UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
CÂMPUS DE PRESIDENTE PRUDENTE

Evelyn Rocha

Segmentação do Perfil de Clientes Inadimplentes Utilizando Ferramentas Computacionais

Presidente Prudente

2021/2022

Evelyn Rocha

Segmentação do Perfil de Clientes Inadimplentes Utilizando Ferramentas Computacionais

Relatório Final para Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Estatística da FCT/Unesp para aproveitamento na disciplina Trabalho de Conclusão de Curso. Orientador: Prof. Dr. Manoel Ivanildo Silvestre Bezerra.

Presidente Prudente

2021/2022

R672s

Rocha, Evelyn

Segmentação do perfil de clientes inadimplentes utilizando ferramentas computacionais / Evelyn Rocha. -- Presidente Prudente, 2022

79 p.

Trabalho de conclusão de curso (Bacharelado - Estatística) - Universidade Estadual Paulista (Unesp), Faculdade de Ciências e Tecnologia, Presidente Prudente

Orientador: Manoel Ivanildo Silvestre Bezerra

1. Estatística. 2. Análise de regressão logística. 3. Inadimplência (Finanças). 4. Big data. 5. Programas de computador. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Faculdade de Ciências e Tecnologia, Presidente Prudente. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

TERMO DE APROVAÇÃO

Evelyn Rocha

Segmentação do Perfil de Clientes Inadimplentes Utilizando Ferramentas Computacionais

Relatório Final de Trabalho de Conclusão de Curso aprovado como requisito para obtenção de créditos na disciplina Trabalho de Conclusão do curso de graduação em Estatística da Faculdade de Ciências e Tecnologia da Unesp, pela seguinte banca examinadora:

Orientador: _____

Prof. Dr. Manoel Ivanildo Silvestre Bezerra
Departamento de Estatística

_____

Prof. Dr. José Gilberto Spasiani Rinaldi
Departamento de Estatística

_____

Profª. Drª. Silvely Nogueira De Almeida Salomão Néia
Departamento de Estatística

Presidente Prudente, 26 de março de 2022.

Agradecimentos

Primeiramente agradeço ao professor Manoel por todo o carinho e atenção que teve comigo durante toda a minha gradução, fazendo muitas vezes mais do que o papel de professor mas sim um amigo, um familiar. Agradeço também a todos professores que passaram pelo menos um pouquinho de seu vasto conhecimento pra mim durante todo esse período da faculdade, sem isso eu não estaria nessa etapa.

Por fim mas não menos importante, agradeço a minha família por todo o investimento em mim, por acreditarem que eu me tornaria algo e hoje posso dizer que me tornei, creio que a conclusão desse ciclo não é só meu, é nosso. Em especial deixo esse trabalho em homenagem a minha vó dona Maria Aparecida que perdi recentemente, minha maior inspiração de bondade e de amor.

Resumo

A inadimplência é um termo muito presente na vida dos brasileiros, visto que em 2021 o Brasil teve o seu maior nível de endividamento médio em 11 anos. Diante disso, áreas de cobrança e recuperação de crédito buscam esforços para melhorar cada vez mais seus métodos de abordagem ao cliente, sempre visando a melhor estratégia e a busca pelo lucro. A partir disso, é possível ver o quanto estudos com abordagem em recuperação de crédito são importantes para melhorar processos e proporcionar ações preventivas, sendo esses os objetivos deste trabalho.

No decorrer do texto é possível entender o caminho do crédito até o processo de cobrança e recuperação de crédito, entender como algumas ferramentas computacionais funcionam, absorver de forma sucinta a teoria de regressão logística além de vê-la na prática a partir de uma abordagem mista entre algoritmos de software Python e R. Nas análises realizadas, foi demonstrado ser viável a construção de um modelo preditivo para segmentar o perfil de clientes inadimplentes, sendo a variável resposta o cliente ter um perfil digital ou não.

Palavras-chave: Inadimplência. Regressão logística. Ferramentas Computacionais.

Abstract

Default is a term that is very present in the lives of Brazilians, since in 2021 Brazil had its highest average debt level in 11 years. In addition, credit collection and recovery areas sought to increasingly improve their methods of approaching the customer, always aiming at the best strategy and the pursuit of profit. From this, it is possible to see how many studies with a credit recovery approach are important to improve processes and provide preventive actions, being these objectives of this work.

Throughout the text it is possible to understand the credit path or the credit collection and recovery process, understand how some computational tools work, succinctly absorb the resource theory in logistics in addition to seeing it in practice from a mixed approach. among other Python and R software. In those evaluated under construction, the pre-feasible model to segment the profile of customers into non-defaulting customers was shown to be viable, whether a digital profile or not.

Keywords: Default. Logistic Regression. Computational Tools.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Instituições financeiras especializadas em intermediar e custodiar o dinheiro de grande parte da população desde sempre lidam com um tema não muito agradável que segue atrelado ao risco de crédito, a inadimplência.

No decorrer dos anos, a grande expansão do mercado de crédito foi tratada como um mar de sucesso e oportunidades para a economia como um todo. Junto a essa expansão, atrelou-se o aumento dos devedores no país, pessoas que se tornaram inadimplentes ao descumprirem suas obrigações financeiras. Esse tema vem obtendo cada vez mais força no último ano com a chegada da pandemia do novo coronavírus, que foi responsável por um aumento no número de devedores.

Segundo PEREIRA (2019), o Brasil é conhecido como um dos países com o maior número de inadimplentes, chegando aproximadamente a 60 milhões de pessoas com contas em atraso. Esse número chega a ser maior do que a população de quase todos os países da América Latina, e equivalente ao tamanho da Itália.

Com a ascendência da inadimplência no mercado financeiro, a criação de um setor especializado na gestão de faturas em aberto foi necessária e de grande importância, já que o mesmo é responsável pela renegociação de dívidas, prevenção de inadimplência, criação de estratégias de cobrança e minimização de *Churn* (indicador de perda de clientes), que pode ser traduzido na área de recuperação como a quebra de uma promessa ou o cancelamento de um cartão pela falta de pagamento.

Para tornar a recuperação de crédito efetiva e contemplar o maior número de inadimplentes, escritórios de cobrança são contratados para fazer o acionamento dos clientes a partir de cinco principais ferramentas: *SMS* (Short Message Service), *E-Mail*, *Voicer* (mensagem de áudio), P.A. Humana (Ponto de Atendimento em que o cliente conversa em tempo real com um atendente por ligação), P.A. Digital (Ponto de Atendimento em que o cliente conversa por texto em tempo real com um *chatbot* (atendente virtual)).

No contexto de escritórios de cobrança temos os chamados: escritórios digitais e escritórios não digitais. Os escritórios de cobrança tradicionais ou aqui tratados como "não digitais", são os escritórios que envolvem o atendimento humano, que hoje ainda são a maioria no mercado de recuperação de crédito, eles oferecerem uma maior comodidade para o cliente, porém possuem as maiores taxas de cobrança do mercado. Além dos escritórios de cobrança tradicionais, existem os chamados escritórios digitais, criados com a intenção de agilizar o atendimento e diminuir o custo associado a cobrança em comparação aos escritórios tradicionais, tornando-se assim um meio de cobrança que vem ganhando cada vez mais força na área de recuperação.

Dessa forma fica claro como conhecer seu cliente é algo importante na atualidade, a segmentação do mercado de inadimplentes pode gerar muitas oportunidades e *insights* (capacidade de obter uma compreensão intuitiva precisa e profunda) para a área, auxiliando na definição de estratégias para o aumento da arrecadação e a diminuição de custos envolvidos no processo, objetivos desse trabalho. Para isso serão utilizadas técnicas estatísticas de Regressão Logística e *Random Forest* com o auxílio de ferramentas computacionais utilizadas no mercado de trabalho.

2 Mercado de Crédito e Cobrança

Nascido com a meta de ser uma oportunidade de conquistar mais clientes, o crédito se estabeleceu como uma das palavras mais importantes do mundo financeiro, atrelando-se a recuperação de crédito, área que cresce exponencialmente e eleva a gestão de crédito a um novo e importante patamar. Por isso, este capítulo visa apresentar algumas etapas importantes do mercado de crédito e cobrança, área a ser explorada neste trabalho.

2.1 Crédito

Crédito, palavra originária do latim *creditum* com o significado de confiança, empréstimos ou dívidas, refere-se basicamente à troca de um valor presente por uma promessa de reembolso no futuro, que pode ocorrer ou não em virtude do fator de risco, segundo SANTOS (2006).

A oferta de crédito por parte de empresas e instituições financeiras é vista como um recurso estratégico para gerar mais receita e também oferecer um maior poder de compra para os clientes. As vendas a prazo ou usualmente chamadas de vendas no crédito continuam sendo a melhor forma de facilitar e alavancar vendas, por possibilitar que o consumidor obtenha um produto no momento que desejar, segundo SILVA (2000). Dessa forma é possível concluir que mesmo correndo riscos de perdas, a concessão de crédito pode ser considerada como um dos pilares da economia atual.

2.1.1 Políticas de Crédito

Determinada pela empresa ou instituição financeira, a política de crédito é um conjunto de normas e critérios para aprovar ou negar a concessão do crédito a seus clientes. Por ser um recurso essencial para a segurança financeira do negócio, é preciso assegurar que ela esteja alinhada com a realidade e objetivos da companhia para que seus resultados sejam consistentes e seja possível identificar qual é o perfil de cliente mais adequado ou interessante para a empresa. Apesar de ser visível o quão importante é possuir uma boa política de crédito, muitas empresas utilizam critérios generalistas para sua criação, o que pode refletir negativamente em seu resultado.

Para a criação de uma política de crédito, segundo SÁ (2004), vários fatores devem ser considerados com base em questionamentos e particularidades do negócio para que seja possível atingir os objetivos estratégicos e garantir o equilíbrio financeiro. Antes de começar a estruturá-la, é importante fazer um levantamento de todos os dados existentes

dentro da empresa para conhecer bem o perfil dos clientes, além de buscar alguns conceitos de mercado, como os cinco C's de crédito, definidos abaixo:

1. Caráter: trata-se do risco moral, a propensão do cliente em pagar ou não o valor;

2. Capacidade: condições do cliente para o pagamento;

3. Capital: situação financeira do cliente;

4. Condições: variáveis da economia no momento em que o crédito é concedido;

5. Colateral: garantias que o cliente oferece para pagamento do crédito.

O benefício de possuir uma política de crédito bem estruturada é possibilitar a redução dos índices de inadimplência, permitindo um volume de concessão de crédito mais alto e equilibrado, além de proporcionar diretamente mais vendas para a empresa. Também há uma diminuição de custos operacionais e maior segurança nas análises, deixando de serem imparciais e subjetivas para desempenharem um papel padronizado e embasado em informações fundamentadas.

2.1.2 Análise de Crédito

Segundo HOJI (2000), ao solicitar crédito no mercado sempre há uma expectativa sobre a aprovação do mesmo por parte dos consumidores, porém, antes de receber uma resposta, existe uma etapa muito importante: a análise de crédito. A partir dessa etapa, a instituição financeira irá definir o valor liberado de crédito, as taxas de juros envolvidas na operação e a quantidade de parcelas oferecidas para o pagamento do montante.

A análise de crédito é um processo criado para dar segurança às empresas e instituições financeiras que oferecem crédito a um consumidor. Ela é um processo necessário em qualquer operação financeira, como por exemplo, solicitar um novo cartão de crédito, limite de cheque especial, empréstimos, financiamentos, crediários e até para abrir uma conta bancária, já que as informações do consumidor precisam ser avaliadas pela empresa.

O processo de análise de crédito pode variar de acordo com a instituição, mas costuma ter alguns pontos em comum, como a consulta dasvseguintes informações sobre o solicitante:

Dados Pessoais: Informações básicas sobre o consumidor, como o CPF, telefone, estado civil, nível de escolaridade, profissão, renda e endereço.

Restrições no Nome: Além dos dados pessoais, a instituição precisará saber se o cliente tem alguma pendência com ela, que, em caso positivo, pode impossibilitar a tomada de um novo crédito, já que as chances de inadimplência são maiores. A maioria

das empresas também costuma consultar os "birôs de crédito", ou órgãos de proteção ao crédito, para conseguir informações mais detalhadas, como saber se o consumidor tem dívidas em atraso com outras instituições, o que pode indicar que ele é um mau pagador.

Os birôs de crédito também fornecem informações sobre o *score*, ou pontuação de crédito do consumidor. O valor varia de 0 a 1.000 e é calculado pelos próprios birôs por meio da avaliação de seu histórico de dívidas e de pagamentos, sendo o objetivo dessa consulta entender qual é o risco de inadimplência do consumidor ao pedir crédito no mercado. Quanto mais perto do 1.000, melhor é o score e maiores são as chances de ter crédito aprovado.

Cadastro Positivo: O cadastro positivo reúne informações sobre bons pagadores, onde as empresas podem acessar os dados do comportamento e situação financeira de cada indivíduo ou empresa, com todos os pagamentos feitos em dia. Após consultar as informações do consumidor que paga as contas sem atraso, as instituições financeiras têm condições de oferecer crédito com taxas menores e melhores condições de pagamento, além de conhecer a parcela da população sem emprego formal, que também pode ser beneficiada por terem seus dados no sistema, servindo como comprovação de capacidade de pagamento.

Renda: A renda mensal do consumidor também é uma informação importante para garantir a segurança da operação e do próprio solicitante. Isso serve para saber se o cliente realmente tem a possibilidade de pagar o valor de uma parcela ou se o limite do cartão está adequado ao salário, sendo que geralmente uma despesa não pode ultrapassar 30% do seu salário para que o orçamento não fique comprometido.

Garantia: No caso das operações de crédito que envolvem um bem como garantia de pagamento, como o refinanciamento de imóvel ou refinanciamento de veículo, a propriedade também é analisada. Nesse caso, a casa ou carro do solicitante passam por uma vistoria para que a instituição consiga avaliar o estado de conservação do bem e estimar o seu valor de mercado. Essas informações também são importantes, pois quanto mais valioso for o imóvel ou automóvel, menores ficam as taxas de juros da operação.

Para MARQUES (2020), uma questão importante é que nenhuma instituição pode ter acesso aos perfil de crédito do consumidor e fazer uma análise sem a autorização do cliente, dada pela resolução n 4.571, parágrafo 4, em que o Banco Central determina que a credora deve guardar por pelo menos cinco anos a autorização de consulta ao SCR (Sistema de Informações de Créditos do Banco Central) feita por meio físico ou eletrônico, independentemente da realização da operação de crédito. Ou seja, as empresas precisam do consentimento do cliente para fazer qualquer consulta, mesmo que o empréstimo de crédito não seja concretizado, pois isso garante que as informações do cliente mantenham-se confidenciais e não possam ser utilizadas para outra finalidade além da informada ao solicitante.

2.1.3 Risco de Crédito

Risco de crédito é a probabilidade de um cliente não pagar o valor devido a uma empresa ou instituição financeira que lhe concedeu algum crédito, segundo BLATT (1999). Esse risco existe em qualquer operação financeira que envolva confiança, liberação de algum tipo de crédito com a pressuposição de recebimento do pagamento. Abaixo é possível ver alguns exemplos de risco de crédito:

Empréstimos: Quando um consumidor solicita um empréstimo, a instituição financeira pode liberar o valor pedido, mas não tem qualquer garantia de que irá recebê-lo, portanto, está exposta ao risco de crédito.

Cartões de Crédito: As administradoras de cartões de crédito disponibilizam um limite de uso aos seus clientes, mas podem sofrer calote se eles não pagarem a fatura.

Parcelamentos em Lojas e Comércios: Seja por meio de crediários ou ao vender "fiado", os proprietários dos estabelecimentos também enfrentam o chamado risco de crédito.

Contratos de Aluguel: Quando uma imobiliária firma um contrato de aluguel por determinado período, em tese, confia que o locatário cumprirá as exigências descritas e pagará os valores combinados todos os meses.

Após a etapa de análise de crédito, o risco de crédito de cada cliente pode ser classificado em duas categorias:

Risco de Primeira Classe: O risco de primeira classe é quando o crédito tem grandes chances de não ser quitado, sendo que, nesses casos, a instituição financeira pode optar por não liberar o valor solicitado ou fazer a aprovação do crédito com condições menos atrativas.

Risco de Segunda Classe: O risco de segunda classe é quando a empresa entende que sofre menos riscos ao oferecer crédito para um cliente. Dessa forma, os valores liberados e as condições de pagamento costumam ser mais interessantes.

Como o Brasil é um país com alto índice de endividamento, a análise de risco de crédito é muito importante, pois graças a ela as instituições financeiras e demais empresas credoras conseguem avaliar individualmente o perfil de cada consumidor e oferecer condições de pagamento personalizadas e mais justas. Conforme SANTOS (2006), sem essa avaliação, as empresas considerariam risco de crédito alto para todos, o que tornaria os empréstimos mais caros e as condições de pagamento menos atrativas.

2.2 Inadimplência

Inadimplência é o descumprimento de alguma obrigação financeira, quando não é realizado algum pagamento previsto em contrato até a sua data de vencimento. Esse é um termo de grande foco, já que em janeiro de 2021 aproximadamente 67 milhões de brasileiros, ou 41% da população, estavam em inadimplência segundo o SPC Brasil e CNDL (Confederação Nacional de Dirigentes Lojistas).

Vale ressaltar que a inadimplência é diferente de dívida, pois pode-se ficar inadimplente com uma dívida mas nem toda dívida é uma inadimplência. Por exemplo, uma compra parcelada no cartão de crédito é uma dívida, já que são parcelas futuras que deverão ser pagas em determinado momento, porém, na prática o parcelamento é um montante em aberto que o cliente ainda deve pagar. A inadimplência, por sua vez, acontece quando alguma dívida não é paga dentro de determinado período, por isso, não pagar determinada dívida pode deixar o consumidor inadimplente.

No Brasil, a consequência mais comum da inadimplência é ter uma restrição de nome e CPF gerada pelos órgãos de proteção ao crédito (SPC, Serasa e Boa Vista), indicando para o mercado que a pessoa é má pagadora, já que ela possui débitos em aberto e isso pode gerar as seguintes dificuldades:

- Conseguir um empréstimo;
- Fazer qualquer tipo de financiamento, seja de imóvel, carro ou outros bens;
- Abrir uma conta corrente ou adquirir um novo cartão de crédito (para quem já é correntista, o banco pode bloquear o cheque especial e cancelar a emissão de novos talões de cheque);
- Ter o valor devido inicialmente aumentar cada vez mais: quanto maior o tempo que a pessoa fica inadimplente, maiores os juros que vão incidir sobre o montante.

2.3 Cobrança e Recuperação de Crédito

Para uma empresa ou instituição financeira se manter e obter sucesso a longo prazo, é preciso manter um fluxo de caixa saudável, algo desafiador no contexto de inadimplência, onde a empresa é obrigada a arcar com os custos de empréstimoss de crédito mal sucedidos, resultando na falta de recursos para cumprir obrigações e obter lucro. Para lidar com essa questão e garantir a manutenção da saúde financeira do negócio, a empresa pode adotar algumas estratégias, como melhorar a gestão de crédito, recuperação de crédito e cobranças, segundo TAVARES (1988).

As operações de cobrança são realizadas em razão de um valor devido pelo consumidor. Em resumo, elas são usadas para cobrar ou renegociar uma dívida específica, e por isso são mais complexas. Nesses casos, o profissional responsável pelo contato com o cliente deve apresentar uma oportunidade interessante de regularização. Normalmente, as empresas desenvolvem uma régua de cobrança para definir a linguagem usada durante as operações e quais vantagens poderão ser oferecidas ao devedor para que ele regularize a sua situação. É importante lembrar que uma campanha de cobrança pode ter motivação judicial, pois trata-se de um aviso amistoso antes que a quantia seja protestada na Justiça.

A recuperação de crédito é diferente da operação de cobrança porque o seu foco não é somente fazer consumidor liquidar uma dívida, mas sim ajudar o inadimplente a regularizar sua situação para voltar a ter acesso a financiamentos e outras modalidades de crédito. Esse tipo de ação é normalmente voltado a consumidores endividados que já foram inseridos em cadastros negativos de órgãos como a Boa Vista SCPC e, por esse motivo, perderam o direito ao crediário em uma instituição financeira.

Nesses casos, a negociação pode até envolver mais de uma instituição, e a abordagem também precisa ser bem elaborada. Assim, o consumidor terá uma clara visão dos benefícios de ter acesso ao crédito.

No final das contas, o objetivo da empresa é receber o dinheiro que cada cliente lhe deve. Conforme SANTOS (2006), utilizar indicadores de performance, que avaliem periodicamente os serviços prestados, é uma boa maneira de garantir que as cobranças estão sendo feitas de maneira eficaz. No final do processo, um indicador positivo para averiguar o sucesso desse tipo de operação é o número de pessoas com crédito restabelecido e dívidas quitadas após o contato.

2.3.1 PDD

PDD, provisão para devedores duvidosos é uma reserva realizada para cobrir eventuais perdas com a inadimplência. Não trata-se, ainda, de um crédito não recebido mas sim da possibilidade de não recebê-lo. Contabilmente, também é chamada de PCLD (provisão para crédito de liquidação duvidosa) e reduz o lucro da empresa quando é provisionado e aumenta quando são feitas "retiradas" da conta de provisionamento pelo não acontecimento da inadimplência.

No caso do segmento bancário, em específico, a Resolução 2682 de 21 de dezembro de 1999 do CMN e suas alterações posteriores determina que as instituições financeiras classifiquem suas operações de crédito em ordem crescente de nível de risco. Serve como base para classificação, no mínimo, as informações em relação ao devedor e as garantias e a operação em si, uma vez que diferentes clientes e diferentes operações possuem diferentes riscos. Os bancos têm alguma liberdade para determinar seus modelos de

rating (Classificação de risco de crédito), desde que esse modelo seja aprovado pelo órgão regulador.

A tabela mínima obrigatória é a seguinte (sendo que muitas instituições utilizam valores maiores):

Quadro 1 - PDD (Provisão de Devedores Duvidosos)
Resolução 2682/99

Nível de Risco	**Faixa de Atraso**	**% Perda Esperada**
AA	Em dia	0,00 %
A	Até 15 dias de atraso	0,50 %
B	15-30 dias de atraso	1,00 %
C	31-60 dias de atraso	3,00 %
D	61-90 dias de atraso	10,00 %
E	91-120 dias de atraso	30,00 %
F	121-150 dias de atraso	50,00 %
G	151-180 dias de atraso	70,00 %
H	>180 dias de atraso	100,00 %

Fonte: Elaborado pela autora

Os bancos revisam suas operações mensalmente, realizando assim o ajuste de crédito ou débito na respectiva conta do balanço. Outro conceito importante é o de arrasto por contágio: uma operação de pior *rating* de um cliente ou grupo econômico acaba por contagiar todas as operações daquele cliente ou grupo. Algumas situações que afetam o volume provisionado são:

- Manutenção de *rating* + aumento das exposições = mais despesa
- Manutenção de *rating* + redução nas exposições = mais receita
- Melhora na classificação de *rating* + manutenção das exposições = mais receita
- Piora na classificação de *rating* + manutenção das exposições = mais despesa

É importante saber que a renegociação de uma dívida mantém o *rating* da nova operação, pelo menos, no mesmo nível da operação original. Quando uma operação é baixada a prejuízo, todo o valor provisionado é revertido, pois ela de fato passa a ser considerada um prejuízo então perde sentido provisionar sobre ela.

A regra da PDD tem impacto direto e significativo no custo do crédito no Brasil, girando em torno de 40% em tempos considerados normais, pois é uma espécie de "margem

de segurança" acumulada pelas instituições a fim de manter o lucro e a liquidez dos negócios, já em momentos de crise ecônomica como o vivido hoje, os banco fazem provisionamentos ainda maiores para melhor enfrentar o futuro incerto.

2.3.2 Métodos de Cobrança

Uma das maiores dificuldades e principal objetivo do meio de cobrança e recuperação de crédito é conhecer o perfil de seus consumidores. Como pôde ser visto até aqui, existem várias etapas, sendo que desde a concessão de crédito até a cobrança e recuperação de crédito, todas são de grande importância.

Neste ponto, os métodos de cobrança relacionados ao banco de dados utilizado neste trabalho serão retratados, de tal forma que seja de fácil entendimento o objetivo da análise aqui realizada, objetivo esse de conhecer, generalizadamente, qual meio de cobrança é mais eficiente para a carteira de clientes da financeira de uma empresa do varejo, podendo ter como resposta métodos de cobrança digital ou não.

2.3.2.1 Digitais

A cobrança digital é uma forma de enviar faturas e contas utilizando ferramentas online para realizar todos os processos. Dessa forma, as empresas podem usar canais e dispositivos digitais para realizar essa demanda de forma muito mais prática.

As soluções mais modernas do mercado permitem à empresa administrar a cobrança de dívidas de forma organizada, mantendo o controle de todo o processo. Mesmo para quem opta por terceirizar a cobrança de dívidas, esse tipo de organização é essencial. O faturamento digital também significa custos de negócios mais baixos, assim, as despesas do setor de cobrança acabam sendo menores e a empresa pode utilizar essa economia para investir em outras demandas.

A cobrança digital como uma solução eficaz e moderna acaba tendo resultados mais efetivos, tanto na recuperação de crédito quanto na manutenção de uma carteira de clientes, a tendência é que essa forma de cobrança continue a ser cada vez mais presente no mundo dos negócios. É importante salientar que a cobrança digital não substitui a necessidade de profissionais que atuem frente ao setor de cobrança, no entanto, ela surge de forma complementar, auxiliando na gestão e promovendo melhores resultados, que também dependem do perfil da carteira de clientes.

Por meio dos canais digitais, a empresa pode enviar notificações de cobrança por SMS ou e-mail, tornando a comunicação com o devedor mais efetiva, além da utilização de um portal de negociação que facilita o acordo com o cliente e o pagamento de valores em aberto. Os canais digitais utilizados para cobrança dos clientes da empresa em estudo são:

- **SMS** (***Short Message Service***): O SMS é um dos canais mais velozes e de maior alcance que existem, além de ser um forte meio de cobrança para as empresas. Segundo uma pesquisa, mais de 90% das mensagens enviadas são lidas imediatamente após o recebimento e a taxa de retorno chega até 45%. Portanto, o envio de mensagens se enquadra como uma estratégia poderosa para sanar o problema de inadimplência.

  Outra vantagem é que as mensagens por esse canal são mais diretas. Por isso, é comum o envio de código de barras de boletos para pagamento, segunda via de boletos e lembretes de data de vencimento de compra.

- ***Email***: Correio eletrônico que permite o envio de cobranças, código de barras de boletos para pagamento, segunda via de boletos e lembretes de data de vencimento de compra.

- ***Voicer ou Voicebot (mensagem de áudio)***: O voicebot permite automatizar todo o processo de cobrança, localização, abordagem, negociação e auditoria de uma operação, tudo isso de forma humanizada e interativa. Dessa forma, os agentes humanos não precisam mais se preocupar com as tarefas repetitivas e previsíveis do processo de cobrança, podendo focar em assuntos mais complexos e com maior valor agregado. Portanto, o robô de voz pode atuar também na prevenção de atrasos de pagamento, realizando avisos e lembranças da data de pagamento, além de enviar segunda via de boletos.

- **P.A. Digital**: Ponto de Atendimento digital em que o cliente conversa em tempo real com um *chatbot* ou atendente virtual.

2.3.2.2 Não Digitais

Ao contrário da cobrança digital, a cobrança tradicional não é feita por meio de canais digitais e mesmo sendo o meio de cobrança mais caro para a empresa contratante, atualmente ainda é o método mais bem aceito pelos consumidores. Os canais não digitais ou tradicionais utilizados para cobrança dos clientes da empresa em estudo são:

- **P.A. Humana**: Ponto de Atendimento Humano em que o cliente negocia seus débitos com uma pessoa, sendo ela da própria empresa que o cliente deve ou de um escritório de cobrança terceirizado.

- **Negativação**: Indicação da dívida vencida e não paga junto ao banco de dados da entidade de proteção de crédito, identificando o devedor e informando o não pagamento.

Mesmo havendo a comprovação de que os métodos de cobrança digital sejam efetivos e mais econômicos, eles não podem ser utilizados em todos os tipos de cobranças.

Isso acontece pelo fato de que o perfil de clientes pode variar muito de uma empresa para outra ou até mesmo de um produto para outro.

Enquanto alguns perfis preferem um atendimento mais rápido e sem contato humano que os meios de cobrança digital podem oferecer, outros descartam essa opção e preferem a abordagem tradicional feita por humanos. Isso pode ocorrer pelo alto volume de golpes que em 2021 bateu a marca de 4 milhões de tentativas de fraudes digitais segundo o SERASA (2022), ou até mesmo pela inabilidade de algumas pessoas com os meios digitais.

3 Ferramentas Computacionais

Este capítulo tem o propósito de demonstrar algumas ferramentas utilizadas no contexto de *Big Data* e no presente trabalho, explicando suas funcionalidades, além de vantagens e desvantagens nos processos a serem citados.

3.1 Big Data

Apesar de seu uso ter se tornado mais frequente em tempos recentes, o termo *Big Data* nasceu ainda na década de 1990 na NASA (*National Aeronautics and Space Administration* – Administração Nacional da Aeronáutica e Espaço). Na época, o *Big Data* era utilizado na descrição de conjuntos de dados complexos que desafiavam os tradicionais limites computacionais de captura, processamento, análise e armazenamento de informações.

Em 2001, o então vice-presidente e diretor de pesquisas da *Enterprise Analytics Strategies* (Estratégias Analíticas Empresariais), Doug Laney, articulou a definição de *Big Data* em três V's: Volume, Variedade e Velocidade, que 12 anos mais tarde se tornaram seis V's com a adição de: Valor, Volatilidade e Veracidade, pelo chefe de dados da *Express Scripts*, Inderpal Bhandar. Com o avanço do *Big Data*, notou-se necessária a criação de um sétimo V referente a visualização, visando facilitar e agilizar de forma intuitiva as tomadas de decisões.

Devido à sua eficiência, as organizações começaram a perceber o poder do uso do *Big Data* e segundo segundo a Forbes, já em 2015 cerca de 90% da empresas de nível médio a grande porte já haviam investido em *Big Data*.
Os sete V's do *Big Data* são definidos por:

1. Volume
O *Big Data* agrupa uma enorme quantidade de dados que são gerados a cada segundo, tendo que lidar com eficiência e possibilitando o seu agrupamento através de softwares.

2. Velocidade
É a agilidade com a qual os dados são produzidos e manipulados, muitas vezes analizados no instante em que são criados sem a necessidade de serem armazenados.

3. Variedade
Os dados podem ser gerados em formatos estruturados, como os numéricos ou não-estruturados, como arquivos de áudio, texto e vídeo.

4. Valor
De nada adianta ter acesso a uma grande quantidade de informação se ela não puder agregar valor, portanto, o valor do *Big Data* está na análise precisa dos dados, nas informações e *insights* fornecidos para as empresas a partir do seu conteúdo.

5. Veracidade
É de suma importância que as informações reunidas sejam verdadeiras, parecendo uma tarefa difícil em tempos de *fake news*. Porém com o uso do *Big Data*, é possível analisar grandes volumes de dados que acabam compensando possíveis informações equivocadas.

6. Volatilidade
Os fluxos de dados são crescentes em relação à velocidade e variedade, mas também possuem picos periódicos que variam de acordo com as tendências, sendo alguns deles difíceis de serem gerenciados como os não-estruturados.

7. Visualização
Os dados precisam ser apresentados de forma acessível e legível.

3.2 KDD

Em um mundo cada vez mais tecnológico e imerso no universo da internet, o aumento do volume de dados gerados diariamente é inevitável, criando uma grande demanda por armazenamento e processamento dos mesmos. Dessa forma, a análise manual é totalmente inviável, e a implementação de métodos computacionais é de grande importância no processo de extração de informações.

KDD (Knowledge Discovery in Databases) – em português, Descoberta de Conhecimento em Bases de Dados – é um procedimento definido em Fayyad, Piatetsky-Shapiro e Smyth (1996) como sendo um processo não trivial de identificação de padrões, a partir de dados que sejam compreensíveis e potencialmente úteis. Uma outra definição apresentada em Han e Kamber (2012) diz que, o processo de KDD é visto como sendo o procedimento de descobrir padrões interessantes e conhecimento de grandes quantidades de dados. Já a descrição vista em Witten, Eibe e Hall (2011) define esse processo como sendo a extração de dados implícitos, anteriormente desconhecidos, e potencialmente úteis para retirar informações dos dados. Nota-se que as definições apresentadas por esses diferentes autores não divergem muito.

O processo de busca de conhecimento em banco de dados é apresentado em Fayyad, Piatetsky-Shapiro e Smyth (1996), em que é dividido nas seguintes etapas:

1. Preparação dos Dados: Consiste em desenvolver um entendimento e integração das informações relevantes para resolver o problema, além de identificar o objetivo do

processo KDD do ponto de vista do usuário final.

2. Definir o Conjunto de Dados: Consiste em selecionar um conjunto de dados ou focar em um subconjunto de variáveis ou amostras de dados, no qual o processo KDD será realizado.

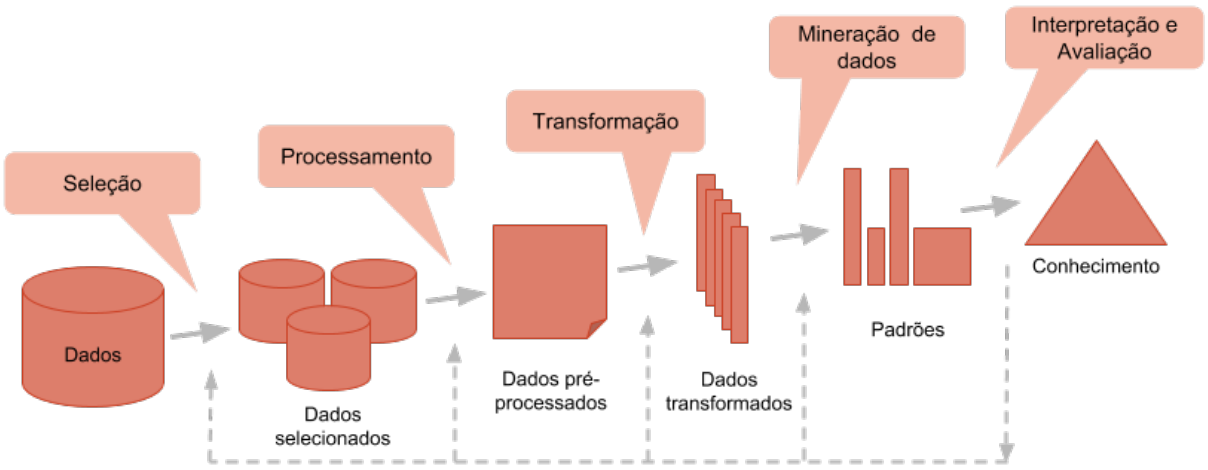
3. Limpeza e Pré-processamento de Dados: Consiste em remover o ruído, coletando as informações necessárias para modelar ou contabilizar o ruído, decidir sobre estratégias para lidar com campos de dados ausentes e contabilizar informações de sequência temporal e alterações conhecidas.

4. Redução dos Dados: Consiste em realizar a redução ou transformação de dimensionalidade dos dados, o número efetivo de variáveis em consideração pode ser reduzida dependendo do método aplicado.

5. Mineração de Dados: Consiste em determinar qual a tarefa de Mineração de Dados será necessária para responder a pergunta de negócio, ou seja, decidir entre regressão, classificação e agrupamento. Outro ponto de atenção é selecionar qual o melhor algoritmo dentro da tarefa escolhida para ser usado na procura de padrões no conjunto dos dados.

6. Interpretação dos Resultados: Essa etapa consiste em visualizar os padrões extraídos e consequentemente realizar a interpretação correta desses resultados, para, assim, chegar no conhecimento extraído do banco de dados, que será usado para auxiliar nas decisões futuras.

Figura 1 – Processo KDD



Fonte: Adaptado de Fayyad, Piatetsky-Shapiro e Smyth (1996)

3.2.1 Mineração de dados

No decorrer do processo KDD tem-se como uma das etapas mais importantes a Mineração de Dados, etapa do processo em que os algoritmos são aplicados ao conjunto de dados. Essa aplicação deve ser bem avaliada e planejada, analisando os tipos de dados disponíveis e se a pergunta inicial que gerou o estudo será respondida.

Han e Kamber (2012) definem os tipos de padrões que podem ser extraídos dos dados, sendo que na etapa de mineração as funcionalidades das técnicas são: caracterização e discriminação, mineração de padrões frequentes e associações e correlações, além das três principais tarefas que são: classificação, regressão e análise de agrupamento.

Neste trabalho iremos utilizar a técnica de Regressão, mais especificamente a Regressão Logística, porém vale ressaltar que para chegar na técnica que mais auxilia na busca de informações deve ser levado em conta as diferenças das técnicas, já que existe um grande leque delas, cada uma com suas particularidades que devem ser consideradas na escolha. Após essa definição pode-se partir para a aplicação com o foco na obtenção das informações ou estimação de interesse.

3.3 Apache

3.3.1 O que é?

O Apache é um servidor de código aberto sendo seu nome oficial *Apache HTTP Server*, ele é gerenciado pela *Apache Software Foundation* e o mesmo está presente em cerca de 46% de todos os sites da internet, de acordo com BOWEN (2009).
A sua primeira versão foi lançada em 1995, sendo ele um dos mais antigos servidores de internet. O Apache possibilita que donos de sites possam mostrar e manter seus conteúdos na internet, por isso o nome "servidor de internet". Quando alguém visita um site, esse visitante entra em um domínio na barra de endereço por um navegador e em seguida, o servidor entrega os arquivos solicitados atuando como se fosse um entregador de encomendas, só que virtual.

3.3.2 Para que serve?

Servidores em geral, sendo eles de arquivos, banco de dados, email e internet usam diferentes categorias de softwares, sendo que cada uma dessas aplicações podem acessar arquivos armazenados em servidores físicos e usá-los para diferentes propósitos. O trabalho de um servidor de internet como o Apache é servir sites na internet agindo como um mediador entre o servidor e as máquinas dos clientes, puxando o conteúdo de um servidor em cada pedido do cliente e realizando essa entrega na internet. Os servidores de internet processam os arquivos escritos em diferentes linguagens de programação, como *PHP*, *Java*,

Python e outras, tendo como maior desafio servir muitos usuários da internet ao mesmo tempo.

3.3.3 Como funciona?

Embora esteja sendo retratado como servidor de internet, vale ressaltar que o Apache não é um servidor físico, já que trata-se de um software executado em um servidor. O trabalho dele é estabelecer uma conexão entre o servidor e os navegadores de sites como *Firefox* e *Google Chrome*, por exemplo, enquanto puxa e entrega arquivos entre eles, sendo essa estrutura chamada de cliente-servidor. Outro ponto a ser ressaltado é que o Apache é um *software* multiplataforma, ou seja, ele funciona tanto em servidor *Unix* como o *Linux* quanto em servidor *Windows*, dessa forma é possível obter suporte para uso da ferramenta utilizando o servidor de preferência.

Para BOWEN (2009), o Apache é altamente personalizável por ter uma estrutura baseada em módulos que permitem que os administradores dos servidores ativem ou desativem novas funcionalidades, sendo esses módulos para segurança, cache, reescrita de URL e autenticação de senhas, além de ser possível fazer suas próprias configurações de servidor por um arquivo chamado *.htaccess*. Toda a comunicação entre servidor e cliente é realizada por HTTP (*Hyper Text Transfer Protocol* - Protocolo de Transferência de Hipertexto), um protocolo de transferência que possibilita que as pessoas que insiram a URL do seu site na Web possam ver os conteúdos e dados que nele existem, sendo o Apache responsável por facilitar e assegurar a comunicação entre os dois lados.

3.3.4 Prós e Contras

3.3.4.1 Prós

1. Código aberto e grátis, mesmo para usos comerciais;

2. *Software* estável e confiável;

3. Atualizado frequentemente e com novidades de segurança;

4. Fácil de configurar e bastante amigável a novos usuários;

5. Suporte a múltiplas plataformas (funciona tanto em servidores *Unix* quanto em servidores *Windows*);

6. Comunidade gigantesca com suporte a dúvidas para qualquer caso de problema.

3.3.4.2 Contras

1. Problemas de desempenho em sites com tráfego muito alto;

2. Muitas opções de configurações podem levar a vulnerabilidades de segurança.

3.4 Hadoop

3.4.1 O que é?

O Hadoop é uma estrutura de *software open-source* (código aberto) para armazenar dados e executar aplicações em *clusters* de *hardwares* comuns. Ele fornece armazenamento massivo para qualquer tipo de dado, grande poder de processamento e a capacidade de lidar quase ilimitadamente com tarefas e trabalhos ocorrendo ao mesmo tempo.

3.4.2 História

Conforme a *World Wide Web* crescia no final dos anos 1990 e início dos anos 2000, mecanismos de busca foram criados para ajudar a localizar informações relevantes em meio a conteúdos textuais. No começo, eram os próprios seres humanos quem devolviam os resultados de buscas, mas, na medida em que a internet cresceu de dezenas para milhares de páginas, a automação se tornou necessária.

Rastreadores *web* foram criados como projetos de pesquisa liderados por universidades e *startups* de mecanismos de busca como o *Yahoo*. Um desses projetos era um mecanismo de busca *open-source* chamado *Nutch* idealizado por Doug Cutting e Mike Cafarella. Eles queriam retornar resultados mais rapidamente ao distribuir dados e cálculos entre computadores distintos para que diferentes tarefas pudessem ser realizadas simultaneamente. Durante esse tempo, outro mecanismo de busca chamado *Google* estava em construção, sendo baseado no mesmo conceito de armazenar e processar dados de forma distribuída e automatizada para que resultados de pesquisa relevantes pudessem ser encontrados rapidamente.

Em 2006, Cutting foi contratado pelo *Yahoo* e levou com ele o projeto *Nutch*, bem como ideias baseadas nos trabalhos iniciais do *Google* de automatizar o armazenamento e o processamento de dados de modo distribuído. O projeto *Nutch* então foi dividido da seguinte forma: o rastreador *web* permaneceu como *Nutch* e a parte de processamento e computação distribuída tornou-se o *Hadoop* (nome do elefantinho de brinquedo do filho do Cutting). Em 2008, o *Yahoo* lançou o *Hadoop* como um projeto *open-source*, sendo que hoje, a estrutura e o ecossistema de tecnologias *Hadoop* são gerenciados e mantidos pela organização sem fins lucrativos *Apache Software Foundation (ASF)*, uma comunidade global de desenvolvedores de *software* e colaboradores, segundo RUSSOM (2013).

3.4.3 Qual a importância do Hadoop?

Figura 2 – Hadoop e suas vantagens

Fonte: Data Science Academy

- **Capacidade de armazenar e processar grandes quantidades de qualquer tipo de dado, e rapidamente**. Com os volumes e tipos de dados disponíveis crescendo constantemente graças às mídias sociais e à Internet das Coisas (IoT), isso é uma consideração importante.

- **Poder computacional**. O modelo computacional distribuído do *Hadoop* processa *Big Data* rapidamente, sendo que quanto maior a quantidade de nós computacionais for utilizado, mais poder de processamento será obtido.

- **Tolerância a falhas**. O processamento de dados e aplicações é protegido contra falhas de *hardware*. Se um nó cai, os trabalhos são automaticamente redirecionados para outros nós para garantir que a computação distribuída não falhe. Múltiplas cópias de todos os dados são armazenadas automaticamente.

- **Flexibilidade**. Ao contrário dos bancos de dados relacionais tradicionais, não é necessário pré-processar os dados antes de armazená-los. É possível armazenar qualquer volume de dados e decidir como usá-los depois, incluindo dados não-estruturados como texto, imagens e vídeos.

- **Custo baixo**. A estrutura *open-source* é gratuita e utiliza *hardwares* comuns para armazenar grandes quantidades de dados.
- **Escalabilidade**. É possível aumentar facilmente o sistema para lidar com mais dados ao adicionar nós, não necessitando de muita administração.

3.4.4 Quais são os desafios em utilizar o Hadoop?

- **A programação de *MapReduce* não é uma boa solução para todos os problemas.** Ela é ótima para pedidos de informação simples e problemas que podem ser divididos entre unidades independentes, mas não é eficiente para tarefas de inteligência analítica iterativas e interativas. O *MapReduce* é focado em arquivos, como os nós não se comunicam, exceto através de misturas e classificações, algoritmos iterativos precisam de diversas fases de map-shuffle e *sort-reduce* para se completarem. Isso cria muitos arquivos entre as fases de *MapReduce* e não é eficiente para computação analítica avançada.
- **Há uma lacuna de talento amplamente notada.** Pode ser difícil encontrar programadores iniciantes que tenham habilidades suficientes em *Java* para serem produtivos com *MapReduce*. Esse é um dos motivos pelos quais os fornecedores estão correndo para incluir tecnologia relacional (SQL) em *Hadoop*, já que é muito mais fácil encontrar programadores com habilidades em SQL do que em *MapReduce*. E a administração do Hadoop parece ser parte arte e parte ciência, requerendo pouco conhecimento técnico em operação de sistemas, *hardware* e configurações centrais de *Hadoop*.
- **Segurança dos dados.** Outro desafio gira em torno dos problemas de segurança de dados fragmentados, embora novas ferramentas e tecnologias estejam surgindo.
- **Gestão e governança de dados completos.** O Hadoop não possui ferramentas completas e fáceis de usar para gerenciamento de dados, governança ou qualidade e padronização de dados.

3.5 Spark

3.5.1 O que é?

Spark é um poderoso mecanismo de processamento de código aberto construído em torno de velocidade, facilidade de utilização, e análises sofisticadas.

3.5.2 História

De acordo com CHAMBERS e ZAHARIA (2018), o Spark foi originalmente desenvolvida na Universidade de Berkeley em 2009, sendo um *framework* 100% *open source* hospedado no *Apache Software Foundation*. Potências da Internet como *Netflix, Yahoo e eBay* o implementaram e começaram a processar coletivamente múltiplos petabytes de dados em *clusters* de mais de 8.000 nós, transformando o *Spark* na maior comunidade *open source* em *big data*, com mais de 1.000 colaboradores de mais de 250 organizações.

O projeto nasceu com o intuito de resolver problemas de performance e processamento paralelo, criando assim um poderoso ambiente de execução em memória nunca visto anteriormente. Rapidamente adotado, ele se transformou em base para diversas aplicações como : *BigData*, Machine Learning, SQL, entre outras.

3.5.3 Arquitetura do Spark

A arquitetura de uma aplicação *Spark* é constituída por três partes principais:

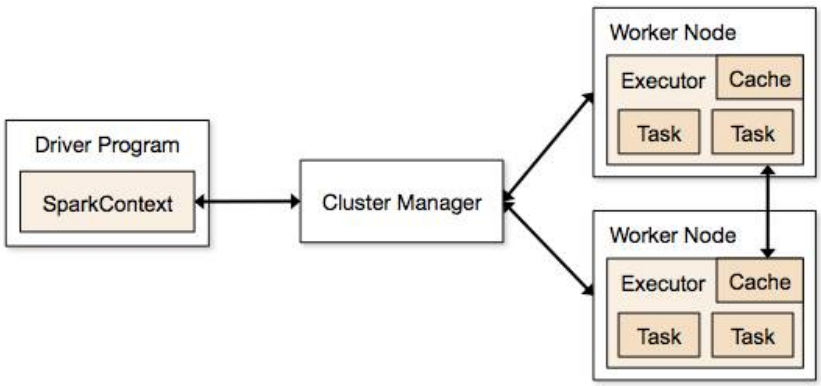
1. O *Driver Program* é a aplicação principal que gerencia a criação e é quem executará o processamento definido pelo programados;

2. O *Cluster Manager* é um componente opcional que só é necessário se o *Spark* for executado de forma distribuída. Ele é responsável por administrar as máquinas que serão utilizadas como *workers*;

3. Os *Workers* são as máquinas que realmente executarão as tarefas que são enviadas pelo *Driver Program*. Se o *Spark* for executado de forma local, a máquina desempenhará tanto o papel de *Driver Program* como de *Worker*.

A Figura 3 mostra a arquitetura do *Spark* e seus principais componentes.

Figura 3 – Arquitetura Spark



Fonte: Data Science Academy

Além da arquitetura, é importante conhecer os principais componentes do modelo de programação do *Spark*. Existem três conceitos fundamentais que serão utilizados em

todas as aplicações desenvolvidas:

1. *Resilient Distributed Datasets (RDD)*: abstraem um conjunto de objetos distribuídos no *cluster*, geralmente executados na memória principal. Estes podem estar armazenados em sistemas de arquivo tradicional, no HDFS (*Hadoop Distributed File System*) e em alguns Banco de Dados *NoSQL*, como *Cassandra* e *HBase*. Ele é o objeto principal do modelo de programação do *Spark*, pois são nesses objetos que serão executados os processamentos dos dados.

2. Operações: representam transformações (como agrupamentos, filtros e mapeamentos entre os dados) ou ações (como contagens e persistências) que são realizados em um RDD. Um programa Spark normalmente é definido como uma sequência de transformações ou ações que são realizadas em um conjunto de dados.

3. *Spark Context*: o contexto é o objeto que conecta o *Spark* ao programa que está sendo desenvolvido. Ele pode ser acessado como uma variável em um programa para utilizar os seus recursos.

3.5.4 Quais são os benefícios do Spark?

1. Velocidade
O *Spark* pode ser 100x mais rápido do que o *Hadoop* para o processamento de dados em grande escala, explorando em computação memória e outras otimizações. Ele também é rápido quando os dados são armazenados no disco e atualmente detém o recorde mundial de grande escala de classificação no disco. A distribuição dos dados em memória e seu processamento paralelo tornam o *framework* muito performático para qualquer tipo de processamento.

2. Fácil de Usar
O Projeto tem APIs (*Application Programming Interface* - Interface de Programação de Aplicativos - um conjunto de definições e protocolos para criar e integrar softwares de aplicações), fáceis de usar para operar em grandes conjuntos de dados. Isso inclui uma coleção de mais de 100 operadores para transformar *APIs* e estrutura de dados familiar para manipulação de dados semi-estruturados. A possibilidade de integrá-lo com diversas bibliotecas e linguaguens facilita muito o dia a dia do desenvolvedor que pode utilizar linguagens como : *Python*, *Scala*, *Java* ou R.

A opção de usar o SQL também torna mais fácil a programação das consultas e extrações de dados, visto que o SQL é a linguagem de consulta de dados mais comum em tecnologia, o Spark SQL abstrai a complexidade do *big data* transformando a consulta de dados de volumes gigantescos em algo muito simples.

3. Ampla Biblioteca
O *Spark* conta com bibliotecas de alto nível para suporte de consultas SQL, aprendizado de máquina e processamento gráfico.

3.6 Hadoop vs. Spark

3.6.1 Comparação

Quando o assunto é comparação entre *Hadoop* e *Spark*, é importante esclarecer que o *Spark* é na verdade um aprimoramento do *Hadoop* para *MapReduce* (modelo de programação desenhado para processar grandes volumes de dados em paralelo, dividindo o trabalho em um conjunto de tarefas independentes). A principal diferença entre os dois é que o *Spark* processa e retém os dados na memória para as etapas subsequentes, enquanto o *Hadoop MapReduce* processa os dados no disco. Como resultado, para cargas de trabalho menores, as velocidades de processamento de dados do *Spark* são até 100 vezes mais rápidas que o *Hadoop*, segundo CHAMBERS e ZAHARIA (2018).

Além disso, ao contrário do processo de execução de dois estágios no *Hadoop MapReduce*, o *Spark* cria um DAG (*Directed Acyclic Graph - Grafos acíclicos dirigidos*) para agendar tarefas e a orquestração de nós no cluster *Hadoop*. Esse processo de rastreamento de tarefas permite a tolerância a falhas, que reaplica as operações gravadas aos dados de um estado anterior.

As principais diferenças entre *Hadoop* e *Spark* são dadas por:

1. Desempenho: O *Spark* é mais rápido por usar memória RAM (*Random Access Memory* - Memória de Acesso Aleatório), em vez de ler e gravar dados intermediários em discos, já o *Hadoop* armazena dados em várias fontes e os processa em lotes via *MapReduce*.

2. Custo: Enquanto o *Hadoop* é executado a um custo baixo por utilizar qualquer tipo de armazenamento em disco para processamento de dados, o *Spark* é executado a um custo mais alto por depender de cálculos na memória para processamento de dados em tempo real, o que exige o uso de grandes quantidades de RAM para ativar nós.

3. Processamento: Embora ambas as plataformas processem dados em um ambiente distribuído, o *Hadoop* é ideal para processamento em lote e processamento linear de dados. Já o *Spark* é ideal para processamento em tempo real e processamento de fluxos de dados não estruturados ao vivo.

4. Escalabilidade: Quando o volume de dados cresce rapidamente, o *Hadoop* consegue acomodar a demanda por meio do HDFS (*Hadoop Distributed File System*), porém, o *Spark* conta com o HDFS tolerante a falhas para grandes volumes de dados.

5. Segurança: O *Spark* aprimora a segurança com autenticação por meio de

segredo compartilhado ou registro de eventos, enquanto o *Hadoop* usa vários métodos de autenticação e controle de acesso. Embora, em geral, o *Hadoop* seja mais seguro, o *Spark* pode se integrar ao *Hadoop* para alcançar um nível de segurança mais alto.

6. Aprendizado de máquina (ML): O *Spark* é a plataforma superior nesta categoria por incluir a MLlib, biblioteca que executa cálculos iterativos de *Machine Learning* (Aprendizado de Máquina) na memória. Ele também inclui ferramentas que realizam regressão, classificação, persistência, construção de pipeline, avaliação, etc.

3.6.2 Alguns enganos sobre *Hadoop* e *Spark*

3.6.2.1 Hadoop

O *Hadoop* é barato: Embora seja de código aberto e fácil de configurar, manter o servidor funcionando pode ser caro. Ao usar recursos como computação em memória e armazenamento em rede, o gerenciamento de *big data* pode custar até US$ 5.000.

***Hadoop* é um banco de dados:** Mesmo que o *Hadoop* seja usado para armazenar, gerenciar e analisar dados distribuídos, não há consultas envolvidas ao extrair dados. Isso o torna um *data warehouse* (armazém de dados, ou apenas um ambiente informacional) em vez de um banco de dados.

O *Hadoop* é difícil de configurar: Embora o gerenciamento do *Hadoop* seja difícil nos níveis mais altos, existem muitas interfaces gráficas de usuário (GUIs) que simplificam a programação do *MapReduce*.

3.6.2.2 Spark

O *Spark* é uma tecnologia em memória: Embora o *Spark* utilize efetivamente o algoritmo LRU (*Least Recently Used* - Menos Utilizado Recentemente) que escolhe preferencialmente páginas antigas e menos usadas para remoção, ele não é, em si, uma tecnologia baseada em memória.

O *Spark* sempre funciona 100 vezes mais rápido que o *Hadoop*: Por mais que o *Spark* possa ter um desempenho até 100 vezes mais rápido que o *Hadoop* para pequenas cargas de trabalho, de acordo com o *Apache*, ele normalmente só funciona até 3 vezes mais rápido para grandes cargas de trabalho.

O *Spark* apresenta novas tecnologias no processamento de dados: Embora o *Spark* utilize efetivamente o algoritmo LRU e o processamento de dados em *pipeline* (técnica de *hardware* que permite que a CPU realize a busca de uma ou mais instruções além da próxima a ser executada), esses recursos existiam anteriormente em bancos de dados de processamento paralelo em massa (MPP). No entanto, o que diferencia o *Spark* do MPP é sua orientação de código aberto.

3.6.3 Casos de uso do *Hadoop* e *Spark*

Após as análises comparativas, prós e contras sobre o *Hadoop* e o *Spark*, é possível ter uma ideia da usabilidade geral de cada ferramenta.

O *Hadoop* é mais eficaz para os seguintes cenários:

- Ao processar conjuntos de *big data* em ambientes onde o tamanho dos dados excede a memória disponível;
- No processamento em lote com tarefas que exploram operações de leitura e gravação de disco;
- Na criação de infraestrutura de análise de dados com um orçamento limitado;
- Na conclusão de trabalhos que não são sensíveis ao tempo;
- Ao realizar análise de dados históricos e de arquivo.

O *Spark* é mais eficaz para os seguintes cenários:

- Ao lidar com cadeias de operações paralelas usando algoritmos iterativos;
- No alcance de resultados rápidos com cálculos na memória;
- Ao analizar a análise de dados de fluxo em tempo real;
- No processamento paralelo ao gráfico para modelar dados;
- Na utilização de todos os aplicativos de Aprendizado de Máquina.

3.7 Hive

3.7.1 O que é?

O *Hive* é um *framework* (estrutura) para soluções de *Data Warehousing* (repositório central de informações que podem ser analisadas para tomar decisões) executado no ambiente *Hadoop*. A escolha para utilização do Hadoop em seu desenvolvimento foi incentivada pelo baixo custo, escalabilidade e para evitar a dependência sobre custos de licenças e manutenção anual, que são comuns nos bancos de dados do mercado.

3.7.2 História

Construído inicialmente pelo time de desenvolvimento do *Facebook* em 2007 mas atualmente fazendo parte da*Apache*, o *Hive* nasceu a partir da necessidade de gerenciar, analisar e aprender sobre o comportamento dos usuários a partir dos imensos volumes de dados gerados a cada dia. Ele foi criado com uma interface SQL, visando aproveitar o conhecimento nessa linguagem de progração que os analistas e desenvolvedores do *Facebook* estavam familiarizados, já que na época eles não eram tão proficientes na linguagem *Java* para utilizar o *Hadoop MapReduce*.

3.7.3 Para que serve?

A finalidade principal do *Hive* é analisar dados e ser capaz de se integrar com soluções de *Business Intelligence* (Inteligência de negócios), podendo fazer parte do processo de coleta, organização, análise, compartilhamento e monitoramento dos dados.

3.7.4 Como funciona?

O *Hive* utiliza uma linguagem chamada HiveQL (*Hive Query Language*), que transforma as sentenças escritas na linguagem SQL em *Jobs* de *MapReduce* que são executados no cluster *Hadoop*. Vale ressaltar que mesmo sendo muito parecida a linguagem SQL tradicional com a existente no Hive, há algumas diferenças de comandos.

Por utilizar o sistema de *Mapreduce* do *Hadoop*, o *Hive* trabalha com o particionamento de tabelas, que permite maior nível de performance das consultas executadas. Essas partições são armazenadas em diretórios diferentes, buscando otimizar o acesso na leitura do disco. O *Hive* não foi desenhado para executar consultas em tempo real, por isso, tratando-se de consultas em linguagem SQL, ele não é um dos mais rápidos para uma amostra pequena, porém isso muda quando trata-se de um grande volume de dados, onde ele costuma ter uma melhor performance.

3.8 Impala

3.8.1 O que é?

O *Impala* é um *framework* que assim como o *Hive* fornece consultas SQL, além de permitir a exploração interativa e o ajuste de consultas analíticas em vez de longos trabalhos em lote tradicionalmente associados a tecnologias *SQL-on-Hadoop*. O alto nível de integração com o *Hive* e a compatibilidade com a sintaxe *HiveQL* permite a escolha de utilização da ferramenta Impala ou Hive para criar tabelas, emitir consultas ou carregar dados, segundo RUSSELL (2014).

3.8.2 Principais vantagens

- O *Impala* fornece acesso a dados armazenados em *Hadoop* sem exigir as habilidades necessárias para trabalhos em *Java*;
- Pode acessar dados diretamente do sistema de arquivos *Hadoop*;
- Fornece interface SQL para acessar dados no sistema de banco de dados;
- Retorna resultados normalmente em segundos ou alguns minutos, em vez dos muitos minutos ou horas que geralmente são necessários para a conclusão das consultas do *Hive*;
- O *Impala* é pioneiro no uso do formato de arquivo *Parquet*, um *layout* de armazenamento otimizado para consultas em larga escala, típicas de *data warehouse*.

3.9 Hive vs. Impala

A principais diferenças entre as duas plataformas são:

- A velocidade de processamento de consultas no *Hive* é lenta, mas o *Impala* é de 6 a 69 vezes mais rápido que o *Hive*.
- No *Hive*, a latência (tempo de resposta) é alta, mas no *Impala*, a latência é baixa.

3.10 Python

Desde que foi criada pelo pesquisador holandês Guido Van Rossum, no final dos anos 1980, a linguagem de programação de código aberto conhecida como *Python* se tornou famosa por sua agilidade e interatividade.

3.10.1 Como funciona?

O *Python* é uma linguagem interpretada, ou seja, ela traduz o código analisado e o executa tendo a grande vantagem de proporcionar soluções para os mais variados tipos de problemas, além de ser uma ferramenta multiplataforma independente, que pode ser aplicada em *Windows*, Macintosh e *Linux*.

3.10.2 Aplicações

Por mais simples que pareça ser utilizar a linguagem *Python*, ela serve para produtos complexos como: criação de jogos, análises estatísticas (*data science*, *machine learning*,

etc...), desenvolvimento *web*, desenvolvimento de aplicativos, automações e etc. Com o poder de ser usado em diferentes tipos de projetos, o *Python* abre um grande leque de oportunidades.

3.11 R

3.11.1 O que é?

O R é uma tecnologia estatística criada por Ross Ihaka e Robert Gentleman para suprir uma deficiência de ferramentas gratuitas e simples para análises da área. Justamente por possibilitar análises estatísticas, cálculos e manipulações gráficas, a linguagem R já é usada em diversos campos do conhecimento, seja para fins acadêmicos oupara o mercado. O objetivo dos fundadores era desenvolver algo que fosse baseado em *scripts*, ou rápidos pedaços de códigos, e que não apresentasse grandes dificuldades para quem não é programador.

Alguns benefícios da linguagem são:

- **Suporte ao Big Data**

  Concorrendo diretamente com outra linguagem muito famosa, a *Python*, o R é uma ótima ferramenta para o *Big Data*, pois possibilita o estudo desses dados, em busca de valor, com a identificação de correlações, padrões e tendências. Além disso, é possível gerar rapidamente inúmeras maneiras de visualizar as informações em formas de gráficos e relatórios estruturados, o que facilita a compreensão e o compartilhamento dessas informações.

- **Análises Robustas**

  O R é fundamental para analisar os dados e manipulá-los antes do processo de alimentação de um algoritmo mais robusto computacionalmente como o *Machine Learning* e o *Random Forest*, de modo a otimizar os resultados. Sendo esses modelos capazes de predizer o futuro e identificar riscos ou oportunidades, no R existe uma série de bons recursos para ajudar a aumentar a precisão dessas predições, o que contribui com os resultados e com a qualidade das escolhas das companhias.

- **É gratuito**

  Outra vantagem que deve ser considerada: o R é uma solução gratuita e de código aberto. Ou seja, qualquer programador pode ajudar a melhorar a ferramenta, o que a torna ainda mais robusta, madura e segura. Além disso, não há custos para começar com a tecnologia nem para obter resultados rápidos para uma empresa. Assim, o retorno é extremamente benéfico, com um índice de lucro positivo.

- **Tem uma comunidade**

  Com cerca de 2 milhões de usuários, a comunidade do R é outro ponto a favor do poder desse padrão de programação. A comunidade de uma linguagem é muito importante para quem está começando a aprender, pois consiste em um grupo de pessoas que ajudam a solucionar os principais problemas, compartilhando dicas e eliminando possíveis dúvidas. Da mesma forma, é muito útil para quem já é experiente também, com a solução de questões mais complexas e específicas. Quanto maior a comunidade, mais ajuda há disponível e, portanto, melhor é o aprendizado.

Concluindo, o R é uma tecnologia poderosa e extremamente versátil, já que promove uma integração com outras ferramentas como o *Python* e também possui uma extensão de pacotes diversos para todo o tipo de análise. Seu uso para análise de dados é bem conhecido, por ser uma das principais soluções desse mercado.

3.12 Python vs. R

O contexto de uma rivalidade entre *Python* e R não é de hoje, sempre é abordado o questionamento de qual linguagem é melhor. A verdade é que as duas linguagens são ótimas e resolvem o que é necessário no quesito estatística, tendo apenas algumas diferenças devido ao estilo de criação das ferramentas. As principais diferenças entre as linguagens de programação *Python* e *R* são:

3.12.1 Objetivos da análise de dados

A principal distinção entre as duas linguagens está em sua abordagem, já que ambas as linguagens de programação de código aberto são suportadas por grandes comunidades, estendendo continuamente suas bibliotecas e ferramentas. Mas enquanto R é usado principalmente para análise estatística o *Python* fornece uma abordagem mais geral.

Python

O *Python* é uma linguagem multifuncional, muito parecida com C++ e Java, com uma sintaxe legível e fácil de aprender. Os programadores usam o *Python* para aprofundar a análise de dados ou usar o aprendizado de máquina em ambientes de produção escaláveis.

R

O R, por outro lado, é construído por estatísticos e se apoia fortemente em modelos estatísticos e análises especializadas. Os cientistas de dados usam R para análises estatísticas profundas, apoiadas por apenas algumas linhas de código e belas visualizações de dados.

3.12.2 Coleta de dados

Python

o Python oferece suporte a todos os tipos de formatos de dados, desde arquivos com valores separados por vírgula (CSV) até JSON originados da *web*. Também é possível importar tabelas SQL diretamente no código *Python*. Para desenvolvimento da Web, a biblioteca de solicitações do *Python* permite obter-se facilmente dados da *web* para criar conjuntos de dados.

R

Em contraste, o R foi projetado para analistas de dados importarem dados do Excel, CSV e arquivos de texto. Arquivos criados no *Minitab* ou no formato SPSS também podem ser transformados em dataframes R. Embora o *Python* seja mais versátil para extrair dados da *Web*, os pacotes modernos do R ajudam na criação de boas análises.

3.12.3 Exploração de dados

Python

Em *Python* é possível explorar dados com o Pandas, a biblioteca de análise de dados para *Python*, sendo possível filtrar, classificar e exibir dados em questão de segundos.

R

O R, por outro lado, é otimizado para análise estatística de grandes conjuntos de dados e oferece várias opções diferentes para explorar dados. Com o R é possível criar distribuições de probabilidade, aplicar diferentes testes estatísticos e usar técnicas padrão de aprendizado de máquina e mineração de dados.

3.12.4 Modelagem de dados

Python

O *Python* possui bibliotecas padrão para modelagem de dados, incluindo Numpy para análise de modelagem numérica, SciPy para computação e cálculos científicos e scikit-learn para algoritmos de aprendizado de máquina.

R

Para análise de modelagem específica em R, às vezes será necessário confiar em pacotes fora da funcionalidade principal do R. Mas o conjunto específico de pacotes conhecido como Tidyverse facilita a importação, manipulação, visualização e relatórios de dados.

3.12.5 Visualização de dados

Python

Embora a visualização não seja um ponto forte em *Python*, é possível utilizar a biblioteca Matplotlib para gerar gráficos e tabelas básicas. Além disso, a biblioteca Seaborn permite desenhar gráficos estatísticos mais atraentes e informativos.

R

O R foi construído para demonstrar os resultados da análise estatística, com o módulo gráfico básico que permite criar facilmente gráficos e gráficos básicos. Também é possível usar ggplot2 para gráficos mais avançados, como gráficos de dispersão complexos com linhas de regressão.

4 Regressão Logística

Considerada uma das técnicas estatísticas mais utlizadas no mercado de trabalho e com aplicações em diversas áreas, a Regressão Logística é uma técnica de Análise Multivariada, utilizada para descrever a relação entre uma variável resposta, sendo ela binária ou multinomial, dentro de um conjunto de dados com variáveis explicativas.

Assim como em qualquer análise utilizando outras técnicas de modelagem, no presente trabalho a meta é obter um modelo mais econômico e que descreva de forma razoável a relação entre a variável de interesse (se o cliente inadimplente possui um perfil digital ou não) e o conjunto de dados presente utilizando a Regressão Logística, que será retratada neste capítulo desde seu nascimento até sua interpretação.

4.1 História

A função logística foi inventada no século XIX para a modelagem do crescimento populacional e de reações químicas. (CRAMER, 2002, p.3)

Considerando uma quantidade W(t), onde t é o tempo, a variação é dada pela primeira derivada:

$$W'(t) = \frac{dW(t)}{dt} \tag{4.1}$$

Supondo que W'(t) seja proporcional a W(t), tem-se:

$$W'(t) = \beta \frac{dW(t)}{dt} \tag{4.2}$$

Onde β é uma taxa constante de crescimento, que leva a:

$$(t) = Ae^{\beta t} \tag{4.3}$$

Segundo Cramer (2002), esse modelo é adequado para o crescimento populacional de países jovens. Alphonse Quetelet (1795-1874) sabia que esse ritmo implica valores impossíveis. Então, pediu a seu aluno Pierre-François Verhulst (1804-1849) para ajudá-lo, que acrescentou uma suposição:

$$W'(t) = \beta W(t)(\Omega - W(t)) \tag{4.4}$$

Em que, Ω é o limite superior de W. Escrevendo W(t) como proporção, que é proporcional a W(t) e (Ω-W(t)):

$$P(t) = \frac{W(t)}{\Omega} \tag{4.5}$$

$$P(t) = \beta P(t)(1 - P(t)) \tag{4.6}$$

A solução para essa equação diferencial é:

$$P(t) = \frac{exp(\alpha + \beta t)}{1 + exp(\alpha + \beta t)} \tag{4.7}$$

Verhulst chamou essa função de logística. Portanto, a população W(t) segue:

$$P(t) = \Omega \frac{exp(\alpha + \beta t)}{1 + exp(\alpha + \beta t)} \tag{4.8}$$

Verhulst publicou seu trabalho entre 1838 e 1847. Ele modelou o crescimento das populações de países como França, Bélgica, Rússia e o condado de Essex. A curva foi bem ajustada e os parâmetros α, β, Ω foram estimados determinando três pontos em que a curva deveria passar. (CRAMER, 2002)

A função logística foi descoberta novamente, de maneira independente, por Pearl e Reed em 1920, ao estudarem o crescimento da população dos Estados Unidos. Na época, Raymond Pearl era diretor do departamento de Biometria e Estatísticas Vitais da universidade John Hopkins. Pearl era biólogo e familiarizou-se com a Estatística após passar um ano com Karl Pearson (1857-1936). Lowell J. Reed era matemático, seguiu uma carreira na área de Bioestatística e em 1920 era assistente de Pearl. Posteriormente, tornou-se presidente da universidade John Hopkins.

Uma publicação importante, ocorreu em 1925 por Udny Yule (1871-1951), na Sociedade Estatística Real. Yule cita Reed e Verhulst ao nomear a função como logística. (CRAMER, 2002)

4.2 Modelo da Regressão Logística

Seja Y uma variável aleatória, onde

$$Y = \begin{cases} 1, & \textit{evento de interesse} \\ 0, & \textit{caso contrário} \end{cases}$$

cada Y terá distribuição de Bernoulli, com função de probabilidade dada por:

$$P(y|\theta) = \theta^y (1 - \theta)^{1-y}, \quad y = 0, 1, 2, ..., n \tag{4.9}$$

sendo:

y = Evento Ocorrido

θ= Probabilidade de interesse para a ocorrência do evento

Como há uma sequência desses eventos, a soma dos valores observados terão uma distribuição Binomial com n (observações) e θ (probabilidade de sucesso). Cuja função de probabilidade Binomial será

$$P(y|n,\theta) = \begin{pmatrix} n \\ y \end{pmatrix} \theta^y (1-\theta)^{1-y} \tag{4.10}$$

Onde:

- Em cada tentativa é possível apenas dois resultados, sendo sucesso ou fracasso;
- Cada tentativa é independente das outras;
- A probabilidade de sucesso θ em cada tentativa permanece constante e independente das demais;
- A variável de interesse é o número de sucessos y nas n tentativas.

A transformação logística será o logaritmo da razão de probabilidades. A função de ligação do modelo de regressão logística univariado é dada por

$$g(x) = ln\left(\frac{\pi(x)}{(1-\pi(x))} \right) = \beta_0 + \beta_1 x, \tag{4.11}$$

Já para o caso multivariado será:

$$g(\mathbf{x}) = ln\left(\frac{\pi(\mathbf{X})}{(1-\pi(\mathbf{X}))} \right) = \beta_0 + \beta_1 x_1, + ... + B_n x_n \tag{4.12}$$

A probabilidade do evento Y ocorrer dado x, é dada por

$$\pi(x) = E(Y|x) = \frac{e^{g(x)}}{1+e^{g(x)}} = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \tag{4.13}$$

e a probabilidade do evento Y ocorrer dado o vetor $\mathbf{x}$, será

$$\pi(\mathbf{x}) = E(Y|\mathbf{x}) = \frac{e^{g(\mathbf{X})}}{1+e^{g(\mathbf{X})}} = \frac{e^{\beta_0+\beta_1 x_1+...+\beta_n x_n}}{1+e^{\beta_0+\beta_1 x_1+...+\beta_n x_n}} \tag{4.14}$$

O princípio de máxima verossimilhança afirma que se usa como estimativa de *beta* o valor que maximiza a expressão

$$l(\beta) = \prod_{i=1}^{n} \pi(x_i)^{y_i} [1-\pi(x_i)]^{1-y_i} \tag{4.15}$$

em que n representa o número de indivíduos na amostra.
No entanto, é mais fácil trabalhar matematicamente com o *log* da equação, definida como

$$L(\beta) = ln[l(\beta)] = \sum_{i=1}^{n} \{y_i ln[\pi(x_i)] + (1-y_i) ln[1-\pi(x_i)]\} \tag{4.16}$$

Para que seja possível encontrar o vetor de $\boldsymbol{\beta}$, maximiza-se $L(\boldsymbol{\beta})$ e diferencia-se $L(\boldsymbol{\beta})$ em relação a β_0 e β_1, definindo assim as expressões resultantes iguais a zero. As equações são denotadas da seguinte forma:

$$\sum[y_i - \pi(x_i)] = 0 \tag{4.17}$$

e

$$\sum x_i[y_i - \pi(x_i)] = 0 \tag{4.18}$$

Como essas equações são não-lineares, para solucioná-las é necessário utilizar métodos numéricos iterativos, como Newton-Raphson (Gourieroux e Monfort, 1995). Expandindo a função U($\boldsymbol{\beta}$) em torno do ponto inicial $\boldsymbol{\beta}^{(0)}$, teremos

$$U(\boldsymbol{\beta}) \approx U(\boldsymbol{\beta}^{(0)}) + U'(\boldsymbol{\beta}^{(0)})(\boldsymbol{\beta} - \boldsymbol{\beta}^{(0)}) \tag{4.19}$$

Sendo que U($\boldsymbol{\beta}$) são as derivadas de primeira ordem e U'($\boldsymbol{\beta}$) são as derivadas de segunda ordem do logaritmo da função de verossimilhança. A repetição do processo (4.19) resulta no seguinte processo iterativo

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + [-U'(\boldsymbol{\beta}^{(k)})]^{-1}U'(\boldsymbol{\beta}^{(k)}), \quad k = 0, 1, ... \tag{4.20}$$

como a matriz -U'($\boldsymbol{\beta}$) pode não ser positiva definida, não admitindo sua matriz inversa, então, substitui-se pela matriz de informação de Fisher.

$$\boldsymbol{\beta}^{(k+1)} = \boldsymbol{\beta}^{(k)} + [-I(\boldsymbol{\beta}^{(k)})^{-1}]U'(\boldsymbol{\beta}^{(k)}), \quad k = 0, 1, ... \tag{4.21}$$

Dessa forma, a matriz de informação de Fisher com uma variável para o modelo logístico pode ser escrita como:

$$I(\widehat{\boldsymbol{\beta}}) = -\begin{bmatrix} \frac{\partial^2}{\partial\beta_0^2} ln\, Ł(\beta_0, \beta_1) & \frac{\partial^2}{\partial\beta_0\beta_1} ln\, Ł(\beta_0, \beta_1) \\ \\ \frac{\partial^2}{\partial\beta_0\beta_1} ln\, Ł(\beta_0, \beta_1) & \frac{\partial^2}{\partial\beta_1^2} lnŁ(\beta_0, \beta_1) \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1}^{k} n_i \frac{e^{\beta_0+\beta_1 x_i}}{(1+e^{\beta_0+\beta_1 x_i})^2} & \sum_{i=1}^{k} n_i x_i \frac{e^{\beta_0+\beta_1 x_i}}{(1+e^{\beta_0+\beta_1 x_i})^2} \\ \\ \sum_{i=1}^{k} n_i x_i \frac{e^{\beta_0+\beta_1 x_i}}{(1+e^{\beta_0+\beta_1 x_i})^2} & \sum_{i=1}^{k} n_i x_i^2 \frac{e^{\beta_0+\beta_1 x_i}}{(1+e^{\beta_0+\beta_1 x_i})^2} \end{bmatrix} \tag{4.22}$$

Com as estimativas dos parâmetros do modelo, é possível calcular as probabilidades estimadas a partir da expressão (4.14).

4.3 Teste da Razão de Verossimilhanças

Para a utilização da regressão logística, verificar se as variáveis da base de dados são significativas é uma etapa essencial. Essa verificação é baseada no log da verossimilhança. Para isso, a hipótese a ser testada é a seguinte:

$$\begin{cases} H_0 : \beta_j = 0, \quad j = 1, 2, ..., k \\ H_1 : \beta_j \neq 0 \end{cases}$$

A estatística do teste é dada por

$$G = -2ln \left[\frac{(verossimillhança\ sem\ a\ variável)}{(verossimilhança\ com\ a\ variável)} \right]$$

ou

$$G = -2ln(L_s) + 2ln(L_c),$$

em que L_s é a verossimilhança do modelo sem a covariável e L_c é a verossimilhança do modelo com covariável.

4.4 Teste Z-Wald

O teste de Wald também é usado para determinação da significância dos coeficientes logísticos, sendo outra ferramenta para verificar se as variáveis da base de dados são significativas.
O cálculo da estatística do teste Wald, é dado pela divisão do coeficiente estimado $\widehat{\beta}_j$ pelo seu desvio padrão (DP), esse resultado terá uma distribuição aproximadamente normal padrão

$$Wald = \frac{\widehat{\beta}_j}{\widehat{DP(\beta}_j)} \sim N(0, 1),$$

Ao elevar ao quadrado essa estatística Z, tem-se uma aproximação a qui-quadrado com 1 grau de liberdade

$$Wald = \left(\frac{\widehat{\beta}_j}{\widehat{DP(\beta}_j)} \right) \sim \chi^2_1,$$

e as hipóteses são

$$\begin{cases} H_0 : \beta_j = 0 \\ H_1 : \beta_j \neq 0 \end{cases}$$

dessa forma, quando $\chi^2_{(Calculado)} > \chi^2_{(Tabelado)}$, rejeita-se H_0 ao nível de confiança α.

4.5 Intervalo de Confiança para os Parâmetros

Os intervalos de confiança são baseados nos testes de Wald. O intervalo de confiança de 100(1-α) para o parâmetro $\beta_j (j = 1, 2, ..., k)$ é:

$$IC(\beta_j, 1-\alpha) = [\widehat{\beta}_j - z_{1-\alpha/2} DP(\widehat{\beta}_j);\ \widehat{\beta}_j + z_{1-\alpha/2} DP(\widehat{\beta}_j)],$$

em que $z_{1-\alpha/2}$ é o ponto da normal padrão correspondente a 100(1-α/2).

4.6 Intervalo de Confiança para os Valores Ajustados

Com o estimador da função de ligação e seu intervalo de confiança, pode-se estimar os valores ajustados e calcular os respectivos intervalos:

$$IC(\pi, 1-\alpha) = \left[\frac{e^{\hat{g}(x) - z_{1-\alpha/2} DP(\hat{g}(x))}}{1+e^{\hat{g}(x) - z_{1-\alpha/2} DP(\hat{g}(x))}};\ \ \frac{e^{\hat{g}(x) + z_{1-\alpha/2} DP(\hat{g}(x))}}{1+e^{\hat{g}(x) + z_{1-\alpha/2} DP(\hat{g}(x))}} \right]$$

4.7 Intervalo de Confiança para a Razão das Chances (*Odds Ratio*)

Considere os limites do intervalo de confiança para β_j:

$$L.I. = \widehat{\beta}_j - z_{1-\alpha/2} DP(\widehat{\beta}_j) \ \text{ e } \ L.S. = \widehat{\beta}_j + z_{1-\alpha/2} DP(\widehat{\beta}_j),$$

o intervalo de confiança para a razão das chances será:

$$IC(RC, 1-\alpha) = [e^{\beta_I}\ ;\ e^{\beta_S}]$$

4.8 Resíduo de Pearson

Na regressão logística, uma das formas de medir a diferença entre os valores observados e os valores preditos é calcular os valores ajustados para cada combinação de níveis diferentes das variáveis explicativas, denominada de padrão de covariável.

Assim, o conjunto de variáveis explicativas do modelo será: $\mathbf{X} = (X_1, X_2, ..., X_p)$, sendo n o número de valores que x assumirá. O número de indivíduos na amostra com valores iguais $\mathbf{x} = \mathbf{x}_j$ é denotado por: m_j, e o número de indivíduos y = 1 será denotado por y_j em m_j, tendo a probabilidade ajustada dos indivíduos em j como π_j, assim, obtendo:

$$\hat{y}_j = m_j\hat{\pi}_j,$$

e a medida de Pearson para a diferença entre o observado e predito é:

$$r(y_j, \hat{\pi}_j) = \frac{(y_j - m_j\hat{\pi}_j)}{\sqrt{m_j\hat{\pi}_j(1-\hat{\pi}_j)}}$$

dessa forma, a estatística χ^2 de Pearson será:

$$\chi^2 = \sum_{j=1}^{J} r(y_j, \hat{\pi}_j)^2$$

4.9 Deviance

O *deviance* (D) é a soma dos quadrados dos resíduos no modelo logístico, equivalente a ANOVA, etapa realizada na regressão linear. Ele é definido da seguinte forma:

$$d(y_j, \hat{\pi}_j) = \pm \left[\, 2 \left[\, y_j \, ln \, \left(\, \frac{y_j}{m_j\hat{\pi}_j} \, \right) + (m_j - y_j) \, ln \, \left(\, \frac{(m_j - y_j)}{m_j\hat{\pi}_j(1-\hat{\pi}_j)} \, \right) \, \right] \, \right]^{\frac{1}{2}}$$

Os sinais + ou - é o mesmo sinal que $(y_j - m_j\hat{\pi}_j)$, mj é o número de observações no padrão de covariável j, e $\hat{\pi}_j$ é a probabilidade ajustada dos indivíduos no grupo j e y_j, o número de indivíduos em j com y = 1.
A estatística é dada por:

$$D = \sum_{j=1}^{J} d(y_j, \hat{\pi}_j)^2$$

Sob a hipótese de que o modelo ajustado é adequado, o mesmo possui a distribuição χ^2 com $J - (p + 1)$ graus de liberdade.

4.10 Teste de Hosmer e Lemeshow

O teste de Hosmer-Lemeshow é muito utilizado na regressão logística para verificar o ajuste. O teste verifica as distâncias entre as probabilidades ajustadas e as probabilidades observadas. Estima-se as frequências esperadas dentro de cada grupo, então divide-se a variável dependente.
Para Y = 1, a frequência é observada pela soma das probabilidades estimadas de todos os indivíduos dentro de um grupo. Para Y = 0, a frequência é dada pela soma de 1 menos a probabilidade de todos os indivíduos dentro daquele grupo.
Com as frequências, calcula-se a estatística de Hosmer e Lemeshow, $\hat{C}$, que é dada pela seguinte fórmula

$$\hat{C} = \sum_{k=1}^{g} \frac{(o_k - n_k\bar{\pi}_k)^2}{n_k\pi_k(1-\pi_k)}$$

em que:

- n_k é o número de indivíduos no k-ésimo grupo.
- $\bar{\pi}_k = \sum_{j=1}^{C_k} \frac{m_j\bar{\pi}_j}{n_k}$
- C_k: número total de combinações de níveis.
- $O_k = \sum_{j=1}^{C_k} y_j$: total de respostas dentro do grupo k.
  A estatística do teste segue aproximadamente uma distribuição χ^2 com g-2 graus de liberdade.

4.11 Matriz de confusão

A matriz de confusão fornece a visualização da classificação dos valores preditos versus os verdadeiros valores, além de contribuir na elaboração da curva ROC.
Analisar algumas métricas na matriz de confusão é importante para verificar não só a taxa de acerto geral, como individualmente nos dois grupos.

Quadro 1 - Matriz de Confusão

	Valor Predito	
Valor Verdadeiro	1	0
1	Positivos Verdadeiros (PV)	Negativos Falsos (NF)
0	Positivos Falsos (PF)	Negativos Verdadeiros (NV)

Fonte: Johnson e Wichern (2007)

Acurácia: $\frac{NV+PV}{PV+NF+PF+NV}$ é acerto geral do modelo.

Erro: $\frac{NF+PF}{PV+NF+PF+NV}$ é o erro geral do modelo.

Sensibilidade: $\frac{PV}{PV+NF}$ é a proporção de verdadeiros positivos, avalia a capacidade do modelo classificar um indivíduo como evento ($\hat{Y} = 1$), dado que realmente ele é evento (Y = 1).

Especificidade: $\frac{NV}{NV+PF}$ é a proporção de verdadeiros negativos, avalia a capacidade do modelo classificar um indivíduo como não evento ($\hat{Y} = 0$) dado que ele realmente é não evento (Y = 0).

Verdadeiro Preditivo Positivo: $\frac{PV}{PV+PF}$ é a proporção de verdadeiros positivos em relação a todas as predições positivas.

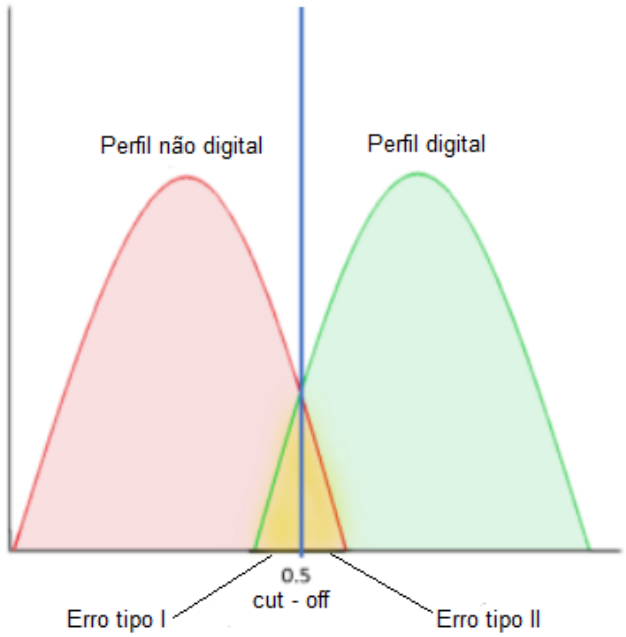
Verdadeiro Preditivo Negativo: $\frac{NV}{NV+NF}$ é a proporção de verdadeiros negativos em relação a todas as predições negativas.

4.12 Cut-off

O valor que se usa para classificar o valor predito em acerto ou erro em relação ao ajuste do modelo é nomeado como *Cut-off* (ponto de corte). Em termos teóricos, quando a probabilidade ajustada é ≥ 0.5, por exemplo, atribui-se 1, e 0 caso contrário.

Mesmo utilizando o ponto de corte que maximiza a taxa de acertos, nunca será possível acertar 100% dos resultados, uma vez que haverá indivíduos de perfil não digital (status 0) com a mesma característica de indivíduos de perfil digital (status 1), e o mesmo para o caso contrário, assim gerando dois tipos de erros:

Figura 4 – Erro tipo I (Negativos Falsos) e Erro tipo II (Positivos Falsos)



Fonte: Elaborado pela autora

Erro tipo 1: Classificar como perfil não digital um indivíduo com perfil digital.

Erro tipo 2: Classificar como perfil digital um indivíduo que não tenha o perfil de cliente digital.

O objetivo de achar um cut-off ideal, é equilibrar os erros dentro dos grupos de perfil digital e não digital, para isso, analisa-se as porcentagens da sensibilidade (Verdadeiro positivo) e especificidade (Verdadeiro negativo), quando elas tiverem resultados semelhantes atribui-se o ponto de corte na probabilidade que foi usada para construir a matriz de confusão.

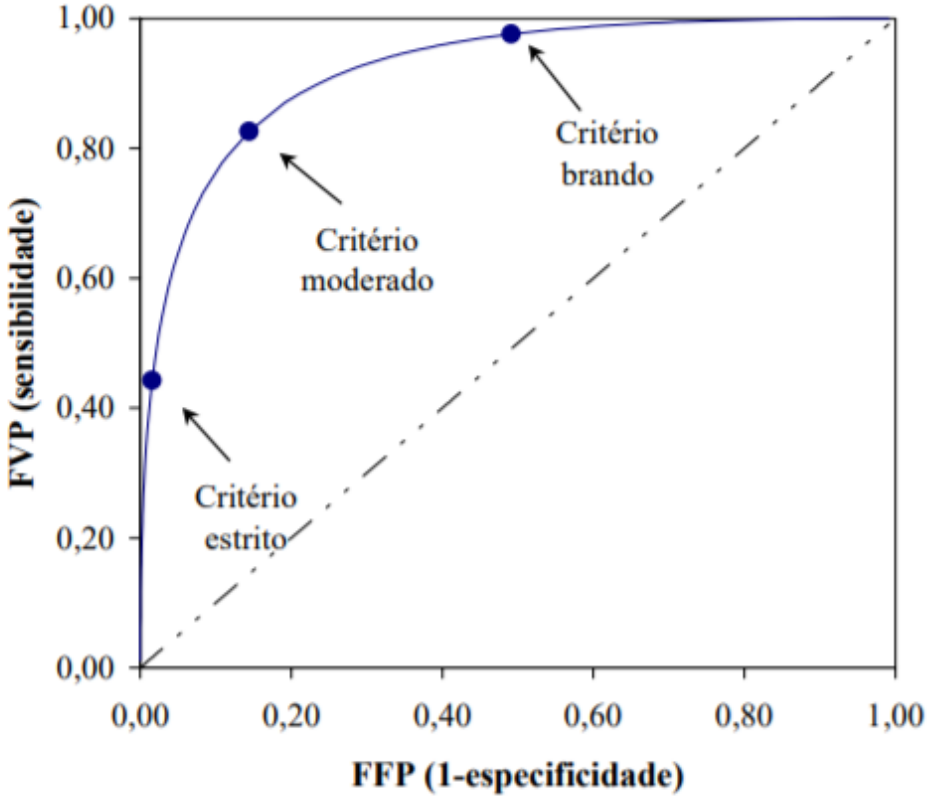
4.13 Curva ROC

Uma das técnicas existentes para facilitar a interpretação da classificação 0 ou 1 conforme as variáveis explicativas, é o método gráfico da curva ROC. Sabe-se que o modelo nunca terá um acerto de 100%, já que sempre haverá situações em que o mesmo previu 1, dado que o verdadeiro valor é 0, e o mesmo para o caso contrário.
O valor da área sob a curva (AROC) fornece a capacidade preditiva do modelo classificar corretamente as observações, sendo os seguintes intervalos a serem considerados:

$$\begin{cases} ROC = 0,5 \text{ Não discrimina} \\ 0,5 \leq \text{ROC} < 0{,}7 \text{ Baixa} \\ 0,7 \leq \text{ROC} < 0{,}8 \text{ Aceitável} \\ 0,8 \leq \text{ROC} < 0{,}9 \text{ Muito bom} \\ ROC \geq \ 0{,}9 \text{ Excelente} \end{cases}$$

Na figura 3 descrita abaixo, pode-se definir, um critério "estrito"como sendo aquele que conduz a uma pequena fração de falsos positivos e também a uma relativamente pequena fração de verdadeiros positivos, isto é, gera um ponto na curva ROC que se situa no canto inferior esquerdo do espaço ROC. Progressivamente critérios menos estritos conduzem a maiores frações de ambos os tipos, isto é, pontos colocados no canto superior direito da curva no espaço ROC. Nomeando os eixos tem-se sensibilidade ou FVP (ordenadas) e 1-especificidade ou FFP (abcissas).

Figura 5 – Curva ROC, para uma dada capacidade de discriminação, com a variação do critério de decisão.



Fonte: Braga (2018)

Figura 6 – Três graus de capacidade de discriminação da curva ROC

Fonte: Adaptado de Markham (2014)

Na figura 4, apresentam-se três graus de discriminação possíveis fornecidos pelas

curvas ROC, sendo que um bom classificador terá um arco ou curva e estará mais longe da linha do classificador aleatório (*Random Classifier*). Para quantificar um bom classificador de um mau usando uma curva ROC, é utilizada a AUC (Área sob a Curva), que a partir do gráfico fica claro que um bom classificador terá uma área sob a curva maior do que um classificador ruim, pois a área sob a curva será maior para o primeiro.

5 Random Forest

5.1 O que é?

Random Forest significa floresta aleatória em português. Sendo que o nome explica muito bem o funcionamento do algoritmo, que irá criar muitas árvores de decisão, de maneira aleatória, formando o que podemos enxergar como uma floresta, onde cada árvore será utilizada na escolha do resultado final.

5.2 Métodos Ensemble

Para entender o algoritmo *Random Forest*, é necessário de antemão conhecer os métodos *ensemble*, dos quais ele faz parte. Estes métodos são construídos da mesma forma que algoritmos mais básicos, como regressão linear eárvore de decisão, mas possuem uma característica principal que os diferenciam, a combinação de diferentes modelos para se obter um único resultado. Essa característica torna esses algoritmos mais robustos e complexos, levando a um maior custo computacional que costuma ser acompanhando de melhores resultados.

Normalmente na criação de um modelo, escolhe-se o algoritmo que apresenta o melhor desempenho para os dados em questão. Testa-se diferentes configurações do algoritmo escolhido, gerando diferentes modelos, mas no fim do processo de *machine learning*, apenas 1 é escolhido. Com um método *ensemble* vários modelos diferentes serão criados a partir de um algoritmo, mas não será escolhido apenas um para utilização final, e sim todos.

Com esta metodologia é possível ter ter um resultado para cada modelo criado. Em problemas de regressão a média dos valores é aplicada para a obtenção do resultado final, já nos problemas de classificação o resultado que mais se repete será o escolhido. Há casos onde o resultado de um modelo será utilizado na criação do próximo, criando uma dependência entre os modelos, e levando a um único resultado final, gerado a partir de vários resultados intermediários.

5.3 Como funciona?

5.3.1 Árvore de Decisão

As Árvores de Decisão, estabelecem regras para tomada de decisão, a partir de um algoritmo que criará uma estrutura similar a um fluxograma, com “nós” onde uma

condição é verificada, e se atendida o fluxo segue por um ramo, caso contrário, por outro, sempre levando ao próximo nó, até a finalização da árvore. Com os dados de treino, o algoritmo busca as melhores condições, e onde inserir cada uma dentro do fluxo.

5.3.2 Seleção de Amostras

Ao utilizar o RandomForest, o primeiro passo executado pelo algoritmo será selecionar aleatoriamente algumas amostras dos dados de treino, sendo utilizado nessa etapa o *bootstrap*, que é um método de reamostragem onde as amostras selecionadas podem ser repetidas na seleção. Com esta primeira seleção de amostras é possível construir a primeira árvore de decisão.

5.4 Seleção das variáveis para cada nó

Na árvore de decisão, é preciso definir o primeiro nó da árvore (nó raiz) para dar início ao modelo, sendo que essa será a primeira condição verificada, dando origem aos dois primeiros ramos. Utilizando o algoritmo de entropia ou o índice Gini, será escolhida a melhor variável para compor o nó raiz, variando de acordo com o método utilizado.

A definição desta variável no *Random Forest* não acontece com base em todas as variáveis disponíveis. O algoritmo irá escolher de maneira aleatória duas ou mais variáveis, e então realizar os cálculos com base nas amostras selecionadas, para definir qual dessas variáveis será utilizada no primeiro nó. Para escolha da variável do próximo nó, novamente serão escolhidas duas (ou mais) variáveis, excluindo as já selecionadas anteriormente, e o processo de escolha se repetirá. Desta forma a árvore será construída até o último nó. A quantidade de variáveis a serem escolhidas pode ser definida na criação do modelo.

É evidente que este não é o melhor método para construção de uma árvore de decisão. O algoritmo pode, sem querer, selecionar as duas piores variáveis na primeira seleção, escolhendo uma variável péssima para o primeiro nó. Mas como serão construídas muitas árvores, essa estratégia se torna poderosa, e costuma evitar o *overfitting*.

5.5 Construção das próximas árvores

Na construção da próxima árvore, os dois processos anteriores se repetirão, levando a criação de uma nova árvore. Provavelmente essa árvore será diferente da primeira, pois tanto na seleção das amostras, quanto na seleção das variáveis, o processo acontece de maneira aleatória. Podemos construir quantas árvores quisermos, sendo que quanto mais árvores criadas, melhor serão os resultados do modelo, até determinado ponto, onde uma nova árvore não conseguirá levar a uma melhora significativa no desempenho do modelo.

5.6 Prevendo novos valores

Tendo o modelo de *machine learning* devidamente criado, podemos apresentar novos dados e obter o resultado da previsão. Cada árvore criada irá apresentar o seu resultado, sendo que em problemas de regressão será realizada a média dos valores previstos, e esta média informada como resultado final, e em problemas de classificação o resultado que mais vezes foi apresentado será o escolhido.

6 Análise dos Dados Reais

Neste trabalho os resultados apresentados serão referentes a um banco de dados cedido por uma instituição financeira de uma empresa do varejo. O modelo a ser desenvolvido neste capítulo será baseado em técnicas de Regressão Logística utilizando ferramentas de *Big Data*, tendo o intuito de auxiliar estrategicamente o modo de cobrança realizado por uma empresa, passando de um estilo comum, sem nenhum conhecimento do perfil dos clientes, para uma cobrança estratégica e baseada em dados.

Mais especificamente, os dados são referentes a uma base de clientes inadimplentes, ou seja, com débitos em aberto, que realizaram algum tipo de acordo para a realização do pagamento de suas dívidas. A partir disso, o objetivo principal é descobrir se os clientes inadimplentes dessa empresa possuem um perfil mais voltado para canais digitais ou não. Sendo canais digitais: *SMS*, *Email*, *Voicer* (mensagem de áudio) e P.A. Digital (Ponto de Atendimento em que o cliente conversa por texto em tempo real com um *chatbot* (atendente virtual)), e canal não digital: P.A. Humana (Ponto de Atendimento em que o cliente conversa emtempo real com um atendente por ligação).

É de grande valor para o negócio saber se o cliente inadimplente tem um perfil digital ou não, já que um dos maiores desafios do sistema de cobrança é conseguir contatar o cliente, o que muitas vezes não é possível ou necessita de muitas tentativas por vários tipos de canais. O problema disso é que cada tentativa de contato gera um custo, sendo este praticamente o dobro no caso da P.A. Humana que não é um canal digital.

Sendo assim, com o auxílio do modelo a ser desenvolvido, a área poderá direcionar esforços de forma inteligente, entrando em contato com um cliente que tenha maior propensão de ser do grupo denominado digital por um canal digital, e um cliente com menor propensão ao grupo digital por um canal não digital. Colocando em prática um grande ditado popular: *"Tempo é dinheiro"*, visando a meta de aumentar as chances de um possível contato com o cliente em menos tentativas, diminuindo os custos buscando cada vez mais direcionar o meio de cobrança para os canais digitais.

6.1 Variáveis

O banco de dados utilizado conta com aproximadamente 440 mil linhas e 10 variáveis que serão descritas a seguir:

Quadro 2 - Variáveis

Variável	**Tipo**	**Categoria**	**Descrição**
Atraso	Contínua	-	Tempo que o cliente está em atraso em dias
Behavior	Ordinal	-	Pontuação que analisa o comportamento do consumidor com base em dados gerados no histórico de transações
Cadastro	Contínua	-	Tempo de cadastro
Empréstimo	Nominal	Sim ou Não	Sim: Cliente tem empréstimo ativo ou Não: Cliente não tem empréstimo ativo
Gasto	Discreta	-	% de gasto nos últimos 2 anos em ecommerce
Idade	Contínua	-	Idade em anos
Parcelas	Discreta	-	Quantidade de parcelas que o cliente acordou
Resultado	Nominal	Digital ou Não Digital	Digital:Cliente possui perfil digital ou Não Digital: Cliente não possui perfil digital
UF	Nominal	-	Unidade Federativa do Cliente
Valor	Contínua	-	Valor do acordo que o cliente concordou em pagar

Fonte: Elaborado pela autora

6.2 Variáveis Categorizadas

Quadro 3 - Variáveis Categorizadas

Variável	**Categoria**	**Descrição**
Atraso	1 a 8	1 = 001 a 006 dias 2 = 007 a 030 dias 3 = 031 a 060 dias 4 = 061 a 090 dias 5 = 091 a 120 dias 6 = 121 a 150 dias 7 = 151 a 180 dias 8 = maior ou igual a 180 dias
Behavior	1 a 4	1 = 01 a 03 (Ruim) 2 = 04 a 06 (Médio) 3 = 07 a 09 (Bom) 4 = 10 a 11 (Ótimo)
Cadastro	1 a 3	1 = 0 a 2 anos 2 = 3 a 5 anos 3 = maior que 5 anos
Empréstimo	0 ou 1	0 = Não tem empréstimo ativo 1 = Tem empréstimo ativo
Gasto	1 a 4	1 = 0 2 = 1 a 30% 3 = 31 a 70% 4 = 71 a 100%
Idade	1 a 3	1 = Até 30 anos 2 = 31 a 50 anos 3 = mais que 50 anos
Parcelas	1 a 5	1 = 1 (à vista) 2 = 2 a 5 3 = 6 a 10 4 =11 a 20 5 =21 a 30

(Continua na próxima página)

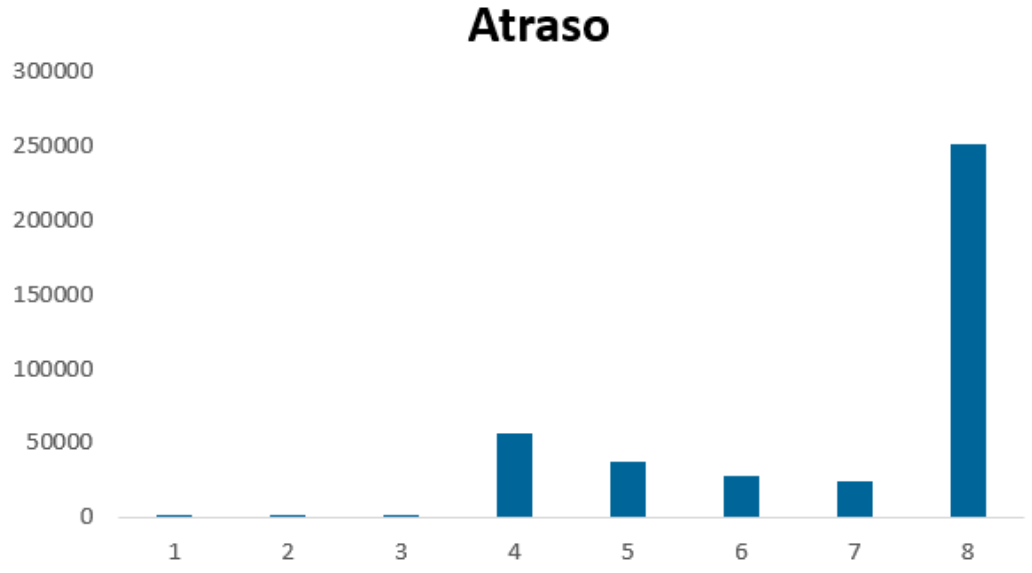
Resultado	0 ou 1	0 = Não possui perfil digital 1 = Possui perfil digital
UF	1 a 5	1 = Sul 2 = Sudeste 3 = Norte 4 = Nordeste 5 = Centro - Oeste
Valor	1 a 5	1 = R$ 0,00 a R$ 100,00 2 = R$ 100,01 a R$ 200,00 3 = R$ 200,01 a R$ 500,00 4 = R$ 500,01 a R$ 1.000,00 5 = maior que R$ 1.000,00

Fonte: Elaborado pela autora

6.3 Análise Exploratória de Dados

Nos gráficos abaixo é possível ver a distribuição de frequência das variáveis categorizadas que serão utilizadas no modelo.

Figura 7 – Distribuição Variável Atraso

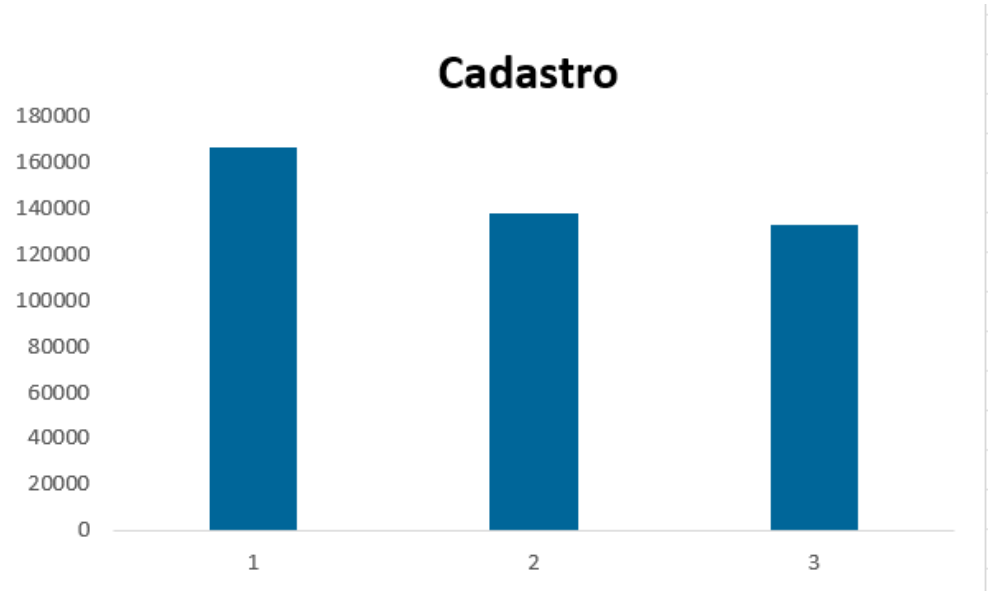


Fonte: Elaborado pela autora

Pelo gráfico da variável Atraso é possível perceber que a maior parte dos clientes inadimplentes estão concentrados na categoria 8, sendo um atraso maior ou igual a 180 dias. A segunda categoria mais relevante é a número 4 que representa um atraso de 61 a

90 dias.

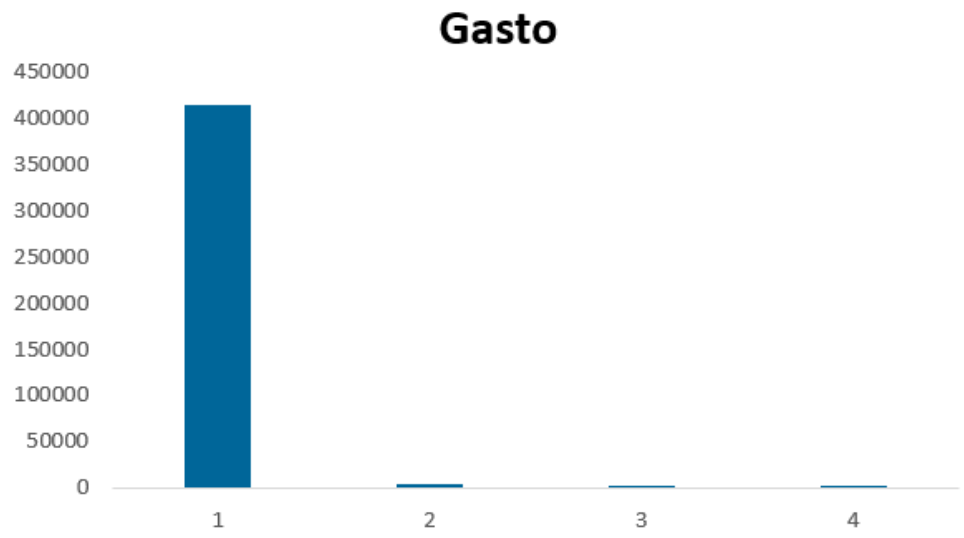
Figura 8 – Distribuição Variável Cadastro



Fonte: Elaborado pela autora

Pelo gráfico da variável Cadastro é possível ver que não há muita discrepância entre as categorias, mas ainda sim a categoria 1 se destaca representando os clientes que possuem de 0 a 2 anos de cadastro na empresa, ou seja, são clientes mais novos.

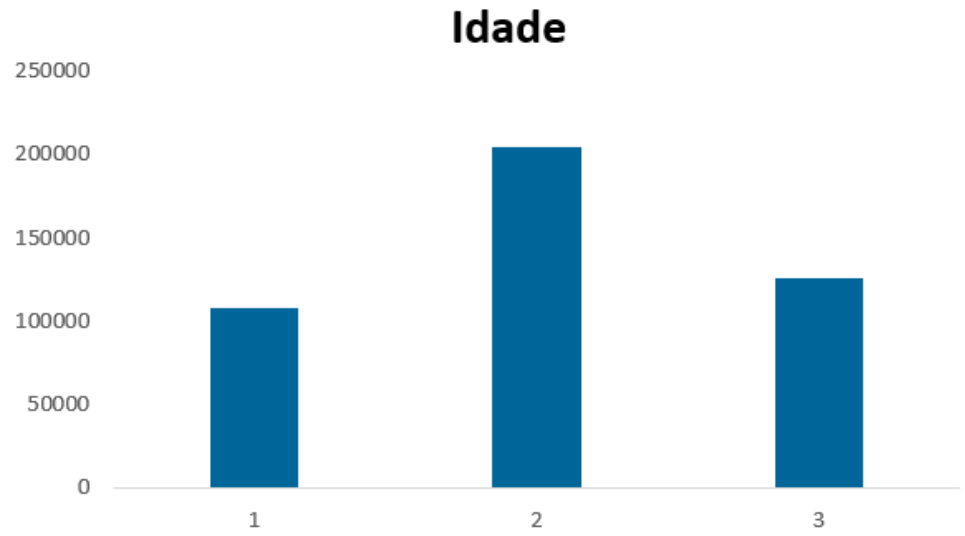
Figura 9 – Distribuição Variável Gasto



Fonte: Elaborado pela autora

Pelo gráfico da variável Gasto é possível ver que a primeira categoria se destaca, representando pessoas que não tiveram nenhum gasto em e-commerce.

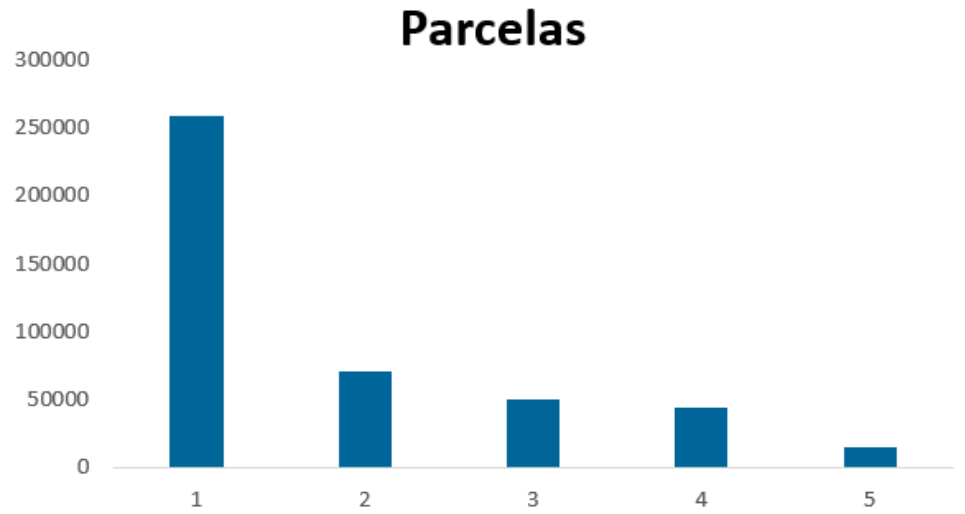
Figura 10 – Distribuição Variável Idade



Fonte: Elaborado pela autora

Pelo gráfico da variável Idade é possível ver que o volume maior de clientes está concentrado na categoria 2, equivalente a pessoas de 31 a 50 anos.

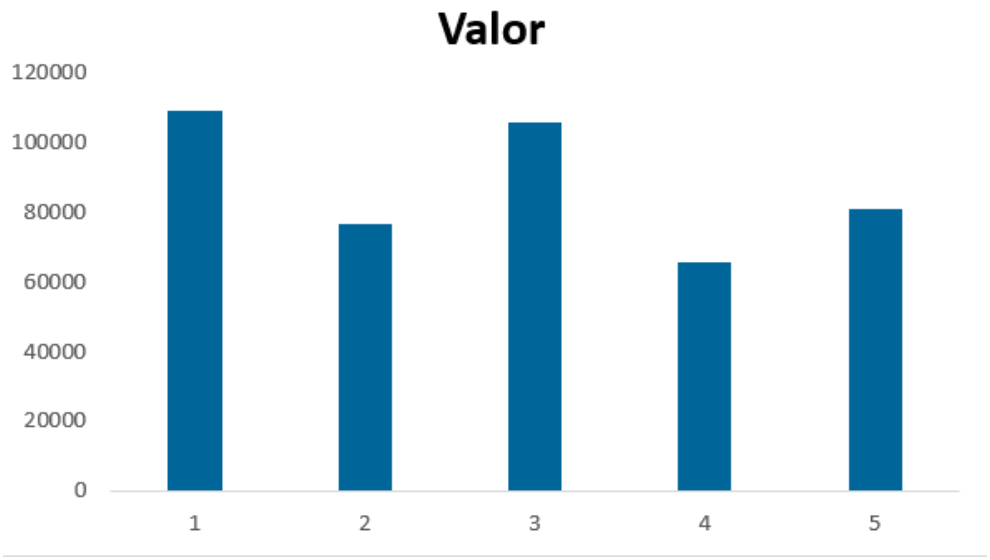
Figura 11 – Distribuição Variável Parcelas



Fonte: Elaborado pela autora

Pelo gráfico da variável Parcelas é possível ver que a primeira categoria se destaca, representando pessoas que preferem pagar sua dívida à vista, muitas vezes por possuir um desconto diferenciado para essa modalidade de pagamento.

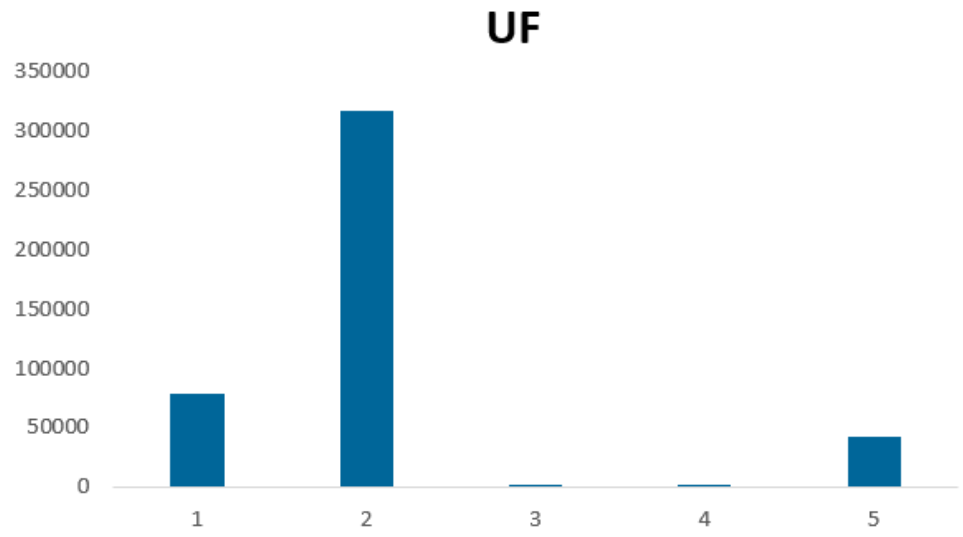
Figura 12 – Distribuição Variável Valor



Fonte: Elaborado pela autora

Pelo gráfico da variável Valor é possível ver que a variável está bem distríbuida, sendo que as categorias mais altas são a 1 e a 3, sendo a categoria 1 definida por pessoas que gastaram de R$1,00 a R$100,00 e a categoria 3 por pessoas que gastaram de R$200,01 a R$500,00.

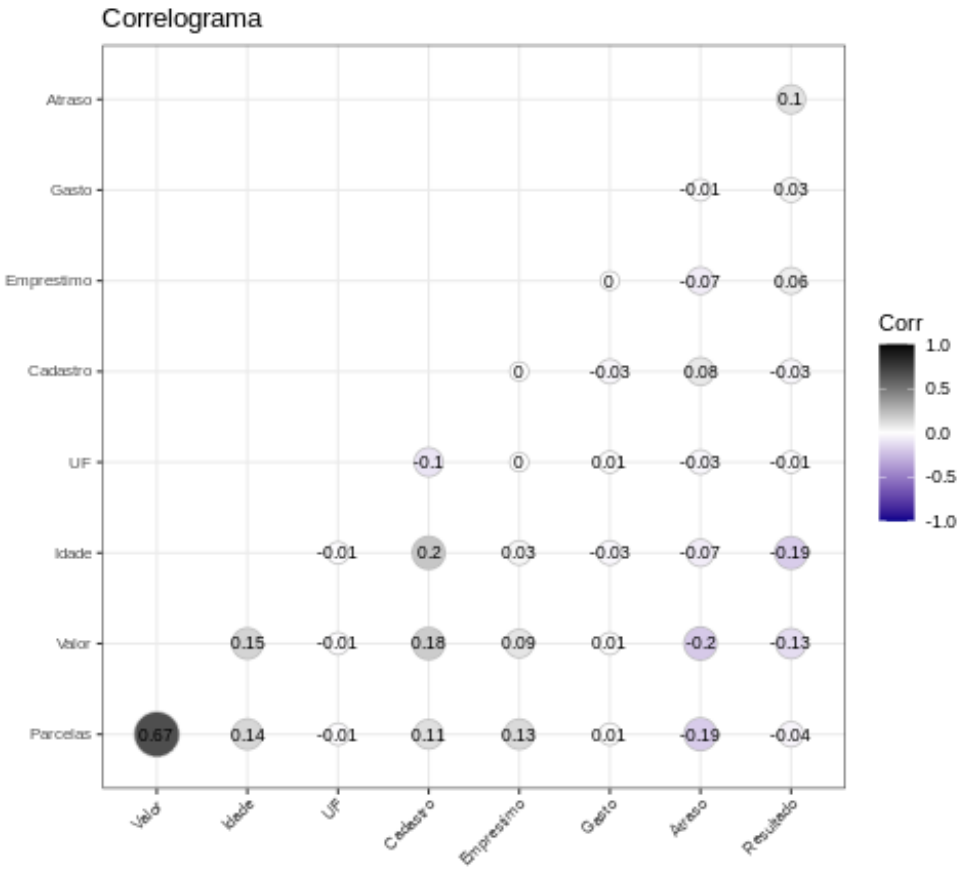
Figura 13 – Distribuição Variável UF



Fonte: Elaborado pela autora

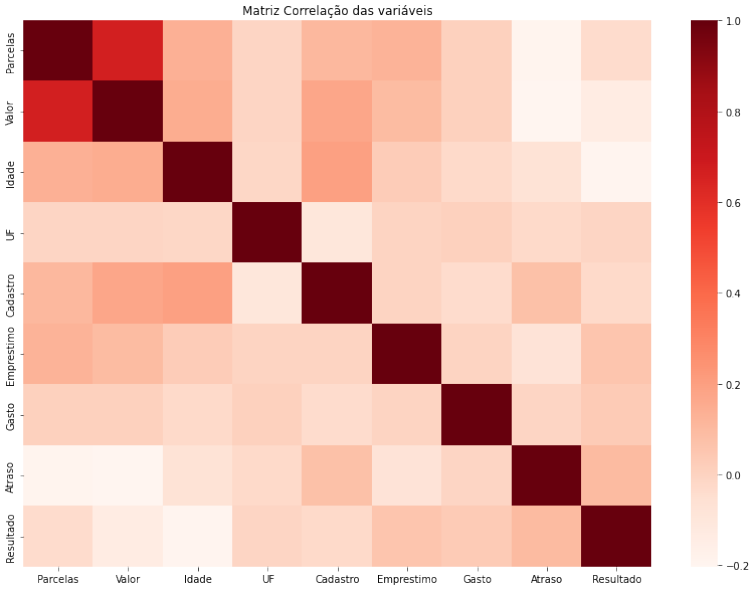
Pelo gráfico da variável UF é possível observar que a maior parte dos clientes está na categoria 2, classificada como região sudeste.

Figura 14 – Correlograma



Fonte: Elaborado pela autora

Figura 15 – Matriz de Correlação das Variáveis



Fonte: Elaborado pela autora

6.4 Construção do Modelo

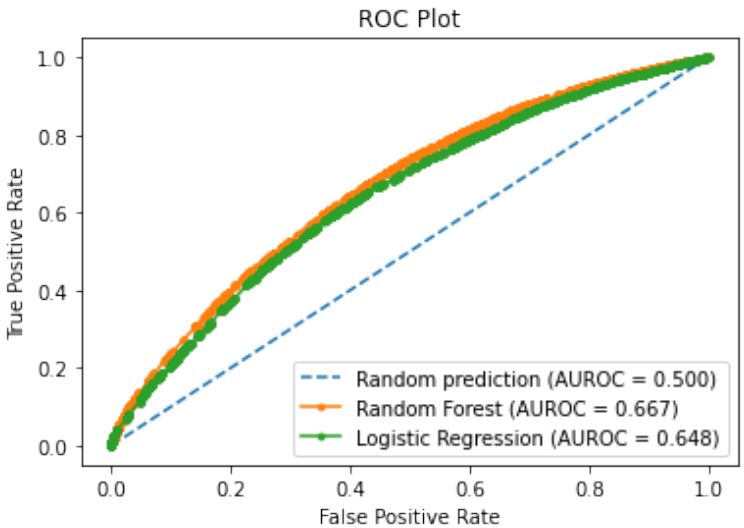
Inicialmente, a base utilizada nesse trabalho era constituída de 440 mil observações, porém, após alguns tratamentos realizados que serão descritos a seguir, o tamanho do banco de dados utilizado para modelagem foi de 275 mil observações. Na etapa de pré-processamento, os dados foram tratados nos softwares com interface SQL: *Hive* e Impala. Nesse processo foram feitas junções de bases e vários tipos de tratamentos de variáveis, como por exemplo, a data em que o cliente começou a atrasar seu pagamento foi transformada na variável atraso, que possui os dias em atraso.

Após essa etapa inicial, a base de dados passou por uma segunda etapa de tratamentos, dessa vez utilizando a ferramenta *Python*. Foi realizado um balanceamento na base a partir da técnica KNN (*K-Nearest Neighbors Algorithm*) ou K-ésimo Vizinho mais Próximo que exclui da variável resposta com maior volume, as observações que possuem maior semelhança, procurando deixar a base final balanceada mais heterogênea. Além do balanceamento, houve o tratamento de variáveis nulas que resultou na exclusão da variável *Behavior*, já que a mesma possuia mais de 70% de suas observações nulas, deixando a variável pouco representativa. Por fim, as variáveis categóricas utilizadas foram transformadas em *dummies* já que sempre que deseja-se incluir variáveis categóricas em modelos que aceitam apenas variáveis numéricas devemos realizar essa transformação.

6.5 Resultados

6.5.1 Regressão Logística

Figura 16 – Curva ROC



Fonte: Elaborado pela autora

O modelo foi construído considerando 80% dos dados com observações que foram

escolhidas de forma aleatória para compor o conjunto de treinamento. Os 20% de dados restantes foi definido como conjunto teste e será destinado a avaliação do modelo.

Os parâmetros significativos estimados do modelo de Regressão Logística, assim como seus respectivos intervalos de confiança são apresentados na Tabela 1.

Tabela 1 – Estimativa dos parâmetros significativos e respectivos intervalos de confiança

Parâmetro	Coeficiente	Intervalo de confiança	P-Valor
Atraso 8	1.225705	[0.76375863 ; 1.691280545]	9.63e-12
Cadastro 1	0.171520	[0.14361421 ; 0.199430537]	$<$ 2e-16
Idade 2	-0.641128	[-0.69288097 ; -0.589401027]	$<$ 2e-16
Parcela 1	0.260885	[0.21672156 ; 0.305158772]	$<$ 2e-16
Parcela 2	0.447751	[0.38377102 ; 0.512121256]	$<$ 2e-16
Valor 1	0.103883	[0.05921869 ; 0.148576209]	2.10e-09
Valor 3	0.080119	[0.04426131 ; 0.116004909]	8.74e-09

Fonte: Elaborado pelo autor

6.5.1.1 Avaliação do modelo

Pode-se notar pelo P-Valor, que as variáveis presentes na Tabela 1 obtiveram o nível de significância próximo de 1%, indicando que essas são as variáveis mais explicativas e que devem estar no modelo final.

Após ser realizado o treinamento no modelo com os 80% de dados, o modelo foi aplicado nos 20% dos dados restantes, para verificar o desempenho nos dados de teste.

Segue abaixo a matriz de confusão para o modelo em que tem-se o perfil digital como evento de interesse, de modo que, Y(1) = Possui perfil digital e Y(0) = Não possui perfil digital.

Quadro 5 - Matriz de Confusão

	Valor Predito	
Valor Verdadeiro	0	1
0	21.953	551
1	473	15.469

Fonte: Elaborado pelo autor

E as métricas de desempenho ficam,

Quadro 6 - Métricas de desempenho

Fonte: Elaborado pelo autor

Métrica	**Valor**
Acurácia	0.64
Precisão	0.58
Sensibilidade	0.62
F1 - Score	0.61

Como a taxa de acerto do modelo presente no Quadro 6 de 0,64, conclui-se que o modelo possui uma boa capacidade preditiva para o tema em questão.

6.5.2 Random Forest

Dando prosseguimento na construção dos modelos, um outro experimento foi realizado, dessa vez testando um modelo de floresta aleatória. Esse modelo foi ajustado usando o mesmo conjunto de dados usado para o treinamento do modelo.

6.5.2.1 Avaliação do modelo

Quadro 7 - Matriz de Confusão

	Valor Predito	
Valor Verdadeiro	0	1
0	23.434	479
1	402	16.386

Fonte: Elaborado pelo autor

Após ser realizado o treinamento no modelo com os 80% de dados, o modelo foi aplicado nos 20% dos dados restantes para verificar o desempenho em dados de teste. E as métricas de desempenho ficam,

Quadro 8 - Métricas de desempenho

Métrica	**Valor**
Acurácia	0.667
Precisão	0.62
Sensibilidade	0.64
F1 - Score	0.63

Fonte: Elaborado pelo autor

Como a taxa de acerto do modelo apresentada no Quadro 8 de 0,667 conclui-se que, o modelo possui uma boa capacidade preditiva, tendo um pouco mais de acurácia do que o modelo de regressão logística.

7 Considerações Finais

Neste trabalho foi apresentado como funciona o ciclo do crédito até sua chegada a inadimplência, além da predição do perfil de clientes inadimplentes da empresa em estudo.

Para isso, foram utilizados modelos de Regressão Logística e *Random Forest* que tiveram efeitos semelhantes em relação às sensibilidades, especificidades e acurácia. Quando comparados, é possível perceber que o *Random Forest* com aproximadamente 67% de acurácia se saiu um pouco melhor do que o de Regressão Logística com 65% .

A partir da análise exploratória e do resultado dos modelos, conclui-se que os clientes inadimplentes dessa empresa, em sua maioria, não possuem um perfil digital. Além disso, fica claro que boa parte deles possuem idade entre 31 e 50 anos e vivem na região sudeste, possuindo dívidas entre R$100,00 e R$500,00 com atraso acima de 1 ano.

Por fim, a maioria das pessoas que fizeram acordos para realizarem o pagamento de suas dívidas e deixarem de ser inadimplentes, preferiu realizar o pagamento à vista, sendo que isso ocorre na maioria das vezes pelo fato das empresas de cobrança darem maior desconto para essa modalidade, visando que o cliente quite sua dívida o mais rápido possível, ao invés de prometer um pagamento parcelado e não cumprir com todas as parcelas. Outro ponto observado é que essa base de clientes não tem um perfil de gasto em *e-commerce*, reforçando que a utilização de métodos de cobrança digitais pode não ter muito sucesso, já que os clientes em sua maioria não fazem compras por canais digitais.

É importante ressaltar que essas conclusões são válidas para os dados utilizados neste trabalho, não sendo possível afirmar que os modelos de cobrança ou recuperação de crédito em geral serão sempre semelhantes. Esses resultados podem ser diferentes dependendo das variáveis que compõem um conjunto de dados, do tipo de amostra e da forma como foram coletadas e tratadas.

Referências

HOSMER, D. W.; LEMESHOW, S. Applied logistic regression, 2 ed. New York: John Wiley Sons, 1989.

SARMENTO, A. M. B.; Regressão Logística Uma introdução ao modelo estatístico. 1 ed. Portugal: Laureata, 2015.

BREIMAN, L.; FRIEDMAN, J. H.; OLSHEN, R. A.; STONE, C. J. Classification and Regression Trees; California: Wadsworth International, 1984. 358p.

CRAMER, J. S. The Origins of the Logistic Regression; Tinbergen Institute Discussion Paper, No 2002-119/4. Disponível em: <https://ssrn.com/abstract=360300>. Acesso em 02 de set. de 2021.

HAN, J.; KAMBER, M.; JIAN, P. Data Mining Concepts and Techniques. 3. ed. Waltham: Elsevier, 2012.

WITTEN, I. H.; FRANK, E.; HALL, A. M. Data mining: practical machine learning tools and techniques. 3. ed. Burlington: Elsevier, 2011.

SANTOS, Jose Odalio dos. Análise de crédito: empresas e pessoas físicas. 2.ed.São Paulo: Atlas, 2006.

SILVA, José Pereira da. Gestão e análise de risco de crédito.3.ed. São Paulo:Atlas, 2000.

SÁ, A. Estabelecimento de limite de crédito: uma nova abordagem para um velho problema.Rio de Janeiro: Qualitymark, 2004.

BLATT, A. Avaliação de risco e decisão de crédito: um enfoque prático. São Paulo: Nobel, 1999.

TAVARES, Ricardo Ferro. Crédito e cobrança. São Paulo: Atlas, 1988.

SCHMARZO, Bill. Big Data: Understanding How Data Powers Big Business. Wiley, 2013.

ZAHARIA, M.; CHAMBERS B. Spark: The Definitive Guide: Big Data Processing

Made Simple. O'Reilly Media, 2018.
BOWEN, R.; COAR, K. Apache - Guia Prático. Alta Books, 2009.

WHITE, Tom. Hadoop: The Definitive Guide: Storage and Analysis at Internet Scale. O'Reilly Media, 2015.

RUSSELL, John. Getting Started with Impala: Interactive SQL for Apache Hadoop. O'Reilly Media, 2014.

BRAGA, A. C. Curva ROC: Aspectos funcionais e aplicações, 2000. Tese de Doutorado, Universidade do Minho, Braga, 2000.

FAYYAD, U.; PIATETSKY-SHAPIRO, G.; SMYTH, P. From data mining to knowledge discovery in databases. AI magazine, Rhode Island, v.17, n.3, p.37-54, Jul. 1996. Disponível em: https://www.aaai.org/ojs/index.php/aimagazine/article/view/1230/1131. Acesso em: 24 Set. 2021.

BEZERRA, Paula.Disponível em: <https://www.creditas.com/exponencial/inadimplencia-no-brasil/>. Acesso em: 27 de fev. de 2021.

PEREIRA, Renée. Disponível em: <https://economia.uol.com.br/noticias/estadao-conteudo/2019/12/02/maior-parte-dos-inadimplentes-do-pais-tem-divida- contraida-ha-mais-de-7-anos.htm>.Acesso em: 27 de fev. de 2021.

LEOPARD, S.; SONG, J. Disponível em: <https://www.sas.com/content/dam/SAS/en-us/doc/event/analytics-experience-2016/ using-logistic-regression-predict-credit-default.pdf>. Acesso em: 14 de mar. de 2021.

CNC – Confederação Nacional do Comércio. Pesquisa de Endividamento e Inadimplência do Consumidor, 2013. Disponível em: < http://www.ibegi.org.br/indicadores.php> Acesso em: 28 de fev. de 2021.

Banco Central do Brasil. Relatório de Inflação, 2021. Disponível em: <https://www.bcb.gov.br/content/ri/relatorioinflacao/202103/ri202103p.pdf>. Acesso em: 10 de abr. de 2021.

8 Apêndices

8.1 Códigos Python e R

Atrávés desse código mesclado entre as linguagens de programação *Python* e R, é possível realizar vários tratamentos no banco de dados como o balanceamento da base, tratamento dos nulos e criação de variáveis dummy. Também é possível criar os modelos de Regressão Logística e *Random Forest*.

Listing 8.1 – Elaborado pelo autor.

```
%load_ext rpy2.ipython

import warnings
warnings.filterwarnings("ignore", category=DeprecationWarning)

!pip install imblearn

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.formula.api as smf
import statsmodels.api as sm
from sklearn.linear_model import LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,
confusion_matrix, classification_report
from imblearn.over_sampling import SMOTE
from imblearn.under_sampling import NearMiss
from scipy import stats
%matplotlib inline

from google.colab import drive
drive.mount('/content/drive')

df = pd.read_csv('/content/drive/My Drive/Base_modelo.csv', sep = ';')
```

```
df.dropna(inplace = True)

%%R -i df
df <- data.frame(df)

%%R
library(magrittr)
library(dplyr)
install.packages("ggcorrplot")
library(ggcorrplot)
install.packages("fastDummies")
library(fastDummies)
install.packages("caret")
install.packages("prediction")
library(prediction)
library(caret)
library(MASS)


%%R

df<- df %>%
mutate(parcelas = as.factor(Parcelas),
valor = as.factor(Valor),
uf = as.factor(UF),
cadastro = as.factor(Cadastro),
gasto = as.factor(Gasto),
atraso = as.factor(Atraso),
idade = as.factor(Idade))

df.head()

%%R

df %>%
select_if(is.numeric) %>%
cor(use = "pairwise.complete.obs") %>%
ggcorrplot::ggcorrplot(hc.method = TRUE,
type = "lower",
```

```
lab = TRUE,
lab_size = 3,
tl.cex = 8,
method="circle",
colors = c("darkblue", "white", "black"),
title="Correlograma",
ggtheme=theme_bw)



corr = df.corr()
corr.style.background_gradient(cmap='coolwarm')

plt.subplots(figsize=(14,10))
ax = plt.axes()
ax.set_title("Matriz Correla   o das vari veis")
corr = df.corr()
sns.heatmap(corr,
xticklabels=corr.columns.values,
yticklabels=corr.columns.values,
cmap="Reds")
plt.show()

%%R
df_dummy <- dummy_columns(.data = df,
select_columns = c("Parcelas",
"Valor","Idade","uf","Cadastro","Gasto","Atraso"),
remove_selected_columns = T,
remove_most_frequent_dummy = T)

%%R
# Base teste
base_teste_x <- createDataPartition
(df_dummy$Resultado, p=0.10, list=FALSE)
base_teste <- df_dummy[base_teste_x,]

#Base modelo
base_modelo <- df_dummy[-base_teste_x,]
```

```
# Dividir a base em 80% para treino e 20% para validacao
trainIndex <- createDataPartition
(base_modelo$Resultado, p=0.80, list=FALSE)
trainData <- base_modelo[trainIndex,]
validationData <- base_modelo[-trainIndex,]

%%R
modelo_database_dummies <- glm(formula = Resultado ~ .,
data = trainData,
family = "binomial")

%%R
summary(modelo_database_dummies)

%%R
step_database_dummies <- stepAIC
(modelo_database_dummies, direction = "both",
trace =  FALSE, k = qchisq(p = 0.05, df = 1, lower.tail = FALSE))

%%R
summary(step_database_dummies)

%%R
ROC <- roc(response = validationData_v2$Resultado,
predictor = pred.validationData)

ggplotly(
ggroc(ROC, color = "#440154FF", size = 1) +
geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1),
color="grey40",
size = 0.2) +
labs(x = "Especificidade",
y = "Sensitividade",
title = paste("Area abaixo da curva:",
round(ROC$auc, 3),
"|",
"Coeficiente de Gini",
round((ROC$auc[1] - 0.5) / 0.5, 3))) +
```

```
theme_bw())

#Random Forest

from sklearn.ensemble import RandomForestClassifier

rf = RandomForestClassifier(max_features=5, n_estimators=500)
rf.fit(x_train, y_train)

r_probs = [0 for _ in range(len(y_test))]
rf_probs = rf.predict_proba(x_test)
lr_probs = lr.predict_proba(x_test)

rf_probs = rf_probs[:, 1]
lr_probs = lr_probs[:, 1]

from sklearn.metrics import roc_curve, roc_auc_score

r_auc = roc_auc_score(y_test, r_probs)
rf_auc = roc_auc_score(y_test, rf_probs)
lr_auc = roc_auc_score(y_test, lr_probs)

print('Random (chance) Prediction: AUROC = %.3f' % (r_auc))
print('Random Forest: AUROC = %.3f' % (rf_auc))
print('Logistic Regression: AUROC = %.3f' % (lr_auc))

r_fpr, r_tpr, _ = roc_curve(y_test, r_probs)
rf_fpr, rf_tpr, _ = roc_curve(y_test, rf_probs)
lr_fpr, lr_tpr, _ = roc_curve(y_test, lr_probs)

import matplotlib.pyplot as plt

plt.plot(r_fpr, r_tpr, linestyle='--',
label='Random prediction (AUROC = %0.3f)' % r_auc)
plt.plot(rf_fpr, rf_tpr, marker='.',
label='Random Forest (AUROC = %0.3f)' % rf_auc)
plt.plot(lr_fpr, lr_tpr, marker='.',
label='Logistic Regression (AUROC = %0.3f)' % lr_auc)
```

```
# Title
plt.title('ROC Plot')
# Axis labels
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
# Show legend
plt.legend() #
# Show plot
plt.show()
```