NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENTO DE MATEMÁTICA

ANDRAWS STEVE SANTOS

Licenciatura em Matemática

MODELOS DE MACHINE LEARNING PARA IDENTIFICAÇÃO DE CLIENTES COM RISCO ACRESCIDO DE BRANQUEAMENTO DE CAPITAIS E FINANCIAMENTO DO TERRORISMO

MESTRADO EM MATEMÁTICA E APLICAÇÕES

Universidade NOVA de Lisboa
Setembro, 2023

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTAMENTO DE MATEMÁTICA

MODELOS DE MACHINE LEARNING PARA IDENTIFICAÇÃO DE CLIENTES COM RISCO ACRESCIDO DE BRANQUEAMENTO DE CAPITAIS E FINANCIAMENTO DO TERRORISMO

ANDRAWS STEVE SANTOS

Licenciatura em Matemática

Orientador: Pedro José dos Santos Palhinhas Mota
Professor, NOVA University Lisbon

Coorientadora: Sofia Medeiros
Consultant, KPMG

Júri

Presidente: Ayana Maria Xavier Furtado Mateus
Assistant Professor, FCT-NOVA

Arguente: Marta Cristina Vieira Faias Mateus
Associate Professor, FCT-NOVA

Vogal: Pedro José dos Santos Palhinhas Mota
Full Professor, FCT-NOVA

MESTRADO EM MATEMÁTICA E APLICAÇÕES
ESPECIALIDADE EM CIÊNCIAS DOS DADOS E DA DECISÃO

Universidade NOVA de Lisboa
Setembro, 2023

Modelos de Machine Learning para identificação de Clientes com risco acrescido de Branqueamento de Capitais e Financiamento do Terrorismo

Copyright © Andraws Steve Santos, Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa.
A Faculdade de Ciências e Tecnologia e a Universidade NOVA de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Este documento foi gerado com o processador (pdf/Xe/Lua)LaTeXe o modelo NOVAthesis (v6.9.4) [18].

Agradecimentos

À minha jornada académica, agradeço a todos os professores que, com dedicação e sabedoria, partilharam o conhecimento que me guiou ao longo do meu percurso. Cada um de vocês desempenhou um papel fundamental na minha formação, e sou imensamente grato por isso.

Em particular, quero expressar a minha sincera gratidão ao meu orientador, o professor Pedro Mota, e à minha coorientadora, Sofia Medeiros. A vossa orientação e apoio foram essenciais para o sucesso deste trabalho e para o meu crescimento académico.

A nível institucional, gostaria de agradecer ao Departamento de Matemática da Faculdade de Ciências e Tecnologias da Universidade Nova de Lisboa por proporcionar o ambiente propício à minha aprendizagem como aluno e pessoa.

Não posso deixar de mencionar a importância da minha mãe, amigos e entes queridos. O vosso apoio inabalável, encorajamento e amor foram a força motriz por trás de cada passo que dei nesta jornada académica. Obrigado por estarem sempre ao meu lado.

Por fim, quero expressar a minha gratidão a Deus por me ter dado fé e clareza de visão nas minhas capacidades e orientação ao longo deste caminho.

A todos vocês, o meu mais sincero agradecimento. Este percurso não teria sido possível sem a vossa presença e apoio constante.

Resumo

A identificação de clientes associados a risco acrescido de Branqueamento de Capitais e Financiamento do Terrorismo (BC/FT) é preocupação primordial para as Instituições Financeiras, em conformidade com a Legislação atual sobre esta temática. Este estudo foca-se no cálculo do Risco BC/FT associado a clientes, um desafio complexo e essencial na atual conjuntura financeira.

O problema em análise é a necessidade de identificar clientes de alto risco, assegurando a conformidade legal e permitindo assim às Instituições Financeiras aplicar as medidas de diligência reforçada sempre que identifique clientes com potencial risco acrescido de BC/FT. Com a crescente ênfase nas abordagens estatísticas, a transição de modelos heurísticos para modelos de classificação em *Machine Learning* (ML) torna-se crucial, uma vez que estes modelos tradicionais não são suportados por um racional matemático, sendo baseados apenas em boas-práticas relacionadas com estas temáticas.

Este estudo propõe uma análise da Avaliação do Risco de BC/FT, explorando a adoção de modelos de classificação em ML que permitem ter em consideração que os clientes particulares e coletivos têm caraterísticas diferentes, e por esse motivo, são aplicados modelos específicos para cada tipologia de cliente.

Nesta dissertação, examinaremos a metodologia, incluindo a exploração de dados estatísticos relacionados com os clientes, o processo de preparação dos dados e uma descrição detalhada dos modelos utilizados. A nossa pesquisa culminou na identificação de modelos de ML altamente precisos na previsão de clientes com risco BC/FT alto com base nos dados considerados.

O uso eficaz dos modelos de classificação em ML contribui para a mitigação do Risco BC/FT fortalecendo a capacidade das instituições financeiras de identificar e monitorizar clientes de alto risco. Este estudo visa fornecer uma abordagem mais precisa e fundamentada na avaliação do risco BC/FT.

Palavras-chave: Machine Learning, Risco BC/FT, Modelos de Classificação, Modelos de Ensemble Learning

Abstract

The identification of customers associated with an increased risk of AML/CFT is a primary concern for Financial Institutions, in compliance with current legislation on this subject. This study focuses on calculating AML/CFT Risk associated with customers, a complex and essential challenge in the current financial landscape.

The problem under analysis is the need to identify high-risk customers, ensuring legal compliance and enabling Financial Institutions to apply enhanced due diligence measures whenever they identify customers with the potential for increased AML/CFT risk. With the growing emphasis on statistical approaches, the transition from heuristic models to Machine Learning (ML) classification models becomes crucial, as these traditional models lack mathematical rationale and are based solely on best practices related to these topics.

This study proposes an analysis of AML/CFT Risk Assessment, exploring the adoption of ML classification models that take into account that individual and corporate customers have different characteristics, and therefore, specific models are applied to each customer type.

In this dissertation, we will examine the methodology, including the exploration of statistical customer data, the data preparation process, and a detailed description of the models used. Our research has resulted in the identification of highly accurate ML models for predicting high AML/CFT risk customers based on the considered data.

The effective use of ML classification models contributes to the mitigation of AML/CFT Risk by strengthening the ability of financial institutions to identify and monitor high-risk customers. This study aims to provide a more precise and evidence-based approach to AML/CFT risk assessment.

Keywords: Machine Learning, AML/CTF Risk, Classification Models, Ensemble Learning Models

Índice

Índice de Figuras

Índice de Tabelas

Siglas

BC/FT Branqueamento de Capitais e Financiamento do Terrorismo vii, 1–3, 5, 6, 8, 16, 63–65, 68, 71, 72

CRISP-DM *Cross-Industry Standard Process for Data Mining* 12, 13

KNN *K-Nearest Neighbors* xi, 28–30, 51, 53, 57–59, 61, 62, 67, 68, 78, 80, 82, 84, 86, 88, 90, 93, 96, 98, 100, 102, 104, 106, 108, 110, 112, 116

KYC *Know Your Client* 1, 2, 17

ML *Machine Learning* vii, xi, xv, 2, 3, 5–19, 21–23, 25, 27–33, 35–41, 43, 45, 48, 49, 65, 68–72

PIB Produto Interno Bruto 1

ROC-AUC *Receiver Operating Characteristic-Area Under the Curve* xv–xvii, 48, 55–64, 67, 78, 84, 91, 98, 111

SVM *Support Vector Machine* xii, 3, 4, 7, 33–35

1

Introdução

A Avaliação do Risco de BC/FT emerge como uma responsabilidade incontornável para as Instituições Financeiras, compelidas a operar em estrito alinhamento com regulamentações de caráter rigoroso. No seio deste contexto regulatório que se aprofunda, a identificação precisa e eficaz do risco associado a estas práticas ilícitas assume uma relevância de primordial importância.

As consequências económicas dos crimes financeiros, como o BC/FT, são assuntos de significativa preocupação global. Estudos indicam que essas infrações podem representar até 5% do Produto Interno Bruto (PIB) [21]. Para enfrentar esses desafios, as autoridades reguladoras têm imposto exigências abrangentes às Instituições Financeiras, investindo quantias substanciais, da ordem dos milhares de milhões de euros, para mitigar o BC/FT. No entanto, apesar dos esforços, revela-se evidente que algumas instituições não aderem de forma estrita às diretrizes e melhores práticas de prevenção do BC/FT, resultando em consequentes multas vultuosas. Apenas em 2016, registaram-se multas de aproximadamente 42 mil milhões de dólares [21], indicando o compromisso das entidades reguladoras em combater este problema com vigor. Concomitantemente, torna-se claro que os criminosos estão a adaptar as suas táticas para contornar o enquadramento legal vigente.

A avaliação de risco BC/FT dos clientes é um processo intrinsecamente ligado a dados e informações pessoais dos clientes. A falta de precisão desses métodos pode levar a classificações incorretas, identificando uma grande percentagem de clientes de baixo risco como de alto risco [4]. A fraca qualidade dos dados, juntamente com a necessidade de atualizações frequentes, adiciona complexidade ao processo. A limpeza, transformação e atualização dos dados são cruciais para o pré-processamento de dados não padronizados. Além disso, os padrões de comportamento relacionados ao BC/FT podem variar entre diferentes países e empresas, o que complica ainda mais a detecção precisa, permitindo que os criminosos evitem ser detectados. Portanto, a compreensão detalhada do contexto de *Know Your Client* (KYC) é essencial para garantir a eficácia dos métodos de avaliação

de risco e aprimorar a detecção de atividades suspeitas.

No âmbito das instituições financeiras, ferramentas de monitorização de Clientes e Transações desempenham um papel crucial na identificação de situações de risco acrescido de BC/FT. Isso requer a classificação dos clientes de acordo com o nível de risco associado. No entanto, a precisão dessa classificação é vital, uma vez que uma avaliação subestimada pode expor as instituições a violações regulatórias significativas e riscos financeiros, alocando recursos financeiros substanciais na revisão de um grande número de falsos positivos.

Além disso, à medida que as autoridades reguladoras reforçam as suas capacidades de identificar transações suspeitas, o conceito de KYC ganha relevância, incentivando uma compreensão profunda das informações dos clientes a fim de melhor avaliar e mitigar riscos associados. A adoção de modelos de classificação robustos, em contraste com abordagens heurísticas, pode desempenhar um papel crucial nesse processo, uma vez que esses modelos se baseiam em algoritmos de ML capazes de analisar grandes volumes de dados e identificar padrões complexos. A utilização de modelos de classificação robustos pode reduzir os erros de classificação em até 50% [21], contribuindo para uma avaliação mais precisa e eficaz dos riscos associados ao BC/FT.

O propósito subjacente deste projeto consiste em desenvolver uma ferramenta de avaliação de risco de alta precisão, fundamentada na aplicação de métodos de ML e na análise de dados. O objetivo primordial é mitigar as dificuldades anteriormente delineadas por meio de abordagens robustas. Posteriormente, tal classificação será seguida por procedimentos de diligência reforçada para clientes de risco alto, o que implica que o seu comportamento é monitorizado de forma mais rigorosa e atenta. No entanto, é importante destacar que essa monitorização mais intensiva não implica automaticamente em relatórios às autoridades; tais relatórios resultam da análise contínua das transações.

2

Revisão da Literatura

Nesta secção, será realizada uma breve, mas fundamental, revisão da literatura existente na nossa área científica, focada no combate ao Branqueamento de Capitais e Financiamento do Terrorismo (BC/FT). Esta fase do projeto desempenha um papel crucial, uma vez que é aqui que identificaremos lacunas e áreas ainda pouco exploradas no campo do BC/FT. A partir destas lacunas, desenvolveremos a nossa estratégia de pesquisa para as etapas subsequentes deste trabalho.

O BC/FT é um domínio que, apesar da sua crescente importância, ainda carece de um corpo substancial de pesquisa. Assim, a literatura específica sobre este tema permanece relativamente escassa. Como resposta a esta carência, iremos também explorar literatura relacionada a tópicos paralelos, mas igualmente pertinentes, como o risco de crédito [2, 15, 5, 26, 7, 16, 8]. Estes temas, embora distintos, partilham o uso comum de modelos de *Machine Learning* (ML), o que pode proporcionar valiosas perspetivas estratégicas à nossa pesquisa no contexto do BC/FT.

2.1 Branqueamento de capitais

Uma área de grande interesse nestes estudos é a classificação de clientes, baseada no risco associado ao BC/FT. Diversos métodos têm sido aplicados com o objetivo de avaliar e classificar o nível de risco de clientes em instituições financeiras. Entre os métodos mais amplamente utilizados nestes estudos, destacam-se o *Support Vector Machine* (SVM) [14, 27, 19], árvores de decisão [3, 17, 29], redes neuronais [19, 3].

No artigo [14], apresentado na 5ª Conferência Internacional sobre Gestão de Comércio Eletrónico e *e-Government* em 2011 por L. Keyan e Y. Tingting, explora uma abordagem otimizada para detetar atividades suspeitas de branqueamento de capitais utilizando as SVM. Neste estudo, os autores abordam o desafio de detetar atividades suspeitas de branqueamento de capitais no contexto financeiro. Para alcançar esse objetivo, propõem uma abordagem baseada em SVM, que é uma técnica de ML. O foco central deste trabalho é

melhorar o desempenho do modelo SVM na identificação de transações financeiras potencialmente relacionadas com o branqueamento de capitais. O processo de investigação envolve várias etapas cruciais:

- **Geração de Dados Simulados:** Os autores iniciaram o estudo utilizando dados simulados, gerados com base em informações obtidas do Banco Agrícola da China. Esses dados simulados foram concebidos para representar transações financeiras que poderiam estar associadas ao branqueamento de capitais. A utilização de dados simulados permitiu criar um ambiente controlado para testar o modelo SVM.

- **Treino e Avaliação do Modelo SVM:** Com os dados simulados, os autores procederam ao treino do modelo SVM. Isso envolveu expor o modelo a exemplos rotulados de transações financeiras, permitindo-lhe aprender a distinguir entre transações legítimas e transações suspeitas de branqueamento de capitais.

- **Aprimoramento do Modelo:** A contribuição principal deste estudo reside na otimização do modelo SVM. Embora os autores não tenham detalhado especificamente os métodos de otimização, afirmam que essa melhoria conduziu a resultados mais precisos na deteção de atividades suspeitas.

- **Resultados Experimentais:** O desempenho do modelo SVM otimizado foi avaliado por meio de experiências ao utilizar dados simulados. Os resultados experimentais demonstraram a capacidade do modelo em identificar corretamente transações suspeitas de branqueamento de capitais. Esses resultados foram comparados com abordagens anteriores para determinar a eficácia do modelo otimizado.

- **Conclusões e Implicações:** Com base nos resultados experimentais, os autores concluíram que o modelo SVM otimizado superou as abordagens anteriores em termos de precisão na deteção de atividades suspeitas de branqueamento de capitais. Isso sugere que essa abordagem aprimorada pode ser promissora para instituições financeiras que procuram identificar eficazmente atividades ilícitas relacionadas com o branqueamento de capitais.

Os artigos [19] e [27] tratam da mesma questão, investigando o uso de SVM, e todos eles e utilizam dados simulados originados de instituições bancárias chinesas.

O artigo [19] apresenta ainda uma abordagem baseada em redes neuronais. O estudo foi conduzido por Lin-Tao Lv, Na Ji e Jiu-Long Zhang e foi publicado na *International Conference on Wavelet Analysis and Pattern Recognition* de 2008.

A principal motivação deste estudo é desenvolver um modelo eficaz para identificar transações financeiras que possam estar relacionadas com o branqueamento de capitais. Os autores destacam a importância dessa deteção, dada a crescente preocupação global com a prevenção do BC/FT.

A abordagem proposta neste artigo baseia-se em redes neuronais de função de base radial, que são conhecidas pela sua capacidade de aprendizagem em dados não lineares. Os autores aplicaram esta técnica a um conjunto de dados simulados, gerados a partir de um banco chinês, que consistia em transações financeiras. O objetivo era classificar essas transações como suspeitas ou não suspeitas de branqueamento de capitais.

O artigo descreve em detalhes a metodologia utilizada, incluindo a preparação dos dados, o treino da rede neuronal de função Radial e a avaliação do desempenho do modelo. Os resultados obtidos mostraram que o modelo de função Radial teve um desempenho promissor na deteção de atividades suspeitas de branqueamento de capitais, com uma taxa de precisão considerável. Isso sugere que esta abordagem pode ser valiosa para instituições financeiras que buscam identificar eficazmente transações financeiras relacionadas ao branqueamento de capitais.

O artigo [3] também se foca nas redes neuronais para resolver problemas de faturações fraudulentas, com dados de uma instituição financeira norte americana.

Em [29], que foi apresentado por Su-Nan Wang e Jian-Gang Yang na *International Conference on Machine Learning and Cybernetics* de 2007, concentra-se em desenvolver um método de avaliação de risco de branqueamento de capitais tendo o seu foco nas árvores de decisão.

A motivação por trás deste artigo é a crescente preocupação com a deteção e prevenção de atividades suspeitas de branqueamento de capitais no setor financeiro. Identificar transações financeiras que possam estar relacionadas com o branqueamento de capitais é crucial para evitar que fundos ilícitos sejam introduzidos no sistema financeiro.

A abordagem proposta pelos autores envolve o uso de árvores de decisão, que são uma técnica de ML amplamente aplicada na classificação de dados. Os dados utilizados foram simulados, baseados em transações financeiras de bancos chineses.

Os autores descrevem a metodologia usada para treinar e avaliar o modelo de árvore de decisão em relação ao risco de branqueamento de capitais. O artigo explora como o modelo de árvore de decisão pode ser aplicado para classificar transações financeiras como de alto ou baixo risco de branqueamento de capitais. Os resultados obtidos indicam que as árvores de decisão conseguiram identificar padrões eficazmente e alcançaram resultados satisfatórios na deteção de atividades suspeitas de branqueamento de capitais.

Os artigos [3] e [17] tratam da mesma questão, ao promover o uso de Árvores de Decisões. Em [17], utilizam dados gerados a partir de um banco Sueco e em [3] utilizam

dados gerados a partir de uma instituição financeira norte americana.

O artigo [24] intitulado "*Addressing AML Regulatory Pressures by Creating Customer Risk Rating Models with Ordinal Logistic Regression*", escrito por Edwin Rivera, Jimmie L. West e Carl Suplee em 2015, aborda as pressões regulatórias no campo do combate ao branqueamento de capitais. O foco deste estudo está na criação de modelos de classificação de risco de clientes em instituições financeiras.

Este estudo descreve a diferença entre modelos heurísticos baseados em regras e modelos baseados em estatísticas na criação de modelos de classificação de risco de clientes em instituições financeiras. Ambos os tipos de modelos consideram variáveis como tipo de relacionamento do cliente, geografia, características da conta, clientes de alto risco, histórico de alertas e registos e atividade transacional esperada.

Descrevem que, modelos heurísticos são fórmulas analíticas simples usadas para atribuir uma pontuação com base em variáveis que a instituição considera importantes. Esses modelos são frequentemente criados a partir de todas as variáveis disponíveis, pois a importância relativa de cada variável individual geralmente é desconhecida. Cada cliente recebe uma pontuação com base nas variáveis e, em seguida, as pontuações são agregadas para atribuir uma categoria de risco ao cliente. No entanto, esses modelos não seguem uma metodologia estatística específica, tornando difícil a justificação perante Reguladores.

Por outro lado, modelos que se baseados em fundamentos estatísticos usam uma abordagem mais científica na criação de modelos de risco de clientes. Estes modelos seguem uma estrutura estatística específica e permitem um ajuste sistemático de parâmetros e validação do modelo. Essa abordagem é preferida atualmente devido à sua capacidade de resistir ao escrutínio regulatório e fornecer um método mais rigoroso para o ajuste de parâmetros e validação de modelos.

A literatura existente sugere que o uso de modelos de ML se tem mostrado promissor na identificação de transações financeiras suspeitas e na avaliação de risco associado ao BC/FT em contraste com modelos heurísticos. No entanto, vale ressalvar que muitos desses estudos se concentram em contextos específicos, como instituições financeiras nos Estados Unidos ou na China. Portanto, há espaço para investigar como esses modelos podem ser aplicados em diferentes cenários e como o contexto geográfico, como o país de residência dos clientes, pode influenciar os resultados. É relevante notar que a escolha destes métodos de classificação pode depender das características específicas dos dados e do contexto da instituição financeira.

Na literatura, é prevalente o uso recorrente dos mesmos modelos de classificação, resultando em abordagens e resultados notavelmente similares em diversos artigos, sendo as únicas variações os conjuntos de dados utilizados.

2.2 Risco de crédito

Na área de pesquisa sobre risco de crédito , é interessante observar que, embora ambos compartilhem um objetivo fundamental relacionado à avaliação de riscos financeiros, as suas abordagens apresentam nuances distintas. Enquanto que, o branqueamento de capitais concentra-se na detecção de atividades financeiras ilícitas e lavagem de dinheiro, o risco de crédito lida com a avaliação da probabilidade de inadimplência de empréstimos ou financiamentos.

Uma tendência notável na literatura é o amplo uso de modelos de ML. Esses modelos incluem a regressão logística [2], SVM [15, 5, 7], técnicas de boosting e bagging [26, 8], árvores de decisão e florestas aleatórias [2, 15, 16]. Esta diversidade de abordagens sugere a busca por métodos eficazes de previsão e classificação, que podem ser adaptados às características específicas dos conjuntos de dados em cada contexto.

Outro ponto relevante prende-se com a discrepância nos dados e contextos utilizados em ambas as áreas. Enquanto as investigações sobre branqueamento de capitais frequentemente se concentram em bancos norte-americanos e chineses, neste contexto temos mais estudos relacionados a países europeus, como Alemanha, Itália, França, assim como Austrália e Japão. Isso pode ser atribuído às diferentes dinâmicas financeiras e regulamentações em vigor em cada região, o que impacta a natureza dos dados disponíveis. Este facto sublinha a necessidade de uma abordagem regionalizada na análise do branqueamento de capitais, considerando as especificidades inerentes a cada jurisdição.

A variação nos modelos e a escolha de diferentes conjuntos de dados podem ser explicadas pela busca constante por abordagens que se adaptem melhor às características específicas de cada problema e região. Em última análise, esta diversidade na pesquisa demonstra a flexibilidade e a capacidade de adaptação dos investigadores para enfrentar desafios complexos nas suas respectivas áreas de estudo.

Conclusões

Nesta análise abrangente da literatura disponível, fica claro que o foco predominante está no combate a uma ampla gama de crimes financeiros, incluindo o branqueamento de capitais e os riscos de crédito. Isso ocorre em resposta à crescente pressão regulatória exercida pelas autoridades, que têm aplicado multas substanciais às instituições financeiras que não cumprem rigorosamente esses requisitos.

Para enfrentar esses desafios, as instituições financeiras estão a adotar cada vez mais modelos de classificação com suporte estatístico, deixando em segundo plano os modelos heurísticos. Esta revisão da literatura revela que estes modelos de ML têm o potencial de melhorar significativamente as previsões, reduzindo, assim, os incidentes de crimes financeiros.

No entanto, a diversidade de modelos disponíveis e a variação nos conjuntos de dados utilizados por diferentes autores indicam que o contexto geográfico pode ter uma influência significativa no desempenho desses modelos.

Além disso, a utilização de dados simulados apresenta desafios, uma vez que nem todas as instituições têm acesso a dados reais. Isso pode afetar o desempenho e a generalização dos modelos de ML.

Portanto, nesta dissertação iremos focar em:

- Demonstrar que o uso de modelos de ML confere resultados sólidos na classificação de risco de BC/FT.
- Realizar uma análise abrangente comparando o desempenho de diversos modelos de ML, com o intuito de identificar discrepâncias nas suas *performances*.
- Investigar quais as variáveis mais relevantes e que exercem mais influência nos resultados e comportamento dos modelos de ML.

Esta pesquisa busca contribuir para a compreensão das melhores práticas na implementação de modelos de ML no setor financeiro, considerando os desafios específicos relacionados à escolha do modelo, aos dados disponíveis e ao contexto geográfico.

3

Metodologia

Neste capítulo, apresentaremos em detalhe a metodologia que orientará esta dissertação.

3.1 Linguagem de Programação

Neste subcapítulo, detalharemos a implementação dos modelos de ML utilizando a linguagem de programação *Python*. Abordaremos as razões por trás da escolha dessa linguagem, as bibliotecas e ferramentas específicas utilizadas, bem como os detalhes de como os modelos foram implementados e treinados.

3.1.1 Razões para Escolher *Python*

Exploraremos as justificações que sustentam a decisão de optar pela linguagem de programação *Python* como a principal ferramenta para a implementação dos modelos de ML neste projeto.

Esta escolha foi orientada por diversas razões que se unem para fazer do *Python* uma escolha vantajosa para esta tarefa:

Em primeiro lugar, o *Python* é amplamente adotado na comunidade de ciência de dados e ML. Esta popularidade resulta numa vasta gama de bibliotecas, recursos e tutoriais disponíveis, facilitando a implementação de algoritmos e a superação de desafios específicos.

A sintaxe do *Python* é conhecida pela sua clareza e legibilidade. Esta característica torna a linguagem adequada para prototipagem rápida e promove a colaboração em projetos de equipa, facilitando a compreensão do código por diversos profissionais.

O ecossistema do *Python* oferece uma variedade de bibliotecas e *frameworks* orientados para a ciência de dados e ML, como *NumPy*, *Pandas*, *Scikit-learn*, *TensorFlow* e *Keras*. Estas

ferramentas agilizam o desenvolvimento e a implementação de modelos, permitindo um progresso eficiente.

A flexibilidade e versatilidade do *Python* são traços fundamentais. A linguagem pode ser utilizada em diferentes fases do processo de análise de dados, abrangendo desde a manipulação dos dados até à visualização e apresentação dos resultados finais.

A comunidade robusta do *Python* contribui para um suporte substancial. Questões são rapidamente respondidas, promovendo um ambiente de aprendizagem e resolução de problemas ágil e eficaz ao longo de todo o desenvolvimento do projeto.

Além disso, a capacidade de integração do *Python* com outras tecnologias e linguagens amplia o seu potencial. Isso permite criar fluxos de trabalho personalizados e combinar de forma estratégica diversas ferramentas.

Outro ponto a considerar é a acessibilidade do *Python*, que é uma linguagem de código *Open Source* e gratuita. Esta característica torna o *Python* acessível a uma ampla variedade de programadores e organizações, proporcionando igualdade de oportunidades.

Em resumo, a escolha do *Python* para a implementação das técnicas de ML neste projeto foi baseada na sua ampla adoção na comunidade de ciência de dados, na sua sintaxe clara e legível, na disponibilidade de bibliotecas e *frameworks* robustos, na flexibilidade e versatilidade, no suporte dinâmico da comunidade, na integração fluida com outras tecnologias e na acessibilidade. Estas razões combinadas destacam o *Python* como uma escolha sólida e bem fundamentada para atingir os objetivos deste estudo.

3.1.2 Ambiente Interativo

A relevância do ambiente interativo do *Jupyter Notebook* assume um papel de destaque no desenvolvimento e exploração dos modelos de ML. A sua natureza interativa permite a combinação harmoniosa de código, documentação e visualização dos resultados num único ambiente. Esta abordagem integrada facilita a experimentação iterativa, permitindo que cientistas de dados e investigadores explorem diferentes abordagens, ajustem parâmetros e observem imediatamente os resultados.

Uma das vantagens distintas do *Jupyter Notebook* reside na capacidade de integrar blocos de código *Python* com texto explicativo, gráficos e tabelas. Esta integração não só torna a documentação dos passos de implementação mais clara e compreensível, mas também possibilita uma análise detalhada dos resultados obtidos. A visualização de gráficos e tabelas diretamente no *Notebook* agiliza a interpretação dos dados, facilitando a

identificação de padrões, tendências e anomalias.

Para além disso, o *Jupyter Notebook* é uma ferramenta altamente colaborativa, simplificando a partilha de código, análises e resultados com colegas de equipa ou com a comunidade em geral. A sua natureza interativa e a capacidade de criar relatórios completos no âmbito do *Notebook* facilitam a comunicação de resultados e insights.

Em resumo, o ambiente interativo do *Jupyter Notebook* desempenha um papel crucial na experimentação e visualização dos resultados dos modelos de ML. A conjugação única de código, documentação e visualização contribui para um fluxo de trabalho mais eficiente, permitindo uma exploração detalhada dos dados e a tomada de decisões informadas na implementação dos modelos.

3.1.3 Bibliotecas e *Frameworks* Utilizados

Neste subcapítulo, iremos explorar as principais bibliotecas e *frameworks* utilizados na implementação dos modelos de ML que foram escolhidos para este projeto. Destacaremos o papel crucial que essas ferramentas desempenham na construção, treino e avaliação dos modelos, fornecendo uma base sólida para a aplicação das técnicas de ML selecionadas..

Para atingir os objetivos deste projeto, optamos por utilizar um conjunto de bibliotecas e *frameworks* amplamente reconhecidos e utilizados na comunidade de ciência de dados e ML. As seguintes bibliotecas e *frameworks* foram escolhidos devido às suas capacidades, eficácia e facilidade de uso:

- *NumPy*: *NumPy* [11] é uma biblioteca fundamental para computação numérica em *Python*. Oferece suporte para *arrays* multidimensionais e funções matemáticas avançadas, o que é essencial para manipulação e processamento de dados.
- *Pandas*: O *Pandas* [20] é uma biblioteca que fornece estruturas de dados de alto desempenho para análise de dados, incluindo séries temporais, *DataFrames* e ferramentas para manipulação de dados.
- *Scikit-learn*: Também conhecido como *sklearn*, o *Scikit-learn* [23] é uma biblioteca de ML *Open Source* que oferece uma ampla gama de algoritmos de classificação, regressão, clusterização, entre outros.
- *Matplotlib* e *Seaborn*: *Matplotlib* [13] é uma biblioteca para criação de visualizações gráficas em *Python*, enquanto *Seaborn* [30] é uma biblioteca baseada em *Matplotlib* que fornece uma interface de alto nível para criar gráficos estatísticos atraentes e informativos.

Estas bibliotecas e *frameworks* oferecem uma base sólida para a implementação das técnicas de ML propostas. A escolha destas ferramentas foi guiada pela popularidade,

versatilidade, documentação abrangente e suporte ativo da comunidade. A combinação destes recursos auxiliará na execução eficaz das etapas de preparação, treino e avaliação dos modelos, além de possibilitar a criação de visualizações claras e elucidativas para a interpretação dos resultados obtidos.

3.2 CRISP-DM

O processo metodológico *Cross-Industry Standard Process for Data Mining* (CRISP-DM) [32], amplamente adotado pela comunidade científica, oferece uma estrutura bem definida para guiar a execução de projetos de mineração de dados de forma estruturada.

O CRISP-DM é composto por seis fases inter-relacionadas, cada qual contribuindo para uma compreensão mais profunda dos dados e apropriação adequada para a modelagem. As etapas englobam:

1. Compreensão do Contexto de Negócios e dos Dados: Nessa fase, busca-se uma apreensão integral dos objetivos do projeto e empreende-se a exploração da natureza intrínseca dos dados disponíveis. Por meio de análises preliminares, identificam-se variáveis pertinentes para o problema em causa.

2. Compreensão dos Dados: Aprofunda-se a exploração mediante a Análise Exploratória de Dados, que envolve a aplicação de técnicas estatísticas e de visualização para explorar possíveis padrões, tendências e eventuais problemáticas nos dados sob análise.

3. Preparação dos Dados: Nesta etapa, procede-se à seleção das variáveis mais pertinentes, depuração de dados inconsistentes ou incompletos e transformação das variáveis, visando uma melhor adequação aos modelos de ML.

4. Modelação: O foco principal recai sobre a criação dos Modelos de Classificação e *Ensemble Learning*.

5. Avaliação: Nesta fase, procede-se à avaliação dos resultados provenientes dos modelos aplicados aos dados preparados. Isso contribui para uma compreensão mais precisa da eficácia das técnicas de ML em relação ao problema em questão.

6. Implementação: Embora esta fase esteja mais associada à implementação de soluções em contextos corporativos, o foco principal desta dissertação está direcionado principalmente para as etapas anteriores do processo.

Ao adotar o CRISP-DM, é estabelecida uma abordagem metodológica sistemática e estruturada na preparação dos dados, o que, por sua vez, fortalece a solidez e a eficácia dos modelos de classificação e *ensemble* que serão posteriormente aplicados.

É importante ressalvar que, dada a natureza flexível do CRISP-DM e seu potencial de adaptação a diferentes cenários empresariais, os detalhes específicos de implementação de cada etapa serão discutidos de maneira mais aprofundada em capítulos subsequentes. A aplicação do CRISP-DM será moldada e contextualizada de acordo com as necessidades e objetivos específicos desta pesquisa, garantindo assim uma abordagem direcionada e eficiente para a análise de dados.

3.3 Descrição da Base de Dados

Neste subcapítulo, aprofundamos a nossa compreensão da base de dados que servirá como a espinha dorsal para a nossa análise de ML. É fundamental conhecer a estrutura e a natureza dos dados com os quais estamos a trabalhar antes de aplicar qualquer técnica ou modelo. Vamos explorar detalhadamente os conjuntos de dados *"account_info"* e *"customer_info"*, examinando as suas variáveis e características. Esta análise detalhada proporcionará um contexto valioso para a preparação, pré-processamento e modelagem dos dados, tópicos que serão abordados nas secções subsequentes deste capítulo.

3.3.1 Origem dos Dados

A origem das bases de dados utilizadas neste estudo está na criação de dados sintéticos baseados na tipologia de informação de uma Instituição Financeira localizada em Portugal, abrangendo clientes particulares e coletivos. Esses dados sintéticos foram gerados através de um processo cuidadoso de síntese que mapeou e extrapolou informações essenciais dos clientes e contas do banco. Esse processo de criação de dados sintéticos foi elaborado para preservar a privacidade e a confidencialidade dos dados originais, não destoando o realismo do conjunto de dados para a nossa análise de ML.

3.3.2 Base de Dados

Para a realização deste estudo, tivemos a oportunidade de aceder a três bases de dados recentes, cada uma contendo informações valiosas e detalhadas. Uma base de dados abrange informações relacionadas com as contas dos clientes, outra fornece *insights* abrangentes sobre os próprios clientes, e há ainda uma terceira, denominada *risk_score*, que desempenha um papel crucial na avaliação de riscos. Essas bases de dados constituem a espinha dorsal da nossa análise de ML e são fundamentais para a compreensão do nosso trabalho.

Base de Dados *account_info*

A primeira base de dados, intitulada *account_info*, inclui uma variedade de informações relacionadas às contas dos clientes. Esta base de dados revela detalhes importantes, como os titulares das contas, os tipos de produtos associados e informações exclusivas de

cada conta. Com um total de 162942 registos e 6 variáveis, o conjunto de dados *account_info* fornece informações detalhadas sobre as contas dos clientes, essenciais para a nossa análise de ML.

Aqui estão as variáveis presentes neste conjunto de dados:

- ***account_id***: Variável categórica de identificador único da conta.
- ***first_holder***: Variável categórica de identificador único do primeiro titular da conta.
- ***product_source_type_code***: Variável categórica que contém o código que indica a fonte ou tipo de produto associado à conta.
- ***customer_id***: Variável categórica de identificador único do cliente vinculado a essa conta.
- ***customer_role***: Variável categórica que descreve a função ou papel do cliente em relação a essa conta.
- ***bef_flag***: Variável numérica binária que indica algum estado de risco anterior à abertura da conta.

Base de Dados *customer_info*

A segunda base de dados, chamada *customer_info*, fornece informações detalhadas sobre os clientes em si. Este conjunto de dados abrange o perfil demográfico dos clientes, o seu histórico e o seu estatuto em relação às atividades financeiras. Com um total de 40000 registos e 19 variáveis, o conjunto de dados *customer_info* oferece informações cruciais para compreender os clientes em profundidade.

Aqui estão as variáveis presentes neste conjunto de dados:

- ***customer_id***: Variável categórica ou categórica de identificador único da conta.
- ***customer_type_code***: Variável categórica que indica o tipo de cliente.
- ***nationality_code***: Variável categórica que representa o código da nacionalidade do cliente.
- ***country_of_residence***: Variável categórica que indica o país de residência do cliente.
- ***country_of_origin***: Variável categórica que indica o país de origem do cliente.
- ***incorporation_date***: Variável de data que indica a data de incorporação da empresa do cliente.
- ***business_type***: Variável categórica que descreve o tipo de negócio do cliente.

- ***occupation***: Variável categórica que indica a ocupação do cliente.
- ***channel***: Variável categórica que representa o canal de comunicação preferido pelo cliente.
- ***date_of_birth***: Variável de data que indica a data de nascimento do cliente.
- ***sa_proposito_rela***: Variável categórica que descreve o propósito, razão ou objetivo da relação do banco com o cliente.
- ***pep_flag***: Variável binária que indica se o cliente é uma Pessoa Politicamente Exposta (PEP)
- ***segmento***: Variável categórica que indica o segmento ao qual o cliente pertence.
- ***tocpp***: Variável categórica que descreve se o cliente ocupa ou ocupou outros cargos políticos ou públicos.
- ***formjur***: Variável categórica que descreve a forma jurídica do cliente.
- ***assigned_risk_id***: Variável numérica (10, 30 ou 50) que representa a identificação de risco atribuída ao cliente
- ***manual_risk_flag***: Variável binária que indica se existe algum risco associado manualmente ao cliente.
- ***adversemedia_flag***: Variável binária que indica a presença de informações negativas sobre o cliente.
- ***sar_flag***: Variável binária que indica relatórios de atividades suspeitas

Base de Dados *risk_score*

A terceira base de dados, chamada *risk_score*, desempenha um papel fundamental na avaliação de riscos relacionados a contas e clientes. Este conjunto de dados contém informações específicas, como códigos de risco, descrições, pesos e fatores de risco. Com um total de 1347 registos e 4 variáveis, o conjunto de dados *risk_score* é essencial para preparar os dados e modelar nossos algoritmos de ML.

Aqui estão as variáveis presentes neste conjunto de dados:

- ***code***: Variável categórica que pode ser numérica ou alfanumérica que contem códigos.
- ***description***: Variável de texto que contém descrições detalhadas das entradas associadas aos códigos em *code*.

- ***risk_weight***: Variável numérica que contém os valores dos riscos.

- ***risk_factor***: : Variável de texto que contém descrições gerais.

As variáveis no conjunto de dados *risk_score* desempenham um papel crítico na conversão das informações contidas nas bases de dados *account_info* e *customer_info* em pesos de risco relevantes. Essa conversão é essencial, pois os níveis de risco para os atributos dos clientes e contas dos clientes foram previamente classificados. Essa classificação, baseada em uma abordagem de Risco BC/FT conforme estipulada pela legislação e boas práticas da indústria, foi disponibilizada juntamente com os dados dos clientes. Isso permite que os modelos de ML compreendam e processem eficazmente as informações associadas às contas e aos clientes, capacitando-nos a tomar decisões informadas com base nos riscos identificados.

3.4 Preparação e Pré-Processamento dos Dados

A etapa de preparação e pré-processamento dos dados é uma pedra angular incontestável em todos os projetos de ML. No entanto, antes de utilizarmos os dados nos modelos, é de imperativa importância assegurar a formatação adequada, identificando e eliminando inconsistências, e preparar os dados para a transição eficiente em conhecimento utilizável. A magnitude dessa fase não deve ser subestimada, uma vez que exerce influência decisiva tanto na qualidade quanto no desempenho subsequente dos modelos de ML a serem concebidos.

Este processo de preparação e pré-processamento dos dados será composto por oito pontos essenciais. É importante destacar que não existe uma ordem ou quantidade fixa de passos a seguir, uma vez que cada base de dados terá as suas necessidades específicas, e cada indivíduo adotará a sua abordagem única para atender aos requisitos do projeto.

1. **União das Bases de Dados**

   Neste contexto, concentramos a nossa principal atenção nas bases de dados *customer_info* e *account_info*, uma vez que o *risk_score* apenas contem variáveis numéricas que substituem variáveis categóricos nas bases de dados anteriormente mencionados. Para alcançar essa concentração, realizaremos a união das bases de dados utilizando a variável comum *customer_id* presente em todas elas. Essa etapa é crucial para consolidar as informações relevantes e criar uma única base de dados que agregue informações de clientes e de suas contas.

2. **Análise Inicial das Variáveis**

   Iniciar a análise das variáveis é uma etapa fundamental no processo de preparação e pré-processamento dos dados. É importante compreender o significado de cada variável, identificar aquelas que serão eliminadas posteriormente e, especialmente, identificar a variável que nos indica o risco atual de cada cliente, *assigned_risk_id*. No contexto dos modelos de ML, esta variável desempenha o papel de variável de resposta, sendo crucial para o desenvolvimento dos nossos modelos. A variável *customer_type_code* também desempenha um papel central, uma vez que nos indica se estamos a lidar com um cliente particular ou corporativo. É essencial considerar os requisitos do *Know Your Client* (KYC) e compreender que cada tipo de cliente pode envolver variáveis específicas aplicáveis apenas a esse grupo. Portanto, a nossa estratégia envolve a divisão da base de dados em clientes particulares e corporativos, permitindo a tomada de decisões mais adequadas e personalizadas para cada um desses grupos. Esta abordagem contribuirá para a criação de modelos de ML mais precisos e eficazes, tendo em consideração as características distintas de cada tipo de cliente.

3. **Eliminação de Variáveis Irrelevantes**

   Na fase de preparação dos dados, é importante analisar as variáveis categóricas e identificar aquelas que não contêm informações relevantes para os modelos. Isso inclui variáveis que servem apenas para identificar contas ou clientes, bem como aquelas que descrevem ou categorizam outras variáveis, neste caso: *account_id*, *primeiro_titular*, *customer_id* e *customer_type_code*. No que diz respeito a clientes corporativos, identificamos que eles não fornecem informações relevantes para variáveis como *country_of_origin*, *occupation*, *pep_flag*, *bef_flag* e *tocpp*, tendo em conta os requisitos KYC. Estas variáveis serão eliminadas para simplificar o conjunto de dados e torná-lo mais eficiente para a construção de modelos de ML.

4. **Tratamento de Valores Ausentes**

   A identificação de valores ausentes é uma consideração crítica na preparação e pré-processamento dos dados. Foi possível identificar valores ausentes em 9 variáveis. Essa ocorrência de valores ausentes requer uma abordagem cuidadosa. As variáveis afetadas e a percentagem de valores em falta são as seguintes:

   - *country_of_origin* - 0,0025% (1)
   - *bussines_type* - 0,0125% (5)

- *occupation* - 0,0325% (13)
- *sa_proposito_rela* - 6,165% (2466)
- *segmento* - 38,44% (15376)
- *formjur* - 94,11% (37644)
- *manual_risk_flag* - 0,0025% (1)
- *adversemedia_flag* - 0,0025% (1)
- *sar_flag* - 0,0025% (1)

Com base nesta análise, concluímos que as variáveis *formjur* e *segmento* podem ser excluídas da base de dados. A decisão de exclusão baseia-se na alta percentagem de valores ausentes nessas variáveis e na impossibilidade de imputar valores com precisão, o que comprometeria a qualidade e a confiabilidade dos resultados.

5. **Transformação de Variáveis Categóricas em Numéricas**

Uma etapa crucial na preparação dos dados é a transformação das variáveis categóricas em variáveis numéricas. Para realizar essa transformação, utilizaremos a base de dados *risk_score* como referência, garantindo que todas as variáveis categóricas sejam adequadamente convertidas em formato numérico.

6. **Imputação de Valores Ausentes**

No que diz respeito ao tratamento de valores ausentes, é imperativo adotar uma abordagem cuidadosa para preservar a integridade e a completude dos nossos conjuntos de dados. Optamos por uma estratégia de imputação, na qual utilizamos a mediana de cada variável para substituir os valores em falta. Essa escolha fundamenta-se na capacidade da mediana de representar um valor central robusto em relação a *outliers*, garantindo uma imputação mais estável e confiável.

7. **Normalização dos Dados**

A normalização dos dados é uma etapa crucial no pré-processamento dos mesmos, especialmente quando se trabalha com variáveis que abrangem diferentes intervalos de valores. A normalização assegura que todas as variáveis tenham uma escala semelhante, o que é fundamental para muitos modelos de ML descritos em 3.5, uma vez que alguns deles podem ser sensíveis a dados não normalizados.

Uma das técnicas de normalização disponíveis em *Python*, através da biblioteca *Scikit-learn* 3.1.3, é conhecida como *Min-Max Scaler*. Esta técnica redimensiona os

valores de uma variável para um intervalo específico, geralmente entre 0 e 1, embora seja possível especificar um intervalo personalizado. Eis como funciona:

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{3.1}$$

Onde:

- X_{norm} representa o valor normalizado
- X representa o valor original
- X_{min} representa o valor mínimo da variável
- X_{mx} representa o valor máximo da variável

8. **Correlação de Variáveis**

Finalmente, antes de introduzir os dados nos modelos de ML, realizaremos uma análise de correlação entre as variáveis. Esse processo permitirá avaliar se algumas variáveis estão altamente correlacionadas, o que pode afetar a qualidade dos modelos.

Para esse efeito iremos utilizar a correlação de *Pearson* [9] , também conhecida como coeficiente de correlação de *Pearson* ou simplesmente coeficiente de correlação, é uma medida estatística que quantifica o grau de associação linear entre duas variáveis contínuas. A fórmula para calcular o coeficiente de correlação de *Pearson*, representado por r, entre duas variáveis X e Y é a seguinte:

$$r = \frac{cov(X, Y)}{\sqrt{var(X) \cdot var(Y)}} \tag{3.2}$$

Onde:

- $cov(X, Y)$ representa a covariância entre X e Y
- $var(X)$ representa a variância de X.
- $var(Y)$ representa a variância de Y.

O valor de r varia entre -1 e 1, e para fins de interpretação, consideraremos a seguinte faixa de valores:

- $r = -1$: Correlação perfeita e negativa, indicando que quando uma variável aumenta, a outra diminui na mesma proporção.
- $r = 0$: Ausência de correlação linear.
- $r = 1$: Correlação perfeita e positiva, indicando que quando uma variável aumenta, a outra aumenta na mesma proporção.
- $(-0,3 < r < 0,3) \wedge (r \neq 0)$: Correlação fraca.
- $(-0,8 < r \leq -0,3) \wedge (0,3 \leq r < 0,8)$: Correlação moderada.
- $(-1 \leq r \leq -0,8) \wedge (0,8 \leq r \leq 1)$: Correlação forte.

Para uma melhor visualização das correlações de *Pearson* entre variáveis, iremos utilizar uma matriz de correlação.

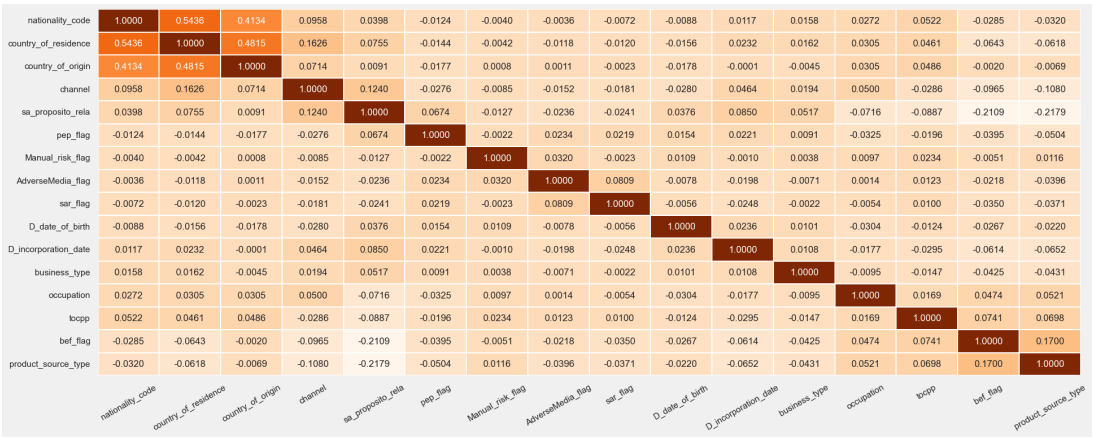


Figura 3.1: Matriz de Correlação para Clientes Particulares

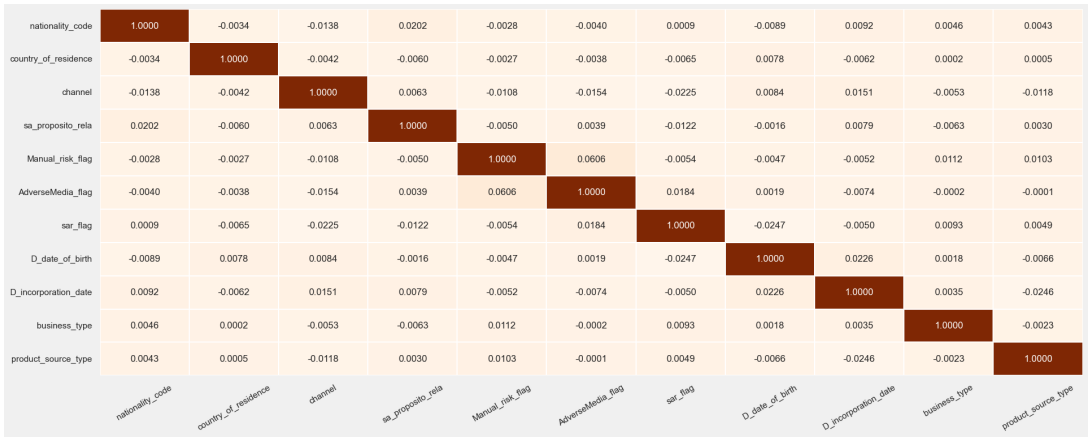


Figura 3.2: Matriz de Correlação para Clientes Corporativos

Na Figura 3.1, podemos observar que existe uma correlação moderada entre as variáveis *nationality_code*, *country_of_residence* e *country_of_origin*. Essa correlação moderada faz sentido, uma vez que todas essas variáveis estão relacionadas com

a origem do cliente, envolvendo informações como nacionalidade, naturalidade e o país onde a conta do cliente foi criada. Dado que essas correlações são moderadas, optamos por mantê-las no conjunto de dados e não eliminá-las. As restantes variáveis apresentam uma correlação fraca ou inexistente.

Por outro lado, na figura 3.2, todas as variáveis apresentam correlações fracas ou inexistentes entre si.

Podemos, assim, concluir que não será necessário eliminar ou agregar quaisquer variáveis dado que, as correlações fracas entre variáveis são benéficas para os nossos modelos de ML, pois evita multicolinearidade, que ocorre quando duas ou mais variáveis independentes estão fortemente correlacionadas entre si. A multicolinearidade pode levar a resultados instáveis e menos interpretáveis nos modelos. Portanto, a presença de correlações fracas ou moderadas é desejável, pois mantém a independência entre as variáveis e facilita a interpretação dos resultados do modelo.

Dispomos de um total de 30000 registos descritas por 16 variáveis para clientes particulares e 10000 registos descritas por 11 variáveis para clientes corporativos.

9. **Divisão dos Dados em Treino e Teste**

Antes de avançar para a construção dos nossos modelos de ML, é fundamental dividir as bases de dados de Clientes Particulares e Corporativos em conjuntos de treino e teste. Optámos por uma proporção de 80% dos dados para treino e 20% para teste, a fim de avaliar o desempenho dos nossos modelos num ambiente controlado (*seen data*), onde será possível otimizar os modelos alterando hiperparâmetros e, posteriormente, num ambiente de teste (*unseen data*), onde podemos avaliar a capacidade de generalização para dados não utilizados no treino. Esta estratégia de divisão garante que tenhamos conjuntos de dados independentes para treinar e testar os nossos modelos, permitindo-nos avaliar a sua capacidade de generalização para situações não previamente observadas no mundo real. Além disso, esta abordagem pode também proporcionar *insights* valiosos sobre se os modelos podem estar a sofrer de *overfitting*, o que acontece quando um modelo se ajusta demasiado bem aos dados de treino, mas não apresenta um bom desempenho nos dados de teste.

Para clientes particulares, o conjunto de treino é composto por 24000 registos, dos quais 1211 são classificados como clientes de alto risco com base no campo *assigned_risk_id*. O conjunto de teste inclui 6000 registos, com 290 deles identificados como clientes de alto risco.

No caso de clientes corporativos, o conjunto de treino consiste em 8000 registos, dos quais 327 são classificados como clientes de alto risco com base no campo *assigned_risk_id*. O conjunto de teste abrange 2000 registos, com 78 deles identificados como clientes de alto risco.

Com dados devidamente preparados, estamos agora preparados para prosseguir com a construção e treino de modelos de ML, com o objetivo de prever a nossa variável de resposta, *assigned_risk_id*, e gerir riscos de forma eficiente e eficaz. Esta sólida preparação de dados serve como alicerce para o sucesso do nosso estudo.

3.5 Modelos de ML

Os modelos de ML representam uma evolução significativa na forma como abordamos problemas complexos de análise de dados e tomada de decisões. Estes modelos são algoritmos computacionais com a capacidade de aprender padrões e fazer previsões com base em dados de treino. São amplamente utilizados em diversas áreas, desde o reconhecimento de imagens e processamento de linguagem natural até à previsão financeira e ao diagnóstico médico.

A essência do ML reside na capacidade dos modelos aprenderem e adaptarem-se a partir de dados, permitindo-lhes fazer generalizações e tomar decisões precisas em novos conjuntos de dados. Esta capacidade de automação e aprendizagem contínua torna estes modelos indispensáveis num mundo cada vez mais orientado por dados.

À medida que a tecnologia avança e os conjuntos de dados se tornam mais complexos, os modelos de ML continuam a evoluir, oferecendo soluções poderosas para uma ampla gama de desafios. Dominar estas técnicas não só impulsiona a capacidade de resolver problemas complexos, mas também desempenha um papel fundamental na inovação e no avanço da ciência e da tecnologia.

3.5.1 Modelos de Classificação

Neste capítulo, centramo-nos na exploração de uma variedade de modelos de classificação, que são técnicas de ML projetadas para categorizar dados em classes ou categorias pré-definidas. Os modelos de classificação desempenham um papel fundamental na análise de dados e na tomada de decisões, abrangendo diversos campos, desde a medicina até as finanças. A classificação envolve atribuir rótulos de classe a dados não rotulados, com base em padrões extraídos dos dados rotulados de treino.

Ao explorar estes modelos de classificação, procuramos obter um entendimento mais profundo das suas características, aplicabilidades e limitações. Cada um destes modelos

possui abordagens únicas para resolver problemas de classificação, e a escolha dependerá das características dos dados e dos objetivos do projeto.

3.5.1.1 Modelos Baseados em Árvores

Iniciamos por explorar os Modelos Baseados em Árvores, onde examinamos detalhadamente a Árvore de Decisão e as Florestas Aleatórias. A Árvore de Decisão é uma estrutura hierárquica que toma decisões sequenciais com base em atributos, enquanto que as Florestas Aleatórias consistem num conjunto de árvores de decisão independentes que trabalham em conjunto para obter uma previsão final.

3.5.1.1.1 Árvore de Decisão

A Árvore de Decisão é uma técnica de ML que representa uma estrutura hierárquica de decisões baseadas em atributos, utilizada para resolver problemas de classificação e regressão [33] . Essa abordagem tem suas origens em diversas disciplinas, como a estatística e a teoria da decisão, sendo amplamente adotada no campo da inteligência artificial e mineração de dados.

Formulação, Funcionamento e Construção da Árvore de Decisão

Uma árvore de decisão é construída como uma estrutura de árvore, na qual cada nó interno representa uma decisão baseada em um atributo e cada folha representa um resultado final ou classe. O processo de construção começa escolhendo o atributo mais relevante para dividir o conjunto de dados em subconjuntos mais homogéneos. A seleção do atributo é feita com base em critérios como o índice Gini ou a Entropia.

Construção da Árvore de Decisão

1. **Escolha do Atributo Inicial**:
   O processo começa pela seleção de um atributo relevante para a primeira divisão do conjunto de dados. Normalmente, escolhe-se o atributo que melhor distingue as diferentes classes de saída, com base em medidas de impureza, como o índice Gini ou a Entropia.

2. **Cálculo da Impureza e Divisão**:
   O atributo escolhido é utilizado para dividir o conjunto de dados em subconjuntos com base nos valores desse atributo. A medida de impureza (como o índice Gini ou a Entropia) é calculada para cada subconjunto. O objetivo é maximizar a redução da impureza após a divisão.

A fórmula geral para o cálculo do índice Gini em um nó é:

$$Gini(D) = 1 - \sum_{i=1}^{c} (p_i)^2 \tag{3.3}$$

Onde:

- D é o conjunto de dados.
- c é o número de classes possíveis.
- p_i é a proporção de observações da classe i em D.

A Entropia de *Shannon*, outra medida de impureza, é calculada da seguinte forma:

$$Entropia(D) = -\sum_{i=1}^{c} p_i \log_2(p_i) \tag{3.4}$$

Onde:

- D é o conjunto de dados.
- c é o número de classes possíveis.
- p_i é a proporção de observações da classe i em D.

Estas métricas medem o grau de impureza dos dados em um nó e são usadas para decidir qual atributo dividirá os dados de forma mais eficaz.

3. **Recursão para os Nós Filhos**:
   O procedimento é repetido para cada subconjunto gerado após a divisão. Ou seja, cada subconjunto se transforma num nó interno da árvore, e o processo de seleção de atributo e divisão é aplicado novamente a esses subconjuntos. Essa recursão prossegue até que um critério de paragem seja atingido.

4. **Critérios de Paragem**:
   Os critérios de paragem podem ser definidos para controlar o crescimento da árvore. Isso inclui limitar a profundidade da árvore, o número mínimo de exemplos por nó ou uma medida mínima de impureza. Quando um critério de paragem é alcançado, a recursão é interrompida e os nós finais são criados.

5. **Criação das Folhas**:
   Quando a recursão é interrompida, os nós finais da árvore são denominados folhas. Cada folha representa uma classe ou valor de regressão. O rótulo da classe é determinado pela classe mais comum nos exemplos presentes na folha. Essas folhas são responsáveis por fornecer a previsão final para um novo exemplo.

6. **Classificação de Novos Exemplos**:
   Para classificar um novo exemplo utilizando a árvore, começa-se pelo nó raiz e segue-se pelo caminho de decisões adequado com base nos valores dos atributos do exemplo. Esse processo repete-se até atingir uma folha, que fornece a previsão final para a classe ou valor de regressão do exemplo.

As árvores de decisão oferecem uma abordagem intuitiva e interpretação direta, sendo capazes de capturar padrões complexos nos dados. No entanto, podem ser suscetíveis ao *overfitting* (ajuste excessivo ao conjunto de treino) se não forem devidamente controladas. Apesar disso, continuam a ser uma ferramenta valiosa para resolver problemas complexos de classificação e regressão.

Vantagens

1. **Interpretação Intuitiva**:
   As árvores de decisão são fáceis de entender e interpretar, pois a sua estrutura é semelhante a um fluxograma de decisões. Isso facilita a comunicação dos resultados para não especialistas.

2. **Lida com Dados Mistos**:
   As árvores de decisão podem lidar com uma combinação de variáveis categóricas e numéricas, tornando-as versáteis para diferentes tipos de dados.

3. **Captura de Padrões Complexos**:
   As árvores podem capturar relações não lineares e interações complexas entre variáveis, tornando-as úteis para problemas em que as relações são intrincadas.

4. **Não Exige Normalização de Dados**:
   Diferentemente de muitos algoritmos, as árvores de decisão não requerem que os dados sejam normalizados ou padronizados, o que poupa tempo no pré-processamento.

5. **Robustez a *Outliers***:
   As árvores de decisão são menos afetadas por valores atípicos em comparação com algoritmos sensíveis, como a regressão linear.

Desvantagens

1. ***Overfitting***:
   As árvores de decisão podem ajustar-se muito bem aos dados de treino e, assim, podem capturar ruído e levar a uma má generalização para novos dados.

2. **Instabilidade**:
   Pequenas mudanças nos dados de treino podem levar a grandes variações na estrutura da árvore, tornando-as menos estáveis.

3. **Tendência para Classes Raras**:
   Em conjuntos de dados desequilibrados, as árvores de decisão podem ser enviesadas em direção às classes mais frequentes, tornando difícil a previsão das classes menos comuns.

4. **Limitação em Regressão**:
   As árvores de decisão não são tão adequadas para problemas de regressão, especialmente quando os dados têm uma relação mais complexa.

5. **Seleção de Atributos**:
   A escolha inadequada de atributos pode resultar em árvores subótimas ou excessivamente complexas.

As árvores de decisão são uma ferramenta poderosa para resolver problemas de classificação e regressão. Embora tenham vantagens notáveis, como interpretação fácil e flexibilidade em relação aos tipos de dados, também apresentam desafios, como *overfitting* e instabilidade. Ao aplicar árvores de decisão, é fundamental considerar as suas vantagens e desvantagens específicas em relação ao contexto do problema e à qualidade dos dados. Além disso, técnicas de controlo de overfitting, poda de árvore e uso de ensemble learning podem ajudar a mitigar algumas das suas limitações.

3.5.1.1.2 Florestas Aleatórias

As Florestas Aleatórias, também conhecidas como *Random Forests*, são uma técnica avançada no campo dos modelos de árvores de decisão. Essa abordagem visa melhorar a precisão e a robustez das previsões, superando algumas das limitações das árvores de decisão individuais. As Florestas Aleatórias são um exemplo de *Ensemble Learning*, em que várias árvores individuais são combinadas para formar uma abordagem mais poderosa e precisa.

Funcionamento e Construção das Florestas Aleatórias

O funcionamento das Florestas Aleatórias baseia-se na criação de um conjunto diversificado de árvores de decisão [12]. Cada árvore é treinada com uma amostra aleatória de conjunto de treino e utiliza uma abordagem de amostragem com reposição, conhecida como Bootstrap Aggregating ou Bagging. Além disso, durante a construção de cada árvore, uma seleção aleatória de atributos é usada para divisões nos nós. Esses dois aspectos

de aleatoriedade - amostragem dos dados e seleção de atributos - garantem que as árvores individuais sejam diferentes entre si.

Combinação de Previsões
Uma vez que todas as árvores individuais são construídas, a previsão de um novo exemplo é obtida coletando as previsões de cada árvore e realizando uma votação ou média para determinar a classe final. Essa abordagem de combinação de previsões ajuda a reduzir o risco de *overfittin*g e a melhorar a generalização para novos dados.

Vantagens

1. **Alta Precisão**:
   As Florestas Aleatórias geralmente produzem resultados altamente precisos devido à combinação de várias árvores de decisão independentes, resultando em previsões mais robustas e confiáveis.

2. **Resistência ao *Overfitting***:
   A técnica de construir várias árvores e combinar seus resultados reduz a tendência de overfitting, onde o modelo se ajusta em excesso ao conjunto de treino, tornando-o mais generalizável para novos exemplos.

3. **Lida com Dados Desbalanceados**:
   As Florestas Aleatórias são eficazes em lidar com conjuntos de dados desbalanceados, onde há uma diferença significativa no número de exemplos entre classes, mitigando o viés em direção à classe maioritária.

Desvantagens

1. **Complexidade Computacional**:
   Devido à construção de múltiplas árvores e sua posterior combinação, as Florestas Aleatórias podem ser computacionalmente intensivas, exigindo mais recursos de processamento e tempo de treino.

2. **Interpretação Desafiadora**:
   Enquanto uma única árvore de decisão é relativamente fácil de interpretar, a combinação de várias árvores nas Florestas Aleatórias pode dificultar a compreensão das razões subjacentes por trás das previsões.

Explorar essas vantagens e desvantagens nos permitirá tomar decisões informadas ao empregar as Florestas Aleatórias em cenários de modelagem específicos, ponderando seus benefícios e limitações.

3.5.1.2 Modelos de Vizinhança e Similaridade

Em seguida, aprofundamos os Modelos de Vizinhança e Similaridade , destacando o *K-Nearest Neighbors* (KNN) [22]. O KNN classifica os dados com base na proximidade entre pontos de dados, em que a classe de um ponto não rotulado é determinada pela maioria das classes dos k pontos mais próximos.

3.5.1.2.1 KNN

O KNN é uma técnica de ML que se baseia num princípio fundamental: a proximidade entre dados semelhantes. Esta abordagem tem as suas origens em conceitos intuitivos e oferece uma forma poderosa de realizar tarefas de classificação e regressão. Com a sua simplicidade e eficácia, o KNN destacou-se como uma ferramenta versátil em várias situações de análise de dados.

Formulação e Funcionamento

Como já descrito anteriormente, o objetivo do KNN é realizar classificações ou previsões baseadas na similitude entre exemplos. Ao receber um novo exemplo, o KNN identifica os k exemplos de treino mais próximos com base em uma métrica de distância.

A formulação geral do KNN envolve os seguintes passos:

1. **Cálculo de Distâncias**:
   O primeiro passo consiste em calcular as distâncias entre o exemplo de teste (aquele a ser classificado ou previsto) e todos os exemplos de treino presentes no conjunto de dados. Para isso, diferentes métricas podem ser utilizadas, como a distância euclidiana.

   A fórmula para calcular a distância euclidiana é:

$$D = \sqrt{\sum_{i=1}^{d} (p_i - q_i)^2} \tag{3.5}$$

   Onde:

   - p e q são dois pontos no espaço.
   - d é o número da dimensão do espaço.

2. **Seleção dos Vizinhos**:
   Com as distâncias calculadas, os k exemplos de treino mais próximos do exemplo de teste são escolhidos como seus vizinhos mais próximos.

3. **Decisão de Classe ou Regressão**:
   No caso de tarefas de classificação, a classe mais frequente entre os k vizinhos é atribuída ao exemplo de teste. Já em situações de regressão, a média ou mediana dos valores dos k vizinhos é calculada e adotada como a estimativa de valor para o exemplo de teste.

A escolha do valor de k tem um papel crucial no desempenho do KNN. Um k pequeno pode tornar o modelo sensível a ruídos, enquanto um k grande pode suavizar as decisões, perdendo nuances dos dados.

Vantagens

1. **Simplicidade**:
   A sua simplicidade intrínseca torna o KNN acessível mesmo para iniciantes.

2. **Versatilidade**:
   Pode ser aplicado tanto em problemas de classificação como de regressão, adaptando-se às necessidades específicas.

3. **Natureza Não Paramétrica**:
   A falta de pressupostos rígidos sobre a distribuição dos dados confere-lhe flexibilidade.

4. **Capacidade de Abordar Padrões Complexos**:
   É capaz de identificar padrões complexos, especialmente em problemas com fronteiras de decisão não lineares.

5. **Mitigação do Overfitting**:
   O KNN mostra menor tendência ao overfitting quando o valor de k é selecionado adequadamente.

Desvantagens

1. **Sensibilidade a *Outliers***:
   Outliers podem ter um impacto significativo nas previsões, especialmente em cenários com um valor pequeno de K.

2. **Custo Computacional**:
   O cálculo das distâncias entre exemplos pode ser computacionalmente dispendioso para conjuntos de dados extensos.

3. **Escolha do Valor de K**:
   A definição adequada do valor de K é um desafio, uma vez que influencia o desempenho do algoritmo.

4. **Desempenho em Dimensões Elevadas**:
   A eficácia do KNN pode ser afetada adversamente em cenários de alta dimensionalidade, resultando em desafios como a dispersão dos pontos de dados no espaço e a diluição da densidade dos vizinhos mais próximos.

5. **Desbalanceamento de Classess**:
   Em conjuntos de dados desbalanceados, o KNN pode favorecer classes maioritárias.

3.5.1.3 Modelos Probabilísticos

Exploramos, agora, os Modelos Probabilísticos, incluindo a Regressão Logística [6] e o *Naive Bayes* [31], que se baseiam em probabilidades para fazer previsões. A Regressão Logística é uma técnica de regressão usada para prever a probabilidade de uma classe binária, enquanto que o *Naive Bayes* se baseia no Teorema de *Bayes* e na hipótese de independência condicional entre as variáveis.

3.5.1.3.1 Regressão Logística

A regressão logística é uma técnica de ML amplamente utilizada para tarefas de classificação. Baseia-se numa abordagem estatística que visa modelar a relação entre variáveis independentes e uma variável dependente binária ou categórica.

Formulação e Funcionamento

A regressão logística implica a modelação da probabilidade de uma variável dependente pertencer a uma classe específica. Este processo é realizado através de uma função logística que transforma uma combinação linear das variáveis independentes numa probabilidade situada entre 0 e 1. A função sigmoid permite uma adaptação eficaz do modelo a dados binários ou multicategóricos. Durante o processo de treino, os coeficientes das variáveis independentes são ajustados para otimizar a probabilidade prevista.

A regressão logística faz parte dos Modelos Lineares Generalizados (GLM), que são uma extensão dos modelos de regressão linear para diferentes tipos de variáveis de resposta.

A função logística, também conhecida como função sigmoid, é definida da seguinte forma:

$$f(z) = \frac{1}{1 + e^{-z}} \tag{3.6}$$

Onde:

- $f(z)$ representa a probabilidade estimada de pertencer à classe positiva.
- z é a combinação linear das variáveis independentes ponderadas pelos coeficientes:

$$z = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \tag{3.7}$$

O modelo utiliza o método de máxima verossimilhança para estimar os coeficientes $\beta_0, \beta_1, \ldots, \beta_n$. Após a estimativa dos coeficientes, a probabilidade prevista $f(z)$ é utilizada para efetuar a classificação. Um limiar é definido, como por exemplo 0,5, para determinar a classe final: se $f(z)$ for superior ao limiar, o valor de 1 é atribuído; caso contrário, é atribuído valor de 0.

Vantagens

1. **Interpretabilidade**:
   Os coeficientes da regressão logística fornecem informações claras sobre a direção e magnitude do impacto das variáveis independentes na variável dependente.
2. **Ampla Aplicabilidade**:
   A regressão logística pode ser usada em uma variedade de problemas de classificação, desde tarefas binárias até problemas multicategóricos.
3. **Lida com Dados Desbalanceados**:
   Pode lidar bem com conjuntos de dados desbalanceados, o que é comum em problemas de classificação do mundo real.
4. **Eficiente para Grandes Conjuntos de Dados**:
   É computacionalmente eficiente mesmo em grandes conjuntos de dados, tornando-o prático para cenários do mundo real.

Desvantagens

1. **Assume Linearidade**:
   A regressão logística assume uma relação linear entre as variáveis independentes e a probabilidade da variável dependente, o que pode ser limitante em alguns casos.

2. **Suscetível a *Outliers***:
   Outliers podem distorcer os coeficientes da regressão e influenciar significativamente as previsões.

3. **Pode Sofrer de *Overfitting***:
   Como qualquer modelo, a regressão logística pode sofrer de *overfitting* se não for controlada adequadamente.

4. **Dependência de Atributos Relevantes**:
   A qualidade das previsões da regressão logística depende da inclusão de atributos relevantes e da remoção de atributos irrelevantes.

3.5.1.3.2 *Naive Bayes*

O Naive Bayes é uma técnica de ML baseada no teorema de Bayes, que utiliza a probabilidade condicional para realizar tarefas de classificação e análise de texto.

Formulação e Funcionamento

O Naive Bayes baseia-se na aplicação do teorema de Bayes para calcular a probabilidade de um exemplo pertencer a uma determinada classe, dado um conjunto de características observadas.

A fórmula geral é expressa como:

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)} \tag{3.8}$$

Onde:

- $P(C|X)$ é a probabilidade de pertencer à classe C dado o conjunto de características X.
- $P(X|C)$ é a probabilidade das características X dado que pertencem à classe C.
- $P(C)$ é a probabilidade de pertencer à classe C independentemente das características.
- $P(X)$ é a probabilidade do conjunto de características X.

A suposição *"naive"* é que as características são independentes, o que simplifica o cálculo das probabilidades condicionais.

Vantagens

1. **Eficiente**:
   O Naive Bayes é computacionalmente eficiente e rápido de treinar.

2. **Lida com dados categóricos**:
   É adequado para dados categóricos e de texto.

3. **Tratamento de atributos em falta**:
   Lida bem com atributos em falta.

Desvantagens

1. **Suposição de independência**:
   A suposição de independência condicional nem sempre é realista.

2. **Sensibilidade a variáveis irrelevantes**:
   Variáveis irrelevantes podem afetar negativamente o desempenho.

3. **Problemas com dados desbalanceados**:
   Pode ter dificuldade em lidar com classes desbalanceadas.

3.5.1.4 Modelos Baseados em Otimização

Posteriormente, examinamos os Modelos Baseados em Otimização, com foco na SVM [7]. A SVM procura encontrar um hiperplano que maximize a margem entre as classes, permitindo a classificação linear ou não linear dos dados.

3.5.1.4.1 SVM

As SVM, também conhecidas como Máquinas de Vetores de Suporte, representam uma classe distintiva de algoritmos no campo do ML que têm profundas raízes na teoria das margens de separação máxima. Influenciadas por avanços na área de estatística e teoria da otimização, as SVM foram formuladas com o objetivo primordial de alcançar uma separação ideal entre as classes de dados.

Formulação e Funcionamento

As SVM têm a sua essência na busca pela maximização da margem entre as classes de dados. Essa margem é a distância entre os vetores de suporte, que representam os exemplos de treino mais próximos da fronteira de decisão, conhecida como hiperplano de separação. O objetivo das SVM é encontrar o hiperplano que maximiza essa margem, o que resulta numa melhor capacidade de generalização para novos dados.

Matematicamente, o hiperplano de separação pode ser definido como:

$$w \cdot x + b = 0 \tag{3.9}$$

Onde:

- w é um vetor normal ao hiperplano.
- b é o termo de viés.

O funcionamento das SVM envolve a transformação dos dados de entrada x para um espaço de maior dimensão, onde podem ser separados linearmente. Essa transformação é realizada através de uma função de *kernel* $K(x, x')$, que determina o produto interno entre os vetores transformados. A escolha do *kernel* é crucial, e diferentes *kernel*, como linear, polinomial e *radial basis function* (RBF), podem ser utilizados para a transformação.

A equação que define a classificação usando o *kernel* é:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x, x_i) + b \tag{3.10}$$

Onde:

- α_i são os coeficientes de *Lagrange*.
- y_i são as classes dos observações de treino.
- x_i são os vetores de treino.
- b é o termo de viés.

O termo b nas SVM refere-se ao viés, que é um parâmetro responsável por controlar o deslocamento vertical do hiperplano de separação em relação à origem no espaço de atributos. Este parâmetro permite ajustar a posição do hiperplano de modo a otimizar a separação entre as classes de dados. Durante o treino das SVM, o valor de b é determinado de forma a minimizar os erros de classificação. O viés é especialmente importante quando os dados não são linearmente separáveis, permitindo que o hiperplano seja ajustado de maneira adequada para alcançar uma melhor separação das classes.

Vantagens

1. **Eficiência com Espaços de Alta Dimensão**:
   As SVM são especialmente eficazes quando os dados estão em espaços de alta dimensão, tornando-as ideais para problemas com muitos atributos.

2. **Capacidade de Lidar com Dados Não Lineares**:
   Através da utilização de funções de *kernel*, as SVM têm a capacidade de lidar com problemas de classificação não linear.

3. **Generalização Robusta**:
   Buscam a maximização da margem, o que pode levar a uma melhor capacidade de generalização para novos dados.

Desvantagens

1. **Sensibilidade a Parâmetros**:
   A escolha do *kernel* e outros parâmetros das SVM pode afetar significativamente o desempenho do modelo.

2. **Complexidade de Treino**:
   Para grandes conjuntos de dados, o treino das SVM pode ser computacionalmente intensivo.

3. **Requer Dados Rotulados**:
   São eficazes para classificação supervisionada, mas requerem dados rotulados para o treino.

3.5.2 Modelos *Ensemble Learning*

Nesta secção, concentramo-nos nos modelos *Ensemble Learning*, uma abordagem avançada que procura combinar as previsões de vários modelos individuais para melhorar a precisão e a robustez das previsões finais. Ao unir as capacidades de diferentes algoritmos, os modelos *Ensemble Learning* procuram alcançar resultados superiores aos modelos individuais.

Ao explorar os modelos *Ensemble Learning*, estamos a abordar uma técnica sofisticada que procura tirar partido dos pontos fortes de vários modelos para melhorar a qualidade das previsões. Esta técnica é especialmente útil quando se pretende maximizar a precisão e a robustez dos resultados, sendo amplamente aplicada em competições de ciência de dados e problemas complexos de ML. Cada subcategoria de modelos *Ensemble Learning* possui características distintas e é escolhida com base nas características dos dados e nos objetivos do projeto.

3.5.2.1 Modelos Bagged

Nesta seção, exploramos os *Modelos Bagged* [1], uma estratégia de *Ensemble Learning* que envolve a geração de múltiplos modelos idênticos, cada um treinado em um conjunto de dados de treino diferente. Essa abordagem ajuda a reduzir a variância, uma vez que os modelos individuais são treinados em amostras diferentes dos dados originais. O método de *Bagging (Bootstrap Aggregating)* é frequentemente aplicado a modelos baseados em árvores, como a Árvore de Decisão, criando variações nos dados de conjunto de treino por meio do reamostragem com reposição. Isso permite que cada modelo aprenda diferentes padrões, e suas previsões são então combinadas para produzir uma previsão final mais estável e geralmente mais precisa.

3.5.2.1.1 *Bootstrap Aggregating*

Os modelos *Bagged*, ou *Bootstrap Aggregating*, são uma técnica de ML que tem como objetivo melhorar o desempenho e a estabilidade dos modelos de base através da agregação de múltiplos modelos treinados em amostras de dados diferentes.

Formulação e Funcionamento

A técnica inicia-se com a criação de várias amostras de treino a partir do conjunto de dados original com o método de *bootstrap*. O *bootstrap* envolve a seleção aleatória de observações do conjunto de dados com reposição, resultando em conjuntos de treino de tamanho igual ou menor que o conjunto de dados original.

Para cada uma das amostras de treino geradas pelo *bootstrap*, um modelo de base é treinado. Estes modelos de base podem ser de qualquer algoritmo de ML, como árvores de decisão, regressão linear ou até mesmo redes neuronais.

- **Treino dos Modelos de Base**: Cada amostra de treino é utilizada para treinar um modelo de base separadamente. Como as amostras de treino são geradas aleatoriamente com reposição, os modelos de base aprenderão de maneira ligeiramente diferente devido às variações nos dados de treino.
- **Agregação**: Após o treino de todos os modelos de base, as suas previsões individuais são agregadas para produzir uma previsão final. A forma como as previsões são combinadas depende do tipo de problema. Por exemplo, em problemas de classificação, pode ser realizada uma votação para determinar a classe mais frequente entre os modelos. Para problemas de regressão, é calculada a média das previsões.

Esta abordagem de *Bagging* ajuda a reduzir a variância dos modelos, tornando-os mais robustos e geralmente levando a um melhor desempenho preditivo, evitando técnicas

como o *K-Fold Cross Validation*.

Vantagens

1. **Redução de Variância**: A principal vantagem dos modelos *Bagged* é a redução da variância. Uma vez que os modelos de base são treinados em diferentes amostras de dados, eles cometem erros diferentes. Quando as suas previsões são agregadas, os erros tendem a compensar-se, resultando num modelo agregado com menor variância em comparação com os modelos individuais.

2. **Robustez**: Os modelos *Bagged* são menos sensíveis a *outliers* e ruído nos dados devido à técnica de *Bootstrap*, que introduz aleatoriedade no processo de treino.

3. **Melhora a Estabilidade**: Esta técnica pode tornar modelos instáveis mais estáveis, melhorando o seu desempenho em conjuntos de dados com maior variabilidade.

Desvantagens

1. **Aumento no Tempo de Treino**: Treinar vários modelos de base pode ser computacionalmente dispendioso e demorado, especialmente se os modelos de base forem complexos.

2. **Complexidade do Modelo Final**: Em alguns casos, a agregação de muitos modelos de base pode resultar num modelo final complexo e de difícil interpretação.

3. **Limitação nas Melhorias**: Nem todos os modelos beneficiam igualmente do *Bagging*. Em problemas nos quais os modelos de base já possuem uma baixa variância, o *Bagging* pode não proporcionar melhorias significativas. É importante selecionar os modelos de base apropriados.

3.5.2.2 Modelos *Boost*

Por fim, exploramos os Modelos *Boost*, um conjunto de algoritmos que têm como objetivo ajustar sequencialmente modelos individuais para corrigir os erros cometidos pelos modelos anteriores. Entre estes modelos, destacamos o *AdaBoost*, que se concentra em identificar e ponderar classificações que foram incorretamente feitas pelo modelo anterior, permitindo que o próximo modelo se concentre nos exemplos mais desafiadores. Além disso, discutimos o *Stochastic Gradient Boosting*, que combina o conceito de *Gradient Boosting* com amostragem estocástica, melhorando a velocidade e eficiência do processo de aprendizagem. Por fim, analisamos o *XGBoost* , uma implementação otimizada de *Gradient Boosting* que apresenta funcionalidades avançadas, como o tratamento de valores em falta e regularização.

3.5.2.2.1 *AdaBoost*

O *AdaBoost (Adaptive Boosting)* [25] é um algoritmo de ML que visa melhorar o desempenho de modelos de árvore de decisão fracos por meio da combinação ponderada deles.

Formulação e Funcionamento

O algoritmo opera da seguinte forma:

1. **Inicialização dos Pesos das Amostras**: No início do processo de treino, todos os pesos das amostras são inicializados com valores iguais, de modo que cada amostra tenha a mesma importância:

$$w_{1,i} = \frac{1}{N} \tag{3.11}$$

   Onde:

   - N é o número total de amostras de treino.
   - $i = 1, 2, ..., N$

2. **Seleção e Treino dos Modelos de Árvore Fracos:** O *AdaBoost* itera sobre um conjunto de modelos de árvore de decisão fracos. Em cada iteração t, o algoritmo faz o seguinte:

   a) Seleciona um modelo de árvore fraco $h_t(x)$ que se concentre nas amostras que foram classificadas incorretamente nas iterações anteriores. A escolha de $h_t(x)$ é baseada no seu desempenho ponderado nas amostras de treino.

   b) Calcula o erro ponderado do modelo fraco ϵ_t na iteração t:

$$\epsilon_t = \sum_{i=1}^{N} w_{t,i} \cdot I(h_t(x_i) \neq y_i) \tag{3.12}$$

   Onde:

   - $h_t(x_i)$ é a previsão do modelo fraco t para a amostra x_i.
   - y_i é o rótulo verdadeiro da amostra x_i.

- $I(\cdot)$ é uma função indicatriz que retorna 1 se a previsão do modelo for incorreta e 0 se for correta.

c) Calcula o peso α_t atribuído ao modelo fraco t com base no erro ponderado ϵ_t:

$$\alpha_t = \frac{1}{2} \ln\left(\frac{1-\epsilon_t}{\epsilon_t}\right) \tag{3.13}$$

3. **Atualização dos Pesos das Amostras:** Os pesos das amostras são atualizados para a próxima iteração $t+1$ da seguinte forma:

$$w_{t+1,i} = w_{t,i} \cdot \exp(-\alpha_t \cdot y_i \cdot h_t(x_i)) \tag{3.14}$$

Onde:

- α_t é o peso do modelo fraco na iteração t.
- y_i é o rótulo verdadeiro da amostra x_i.
- $h_t(x_i)$ é a previsão do modelo fraco t para a amostra x_i.

4. **Normalização dos Pesos:** Após a atualização dos pesos das amostras, serão normalizados de forma que a soma de todos os pesos seja igual a 1:

$$w_{t+1,i} = \frac{w_{t+1,i}}{\sum_{i=1}^{N} w_{t+1,i}} \tag{3.15}$$

5. **Agregação dos Modelos Fracos:** O modelo *AdaBoost* final é uma combinação ponderada dos modelos de árvore fracos. A previsão do modelo *AdaBoost* é calculada como:

$$H(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)\right) \tag{3.16}$$

Onde:

- T é o número total de iterações (ou modelos fracos).
- α_t é o peso do modelo fraco t.
- $h_t(x)$ é a previsão do modelo fraco t para a amostra x.
- sign(·) é a função de sinal que retorna 1 se o valor for positivo e -1 se for negativo.

Vantagens

1. **Melhoria do Desempenho:** O *AdaBoost* é capaz de melhorar significativamente o desempenho de modelos fracos, resultando num modelo forte de alta precisão.
2. **Adaptação:** O algoritmo é adaptativo, dando mais peso às amostras mal classificadas em cada iteração, o que o torna eficaz na resolução de problemas difíceis.
3. **Flexibilidade de Modelo Fraco:** O *AdaBoost* pode ser combinado com diferentes modelos de ML fracos, permitindo flexibilidade na escolha do modelo adequado para o problema.

Desvantagens

1. **Sensibilidade a *Outliers*:** O *AdaBoost* pode ser sensível a *outliers*, uma vez que atribui pesos maiores às amostras mal classificadas, o que pode resultar em modelos suscetíveis a ruídos nos dados.
2. ***Overfitting***: Se não for controlado adequadamente, o *AdaBoost* pode sofrer de *overfitting*, especialmente quando modelos fracos muito complexos são usados.
3. **Computacionalmente Intensivo:** O algoritmo pode ser computacionalmente intensivo, uma vez que envolve várias iterações de treino de modelos fracos.

3.5.2.2.2 *Stochastic Gradient Boosting*

O *Stochastic Gradient Boosting (SGD Boost)* [10] é um algoritmo de ML que visa melhorar o desempenho de modelos fracos por meio da combinação ponderada deles.

Formulação e Funcionamento
O algoritmo opera da seguinte forma:

1. **Inicialização dos Pesos das Amostras:** No início do processo de treino, todos os pesos das amostras são inicializados com valores iguais, à semelhança do *AdaBoost*, de modo que cada amostra tenha a mesma importância:

$$w_{1,i} = \frac{1}{N} \tag{3.17}$$

Onde:

- N é o número total de amostras de treino.
- $i = 1, 2, ..., N$

2. **Seleção e Treino dos Modelos de Regressão Fracos:** O *SGD Boost* itera sobre um conjunto de modelos de regressão fracos. Em cada iteração t, o algoritmo faz o seguinte:

   a) Seleciona um modelo de regressão fraco $h_t(x)$ que se concentre nas amostras que foram classificadas incorretamente nas iterações anteriores. A escolha de $h_t(x)$ é baseada no seu desempenho ponderado nas amostras de treino.

   b) Calcula o erro residual ϵ_t na iteração t:

$$\epsilon_t = y_i - H_{t-1}(x_i) \tag{3.18}$$

   Onde:

   - y_i é o rótulo verdadeiro da amostra x_i.
   - $H_{t-1}(x_i)$ é a previsão acumulada dos modelos anteriores para a amostra x_i.

   c) Treina $h_t(x)$ para ajustar os resíduos ϵ_t.

3. **Atualização dos Pesos das Amostras:** Os pesos das amostras são atualizados para a próxima iteração $t + 1$ da seguinte forma:

$$w_{t+1,i} = w_{t,i} \cdot \exp(-\alpha_t \cdot \epsilon_t) \tag{3.19}$$

   Onde:

   - α_t é o peso do modelo fraco na iteração t.
   - ϵ_t é o erro residual calculado anteriormente.

4. **Normalização dos Pesos:** Após a atualização dos pesos das amostras, eles são normalizados de forma que a soma de todos os pesos seja igual a 1:

$$w_{t+1,i} = \frac{w_{t+1,i}}{\sum_{i=1}^{N} w_{t+1,i}} \tag{3.20}$$

5. **Agregação dos Modelos Fracos:** O modelo final do *SGD Boost* é uma combinação ponderada dos modelos de regressão fracos. A previsão do modelo *SGD Boost* é calculada como:

$$H_t(x) = H_{t-1}(x) + \alpha_t \cdot h_t(x) \tag{3.21}$$

Onde:

- $t = \{1, 2, ..., T\}$.
- α_t é o peso do modelo fraco t.
- $h_t(x)$ é a previsão do modelo fraco t para a amostra x.

Vantagens

1. **Melhoria do Desempenho:** O *SGD Boost* é capaz de melhorar significativamente o desempenho de modelos fracos, resultando em um modelo forte de alta precisão.
2. **Adaptação:** O algoritmo é adaptativo, ajustando-se aos erros residuais nas iterações subsequentes, o que o torna eficaz na resolução de problemas difíceis.
3. **Flexibilidade de Modelo Fraco:** O *SGD Boost* pode ser combinado com diferentes modelos de regressão fracos, permitindo flexibilidade na escolha do modelo adequado para o problema.

Desvantagens

1. **Sensibilidade a *Outliers*:** O *SGD Boost* pode ser sensível a *outliers*, uma vez que atribui pesos maiores às amostras com erros residuais maiores.
2. ***Overfitting*:** Se não for controlado adequadamente, o *SGD Boost* pode sofrer de *overfitting*, especialmente quando modelos fracos muito complexos são usados.
3. **Computacionalmente Intensivo:** O algoritmo pode ser computacionalmente intensivo, uma vez que envolve várias iterações de treinamento de modelos fracos.

3.5.2.2.3 *XGBoost*

O *XGBoost (Extreme Gradient Boosting)* é um algoritmo de ML que visa melhorar o desempenho de modelos fracos por meio da combinação ponderada deles.

Formulação e Funcionamento

O algoritmo opera da seguinte forma:

1. **Inicialização dos Pesos das Amostras:** No início do processo de treino, todos os pesos das amostras são inicializados com valores iguais, à semelhança do *AdaBoost* e *SGDBoost*, de modo que cada amostra tenha a mesma importância:

$$w_{1,i} = \frac{1}{N} \tag{3.22}$$

   Onde:

   - N é o número total de amostras de treino.
   - $i = 1, 2, ..., N$

2. **Seleção e Treino dos Modelos de Regressão Fracos:** O *XGBoost* itera sobre um conjunto de modelos de regressão fracos. Em cada iteração t, o algoritmo faz o seguinte:

   a) Seleciona um modelo de regressão fraco $h_t(x)$ que se concentre nas amostras que foram classificadas incorretamente nas iterações anteriores. A escolha de $h_t(x)$ é baseada no seu desempenho ponderado nas amostras de treino.

   b) Calcula o gradiente ∇ da função de perda com base nas previsões acumuladas até a iteração atual e nos rótulos verdadeiros:

$$\nabla = \nabla L(y_i, H_{t-1}(x_i)) \tag{3.23}$$

   Onde:

   - y_i é o rótulo verdadeiro da amostra x_i.
   - $H_{t-1}(x_i)$ é a previsão acumulada dos modelos anteriores para a amostra x_i.

- L é a função de perda utilizada.

c) Treina $h_t(x)$ para ajustar o gradiente ∇.

3. **Atualização dos Pesos das Amostras:** Os pesos das amostras são atualizados para a próxima iteração $t+1$ da seguinte forma:

$$w_{t+1,i} = w_{t,i} \cdot \exp(-\alpha_t \cdot \nabla) \tag{3.24}$$

Onde:

- α_t é o peso do modelo fraco na iteração t.
- ∇ é o gradiente calculado anteriormente.

4. **Normalização dos Pesos:** Após a atualização dos pesos das amostras, eles são normalizados de forma que a soma de todos os pesos seja igual a 1:

$$w_{t+1,i} = \frac{w_{t+1,i}}{\sum_{i=1}^{N} w_{t+1,i}} \tag{3.25}$$

5. **Agregação dos Modelos Fracos:** O modelo final do *XGBoost* é uma combinação ponderada dos modelos de regressão fracos. A previsão do modelo *XGBoost* é calculada como:

$$H_t(x) = H_{t-1}(x) + \alpha_t \cdot h_t(x) \tag{3.26}$$

Onde:

- $t = \{1, 2, ..., T\}$.
- α_t é o peso do modelo fraco t.
- $h_t(x)$ é a previsão do modelo fraco t para a amostra x.

Vantagens

1. **Melhoria do Desempenho:** O *XGBoost* é capaz de melhorar significativamente o desempenho de modelos fracos, resultando em um modelo forte de alta precisão.

2. **Adaptação:** O algoritmo é adaptativo, ajustando-se aos gradientes das funções de perda nas iterações subsequentes, o que o torna eficaz na resolução de problemas difíceis.

3. **Flexibilidade de Modelo Fraco:** O *XGBoost* pode ser combinado com diferentes modelos de regressão fracos, permitindo flexibilidade na escolha do modelo adequado para o problema.

Desvantagens

1. **Sensibilidade a Outliers:** O *XGBoost* pode ser sensível a outliers, uma vez que atribui pesos maiores às amostras com gradientes maiores.

2. **Overfitting:** Se não for controlado adequadamente, o *XGBoost* pode sofrer de overfitting, especialmente quando modelos fracos muito complexos são usados.

3. **Computacionalmente Intensivo:** O algoritmo pode ser computacionalmente intensivo, uma vez que envolve várias iterações de treinamento de modelos fracos.

3.6 Técnicas e Métricas de Avaliação

Neste subcapítulo, abordaremos as técnicas e métricas de avaliação de desempenho dos modelos de ML. É fundamental compreender como medimos a eficácia e a capacidade de generalização dos modelos que foram apresentados anteriormente na secção 3.5. Para isso, exploraremos as ferramentas e métricas utilizadas para avaliar a precisão, a robustez e a relevância dos resultados obtidos. Esta análise crítica permitirá comparar as *performances* dos modelos entre si, bem como orientar a otimização de hiperparâmetros na secção subsequente, 3.7.

- **Matriz de Confusão**

  A matriz de confusão [28] uma técnica que mostra o desempenho de um modelo de classificação em termos de verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN). A partir da matriz de confusão, é possível calcular outras métricas, como *Accuracy*, *Precision*, *Recall* e *F1-Score*.

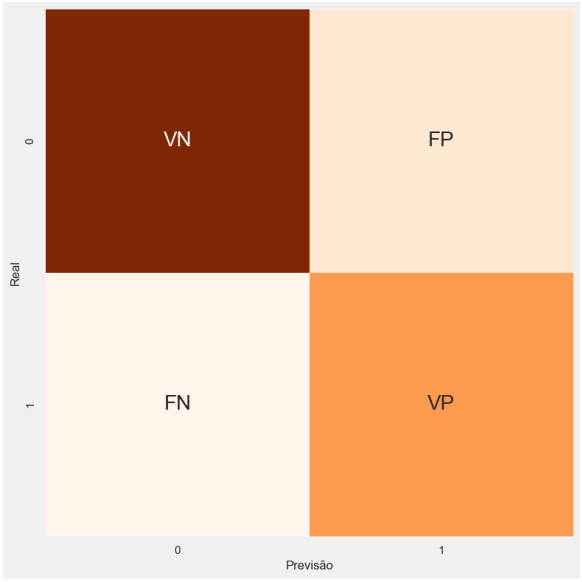


Figura 3.3: Aspeto de uma Matriz de Confusão

- ***Accuracy***

  A *Accuracy* [23] é uma métrica que mede a taxa geral de acertos do modelo, ou seja, a proporção de todas as previsões corretas (VP e VN) em relação ao total de previsões (VP, VN, FP e FN). É uma medida geralizada de desempenho. É expressa como:

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \tag{3.27}$$

  Onde:

  - VP - Verdadeiros Positivos
  - VN - Verdadeiros Negativos
  - FP - Falsos Positivos
  - FN - Falsos Negativos

- ***Precision***

  Precision [23] é a proporção de verdadeiros positivos (VP) em relação ao total de previsões positivas feitas pelo modelo. Em fórmula, é expressa como:

$$Precision = \frac{VP}{VP + FP} \tag{3.28}$$

Onde:

- VP - Verdadeiros Positivos
- FP - Falsos Positivos

- ***Recall***

Recall [23] é a proporção de verdadeiros positivos (VP) em relação ao total de casos verdadeiramente positivos no conjunto de dados. É expresso como:

$$Recall = \frac{VP}{VP + FN} \tag{3.29}$$

Onde:

- VP - Verdadeiros Positivos
- FN- Falsos Negativos

- ***F1-Score***

F1-Score [23] é a média harmónica entre o *precision* e o *recall*. Esta métrica equilibra as fragilidades entre essas duas métricas e é particularmente útil quando se deseja ter uma métrica única que leve em consideração ambos os falsos positivos e falsos negativos. É expresso como:

$$F1\text{–}Score = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \tag{3.30}$$

- **Especificidade**

A especificidade [23] mede a proporção de verdadeiros negativos (VN) em relação ao total de casos verdadeiramente negativos no conjunto de dados. É expressa como:

$$Especificidade = \frac{VN}{VN + FP} \tag{3.31}$$

Onde:

- VN - Verdadeiros Negativos
- FP - Falsos Positivos

- ***Receiver Operating Characteristic-Area Under the Curve* (ROC-AUC)**

  A métrica ROC-AUC [23] é uma medida usada para avaliar a capacidade discriminativa de um modelo de classificação binária. Esta métrica não é calculada por meio de uma equação matemática simples como muitas métricas tradicionais. Em vez disso, envolve a construção da curva ROC, que é criada traçando a curva da Taxa de Verdadeiros Positivos (TVP) no eixo vertical (y) e a Taxa de Falsos Positivos (TFP) no eixo horizontal (x) para diferentes valores de limiar de classificação. A ROC-AUC é a área sob a curva ROC, que varia de 0 a 1. Valores mais próximos de 1 indicam um melhor desempenho do modelo em distinguir entre as classes positiva e negativa.

- ***Average Precision***

  A *Average Precision* [23] é uma métrica comumente usada em problemas de classificação binária para avaliar o desempenho de um modelo de ML. Quantifica a qualidade das previsões do modelo, considerando a *precision* e o *recall* em vários pontos de limite de decisão. Expressa-se da seguinte forma:

$$AvaragePrecision = \sum_{n} (R_n - R_{n-1}) \cdot P_n \tag{3.32}$$

Onde:

- R_n é o*recall* no ponto n da curva *Precision-Recall*.
- P_n é a *precision* no ponto n da curva *Precision-Recall*.

- ***K-fold Cross Validation***

  A validação cruzada *k-fold* [23] é uma técnica usada para avaliar o desempenho de modelos de ML. Nesta técnica, o conjunto de dados é dividido em k subconjuntos (ou "dobras") de tamanho igual. O modelo é treinado k vezes, em cada treino usando $k-1$ dessas dobras como conjunto de treino e a dobra restante como conjunto de validação. Isto é repetido k vezes, de forma a que cada dobra seja utilizada como conjunto de validação uma vez.

  A principal vantagem da validação cruzada *k-fold* é que fornece uma estimativa mais robusta do desempenho do modelo, uma vez que todos os dados são usados para treino e validação em algum momento. Isto ajuda a evitar problemas de *overfitting* (ajustamento excessivo) ou *underfitting* (ajustamento insuficiente) e também variância do modelo. A métrica de desempenho é calculada como a média das métricas de desempenho obtidas em cada uma das k iterações.

3.7 Hiperparâmetros

Neste capítulo, iremos aprofundar a importância dos hiperparâmetros no contexto do nosso estudo. Os hiperparâmetros desempenham um papel fundamental na configuração e no desempenho dos modelos de ML. São cruciais para ajustar e personalizar os modelos, permitindo que se adaptem melhor aos dados e, consequentemente, melhorem as suas capacidades de previsão.

3.7.1 Otimização de Hiperparâmetros

A otimização de hiperparâmetros é uma etapa fundamental no processo de desenvolvimento de modelos de ML. Os hiperparâmetros são configurações que não são aprendidas pelo modelo durante o treino, mas têm um impacto significativo no seu desempenho. Desempenham um papel crucial ao determinar como o modelo se adapta aos dados, influenciando fatores como complexidade, generalização e capacidade de previsão.

A busca pelos melhores hiperparâmetros é um desafio, uma vez que existem várias combinações possíveis e não há uma solução única para todos os casos. No entanto, é essencial realizar essa otimização para garantir que os modelos atinjam o seu potencial máximo e possam fornecer resultados precisos e confiáveis.

É importante salientar que nem sempre é possível melhorar o desempenho dos modelos através da otimização de hiperparâmetros, uma vez que os valores padrão frequentemente já representam configurações adequadas para um determinado modelo e conjunto de dados.

3.7.2 Significado dos Hiperparâmetros

Antes de procedermos à seleção dos hiperparâmetros ótimos, é fundamental que compreendamos o significado e a influência de cada hiperparâmetro em relação a cada modelo. Iremos, portanto, realizar uma descrição detalhada de cada hiperparâmetro associado a cada modelo utilizado pela biblioteca *Scikit-learn*.

- **SVM**
  - **`C`**: Parâmetro escalar de regularização, que controla o equilíbrio entre maximizar a margem de separação entre classes e minimizar a classificação incorreta. Valores maiores de C permitem classificações incorretas no conjunto de treino, enquanto valores menores priorizam a maximização da margem.
  - **`kernel`**: Define o tipo de `kernel` a ser usado, como linear, polinomial ou radial. O `kernel` determina a forma da função de decisão que separa as classes no espaço de características.
  - **`gamma`**: Parâmetro escalar que influencia a forma da função de decisão. Um valor baixo de gamma gera uma função de decisão mais suave, enquanto um valor alto pode levar a uma função mais complexa que se ajusta aos dados de treino com mais detalhes. Um valor muito alto pode levar ao *overfitting*.
- **Regressão Logística**
  - **`penalty`**: Define a norma utilizada na penalização da regressão logística. Pode ser *"l1"* para penalização L1, *"l2"* para penalização L2 ou *"none"* para nenhuma penalização.
  - **`C`**: É o inverso do parâmetro escalar de regularização. Valores menores de C aumentam a força da regularização, tornando o modelo mais resistente ao *overfitting*.
  - **`solver`**: Define o algoritmo a ser utilizado na otimização da regressão logística. Opções incluem *"liblinear"* para conjuntos de dados pequenos e *"lbfgs"* (`kernel` Gaussiano) para conjuntos de dados maiores.
  - **`max_iter`**: Define o número máximo de iterações para a otimização.
- **Árvore de Decisão**
  - **`criterion`**: Define a métrica usada para medir a qualidade da divisão dos nós. Pode ser *"gini"* para o índice Gini ou *"entropy"* para a entropia de *Shannon*.
  - **`splitter`**: Define a estratégia utilizada para escolher a divisão em cada nó da árvore. Pode ser *"best"* para escolher a melhor divisão ou *"random"* para escolher uma divisão aleatória.

- **`max_depth`**: Escalar que limita a profundidade máxima da árvore de decisão. Isso ajuda a evitar árvores muito profundas que podem levar ao *overfitting*.
- **`max_features`**: Define o número máximo de recursos (variáveis) considerados ao procurar a melhor divisão em cada nó da árvore de decisão. Pode ser um número inteiro, um valor decimal (proporção), ou usar valores especiais como *"sqrt"* (raiz quadrada do número total de recursos), *"log2"* (logaritmo na base 2 do número total de recursos) ou *"None"* (usar todos os recursos disponíveis).
- **`min_samples_split`**: Define o número mínimo de amostras necessárias para realizar uma divisão em um nó interno da árvore.
- **`min_samples_leaf`**: Define o número mínimo de amostras necessárias para que uma folha (nó terminal) seja criada na árvore.

- **Florestas Aleatórias**
  - **`n_estimators`**: Define o número de árvores na floresta aleatória. Um maior número de árvores geralmente leva a um modelo mais robusto, mas aumenta o tempo de treino.
  - **`criterion`**: Define a métrica usada para medir a qualidade das divisões nas árvores individuais da floresta. Pode ser *"gini"* ou *"entropy"*.
  - **`max_depth`**: Limita a profundidade máxima das árvores individuais na floresta.
  - **`min_samples_split`**: Define o número mínimo de amostras necessárias para realizar uma divisão em um nó interno da árvore em cada árvore da floresta.
  - **`min_samples_leaf`**: Define o número mínimo de amostras necessárias para criar uma folha (nó terminal) em cada árvore da floresta.

- **KNN**
  - **`n_neighbors`**: Define o número de vizinhos a serem considerados para determinar a classe de um novo ponto de dados. Um valor maior de `n_neighbors` suaviza a fronteira de decisão.
  - **`weights`**: Define a estratégia de atribuição de pesos aos vizinhos. Pode ser *"uniform"* para pesos iguais ou *"distance"* para dar mais peso aos vizinhos mais próximos.
  - **`metric`**: Define a métrica de distância a ser usada para medir a proximidade entre pontos. Pode ser *"euclidean"*, *"manhattan"*, *"chebyshev"* ou outras métricas disponíveis.

- ***Naive Bayes***
  Não possui hiperparâmetros de ajuste.

3.7.3 Métodos de Otimização

Existem várias abordagens para a otimização de hiperparâmetros, sendo uma das mais simples o *Grid Search* [23]. Neste método, definimos um conjunto de valores possíveis para cada hiperparâmetro e testamos todas as combinações. Embora seja computacionalmente intensivo, é uma abordagem sistemática e eficaz.

Outra técnica é a otimização*Bayesiana* [23], que utiliza um processo de otimização probabilístico para encontrar os melhores hiperparâmetros. Esta abordagem é especialmente útil quando se lida com um grande espaço de hiperparâmetros e recursos computacionais limitados.

Decidimos utilizar o *Grid Search* como método de otimização de hiperparâmetros, aproveitando a capacidade de processamento da nossa máquina e o desejo de explorar todas as combinações possíveis de hiperparâmetros. O nosso objetivo é garantir que escolheremos a configuração ideal para os nossos modelos ao percorrermos minuciosamente este espaço de possibilidades. Usaremos a nossa base de dados inteira para este método de otimização.

3.7.4 Escolhas de Hiperparâmetros

Na figura 3.1, apresentamos na primeira coluna os hiperparâmetros padrão para cada modelo de classificação e nas segunda e terceira colunas, exibimos os hiperparâmetros otimizados para Clientes Particulares e Corporativos, respectivamente.

Tabela 3.1: Hiperparâmetros dos Modelos

Hiperparâmetros	*Standard*	Clientes Particulares	Clientes Corporativos
SVM			
C	1	1	0,9
kernel	'rbf'	'rbf'	'rbf'
gamma	'scale'	1	1
Regressão Logística			
penalty	'l2'	'none'	'l1'
C	1	0.0001	1.6238
solver	'lbfgs'	'lbfgs'	'liblinear'
max_iter	100	100	100
Árvore de Decisão			
criterion	'gini'	'entropy'	'gini'
splitter	'best'	'best'	'best'
max_depth	-	20	30
max_features	'none'	'sqrt'	'sqrt'
min_samples_split	2	5	10
min_samples_leaf	1	1	1
Florestas Aleatórias			
n_estimators	100	900	100
criterion	'gini'	'gini'	'gini'
max_depth	-	-	-
min_samples_split	2	2	2
min_samples_leaf	1	1	1
KNN			
n_neighbors	5	1	3
weights	'uniform'	'uniform'	'uniform'
metric	'minkowski'	'minkowski	'minkowski'

4

Resultados e Discussão

4.1 Resultados

Neste capítulo, iremos apresentar em detalhe os resultados da nossa análise dos modelos de classificação e modelos *Ensemble Learning*. Ao longo das próximas secções, iremos explorar os resultados iniciais, a avaliação da robustez, a otimização de hiperparâmetros, as matrizes de confusão e a importância das características. Estes resultados são fundamentais para compreender o desempenho e a eficácia dos modelos no nosso estudo.

4.1.1 Clientes Particulares

Neste início da nossa análise, concentrar-nos-emos nos clientes particulares, uma vez que representam 80% da nossa base de dados. Além disso, dispomos de um conjunto mais rico de variáveis para os clientes particulares em comparação com os clientes corporativos. Esta disponibilidade de dados mais abrangente alimenta a expectativa de que os modelos exibirão um desempenho aprimorado ao lidar com este segmento de clientes.

4.1.1.1 Modelos de Classificação

Nesta secção, vamos começar a nossa análise exploratória dos modelos de classificação mais amplamente utilizados na literatura.

4.1.1.1.1 Avaliação de Robustez

Como descrito em 3.6, o uso do *Cross Validation* com a técnica *K- Fold*, é essencial para controlar a variância dos modelos e evitar *overfitting*. Isto ajuda a garantir que os modelos sejam capazes de generalizar bem para novos dados, em vez de se ajustarem excessivamente aos dados de treino específicos. A validação cruzada *k-fold* fornece uma estimativa mais robusta do desempenho do modelo, uma vez que valida o modelo em múltiplos conjuntos de treino diferentes.

Nesta avaliação de robustez, optámos por utilizar a métrica ROC-AUC uma vez que esta métrica se adapta melhor a contextos com classes não balanceadas.

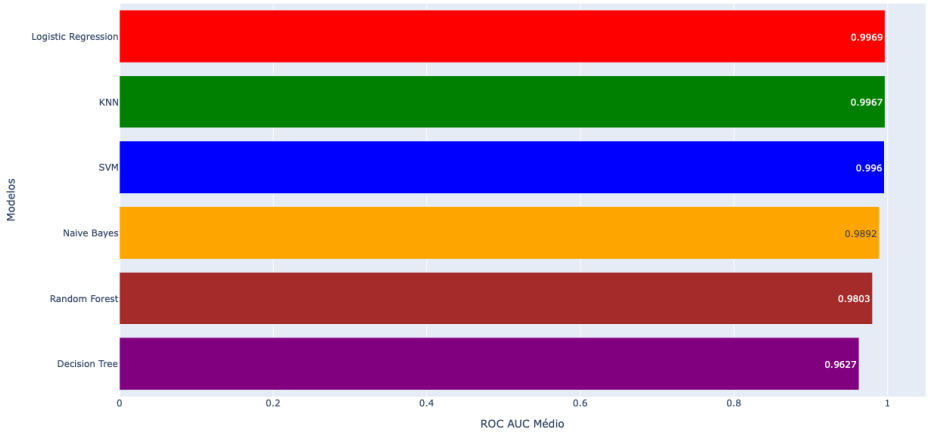


Figura 4.1: Gráfico de barras de Modelos de Classificação para Clientes Particulares

A partir da figura 4.1, podemos notar que os valores médios de ROC-AUC obtidos por meio da validação cruzada refletem a capacidade de generalização dos modelos em relação ao conjunto de treino.

Agora, vamos apresentar um gráfico de velas para uma visualização mais detalhada das estatísticas do ROC-AUC, incluindo a identificação de possíveis *outliers*.

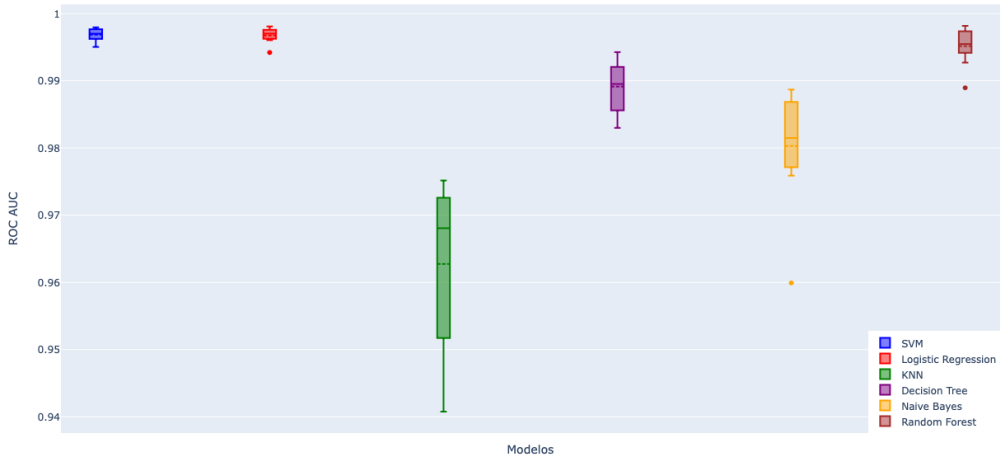


Figura 4.2: Gráfico de velas de Modelos de Classificação para Clientes Particulares

Pode-se observar que, na figura 4.2, entre os modelos avaliados, apenas um deles demonstra um desvio padrão maior em relação aos outros. Além disso, identificamos a presença de três *outliers*. Contudo, graças ao uso do *cross-validation*, conseguimos mitigar os efeitos de conjuntos de treino adversos, garantindo uma avaliação mais robusta e confiável.

4.1.1.1.2 *Performance* dos Modelos

Após avaliarmos a robustez do nosso modelo e validarmos o conjunto de treino, iremos agora apresentar, na tabela 4.1, o desempenho destes modelos no conjunto de teste.

Tabela 4.1: ROC-AUC e *Accuracy* para o conjunto de testes.

Modelos	ROC-AUC	*Accuracy*
SVM	0,9266	0,9912
Regressão Logística	0,8906	0,988
Árvore de Decisão	0,9332	0,9912
Florestas Aleatórias	0,9438	0,9927
KNN	0,8745	0,9855
Naive Bayes	0,911	0,9833

Embora as métricas de avaliação geral, como ROC-AUC e *Accuracy*, forneçam uma visão abrangente do desempenho dos modelos, é igualmente essencial compreender como esses modelos realizam previsões para cada classe individualmente. Essa análise proporciona uma visão mais detalhada e esclarecedora do comportamento dos modelos em relação a cada classe específica.

Tabela 4.2: *Precision*, *Recall*, *F1-Score* e Especificidade para o conjunto de testes.

Modelos	*Precision*	*Recall*	*F1-Score*	Especificidade
SVM	0,9575	0,8552	0,9035	0,9981
Regressão Logística	0,9619	0,7828	0,8631	0,9984
Árvore de Decisão	0,9438	0,8690	0,9048	0,9974
Florestas Aleatórias	0,9556	0,8867	0,9214	0,9979
KNN	0,9356	0,7517	0,8337	0,9974
Naive Bayes	0,8253	0,831	0,8282	0,9911

No anexo I, é possível visualizar a curva ROC, tabelado em 4.1 e *Precision vs. Recall*, tabelado em 4.2.

Para uma compreensão mais detalhada das métricas em 4.2, é benéfico considerar a matriz de confusão, apresentado no anexo II. Essa matriz fornece valores concretos que representam os acertos reais e previstos para cada modelo, permitindo uma análise mais aprofundada do desempenho dos modelos em cada classe.

4.1.1.1.3 Otimização de Hiperparâmetros

Após a avaliação dos modelos nos conjuntos de teste, iremos investigar a otimização dos hiperparâmetros. Esta análise visa determinar se ajustar criteriosamente as configurações dos modelos pode conduzir a melhorias substanciais no desempenho. Os hiperparâmetros são os determinados em 3.7.1.

Analogamente à estrutura de 4.1.1.1.2, nas tabelas 4.3 e 4.4, temos os desempenhos dos modelos otimizados no conjunto de teste. Daqui em diante, Modelos* refere-se a modelos com hiperparâmetros otimizados.

Tabela 4.3: ROC-AUC e *Accuracy* para o conjunto de testes.

Modelos*	ROC-AUC	*Accuracy*
SVM	0,9298	0,991
Regressão Logística	0,9435	0,9922
Árvore de Decisão	0,9522	0,9932
Florestas Aleatórias	0,9505	0,993
KNN	0,9101	0,9878

Tabela 4.4: *Precision*, *Recall*, *F1-Score* e Especificidade para o conjunto de testes.

Modelos*	*Precision*	*Recall*	*F1-Score*	Especificidade
SVM	0,947	0,8621	0,9025	0,9975
Regressão Logística	0,9451	0,8897	0,9165	0,9974
Árvore de Decisão	0,9495	0,9069	0,9277	0,9975
Florestas Aleatórias	0,9493	0,9034	0,9258	0,9975
KNN	0,9157	0,8241	0,8675	0,9961

No anexo III, é possível visualizar a curva ROC, tabelado em 4.3 e *Precision vs. Recall*, tabelado em 4.4, assim como as matrizes de confusão, em IV.

4.1.1.2 Modelos *Ensemble Learning*

Após a melhoria do desempenho dos nossos modelos de classificação por meio da otimização dos hiperparâmetros, estamos agora preparados para explorar estratégias avançadas visando aprimorar ainda mais nossa capacidade preditiva. Esta etapa envolve a incorporação de modelos de *Ensemble Learning*. Notavelmente, esses modelos *Ensemble* tendem a ser mais robustos e menos suscetíveis a *overfitting*, reduzindo a necessidade de uma análise de robustez intensiva quanto a dos modelos de classificação individuais.

4.1.1.2.1 *Performance dos Modelos*

Analogamente à estrutura de 4.1.1.1.3, nas tabelas 4.5 e 4.6, temos os desempenhos dos modelos *Ensenble Learning* no conjunto de teste.

Tabela 4.5: ROC-AUC e *Accuracy* para o conjunto de testes.

Modelos	ROC-AUC	*Accuracy*
Bagged SVM	0,9612	0,9947
Bagged Regressão Logística	0,954	0,9935
Bagged Árvore de Decisão	0,9596	0,9948
Bagged Florestas Aleatórias	0,9578	0,9945
Bagged KNN	0,9489	0,993
Bagged Naive Bayes	0,9108	0,9828
AdaBoost	0,9566	0,9943
SGDBoost	0,9614	0,995
XGBoost	0,9613	0,9948

Tabela 4.6: *Precision*, *Recall*, *F1-Score* e Especificidade para o conjunto de testes.

Modelos	*Precision*	*Recall*	*F1-Score*	Especificidade
Bagged SVM	0,964	0,9241	0,9437	0,9982
Bagged Regressão Logística	0,9531	0,9103	0,9312	0,9977
Bagged Árvore de Decisão	0,9709	0,9207	0,9451	0,9986
Bagged Florestas Aleatórias	0,9673	0,9172	0,9416	0,9984
Bagged KNN	0,9526	0,9	0,9255	0,9977
Bagged Naive Bayes	0,8169	0,831	0,8239	0,9905
AdaBoost	0,9204	0,9172	0,9188	0,996
SGDBoost	0,971	0,9241	0,947	0,9986
XGBoost	0,9675	0,9241	0,9453	0,9984

No anexo V, é possível visualizar a curva ROC, tabelado em 4.5 e *Precision vs. Recall*, tabelado em 4.6, assim como as matrizes de confusão, em VI.

4.1.2 Clientes Coorporativos

À medida que nos voltamos para a análise dos clientes corporativos, observamos que esta categoria representa os restantes 20% da nossa base de dados. A diminuição no número de dados e variáveis pode potencialmente afetar a capacidade dos nossos modelos em prever com a mesma precisão que no caso dos clientes particulares.

4.1.2.1 Modelos de Classificação

Nesta secção, vamos começar a nossa análise exploratória dos modelos de classificação mais amplamente utilizados na literatura. Seguiremos exatamente a mesma estrutura do capítulo 4.1.1.

4.1.2.1.1 Avaliação de Robustez

À semelhança de 4.1.1.1.1, continuaremos a utilizar a métrica ROC-AUC para avaliar a robustez e validar o desempenho dos nossos modelos no conjunto de treino.

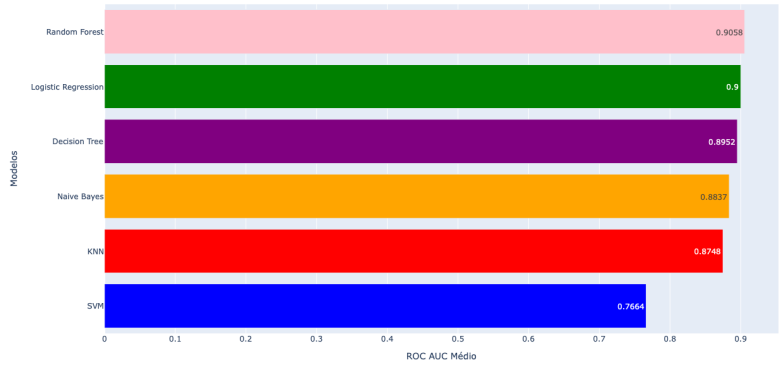


Figura 4.3: Gráfico de barras para Modelos de Classificação para Clientes Corporativos

A partir da figura 4.3, como mencionado em 4.1.2, a diminuição observada no ROC-AUC médio dos modelos pode ser um indicativo de que poderemos obter uma menor robustez nos modelos e, consequentemente, uma menor *performance* no conjunto de teste.

Vamos agora fazer uma análise estatística baseada em diagramas de velas para examinar em detalhe a diminuição no ROC-AUC médio em relação ao conjunto de treino.

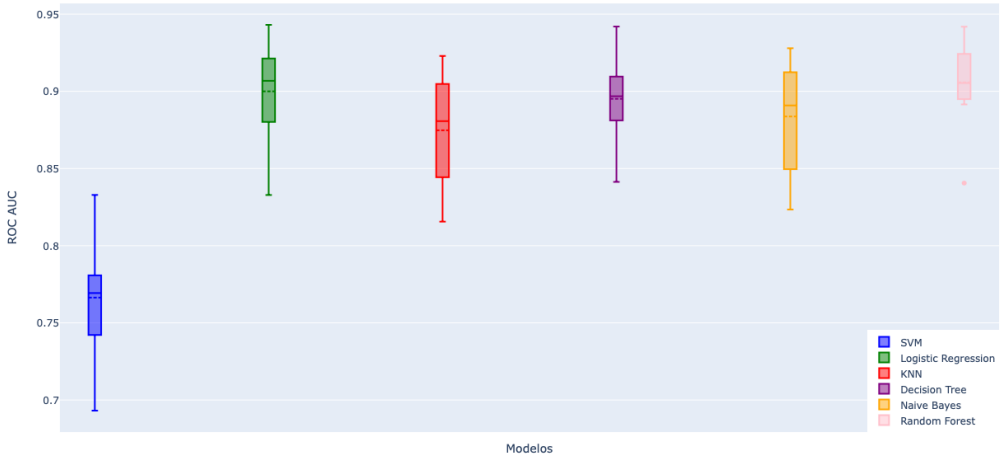


Figura 4.4: Gráfico de velas para Modelos de Classificação para Clientes Corporativos

Ao contrário do cenário anterior, em 4.1.1.1.1, observamos, na figura 4.4, que apenas um dos modelos não apresenta um desvio padrão significativo. Nota-se, também, que a distância entre o ROC-AUC mínimo e máximo é consideravelmente maior e temos a presença de um *outlier*. No entanto, ao empregarmos esta técnica, asseguramos que o modelo utilize o conjunto de treino ótimo, garantindo, assim, a robustez dos modelos.

4.1.2.1.2 *Performance* dos Modelos

Após avaliarmos a robustez dos nossos modelos e validarmos o conjunto de treino, iremos agora apresentar, na tabela 4.7, e 4.8, o desempenho destes modelos no conjunto de teste.

Tabela 4.7: ROC-AUC e *Accuracy* para o conjunto de testes.

Modelos	ROC-AUC	*Accuracy*
SVM	0,8456	0,987
Regressão Logística	0,7436	0,98
Árvore de Decisão	0,8513	0,986
Florestas Aleatórias	0,8649	0,9885
KNN	0,8392	0,9865
Naive Bayes	0,8577	0,9865

Tabela 4.8: *Precision*, *Recall*, *F1-Score* e Especificidade para o conjunto de testes.

Modelos	*Precision*	*Recall*	*F1-Score*	Especificidade
SVM	0,9643	0,6923	0,806	0,999
Regressão Logística	1	0,4872	0,6552	1
Árvore de Decisão	0,9167	0,7051	0,7971	0,9973
Florestas Aleatórias	0,9661	0,7308	0,8321	0,999
KNN	0,9636	0,6795	0,797	0,999
Naive Bayes	0,918	0,7179	0,8058	0,9974

No anexo VII, é possível visualizar a curva ROC, tabelado em 4.7 e *Precision vs. Recall*, tabelado em 4.8, assim como as matrizes de confusão, em VIII.

4.1.2.1.3 Otimização de Hiperparâmetros

Analogamente à estrutura de 4.1.1.1.3, nas tabelas 4.9 e 4.10, temos os desempenhos dos modelos otimizados no conjunto de teste.

Tabela 4.9: ROC-AUC e *Accuracy* para o conjunto de testes.

Modelos*	ROC-AUC	*Accuracy*
SVM	0,8392	0,9865
Regressão Logística	0,8264	0,9855
Árvore de Decisão	0,8649	0,9885
Florestas Aleatórias	0,8649	0,9885
KNN	0,852	0,9875

Tabela 4.10: *Precision*, *Recall*, *F1-Score* e Especificidade para o conjunto de testes.

Modelos*	*Precision*	*Recall*	*F1-Score*	Especificidade
SVM	0,9636	0,6795	0,797	0,999
Regressão Logística	0,9623	0,6538	0,7786	0,999
Árvore de Decisão	0,9661	0,7308	0,8321	0,999
Florestas Aleatórias	0,9661	0,7308	0,8321	0,999
KNN	0,9649	0,7051	0,8148	0,999

Em IX, é possível visualizar a curva ROC, tabelado em 4.9 e *Precision vs. Recall*, tabelado em 4.10, assim como as matrizes de confusão, em X.

4.1.2.2 Modelos *Ensemble Learning*

À semelhança de 4.1.1.2, a intenção é melhorar a *performance* dos modelos otimizados, 4.1.2.1.3.

4.1.2.2.1 *Performance dos Modelos*

Analogamente à estrutura de 4.1.2.1.3, nas tabelas 4.11 e 4.12, temos os desempenhos dos modelos *Ensenble Learning* no conjunto de teste.

Tabela 4.11: ROC-AUC e *Accuracy* para o conjunto de testes.

Modelos	ROC-AUC	*Accuracy*
Bagged SVM	0,859	0,989
Bagged Regressão Logística	0,852	0,9875
Bagged Árvore de Decisão	0,8526	0,9885
Bagged Florestas Aleatórias	0,859	0,989
Bagged KNN	0,859	0,989
Bagged Naive Bayes	0,8577	0,9865
AdaBoost	0,852	0,9875
SGDBoost	0,8526	0,9885
XGBoost	0,859	0,989

Tabela 4.12: *Precision*, *Recall*, *F1-Score* e Especificidade para o conjunto de testes.

Modelos	*Precision*	*Recall*	*F1-Score*	Especificidade
Bagged SVM	1	0,7179	0,8358	0,9974
Bagged Regressão Logística	0,9649	0,7051	0,8148	0,9974
Bagged Árvore de Decisão	1	0,7051	0,8271	0,9974
Bagged Florestas Aleatórias	1	0,7179	0,8358	0,9974
Bagged KNN	1	0,7179	0,8358	0,9974
Bagged Naive Bayes	0,918	0,7179	0,8058	0,9974
AdaBoost	0,9649	0,7051	0,8148	0,9974
SGDBoost	1	0,7051	0,8271	0,9974
XGBoost	1	0,7179	0,8358	0,9974

No anexo XI, é possível visualizar a curva ROC, tabelado em 4.11 e *Precision vs. Recall*, tabelado em 4.12, assim como as matrizes de confusão, em XII.

4.2 Discussão de Resultados

Neste capítulo, após a apresentação dos resultados em 4.1, iremos realizar uma análise criteriosa de todas as métricas disponíveis para determinar quais delas são mais relevantes no contexto do nosso estudo. Esta análise será conduzida separadamente para os clientes particulares e corporativos, levando em consideração as particularidades de cada grupo. O objetivo é identificar as métricas que melhor refletem o desempenho dos modelos em relação aos nossos objetivos e requisitos específicos.

4.2.1 Escolha dos Melhores Modelos

No que diz respeito às métricas que avaliam a *performance* geral dos modelos, a escolha natural recai sobre o ROC-AUC. Esta métrica demonstra ser especialmente útil quando lidamos com conjuntos de dados desequilibrados, como descrito em 3.6, ao contrário da métrica de *Accuracy*. No entanto, ao avaliar medidas mais específicas para cada classe, como *Recall*, *Precision*, *F1-Score* e Especificidade, percebemos a sua relevância, principalmente no contexto de deteção de casos suspeitos.

É crucial considerar o propósito e os objetivos deste trabalho, que, embora seja uma tese de dissertação, foi realizado com uma componente prática muito relevante uma vez que este se encontra alinhado com a realidade atual das Instituições Financeiras. O objetivo principal é detectar situações de risco de BC/FT. Nesse âmbito, não é relevante ter modelos que façam previsões excepcionais para situações que não sejam fraudulentas ou de baixo risco. O foco principal reside em identificar o maior número possível de casos fraudulentos.

É importante notar que nem o *Recall* nem a *Precision* têm uma relação direta e simples com o ROC-AUC. Em cenários em que o *Recall* é elevado, pode acontecer que a *Precision* seja baixa, o que resulta num ROC-AUC relativamente baixo. Por outro lado, em situações onde a *Precision* é alta, o *Recall* pode ser baixo, levando a um ROC-AUC relativamente alto. Assim, o ROC-AUC, o *Recall* e a *Precision* são métricas que refletem diferentes aspetos do desempenho de um modelo de classificação e devem ser avaliados com base nos objetivos específicos do problema. Portanto, as métricas de *Precision* e *Recall* são as que melhor se alinham com os objetivos da empresa e, consequentemente, deste estudo.

Entre essas métricas, surge a questão de qual delas é mais importante: um modelo que acerta mais nas suas previsões ou um modelo que acerta mais nos casos reais? Nesse contexto, faz mais sentido priorizar um modelo que acerte mais nos casos reais no conjunto de dados reais do que um modelo que acerte mais nos casos previstos no conjunto dos dados reais.

Dessa forma, estabelecemos o *Recall* como a métrica principal, seguido por *Precision* e, num contexto mais amplo, o ROC-AUC, que ainda é uma métrica relevante para avaliar o desempenho geral dos modelos. Essa abordagem reflete o nosso compromisso em encontrar e identificar eficazmente casos de BC/FT, que são cruciais para os interesses da empresa e para o sucesso deste estudo.

Na seleção dos modelos, iremos priorizar os três modelos que apresentam o melhor desempenho no *Recall*. Em situações de empate entre modelos com desempenho semelhante em termos de *Recall*, o critério de desempate será o *Precision*. Caso ainda persista um empate após considerar o *Precision*, recorreremos à métrica mais abrangente, o ROC-AUC, para efetuar a decisão final. Havendo uma igualdade entre estas três métricas, os modelos terão a mesma classificação. Esta abordagem garante que os modelos selecionados tenham um alto poder de identificação de casos de BC/FT, ao mesmo tempo que mantêm um nível aceitável de precisão nas suas previsões.

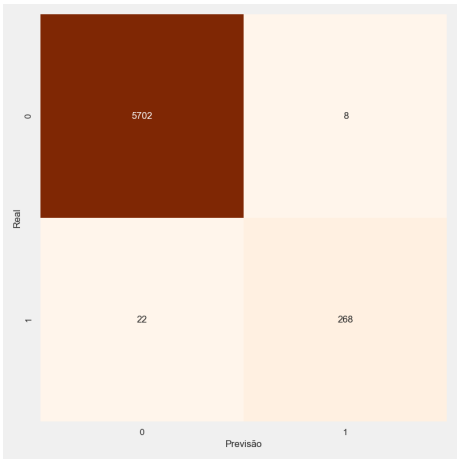
4.2.2 Clientes Particulares

Fazendo uma análise às tabelas 4.1, 4.2, 4.3 e 4.4, podemos concluir, resumido na tabela 4.13, que o modelo com melhor *performance*, segundo o nosso critério, é o G*radient Boost*, seguido do *XGBoost* e do *Bagged SVM*.

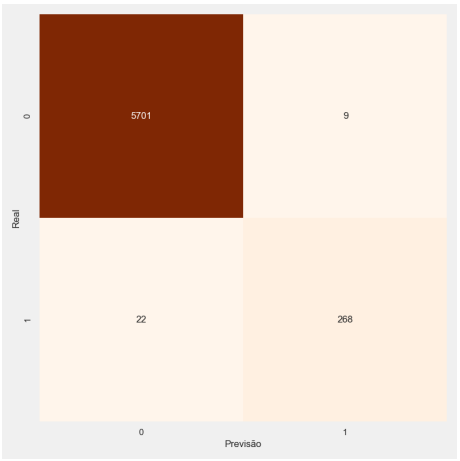
Tabela 4.13: *Precision*, *Recall* e ROC-AUC para o conjunto de testes.

Modelos	*Precision*	*Recall*	ROC-AUC
SGDBoost	0,971	0,9241	0,9614
XGBoost	0,9675	0,9241	0,9613
Bagged SVM	0,964	0,9241	0,9612

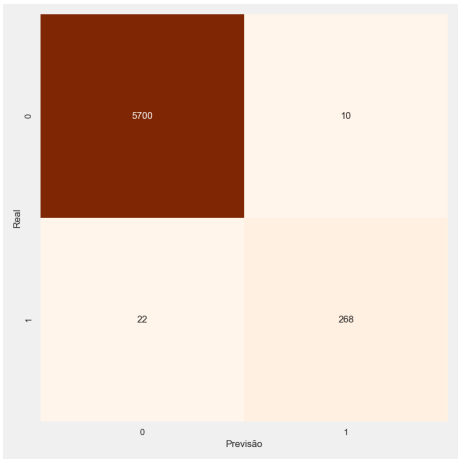
Vamos agora recorrer à matriz de confusão, indexado em VI, para obter uma visualização mais detalhada e compreender o significado destas métricas no contexto dos casos reais.



(a) *SGDBoost*



(b) *XGBoost*



(c) *Bagged* SVM

Figura 4.5: Matrizes de Confusão dos três melhores modelos ML para Clientes Particulares

Na figura 4.5, verificamos que estes modelos efetuam previsões corretas para 268 clientes particulares com alto risco de BC/FT, num total de 290 identificados na nossa base de dados. A única diferença notável ocorre nos casos de falsos positivos, embora a discrepância seja mínima. Os três modelos apresentam um comportamento semelhante nesse aspeto.

É igualmente crucial compreender a influência relativa de cada variável nos modelos e identificar diferenças significativas entre elas, uma vez que isso pode fornecer *insights* valiosos para a interpretação e otimização dos resultados. Ao escolher, aleatoriamente, dois dos três melhores modelos poderemos observar se existem diferença de importância de variáveis para cada modelo.

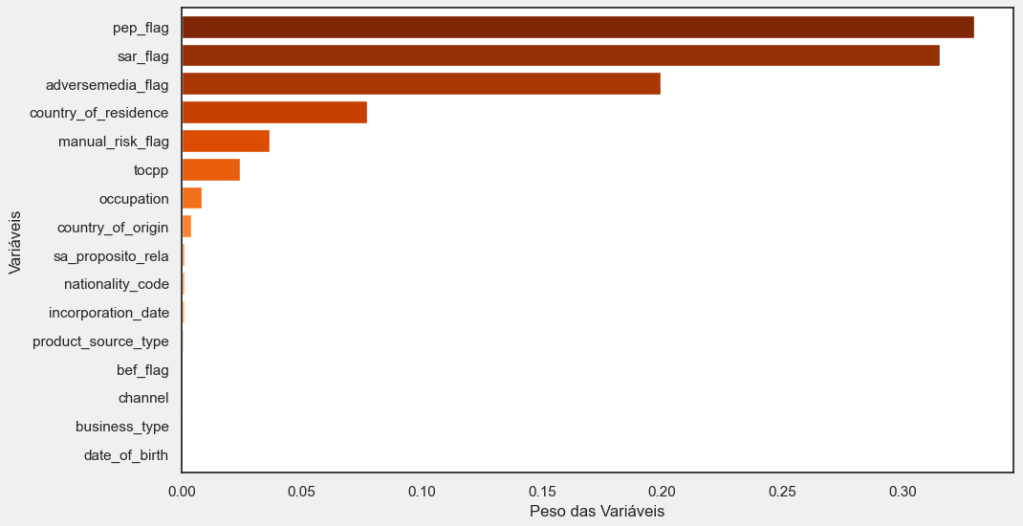


Figura 4.6: Peso de Variáveis no modelo *XGBoost*

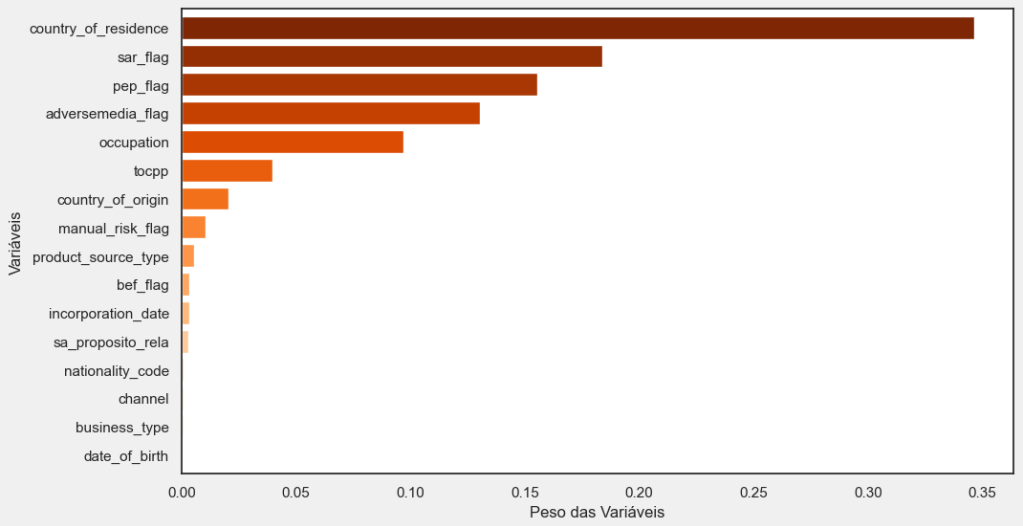


Figura 4.7: Peso de Variáveis no modelo *SGDBoost*

Nas figuras 4.6 e 4.7, é perceptível que, embora tanto o *XGBoost* quanto o *SGDBoost* apresentem métricas de desempenho muito semelhantes, observa-se que cada um desses modelos atribui pesos distintos às variáveis. No entanto, podemos concluir que variáveis como *sar_flag*, *adversemedia_flag*, *country_of_residence* e *pep_flag* têm sempre uma grande importância para estes modelos. Faz sentido que estas variáveis possam ter mais peso do que as restantes, dado que *sar_flag* e *adversemedia_flag* são são clientes reportados às autoridades e com notícias adversas publicadas nos *media* e, nesse caso, é relevante redobrar a análise risco. Além disso, *pep_flag* informa-nos se estamos a lidar com pessoas politicamente expostas, o que aumenta consideravelmente o risco associado a essas pessoas dado que, é uma exigência regulamentar na maioria dos países. Por último, *country_of_residence* fornece informações sobre o local de residência da pessoa, que realmente tem um impacto significativo nos resultados destes modelos. Esta variável poderá, fortemente, indicar que

estes tipos de modelos, para este tipo de clientes, podem ter comportamentos e previsões distintas. Por exemplo, classificando um cliente de alto risco acabar por ser classificado como baixo risco.

4.2.3 Clientes Corporativos

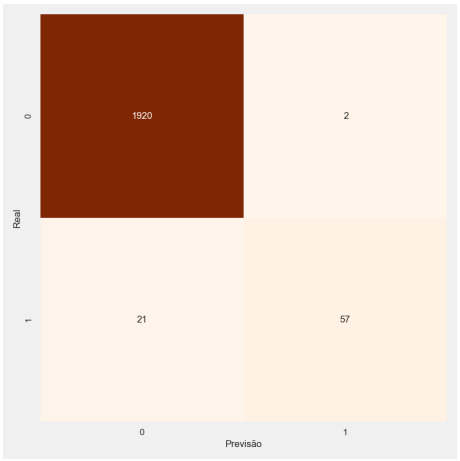
Fazendo uma análise às tabelas 4.7, 4.8, 4.9 e 4.10, podemos concluir, resumido na tabela 4.14, que os modelos com melhor *performance*, segundo o nosso critério, são as Árvores de Decisão e as Florestas Aleatórias, seguido de *Bagged SVM*, *Bagged Florestas Aleatórias*, *Bagged* KNN e *XGBoost*, finalizando com *Bagged Naive Bayes*.

Tabela 4.14: *Precision*, *Recall* e ROC-AUC para o conjunto de testes.

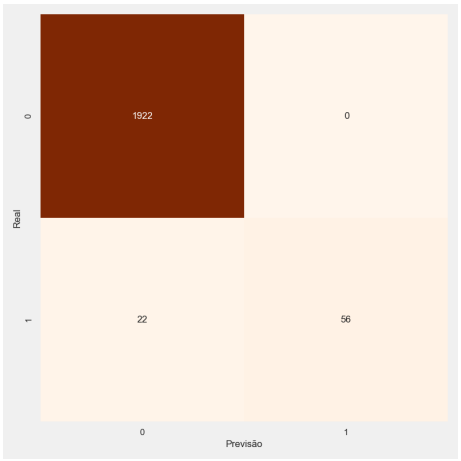
Modelos	*Precision*	*Recall*	ROC-AUC
Florestas Aleatórias*	0,9661	0,7308	0,8649
Árvore de Decisão*	0,9661	0,7308	0,8649
Bagged SVM	1	0,7179	0,859
Bagged Florestas Aleatórias	1	0,7179	0,859
XGBoost	1	0,7179	0,859
Bagged KNN	1	0,7179	0,859
Bagged Naive Bayes	0,918	0,7179	0,8577

Florestas Aleatórias* e Árvore de Decisão* significam que estes modelos têm os hiperparâmetros otimizados.

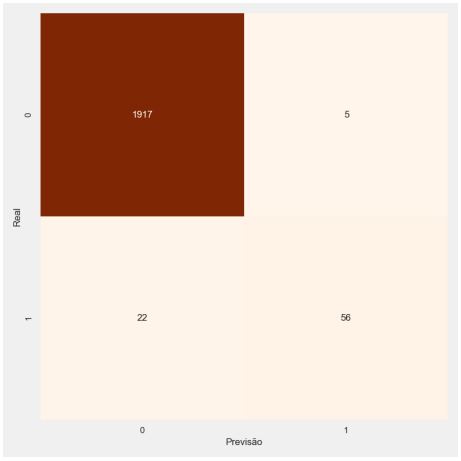
Semelhante aos clientes particulares, vamos recorrer à matriz de confusão, indexado em X e XII, para obter uma visualização mais detalhada e compreender o significado destas métricas no contexto dos casos reais.



(a) Florestas Aleatórias e Árvore de Decisão



(b) *Bagged* SVM, *Bagged* Florestas Aleatórias, *XGBoost* e *Bagged* KNN



(c) *Bagged Naive Bayes*

Figura 4.8: Matrizes de Confusão dos três melhores modelos ML para Clientes Corporativos

Na figura 4.8, verificamos que esses modelos efetuam previsões corretas para 57 clientes corporativos com alto risco de BC/FT, num total de 78 identificados na nossa base de dados. É de notar que os modelos em (b) acertam a totalidade de clientes corporativos não fraudulentos identificados na nossa base de dados, assim como, de entre as previsões feitas, acertam na sua totalidade.

Iremos, novamente, compreender a influência relativa de cada variável nos modelos e identificar diferenças significativas entre elas, uma vez que isso pode fornecer *insights* valiosos para a interpretação e otimização dos resultados. Com apenas dois exemplos poderemos observar se existem diferença de importância de variáveis para cada modelo.

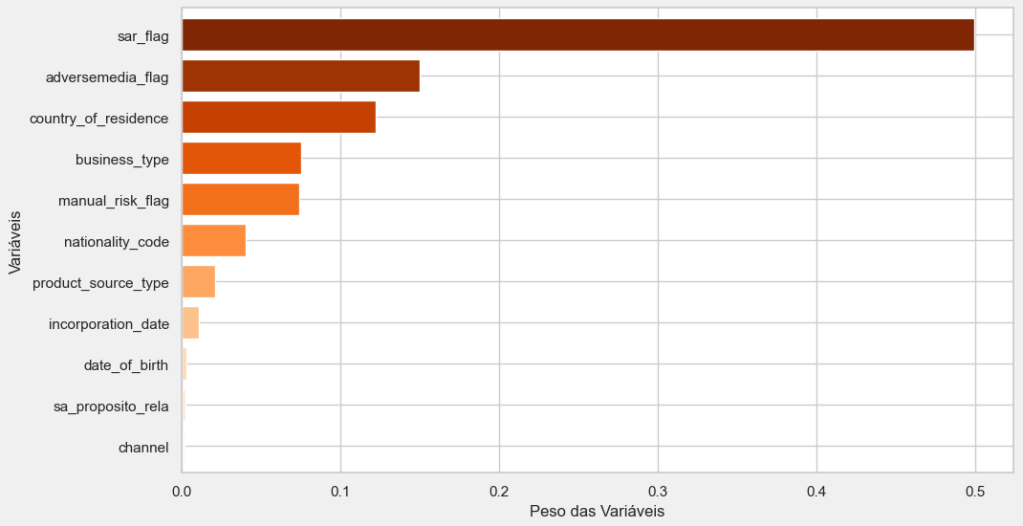


Figura 4.9: Peso de Variáveis no modelo de Florestas Aleatórias

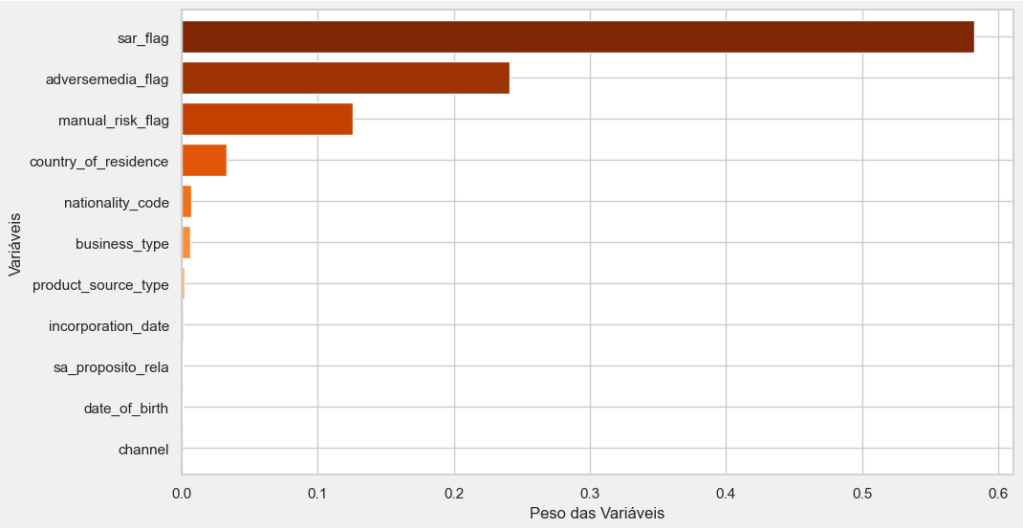


Figura 4.10: Peso de Variáveis no modelo *XGBoost*

Nas figuras 4.9 e 4.10, é perceptível que, embora tanto as Florestas Aleatórias quanto o *XGBoost* apresentam métricas de desempenho muito semelhantes, observa-se que cada um desses modelos atribui pesos distintos às variáveis. Contudo, podemos concluir que variáveis como *sar_flag*, *adversemedia_flag*, *country_of_residence* e *manual_risk_flag* têm sempre uma grande importância para estes modelos. É interessante notar que, à semelhança dos clientes particulares, *sar_flag* e *adversemedia_flag* acabam por ter uma grande preponderância para os clientes corporativos, sendo mesmo as que têm maior peso. Além disso, *country_of_residence* também tem uma importância considerável, semelhante aos clientes particulares. Isso sugere novamente que variáveis de caráter geográfico desempenham um papel significativo nos modelos de ML.

4.2.4 Considerações

É fundamental realçar que os resultados obtidos para os clientes particulares e corporativos revelam diferenças significativas. No caso dos clientes particulares, os modelos *Ensemble Learning* demonstraram, de uma forma geral, um desempenho superior em comparação com os modelos de Classicação. Essa superioridade pode estar relacionada ao acesso a um maior volume de dados ou à presença de um maior número de variáveis nesse conjunto de dados específico. Os modelos *Ensemble Learning* parecem aproveitar esses fatores a seu favor, proporcionando uma classificação mais precisa para os clientes particulares.

No entanto, ao analisar os resultados para os clientes Corporativos, observamos que a diferença de desempenho entre os diversos modelos de Classificação e *Ensemble Learning* não é tão evidente. Isso sugere que, para os Clientes Corporativos, os modelos *Ensemble Learning* podem exigir um conjunto de dados ainda mais extenso ou um número adicional de variáveis para demonstrar melhorias significativas.

Foi possível observar, pelas figuras 4.6, 4.7, 4.9 e 4.10, que, tanto para clientes particulares como corporativos, as variáveis *sar_flag*, *adversemedia_flag* e *country_of_origin* são as que possuem maior peso nos modelos. Vale ressaltar que a variável *pep_flag*, exclusiva para clientes particulares, também demonstrou ter um peso significativo. Isso sugere a importância dessas variáveis na classificação de risco.

Ao analisar os melhores modelos para ambos os tipos de clientes, podemos concluir que a variável *country_of_origin* apresenta uma relevância notável. Um estudo futuro aprofundado poderia confirmar que, de fato, o local de residência tem um grande impacto na classificação de risco. Além disso, observamos que, para o mesmo tipo de cliente, diferentes modelos atribuem importâncias variadas a essa variável. Isso sugere a possibilidade de que existam modelos com melhor desempenho em diferentes países.

Além disso, embora este estudo se tenha concentrado na realização de testes exaustivos com diferentes tipos de modelos de Classificação e *Ensemble Learning*, um dos principais objetivos é determinar se o uso de modelos fundamentados em princípios matemáticos sólidos oferece ganhos relevantes em relação a modelos heurísticos. Os resultados apresentados nas figuras dos anexos I, III, V, VII, IX, XI demonstram consistentemente que, independentemente do modelo de ML utilizado e das suas configurações de hiperparâmetros, superam consistentemente os modelos heurísticos, assinalados pelo tracejado, em termos de desempenho.

Estes resultados fortalecem a conclusão de que a adoção de modelos de ML fundamentados em princípios matemáticos é a abordagem mais eficaz e confiável para a tarefa de classificação.

5

Conclusão e Trabalhos Futuros

A avaliação do risco no contexto do BC/FT é uma responsabilidade crucial para as Instituições Financeiras, dada a crescente regulamentação rigorosa. Este estudo investigou a eficácia de diferentes modelos de classificação e *Ensemble Learning* na identificação de riscos associados ao BC/FT em clientes particulares e corporativos.

No caso dos clientes particulares, os modelos *Ensemble Learning*, como o *SGDBoost* e o *XGBoost*, demonstraram um desempenho superior em relação aos modelos de classificação tradicionais. Isto sugere que esses modelos podem beneficiar de um maior volume de dados ou de um conjunto mais rico de variáveis. No entanto, para os clientes corporativos, a diferença de desempenho entre os modelos de Classificação e *Ensemble Learning* não foi tão evidente. Parece que os modelos *Ensemble Learning* podem precisar de conjuntos de dados ainda maiores ou de mais variáveis para mostrar melhorias significativas. Em resumo, os modelos de Classificação tradicionais podem ser mais adequados quando se lida com bases de dados corporativas com um número menor de variáveis.

Além disso, este estudo demonstrou consistentemente que os modelos de ML conferem uma ferramenta útil e eficaz na classificação de risco de BC/FT, independentemente das configurações de hiperparâmetros. Isto reforça a importância de adotar abordagens baseadas em princípios matemáticos sólidos para a classificação de riscos em BC/FT.

Conseguimos também evidenciar que, dentro do contexto dos dados baseados na tipologia de informação de uma Instituição Financeira localizada em Portugal, a otimização dos modelos por meio dos hiperparâmetros resulta num considerável aumento no desempenho desses modelos.

Em relação à importância das variáveis, podemos afirmar que, apesar dos modelos atribuírem pesos diferentes a cada variável, ainda é possível identificar um padrão de quais variáveis são mais importantes para os clientes particulares e corporativos.

Como trabalho futuro seria interessante explorar a análise de componentes principais (PCA). O PCA pode ajudar a identificar variáveis menos significativas, reduzindo o tempo de computação sem comprometer o desempenho dos modelos e, ao mesmo tempo, possibilitando a detecção de padrões mais robustos.

Além disso, investigar como os modelos *Ensemble Learning* se comportam com bases de dados maiores ou menores, bem como o impacto do número de variáveis, pode fornecer *insights* adicionais sobre o desempenho desses modelos em diferentes contextos.

Investigar se as nossas conclusões sobre a importância das variáveis de origem afetam o desempenho dos modelos, ou seja, se a utilização dos modelos de ML pode variar em desempenho dependendo da origem dos conjuntos de dados.

Outra linha de pesquisa interessante seria a exploração de abordagens em *Deep Learning* com redes neuroais, para avaliar como esses modelos se comparam aos métodos tradicionais em termos de precisão e eficácia na detecção de riscos de BC/FT.

Num cenário global onde as consequências económicas do BC/FT são significativas, este estudo contribui para uma compreensão mais profunda das abordagens eficazes na identificação de riscos. Apreciamos que, embora a busca por melhores métodos seja constante, a base matemática e o uso de modelos de ML emergem como elementos cruciais na mitigação eficaz dessas ameaças.

Bibliografia

[1] L. Breiman. «Bagging predictors». Em: *Machine Learning* 24 (2004), pp. 123–140. URL: https://api.semanticscholar.org/CorpusID:47328136 (ver p. 36).

[2] F. Butaru et al. «Risk and risk management in the credit card industry». Em: *Journal of Banking Finance* 72 (2016), pp. 218–239. ISSN: 0378-4266. DOI: https://doi.org/10.1016/j.jbankfin.2016.07.015. URL: https://www.sciencedirect.com/science/article/pii/S0378426616301340 (ver pp. 3, 7).

[3] P. Castellón González e J. D. Velásquez. «Characterization and detection of taxpayers with false invoices using data mining techniques». Em: *Expert Systems with Applications* 40.5 (2013), pp. 1427–1436. ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2012.08.051. URL: https://www.sciencedirect.com/science/article/pii/S0957417412010123 (ver pp. 3, 5).

[4] T.-H. Chen. «Do you know your customer? Bank risk assessment based on machine learning». Em: *Applied Soft Computing* 86 (2020), p. 105779. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2019.105779. URL: https://www.sciencedirect.com/science/article/pii/S1568494619305605 (ver p. 1).

[5] C. Cortes e V. Vapnik. «Support-vector networks». Em: *Machine learning* 20 (1995), pp. 273–297 (ver pp. 3, 7).

[6] D. R. Cox. «The regression analysis of binary sequences». Em: *Journal of the Royal Statistical Society: Series B (Methodological)* 20.2 (1958), pp. 215–232 (ver p. 30).

[7] N. Cristianini e E. Ricci. «Support Vector Machines». Em: *Encyclopedia of Algorithms*. Ed. por M.-Y. Kao. Boston, MA: Springer US, 2008, pp. 928–932. ISBN: 978-0-387-30162-4. DOI: 10.1007/978-0-387-30162-4_415. URL: https://doi.org/10.1007/978-0-387-30162-4_415 (ver pp. 3, 7, 33).

[8] S. Figini, F. Bonelli e E. Giovannini. «Solvency prediction for small and medium enterprises in banking». Em: *Decision Support Systems* 102 (2017), pp. 91–97. ISSN: 0167-9236. DOI: https://doi.org/10.1016/j.dss.2017.08.001. URL: https://www.sciencedirect.com/science/article/pii/S0167923617301446 (ver pp. 3, 7).

[9] D. Freedman, R. Pisani e R. Purves. «Statistics (international student edition)». Em: *Pisani, R. Purves, 4th edn. WW Norton & Company, New York* (2007) (ver p. 19).

[10] J. Friedman. «Stochastic Gradient Boosting». Em: *Computational Statistics Data Analysis* 38 (fev. de 2002), pp. 367–378. DOI: 10.1016/S0167-9473(01)00065-2 (ver p. 40).

[11] C. R. Harris et al. «Array programming with NumPy». Em: *Nature* 585.7825 (set. de 2020), pp. 357–362. DOI: 10.1038/s41586-020-2649-2. URL: https://doi.org/10.1038/s41586-020-2649-2 (ver p. 11).

[12] T. K. Ho. «Random decision forests». Em: *Proceedings of 3rd international conference on document analysis and recognition*. Vol. 1. IEEE. 1995, pp. 278–282 (ver p. 26).

[13] J. D. Hunter. «Matplotlib: A 2D graphics environment». Em: *Computing in Science & Engineering* 9.3 (2007), pp. 90–95. DOI: 10.1109/MCSE.2007.55 (ver p. 11).

[14] L. Keyan e Y. Tingting. «An improved support-vector network model for anti-money laundering». Em: *2011 Fifth International Conference on Management of e-Commerce and e-Government*. IEEE. 2011, pp. 193–196 (ver p. 3).

[15] A. E. Khandani, A. J. Kim e A. W. Lo. «Consumer credit-risk models via machine-learning algorithms». Em: *Journal of Banking Finance* 34.11 (2010), pp. 2767–2787. ISSN: 0378-4266. DOI: https://doi.org/10.1016/j.jbankfin.2010.06.001. URL: https://www.sciencedirect.com/science/article/pii/S0378426610002372 (ver pp. 3, 7).

[16] P. P. Kumar. *Machine Learning for Model Development in Market Risk*. 2018 (ver pp. 3, 7).

[17] E. A. Lopez-Rojas e S. Axelsson. «Money laundering detection using synthetic data». Em: *Annual workshop of the Swedish Artificial Intelligence Society (SAIS)*. Linköping University Electronic Press, Linköpings universitet. 2012 (ver pp. 3, 5).

[18] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/master/template.pdf (ver p. iii).

[19] L.-T. Lv, N. Ji e J.-L. Zhang. «A RBF neural network model for anti-money laundering». Em: *2008 International conference on wavelet analysis and pattern recognition*. Vol. 1. IEEE. 2008, pp. 209–215 (ver pp. 3, 4).

[20] W. McKinney et al. «Data structures for statistical computing in python». Em: *Proceedings of the 9th Python in Science Conference*. Vol. 445. Austin, TX. 2010, pp. 51–56 (ver p. 11).

[21] D. Mikkelsen, A. Pravdic e B. Richardson. *Flushing out the money launderers with better customer risk-rating models Dramatically improve detection rates by simplifying model architecture, fixing underlying data, and using machine-learning algorithms to identify high-risk behavior*. 2019 (ver pp. 1, 2).

[22] A. Mucherino, P. J. Papajorgji e P. M. Pardalos. «k-Nearest Neighbor Classification». Em: *Data Mining in Agriculture*. New York, NY: Springer New York, 2009, pp. 83–106. ISBN: 978-0-387-88615-2. DOI: 10.1007/978-0-387-88615-2_4. URL: https://doi.org/10.1007/978-0-387-88615-2_4 (ver p. 28).

[23] F. Pedregosa et al. «Scikit-learn: Machine Learning in Python». Em: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830 (ver pp. 11, 46–49, 52).

[24] E. Rivera, J. L. West e C. Suplee. «Addressing AML Regulatory Pressures by Creating Customer Risk Rating Models with Ordinal Logistic Regression». Em: 2015. URL: https://api.semanticscholar.org/CorpusID:52250804 (ver p. 6).

[25] R. E. Schapire. «Explaining adaboost». Em: *Empirical inference*. Springer, 2013, pp. 37–52 (ver p. 38).

[26] Y. Son, H. Byun e J. Lee. «Nonparametric machine learning models for predicting the credit default swaps: An empirical study». Em: *Expert Systems with Applications* 58 (2016), pp. 210–220 (ver pp. 3, 7).

[27] J. Tang e J. Yin. «Developing an intelligent data discriminating system of anti-money laundering based on SVM». Em: *2005 International Conference on Machine Learning and Cybernetics*. Vol. 6. 2005, 3453–3457 Vol. 6. DOI: 10.1109/ICMLC.2005.1527539 (ver pp. 3, 4).

[28] K. M. Ting. «Confusion Matrix». Em: *Encyclopedia of Machine Learning and Data Mining*. Ed. por C. Sammut e G. I. Webb. Boston, MA: Springer US, 2017, pp. 260–260. ISBN: 978-1-4899-7687-1. DOI: 10.1007/978-1-4899-7687-1_50. URL: https://doi.org/10.1007/978-1-4899-7687-1_50 (ver p. 45).

[29] S.-N. Wang e J.-G. Yang. «A money laundering risk evaluation method based on decision tree». Em: *2007 international conference on machine learning and cybernetics*. Vol. 1. IEEE. 2007, pp. 283–286 (ver pp. 3, 5).

[30] M. L. Waskom. «seaborn: statistical data visualization». Em: *Journal of Open Source Software* 6.60 (2021), p. 3021. DOI: 10.21105/joss.03021. URL: https://doi.org/10.21105/joss.03021 (ver p. 11).

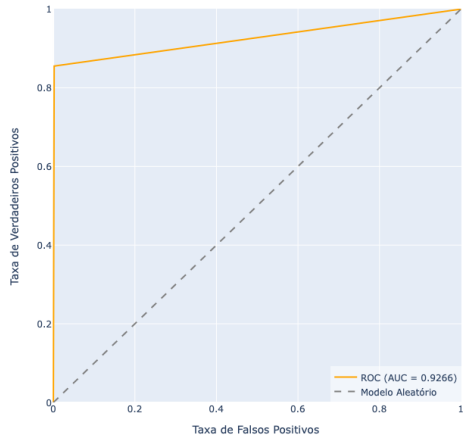
[31] G. I. Webb. «Naïve Bayes». Em: *Encyclopedia of Machine Learning*. Ed. por C. Sammut e G. I. Webb. Boston, MA: Springer US, 2010, pp. 713–714. ISBN: 978-0-387-30164-8. DOI: 10.1007/978-0-387-30164-8_576. URL: https://doi.org/10.1007/978-0-387-30164-8_576 (ver p. 30).

[32] R. Wirth e J. Hipp. «CRISP-DM: Towards a standard process model for data mining». Em: *Proceedings of the 4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining* (jan. de 2000) (ver p. 12).

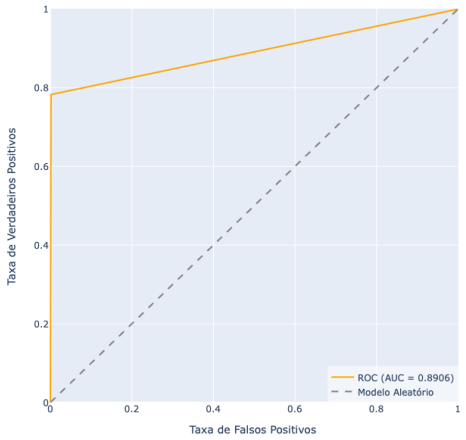
[33] X. Wu et al. «Top 10 algorithms in data mining». Em: *Knowledge and information systems* 14.1 (2008), pp. 1–37 (ver p. 23).

I

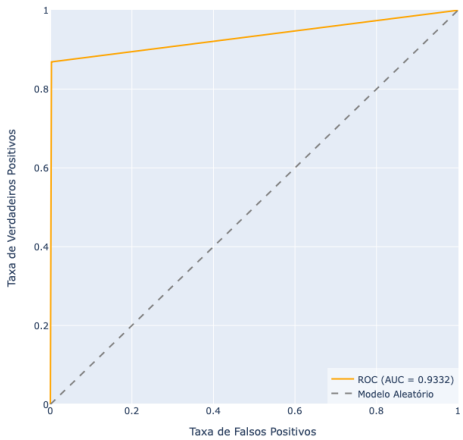
Curvas ROC-AUC e *Precision vs. Recall* para Modelos de Classificação de Clientes Particulares



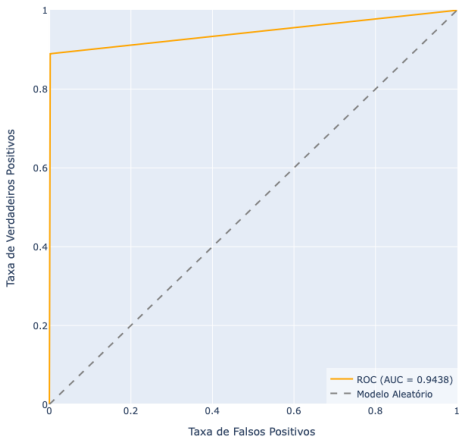
(a) SVM



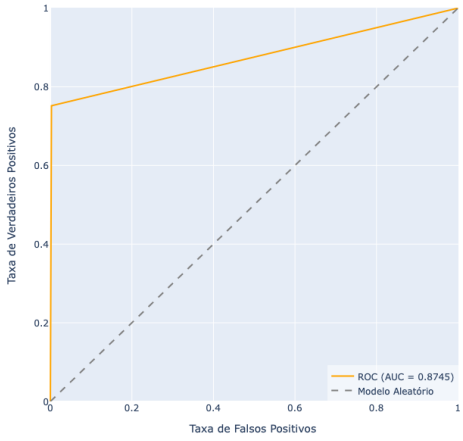
(b) Regressão Logísitca



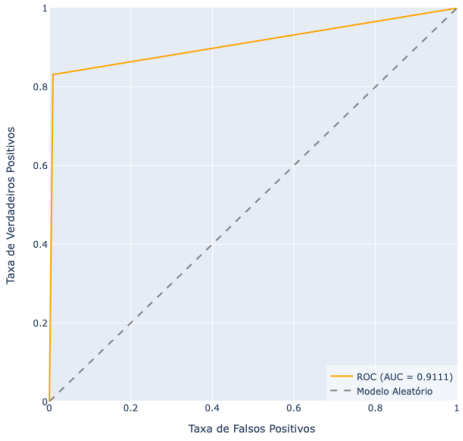
(c) Árvore de Decisão



(d) Florestas Aleatórias

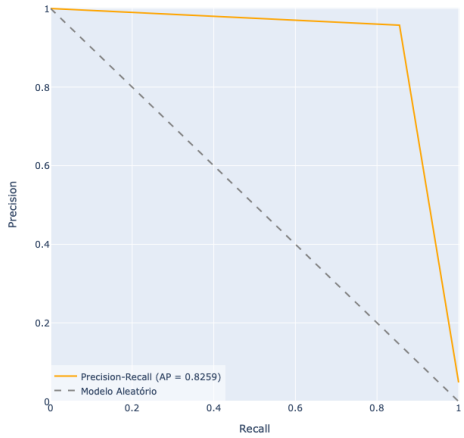


(e) KNN

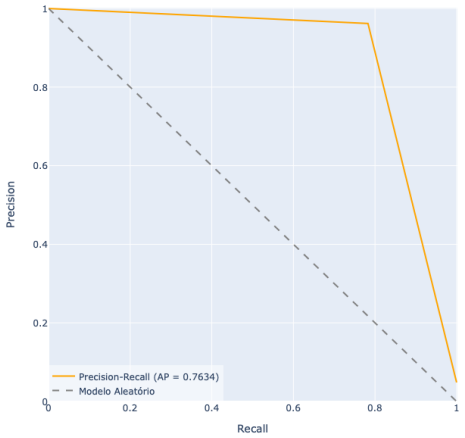


(f) *Naive Bayes*

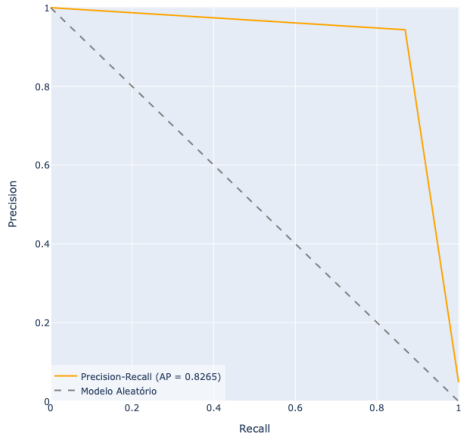
Figura I.1: Curvas ROC-AUC para Modelos de Classificação de Clientes Particulares



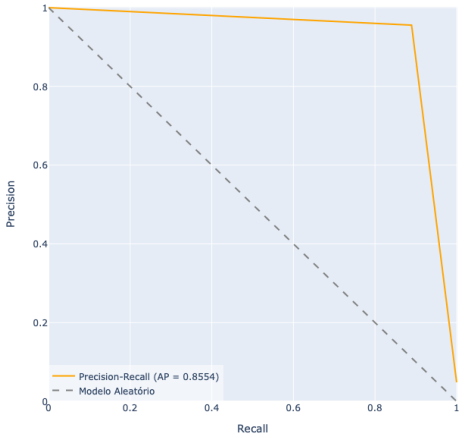
(a) SVM



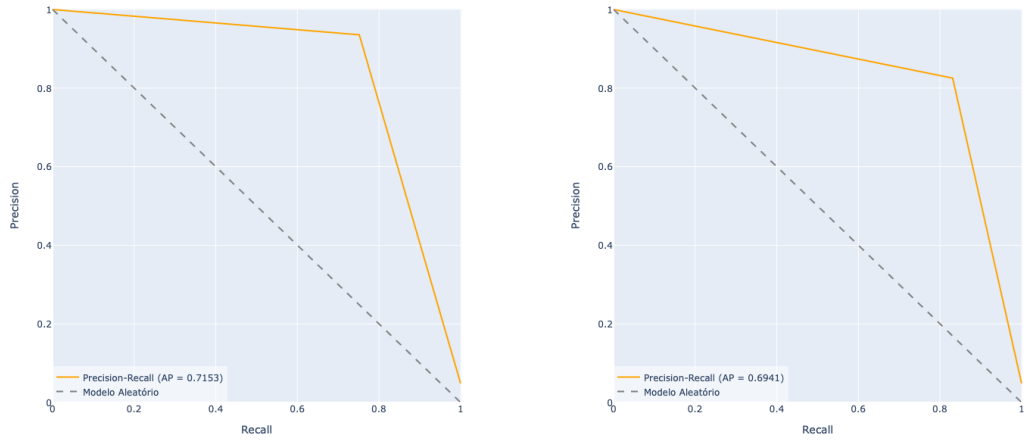
(b) Regressão Logística



(c) Árvore de Decisão



(d) Florestas Aleatórias



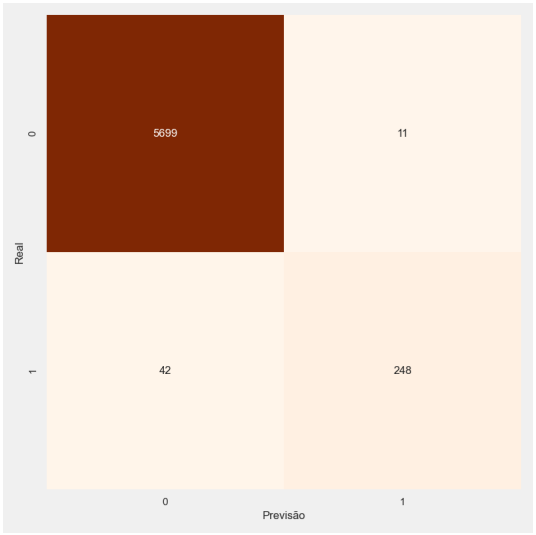
(e) KNN

(f) *Naive Bayes*

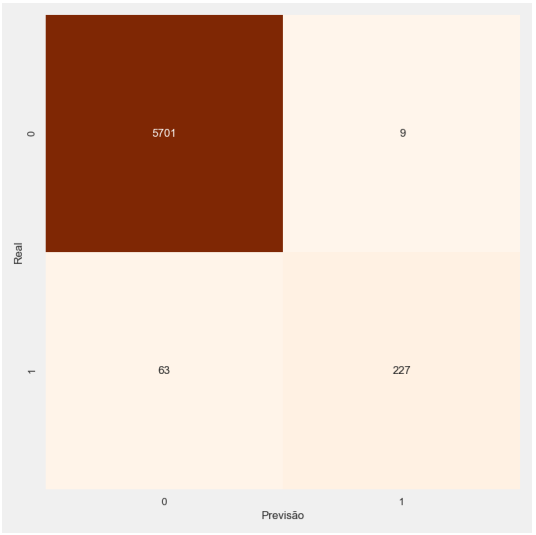
Figura I.2: Curvas *Precision vs. Recall* para Modelos de Classificação de Clientes Particulares

II

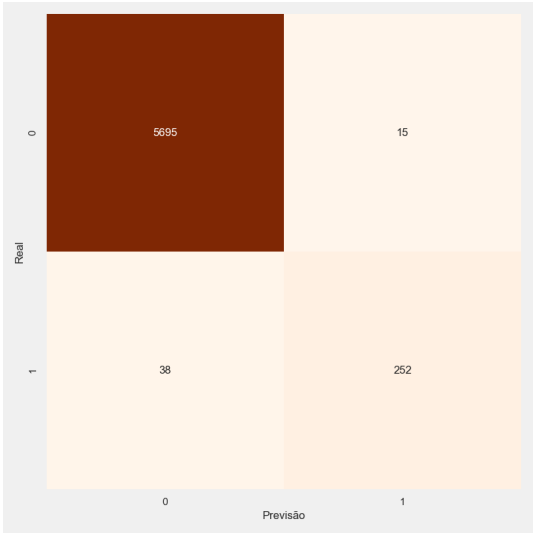
Matrizes de Confusão dos Modelos de Classificação de Clientes Particulares



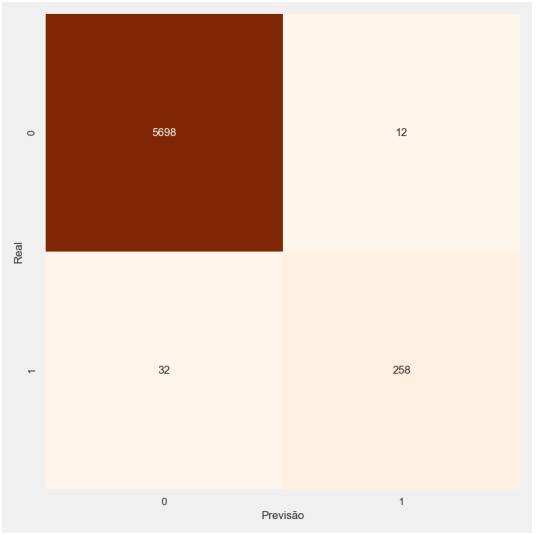
(a) SVM



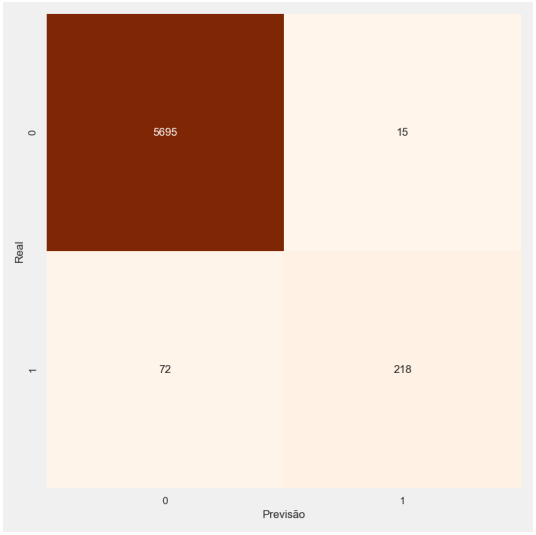
(b) Regressão Logística



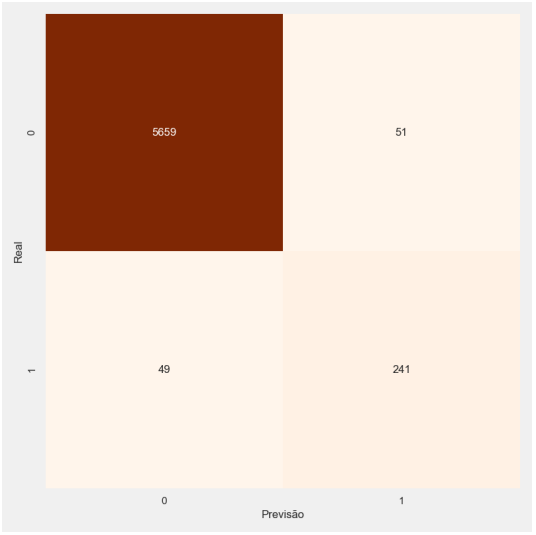
(c) Árvore de Decisão



(d) Florestas Aleatórias



(e) KNN

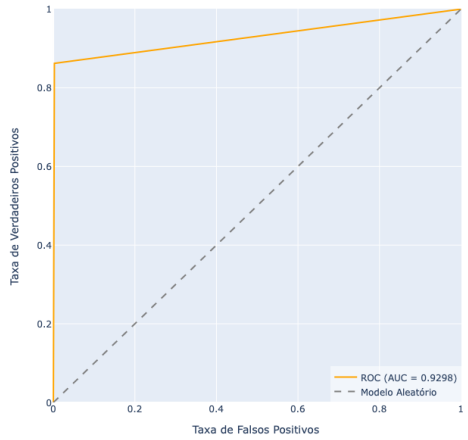


(f) *Naive Bayes*

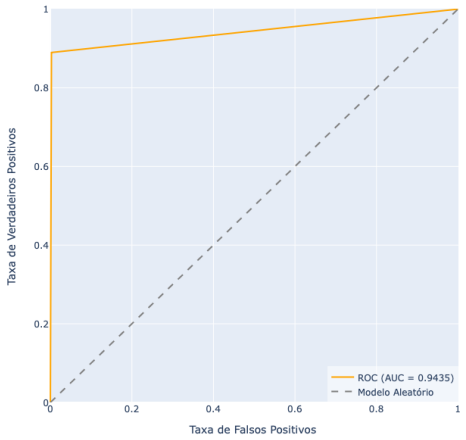
Figura II.1: Matrizes de Confusão para Modelos de Classificação de Clientes Particulares

III

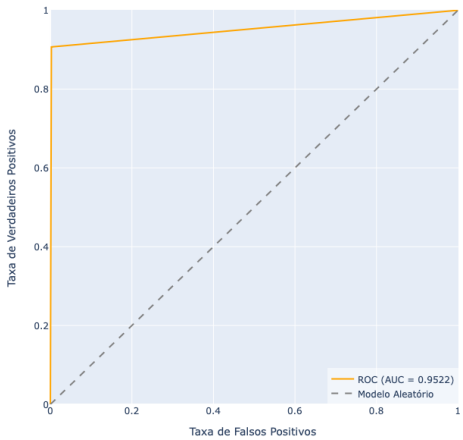
Curvas ROC-AUC e *Precision vs. Recall* para Modelos* de Classificação de Clientes Particulares



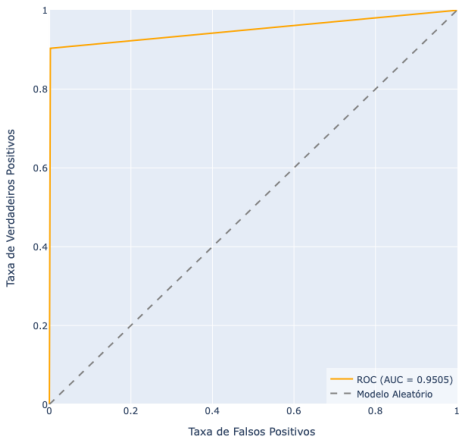
(a) SVM



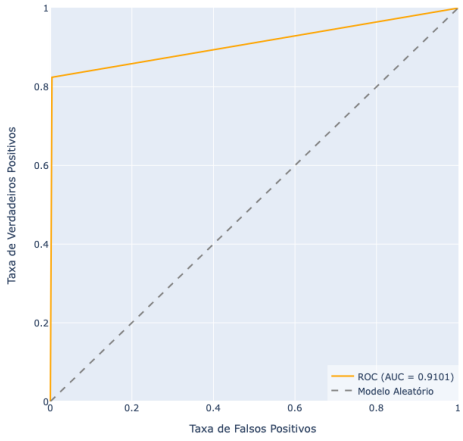
(b) Regressão Logística



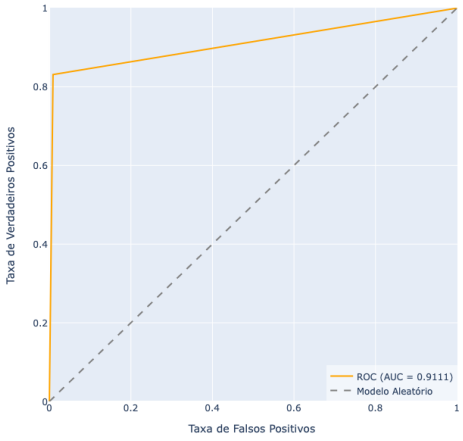
(c) Árvore de Decisão



(d) Florestas Aleatórias

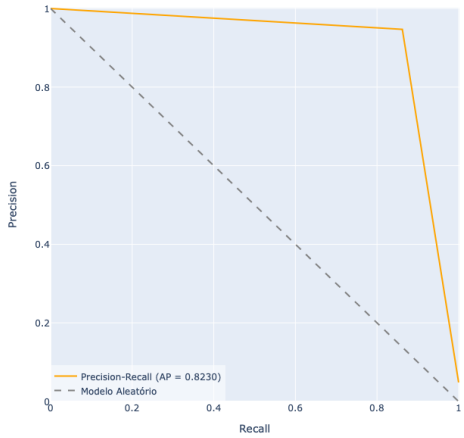


(e) KNN

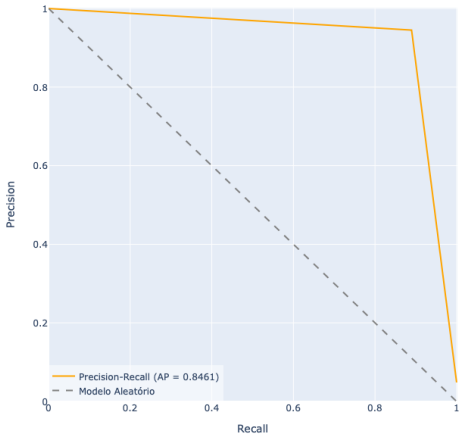


(f) *Naive Bayes*

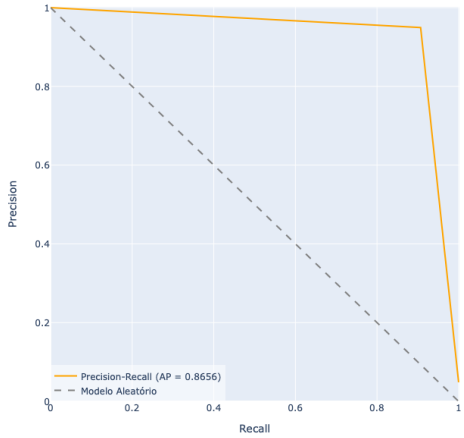
Figura III.1: Curvas ROC-AUC para Modelos* de Classificação para Clientes Particulares



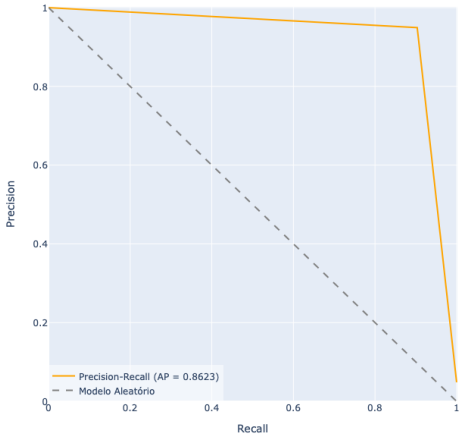
(a) SVM



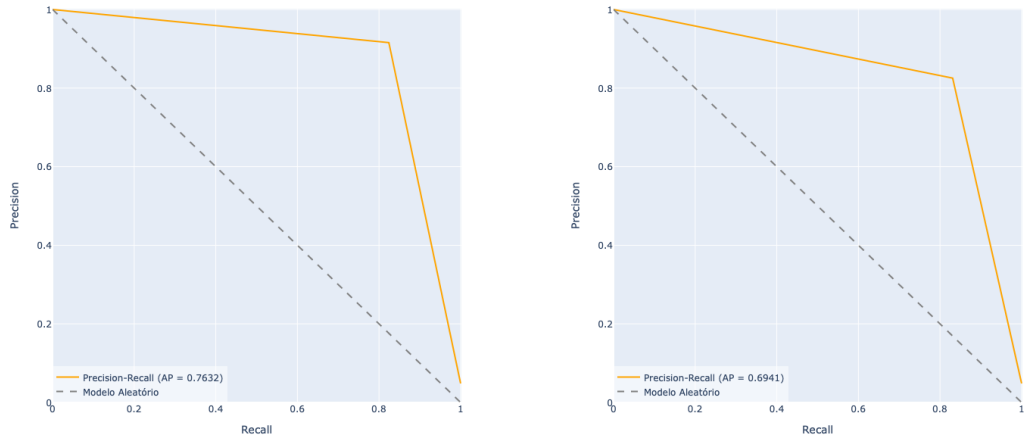
(b) Regressão Logística



(c) Árvore de Decisão



(d) Florestas Aleatórias



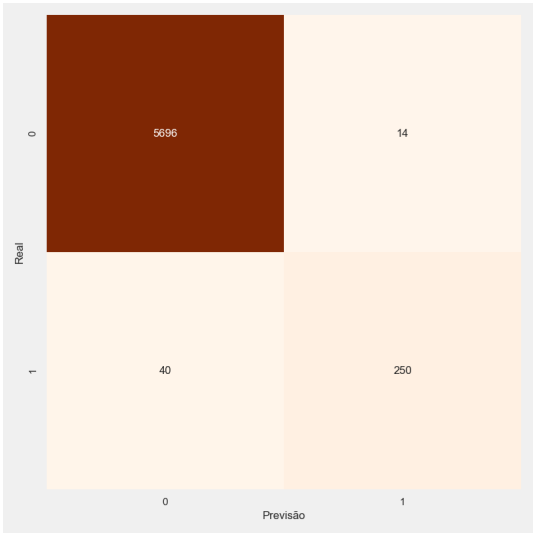
(e) KNN

(f) *Naive Bayes*

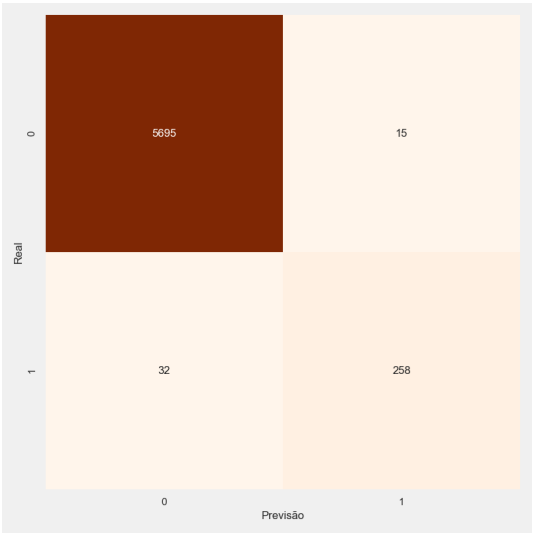
Figura III.2: Curvas *Precision vs. Recall* para Modelos* de Classificação para Clientes Particulares

IV

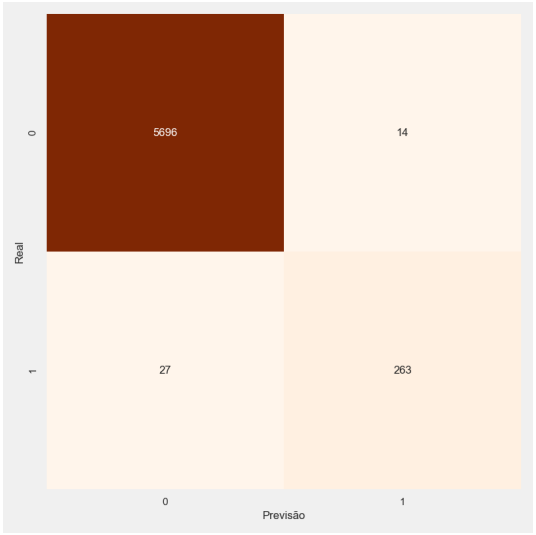
Matriz de Confusão dos Modelos* de Classificação para Clientes Particulares



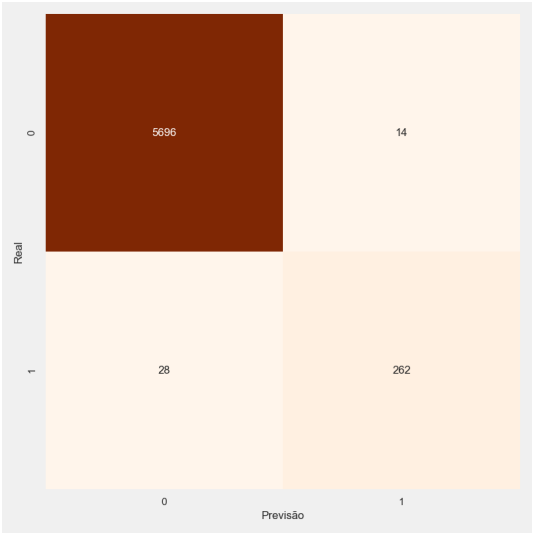
(a) SVM



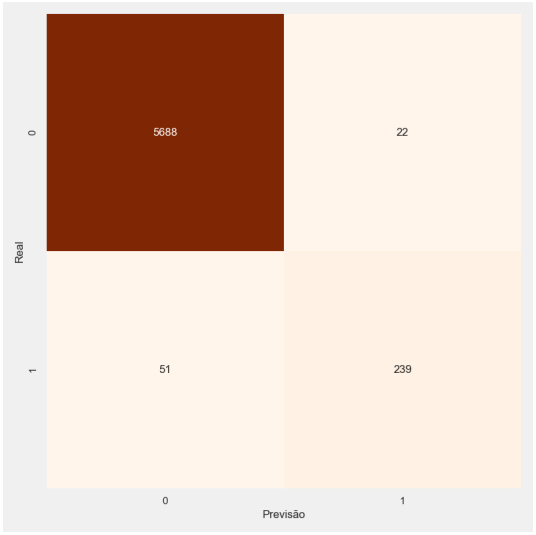
(b) Regressão Logística



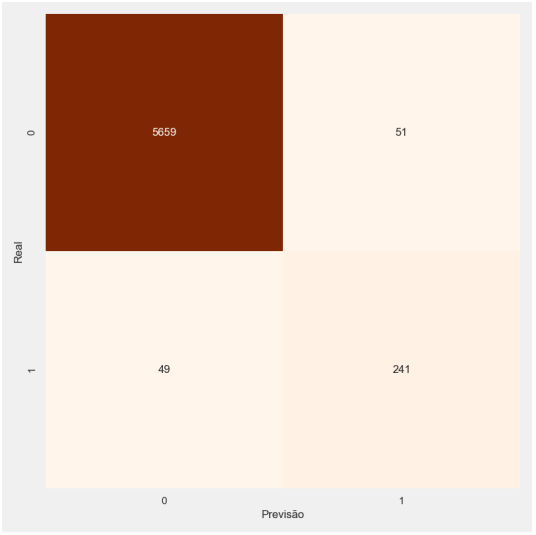
(c) Árvore de Decisão



(d) Florestas Aleatórias



(e) KNN

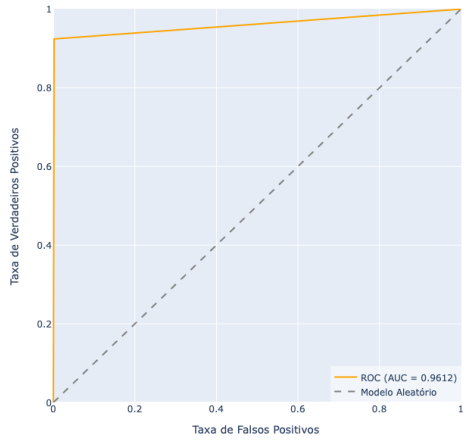


(f) *Naive Bayes*

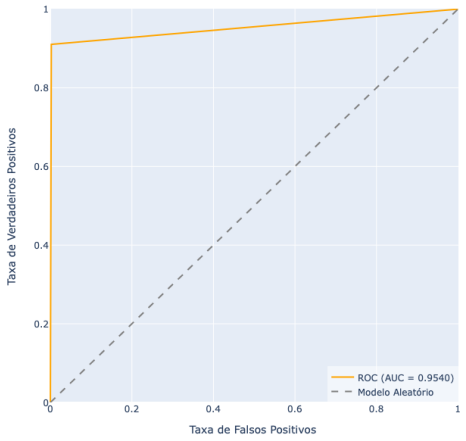
Figura IV.1: Matriz de Confusão dos Modelos* de Classificação para Clientes Particulares

V

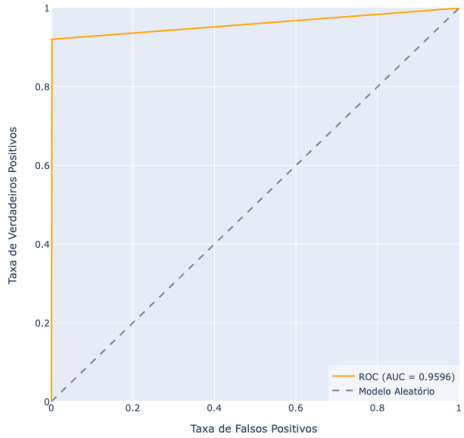
Curvas ROC-AUC e *Precision vs. Recall* de Modelos *Ensemble Learning* para Clientes Particulares



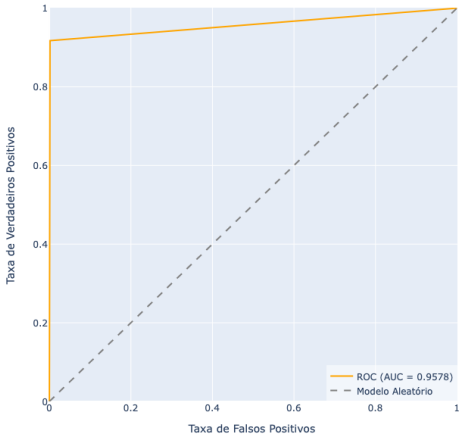
(a) *Bagged* SVM



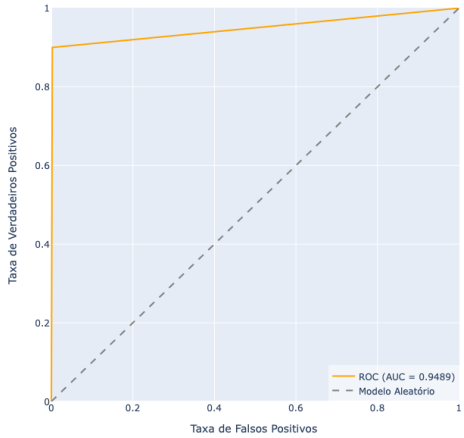
(b) *Bagged* Regressão Logística



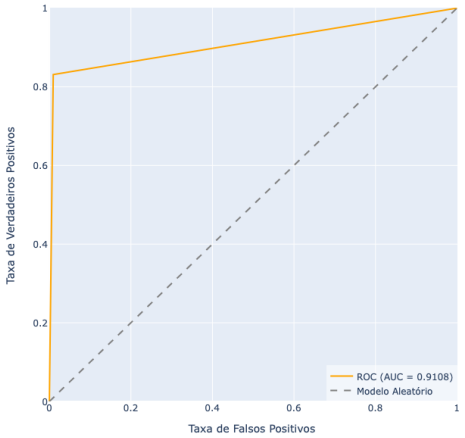
(c) *Bagged* Árvore de Decisão



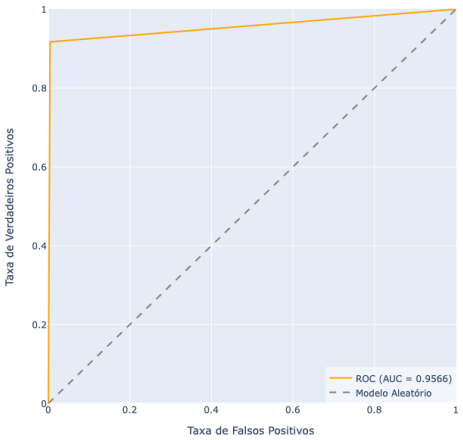
(d) *Bagged* Florestas Aleatórias



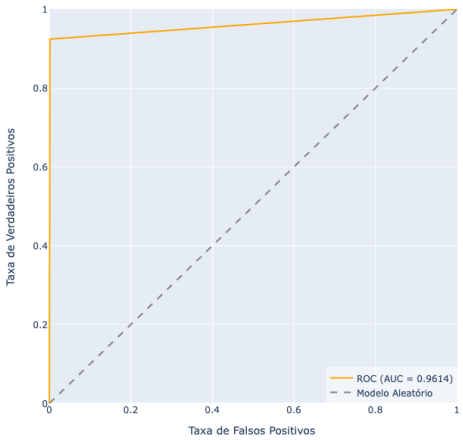
(e) *Bagged* KNN



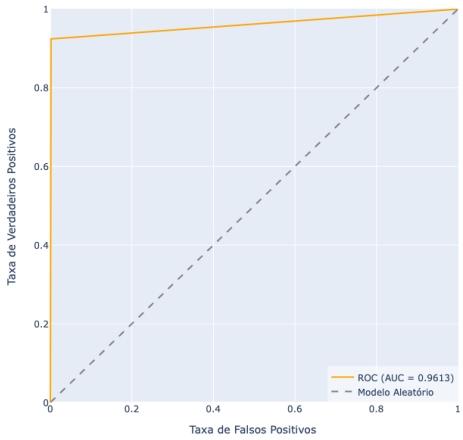
(f) *Bagged Naive Bayes*



(g) *AdaBoost*

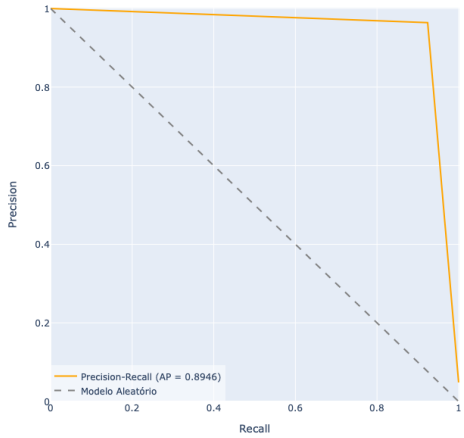


(h) *Gradient Boost*

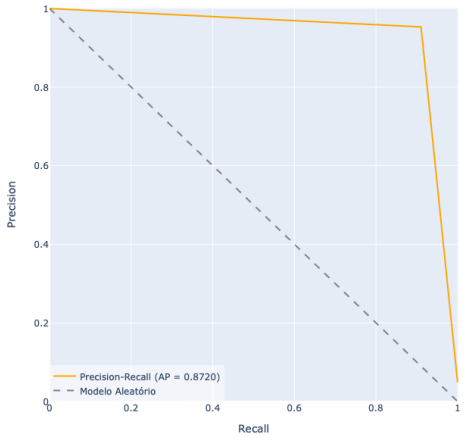


(i) *XGBoost*

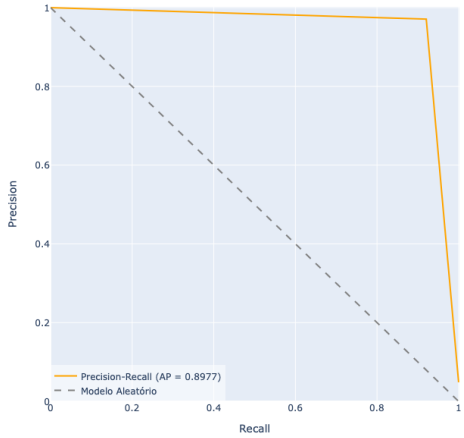
Figura V.1: Curvas ROC-AUC para Modelos *Ensemble Learning* para Clientes Particulares



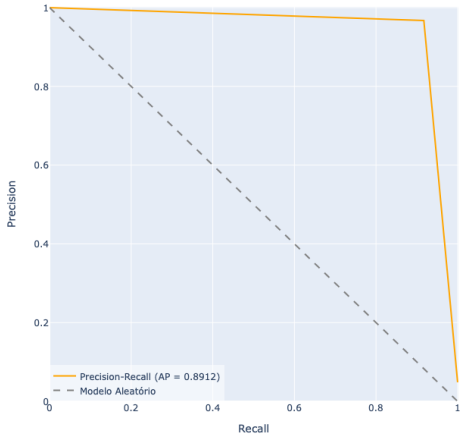
(a) *Bagged* SVM



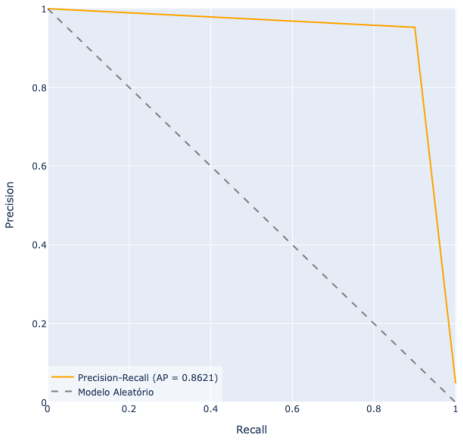
(b) *Bagged* Regressão Logística



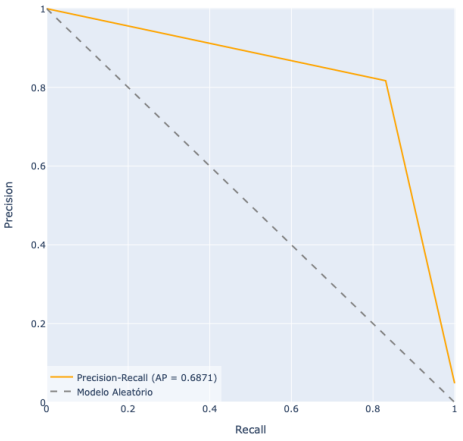
(c) *Bagged* Árvore de Decisão



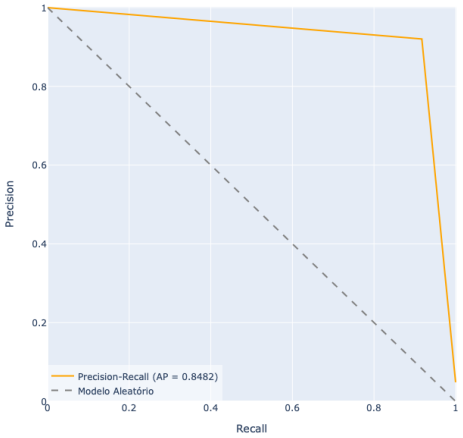
(d) *Bagged* Florestas Aleatórias



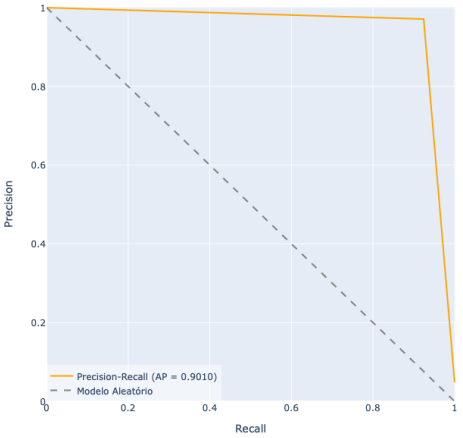
(e) *Bagged* KNN



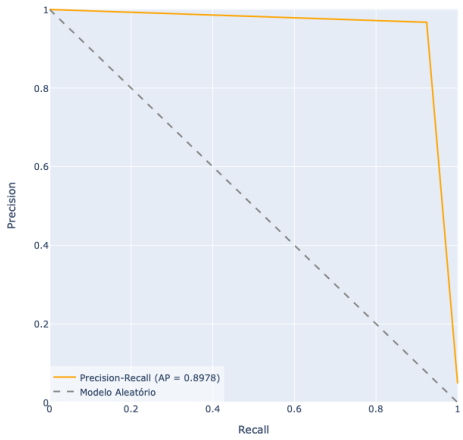
(f) *Bagged Naive Bayes*



(g) *AdaBoost*



(h) *Gradient Boost*

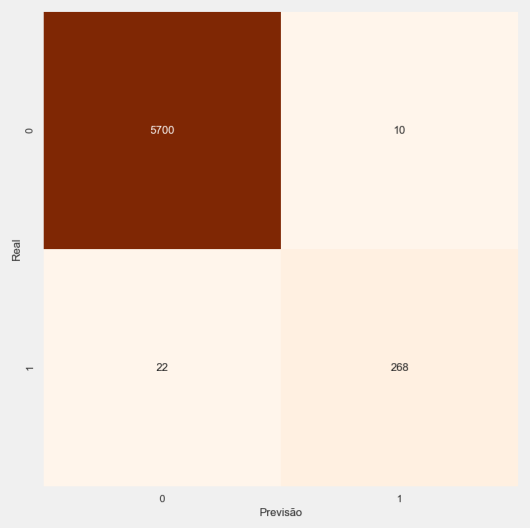


(i) *XGBoost*

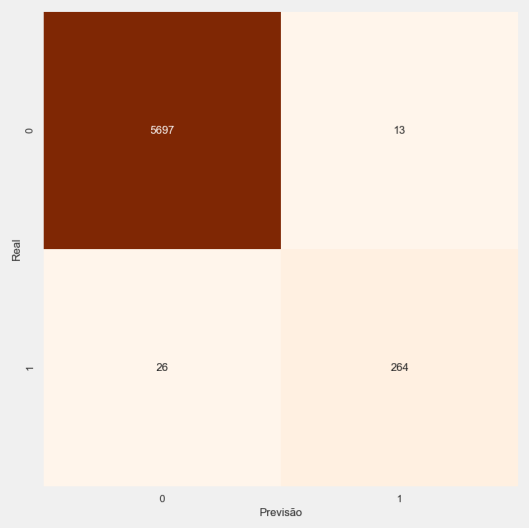
Figura V.2: Curvas *Precision vs. Recall* de Modelos *Ensemble Learning* para Clientes Particulares

VI

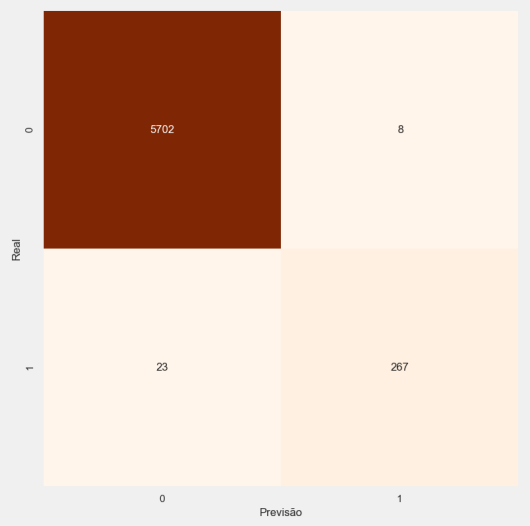
Matriz de Confusão dos Modelos *Esemble Learning* de Clientes Particulares



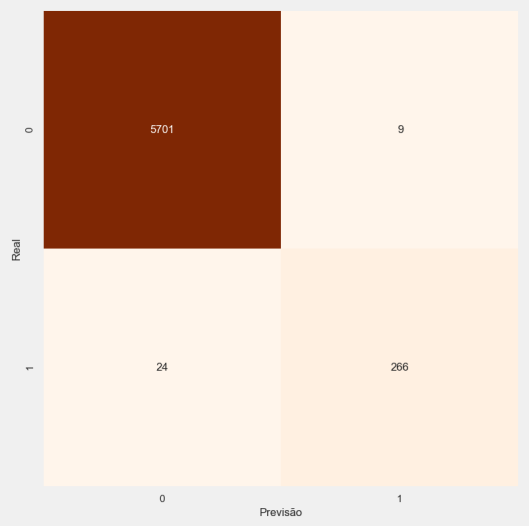
(a) *Bagged* SVM



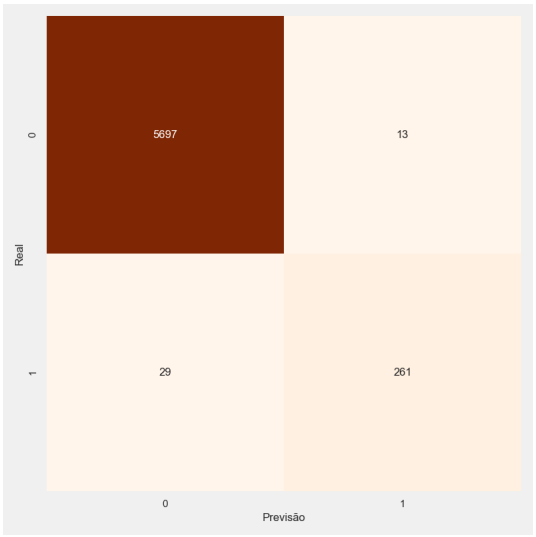
(b) *Bagged* Regressão Logística



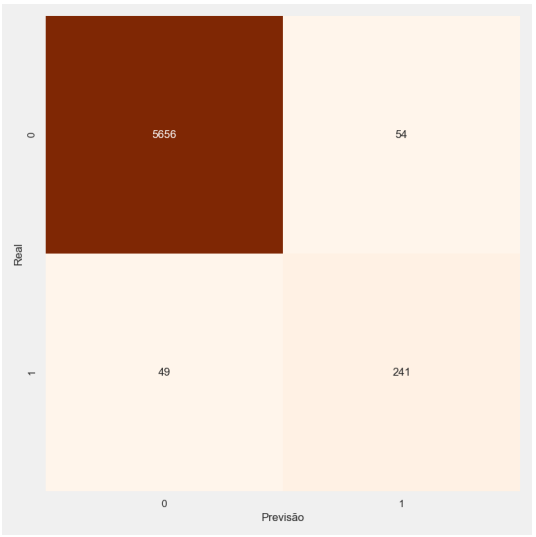
(c) *Bagged* Árvore de Decisão



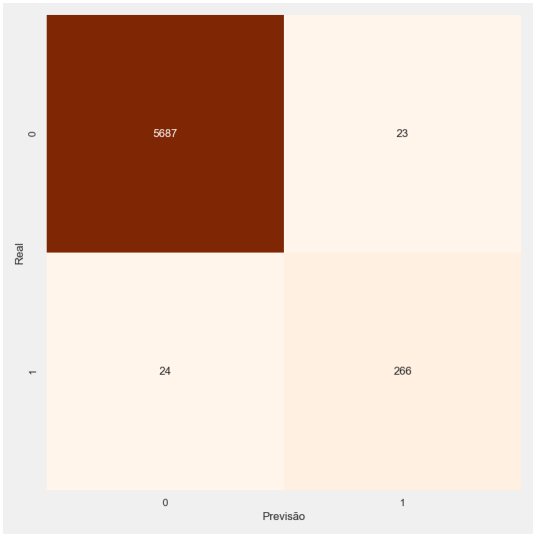
(d) *Bagged* Florestas Aleatórias



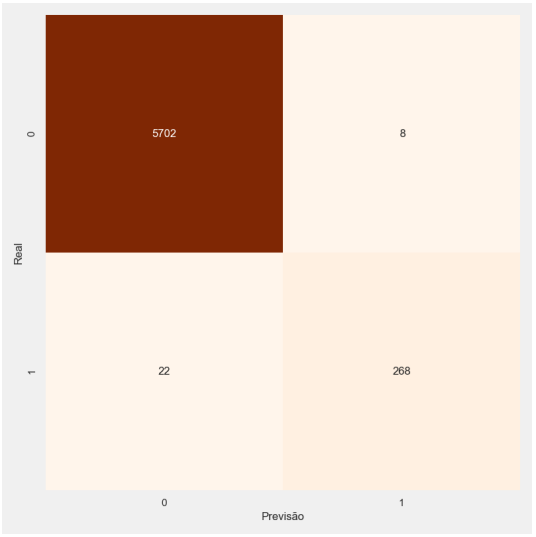
(e) *Bagged* KNN



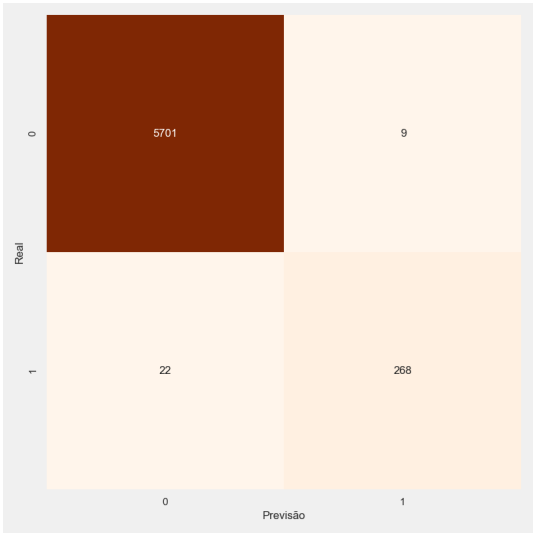
(f) *Bagged Naive Bayes*



(g) *AdaBoost*



(h) *Gradient Boost*

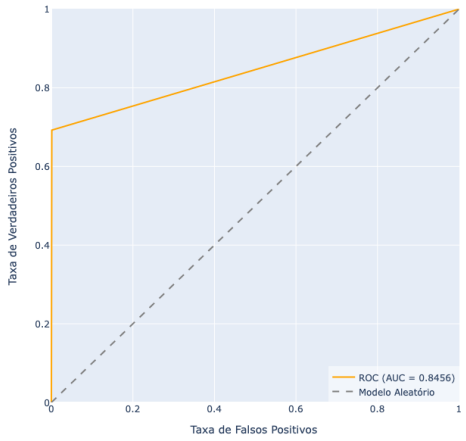


(i) *XGBoost*

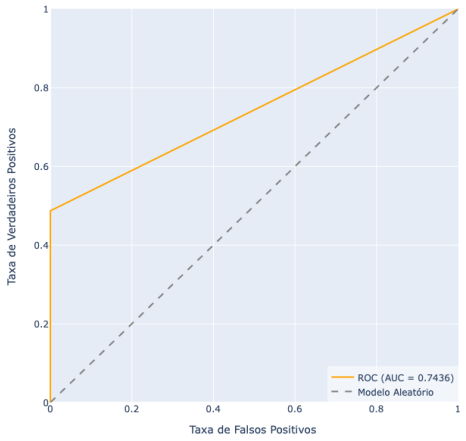
Figura VI.1: Matriz de Confusão de Modelos *Esemble Learning* para Clientes Particulares

VII

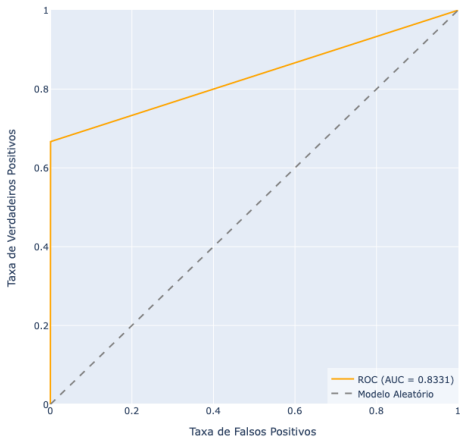
Curvas ROC-AUC e *Precision vs. Recall* para Modelos de Classificação de Clientes Corporativos



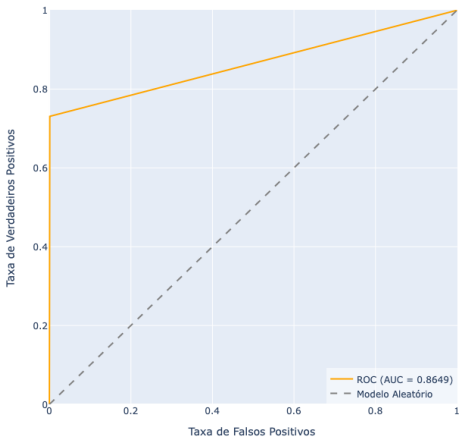
(a) SVM



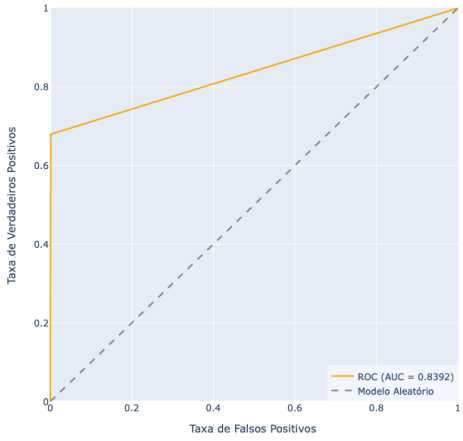
(b) Regressão Logísitca



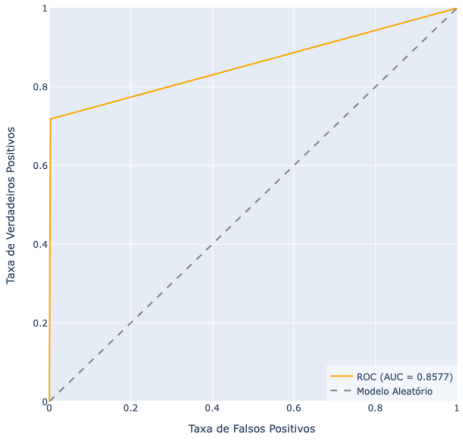
(c) Árvore de Decisão



(d) Florestas Aleatórias

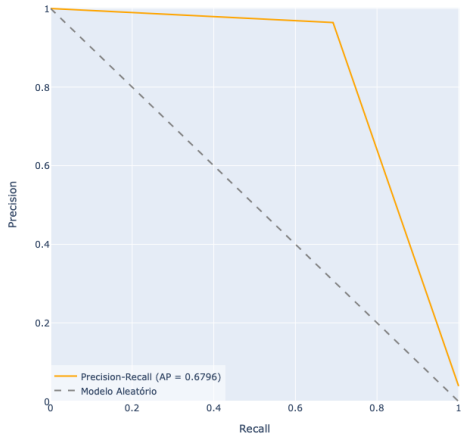


(e) KNN

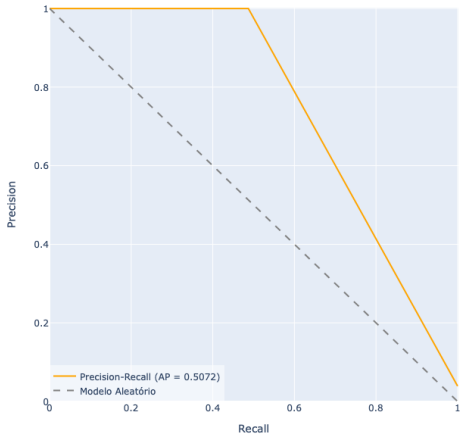


(f) *Naive Bayes*

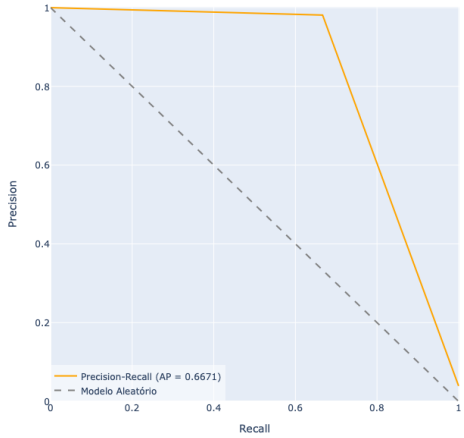
Figura VII.1: Curvas ROC-AUC para Modelos de Classificação de Clientes Corporativos



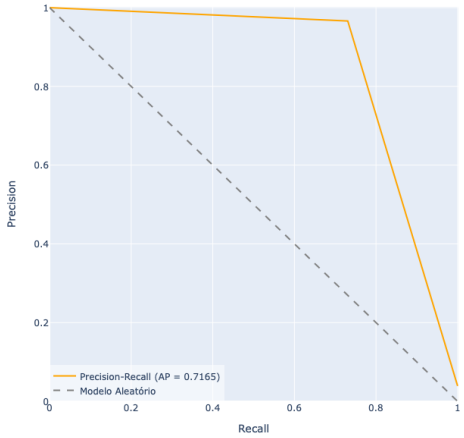
(a) SVM



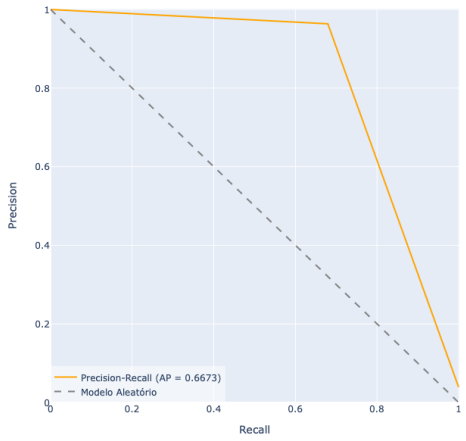
(b) Regressão Logística



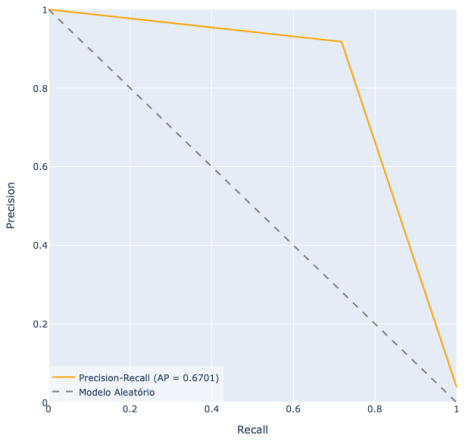
(c) Árvore de Decisão



(d) Florestas Aleatórias



(e) KNN

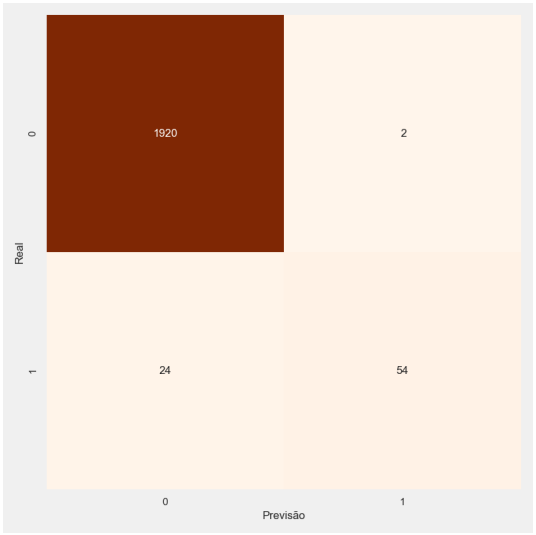


(f) *Naive Bayes*

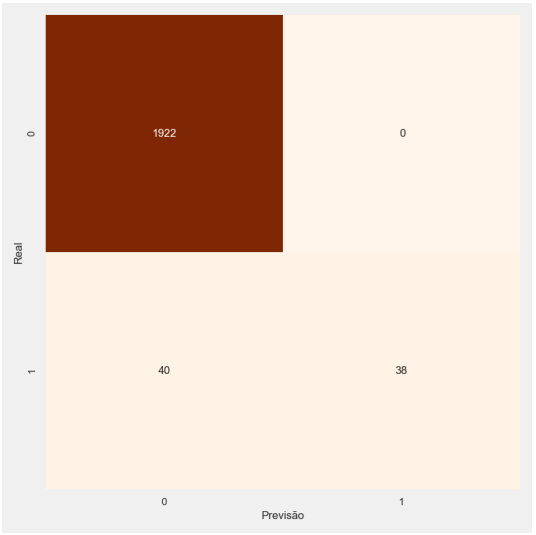
Figura VII.2: Curvas *Precision vs. Recall* para Modelos de Classificação de Clientes Corporativos

VIII

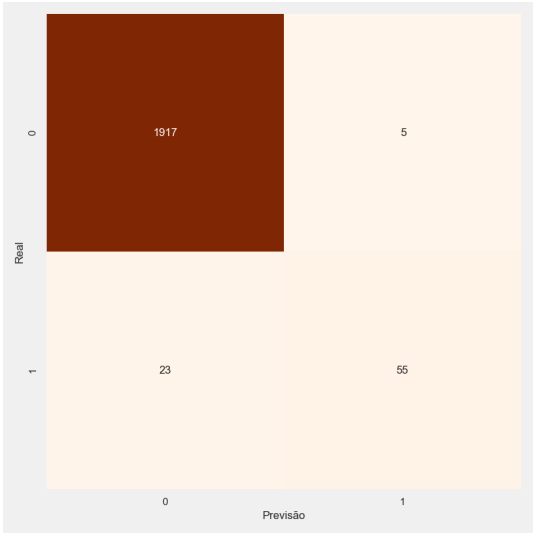
Matrizes de Confusão dos Modelos de Classificação para Clientes Corporativos



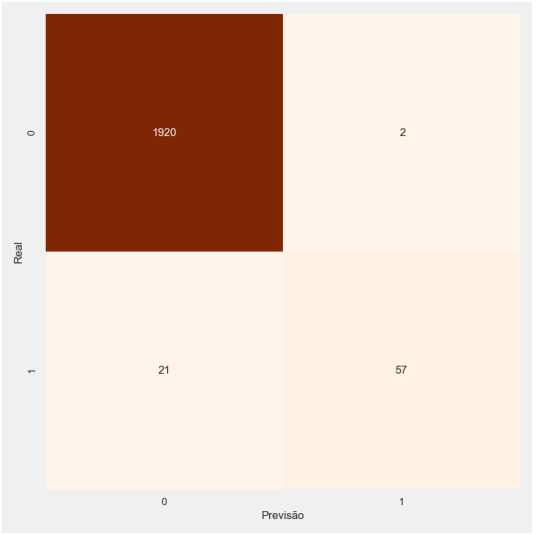
(a) SVM



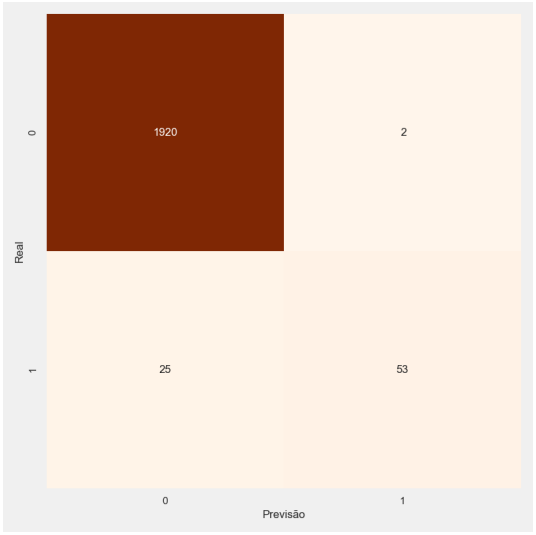
(b) Regressão Logísitca



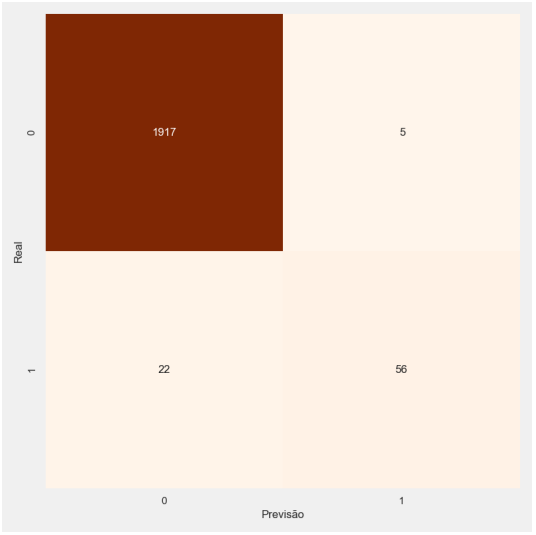
(c) Árvore de Decisão



(d) Florestas Aleatórias



(e) KNN

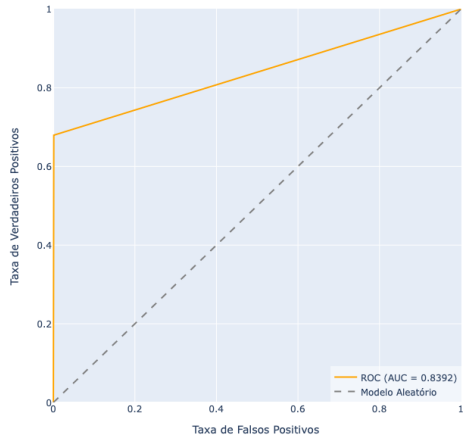


(f) *Naive Bayes*

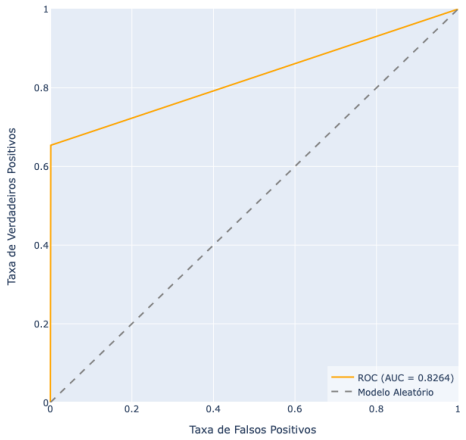
Figura VIII.1: Matrizes de Confusão de Modelos de Classificação para Clientes Corporativos

IX

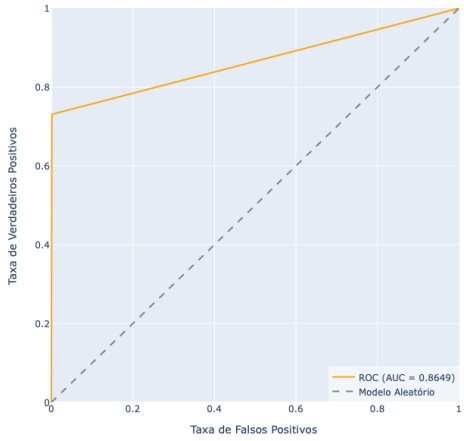
Curvas ROC-AUC e *Precision vs. Recall* para Modelos* de Classificação de Clientes Corporativos



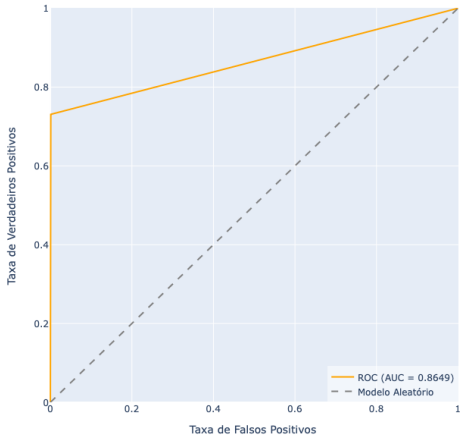
(a) SVM



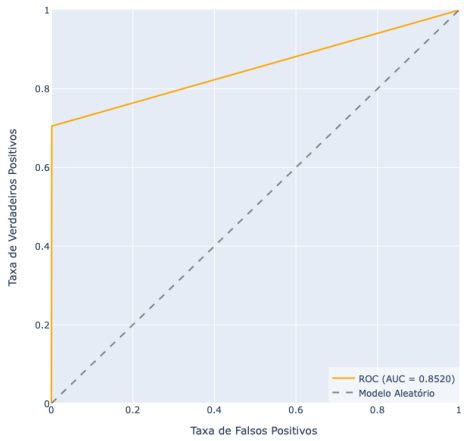
(b) Regressão Logística



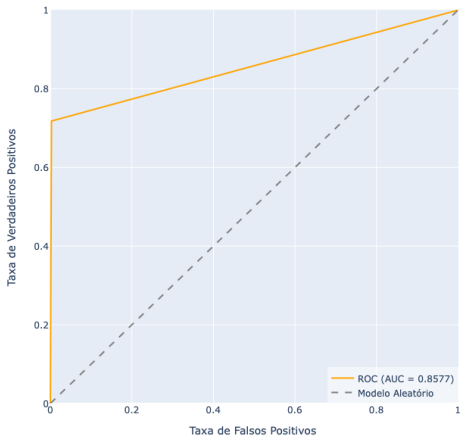
(c) Árvore de Decisão



(d) Florestas Aleatórias

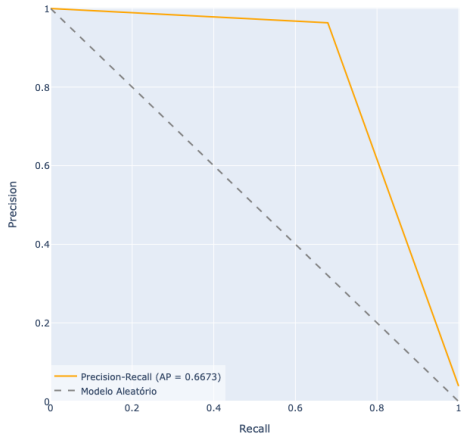


(e) KNN

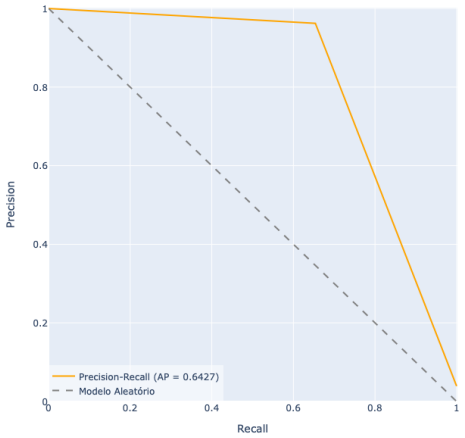


(f) *Naive Bayes*

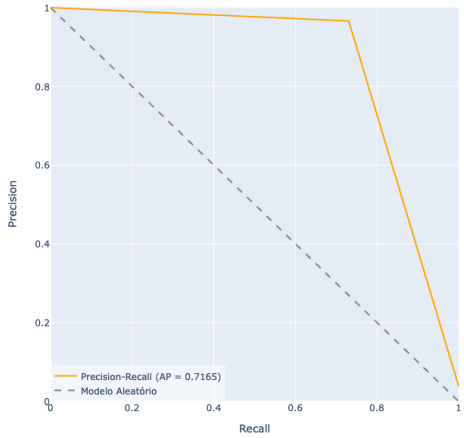
Figura IX.1: Curvas ROC-AUC e *Precision vs. Recall* de Modelos* de Classificação para Clientes Corporativos



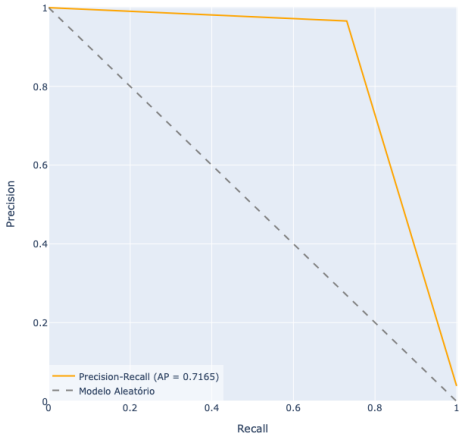
(a) SVM



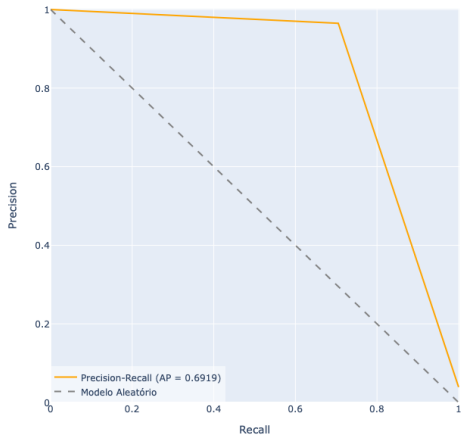
(b) Regressão Logística



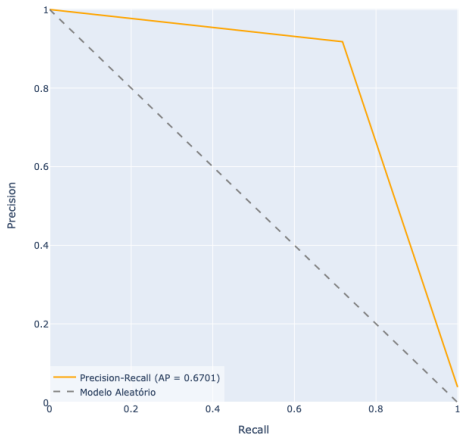
(c) Árvore de Decisão



(d) Florestas Aleatórias



(e) KNN

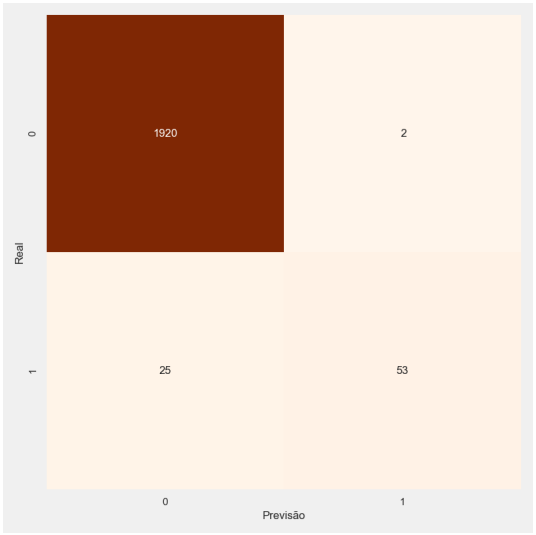


(f) *Naive Bayes*

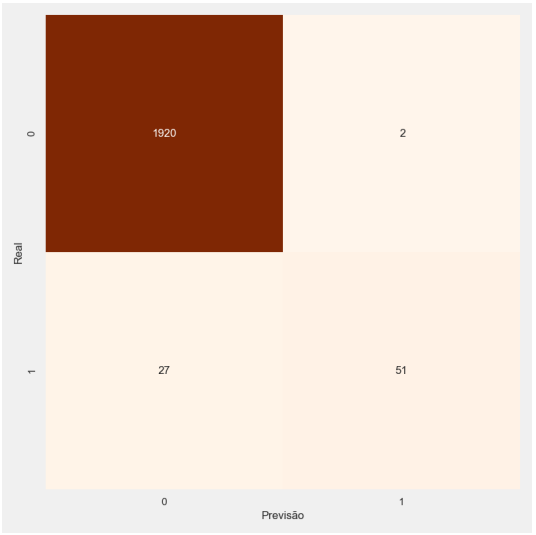
Figura IX.2: Curvas *Precision vs. Recall* de Modelos* de Classificação para Clientes Corporativos

X

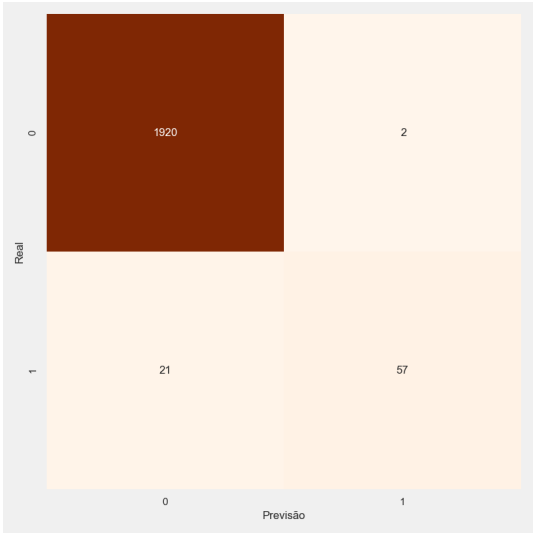
Matriz de Confusão dos Modelos* de Classificação para Clientes Corporativos



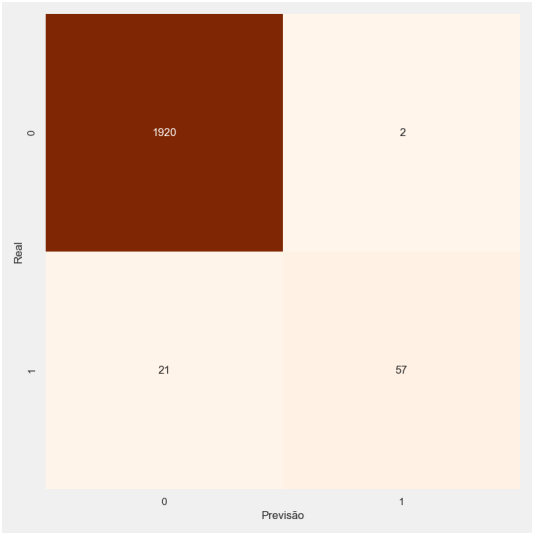
(a) SVM



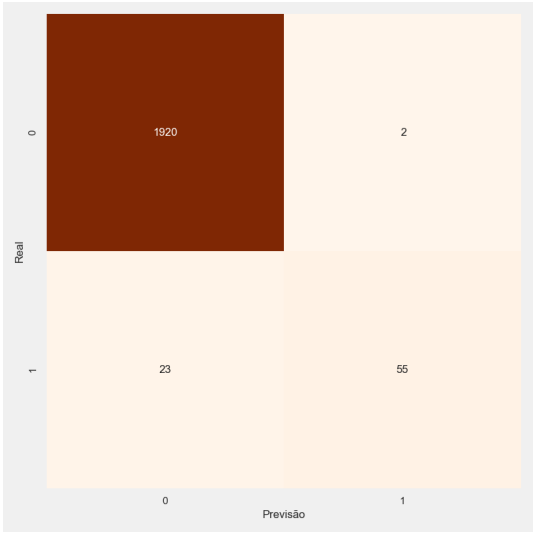
(b) Regressão Logística



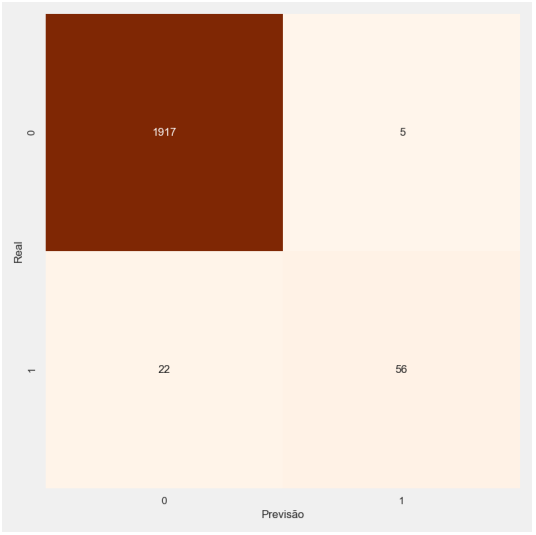
(c) Árvore de Decisão



(d) Florestas Aleatórias



(e) KNN

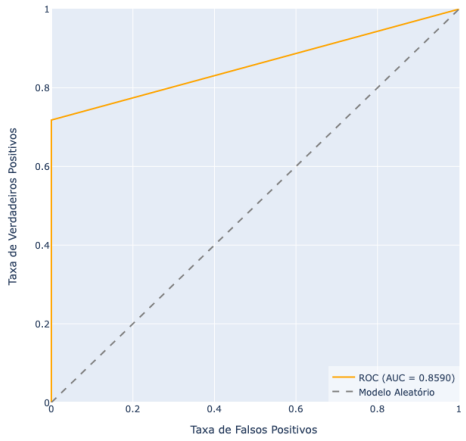


(f) *Naive Bayes*

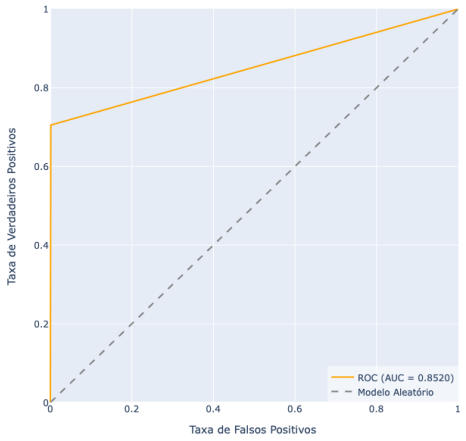
Figura X.1: Matriz de Confusão dos Modelos* de Classificação para Clientes Corporativos

XI

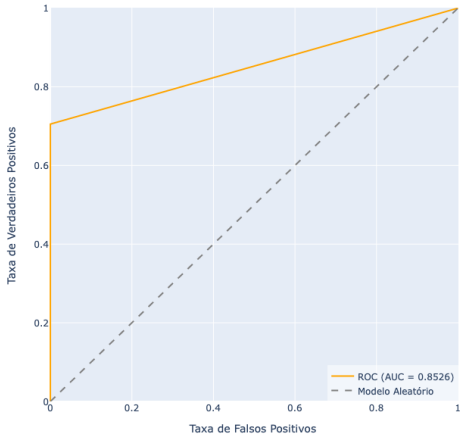
Curvas ROC-AUC e *Precision vs. Recall* de Modelos *Ensemble Learning* para Clientes Corporativos



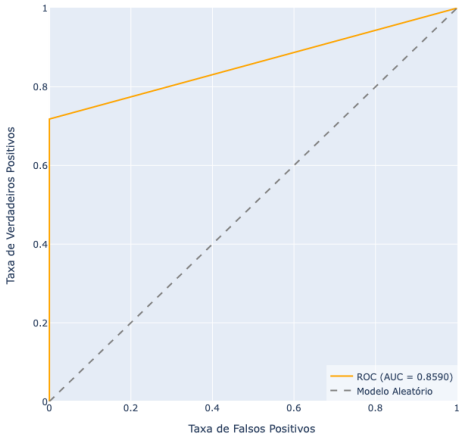
(a) *Bagged* SVM



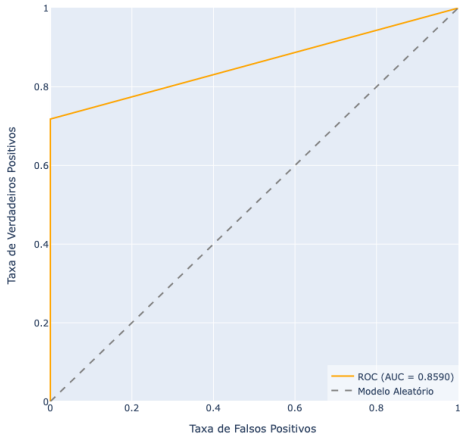
(b) *Bagged* Regressão Logística



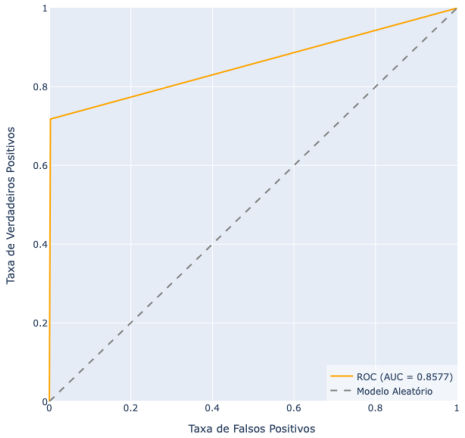
(c) *Bagged* Árvore de Decisão



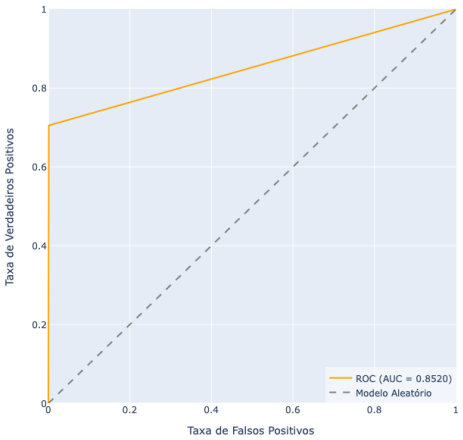
(d) *Bagged* Florestas Aleatórias



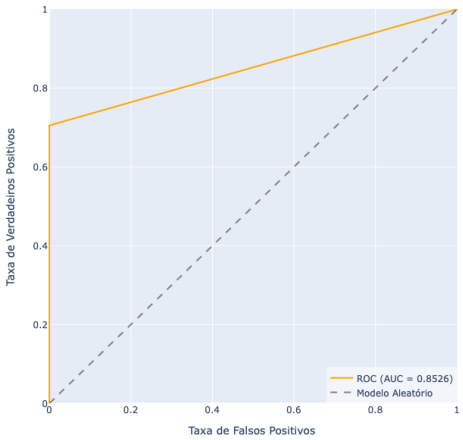
(e) *Bagged* KNN



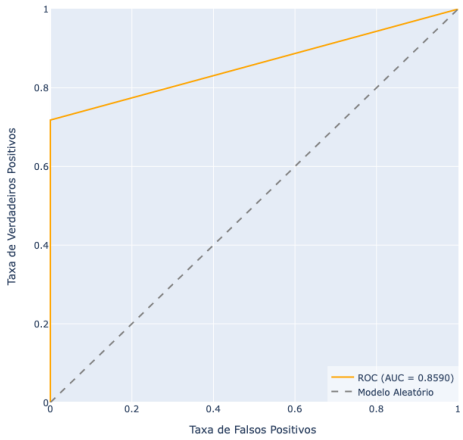
(f) *Bagged Naive Bayes*



(g) *AdaBoost*

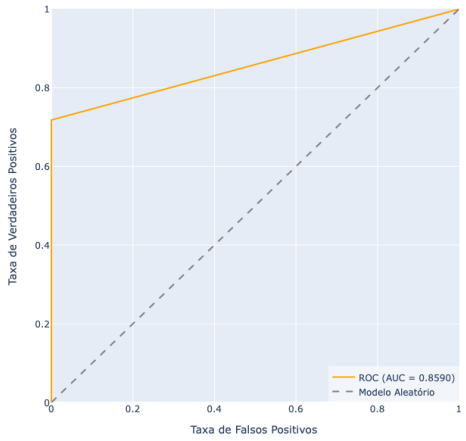


(h) *Gradient Boost*

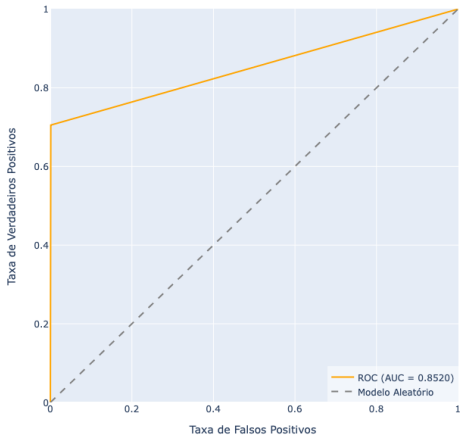


(i) *XGBoost*

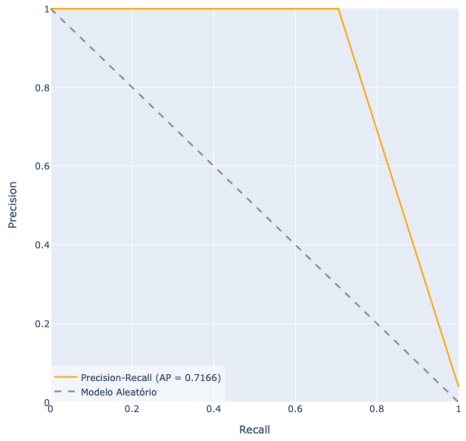
Figura XI.1: Curvas ROC-AUC de Modelos *Ensemble Learning* para Clientes Corporativos



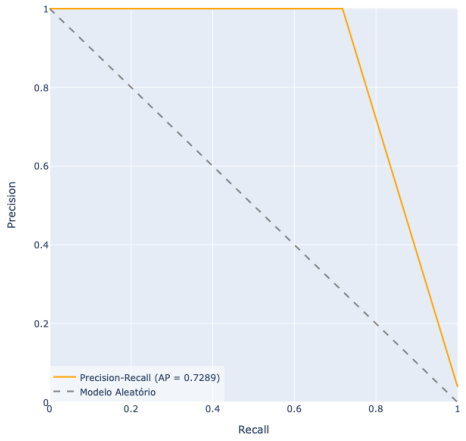
(a) *Bagged* SVM



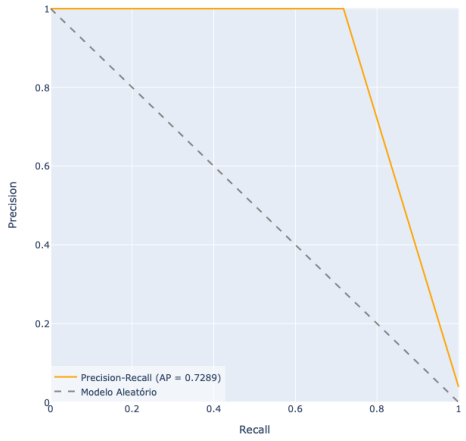
(b) *Bagged* Regressão Logística



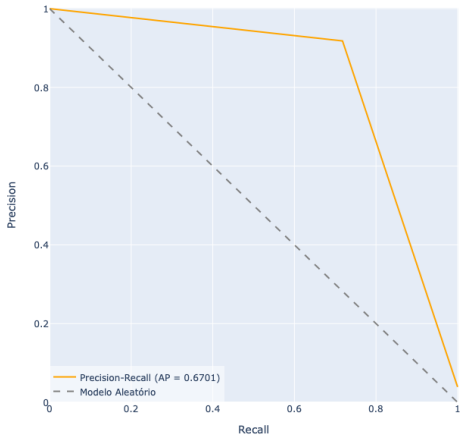
(c) *Bagged* Árvore de Decisão



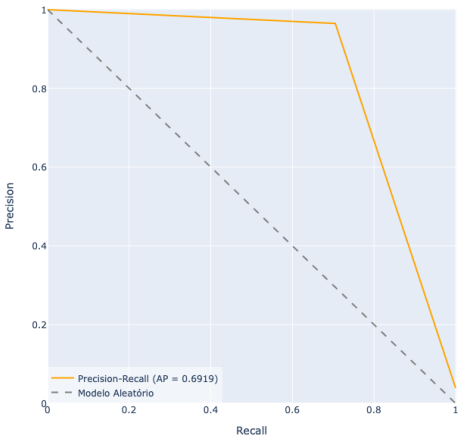
(d) *Bagged* Florestas Aleatórias



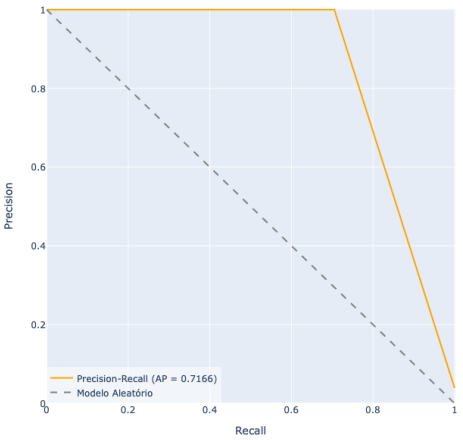
(e) *Bagged* KNN



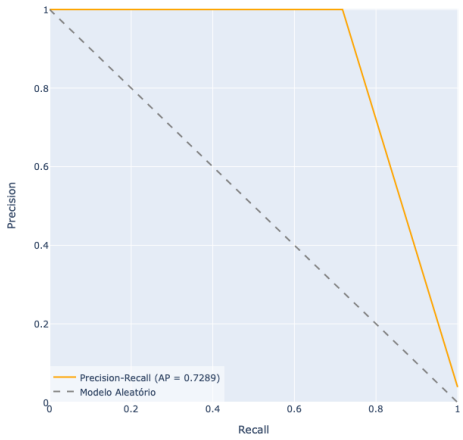
(f) *Bagged Naive Bayes*



(g) *AdaBoost*



(h) *Gradietn Boost*

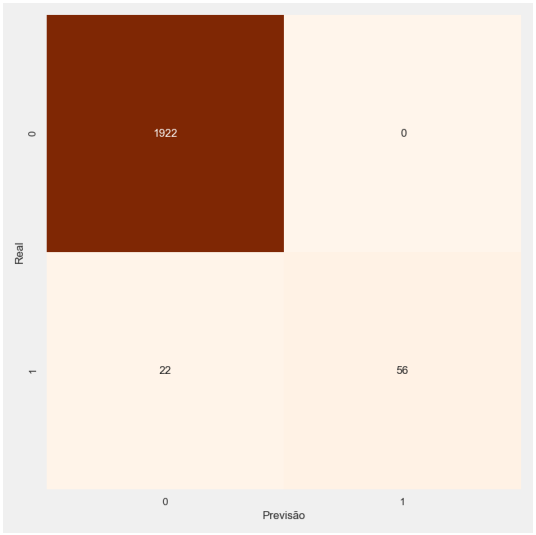


(i) *XGBoost*

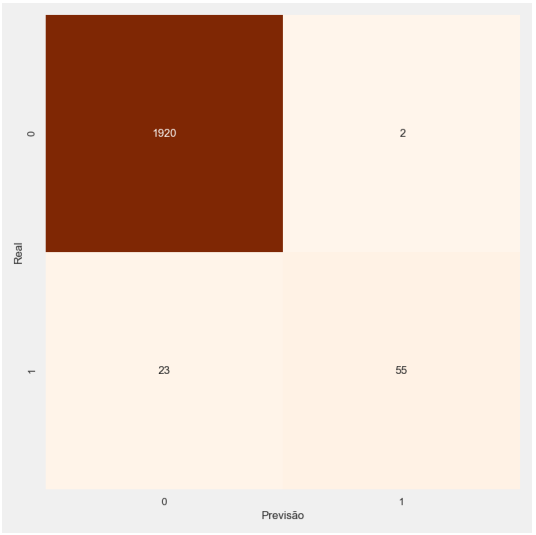
Figura XI.2: Curvas *Precision vs. Recall* de Modelos *Ensemble Learning* para Clientes Corporativos

XII

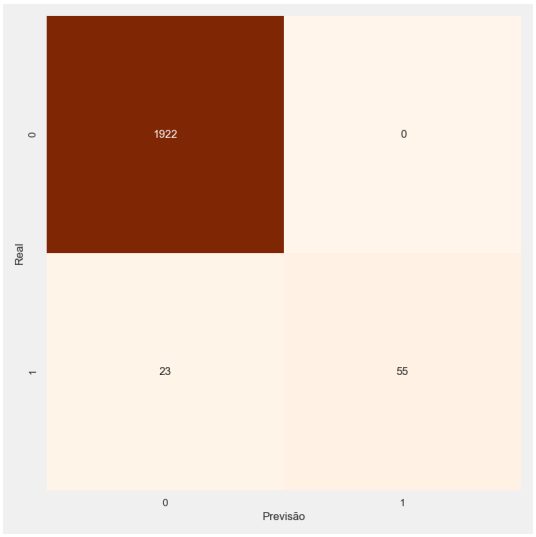
Matriz de Confusão de Modelos *Esemble Learning* para Clientes Corporativos



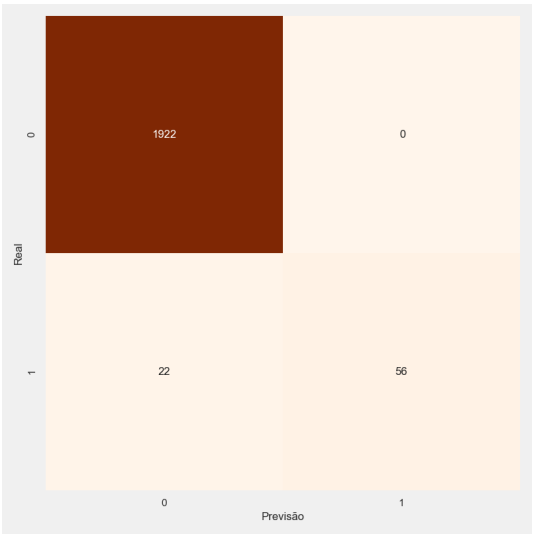
(a) *Bagged* SVM



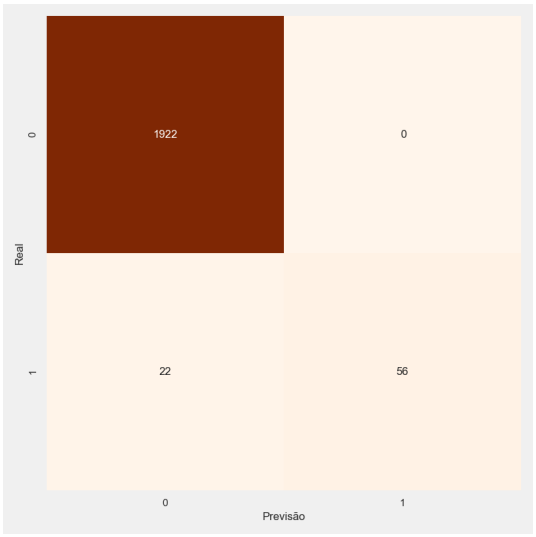
(b) *Bagged* Regressão Logística



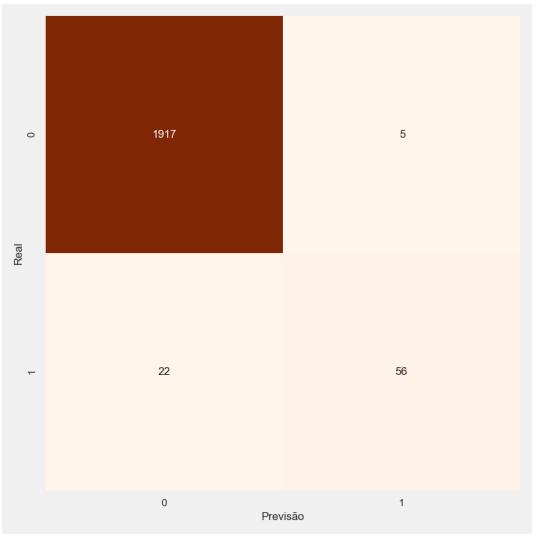
(c) *Bagged* Árvore de Decisão



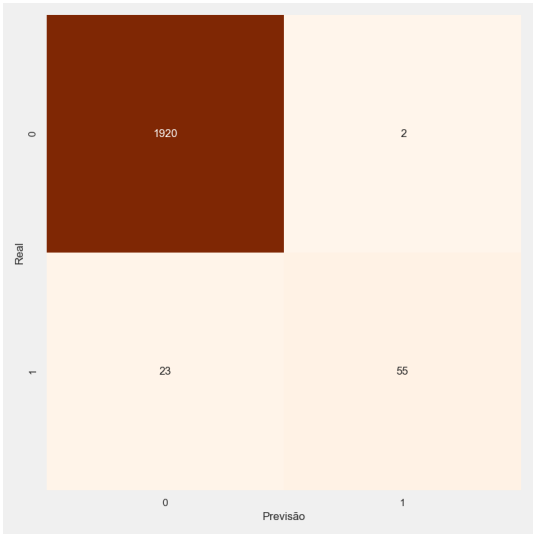
(d) *Bagged* Florestas Aleatórias



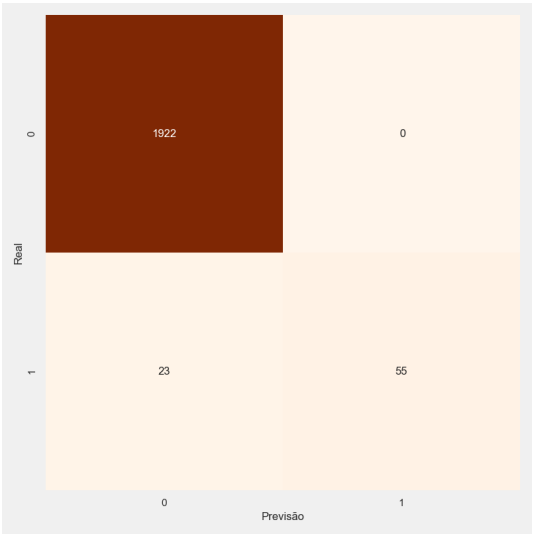
(e) *Bagged* KNN



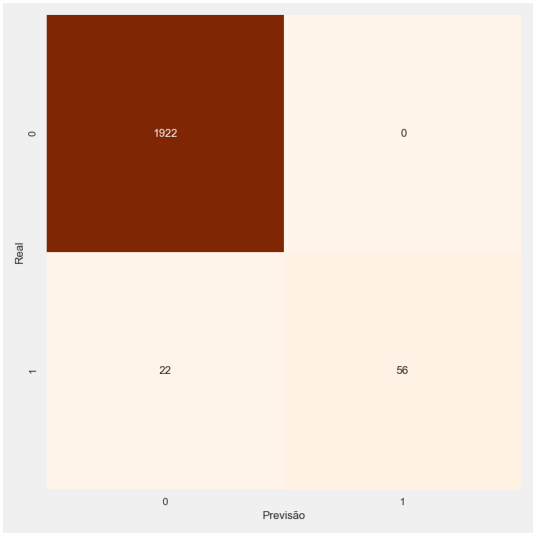
(f) *Bagged Naive Bayes*



(g) *AdaBoost*



(h) *Gradient Boost*



(i) *XGBoost*

Figura XII.1: Matriz de Confusão de Modelos *Esemble Learning* para Clientes Corporativos