UNIVERSIDADE FEDERAL DE JUIZ DE FORA
INSTITUTO DE CIÊNCIAS EXATAS
CURSO DE ESTATÍSTICA

Deiverson Eduardo Oliveira de Almeida

Aplicação de Técnicas de Machine Learning na Identificação de Transações Fraudulentas no E-commerce

Juiz de Fora

2024

Deiverson Eduardo Oliveira de Almeida

Aplicação de Técnicas de Machine Learning na Identificação de Transações Fraudulentas no E-commerce

Monografia apresentada ao Curso de Estatística da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Bacharel em Estatística.

Orientadora: Prof[a]. Dra. Camila Borelli Zeller

Juiz de Fora

2024

Ficha catalográfica elaborada através do Modelo Latex do CDC da UFJF
com os dados fornecidos pelo(a) autor(a)

Almeida, Deiverson Eduardo Oliveira de.
Aplicação de Técnicas de Machine Learning na Identificação de Transações Fraudulentas no E-commerce / Deiverson Eduardo Oliveira de Almeida. – 2024.
75 f. : il.

Orientadora: Camila Borelli Zeller
Trabalho de Conclusão de Curso (graduação) – Universidade Federal de Juiz de Fora, Instituto de Ciências Exatas. Curso de Estatística, 2024.

1. Detecção de fraude. 2. Machine Learning 3. Floresta Aleatória. 4. Regressão Logística. 5. Balanceamento de dados. I. Zeller, Camila Borelli, orient. II. Título.

Deiverson Eduardo Oliveira de Almeida

Aplicação de Técnicas de Machine Learning na Identificação de Transações Fraudulentas no E-commerce

Monografia apresentada ao Curso de Estatística da Universidade Federal de Juiz de Fora como requisito parcial à obtenção do título de Bacharel em Estatística.

Aprovada em 15 de julho de 2024

BANCA EXAMINADORA

Prof[a]. Dra. Camila Borelli Zeller - Orientadora
Universidade Federal de Juiz de Fora

Prof. Dr. Lupércio França Bessegato
Universidade Federal de Juiz de Fora

Prof. Dr. Tiago Maia Magalhães
Universidade Federal de Juiz de Foura

RESUMO

Com a popularização da internet ao longo dos anos, comprar online tornou-se cada vez mais comum, principalmente devido à facilidade e ao conforto oferecidos, além das variadas formas de pagamento, sendo o cartão de crédito um dos principais meios utilizados em transações no comércio eletrônico (*e-commerce*). Esta facilidade, no entanto, também atrai fraudadores, que utilizam a internet para aplicar golpes cada vez mais elaborados. Dessa forma, torna-se imprescindível a identificação eficaz de transações fraudulentas para mitigar este risco e proteger os bons compradores. Um dos principais desafios na identificação de fraudes é o desbalanceamento dos dados, já que a fraude é um evento raro e naturalmente está presente em uma quantidade muito menor do que as transações legítimas, o que tende a diminuir o poder de discriminação das técnicas. Dentre as diversas técnicas de *Machine Learning* comumente utilizadas para este fim, duas delas, a Regressão Logística e a Floresta Aleatória, foram exploradas no presente trabalho. Além disso, testou-se a efetividade do balanceamento dos dados, feito através da técnica de amostragem chamada *Undersampling*, comparando os dois modelos de *Machine Learning* em cenários com dados balanceados e desbalanceados. Para medir o desempenho dos modelos, utilizou-se a métrica da Acurácia Balanceada, considerada adequada para lidar com dados desbalanceados. A aplicação foi feita utilizando uma base de dados, obtida no *Kaggle*, que contém dados artificiais gerados através de um simulador, com transações de cartão de crédito legítimas e fraudulentas.

Palavras-chave: Detecção de fraude; Machine Learning; Floresta Aleatória; Regressão Logística; Balanceamento de dados.

ABSTRACT

With the popularization of the internet over the years, online shopping has become increasingly common, mainly due to the ease and comfort offered, as well as the various payment methods available, with credit cards being one of the main means used in transactions in electronic commerce (e-commerce). However, this convenience also attracts fraudsters who use the internet to perpetrate increasingly sophisticated scams. Thus, it is essential to effectively identify fraudulent transactions to mitigate this risk and protect legitimate buyers. One of the main challenges in fraud detection is data imbalance, as fraud is a rare event and is naturally present in a much smaller quantity than legitimate transactions, which tends to reduce the discriminatory power of the techniques. Among the various Machine Learning techniques commonly used for this purpose, two of them, Logistic Regression and Random Forest, were explored in this work. Additionally, the effectiveness of data balancing was tested using the sampling technique called Undersampling, comparing the two Machine Learning models in scenarios with balanced and unbalanced data. To measure the performance of the models, the Balanced Accuracy metric was used, considered suitable for dealing with imbalanced data. The application was carried out using a dataset obtained from Kaggle, containing artificial data generated through a simulator, with legitimate and fraudulent credit card transactions.

Keywords: Fraud detection; Machine Learning; Random Forest; Logistic Regression; Data balancing.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A ascensão do *e-commerce* trouxe inúmeras vantagens para os consumidores: a conveniência de poder comprar como e quando quiser, de forma mais confortável sem precisar sair de casa; a comparação de preços em tempo real em diversas lojas, possibilitando escolher os melhores preços e condições para aquisição de um produto ou serviço; diversas opções de pagamento são comumente aceitas, de acordo com a ClearSale (2023), como cartões de crédito, boleto bancário e as carteiras digitais. Sendo o cartão de crédito, segundo o Santander (2022), um dos meios de pagamento digital mais utilizados para transações no comércio eletrônico em todo o mundo, ele se torna também um dos mais visados por uma ameaça que assola comerciantes e compradores: a fraude.

A fraude em transações online utilizando cartões de crédito representam uma preocupação crítica para as partes envolvidas. Enquanto ser vítima de uma fraude pode impactar negativamente a confiança do consumidor no *e-commerce* de modo geral e em especial na loja em que ocorreu a situação, transações fraudulentas levam à prejuízos financeiros para os comerciantes que, além de reembolsar o valor para o cliente fraudado, muitas vezes não conseguem recuperar o produto ou serviço comercializado. Sendo assim, identificar e prevenir essas ameaças se tornou uma importante estratégia para as instituições que querem evitar prejuízos financeiros e ao mesmo tempo manter a confiança dos consumidores.

Nesse contexto, a detecção de fraudes em transações do *e-commerce* que utilizam cartão de crédito como pagamento, surge como ponto crucial. E, no atual cenário de avanços tecnológicos, a aplicação de técnicas avançadas de análise de dados, aprendizado de máquina (*Machine Learning*, em inglês) e inteligência artificial tornaram-se grandes aliadas para ajudar a identificar padrões suspeitos e comportamentos fraudulentos nessas transações, auxiliando empresas a implementarem sistemas mais eficientes de prevenção e detecção de fraudes.

Existem diversas técnicas de Machine Learning que são comumente utilizadas para esta finalidade de detecção de fraudes, dentre as quais: Redes Neurais Artificiais (Ex: Azevedo, F., 2021); Máquina de Vetores de Suporte (Ex: Pacheco Junior, 2019); Árvores de Decisão (Ex: Assis, 2023); Regressão Logística (Ex: Silva, 2022); Floresta Aleatória (Ex: Azevedo, V. e Figueira, 2020); K-Vizinhos mais Próximos (Ex: Beltran, 2019). No vigente trabalho utilizaremos duas delas, aplicando-as em uma base de dados e comparando os resultados obtidos.

1.1 OBJETIVOS

Este trabalho tem como objetivo explorar e comparar duas diferentes técnicas para detecção de fraudes em transações online que utilizam cartão de crédito como meio de

pagamento: a Regressão Logística, que segundo Agresti (2013) é o modelo mais importante para dados cuja resposta é categórica, como é o caso na detecção de fraudes, onde queremos classificar cada transação em legítima ou fraudulenta; e a Floresta Aleatória, que é um algoritmo de aprendizado de máquina e consiste na combinação de várias árvores de decisão, que de acordo com Cutler, Cutler e Stevens (2012) pode ser utilizado como um classificador em dados que a variável resposta é categórica. Ademais, serão apresentadas métricas para a comparação de desempenho destas diferentes técnicas, além de alternativas para lidar com adversidades inerentes à detecção de fraude, como o extremo desbalanceamento dos dados. Ao final deste estudo, espera-se que os leitores estejam familiarizados com os aspectos fundamentais da detecção de fraudes em transações online com cartões de crédito.

1.2 APRESENTAÇÃO DOS CAPÍTULOS

A estrutura deste trabalho está organizada da seguinte forma: no Capítulo 2 serão apresentados os conceitos iniciais com o objetivo de contextualizar o leitor sobre temas relevantes no mundo da fraude em transações online que utilizam o cartão de crédito como meio de pagamento. No Capítulo 3, temos a metodologia, abrangendo aspectos teóricos, de forma introdutória, dos modelos de *Machine Learning* que serão aplicados no trabalho (Regressão Logística e Floresta Aleatória), além de técnicas para validar, balancear e avaliar o desempenho dos modelos. Seguindo, no Capítulo 4 é feita a aplicação da metodologia descrita no Capítulo 3, utilizando uma base, retirada do *Kaggle*, que contém dados produzidos artificialmente através de um simulador. Por fim, no Capítulo 5 concluiremos sobre os resultados obtidos no presente trabalho.

2 CONCEITOS INICIAIS

Neste capítulo iremos introduzir alguns conceitos importantes para contextualizar o tema tratado neste trabalho.

2.1 E-COMMERCE

A expressão e-commerce vem de *electronic commerce*, em inglês, que significa "comércio eletrônico".

> O e-commerce é utilizado para facilitar ou comercializar produtos ou serviços online, de forma rápida e de fácil acesso para os elementos da sociedade em qualquer parte do mundo, uma vez que se trata de uma forma de comércio à distância, que permite comprar o melhor produto pelo melhor preço, reduzindo significativamente o tempo e os custos envolvidos. (Nascimento; Silva; Santos, 2009, p. 20-21)

É inegável que o comércio eletrônico mudou significativamente a forma como empresas e consumidores se envolvem no processo de compra e venda. Impulsionado pelos avanços tecnológicos presenciados no século XXI e por uma maior acessibilidade à internet no decorrer do século atual, nas palavras de Nascimento, Silva e Santos (2009, p. 23) "o e-commerce se tornou uma ferramenta essencial para toda organização que deseja realizar negócios além das fronteiras".

O e-commerce oferece uma série de vantagens para as duas partes envolvidas, tanto para os consumidores, quanto para as empresas. Além das vantagens anteriormente citadas para os consumidores na Introdução do presente trabalho (conveniência, comparação de preços de forma eficiente e métodos de pagamento), podemos citar também algumas das vantagens para os comerciantes: através dos e-commerces é possível reduzir os custos operacionais, maximizando o lucro ou diminuindo o preço final do produto para o cliente, tornando-o mais atrativo; ter um alcance global de mercado torna-se factível, possibilitando o rompimento de barreiras geográficas; personalização de marketing, identificando qual o público alvo deseja-se atingir com determinado produto ou serviço e direcionando publicidade em sites ou redes sociais para pessoas que se encaixem em tais requisitos.

Com tantas vantagens e ganhando cada vez mais adeptos, tornou-se também alvo frequente de fraudadores, que com o aumento da complexidade das operações e do volume de transações no e-commerce, buscam explorar fragilidades no sistema e realizar transações fraudulentas que prejudicam clientes e comerciantes.

2.2 CARTÃO DE CRÉDITO

As transações utilizando cartões de crédito são bastante populares (já que, como visto na Introdução, trata-se de uma das principais formas de pagamento do mundo) e envolvem várias etapas, visando o processamento correto do pagamento e a gestão do risco de cada transação.

De acordo com Gadi (2008), são 5 os agentes envolvidos no funcionamento do cartão de crédito: Portador, Estabelecimento, Adquirente (também conhecida como Credenciadora), Bandeira e Emissor.

- **Portador:** O portador é o consumidor que possui o cartão de crédito e deseja fazer a compra de um produto ou serviço utilizando-o como meio de pagamento. Ele fornece os detalhes do cartão, iniciando assim a transação.
- **Estabelecimento:** O estabelecimento é o negócio (pessoa física ou jurídica) que vende produtos ou serviços ao portador, aceitando o cartão de crédito como forma de pagamento.
- **Adquirente (ou Credenciadora):** A adquirente é a instituição financeira responsável por processar o pagamento em nome do estabelecimento. Além disso, ela é responsável por intermediar a comunicação entre os estabelecimentos, as bandeiras e os bancos emissores. Alguns exemplos de adquirentes presentes no Brasil são:
  - Cielo;
  - Rede;
  - Stone;
  - Getnet.
- **Bandeira:** As bandeiras de cartões de crédito são responsáveis por regular as políticas associadas ao uso dos cartões, incluindo regras sobre o parcelamento de compras e os tipos de estabelecimentos que aceitam seus cartões. Além disso, elas desempenham o papel de intermediárias na comunicação entre as adquirentes e os emissores, facilitando o processo de validação das transações. Algumas das bandeiras de cartão presentes do Brasil são:
  - Visa;
  - Mastercard;
  - Elo;
  - American Express.

- **Emissor:** Os emissores são as instituições financeiras (normalmente bancos) responsáveis por emitir o cartão de crédito. São responsáveis também por gerenciar o cartão, bem como estabelecer o limite de crédito concedido para o titular e autorizar (ou não) as transações solicitadas.

A Figura 1 demonstra o fluxo em uma transação digital utilizando cartão de crédito como meio de pagamento.

2.3 FRAUDE

"Fraude", segundo o dicionário Dicio (2023), pode ser definida como "qualquer ação ilícita, desonesta, ardilosa que busca enganar ou ludibriar alguém".

Trazendo para o contexto do presente trabalho, a fraude em cartão de crédito, de acordo com Beraldi (2014, p. 15) "pode ser caracterizada por uma transação que não é reconhecida pelo legítimo portador do cartão de crédito". As motivações podem ser variadas mas, a principal delas, é o ganho financeiro.

> Um dos maiores atrativos é o ganho de grandes quantias de dinheiro em um curto espaço de tempo sem exposição a grandes riscos. Isto porque raramente criminosos são descobertos e presos por um longo tempo na atual legislação brasileira para esse tipo de crime. (Beraldi, 2014, p. 15)

2.3.1 Tipos de fraude no e-commerce

Pensando em transações no comércio eletrônico, alguns dos principais tipos de fraude são:

- **Fraude deliberada:** A fraude deliberada é um ato intencional e planejado por uma ou mais pessoas com o objetivo de obter vantagens e/ou ganhos financeiros indevidamente, utilizando-se de dados de terceiros não envolvidos no esquema. Quando o titular do cartão de crédito utilizado percebe a compra feita indevidamente, ocorre a contestação da mesma e a consequente solicitação de reembolso em favor do titular (processo conhecido como chargeback). Porém, nesse meio tempo entre a compra ter sido realizada e o portador do cartão perceber e sinalizar a respeito da transação fraudulenta, o pedido muitas vezes já foi entregue ao fraudador, gerando prejuízos para o e-commerce em que foi efetuado.
- **Teste de cartão:** O teste de cartão consiste em utilizar o comércio eletrônico para, como o nome sugere, "testar" as informações dos cartões de crédito de terceiros os quais os fraudadores tem acesso ilegalmente. Ele serve para descobrir se os cartões estão ativos (e não foram bloqueados pelo titular, que é uma ação comum em caso

Figura 1 – Fluxo de uma transação digital utilizando cartão.

Fonte: Pagar.me (2021).

de perda ou roubo de cartão, por exemplo) e validar informações como a data de validade e se há limite de crédito disponível, antes de realizarem as transações em que realmente estão interessados.

- **Fraude amigável:** A fraude amigável acontece quando alguém que possui algum

tipo de relacionamento com o titular do cartão de crédito (como amigos ou familiares), e que possuem acesso aos dados do cartão, realizam uma compra sem o consentimento do dono do cartão, que quando percebe a transação e sem o devido conhecimento da mesma, realiza a solicitação de reembolso, gerando prejuízos para a loja virtual. Vale frisar que a fraude amigável nem sempre é fruto de má-fé e um exemplo clássico disso é quando um filho pequeno utiliza o cartão dos pais para realizar alguma compra (de um jogo, por exemplo) sem avisá-los e sem saber que está cometendo um ato ilícito.

- **Autofraude:** A autofraude, diferente dos demais tipos de fraude citados neste trabalho onde o portador do cartão é vítima, é realizada por má-fé do próprio proprietário do cartão de crédito. Ela acontece quando o titular realiza uma compra no e-commerce utilizando seus dados e, após o recebimento do produto, solicita o ressarcimento da compra, alegando desconhecimento. Sendo assim, o fraudador fica com o produto e com o dinheiro envolvidos na transação, causando, uma vez mais, prejuízos ao estabelecimento.

- **Fraude de interceptação:** A fraude de interceptação pode ocorrer de duas formas distintas. Na primeira delas, de posse de dados sensíveis da vítima (além do cartão), os criminosos fazem uma compra utilizando esses dados relacionados a vítima, incluindo o endereço real (entre outros) para dar maior legitimidade à transação e, após a compra ser aprovada, entram em contato com a loja em questão solicitando a mudança no endereço de entrega e conseguindo assim acesso ao produto. Uma segunda forma de materializar a fraude de interceptação consiste em, após a efetuação da compra utilizando o máximo de dados reais disponíveis da pessoa fraudada e a mesma ser aprovada, interceptar o produto antes que ele chegue ao endereço da vítima (algumas vezes com envolvimento da transportadora responsável pela entrega e em outras apenas utilizando artimanhas de engenharia social para convencer o entregador a lhe entregar a mercadoria).

Como visto acima, as fraudes que ocorrem no comércio eletrônico são, em sua maioria, dependentes do conhecimento de dados pessoais sensíveis da vítima, já que estes são essenciais para que a transação pareça legítima e obtenha a aprovação do banco emissor e do sistema antifraude (quando este existe). Algumas das formas mais utilizadas pelos fraudadores para conseguirem estes dados são:

- **Phishing:** O phishing é uma das principais técnicas utilizada por criminosos para obter ilegalmente dados pessoais sensíveis de terceiros (entre outros crimes), explorando o erro humano e o senso de urgência. No phishing, os fraudadores normalmente se passam por empresas legítimas, geralmente através de e-mails, mensagens de textos ou até ligações, criam uma falsa situação de urgência (se passando por um provedor de internet, alegando haver uma conta em atraso e

que precisa ser paga nas próximas horas ou personificando o gerente de um banco informando uma suposta transação não reconhecida na conta da vítima e que por isso ela deve compartilhar dados como número da conta e senha, por exemplo). Com isso, muitas vezes convencem as vítimas a revelarem dados pessoas sensíveis, como endereço, telefone, números do cartão, entre outros. Uma outra forma comum é convencer a pessoa a clicar em links ou abrir anexos maliciosos (no caso de e-mails ou mensagens de texto) que irão imediatamente infectar o dispositivo do usuário e assim possibilitar o acesso aos dados pessoais da vítima. E assim, de posse dos dados da pessoa fraudada, realizam transações ilegítimas.

- **Vazamento de dados:** O vazamento de dados é outra forma na qual os fraudadores podem obter acesso aos dados pessoais das vítimas. Vazamento de dados refere-se ao acesso não autorizado ou divulgação indevida de dados pessoais sensíveis, como informações financeiras ou registros de saúde. As causas de um vazamento podem ser diversas, incluindo ataques cibernéticos, falhas de segurança, erro humano, dentre outras. Posteriormente, esses dados são comercializados em grupos de aplicativos de mensagens ou fóruns na *deep web* que permitem o anonimato, possibilitando que diversos criminosos e/ou grupos de criminosos interessados possam adquiri-los e utilizá-los para cometer fraudes.

- **SIM Swap:** O golpe conhecido como SIM Swap é outra forma na qual fraudadores podem obter dados sensíveis das vítimas. O Sim Swap nada mais é do que a clonagem de um número de telefone por meio da troca do SIM card (chip). De modo geral, ele acontece da seguinte forma: o criminoso, de posse de um chip em branco e de dados pessoais do usuário (obtidos anteriormente, podendo ser, inclusive, resultado de um phishing bem sucedido ou de vazamento de dados) liga para a operadora se passando pela vítima e solicitando a ativação de um novo chip para aquele número e usando alguma desculpa como justificativa. A partir disso, é possível que o fraudador consiga muitos outros dados da vítima, como informações bancárias, senhas, informações de contatos e acesso à aplicativos, além de tornar possível burlar a autenticação de dois fatores via SMS (quando ativada), já que o código verificador chegará para o fraudador no número que foi clonado, possibilitando a invasão de contas de redes sociais, aplicativos (incluindo os de lojas do e-commerce), e-mails, etc.

- **Invasão de contas:** A invasão de contas também pode ser uma forma de criminosos conseguirem dados pessoais dos alvos. Através da invasão de uma conta em uma loja do e-commerce, por exemplo, eles podem obter acesso ao nome completo, data de nascimento, CPF, endereço, telefone e, em alguns casos, até mesmo a dados do cartão de crédito da vítima. Com todas essas informações, é possível que o fraudador consiga, inclusive, invadir outras contas do usuário (como e-mails e redes sociais), além de utilizar os dados obtidos para realizar transações fraudulentas. Existem

diversas técnicas utilizadas para se invadir uma conta, como o phishing, tentativas de explorar falhas de segurança de sistemas ou aplicativos e o chamado ataque de força bruta, que é quando os criminosos usam programas automatizados para testar várias combinações de senha até encontrarem a correta e assim concretizar o acesso indevido a conta.

- **Sites falsos:** O roubo de dados através de sites falsos é mais uma forma de obtenção de dados de maneira indevida. Ele funciona basicamente da seguinte forma: criminosos criam páginas da web falsas que possuem visual idêntico a sites legítimos famosos, com o objetivo de enganar as vítimas e coletar informações pessoais e confidenciais (como dados de login e senha, informações do cartão de crédito, entre outros). Normalmente a forma de divulgação desses sites falsos é através de e-mails de phishing e anúncios em redes sociais, muitas vezes atrelados a ofertas tentadoras de produtos sabidamente cobiçados (como computadores e smartphones). Uma vez com os dados coletados, os fraudadores possuem informações suficientes para efetuar transações fraudulentas ou partirem para outros crimes (como a invasão de contas) em busca de mais informações da vítima.

Além das formas de roubo de dados detalhadas acima, existem diversas outras. Sendo assim, algumas dicas simples e que podem ser úteis para evitar ser vítima de cibercriminosos são descritas abaixo:

1. Como relatado acima, existem golpes que buscam invadir contas e conseguir acesso à dados pessoais das vítimas explorando vulnerabilidades das senhas utilizadas pelos usuários (como o ataque de força bruta). Por isso, é importante entender as senhas como uma barreira de proteção e evitar senhas curtas (menos de 8 caracteres), óbvias (como o próprio nome), sequenciais (12345, por exemplo) ou com informações pessoais (como data de nascimento ou nome de um animal de estimação, já que outras pessoas podem "adivinhar" com facilidade), pois esse tipo de senha deixa a conta exposta. Por outro lado, senhas longas e complexas, fruto da combinação de letras (maiúsculas e minúsculas), números e caracteres especiais, ajudam a proteger as contas e aplicativos, dificultando o acesso não autorizado e ataques de força bruta. Vale sempre lembrar que senhas são de uso pessoal e intransferível, e que em hipótese alguma devem ser compartilhadas com quem quer que seja.

2. Estar sempre atento aos detalhes é fundamental para evitar ser vítima de crimes na internet. Por exemplo, em caso de e-mails, verificar com cuidado o remetente, já que os criminosos buscam ser o mais verossímil possível mas, alguns pequenos detalhes podem "denunciar" que não se trata de um contato oficial da empresa na qual eles buscam se passar. Além disso, o tipo de linguagem utilizado no corpo do e-mail

muitas vezes não condiz com o linguajar mais adequado para o mundo corporativo, sendo também um indicativo de que algo está errado.

3. Sempre desconfiar de e-mails, mensagens ou ligações inesperadas, principalmente as que solicitam ações (como pagar um boleto ou clicar em um link) ou o fornecimento de dados confidenciais, sempre com urgência e para evitar consequências "catastróficas". São grandes as chances de se tratar de um phishing, pois empresas sérias, cientes dos crimes cibernéticos contemporâneos, não solicitam este tipo de ação ou informação através destes canais.

4. É importante a utilização de um *software* antivírus, independente do tipo de dispositivo utilizado (seja móvel ou um *desktop*, por exemplo), pois eles são barreiras importantes e que podem evitar o sucesso dos fraudadores. Além disso, é importante manter sistema operacional, *softwares* e aplicativos devidamente atualizados, pois as empresas fornecedoras costumam corrigir vulnerabilidades conhecidas através de atualizações e, com isso, evitar que elas sejam exploradas por criminosos.

5. Quando disponível, é bastante recomendado a utilização da chamada autenticação em dois fatores. A autenticação em dois fatores adiciona uma camada extra de segurança as contas em que está aplicada, fazendo com que, mesmo que um terceiro tenha acesso aos dados de login e senha do usuário, não seja possível invadir a conta sem antes confirmar um código único presente apenas no dispositivo da vítima. As formas mais comuns de obtenção do código de autenticação são através do SMS e de aplicativos especializados que alteram o código (ou *token*) automaticamente de tempos em tempos (os principais são o Google Authenticator e o Microsoft Authenticator). Até pelo supracitado golpe de SIM Swap, é importante dar preferência aos aplicativos especializados em desfavor do código enviado via SMS.

2.3.2 Chargeback

É praticamente impossível falar de fraudes no e-commerce sem citar o processo de chargeback. O chargeback acontece, de acordo com a ClearSale (2021), "quando uma cobrança é contestada pelo titular do cartão e o valor precisa ser devolvido". Vale ressaltar que, embora a fraude seja o motivo mais comum para a contestação de uma transação por parte do portador junto ao emissor, ela não é o único e o chargeback também pode acontecer por outros motivos, como por exemplo:

- Desacordo comercial: pode ocorrer quando há alguma divergência entre o produto comprado pelo titular do cartão e o que foi efetivamente entregue pelo lojista (uma alteração na cor do produto, por exemplo), quando há atrasos na entrega do produto ou o mesmo foi entregue com algum defeito. Nesse caso, embora reconheça ter feito a transação, acontece a solicitação de chargeback por insatisfação do portador junto ao

estabelecimento. Embora não haja má-fé neste tipo de ocorrência, convém destacar que ainda assim gera prejuízos para o estabelecimento, como de taxas envolvidas no processo de chargeback, além de outras possibilidades dependendo da política praticada pela empresa e do acordado no momento da compra (um exemplo são os custos do transporte, caso este por seja por conta do estabelecimento, tanto do trajeto até o endereço do cliente, quanto do caminho reverso até chegar novamente ao estoque do comerciante).

- Erro de processamento: outro motivo de chargeback que não está ligado a transações fraudulentas é quando ocorre algum erro de processamento durante a operação, como uma cobrança duplicada ou com valor incorreto.

O Chargeback é um mecanismo valioso de proteção para os portadores de cartões de crédito em caso de cobranças indevidas, transações fraudulentas ou que não foram entregues exatamente nas condições estabelecidas no momento da compra. No entanto, é uma ferramenta que também pode ser usada por pessoas desonestas para adquirir produtos ou serviços sem pagar por eles, no tipo de fraude conhecido como "autofraude" (detalhada na subseção anterior).

De maneira geral, o fluxo de contestação de compra (ou solicitação de chargeback) acontece da seguinte forma:

1. **Portador não reconhece a compra** O titular do cartão de crédito não reconhece alguma transação pela qual ele está sendo cobrado e decide contestá-la junto ao emissor do cartão.

2. **Contato com o emissor** O titular entra em contato com o banco ou instituição que emitiu o cartão, alegando desconhecer determinada compra e solicitando o cancelamento da cobrança da mesma. A partir da solicitação do cliente, o emissor avalia a abertura do processo de disputa de chargeback e, caso julgue procedente, comunica o fato para a bandeira do cartão.

3. **Bandeira da seguimento ao fluxo** Ao receber a comunicação do emissor a respeito da solicitação de chargeback, a bandeira repassa a informação para as adquirentes.

4. **Adquirente media a disputa** Ao receber a informação de que o cliente deseja cancelar a compra, a adquirente informa ao estabelecimento responsável pela venda e pode solicitar mais informações a respeito da transação contestada, além de abrir uma disputa onde o estabelecimento tem a possibilidade de se defender da contestação caso entenda que seja indevida.

5. **Emissor toma a decisão** Após reunir todas as informações pertinentes ao processo, a adquirente repassa as informações para a bandeira, que por sua vez as encaminha

para o emissor, que é o responsável por tomar a decisão. A partir das informações, o emissor decide se a contestação é válida ou não. Caso entenda que é válida, o titular do cartão não será cobrado pela transação e o estabelecimento será responsabilizado, devendo arcar com o valor. Porém, caso o emissor decida que a contestação não tem fundamento, o titular será cobrado pelo valor da transação, enquanto o estabelecimento não terá qualquer custo.

A Figura 2 ilustra, de forma simplificada, o fluxo do chargeback.

Figura 2 – Fluxo do processo de Chargeback.

Fonte: Equals (2021).

3 METODOLOGIA

Este capítulo apresenta a metodologia empregada na condução do presente trabalho, detalhando os procedimentos e técnicas utilizadas para se atingir os objetivos declarados na Seção 1.1. Inicialmente, abordaremos os dois modelos de *Machine Learning* (Regressão Logística e Floresta Aleatória) que serão aplicados e, posteriormente, serão apresentadas técnicas para validar, balancear e avaliar o desempenho dos modelos. Por fim, uma apresentação dos equipamentos e ferramentas utilizadas para a aplicação descrita no Capítulo 4.

3.1 REGRESSÃO LOGÍSTICA

A Regressão Logística é uma técnica amplamente utilizada para modelar a probabilidade de ocorrência de um evento binário (como sim/não, sucesso/falha, fraude/não fraude, etc.), com base em uma ou mais variáveis independentes. Comumente, é utilizada a codificação 1/0, onde 1 indica a ocorrência do evento de interesse (sucesso) e o 0 indica a não ocorrência do evento de interesse (falha). Por se tratar de um modelo estudado na graduação de Estatística, serão apresentados apenas conceitos básicos, além de referências para os que desejarem uma abordagem mais detalhada a respeito do modelo.

Em linhas gerais, a Regressão Logística modela a relação entre um conjunto de variáveis independentes $\boldsymbol{X} = (X_1, X_2, \ldots, X_p)^T$ e uma variável resposta binária Y. O objetivo é estimar a probabilidade $p(X)$, que é a probabilidade de que Y seja igual a 1, dadas as variáveis independentes X, conforme a seguir:

$$p(X) = \Pr(Y = 1|X).$$

De acordo com James *et al.* (2013), de forma a garantir que a função $p(X)$ produza valores no intervalo entre 0 e 1, a Regressão Logística usa a função logística, dada pela seguinte expressão,

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\ldots+\beta_p X_p}}, \tag{3.1}$$

onde $\beta_0, \beta_1, \ldots, \beta_p$ são os coeficientes de regressão que devem ser estimados a partir dos dados.

Para estimar os coeficientes $\boldsymbol{\beta} = (\beta_0, \beta_1, \ldots, \beta_p)^T$, segundo James *et al.* (2013), é preferível usar o método da máxima verossimilhança por ter melhores propriedades estatísticas. O método busca os valores dos parâmetros do modelo que tornam os dados observados mais verossímeis, através da maximização da função de verossimilhança. Para a Regressão Logística, considerando uma amostra aleatória com n observações, a função de verossimilhança é expressa por:

$$L(\beta) = \prod_{i=1}^{n} p(X_i)^{Y_i} \left[1 - p(X_i)\right]^{1-Y_i}. \tag{3.2}$$

Hosmer, Lemeshow e Sturdivant (2013) indicam ser mais fácil, matematicamente, trabalhar com a maximização da função de log-verossimilhança, definida por:

$$\ell(\beta) = ln\left[L(\beta)\right] = \sum_{i=1}^{n} \left\{Y_i \; ln\left[p(X_i)\right] + (1 - Y_i) \; ln\left[1 - p(X_i)\right]\right\}. \tag{3.3}$$

Em seguida, é necessário derivar (3.3) em relação aos parâmetros $\beta_0, \ldots, \beta_p$ e igualar o resultado a 0. No caso da Regressão Logística, segundo Hosmer, Lemeshow e Sturdivant (2013), são obtidas equações não lineares e é necessário o apoio de métodos iterativos para se chegar ao estimador de máxima verossimilhança.

Para um entendimento mais aprofundado sobre aspectos teóricos do modelo de Regressão Logística, podem ser consultados Hosmer, Lemeshow e Sturdivant (2013); Casella e Berger (2002) e Agresti (2013).

3.1.1 Seleção de variáveis

No presente trabalho, a seleção de variáveis preditoras, no caso do modelo de Regressão Logística, será feita utilizando o conceito de importância das variáveis. Este conceito considera que, quanto maior o valor absoluto de z, maior a importância da variável para o modelo. Vale mencionar que o valor z é obtido a partir da divisão do valor do coeficiente estimado pelo seu erro padrão.

A partir do valor z e considerando um nível de significância de 5%, serão tidas como variáveis significativas para o modelo as que possuírem o valor absoluto de z maiores do que 1,96. Isto porque, considerando a distribuição normal padrão, se o valor absoluto de z for maior que 1,96, então ele pertence à região de rejeição da hipótese nula de que o coeficiente estimado para uma determinada variável seja 0, significando que a variável é relevante para o modelo.

3.2 FLORESTA ALEATÓRIA

A Floresta Aleatória (*Random Forest*, em inglês) é uma técnica de *Machine Learning* amplamente empregada em tarefas de classificação, como no caso deste trabalho, em que queremos classificar transações online utilizando cartão de crédito em legítimas ou fraudulentas. Este método foi introduzido por Breiman (2001) e é baseado na combinação de múltiplas árvores de decisão (*decision trees*, em inglês), levando a um modelo mais eficaz e robusto, não sendo tão sensível quanto árvores de decisão individualmente são.

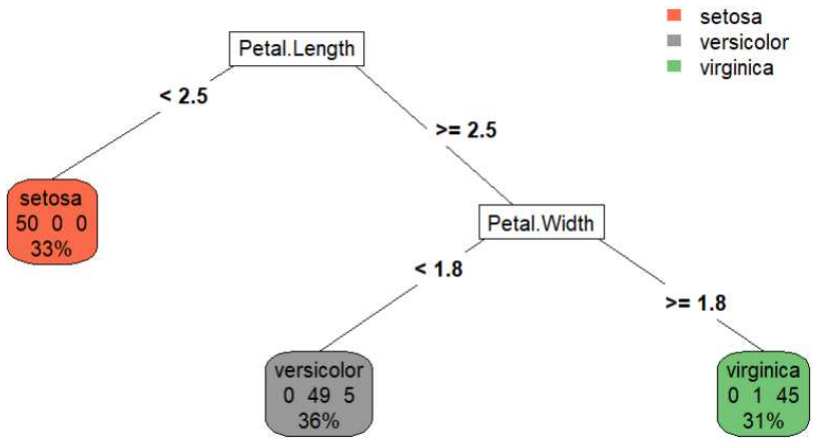
Como mencionado no parágrafo anterior, a Floresta Aleatória é composta por um conjunto de árvores de decisão e, por este motivo, vale a pena entender melhor o

funcionamento das árvores de decisão individualmente antes de falarmos das Florestas Aleatórias em si. As árvores de decisão, como a exemplificada na Figura 3, são construídas a partir de nós, possuindo a seguinte estrutura:

- **Nó raiz**: trata-se do nó inicial da árvore (no exemplo da Figura 3, o nó raiz é *Petal.Length*).
- **Nó de decisão**: são os nós que dividem o conjunto de dados com base em uma condição ou regra (na Figura 3, o nó com a variável *Petal.Width* é um nó de decisão).
- **Nó folha**: no caso de árvores de classificação, são os que representam a classe final (previsão) que será atribuído a amostra após todas as divisões da árvore (no caso da Figura 3, são os nós com os valores *setosa*, *versicolor* e *virginica*).

A escolha sobre qual a variável a ser utilizada em cada nó (seja o nó raiz ou algum nó de decisão) passa pelo conceito de "pureza" dos dados, medido através de métricas como o Índice de Gini e a Entropia (mais detalhes no Capítulo 8 de James *et al.*, 2013). Além da variável em si, a pureza dos dados também possui papel fundamental no ponto de corte que será utilizado na divisão do nó, que pode ser um valor (variáveis contínuas) ou uma categoria (variáveis categóricas), sempre avaliando a qualidade da divisão em termos de pureza (utilizando Gini ou a Entropia), buscando maximizar a separação entre as classes.

Figura 3 – Exemplo de uma Árvore de Decisão.



Fonte: Elaborada pelo autor (2024).

A Floresta Aleatória é formada por um conjunto de árvores de decisão, cada uma gerada a partir de diferentes amostras, extraídas de forma aleatória e com reposição, do conjunto de dados, técnica essa conhecida como *Bootstrap*. Durante a construção de cada

árvore de decisão, em cada nó, são utilizadas como candidatas a variável de divisão (*split*) apenas um subconjunto de tamanho m, selecionado aleatoriamente, dentre as p variáveis preditoras disponíveis, com $m < p$, introduzindo mais aleatoriedade ao processo quando comparado, por exemplo, ao *Bagging* (*Bootstrap Aggregation*), que é outra técnica de *Machine Learning* baseada em múltiplas árvores de decisão. De acordo com James *et al.* (2013, p. 320, tradução nossa) "podemos pensar neste processo como a descorrelação das árvores"[1], o que, ao introduzir mais aleatoriedade ao processo, proporciona melhorias para o modelo de Floresta Aleatória em relação ao *Bagging*.

Considerando o modelo de Floresta Aleatória para classificação, a combinação dos resultados das múltiplas árvores de decisão é feito por voto majoritário. Ou seja, cada árvore irá fazer individualmente uma previsão e atribuir a amostra a uma das classes possíveis (por exemplo, fraude ou não fraude). Ao final do processo, a classe que receber a maioria dos votos dentre os votos de todas as árvores de decisão será a escolhida como a previsão final. Esta estratégia reduz o risco de *overfitting*, que é quando o modelo se ajusta muito bem ao conjunto de dados utilizado na sua criação, mas não consegue generalizar para outros dados e se mostra ineficaz quando se depara com dados nunca visto antes, problema este que é comum em árvores de decisão individuais, as quais James *et al.* (2013) apontam como um modelo não robusto e sensível a pequenas mudanças nos dados.

Quanto a definição do valor do hiperparâmetro m, apresentado anteriormente nesta mesma seção, Hastie, Tibshirani e Friedman (2009) apontam que o valor padrão de m, para o caso de modelos de Floresta Aleatória de classificação, é $m = \sqrt{p}$. Já Kuhn e Johnson (2013), inicialmente, recomendam que sejam ajustados para m cinco valores espaçados igualmente no intervalo de 2 a p, com p sendo o número de variáveis preditoras, e, com base nos resultados obtidos, decidir o melhor valor para m. Ainda de acordo com Kuhn e Johnson (2013), o hiperparâmetro m é denotado como m_{try}. Izbicki e Santos (2020) citam que m (ou m_{try}) pode ser escolhido através da validação cruzada e que, nos casos em que $m = p$, o modelo de Floresta Aleatória se reduz ao *Bagging*. A Figura 4 traz o exemplo do funcionamento de um modelo de Floresta Aleatória para fins de classificação.

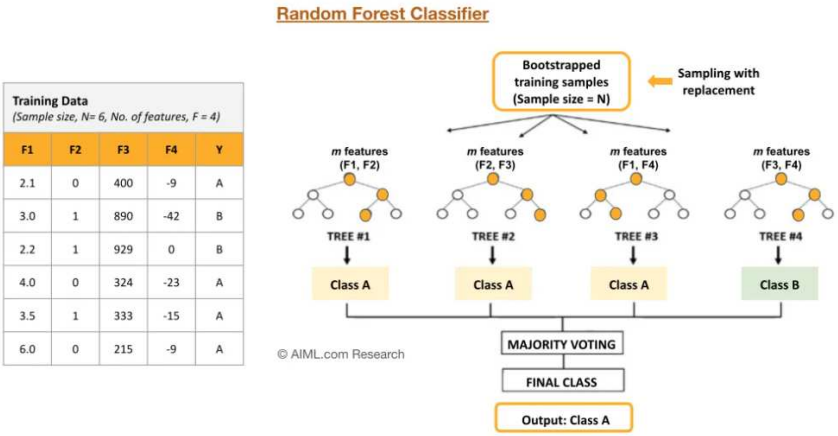
Para mais detalhes a respeito de resultados teóricos do modelo de Floresta Aleatória, veja Breiman (2001).

3.3 VALIDAÇÃO

Uma etapa importante no processo da construção e avaliação de modelos de *Machine Learning* é a validação, visando a criação de modelos que sejam mais robustos e capazes de generalizar os resultados obtidos. Além disso, um modelo devidamente validado, será capaz de fazer melhores previsões em dados não vistos, evitando problemas como o sobreajuste (*overfitting*) e garantindo uma maior confiabilidade ao modelo.

[1] No original: "We can think of this process as decorrelating the trees".

Figura 4 – Exemplo do funcionamento de um modelo de classificação de Floresta Aleatória.



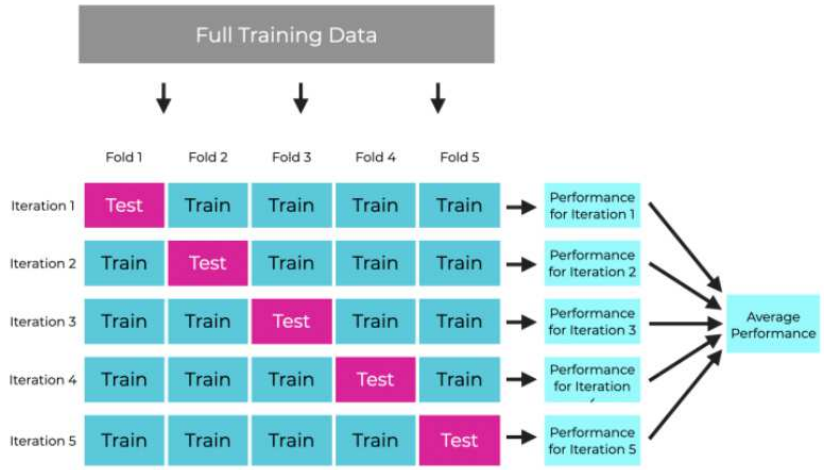
Fonte: AIML (2023).

Existem algumas técnicas de validação de modelo, dentre as mais utilizadas estão a divisão treino-teste (*data splitting*) e técnicas de reamostragem, como *bootstrap* e *k-fold cross-validation*. A técnica que utilizaremos neste trabalho será a *k-fold cross-validation*, detalhada a seguir, porém, mais detalhes sobre técnicas de validação podem ser encontrados em Bramer (2016); Kuhn e Johnson (2013) e Hastie, Tibshirani e Friedman (2009).

Falando especificamente da *k-fold cross-validation*, a técnica consiste em dividir o conjunto de dados em k subconjuntos (ou *folds*) mutuamente exclusivos e de aproximadamente mesmo tamanho. Na sequência, o modelo será ajustado k vezes, utilizando k-1 subconjuntos para treinamento e o subconjunto restante será utilizado para teste. O processo é feito de forma com que em cada um dos k ajustes do modelo, o subconjunto separado para ser utilizado como teste seja diferente e ao final do procedimento todos os k subconjuntos tenham sido utilizados tanto para treino, quanto para teste. A cada repetição (ou iteração), o modelo é avaliado de acordo com uma métrica apropriada para o problema estudado e, ao final das k iterações, avalia-se a performance geral do modelo, normalmente calculando a média do desempenho em cada iteração. A Figura 5 ilustra o processo descrito.

Um último ponto que vale ser destacado a respeito da *k-fold cross-validation* é o valor de k. Tanto Kuhn e Johnson (2013), quanto Hastie, Tibshirani e Friedman (2009) apontam os valores de $k = 5$ e $k = 10$ como os mais comuns, não existindo uma fórmula (ou regra) para a escolha de k. Diante disso, como não há um consenso, neste trabalho aplicaremos a técnica de *k-fold cross-validation* com $k = 5$.

Figura 5 – Ilustração da *k-fold cross-validation* com k = 5.



Fonte: Ebner (2023).

3.4 BALANCEAMENTO DOS DADOS

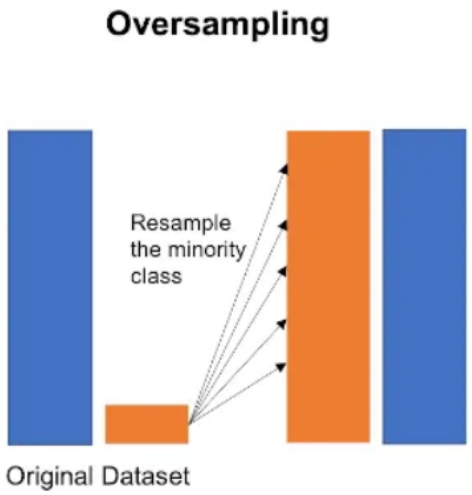
Um problema característico no ajuste de modelos que possuem o objetivo de detectar fraudes é o extremo desbalanceamento dos dados, já que as transações fraudulentas, de forma geral, estarão presentes no conjunto de dados em uma proporção muito menor do que as transações legítimas.

A principal implicação de se treinar um modelo de *Machine Learning* utilizando uma base de dados com grande desequilíbrio entre as classes é que o modelo tende a apresentar um baixo desempenho em relação à classe minoritária. De acordo com Hasanin et al (2019), os algoritmos de *Machine Learning* não conseguem diferenciar com precisão a classe minoritária da majoritária em situações que o conjunto de dados sofre com um grave desequilíbrio entre as classes.

Uma forma de lidar com esse desbalanceamento é utilizar técnicas de amostragem para equilibrar a proporção das classes presentes na base de dados. Segundo Dittman et al (2014), a amostragem de dados modifica o conjunto de dados ao adicionar ou remover instâncias para atingir uma proporção de classes mais equilibrada. Dentre essas técnicas de amostragem, temos a sobreamostragem (*oversampling*) e a subamostragem (*undersampling*).

- **Sobreamostragem** (*oversampling*): na sobreamostragem, o objetivo é aumentar a classe minoritária de forma com que se obtenha o equilíbrio entre as classes. O aumento da classe menos presente nos dados se dá através da duplicação aleatória das observações desta classe, até que atinja o tamanho desejado para a classe minoritária. A Figura 6 ilustra este processo.

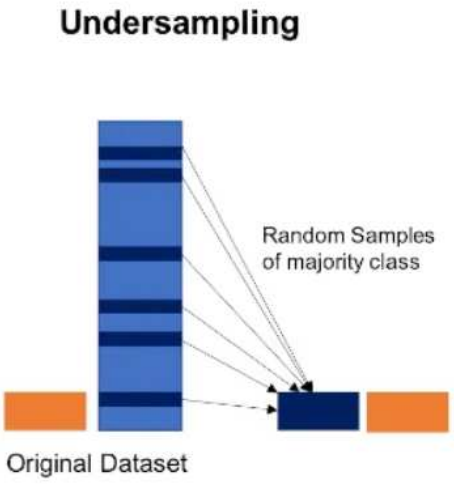
Figura 6 – Ilustração da técnica de sobreamostragem.



Fonte: Jayawardena (2020).

- **Subamostragem** (*undersampling*): a ideia na subamostragem é reduzir a classe majoritária até se atingir o equilíbrio entre as classes. A redução da classe dominante é feita retirando-se observações, através de uma amostra aleatória simples, até que se tenha o tamanho desejado para a classe majoritária, conforme ilustrado na Figura 7.

Figura 7 – Ilustração da técnica de subamostragem.



Fonte: Jayawardena (2020).

No caso específico deste trabalho, no Capítulo 4, mais precisamente na Seção 4.4 onde será feita a aplicação utilizando uma base de dados contendo transações legítimas e fraudulentas, o balanceamento dos dados será feito utilizando a subamostragem (*undersampling*). Isso porque, tanto Dittman et al (2014), quanto Hasanin et al (2019), recomendam

a utilização da subamostragem em vez da sobreamostragem, já que na subamostragem há uma redução do conjunto de dados, o que consequentemente reduz o custo computacional para a implementação, sem haver perda de performance. Além disso, como a base de dados que será utilizada neste trabalho possui um tamanho bastante considerável, é possível utilizar a subamostragem para balancear os dados e ainda assim manter uma base com um tamanho satisfatório.

3.5 MEDIDAS DE DESEMPENHO

Um ponto fundamental na construção de modelos de *Machine Learning* é a avaliação de desempenho do modelo. Utilizar métricas adequadas para o contexto dos dados é essencial para garantir um modelo robusto e que seja generalizável. Discutiremos a seguir, as principais métricas utilizadas para avaliar a eficácia dos modelos cujo objetivo final é a classificação.

3.5.1 Matriz de confusão

De acordo com Izbicki e Santos (2020, p. 138), "é comum avaliar o desempenho de um classificador com base em matrizes de confusão". A matriz de confusão é uma tabela que resume o desempenho do modelo de classificação, trazendo os verdadeiros positivos (VP), verdadeiros negativos (VN), falsos positivos (FP) e falsos negativos (FN), conforme o exemplo a seguir na Tabela 1.

Tabela 1 – Exemplo de uma matriz de confusão quando a variável resposta possui duas classes.

		Real	
		Negativo	Positivo
Predito	Negativo	VN	FN
	Positivo	FP	VP

Fonte: Elaborada pelo autor (2024).

A partir da matriz de confusão, é possível calcular várias métricas que ajudam a avaliar o desempenho do modelo de classificação. A seguir, abordaremos algumas das principais.

- **Acurácia:** é a proporção de previsões corretas feitas pelo modelo, em relação ao total de previsões.

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN}. \tag{3.4}$$

  Apesar de ser uma das medidas de avaliação de modelo mais utilizadas, a acurácia não é uma boa métrica para o contexto de detecção de fraudes. Isso porque, por

conta do grande desbalanceamento entre as classes da variável resposta, uma acurácia elevada pode ser obtida e, ainda assim, o modelo não ter qualquer utilidade na prática. Este fenômeno é conhecido como o Paradoxo da Acurácia e mais detalhes podem ser encontrados em Azank (2020).

- **Acurácia balanceada:** é uma alternativa, recomendada por Peixoto (2023) e Freire (2019), para o caso de dados desbalanceados. Isto ocorre, de acordo com Peixoto (2023), porque a métrica leva em consideração em seu cálculo, tanto a taxa de verdadeiros positivos, quanto a taxa dos verdadeiros negativos.

$$\text{Acurácia Balanceada} = \frac{1}{2}\left(\frac{VP}{VP+FN} + \frac{VN}{VN+FP}\right). \tag{3.5}$$

A fórmula da acurácia balanceada pode ser reescrita da seguinte forma.

$$\text{Acurácia Balanceada} = \frac{1}{2}(\text{Sensibilidade} + \text{Especificidade}). \tag{3.6}$$

Por conta de ser uma métrica mais recomendada para o caso de dados desbalanceados, usaremos a acurácia balanceada como a principal métrica de avaliação de desempenho do modelo neste trabalho.

- **Sensibilidade:** também chamada de *recall*. A sensibilidade é uma medida que avalia a capacidade do modelo de detectar corretamente os casos positivos. No contexto de detecção de fraudes, é uma métrica que avalia a capacidade do modelo de identificar as transações fraudulentas (ou seja, de todas as transações fraudulentas, qual o percentual que o modelo consegue identificar como fraudulenta?).

$$\text{Sensibilidade} = \frac{VP}{VP+FN}. \tag{3.7}$$

- **Especificidade:** é uma métrica que avalia a capacidade do modelo de identificar de forma correta os casos negativos. Dentro do contexto de fraude, a especificidade avalia a capacidade do modelo em identificar transações legítimas corretamente (ou seja, de todas as transações legítimas, qual a taxa que o modelo classifica como legítima?).

$$\text{Especificidade} = \frac{VN}{VN+FP}. \tag{3.8}$$

- **Predição de valores positivos:** também conhecida como precisão. A predição de valores positivos traz o percentual de acertos do modelo em relação a tudo que ele classificou como positivo. No contexto de fraude, a predição de valores positivos indica a taxa de acertos do modelo dentro do que foi classificado como fraude por

ele (ou seja, de todas as transações que o modelo classifica como fraudulentas, qual o percentual que realmente é fraudulenta?).

$$\text{Predição de valores positivos} = \frac{VP}{VP + FP}. \tag{3.9}$$

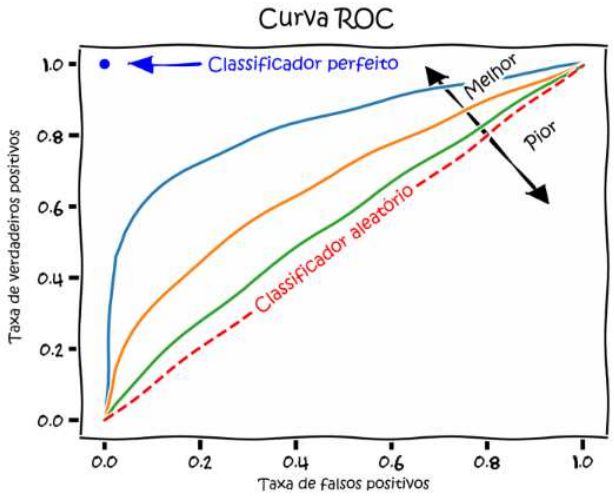
- **Predição de valores negativos:** é uma medida que avalia a taxa de acertos do modelo em classificar os casos negativos. Trazendo para o contexto de fraude, a predição de valores negativos é a taxa de acertos do modelo dentro de tudo que ele classificou como transação legítima (ou seja, de todas as transações classificadas como legítimas pelo modelo, qual o percentual que realmente é legítima?).

$$\text{Predição de valores negativos} = \frac{VN}{VN + FN}. \tag{3.10}$$

3.5.2 Curva ROC

A curva ROC (do inglês *Receiver Operating Characteristic*) é uma ferramenta gráfica usada para avaliar o desempenho de modelos de classificação. Ela traça a relação entre a taxa de verdadeiros positivos (sensibilidade) e a taxa de falsos positivos (1 - especificidade) em vários pontos de cortes possíveis e, com isso, é possível analisar graficamente a troca entre a sensibilidade e a especificidade em diferentes cortes. Um modelo perfeito teria uma curva ROC que passa pelo ponto (0,1), ou seja, com 100% de verdadeiros positivos e 0% de falsos positivos, enquanto que um modelo aleatório teria uma curva ROC sendo uma reta com inclinação de 45 graus, indicando um modelo sem uso. A Figura 8 a seguir ilustra o descrito.

Figura 8 – Exemplo de uma curva ROC.



Fonte: Mariano (2021).

Uma medida capaz de resumir a curva ROC é a área abaixo da curva (*Area Under the Curve* - AUC, em inglês). A AUC varia no intervalo entre 0 e 1 e, quanto maior o valor da AUC, melhor a capacidade de discriminação do modelo, o que é algo desejável.

3.6 AMBIENTE DE DESENVOLVIMENTO

Os seguintes equipamentos e ferramentas foram utilizados para a execução da parte prática deste trabalho (descrita em detalhes no Capítulo 4).

- **Sistema Operacional**: Windows 11 (versão 23H2);
- **Especificações do dispositivo**:
  - Processador (CPU): AMD Ryzen 7 6800H;
  - Memória RAM: 16 GB DDR5 4800MHz;
  - Placa Gráfica (GPU): NVDIA GeForce RTX 3070 Ti;
- **Linguagem de Programação**: R versão 4.3.1 (R Core Team, 2023);
- **Ambiente de Desenvolvimento Integrado**: RStudio versão 2023.6.2.561 (Posit Team, 2023);
- **Pacotes Utilizados**:
  - *tidyverse* (Wickham *et al.*, 2019) - versão 2.0.0 ;
  - *caret* (Kuhn, 2008) - versão 6.0-94;
  - *pROC* (Robin *et al.*, 2011) - versão 1.18.4;
  - *randomForest* (Liaw e Wiener, 2002) - versão 4.7-1.1.

4 APLICAÇÃO

A base de dados utilizada no trabalho foi retirada do *Kaggle*, que é uma conhecida plataforma online dedicada à ciência de dados e ao aprendizado de máquina, onde é possível encontrar, entre outras coisas, bases de dados públicos.

4.1 DESCRIÇÃO DOS DADOS

A base em questão trata-se de uma base sintética, criada por Shenoy e Harris (2020), que contém simulações de transações utilizando cartão de crédito, incluindo tanto transações legítimas, como fraudulentas, e foi produzida através do gerador de dados *Sparkov*. Vale ressaltar que foi necessário recorrer a uma base artificial devido ao fato de que bases reais envolvendo transações de cartão de crédito contém muitos dados sensíveis dos usuários, como e-mail, endereço, telefone e os próprios dados do cartão, sendo o acesso a essas bases bastante limitado por questões de segurança, visando respeitar a privacidade dos usuários e evitar vazamentos.

Composta por 1.852.394 observações e de 23 variáveis, a base foi previamente dividida em uma base de treino e uma base de teste. Esse procedimento de dividir a base em 2 grupos é uma técnica bastante utilizada na construção de modelos com o intuito de evitar o sobreajuste (*overfitting*). Sendo assim, é comum a divisão da base de dados em uma base que será utilizada para construir o modelo (base de treino) e uma base que será utilizada para testar o modelo com dados desconhecidos por ele (base de teste).

As 23 variáveis presentes na base de dados são:

- **index:** identificador único para cada linha;
- **trans_date_trans_time:** data e hora da transação - variável quantitativa;
- **cc_num:** número do cartão de crédito utilizado na transação;
- **merchant:** nome do comerciante;
- **category:** categoria do comerciante - variável qualitativa;
- **amt:** valor (em dólares) da transação - variável quantitativa;
- **first:** primeiro nome do titular do cartão de crédito;
- **last:** último nome do titular do cartão de crédito;
- **gender:** gênero do titular do cartão de crédito - variável qualitativa;
- **street:** endereço do titular do cartão de crédito;

- **city:** cidade do titular do cartão de crédito;
- **state:** estado do titular do cartão de crédito;
- **zip:** código de endereçamento postal (CEP) do titular do cartão de crédito;
- **lat:** latitude do titular do cartão de crédito - variável quantitativa;
- **long:** longitude do titular do cartão de crédito - variável quantitativa;
- **city_pop:** população da cidade do titular do cartão de crédito - variável quantitativa;
- **job:** trabalho do titular do cartão de crédito;
- **dob:** data de nascimento do titular do cartão de crédito - variável quantitativa;
- **trans_num:** código único usado para identificador a transação;
- **unix_time:** tempo da transação contado em UNIX [1] - variável quantitativa;
- **merch_lat:** latitude do comerciante - variável quantitativa;
- **merch_long:** longitude do comerciante - variável quantitativa;
- **is_fraud:** trata-se de uma coluna que sinaliza se a transação é legítima (0) ou fraudulenta (1) - variável qualitativa.

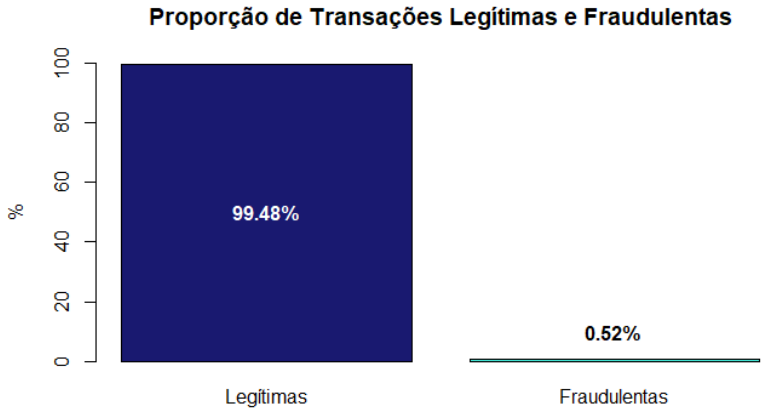
4.2 ANÁLISE DESCRITIVA DOS DADOS

Na presente seção faremos uma análise descritiva dos dados, buscando explorar e encontrar padrões que possam ser úteis para explicar os dados e possíveis *insights* para a construção do modelo. Vale destacar que não há valores faltantes (*Not Available* ou *NA*, em inglês) e temos, portanto, uma base completa. Os códigos utilizados nesta seção encontram-se no Apêndice A.

- **is_fraud:** variável dicotômica que determina se a transação é legítima (0) ou fraudulenta (1) e, sendo assim, a variável resposta da base de dados. Uma característica comum neste tipo de problema de detecção de fraude é o desbalanceamento, isto é, a proporção de uma das categorias da variável é muito superior a outra. Apesar de utilizarmos uma base sintética, esta caraterística foi preservada e a Figura 9 deixa evidente este desbalanceamento:

[1] *UNIX time* é uma representação de tempo que consiste na contagem dos segundos decorridos desde 01/01/1970 00:00:00 UTC.

Figura 9 – Variável *is_fraud*.



Fonte: Elaborada pelo autor (2023).

- **amt:** como dito anteriormente, estamos interessados em detectar quando uma transação é legítima ou fraudulenta. Uma variável que pode ter um papel importante nesse contexto é a variável *amt* (abreviação de *amount*, que significa "valor" em português), variável esta que contém os valores de cada transação (em dólar). Sendo uma variável quantitativa, a Tabela 2 traz um resumo dos dados:

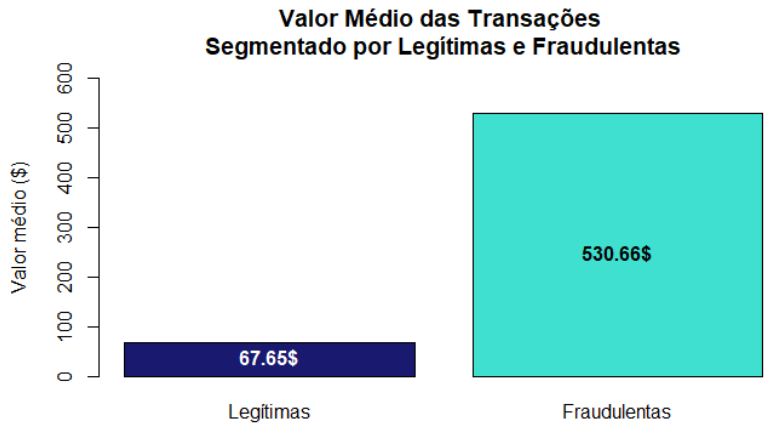
Tabela 2 – Resumo numérico da variável *amt*.

	Mínimo	Mediana	Média	Máximo
amt	1,00	47,45	70,06	28.948,90

Fonte: Elaborada pelo autor (2023).

A Figura 10 traz a média dos valores gastos em transações, segmentando por legítimas e fraudulentas. Fica claro que a média da quantia gasta em transações fraudulentas é bastante superior às legítimas, o que torna esta variável interessante para ser avaliada no modelo.

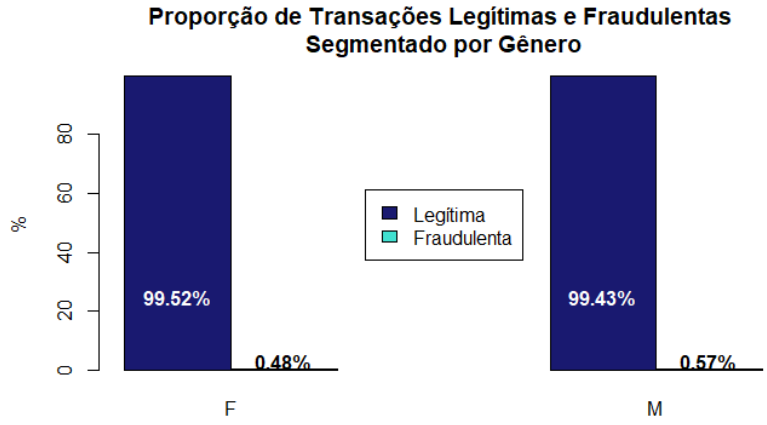
Figura 10 – Variável *amt*.



Fonte: Elaborada pelo autor (2023).

- **gender:** para a variável *gender*, que é qualitativa, temos duas categorias possíveis: masculino (M) e feminino (F). Dito isso, a Figura 11 traz o percentual de transações fraudulentas e legítimas de acordo com o gênero do titular do cartão. Podemos observar números bem próximos entre as duas categorias, sem nada que se destaque neste primeiro momento.

Figura 11 – Variável *gender*.



Fonte: Elaborada pelo autor (2023).

- **age:** a partir da data de nascimento (variável *dob*), podemos criar uma nova variável, que chamaremos de *age*, com a idade do titular do cartão de crédito. Temos na Tabela 3 o resumo numérico desta nova variável.

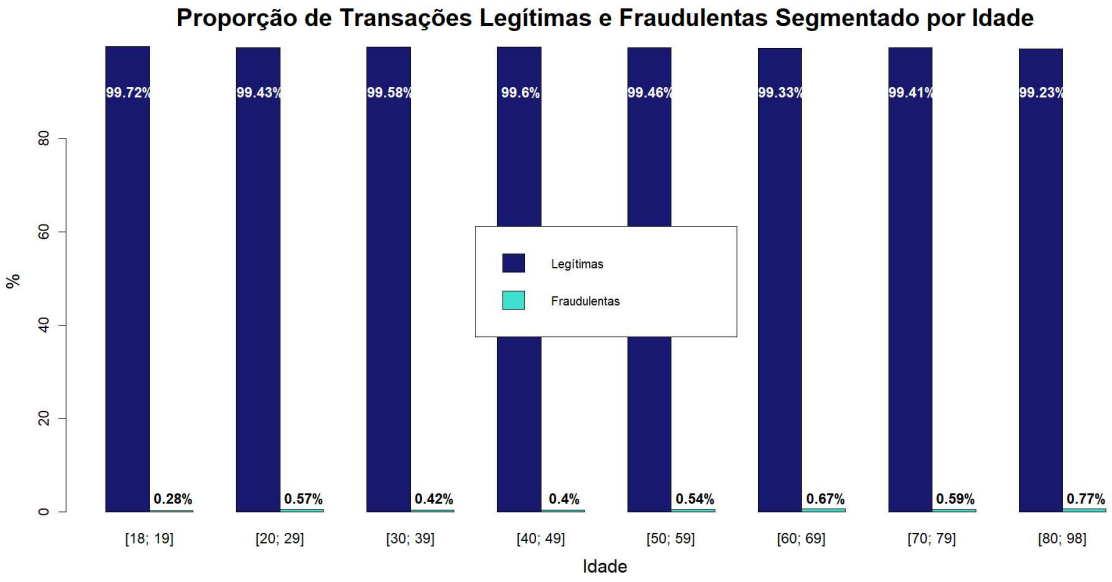
Tabela 3 – Resumo numérico da variável *age*.

	Mínimo	Mediana	Média	Máximo
age	18	47	49,43	98

Fonte: Elaborada pelo autor (2023).

A Figura 12 traz a idade do titular do cartão de crédito, dividida em intervalos, segmentando por transações legítimas e fraudulentas. Podemos observar que, embora os percentuais, no geral, estejam bem próximos, parece haver uma maior propensão à fraude, mesmo que sensível, com o aumento da idade, porém, nada conclusivo.

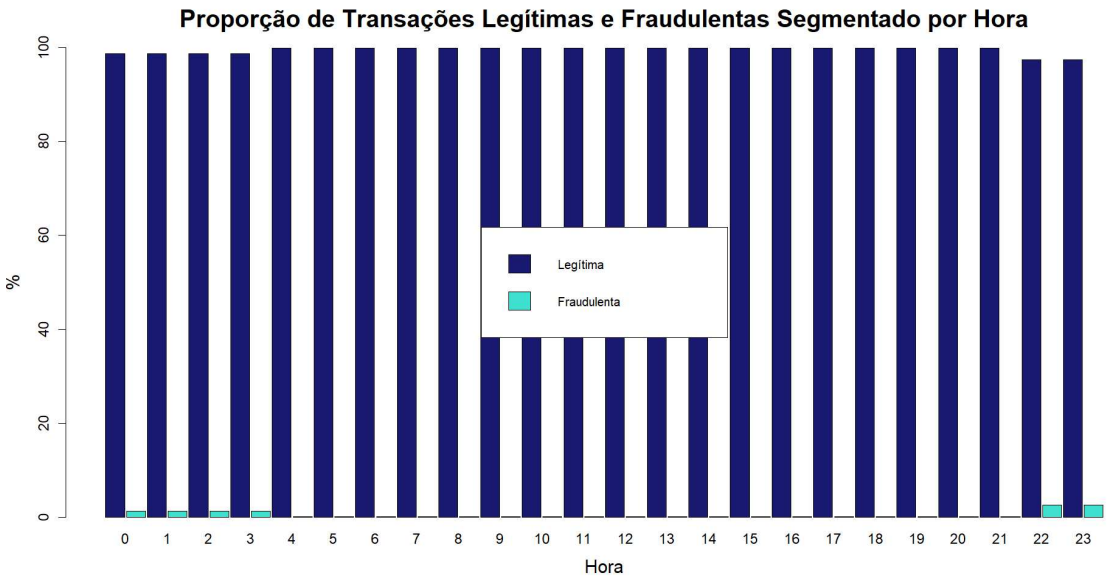
Figura 12 – Variável *age*.



Fonte: Elaborada pelo autor (2023).

- **hour:** a partir da data e hora da transação (variável *trans_date_trans_time*), podemos extrair apenas a hora para criar uma nova variável, que chamaremos de *hour*, com a hora da transação (e apenas a hora, sem minutos e/ou segundos). A Figura 13 traz a hora da transação, segmentando por transações legítimas e fraudulentas.

Figura 13 – Variável *hour*.

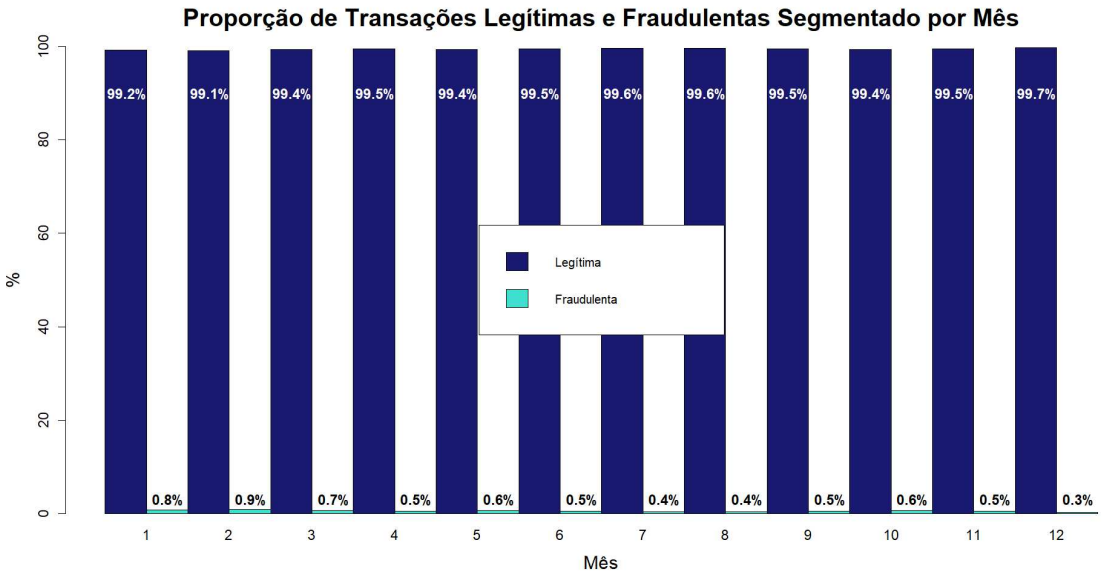


Fonte: Elaborada pelo autor (2023).

É possível notar que, no intervalo entre às 22 horas e às 3 horas, há uma maior proporção de transações fraudulentas em relação ao restante do dia.

- **trans_month:** também a partir da data e hora da transação (variável *trans_date_trans_time*), podemos extrair o mês para criar uma nova variável, que chamaremos de *trans_month*, contendo o mês da transação. A partir da Figura 14, não é possível tirar conclusões claras a respeito da variável.

Figura 14 – Variável *trans_month*.



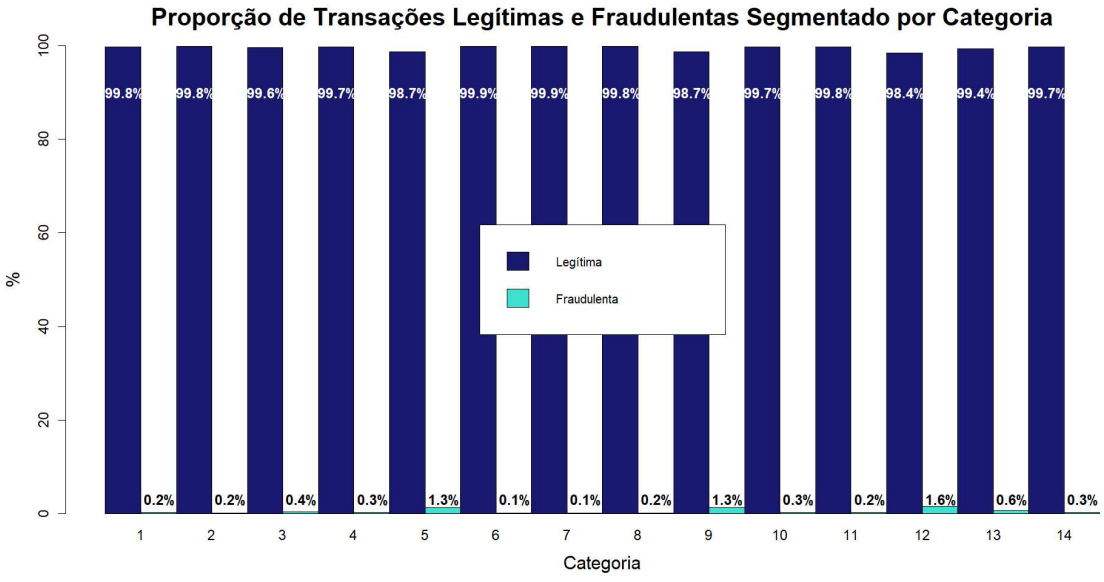
Fonte: Elaborada pelo autor (2023).

- **category:** para a variável qualitativa *category*, que contém a categoria do comerciante, temos 14 categorias possíveis, sendo elas:

  1. *entertainment*;
  2. *food_dining*;
  3. *gas_transport*;
  4. *grocery_net*;
  5. *grocery_pos*;
  6. *health_fitness*;
  7. *home*;
  8. *kids_pets*;
  9. *misc_net*;
  10. *misc_pos*;
  11. *personal_care*;
  12. *shopping_net*;
  13. *shopping_pos*;
  14. *travel*.

  A Figura 15 abaixo traz a categoria do comerciante, segmentando cada uma por transações legítimas e fraudulentas. Nela podemos destacar as categorias 5 (*grocery_pos*), 9 (*misc_net*) e 12 (*shopping_net*), que possuem percentuais de transações fraudulentas mais altos (acima de 1%) que as demais categorias.

Figura 15 – Variável *category*.



Fonte: Elaborada pelo autor (2023).

4.3 TRATAMENTO DOS DADOS

Uma etapa bastante comum antes da utilização de qualquer base de dados é a limpeza/tratamento que se faz necessário nos dados para que possamos utilizá-los na sequência do trabalho, independente do propósito (modelagem, análise de dados, etc.). Esse processo é muitas vezes chamado de *Data Wrangling*, que pode ser definido como sendo "a habilidade de pegar uma fonte de dados bagunçada e não refinada e transformá-la em algo útil." (Boehmke, 2016, p. 3, tradução nossa).[2] No caso desta base em específico, vale reforçar que, como dito no início da Subseção 4.2, não existem dados faltantes, não sendo, portanto, uma preocupação nesta etapa.

Um ponto que vale ressaltar sobre as variáveis inicialmente presentes na base de dados é que, para algumas, não faz sentido incluir no modelo. Isso porque elas se comportam como uma espécie de identificador e não seriam úteis para o propósito do modelo, que é tentar identificar transações fraudulentas. Note que, na Subseção 4.1, quando foram descritas todas as 23 variáveis da base, 11 delas (por exemplo: *cc_num*, *first*, *city*, entre outras) não foram classificadas como qualitativas ou quantitativas propositalmente por este motivo.

De forma a tentar extrair o máximo de informações significativas das variáveis originais da base de dados, foram criadas novas variáveis mais plausíveis, pensando tanto em uma possível utilização posterior no modelo, quanto na melhor compreensão das informações (interpretabilidade). São elas:

- **variável *age*:** a partir da variável *dob*, que contém a data de nascimento do titular do cartão de crédito, podemos calcular a idade do titular e assim termos uma variável mais útil, tanto para análises, como para inserir no modelo;
- **variável *trans_month*:** a partir da variável *trans_date_trans_time*, que possui as informações da data e hora da transação, podemos extrair apenas o mês da transação, criando assim uma nova variável;
- **variável *trans_year*:** também a partir da variável *trans_date_trans_time*, podemos extrair o ano da transação, dando origem a mais uma variável;
- **variável *trans_period*:** ainda a partir da variável *trans_date_trans_time*, podemos extrair a hora da transação. E, a partir da hora, podemos categorizar o horário da transação em dois períodos:
  - Diurna: transações realizadas das 6h às 19h59;
  - Noturna: transações realizadas das 20h às 5h59.

[2] No original: "It's the ability to take a messy, unrefined source of data and wrangle it into something useful."

- **variável *city_size*:** partindo da variável *city_pop*, que contém a população da cidade do titular do cartão de crédito, podemos criar uma nova variável categorizando as cidades de acordo com o tamanho da sua população, com as seguintes categorias:
  - Pequena: população abaixo de 10.000;
  - Média: população entre 10.000 e 100.000;
  - Grande: população acima de 100.000.

- **variável *trans_rec*:** com base nas variáveis *unix_time* e *cc_num*, podemos calcular a diferença, inicialmente em segundos, entre transações utilizando um mesmo cartão. Depois, convertendo a diferença de segundos para horas, podemos categorizar esta diferença e transformar em uma nova variável, com as seguintes categorias possíveis:
  - Primeira transação: primeira transação realizada pelo cartão de crédito;
  - Menos de 1h: a última utilização do cartão de crédito usado na transação atual faz menos de 1h;
  - Entre 1h e 6h: a última utilização do cartão de crédito usado na transação atual tem entre 1h e 6h;
  - Entre 6h e 12h: a última utilização do cartão de crédito usado na transação atual tem entre 6h e 12h;
  - Entre 12h e 24h: a última utilização do cartão de crédito usado na transação atual tem entre 12h e 24h;
  - Mais de 24h: a última utilização do cartão de crédito usado na transação atual faz mais de 24h.

- **variável *dist*:** por fim, derivando das variáveis *lat*, *long*, *merch_lat* e *merch_long*, podemos calcular a distância, em quilômetros, da localização do titular do cartão de crédito para a localização do comerciante. Esse cálculo envolve a utilização da fórmula de Haversine, que é uma equação utilizada para calcular a distância entre dois pontos na superfície de uma esfera com base em suas coordenadas de latitude e longitude. Mais detalhes sobre a fórmula de Haversine podem ser encontrados em Azdy e Darnis (2020).

Com isso, após a criação das 7 novas variáveis acima e com o descarte de algumas outras (seja por ter dado origem a uma outra variável que traz a informação de forma mais útil ou por ser uma espécie de identificador que não faz sentido ser utilizado), ficamos com as seguintes 11 variáveis: **category**, **amt**, **gender**, **is_fraud**, **age**, **trans_month**, **trans_year**, **trans_period**, **city_size**, **trans_rec** e **dist**.

Por fim, um último ajuste a ser feito na base de dados diz respeito a conversão de variáveis após a leitura inicial da base no *software* R e as transformações mencionadas anteriormente. Das 11 variáveis restantes, 8 delas tiveram o seu tipo de dado modificado para que o *software* as reconheça como variáveis qualitativas (ou categóricas), sendo elas: **category**, **gender**, **is_fraud**, **trans_month**, **trans_year**, **trans_period**, **city_size** e **trans_rec**.

Temos, então, após todas as transformações e ajustes necessários nos dados, as 11 variáveis classificadas e detalhadas a seguir:

- **category** - variável qualitativa com as seguintes categorias possíveis:
  - *entertainment*;
  - *food_dining*;
  - *gas_transport*;
  - *grocery_net*;
  - *grocery_pos*;
  - *health_fitness*;
  - *home*;
  - *kids_pets*;
  - *misc_net*;
  - *misc_pos*;
  - *personal_care*;
  - *shopping_net*;
  - *shopping_pos*;
  - *travel*.
- **amt** - variável quantitativa (em dólares).
  - *mínimo:* 1,00;
  - *média:* 70,06;
  - *máximo:* 28.948,90;
- **gender** - variável qualitativa com as seguintes categorias possíveis:
  - *F* (feminino);
  - *M* (masculino).

- **is_fraud** - variável qualitativa com as seguintes categorias possíveis:
  - *0* (transação legítima);
  - *1* (transação fraudulenta).
- **age** - variável quantitativa (em anos completos).
  - *mínimo:* 18;
  - *média:* 49,43;
  - *máximo:* 98.
- **trans_month** - variável qualitativa com as seguintes categorias possíveis:
  - *1* (Janeiro);
  - *2* (Fevereiro);
  - *3* (Março);
  - *4* (Abril);
  - *5* (Maio);
  - *6* (Junho);
  - *7* (Julho);
  - *8* (Agosto);
  - *9* (Setembro);
  - *10* (Outubro);
  - *11* (Novembro);
  - *12* (Dezembro).
- **trans_year** - variável qualitativa com as seguintes categorias possíveis:
  - *2019*;
  - *2020*.
- **trans_period** - variável qualitativa com as seguintes categorias possíveis:
  - *Daytime*;
  - *Nighttime*.
- **city_size** - variável qualitativa com as seguintes categorias possíveis:
  - *Small*;

  - *Avg*;
  - *Big*;

- **trans_rec** - variável qualitativa com as seguintes categorias possíveis:
  - *First transaction*;
  - *Less than 1h*;
  - *Between 1h and 6h*;
  - *Between 6h and 12h*;
  - *Between 12h and 24h*;
  - *After 24h*;

- **dist** - variável quantitativa (em quilômetros).
  - *mínimo:* 0,02;
  - *média:* 76,11;
  - *máximo:* 152,12.

4.4 RESULTADOS

Feita toda a etapa de tratamento dos dados, aplicaremos diferentes técnicas na base resultante, buscando a que traga os melhores resultados dentro do escopo do trabalho, que é identificar transações fraudulentas. Os códigos utilizados nesta seção podem ser consultados no Apêndice B.

4.4.1 Regressão Logística

Em um primeiro momento, aplicaremos a Regressão Logística na base de dados com os dados desbalanceados e, posteriormente, faremos o balanceamento dos dados e aplicaremos novamente a Regressão Logística.

4.4.1.1 Dados desbalanceados

Como informado na Seção 4.1, a base encontrada no *Kaggle* estava previamente dividida em uma base de treino e uma base de teste. Porém, por não sabermos como foi feita essa divisão prévia e para evitar qualquer tipo de viés de seleção, faremos a junção novamente da base e uma nova separação em termos definidos a seguir.

Dito isso, após o tratamento aplicado aos dados, temos uma base com 1.852.394 observações (linhas) e 11 variáveis (colunas). Faremos então a divisão da base em uma base de treino, que será utilizada para construir o modelo; e uma base de teste, que servirá

para testar o modelo construído. Separaremos, de maneira aleatória, 80% da base para treinamento, deixando os 20% restantes para teste. A Tabela 4 resume as bases após o processo de separação da base completa (processo conhecido como *data splitting*).

Tabela 4 – Resumo das bases de dados.

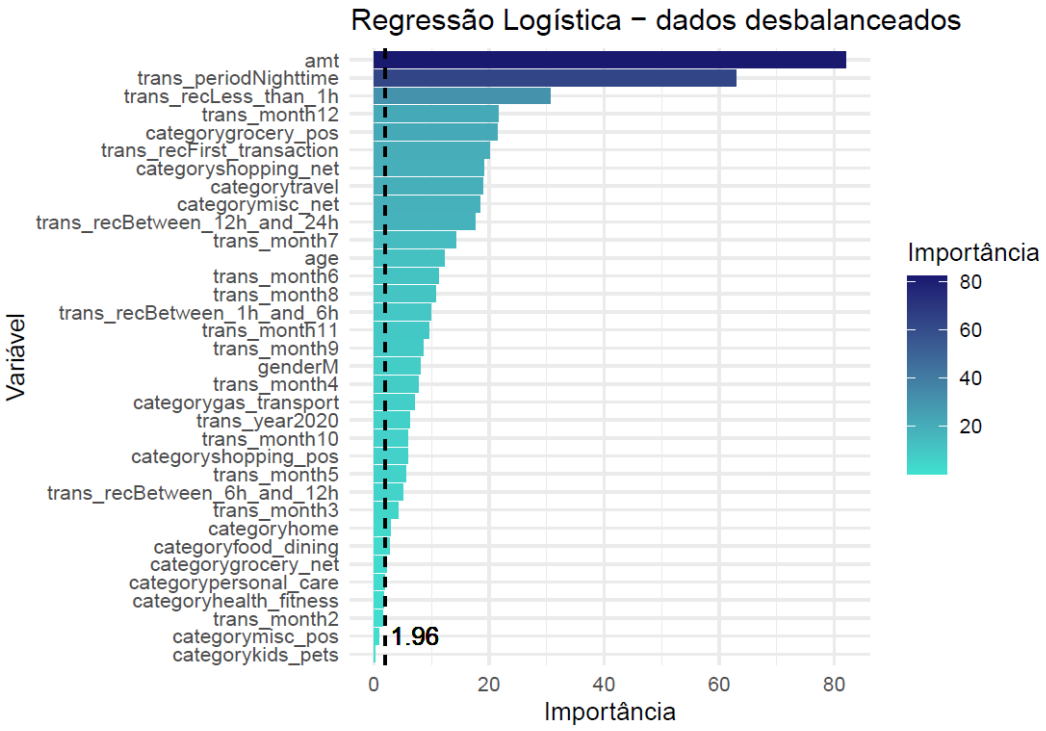
	Observações	Legítimas	Fraudulentas
Base completa	1.852.394	1.842.743	9.651
Base de treino	1.481.916	1.474.195	7.721
Base de teste	370.478	368.548	1.930

Fonte: Elaborada pelo autor (2024).

A construção do modelo se dará utilizando a técnica chamada *k-fold cross-validation*, com k = 5, conforme discutido no Capítulo 3, de forma a construir e validar o modelo ao mesmo tempo, restando apenas a parte de teste que se dará posteriormente. Também conforme discutido no Capítulo 3, por conta do desbalanceamento extremo dos dados, a métrica que buscaremos maximizar durante a construção/validação do modelo será a Acurácia Balanceada.

Inicialmente, ajustou-se o modelo utilizando a base de treino e suas 10 variáveis preditoras. Verificou-se, porém, que as variáveis *dist* e *city_size* não foram estatisticamente significativas considerando o nível de significância de 5%. Ajustou-se então novamente o modelo sem estas duas variáveis e a Figura 16 abaixo ilustra a importância das variáveis no modelo.

Figura 16 – Importância das variáveis no modelo de Regressão Logística com dados desbalanceados.



Fonte: Elaborada pelo autor (2024).

Podemos observar que apenas algumas categorias de certas variáveis ficam abaixo do corte de significância estabelecido e, portanto, as variáveis serão mantidas no modelo já que possuem outras categorias que são significativas. Sendo assim, temos o modelo final para o caso de Regressão Logística com dados desbalanceados.

O passo seguinte é testar o comportamento do modelo com dados desconhecidos por ele (base de teste), e verificarmos seu desempenho através da Matriz de Confusão, de algumas métricas (principalmente a Acurácia Balanceada) e da curva ROC.

Tabela 5 – Matriz de confusão - Regressão Logística com dados desbalanceados.

		Real	
		Legítima	Fraudulenta
Predito	Legítima	368.412	1.874
	Fraudulenta	136	56

Fonte: Elaborada pelo autor (2024).

Como pode ser visto pela matriz de confusão na Tabela 5, fica evidente que a grande maioria das transações fraudulentas foram classificadas erroneamente como legítimas pelo modelo.

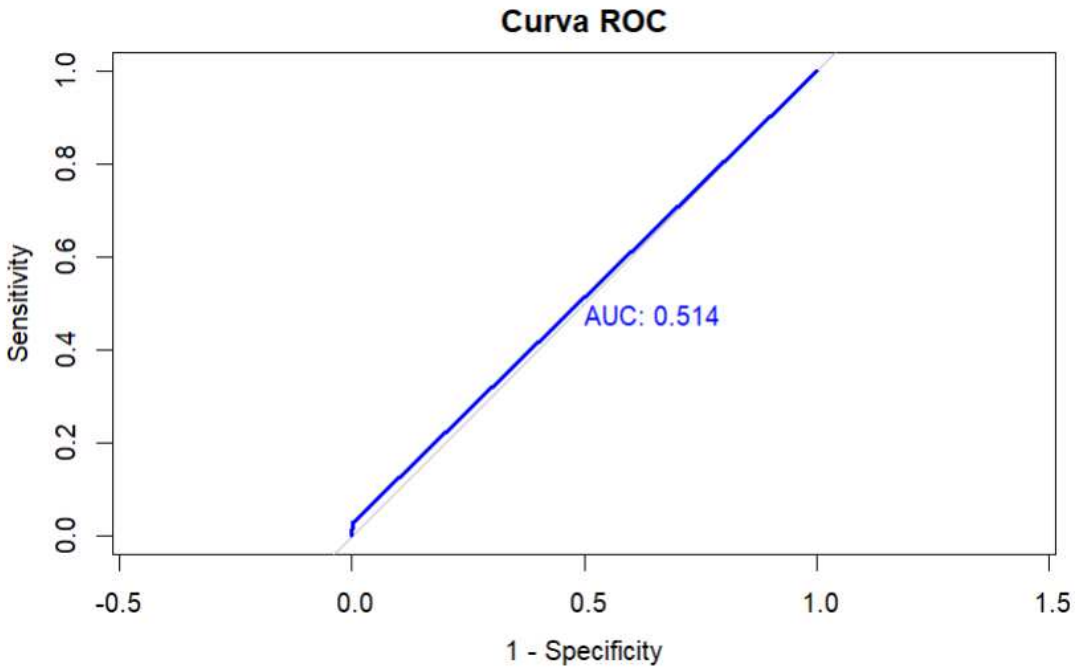
Tabela 6 – Métricas do modelo - Regressão Logística com dados desbalanceados.

Métrica	Valor
Acurácia balanceada	0,5143
Acurácia	0,9946
Sensibilidade	0,0290
Especificidade	0,9996
Pred. de valor pos.	0,2917
Pred. de valor neg.	0,9949

Fonte: Elaborada pelo autor (2024).

Quando observamos as métricas da Tabela 6, notamos que, como discutido no Capítulo 3, a acurácia não serve como parâmetro de desempenho do modelo, neste contexto, por conta do extremo desbalanceamento dos dados. Embora tenhamos uma acurácia excepcional de 99,46%, o modelo não tem utilidade prática e não consegue separar minimamente bem as transações legítimas das fraudulentas, como a matriz de confusão da Tabela 5 deixou evidente juntamente com o baixo valor de sensibilidade e pela acurácia balanceada próxima de 0,5.

Figura 17 – Curva ROC - Regressão Logística com dados desbalanceados.



Fonte: Elaborada pelo autor (2024).

Podemos notar que a área abaixo da curva (AUC) é muito próxima de 0,5 (Figura 17), sendo mais um indício de que o modelo de Regressão Logística não apresenta um

resultado satisfatório dentro do contexto do problema analisado e na presença de uma base de dados extremamente desbalanceada.

4.4.1.2 Dados balanceados

Com o objetivo de verificar o comportamento do modelo de Regressão Logística treinado com uma base de dados balanceada, foi feito o balanceamento da base de treino, de forma a termos a mesma proporção de transações legítimas e fraudulentas.

Conforme detalhado no Capítulo 3, existem algumas formas de se balancear uma base de dados e a técnica escolhida para fazer o balanceamento foi o *undersampling*, que resumidamente consiste em manter todas as observações da classe minoritária da base (nesse caso, as fraudes) e retirar uma amostra aleatória da classe majoritária (transações legítimas) de mesmo tamanho da minoritária, ficando então com uma base de dados com mesma proporção entre as classes (como pode ser visto na Tabela 7).

Vale ressaltar ainda que, embora o modelo será treinado com a nova base de treino após o processo de balanceamento dos dados, o teste será feito com a mesma base de teste original, ou seja, na presença de um extremo desbalanceamento de dados.

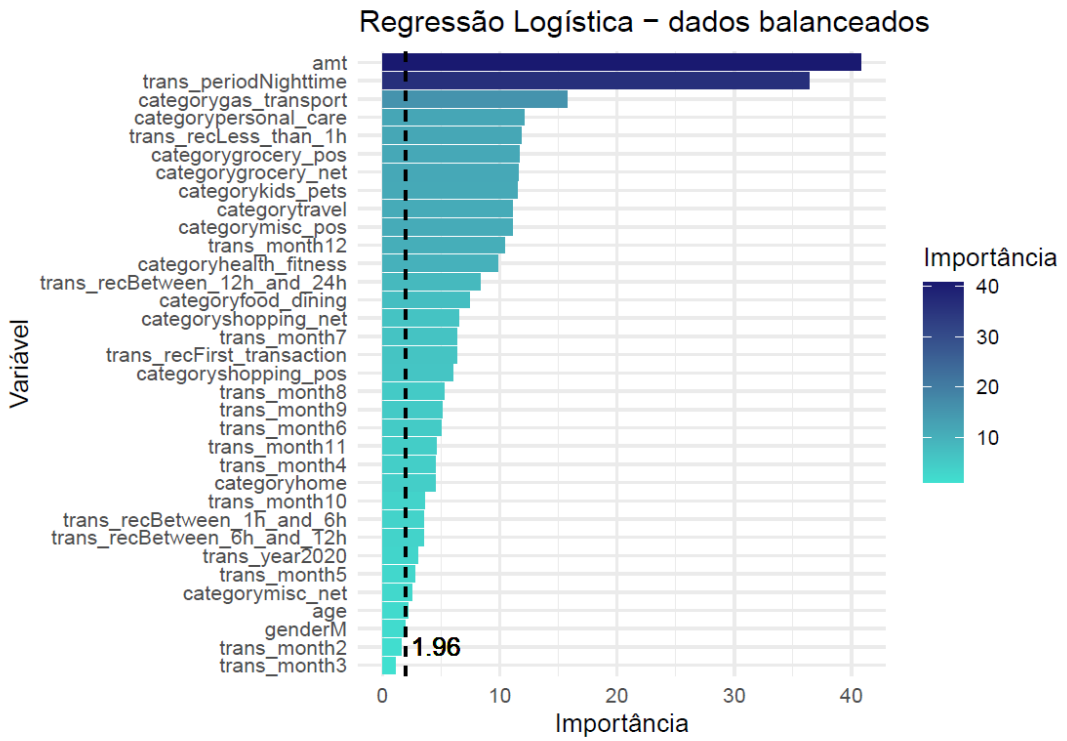
Tabela 7 – Resumo da base de treino após o balanceamento dos dados.

	Observações	Legítimas	Fraudulentas
Base de treino balanceada	15.442	7.721	7.721

Fonte: Elaborada pelo autor (2024).

Após o balanceamento dos dados, foram repetidos todos os passos anteriores para construção/validação e teste do modelo. Novamente o modelo foi construído utilizando a técnica de *k-fold cross-validation*, com k = 5. Mais uma vez, as variáveis *dist* e *city_size* não foram estatisticamente significativas considerando o nível de significância de 5% e foram retiradas do modelo. A seleção do modelo se deu buscando maximizar a Acurácia Balanceada. A Figura 18, a seguir, ilustra os resultados em relação a importância das variáveis no modelo.

Figura 18 – Importância das variáveis no modelo de Regressão Logística com dados balanceados.



Fonte: Elaborada pelo autor (2024).

Observando a Figura 18, é válido fazer uma ressalva. A variável *gender* está na região de "fronteira" do nível de significância estabelecido (possui valor absoluto de $z = 1{,}95$) e será mantida no modelo. Feita esta exceção, existem apenas algumas categorias da variável *trans_month* abaixo do corte estabelecido, porém, como outras categorias desta mesma variável estão acima do corte, a variável será mantida e temos então o modelo final obtido.

Seguindo na análise, utilizaremos a base de teste (que vale lembrar, é desbalanceada) para verificar o comportamento do modelo e posteriormente medirmos o seu desempenho com as mesmas métricas utilizadas anteriormente.

Tabela 8 – Matriz de confusão - Regressão Logística com dados balanceados.

		Real	
		Legítima	Fraudulenta
Predito	Legítima	322.737	274
	Fraudulenta	45.811	1.656

Fonte: Elaborada pelo autor (2024).

Na matriz de confusão, dada na Tabela 8, temos algumas mudanças consideráveis em relação a obtida com o modelo treinado com os dados desbalanceados. A proporção de transações fraudulentas corretamente identificada pelo modelo é bem maior, ao passo que os falso positivos (ou seja, transações preditas pelo modelo como fraudulentas, mas que na realidade são legítimas) também cresceram bastante.

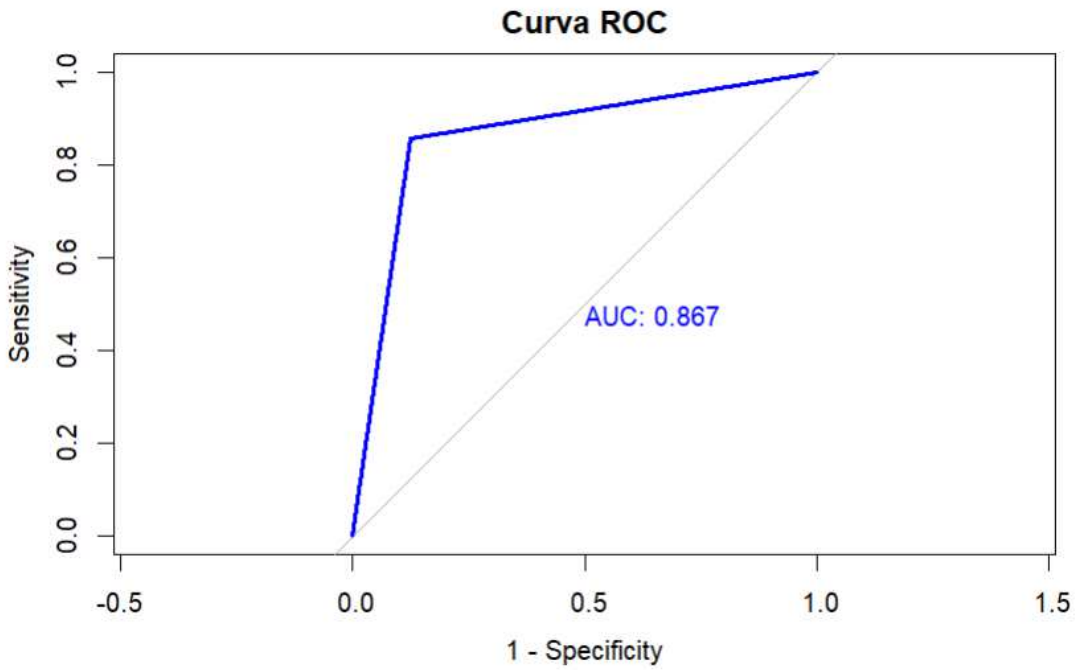
Tabela 9 – Métricas do modelo - Regressão Logística com dados balanceados.

Métrica	Valor
Acurácia balanceada	0,8669
Acurácia	0,8756
Sensibilidade	0,8580
Especificidade	0,8757
Pred. de valor pos.	0,0349
Pred. de valor neg.	0,9992

Fonte: Elaborada pelo autor (2024).

Já na Tabela 9, podemos notar que temos resultados mais equilibrados em relação ao modelo de Regressão Logística com dados desbalanceados. Se por um lado o modelo perdeu acurácia, podemos notar uma discrepância menor entre sensibilidade e especificidade, o que automaticamente causa um aumento bem relevante na acurácia balanceada.

Figura 19 – Curva ROC - Regressão Logística com dados balanceados.



Fonte: Elaborada pelo autor (2024).

Observando a Figura 19, temos uma área abaixo do curva (AUC) consideravelmente maior com o modelo treinado após o balanceamento dos dados.

4.4.2 Floresta Aleatória

Nesta etapa, após a construção dos modelos de Regressão Logística nos dois casos (dados desbalanceados e balanceados), construiremos, de forma similar, modelos de Floresta Aleatória (*Random Forest*).

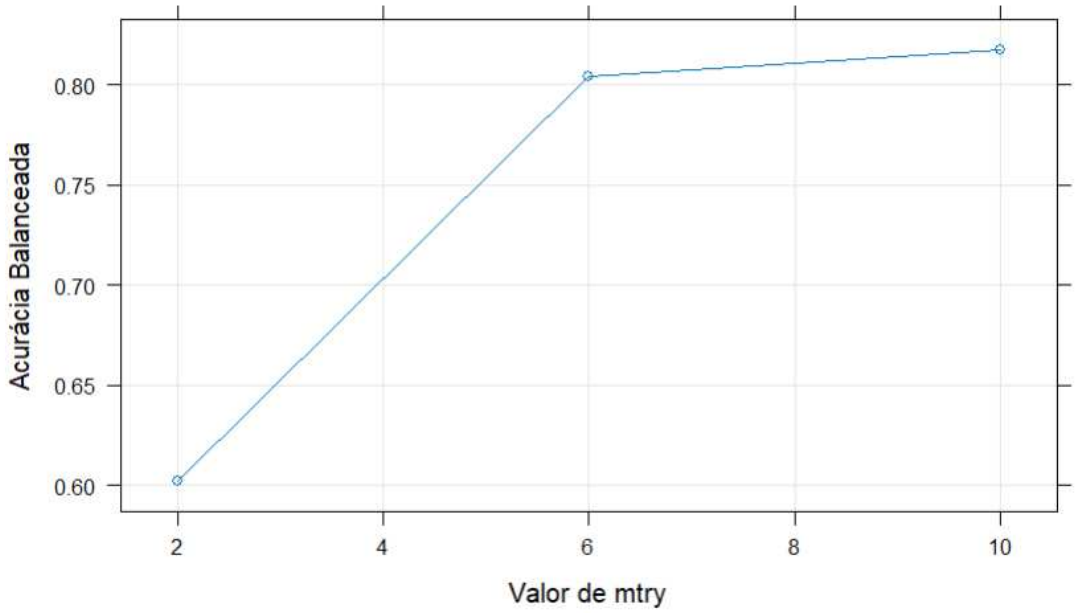
4.4.2.1 Dados desbalanceados

Inicialmente, usando a base de treino original (ou seja, antes do balanceamento), ajustaremos um modelo de Floresta Aleatória. Para o ajuste do modelo, usaremos uma vez mais a *k-fold cross-validation*, com k = 5, como forma de construir e validar o modelo.

Dito isso, temos as definições para a construção do modelo. O número de árvores de decisão a ser construído para se formar a Floresta Aleatória permaneceu o valor *default* do R, que são 500 árvores. O tamanho da amostra da base de treino que será usado para construir cada árvore foi definido como 10.000, que é um tamanho robusto e exequível pensando no custo computacional. Portanto, para a construção da Floresta Aleatória, serão formadas 500 árvores de decisão, sendo cada uma desenvolvida ao retirar-se uma amostra aleatória com reposição de tamanho 10.000 da base de treino.

Já o hiperparâmetro m_{try}, que define a quantidade de variáveis preditoras que será usada na construção do nó de cada árvore de decisão, foi otimizado utilizando a validação cruzada, buscando maximizar a Acurácia Balanceada. Os resultados na Figura 20 demonstram que a métrica é maximizada com $m_{try} = 10$. Como visto no Capítulo 3, em um cenário em que m_{try} é igual ao total de variáveis preditoras disponíveis, como neste caso, o modelo é reduzido ao *Bagging*.

Figura 20 – Valor ótimo do hiperparâmetro m_{try} - Floresta Aleatória com dados desbalanceados.



Fonte: Elaborada pelo autor (2024).

Definido o modelo final, iremos testá-lo utilizando a base de teste e verificar seu desempenho por meio das mesmas métricas empregadas no caso da Regressão Logística.

Tabela 10 – Matriz de confusão - Floresta Aleatória com dados desbalanceados.

		Real	
		Legítima	Fraudulenta
Predito	Legítima	368.322	696
	Fraudulenta	226	1.234

Fonte: Elaborada pelo autor (2024).

Pela matriz de confusão (Tabela 10), é possível observar uma visível melhora em relação ao resultado obtido com a Regressão Logística no caso desbalanceado, já que temos mais transações fraudulentas sendo identificadas corretamente pelo modelo.

Na Tabela 11, temos algumas métricas de desempenho. Notamos uma excelente acurácia de 99,75% e uma boa acurácia balanceada de 81,94%. Enquanto a especificidade beira a perfeição (99,94%), a sensibilidade de 63,94% pode ser melhorada, porém, quando comparada a sensibilidade de 2,90% obtida com o modelo de Regressão Logística treinado com dados desbalanceados, temos um grande salto nesta métrica no modelo de Floresta Aleatória.

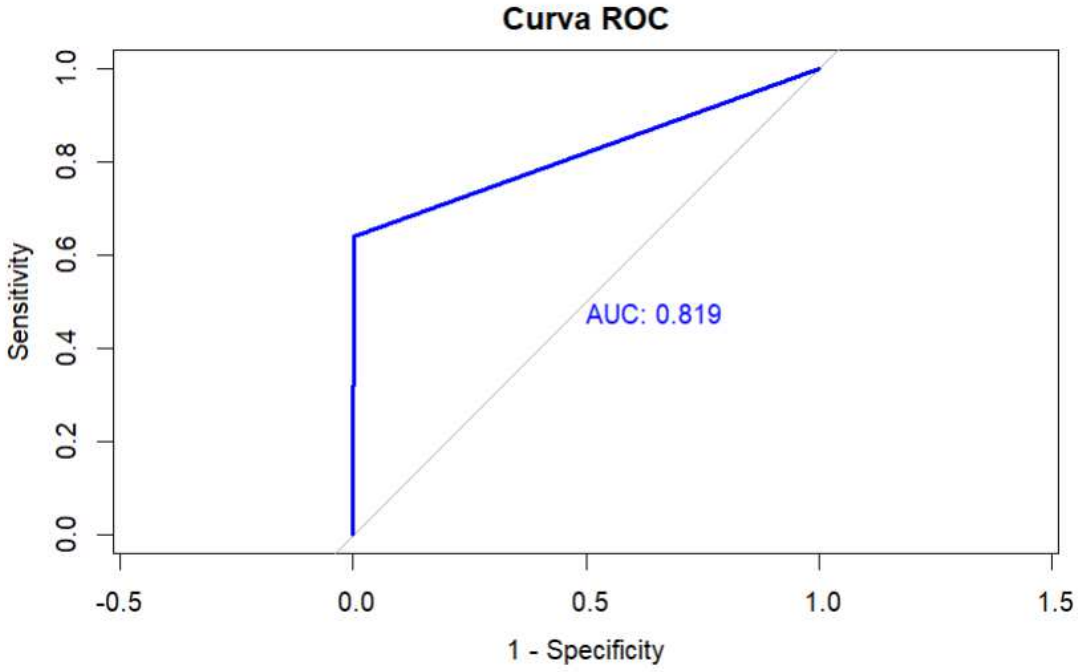
Tabela 11 – Métricas do modelo - Floresta Aleatória com dados desbalanceados.

Métrica	Valor
Acurácia balanceada	0,8194
Acurácia	0,9975
Sensibilidade	0,6394
Especificidade	0,9994
Pred. de valor pos.	0,8452
Pred. de valor neg.	0,9981

Fonte: Elaborada pelo autor (2024).

Na curva ROC (Figura 21), também temos uma melhora considerável na área abaixo da curva (AUC) do modelo de Floresta Aleatória quando comparado com o de Regressão Logística, ambos construídos com dados desbalanceados.

Figura 21 – Curva ROC - Floresta Aleatória com dados desbalanceados.



Fonte: Elaborada pelo autor (2024).

4.4.2.2 Dados balanceados

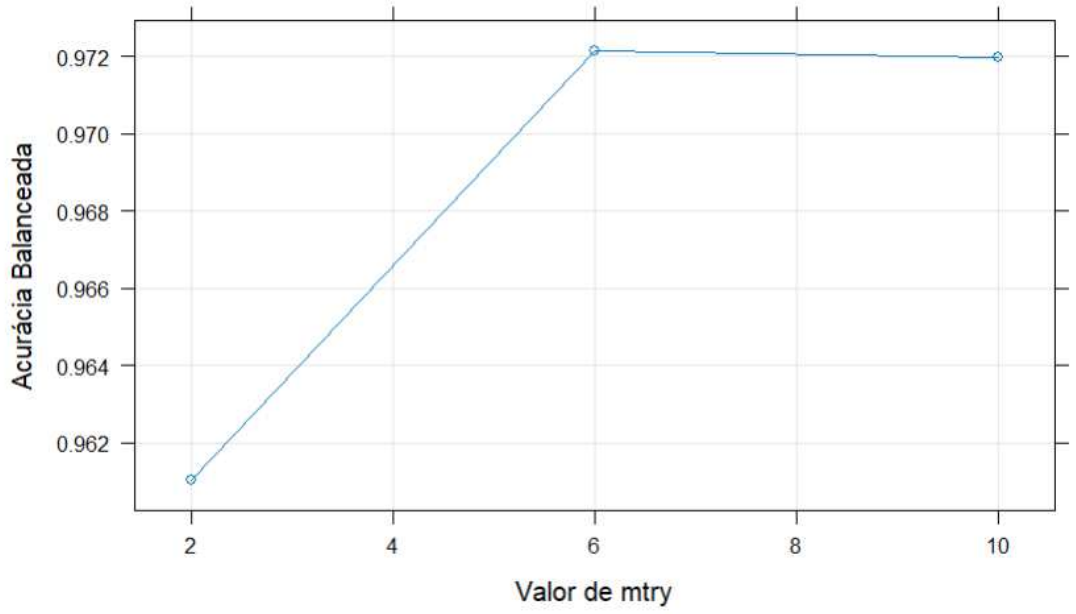
Por fim, repetiremos os mesmos passos adotados anteriormente para o modelo de Floresta Aleatória, com a diferença de que o modelo será treinado com a base de treino após o balanceamento dos dados.

Assim como no caso desbalanceado, o modelo será construído/validado utilizando o método de *k-fold cross-validation*, com k = 5. Manteremos também as mesmas definições

do modelo utilizadas anteriormente, com 500 árvores de decisão formando a Floresta Aleatória, sendo cada uma formada ao retirar-se uma amostra aleatória com reposição de tamanho 10.000 da base de treino após o balanceamento dos dados.

Novamente foi utilizada a validação cruzada para otimizar o hiperparâmetro m_{try}, buscando maximizar a Acurácia Balanceada. Os resultados estão na Figura 22 e pode-se notar que a Acurácia Balanceada é maximizada com $m_{try} = 6$.

Figura 22 – Valor ótimo do hiperparâmetro m_{try} - Floresta Aleatória com dados balanceados.



Fonte: Elaborada pelo autor (2024).

Iremos então avaliar o desempenho do modelo utilizando a base de teste (lembrando, mais uma vez, que a base de teste não foi balanceada), a partir das mesmas métricas aplicadas anteriormente.

Tabela 12 – Matriz de confusão - Floresta Aleatória com dados balanceados.

		Real	
		Legítima	Fraudulenta
Predito	Legítima	354.938	47
	Fraudulenta	13.610	1.883

Fonte: Elaborada pelo autor (2024).

Podemos observar pela matriz de confusão da Tabela 12 que o modelo parece detectar melhor as transações fraudulentas, já que conseguiu identificar 1.883 das 1.930

fraudes presentes na base de teste (cerca de 97%).

Pelas métricas presentes na Tabela 13, observamos que o modelo desempenha muito bem no propósito de detectar fraudes. O ponto de atenção, porém, está na predição de valores positivos (que, neste caso, são as fraudes) de aproximadamente 12%, o que indica que o modelo tem um alto percentual de falso positivos (ou seja, classifica como fraude transações que na realidade são legítimas).

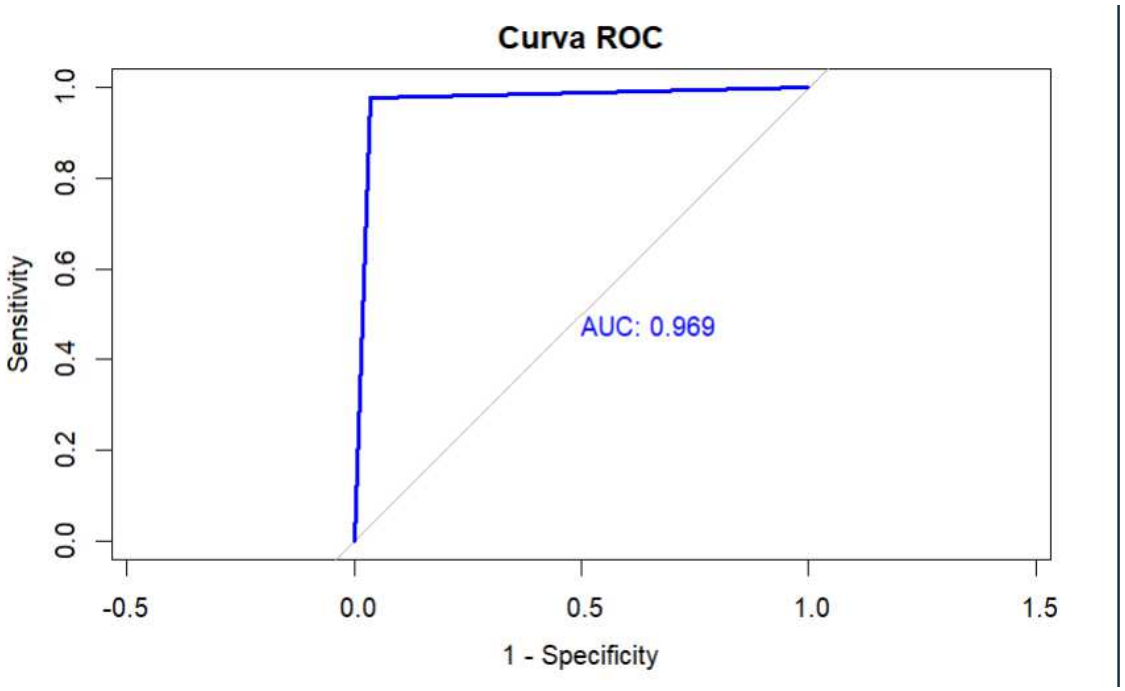
Tabela 13 – Métricas do modelo - Floresta Aleatória com dados desbalanceados.

Métrica	Valor
Acurácia balanceada	0,9694
Acurácia	0,9631
Sensibilidade	0,9756
Especificidade	0,9631
Pred. de valor pos.	0,1215
Pred. de valor neg.	0,9999

Fonte: Elaborada pelo autor (2024).

E, finalmente, notamos na Figura 23 que a área abaixo da curva (AUC) é bem próxima de 1, que seria o valor ideal, sendo o modelo de melhor desempenho nesta métrica dentre os avaliados.

Figura 23 – Curva ROC - Floresta Aleatória com dados desbalanceados.



Fonte: Elaborada pelo autor (2024).

5 CONCLUSÃO

Após os resultados obtidos no Capítulo 4, podemos chegar a algumas conclusões. A Tabela 14 traz as métricas dos quatro casos analisados na Seção 4.4 (modelos de Regressão Logística - RL e Floresta Aleatória - FA, ambos com dados balanceados e desbalanceados), com o intuito de facilitar a comparação.

Tabela 14 – Comparação das métricas obtidas com cada modelo.

Métrica	RL desbalanc.	RL balanceado	FA desbalanc.	FA balanceado
Acurácia balanceada	0,5143	0,8669	0,8194	0,9694
Acurácia	0,9946	0,8756	0,9975	0,9631
Sensibilidade	0,0290	0,8580	0,6394	0,9756
Especificidade	0,9996	0,8757	0,9994	0,9631
Pred. de valor pos.	0,2917	0,0349	0,8452	0,1215
Pred. de valor neg.	0,9949	0,9992	0,9981	0,9999

Fonte: Elaborada pelo autor (2024).

Observando a Tabela 14, comparando os modelos treinados em situações similares (ou seja, RL desbalanceado *vs* FA desbalanceado e RL balanceado *vs* FA balanceado), percebemos que os modelos de Floresta Aleatória apresentam um melhor desempenho, medido pela maior acurácia balanceada, do que os modelos de Regressão Logística nas duas situações. Portanto, dentro do contexto de detecção de transações fraudulentas no e-commerce, fica evidente a partir dos resultados obtidos que a utilização do modelo de Floresta Aleatória é preferível ao modelo de Regressão Logística.

Ainda observando a Tabela 14, é importante ressaltar que a técnica de balancear os dados para treinar o modelo (mesmo que o teste seja feito com dados desbalanceados) traz ganhos de desempenho em métricas importantes para o problema apresentado durante este trabalho, que é a detecção de fraudes em transações online. Ao fazermos o balanceamento para treinar o modelo, diminuímos consideravelmente a diferença entre sensibilidade e especificidade, o que por consequência melhora a acurácia balanceada, que é a métrica que buscamos maximizar (mesmo que haja uma pequena perda na acurácia total do modelo).

Além disso, podemos definir o modelo de Floresta Aleatória treinado com dados balanceados (FA balanceado) como o melhor modelo dentre as 4 opções. Isso porque é o que apresenta a maior acurácia balanceada, com ótimas métricas de especificidade e sensibilidade, além de uma acurácia bastante alta. Seu ponto fraco acaba sendo a predição de valores positivos. Na prática e no contexto de detecção de fraudes, isto significa que o modelo consegue identificar a grande maioria das transações fraudulentas (ótima sensibilidade), porém, acusa muito falso positivo (que é quando o modelo classifica como fraude uma transação que na verdade é legítima), algo evidente pelo baixo percentual

de predição de valores positivos (cerca de 12%). Uma caminho normalmente utilizado no mercado para contornar esta limitação do modelo é combinar a saída do modelo (normalmente em forma de um *score* de fraude) com regras de negócio, como forma de tomar uma decisão mais assertiva e conseguir separar ainda melhor transações fraudulentas das legítimas do que o que é possível apenas utilizando o modelo.

Por fim, conclui-se que o presente trabalho atingiu o objetivo proposto de comparar dois dos modelos mais comumente utilizados com a finalidade de detectar fraudes em transações online utilizando cartão de crédito. Além disso, foram apresentadas alternativas para lidar com o extremo desbalanceamento dos dados que se mostraram eficazes e potencializaram o desempenho dos modelos.

Como sugestão de trabalhos futuros, recomenda-se que sejam comparados outros modelos para detecção de fraude, como Redes Neurais, K-Vizinhos mais Próximos e *Extreme Gradient Boosting (XGBoost)*. Além disso, seria valioso investigar a performance dos modelos em outras bases de dados de fraude.

REFERÊNCIAS

AGRESTI, A. **Categorical Data Analysis**. 3. ed. Hoboken: John Wiley & Sons, 2013.

AIML. Explain the concept and working of the Random Forest model. **AIML.com**, 2023. Disponível em: https://aiml.com/what-is-random-forest-2/. Acesso em: 30 jun. 2024.

AKOBENG, A. K. Understanding diagnostic tests 3: receiver operating characteristic curves. **Acta Paediatrica**, v. 96, n. 5, p. 644-647, 2007. DOI: https://doi.org/10.1111/j.1651-2227.2006.00178.x.

ASSIS, R. L. **Detecção de fraudes em cartões de crédito utilizando métodos de baseados em árvores de decisão**. 2023. Trabalho de Conclusão de Curso (Graduação em Engenharia da Computação) - Universidade Tecnológica Federal do Paraná, Pato Branco, 2023.

AZANK, F. Paradoxo da Acurácia. **Medium**, 2020. Disponível em: https://medium.com/turing-talks/paradoxo-da-acur%C3%A1cia-56baa75334f2. Acesso em: 25 maio 2024.

AZDY, R. A.; DARNIS, F. Use of Haversine Formula in Finding Distance Between Temporary Shelter and Waste End Processing Sites. **Journal of Physics: Conference Series**, v. 1500, 012104, 2020. DOI: https://doi.org/10.1088/1742-6596/1500/1/012104.

AZEVEDO, F. L. de. **Detecção de fraudes de cartão de crédito em uma base brasileira utilizando autoencoder**. 2021. Dissertação (Mestrado em Computação Aplicada) - Programa de Pós-Graduação em Computação Aplicada, Instituto Federal do Espírito Santo, Serra, 2021.

AZEVEDO, V. M.; FIGUEIRA, L. B. Detecção de fraude financeira utilizando Ciência de Dados. *In*: WORKSHOP DE TECNOLOGIA DA FATEC RIBEIRÃO PRETO, 1., 2020, Ribeirão Preto. **Anais** [...]. Ribeirão Preto: Fatec, 2020. Disponível em: http://www.fatecrp.edu.br/WorkTec/edicoes/2020-1/trabalhos/I-Worktec-Victor_Azevedo.pdf. Acesso em: 20 ago. 2023.

BELTRAN, R. D. **Detecção de fraudes bancárias utilizando métodos de clustering**. 2019. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Universidade Federal do Pampa, Alegrete, 2019.

BERALDI, F. **Atualização dinâmica de modelo de Regressão Logística binária para detecção de fraudes em transações eletrônicas com cartão de crédito**. 2014. Dissertação (Mestrado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2014.

BOEHMKE, B. C. **Data Wrangling with R**. Cham, Switzerland: Springer, 2016.

BRAMER, M. **Principles of Data Mining**. 3. ed. London: Springer, 2016.

BREIMAN, L. Random Forests. **Machine Learning**, v. 45, p. 5-32, 2001. DOI: https://doi.org/10.1023/A:1010933404324.

CASELLA, G.; BERGER, R. L. **Statistical Inference**. 2. ed. Pacific Grove: Duxbury, 2002.

CLEARSALE. Chargeback: entenda o que é e evite colocar seu negócio em risco. **Blog ClearSale**, 2021. Disponível em: https://blogbr.clear.sale/chargeback-saiba-o-que-quais-os-riscos-e-como-evit-lo. Acesso em: 24 set. 2023.

CLEARSALE. Meios de Pagamentos Digitais e seus benefícios no aumento das vendas. **Blog ClearSale**, 2023. Disponível em: https://blogbr.clear.sale/meios-de-pagamentos-digitais-e-seus-beneficios-no-aumento-das-vendas. Acesso em: 24 set. 2023.

CUTLER, A., CUTLER, D. R., STEVENS, J. R. Random Forests. *In*: ZHANG, C., MA, Y. (eds) **Ensemble Machine Learning**. New York: Springer, 2012. p. 157-175. DOI: https://doi.org/10.1007/978-1-4419-9326-7_5.

DITTMAN, D. J. *et al.* Comparison of Data Sampling Approaches for Imbalanced Bioinformatics Data *In*: PROCEEDINGS OF THE TWENTY-SEVENTH INTERNATIONAL FLORIDA ARTIFICIAL INTELLIGENCE RESEARCH SOCIETY CONFERENCE, 27., 2014, Florida. **Anais** [...]. Florida: Association for the Advancement of Artificial Intelligence, 2014. p. 268-271. Disponível em: https://cdn.aaai.org/ocs/7850/7850-36780-1-PB.pdf. Acesso em: 04 maio 2024.

EBNER, J. Cross Validation, Explained. **R-craft**, 2023. Disponível em: https://r-craft.org/cross-validation-explained/. Acesso em: 17 maio 2024.

EQUALS. Chargeback: o que fazer para lidar com a contestação de compra?. **Blog Equals**, 2021. Disponível em: https://equals.com.br/blog/chargeback-contestacao-de-compra/. Acesso em: 10 out. 2023.

FRAUDE. *In*: **DICIO, Dicionário Online de Português**. Porto: 7Graus, 2023. Disponível em: https://www.dicio.com.br/fraude/. Acesso em: 20 ago. 2023.

FREIRE, G. F. B. **Comparação de técnicas para aumentar a eficácia e eficiência de análise de sentimento por meio da redução do número de *features***. 2019. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) - Instituto de Computação, Universidade Federal de Alagoas, Maceió, 2019.

GADI, M. F. A. **Uma comparação de métodos de classificação aplicados à detecção de fraude em cartões de crédito**. 2008. Dissertação (Mestrado em Ciência da Computação) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2008.

HASANIN, T. *et al.* Severely imbalanced Big Data challenges: investigating data sampling approaches. **Journal of Big Data**, v. 6, n. 107, p. 1-25, 2019. DOI: https://doi.org/10.1186/s40537-019-0274-4.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning**: Data Mining, Inference, and Prediction. 2. ed. New York: Springer, 2009.

HOSMER, D. W.; LEMESHOW, S.; STURDIVANT, R. X. **Applied Logistic Regression**. 3. ed. Hoboken: John Wiley & Sons, 2013.

IZBICKI, R.; SANTOS, T. M. dos. **Aprendizado de máquina**: uma abordagem estatística. São Carlos: [s.n.], 2020. ISBN: 978-65-00-02410-4.

JAMES, G. *et al.* **An Introduction to Statistical Learning:** with Applications in R. New York: Springer, 2013.

JAYAWARDENA, N. How to Deal with Imbalanced Data. **Medium**, 2020. Disponível em: https://towardsdatascience.com/how-to-deal-with-imbalanced-data-34ab7db9b100. Acesso em: 04 maio 2024.

KUHN, M. Building predictive models in R using the caret package. **Journal of Statistical Software**, v. 28, n. 5, p. 1-26, 2008. DOI: https://doi.org/10.18637/jss.v028.i05.

KUHN, M.; JOHNSON, K. **Applied Predictive Modeling**. New York: Springer, 2013.

LIAW, A.; WIENER, M. Classification and Regression by randomForest. **R News**, v. 2, n. 3, p. 18-22, 2002. Disponível em: https://cran.r-project.org/doc/Rnews/Rnews_2002-3.pdf.

MARIANO, D. Métricas de avaliação em machine learning. **Diego Mariano**, 2021. Disponível em: https://diegomariano.com/metricas-de-avaliacao-em-machine-learning/. Acesso em: 26 maio 2024.

NASCIMENTO, A. R. do; SILVA, B. F. da; SANTOS, G. G. dos. **E-commerce:** o melhor caminho no mercado atual. 2009. Trabalho de Conclusão de Curso (Graduação em Administração - Marketing) - Centro Universitário Eurípides de Marília, Marília, 2009.

PACHECO JUNIOR, J. C. **Modelos para detecção de fraudes utilizando técnicas de aprendizado de máquina**. 2019. Dissertação (Mestrado em Economia) - Escola de Economia de São Paulo, Fundação Getulio Vargas, São Paulo, 2019.

PAGAR.ME. Bandeiras de cartão: entenda seu papel nas transações financeiras. **Blog Pagar.me**, 2021. Disponível em: https://pagar.me/blog/bandeira-de-cartao/. Acesso em: 08 out. 2023.

PEIXOTO, A. C. Métricas de avaliação para modelos de classificação. **Medium**, 2023. Disponível em: https://medium.com/@andreycp17/m%C3%A9tricas-de-avalia%C3%A7%C3%A3o-para-modelos-de-classifica%C3%A7%C3%A3o-1cd0f193d008. Acesso em: 26 maio 2024.

POSIT TEAM. **RStudio**: Integrated Development Environment for R. Posit Software, PBC, Boston, MA. 2023. Disponível em: http://www.posit.co/.

R CORE TEAM. **R**: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. 2023. Disponível em: https://www.R-project.org/.

ROBIN, X. *et al.* pROC: an open-source package for R and S+ to analyze and compare ROC curves. **BMC Bioinformatics**, v. 12, p. 77, 2011. DOI: https://doi.org/10.1186/1471-2105-12-77.

SANTANDER. Digital payment methods: What are they and which ones are most common?. **Banco Santander S.A.**, 2022. Disponível em: https://www.santander.com/en/stories/digital-payment-methods-what-are-they-and-which-ones-are-most-common. Acesso em: 24 set. 2023.

SHENOY, K.; HARRIS, B. Credit Card Transactions Fraud Detection Dataset. **Kaggle**, 2020. Disponível em: https://www.kaggle.com/datasets/kartik2112/fraud-detection. Acesso em: 10 ago. 2023.

SILVA, V. de O. **Detecção de fraudes na utilização de cartões usando a técnica de Regressão Logística:** uma aplicação com dados desbalanceados. 2022. Trabalho de Conclusão de Curso (Graduação em Estatística) - Faculdade de Ciências e Tecnologia, Universidade Estadual Paulista, Presidente Prudente, 2022.

WICKHAM, H. *et al.* Welcome to the Tidyverse. **Journal of Open Source Software**, v. 4, n. 43, p. 1686, 2019. DOI: https://doi.org/10.21105/joss.01686.

APÊNDICE A – Códigos da Seção 4.2

Análise Descritiva dos Dados

```
###### Fazendo a leitura dos dados
# Base de treino
base_treino <- read.csv("Base/fraudTrain.csv")
# Base de teste
base_teste <- read.csv("Base/fraudTest.csv")

###### Juntando as duas bases
base_completa <- rbind(base_teste, base_treino)
colnames(base_completa)[1] <- "index"

###### Checando por valores nulos ou faltantes (NA) na base de dados
sum(is.null(base_completa))
sum(is.na(base_completa))

###### Proporcao de fraude e nao fraude
prop.fraude <- round(prop.table(table(base_completa$is_fraud)),4) * 100
barplot(prop.fraude,
main = "Proporcao de Transacoes Legitimas e Fraudulentas",
ylab = "%",
col = c("midnightblue", "turquoise"),
names.arg = c("Legitimas", "Fraudulentas"),
ylim = c(0,100))
text(x=0.7,y=50, paste0(prop.fraude[[1]], "%"), col = "white", font = 2)
text(x=1.9,y=10, paste0(prop.fraude[[2]], "%"), col = "black", font = 2)

###### Fraude vs Valores
medias_amt <- c(round(mean(base_completa$amt[base_completa$is_fraud ==
0]),2),
round(mean(base_completa$amt[base_completa$is_fraud == 1]),2))
barplot(medias_amt ~ unique(base_completa$is_fraud),
main = "Valor Medio das Transacoes \nSegmentado por Legitimas e
Fraudulentas",
ylab = "Valor medio ($)",
xlab = "",
col = c("midnightblue", "turquoise"),
names.arg = c("Legitimas", "Fraudulentas"),
ylim = c(0,600))
```

```
text(x=0.7,y=40, paste0(medias_amt[1], "$"), col = "white", font = 2)
text(x=1.9,y=250, paste0(medias_amt[2], "$"), col = "black", font = 2)

###### Fraude vs Genero
tabela_genero <- table(base_completa$gender, base_completa$is_fraud)
tabela_genero <- t(tabela_genero)
prop_genero <- round(prop.table(tabela_genero, margin = 2),4) * 100
barplot(prop_genero,
beside = T,
main = "Proporcao de Transacoes Legitimas e Fraudulentas \nSegmentado por
Genero",
ylab = "%",
col = c("midnightblue", "turquoise"),
legend.text = c("Legitima", "Fraudulenta"),
space = c(0,2),
args.legend = list(x="center"))
text(x=2.5,y=25, paste0(prop_genero[[1]], "%"), col = "white", font = 2)
text(x=3.5,y=3, paste0(prop_genero[[2]], "%"), col = "black", font = 2)
text(x=6.5,y=25, paste0(prop_genero[[3]], "%"), col = "white", font = 2)
text(x=7.5,y=3, paste0(prop_genero[[4]], "%"), col = "black", font = 2)

###### Fraude vs Idade
suppressMessages(suppressWarnings(library(lubridate)))
base_completa$dob <- as.Date(base_completa$dob)
dia_atual <- as.Date("2023-09-17")
base_completa$age <- floor(decimal_date(dia_atual) -
decimal_date(ymd(base_completa$dob)))
faixas_idade <- cut(base_completa$age,
breaks = c(-Inf, 19, 29, 39, 49, 59, 69, 79, Inf),
labels = c("[18; 19]", "[20; 29]", "[30; 39]", "[40; 49]",
"[50; 59]", "[60; 69]", "[70; 79]", "[80+]"))
tabela_idade <- table(faixas_idade, base_completa$is_fraud)
tabela_idade <- t(tabela_idade)
prop_idade <- round(prop.table(tabela_idade, margin = 2), 4) * 100
barplot(prop_idade,
beside = TRUE,
cex.names = 1.2,
cex.axis = 1.2,
cex.main = 2,
cex.lab = 1.5,
main = "Proporcao de Transacoes Legitimas e Fraudulentas Segmentado por
Idade",
```

```
ylab = "%",
xlab = "Idade",
col = c("midnightblue", "turquoise"),
legend.text = c("Legitimas", "Fraudulentas"),
names.arg = levels(faixas_idade),
args.legend = list(x="center", cex = 1))
aux <- 2.55
aux2 <- 1.55
for (i in 1:16) {
  if(i %% 2 == 0){
    text(x=aux,y=3, paste0(prop_idade[[i]], "%"), col = "black", font = 2,
cex = 1.2)
    aux <- aux + 3
  }
  else {
    text(x=aux2,y=90, paste0(prop_idade[[i]], "%"), col = "white", font =
2, cex = 1.2)
    aux2 <- aux2 + 3
  }
}

###### Fraude vs Hora
base_completa$hour <-
format(as.POSIXct(base_completa$trans_date_trans_time),
format = "%H")
base_completa$hour <- as.integer(base_completa$hour)
tabela_hora <- table(base_completa$hour, base_completa$is_fraud)
tabela_hora <- t(tabela_hora)
prop_hora <- round(prop.table(tabela_hora, margin = 2),4) * 100
barplot(prop_hora,
beside = T,
cex.names = 1.2,
cex.axis = 1.2,
cex.main = 2,
cex.lab = 1.5,
main = "Proporcao de Transacoes Legitimas e Fraudulentas Segmentado por
Hora",
ylab = "%",
col = c("midnightblue", "turquoise"),
ylim = c(0,100),
xlab = "Hora",
legend.text = c("Legitima", "Fraudulenta"),
```

```
args.legend = list(x="center", cex = 1),
space = c(0.1, 0.1))

###### Fraude vs Mes
base_completa$trans_month <- month(base_completa$trans_date_trans_time)
tabela_mes <- table(base_completa$trans_month, base_completa$is_fraud)
tabela_mes <- t(tabela_mes)
prop_mes <- round(prop.table(tabela_mes, margin = 2),4) * 100
barplot(prop_mes,
beside = T,
cex.names = 1.2,
cex.axis = 1.2,
cex.main = 2,
cex.lab = 1.5,
main = "Proporcao de Transacoes Legitimas e Fraudulentas Segmentado por
Mes",
ylab = "%",
col = c("midnightblue", "turquoise"),
ylim = c(0,100),
xlab = "Mes",
legend.text = c("Legitima", "Fraudulenta"),
args.legend = list(x="center", cex = 1),
space = c(0, 0))
aux <- 1.52
aux2 <- 0.52
for (i in 1:24) {
  if(i %% 2 == 0){
    text(x=aux,y=3, paste0(round(prop_mes[[i]],1), "%"), col = "black",
font = 2, cex = 1.2)
    aux <- aux + 2
  }
  else {
    text(x=aux2,y=90, paste0(round(prop_mes[[i]],1), "%"), col = "white",
font = 2, cex = 1.2)
    aux2 <- aux2 + 2
  }
}

###### Fraude vs Categoria
tabela_categoria <- table(base_completa$category, base_completa$is_fraud)
tabela_categoria <- t(tabela_categoria)
prop_categoria <- round(prop.table(tabela_categoria, margin = 2),4) * 100
```

```
barplot(prop_categoria,
beside = T,
cex.names = 1.2,
cex.axis = 1.2,
cex.main = 2,
cex.lab = 1.5,
main = "Proporcao de Transacoes Legitimas e Fraudulentas Segmentado por
Categoria",
ylab = "%",
col = c("midnightblue", "turquoise"),
ylim = c(0,100),
xlab = "Categoria",
names.arg = 1:14,
legend.text = c("Legitima", "Fraudulenta"),
args.legend = list(x="center", cex = 1),
space = c(0, 0))
aux <- 1.53
aux2 <- 0.53
for (i in 1:28) {
  if(i %% 2 == 0){
    text(x=aux,y=3, paste0(round(prop_categoria[[i]],1), "%"), col =
"black", font = 2, cex = 1.2)
    aux <- aux + 2
  }
  else {
    text(x=aux2,y=90, paste0(round(prop_categoria[[i]],1), "%"), col =
"white", font = 2, cex = 1.2)
    aux2 <- aux2 + 2
  }
}
```

APÊNDICE B – Códigos da Seção 4.4

Tratamento dos Dados e Construção dos Modelos

```
###### Fazendo a leitura dos dados
# Base de treino
base_treino <- read.csv("Base/fraudTrain.csv")
# Base de teste
base_teste <- read.csv("Base/fraudTest.csv")

###### Juntando as duas bases
base_total <- rbind(base_teste, base_treino)

###### Transformando data de nascimento em data e calculando a idade
suppressMessages(suppressWarnings(library(tidyverse)))
base_total$dob <- as.Date(base_total$dob)
dia_atual <- as.Date("2023-09-17")
base_total$age <- floor(decimal_date(dia_atual) -
decimal_date(ymd(base_total$dob)))

###### Extraindo o mes, o ano e a hora (sem min e seg) da transacao
base_total$trans_month <- month(base_total$trans_date_trans_time)
base_total$trans_year <- year(base_total$trans_date_trans_time)
base_total$hour <- format(as.POSIXct(base_total$trans_date_trans_time),
format = "%H")
base_total$hour <- as.integer(base_total$hour)

###### Calculando o periodo da transacao
base_total$trans_period <- ifelse(
base_total$hour >= 6 & base_total$hour < 20, "Daytime", "Nighttime")

###### Segmentando as cidades de acordo com a sua populacao
base_total$city_size <- ifelse(base_total$city_pop < 10000, "Small",
ifelse(base_total$city_pop >= 10000 &
base_total$city_pop < 100000, "Avg",
"Big"))

###### Calculando a diferenca em seg entre transacoes de um mesmo cartao
base_total <- base_total %>%
group_by(cc_num) %>%
mutate(trans_diff = unix_time - lag(unix_time))
```

```
###### Setando como -1 a primeira transacao de um cartao de credito (por
default e NA)
base_total[base_total$trans_diff %in% NA, "trans_diff"] <- -1

###### Convertendo a transacao de segundos para horas
base_total$trans_diff <- base_total$trans_diff/3600

###### Segmentando transacoes de um mesmo cartao
base_total$trans_rec <- ifelse(base_total$trans_diff < 0,
"First_transaction",
ifelse(base_total$trans_diff >= 0 & base_total$trans_diff < 1,
"Less_than_1h",
ifelse(base_total$trans_diff >= 1 & base_total$trans_diff < 6,
"Between_1h_and_6h",
ifelse(base_total$trans_diff >= 6 & base_total$trans_diff < 12,
"Between_6h_and_12h",
ifelse(base_total$trans_diff >= 12 & base_total$trans_diff < 24,
"Between_12h_and_24h", "After_24h")))))

###### Calculando a dist em KM do titular do cartao para o comerciante
# Funcao p/ calcular a dist entre 2 pts usando a formula de Haversine
haversine_distance <- function(lat1, lon1, lat2, lon2) {
  # Raio medio da Terra em km
  raio <- 6371
  # Converter graus para radianos
  lat1 <- lat1 * pi/180
  lon1 <- lon1 * pi/180
  lat2 <- lat2 * pi/180
  lon2 <- lon2 * pi/180
  # Diferenca de latitude e longitude entre os dois pontos
  dlat <- abs(lat2 - lat1)
  dlon <- abs(lon2 - lon1)
  # Formula de Haversine
  a <- sin(dlat/2)^2 + cos(lat1) * cos(lat2) * sin(dlon/2)^2
  aux <- 2 * asin(sqrt(a))
  distance <- raio * aux
  return(distance)
}
base_total$dist <- haversine_distance(base_total$lat,
base_total$long,
base_total$merch_lat,
```

```
base_total$merch_long)

###### Excluindo cols ja utilizadas ou irrelevantes/custosas
base_total$X <- NULL
base_total$trans_date_trans_time <- NULL
base_total$cc_num <- NULL
base_total$merchant <- NULL
base_total$first <- NULL
base_total$last <- NULL
base_total$street <- NULL
base_total$city <- NULL
base_total$state <- NULL
base_total$zip <- NULL
base_total$lat <- NULL
base_total$long <- NULL
base_total$city_pop <- NULL
base_total$job <- NULL
base_total$dob <- NULL
base_total$trans_num <- NULL
base_total$unix_time <- NULL
base_total$merch_lat <- NULL
base_total$merch_long <- NULL
base_total$hour <- NULL
base_total$trans_diff <- NULL
base_total$lat_dist <- NULL
base_total$long_dist <- NULL
base_total <- as.data.frame(base_total)

###### Categorizando as variaveis necessarias
base_total$category <- factor(base_total$category)
base_total$gender <- factor(base_total$gender)
base_total$is_fraud <- factor(base_total$is_fraud)
base_total$trans_month <- factor(base_total$trans_month)
base_total$trans_year <- factor(base_total$trans_year)
base_total$trans_period <- factor(base_total$trans_period)
base_total$city_size <- factor(base_total$city_size)
base_total$trans_rec <- factor(base_total$trans_rec)
str(base_total)
##############################################################################
##############################################################################

##############################################################################
```

```
##            Separando em base de treino e base de teste
#########################################################################

###### Particionando a base em treino (80%) e teste (20%)
suppressMessages(suppressWarnings(library(caret)))
# Semente aleatoria
set.seed(685)
# Particionando a base
aux <- createDataPartition(base_total[,"is_fraud"], p = 0.80, list = F)
# Base de treino
base_treino <- base_total[aux,]
table(base_treino$is_fraud)
# Base de teste
base_teste <- base_total[-aux,]
table(base_teste$is_fraud)
#########################################################################
#########################################################################

#########################################################################
##              Regressao Logistica (dados desbalanceados)
#########################################################################

##### Validacao via k-fold cross-validation com k=5
controle <- trainControl(method = "cv",
number = 5,
p = 0.80,
summaryFunction = multiClassSummary)
set.seed(685)
resultado_desbalanceado <- train(is_fraud ~ .,
data = base_treino,
method = "glm", family = binomial(link="logit"),
trControl = controle,
metric = "Balanced_Accuracy")

##### Verificando a importancia das variaveis no modelo
importancia_variaveis <- varImp(resultado_desbalanceado, scale = F)
importancia_variaveis$importance %>%
as.data.frame() %>%
rownames_to_column() %>%
arrange(Overall) %>%
mutate(rowname = forcats::fct_inorder(rowname)) %>%
ggplot(aes(x = fct_reorder(rowname, Overall), y = Overall, fill =
```

```
Overall)) +
 geom_col() +
 coord_flip() +
 xlab("Variavel")+
 ylab("Importancia")+
 ggtitle("Regressao Logistica - dados desbalanceados")+
 scale_fill_gradient(low = "turquoise", high = "midnightblue") +
 labs(fill = "Importancia")+
 geom_hline(yintercept = 1.96, linetype = "dashed", color = "black")+
 geom_text(x = 1, y = 2, label = "1.96", color = "black",
 hjust = -0.1, vjust = -0.5) +
 theme_minimal()

 ##### Retirando as variaveis nao significativas
 set.seed(685)
 resultado_desbalanceado <- train(is_fraud ~ . -dist -city_size,
 data = base_treino,
 method = "glm", family = binomial(link="logit"),
 trControl = controle,
 metric = "Balanced_Accuracy")

 ##### Verificando a importancia das variaveis no modelo
 importancia_variaveis <- varImp(resultado_desbalanceado, scale = F)
 importancia_variaveis$importance %>%
 as.data.frame() %>%
 rownames_to_column() %>%
 arrange(Overall) %>%
 mutate(rowname = forcats::fct_inorder(rowname)) %>%
 ggplot(aes(x = fct_reorder(rowname, Overall), y = Overall, fill =
Overall)) +
 geom_col() +
 coord_flip() +
 xlab("Variavel")+
 ylab("Importancia")+
 ggtitle("Regressao Logistica - dados desbalanceados")+
 scale_fill_gradient(low = "turquoise", high = "midnightblue") +
 labs(fill = "Importancia")+
 geom_hline(yintercept = 1.96, linetype = "dashed", color = "black")+
 theme_minimal()

 ##########################################################################
 #                          Resultados do modelo
```

```
#########################################################################
resultado_desbalanceado
summary(resultado_desbalanceado)

#########################################################################
#                            Predicao na base de teste
#########################################################################
predicao_desbalanceado <- predict(resultado_desbalanceado, base_teste)

#########################################################################
#                            Matriz de confusao
#########################################################################
MC_RL_desbalanceado <- caret::confusionMatrix(predicao_desbalanceado,
base_teste[,"is_fraud"],
positive = "1")
MC_RL_desbalanceado

#########################################################################
#                                Curva ROC
#########################################################################
suppressMessages(suppressWarnings(library(pROC)))
roc_obj_desbalanceado <- roc(as.numeric(base_teste$is_fraud),
as.numeric(predicao_desbalanceado))
plot(roc_obj_desbalanceado, main = "Curva ROC", col = "blue", lwd = 3,
print.auc = T, legacy.axes = T)

#########################################################################
#########################################################################

#########################################################################
##                            Balanceando os dados
#########################################################################
dados_balanceados <- downSample(base_treino[,-4], base_treino[,4],
yname = "is_fraud")
str(dados_balanceados)

#########################################################################
##              Regressao Logistica (dados balanceados)
#########################################################################
controle <- trainControl(method = "cv",
number = 5,
p = 0.80,
```

```
summaryFunction = multiClassSummary)

set.seed(685)
resultado_balanceado <- train(is_fraud ~ .,
data = dados_balanceados,
method = "glm", family = binomial(link="logit"),
trControl = controle,
metric = "Balanced_Accuracy")

########## Verificando a importancia das variaveis no modelo
importancia_variaveis <- varImp(resultado_balanceado, scale = F)
importancia_variaveis$importance %>%
as.data.frame() %>%
rownames_to_column() %>%
arrange(Overall) %>%
mutate(rowname = forcats::fct_inorder(rowname)) %>%
ggplot(aes(x = fct_reorder(rowname, Overall), y = Overall, fill =
Overall)) +
geom_col() +
coord_flip() +
xlab("Variavel")+
ylab("Importancia")+
ggtitle("Regressao Logistica - dados balanceados")+
scale_fill_gradient(low = "turquoise", high = "midnightblue") +
labs(fill = "Importancia")+
geom_hline(yintercept = 1.96, linetype = "dashed", color = "black")+
theme_minimal()

##### Retirando as variaveis nao significativas
set.seed(685)
resultado_balanceado <- train(is_fraud ~ . -dist -city_size,
data = dados_balanceados,
method = "glm", family = binomial(link="logit"),
trControl = controle,
metric = "Balanced_Accuracy")

########## Verificando a importancia das variaveis no modelo
importancia_variaveis <- varImp(resultado_balanceado, scale = F)
importancia_variaveis$importance %>%
as.data.frame() %>%
rownames_to_column() %>%
arrange(Overall) %>%
```

```
mutate(rowname = forcats::fct_inorder(rowname)) %>%
ggplot(aes(x = fct_reorder(rowname, Overall), y = Overall, fill =
Overall)) +
geom_col() +
coord_flip() +
xlab("Variavel")+
ylab("Importancia")+
ggtitle("Regressao Logistica - dados balanceados")+
scale_fill_gradient(low = "turquoise", high = "midnightblue") +
labs(fill = "Importancia")+
geom_hline(yintercept = 1.96, linetype = "dashed", color = "black")+
theme_minimal()

#########################################################################
#                            Resultados do modelo
#########################################################################
resultado_balanceado
summary(resultado_balanceado)

#########################################################################
#                            Predicao na base de teste
#########################################################################
predicao_balanceado <- predict(resultado_balanceado, base_teste)

#########################################################################
#                            Matriz de confusao
#########################################################################
MC_RL_balanceado <- caret::confusionMatrix(predicao_balanceado,
base_teste[,"is_fraud"],
positive = "1")
MC_RL_balanceado

#########################################################################
#                                Curva ROC
#########################################################################
roc_obj_balanceado <- roc(as.numeric(base_teste$is_fraud),
as.numeric(predicao_balanceado))
plot(roc_obj_balanceado, main = "Curva ROC", col = "blue", lwd = 3,
print.auc = T, legacy.axes = T)
#########################################################################
#########################################################################
```

```
#########################################################################
#             Floresta Aleatoria (dados desbalanceados)
#########################################################################
suppressMessages(suppressWarnings(library(randomForest)))
controle <- trainControl(method = "cv",
number = 5,
p = 0.80,
summaryFunction = multiClassSummary)
set.seed(685)
resultado_desbalanceado_fa <- train(x=base_treino[,-4],
y=base_treino[,4],
method = "rf", sampsize = 10000,
trControl = controle,
metric = "Balanced_Accuracy")

##### Valor otimo de mtry
plot(resultado_desbalanceado_fa,
xlab = "Valor de mtry",
ylab = "Acuracia Balanceada")

#########################################################################
#                           Resultados do modelo
#########################################################################
resultado_desbalanceado_fa
resultado_desbalanceado_fa$finalModel

#########################################################################
#                           Predicao na base de teste
#########################################################################
predicao_desbalanceado_fa <- predict(resultado_desbalanceado_fa,
base_teste)

#########################################################################
#                           Matriz de confusao
#########################################################################
MC_FA_desbalanceado <- caret::confusionMatrix(predicao_desbalanceado_fa,
base_teste[,"is_fraud"],
positive = "1")
MC_FA_desbalanceado
```

```
#########################################################################
#                                Curva ROC
#########################################################################
roc_obj_desbalanceado_fa <- roc(as.numeric(base_teste$is_fraud),
as.numeric(predicao_desbalanceado_fa))
plot(roc_obj_desbalanceado_fa, main = "Curva ROC", col = "blue", lwd = 3,
print.auc = T, legacy.axes = T)
#########################################################################
#########################################################################

#########################################################################
#             Floresta Aleatoria (dados balanceados)
#########################################################################
controle <- trainControl(method = "cv",
number = 5,
p = 0.80,
summaryFunction = multiClassSummary)
set.seed(685)
resultado_balanceado_fa <- train(x=dados_balanceados[,-11],
y=dados_balanceados[,11],
method = "rf", sampsize = 10000,
trControl = controle,
metric = "Balanced_Accuracy")

##### Valor otimo de mtry
plot(resultado_balanceado_fa,
xlab = "Valor de mtry",
ylab = "Acuracia Balanceada")

#########################################################################
#                            Resultados do modelo
#########################################################################
resultado_balanceado_fa
resultado_balanceado_fa$finalModel

#########################################################################
#                            Predicao na base de teste
#########################################################################
predicao_balanceado_fa <- predict(resultado_balanceado_fa, base_teste)

#########################################################################
#                            Matriz de confusao
```

```
#########################################################################
MC_FA_balanceado <- caret::confusionMatrix(predicao_balanceado_fa,
base_teste[,"is_fraud"],
positive = "1")
MC_FA_balanceado

#########################################################################
#                                Curva ROC
#########################################################################
roc_obj_balanceado_fa <- roc(as.numeric(base_teste$is_fraud),
as.numeric(predicao_balanceado_fa))
plot(roc_obj_balanceado_fa, main = "Curva ROC", col = "blue", lwd = 3,
print.auc = T, legacy.axes = T)
#########################################################################
#########################################################################
```