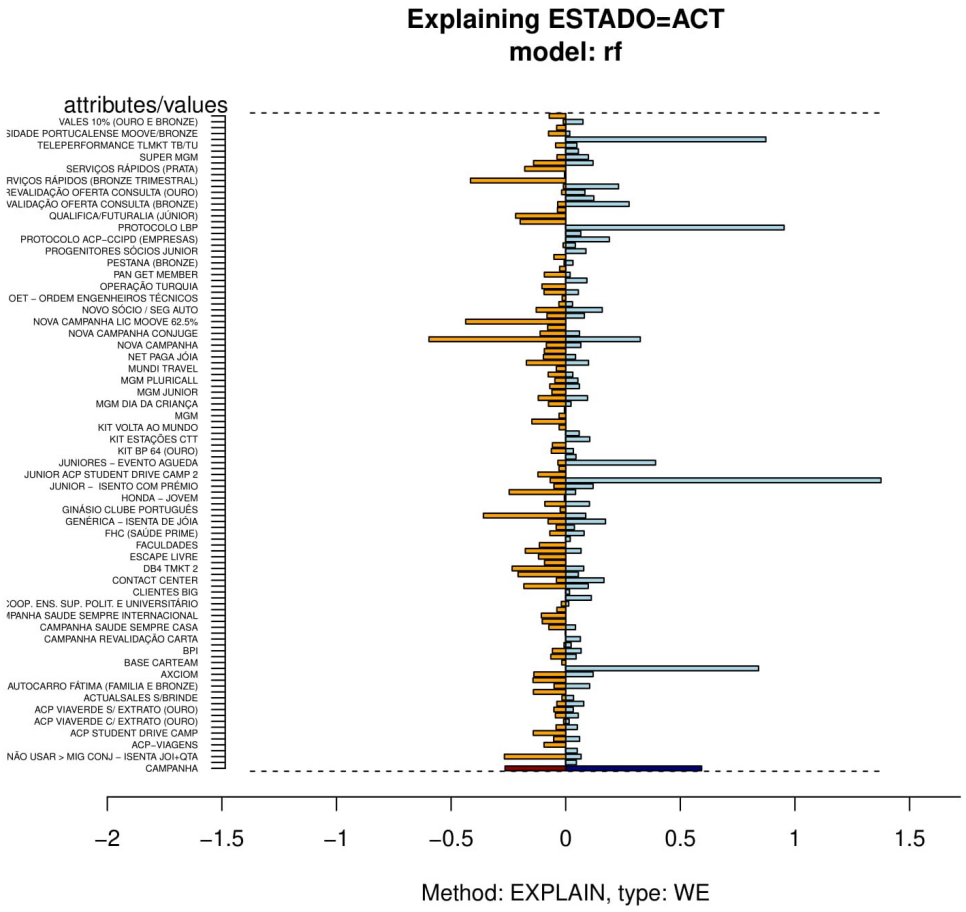


Figura B.14: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo CAMPANHA.

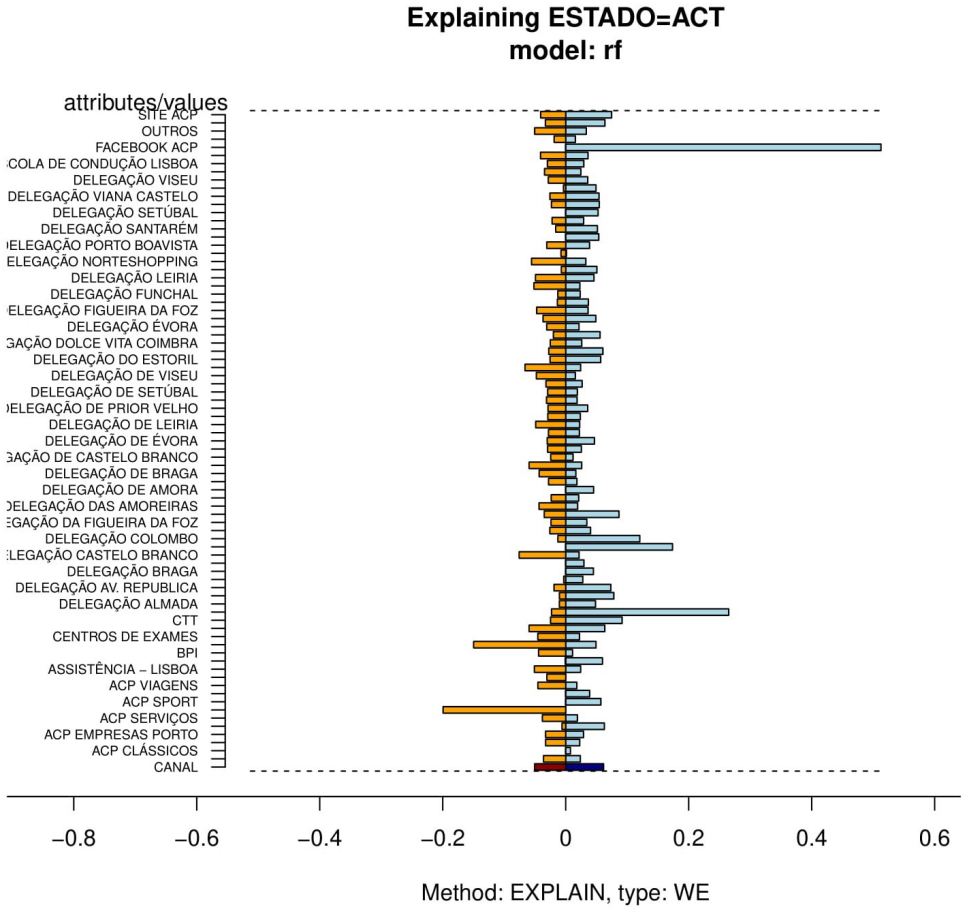


Figura B.15: Explicação para o modelo obtido pelo RF *Classifier* no subconjunto Antiguidade 0-10 para a classe selecionada (Estado = ACT) e para o atributo CANAL.