

Predição de Pagamentos Atrasados Através de Algoritmos Baseados em Árvore de Decisão


Prediction of Late Payments Using Decision Tree-Based Algorithms

Arthur Flor de Sousa Neto ¹  orcid.org/0000-0002-0522-2150

José Fernando Guilhermino da Silva ¹  orcid.org/0000-0001-6439-2435

Glauber Nascimento de Oliveira ¹  orcid.org/0000-0002-2472-9811

Bruno César de Andrade ¹  orcid.org/0000-0003-0503-0770

Alexandre Magno Andrade Maciel ¹  orcid.org/0000-0003-4348-9291

¹ Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil.

E-mail do autor principal: afsn@ecomppoli.br

Resumo

O processo *Invoice-to-Cash* é essencial para a estabilidade financeira de qualquer empresa, tendo em vista a coleta de contas a receber como sua principal atividade. No entanto, apesar da sua importância, a etapa de cobrança é geralmente processada manualmente, no qual ocasiona no contato a todos os clientes em intervalos fixos, mesmo que alguns sempre tenham pago em dia. Assim, o trabalho explora técnicas de mineração de dados com aprendizado de máquina, visando otimizar o processo de coleta através de predições dos pagamentos das faturas. Para isto, foi utilizado oito algoritmos baseados em árvore de decisão, aplicados em três etapas: (i) identificar as faturas com pagamento no prazo ou atrasado; (ii) identificar entre as faturas atrasadas, pagamento no mês de vencimento ou posterior; e (iii) prever entre as faturas atrasadas, quantos dias de atraso terão além do mês de vencimento. Por fim, através dos resultados obtidos dos melhores modelos para cada etapa, foi obtido uma precisão média de 81,85%, 85,63% e 73,98%, respectivamente.

Palavras-Chave: Contas a Receber; Coleta de Pagamento; Mineração de Dados; Aprendizado de Máquina; Árvore de Decisão.

Abstract

The Invoice-to-Cash process is essential for the financial stability of any company, in view of the collection of accounts receivable as its main activity. However, despite its importance, the billing stage is usually processed manually, which causes contact with all customers at fixed intervals, even if some have always paid on time. Thus, the work explores data mining techniques with machine learning, aiming to optimize the collection process through prediction of invoice payments. For this, eight algorithms based on decision tree were used, applied in three steps: (i) identify invoices with payment on time or late; (ii) identify among the overdue invoices, payment in the due date month or later; and (iii) predict among the overdue invoices, how many days of delay they will have beyond the due month. Finally, through the results obtained from the best models for each step, was obtained an average precision of 81.85%, 85.63% and 73.98%, respectively.

Keywords: Accounts Receivable, Payment Collection; Data Mining; Machine Learning; Decision Tree.

1 Introdução

O *Business Analytics* é a prática exploratória dos dados de uma organização com ênfase na análise estatística. Dessa forma, é utilizado por empresas através de procedimento sistemático de coleta de dados para tomar decisões baseadas em dados [1]. Nas últimas décadas, o volume de dados disponíveis aumentou significativamente e esse processo analítico passou por uma grande transição. Logo, métodos automatizados tornaram-se necessários para a análise de dados. Isso abriu espaço para algoritmos de aprendizado de máquina, que segundo Murphy [2], é um conjunto de métodos que detecta automaticamente padrões nos dados para, em seguida, utilizar o conhecimento adquirido na predição de dados novos.

Para este contexto, empresas possuem o processo de *Order-to-Cash*, no qual se refere ao recebimento e processamento de pedidos dos clientes [3]. Embora seu número de etapas possa variar de empresa para empresa (dependendo do seu segmento e tamanho), o conjunto de atividades do setor financeiro, conhecido como *Invoice-to-Cash*, são essenciais para o funcionamento de qualquer negócio [1][3].

1.1 Descrição do Problema

O processo *Invoice-to-Cash*, por sua vez, lida com a priorização de contas, atividades de contato com o cliente, chamadas de cobrança, escalonamento e resolução de disputas [1]. Na maioria das vezes, essas etapas são processadas manualmente e, portanto, lentas, caras e imprecisas, apesar de sua importância para os negócios. Além disso, as ações de cobrança são tipicamente genéricas e não consideram as especificidades do cliente. Nesse sentido, todos os clientes são contatados em intervalos fixos, embora alguns sempre tenham pago em dia; e geralmente quanto mais tarde o contato com um cliente, menor a probabilidade de as faturas serem pagas a tempo [3].

Além disso, o contato repetido de bons clientes pode levar a uma menor satisfação do cliente. Tais ineficiências nas práticas atuais levam a atrasos significativos nas coletas ou até mesmo à falha na cobrança antes dos prazos [4].

1.2 Objetivo

Este trabalho propõe a exploração de técnicas de mineração de dados com aprendizado de máquina, visando otimizar o *Invoice-to-Cash*, especificamente a etapa de coleta de contas a receber. Para este cenário, são analisados modelos baseados em árvore de decisão, no qual conseguem oferecer informações relevantes para auxiliar o processo.

1.3 Justificativa

De fato, é de grande interesse administrar de forma eficaz a coleta de contas a receber, tendo em vista o problema financeiro que pode gerar em uma empresa. Por exemplo, só a indústria de construção canadense em 2012 gerou 111 bilhões de dólares, todos em forma de fatura. Já no mercado global, cerca de 550 bilhões de faturas são criadas anualmente [5][6]. Além disso, as empresas atuais acumulam grandes volumes de dados sobre seus clientes, o que torna possível a eficácia da coleta em grande escala [3].

1.4 Escopo Negativo

Vale ressaltar que o trabalho não se propõe em automatizar completamente o sistema de cobrança, mas sim auxiliar na tomada de decisão no momento de contatar um cliente. Isto é, os modelos vão apenas indicar as faturas com maior probabilidade de atraso.

Por fim, este trabalho está organizado da seguinte maneira: a seção 2 fala sobre a fundamentação teórica, no qual pode-se entender a área do negócio, mineração de dados e os trabalhos relacionados; a seção 3 descreve a base de dados utilizada, assim como o pré-processamento aplicado; a seção 4 apresenta o experimento realizado, bem como os resultados alcançados; e a seção 5 conclui o trabalho realizado, trazendo as considerações finais.

2 Fundamentação Teórica

Nesta seção, são apresentados os temas que constituem a base teórica para realização deste trabalho.

2.1 Accounts Receivable

Accounts Receivable, ou contas a receber, são consideradas uma das partes essenciais das finanças na cadeia de abastecimento e da estabilidade financeira das empresas [7][8].

Existem muitas métricas usadas para medir a eficácia da cobrança de uma empresa. Uma das medidas mais básicas é o Índice de Eficácia de Coleção (*Collection Effectiveness Index*, CEI), que é definido como:

$$CEI = \frac{B + (CS/N) - ET}{B + (CS/N) - EC} \times 100 \quad (1)$$

onde B corresponde ao número de contas a receber iniciais (*Beginning Receivables*), CS ao número de vendas a crédito (*Credit Sales*), ET ao total de contas a receber (*Ending Total Receivables*), EC ao total já recebido (*Ending Current Receivables*) e N ao número de meses ou dias do período analisado. Esta métrica mede principalmente o número e a proporção de contas a receber em um determinado tempo [8].

Outra métrica relacionada é o de Dias de Vendas Pendentes (*Days Sales Outstanding*, DSO). Ela expressa o tempo médio em dias em que as contas a receber estão pendentes e é definido como:

$$DSO = \frac{ET \times NDP}{CSP} \quad (2)$$

onde NDP corresponde ao número de dias no período analisado (*Number of Days in Period Analyzed*) e CSP ao número de vendas a crédito para o período analisado (*Credit Sales for Period Analyzed*) [3].

As métricas comumente utilizadas são funções que auxiliam no tempo necessário para cobrar as faturas. Ou seja, se for possível prever o resultado de uma fatura, pode-se usar essas informações para direcionar o processo de cobrança, de modo a melhorar a métrica de cobrança desejada [2][3].

Segundo Zeng et al. [3], normalmente o setor de cobrança espera até que as faturas estejam inadimplentes para iniciar as ações de cobrança. No entanto, é possível se beneficiar do contato preventivo com potenciais contas inadimplentes. Além disso, mesmo após o vencimento de uma

fatura, é benéfico saber quais faturas provavelmente serão pagas mais cedo ou mais tarde, caso nenhuma ação seja tomada. Dado que os recursos são frequentemente limitados, pode-se escolher priorizar as faturas com base na estimativa do atraso no pagamento.

2.2 Mineração de Dados

Devido ao grande volume de dados atualmente e sua disponibilidade, o campo de estudo de mineração de dados vem atraindo cada vez mais a atenção no meio acadêmico e industrial [9]. De certa forma, a mineração de dados é o processo de explorar e identificar padrões válidos nos dados, com objetivo de extrair conhecimento útil a partir deles. Além disso, através de Inteligência Artificial, modelos de aprendizagem de máquina vêm ganhando cada vez mais espaço nesse contexto [10].

2.3 Aprendizado de Máquina

A partir do reconhecimento de padrões e Inteligência Artificial, o Aprendizado de Máquina (*Machine Learning*), explora a construção de algoritmos que aprendem e fazem previsões em grandes volumes de dados [4]. Dessa forma, o aprendizado de máquina pode ser aproveitado para aprimorar o processo de cobrança, pois permite previsões de datas de pagamento mais precisas, utilizando dados históricos. Além disso, o processo pode ser aplicado para analisar e identificar padrões ao nível de cliente [1][3].

De acordo com Hu [1] e Nanda [4], as abordagens comumente utilizadas para previsão de pagamento no processo de cobrança, são:

- **Modelo de Classificação Binária:** tarefa de classificar os elementos, dado dois grupos (classes), como: (i) pagamentos no prazo; e (ii) pagamento atrasado [4];
- **Modelo de Classificação Multiclasse:** tarefa de classificar os elementos, dado três ou mais grupos (classes). Neste contexto, o modelo classifica as faturas utilizando intervalos (*buckets*) de dias de atraso pré-definidos [11];
- **Modelo de Regressão:** tarefa de prever um valor contínuo referente ao número de dias de atraso. Neste cenário, o modelo prevê os dias

de atraso, no qual pode ser classificado posteriormente, ou não [3].

Diante das abordagens, os principais modelos utilizados são baseados em árvores de decisão [1]. Dessa forma, não somente atendem as três abordagens mencionadas, como também trazem bons resultados através do treinamento com base em atributos relevantes [4].

2.3.1 Árvore de Decisão

Algoritmos baseados em árvore de decisão são utilizados devido ao suporte à tomada de decisão e sua capacidade de construção com base em atributos mais relevantes. Além disso, podem ser utilizados tanto para tarefas de classificação, quanto para regressão [12]. A Figura 1 mostra um exemplo da estrutura de árvore de decisão.

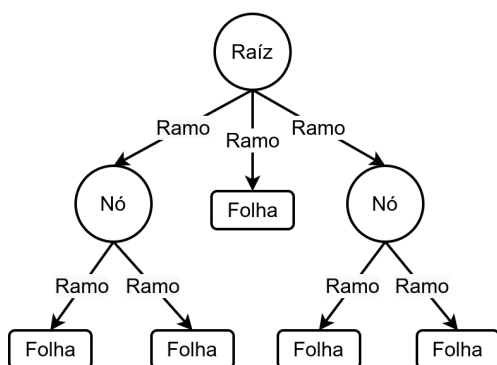


Figura 1: Representação de uma árvore de decisão, sendo os atributos, condições e decisões, representados pelos nós, ramos e folhas, respectivamente.

No contexto de predição de pagamentos e coleta de contas a receber, alguns métodos de construção podem ser utilizados no algoritmo de árvore. O primeiro método, *Bootstrap Aggregating (Bagging)*, consiste em gerar subconjuntos de exemplos através de um sorteio simples com reposição, sobre o conjunto de dados de treinamento original, chamado de "bags". Cada subconjunto é utilizado para a construção de uma nova árvore, tendo como resultado, a combinação das decisões de cada uma [13].

No entanto, através do método *Bagging*, a estrutura da árvore pode ter semelhanças entre os seus subconjuntos, e por sua vez, alta correção em suas previsões. Para evitar a combinação de estruturas semelhantes e obter uma previsão fraca, o

Random Forest altera a maneira de como os subconjuntos são aprendidos, de modo que as previsões resultantes tenham menos correlação através da aleatoriedade nas subdivisões [14].

Outro método de construção, o *Boosting*, explora o processo de combinar e complementar as árvores ao longo do aprendizado. Dessa forma, os modelos não são mais treinados separados, mas sim de forma sequencial, a partir de ajustes dos modelos treinados previamente [15].

No contexto *Boosting*, variações do algoritmo de aprendizado podem melhorar o desempenho do modelo. Inicialmente, o *Adaptive Boosting (AdaBoost)* [15] foi proposto para classificações binárias, no qual o próprio método ajusta seus parâmetros iterativamente (modelo aditivo). Por outro lado, o *Gradient Boosting* [16], propõe ser mais genérico e atender problemas mais complexos (alta dimensionalidade). Além disso, permite a otimização de duas ou mais funções de perda, o que o torna robusto a *outliers* quando comparado ao AdaBoost. Recentemente, foi proposto o método *Random Undersampling Boosting (RUSBoost)* [17], no qual se utiliza do AdaBoost, mas traz o foco no balanceamento das classes. Esse problema de desbalanceamento é aliviado durante o aprendizado, através da subamostragem aleatória da amostra em cada iteração do algoritmo. Por fim, o *eXtreme Gradient Boosting (XGBoost)* [18], traz o foco para o desempenho e construção do modelo através do *Gradient Boosting*. Considerando o processo de aprendizado sequencial, o *XGBoost* visa que a adição de cada nova árvore ao modelo, seja mais certa e otimizada, o que melhora o resultado obtido. Além disso, oferece ajustes de regularização, o que ajuda a reduzir o efeito de sobre-ajuste (*overfitting*) sobre os dados.

2.4 Trabalhos Relacionados

As técnicas de mineração de dados fornecem grande ajuda na otimização do *Invoice-to-Cash*, tendo em vista os desafios ao lidar com grandes volumes de dados financeiros [1].

Zeng et al. [3], aborda o problema da redução de contas a receber pendentes através de melhorias na estratégia de cobrança. Para isso, foi demonstrado como o aprendizado supervisionado pode ser usado para desenvolver modelos que preveem os pagamentos de faturas recém-criadas. Isso permite

ações de cobrança mais assertivas para cada cliente. Os algoritmos utilizados por Zeng et al. [3] foram baseados em árvore de decisão e delimitaram as classes em *ranges* de 30 dias, compondo 5 *buckets*.

O trabalho de Hu [1], por sua vez, apresenta previsões precisas sobre os pagamentos das faturas com base em dados históricos do cliente. Além da criação dos dados históricos, foi realizado um extenso estudo sobre atributos mais relevantes e técnicas com melhor aproveitamento neste cenário. Hu [1] conclui que *Random Forest* obteve os melhores resultados, assim como uma boa flexibilidade entre o desbalanceamento das classes (4 *buckets* de 30 dias de intervalo). Além disso, também reforça o uso do algoritmo *Support Vector Machine* (SVM) como segunda abordagem.

Para o trabalho de Nanda [4], os algoritmos baseados em árvore tiveram três abordagens: (i) regressão, para o modelo prever um número contínuo referente ao número de dias em atraso; (ii) classificação binária, para apenas determinar se a fatura será paga em dia ou atrasada; e (iii) classificação multiclasse, referente aos *buckets*, assim como os trabalhos anteriores. Neste cenário, Nanda [4], alcançou bons resultados com os algoritmos, com destaque para *Random Forest* e o *Gradient Boosting*.

Por fim, Shah [11] apresentou uma forma diferente na criação dos dados históricos. Em seu processo, foi utilizado cada *bucket* definido para delimitar os dados históricos de cada cliente. Desse modo, foi possível criar para cada intervalo de dias de atrasado, seu respectivo histórico. Ainda segundo Shah [11], a desvantagem dessa abordagem é sua limitação aos próprios intervalos, já que para qualquer mudança nos *buckets*, seja por regra de negócio, ou adaptação aos dados, o processo do histórico terá que ser refeito. Além disso, o algoritmo *Adaptive Boosting* (AdaBoost) foi o único utilizado no experimento.

3 Materiais e Métodos

Esta seção consiste em descrever a base de dados do estudo, assim como apresentar as técnicas adotadas e o experimento realizado.

3.1 Descrição da Base de Dados

A base de dados utilizada neste trabalho foi fornecida por uma empresa privada, trazendo dados de sua matriz e filiais. O conjunto comporta registros de todo o processo financeiro interno, no período de janeiro de 2018 até fevereiro de 2021.

Além disso, o conjunto de dados bruto possui cerca de 11 milhões de registros e 131 atributos, entre categóricos e numéricos. Os atributos relacionam informações da fatura, cliente e empresa em uma determinada etapa do processo.

3.2 Análise Descritiva dos Dados

A análise descritiva dos dados é considerada uma etapa fundamental para o processo de descoberta de conhecimento. Tendo em vista a interpretação prévia dos dados, as técnicas de mineração são aplicadas de maneira mais assertiva. Nesta etapa, os dados podem ser organizados utilizando distribuição de frequência e visualizados através de gráficos, mapas ou diagramas.

Dessa forma, a Tabela 1 detalha a distribuição de frequência dos dados, referente a faturas com pagamento no prazo e com atraso.

Tabela 1: Distribuição de frequência de faturas pagas no prazo (*ontime*) e com atraso (*late*).

	Classes	
	<i>ontime</i>	<i>late</i>
Limite Inferior	-368	1
Ponto Médio	-184	256
Limite Superior	0	511
Frequência Absoluta	1.642.075	1.095.940
Frequência Relativa %	59,97	40,03
Frequência Acumulada	1.642.075	2.738.015
Frequência Acumulada %	59,97	100,00

As análises preliminares indicam um grande distanciamento entre o limite inferior de faturas pagas em dia (368 dias antes da data de vencimento) e do limite máximo de faturas pagas com atraso (511 dias após a data de vencimento). Além disso, o número de faturas atrasadas é cerca de 34% menor do que faturas pagas no prazo. A Figura 2 mostra a distribuição das faturas pagas, agrupadas em intervalos de dias de atraso, sendo *ontime* todo pagamento no prazo.

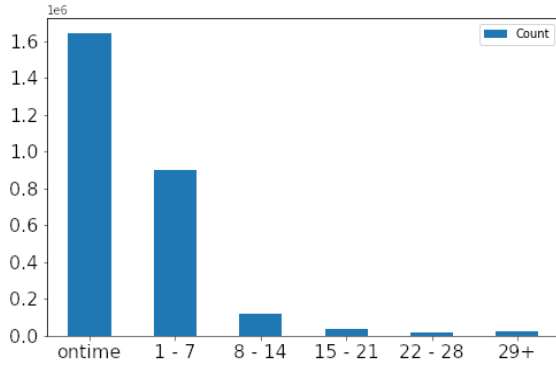


Figura 2: Distribuição de faturas pagas, agrupadas em intervalos de dias de atraso (*ontime* consiste em pagamento no prazo).

Outro fator importante para análise é a distribuição do valor total das faturas entre os dados. De modo geral, há grande dispersão nos dados referente aos valores de faturas, sendo o valor mínimo encontrado de 0,01 *cents*, enquanto o valor máximo chega a mais de 8.5 milhões de dólares. Assim, técnicas de pré-processamento foram aplicadas para minimizar tal dispersão.

3.3 Pré-processamento dos Dados

A etapa de pré-processamento é essencial para a organização e preparação da base, visando a boa qualidade dos dados para a etapa de modelagem. A Figura 3 mostra as etapas do pré-processamento realizadas neste trabalho.

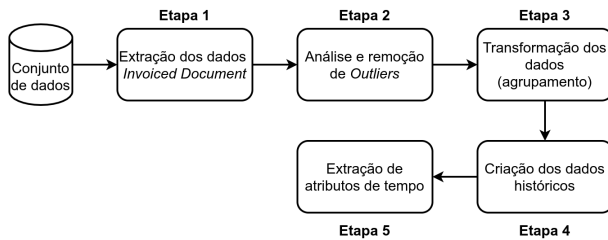


Figura 3: Etapas do pré-processamento realizadas no conjunto de dados.

Tendo em vista que o conjunto de dados bruto consiste em todos os registros do processo financeiro interno, a etapa 1 visa a seleção dos dados de interesse do trabalho, ou seja, do tipo *Invoiced Document*. Também foram removidos registros com datas de processamento inconsistentes, como faturas com data de criação maior que o próprio vencimento e pagamento. Além disso, como regra de negócio, foi desconsiderado faturas com valor menor a mil dólares, por serem consideradas vendas internas.

Ao fim da etapa 1, a base de dados possui cerca de 2.7 milhões de registros. Também foi mantido os 6 principais atributos, tendo em vista que serão utilizados para a criação de outros atributos nas etapas seguintes. Os atributos são:

- **DocumentKey:** Identificação da fatura;
- **CustomerKey:** Identificação do cliente;
- **DocumentDate:** Data de emissão;
- **DueDate:** Data de vencimento;
- **ClearingDate:** Data do pagamento;
- **InvoicedAmount:** Valor total da fatura.

A etapa 2, análise e remoção de *outliers*, foi aplicada visando minimizar a dispersão dos dados com respeito ao tempo de vencimento e ao valor da fatura. Assim, foi utilizado o método *Robust Z-score* [19], já que utiliza o Desvio Mediano Absoluto (*Median Absolute Deviation*), em vez do desvio padrão. O MAD é calculado através da diferença absoluta entre cada valor com a mediana da amostra (\tilde{x}), para então calcular a mediana dessas diferenças:

$$MAD = median\{|x_i - \tilde{x}|\} \quad (3)$$

Por sua vez, o método *Robust Z-score* é calculado a seguir, no qual a constante 0,6745 representa o 0,75 quartil da distribuição, para o qual o MAD converge.

$$M_i = \frac{0.6745(x_i - \tilde{x})}{MAD} \quad (4)$$

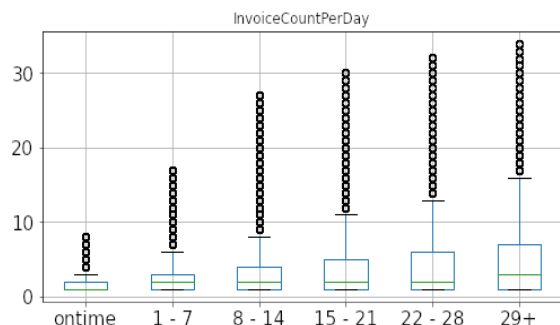
Além disso, como sugerido por Iglewicz e Hoaglin [19], o valor de corte adotado foi de 3,5. Isso significa que todo valor acima do ponto de corte é considerado um *outlier* e é então removido do conjunto.

A etapa 3, consiste na transformação dos dados, no qual foi realizado o agrupamento de faturas de mesmo cliente com mesma data de vencimento. Esse processo foi necessário, pois foi observado a existência de diversas faturas para um mesmo cliente em uma mesma data. De modo que se uma única fatura for paga em atraso, a cobrança ainda será feita. O Quadro 1 descreve os atributos criados nesta etapa.

Quadro 1: Atributos criados na etapa de agrupamento e suas respectivas descrições.

#	Atributo	Descrição
1	InvoiceCount	Número de faturas do cliente na mesma data de vencimento
2	OSInvoiceCount	Número de faturas pendentes do cliente na mesma data de vencimento
3	R_OSInvoiceCount	Proporção de faturas pendentes pelo total de faturas do cliente na mesma data de vencimento
4	InvoiceAmount	Soma dos valores das faturas do cliente na mesma data de vencimento
5	OSInvoiceAmount	Soma dos valores das faturas pendentes do cliente na mesma data de vencimento
6	R_OSInvoiceAmount	Proporção do valor das faturas pendentes pelo valor total das faturas do cliente na mesma data de vencimento

Através dos atributos do Quadro 1, foi possível analisar que a quantidade de faturas dos clientes, acumuladas por dia, está relacionada ao atraso do pagamento. A Figura 4 apresenta a distribuição da quantidade de faturas que os clientes possuem por dia, agrupadas em intervalos de dias.

**Figura 4:** Boxplot referente à quantidade de fatura dos clientes por dia, agrupadas em intervalos de dias.

A etapa 4 consiste na criação de novos atributos baseado nos dados atuais, no qual visa complementar os dados com o histórico do cliente na data de cada fatura [1][3][4]. Assim, criou-se três grupos para os dados históricos. O primeiro é referente à mediana de dias de atraso do cliente, tanto para faturas pagas, quanto para as pendentes. O Quadro 2 apresenta os atributos criados com a mediana de dias de atraso.

Quadro 2: Atributos criados com base na mediana histórica de dias de atraso do cliente.

#	Atributo	Descrição
7	MED_DaysLate	Mediana dos dias de atraso
8	MAD_DaysLate	Desvio absoluto dos dias de atraso
9	MED_DaysLateAM	Mediana dos dias de atraso, posterior ao mês de vencimento
10	MAD_DaysLateAM	Desvio absoluto dos dias de atraso, posterior ao mês de vencimento
11	MED OSDaysLate	Mediana dos dias de atraso das faturas pendentes
12	MAD OSDaysLate	Desvio absoluto dos dias de atraso das faturas pendentes
13	MED OSDaysLateAM	Mediana dos dias de atraso pendentes, posterior ao mês de vencimento
14	MAD OSDaysLateAM	Desvio absoluto dos dias de atraso pendentes, posterior ao mês de vencimento

O segundo grupo de atributos históricos, consiste na quantidade total de faturas pagas em relação à quantidade de faturas pagas com atraso. O Quadro 3 descreve os atributos criados em relação à quantidade de faturas pagas.

Quadro 3: Atributos com relação à quantidade de faturas pagas e pagas com atraso do cliente.

#	Atributo	Descrição
15	PaidCount	Total de faturas pagas do cliente
16	PaidLateCount	Total de faturas pagas com atraso do cliente
17	PaidLateAMCount	Total de faturas pagas com atraso, posterior ao mês de vencimento
18	R_PaidLateCount	Proporção de faturas pagas com atraso pelo total de faturas pagas
19	R_PaidLateAMCount	Proporção de faturas pagas com atraso posterior ao mês de vencimento, pelo total de faturas pagas
20	PaidAmount	Valor total das faturas pagas
21	PaidLateAmount	Valor total das faturas pagas com atraso
22	PaidLateAMAmount	Valor total das faturas pagas com atraso, posterior ao mês de vencimento
23	R_PaidLateAmount	Proporção do valor total das faturas pagas com atraso pelo valor total de faturas pagas
24	R_PaidLateAMAmount	Proporção do valor total das faturas pagas com atraso posterior ao mês de vencimento, pelo valor total de faturas pagas

O terceiro grupo de atributos históricos, é referente a quantidade de faturas pendentes e a quantidade de faturas pendentes com atraso. O

Quadro 4 mostra os atributos históricos criados em relação à quantidade de faturas pendentes.

Quadro 4: Atributos com relação à quantidade de faturas pendentes e pendentes com atraso do cliente.

#	Atributo	Descrição
25	OSCount	Total de faturas pendentes
26	OSLateCount	Total de faturas pendentes com atraso
27	OSLateAMCount	Total de faturas pendentes com atraso, posterior ao mês de vencimento
28	R_OSlateCount	Proporção de faturas pendentes com atraso pelo total de faturas pendentes
29	R_OSlateAMCount	Proporção de faturas pendentes com atraso posterior ao mês de vencimento, pelo total de faturas pendentes
30	OSAmount	Valor total das faturas pendentes
31	OSLateAmount	Valor total das faturas pendentes com atraso
32	OSLateAMAmount	Valor total das faturas pendentes com atraso, posterior ao mês de vencimento
33	R_OSlateAmount	Proporção do valor total das faturas pendentes com atraso pelo valor total de faturas pendentes
34	R_OSlateAMAmount	Proporção do valor total das faturas pendentes com atraso posterior ao mês de vencimento, pelo valor total de faturas pendentes

A etapa 5 tem como objetivo a criação de atributos relacionados com informações de tempo. Dessa forma, o Quadro 5 mostra os atributos criados a partir da data de criação e vencimento.

Quadro 5: Atributos criados com base nas datas de criação e vencimento de cada fatura.

#	Atributo	Descrição
35	DaysToDueDate	Quantidade de dias até a data de vencimento
36	DaysToEndMonth	Quantidade de dias até o fim do mês, a partir da data de vencimento
37	WeekdayEndMonth	Dia da semana do último dia do mês

Por fim, após as etapas do pré-processamento serem aplicadas, a base de dados passa a ter cerca de 580 mil registros e 37 atributos.

3.4 Metodologia Experimental

As etapas da mineração de dados podem variar segundo o modelo de estrutura definido [20]. No

entanto, o modelo comumente utilizado é o *Cross-Industry Standard Process of Data Mining* (CRISP-DM) [21][22], no qual possui cerca de seis fases bidirecionais e organizadas de forma cíclica, assim como mostra a Figura 5.

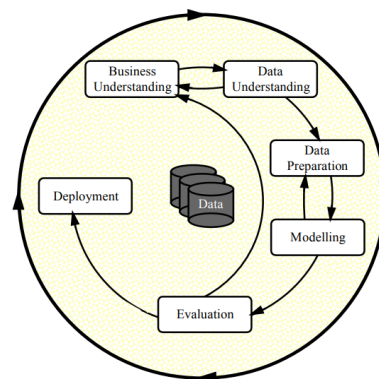


Figura 5: Fases do modelo de processo CRISP-DM para mineração de dados. Fonte: Wirth e Hipp (2000).

As fases do processo CRISP-DM são: (i) Entendimento do Negócio, no qual busca entender o objetivo do negócio que se deseja atingir; (ii) Entendimento dos Dados, visa conhecer os dados, identificando os mais relevantes para solucionar o problema; (iii) Preparação dos Dados, possui o objetivo de tratar e melhorar a qualidade dos dados, através de técnicas de pré-processamento; (iv) Modelagem, define as técnicas e algoritmos de aprendizado de máquina que serão aplicadas, de modo a alcançar o objetivo definido; (v) Avaliação, testa e valida o modelo desenvolvido, visando obter a confiabilidade do processo; e (vi) *Deployment*, no qual desenvolve e documenta um plano para a implantação do modelo, assim como disponibiliza os resultados alcançados.

Com isso, a realização do experimento, assim como as etapas de mineração de dados, foram aplicadas seguindo a metodologia CRISP-DM. De tal forma, foi possível definir quais etapas são necessárias para alcançar o objetivo do trabalho, que é de otimizar o processo *Invoice-to-Cash* no setor de cobrança. As três etapas são compostas por: (i) identificar as faturas que serão pagas no prazo ou com atraso; (ii) identificar se as faturas com atraso, serão pagas ainda no mês de vencimento ou não; e (iii) inferir o número de dias de atraso, para as faturas que serão atrasadas além do mês de vencimento. A Figura 6 sintetiza as etapas definidas.

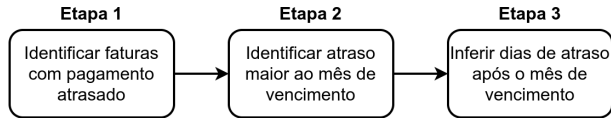


Figura 6: Etapas definidas para a realização do processo de modelagem.

Assim, foi utilizado algoritmos de aprendizado de máquina baseados em árvore de decisão, através da biblioteca Scikit-learn. A abordagem através de árvore de decisão foi escolhida por ser um modelo caixa branca, ou seja, a transparência da decisão do algoritmo, pode ser visualizada pelo usuário. Dessa forma, a implementação dos modelos de classificação e regressão se deu pelos algoritmos: (i) *Bagging*; (ii) *Balanced Bagging*; (iii) *Random Forest*; (iv) *Balanced Random Forest*; (v) *AdaBoost*; (vi) *Gradient Boosting*; (vii) *RUSBoost*; e (viii) *XGBoost*.

Para a execução e validação dos modelos, o conjunto de dados foi particionado em duas partes: (i) dados de teste, composto por toda fatura com data de vencimento em fevereiro de 2021; e (ii) dados de treino, composto por toda fatura com data de vencimento anterior a fevereiro de 2021.

Além disso, são adotadas subconjuntos de dados para cada etapa de acordo com o seu objetivo. Isso significa que, na etapa 1 são utilizados todos os dados para treino (570.270) e teste (13.123), para identificação das faturas pagas com atraso. Já na etapa 2, para identificar os atrasos além do mês de vencimento, são utilizadas as faturas atrasadas, que consiste em 227.048 dados para treino e 5.693 para teste. E por fim, para predição de dias de atraso na etapa 3, também são utilizadas as faturas atrasadas, porém, testadas apenas nas faturas com atraso posterior ao mês de vencimento, que consiste em 1.493. A Figura 7 mostra a distribuição dos dados nas partições de treino e teste.

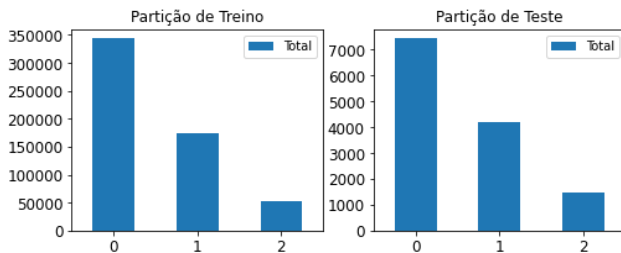


Figura 7: Distribuição dos dados nas partições de treino e teste, agrupados por: pagamento no prazo ("0"); atrasado e pago no mês de vencimento ("1"); atrasado e pago depois do mês de vencimento ("2").

Na fase de treinamento, foi aplicado o procedimento de validação cruzada *Repeated Stratified k-Fold* ($k=10$, $n=3$), no qual consiste em dividir o conjunto de dados em k partições, de forma aleatória e mantendo a proporção original das classes. Assim, cada partição k é utilizada como um conjunto de validação, enquanto todas as outras partições são usadas como um conjunto de treinamento. Esse processo é executado n vezes. Por fim, o desempenho médio do modelo é validado e então aplicado na partição de teste.

Além disso, antes do treinamento dos modelos nas etapas, foram utilizados as técnicas *Recursive Feature Elimination* (RFE) com *cross-validation* e *Grid Search* com *cross-validation*, para selecionar os melhores atributos e as melhores combinações de parâmetros, respectivamente. Para análise dos modelos nas etapas 1 e 2, a métrica *f1-score* com média *macro* foi utilizada [23]. Essa métrica consiste na média de *f1-score*, que é definido por:

$$f1\text{-macro} = \frac{1}{m} \sum_{i=1}^m f1\text{-score}_i \quad (5)$$

O *f1-score*, por sua vez, é a média harmônica entre as métricas de *precision* e *recall*. Essas três métricas fazem parte da matriz de confusão e são definidas nas equações 6, 7 e 8, respectivamente:

$$f1\text{-score} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

$$\text{precision} = \frac{TP}{TP + FP} \quad (7)$$

$$\text{recall} = \frac{TP}{TP + FN} \quad (8)$$

sendo que TP (*True Positive*) representa a classe de interesse classificada corretamente; FP (*False Positive*), a classe de interesse classificada incorretamente; TN (*True Negative*), a classe de não interesse classificada corretamente; e FN (*False Negative*), a classe de não interesse classificada incorretamente.

Para análise dos modelos na etapa 3, foi utilizada a métrica *Root Mean Squared Error* (RMSE) [24]. Essa métrica consiste na raiz quadrática média dos erros entre os valores reais e as predições referente aos dias de atraso. O RMSE é definido a seguir:

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \tilde{y}_j)^2} \quad (9)$$

no qual, y consiste nos valores reais; \tilde{y} nos valores previstos; e n no número de elementos. Além disso, a métrica RMSE traz o benefício de penalizar erros de maior magnitude, o que pode auxiliar na análise dos dias de atraso previstos.

Por fim, também é analisada a acurácia do modelo referente ao número de dias de atraso posterior ao mês de vencimento. Neste contexto, é considerado as predições de dias, somado a uma tolerância de ± 0 até ± 3 dias. A Figura 8 apresenta a visão geral da metodologia realizada para o experimento.

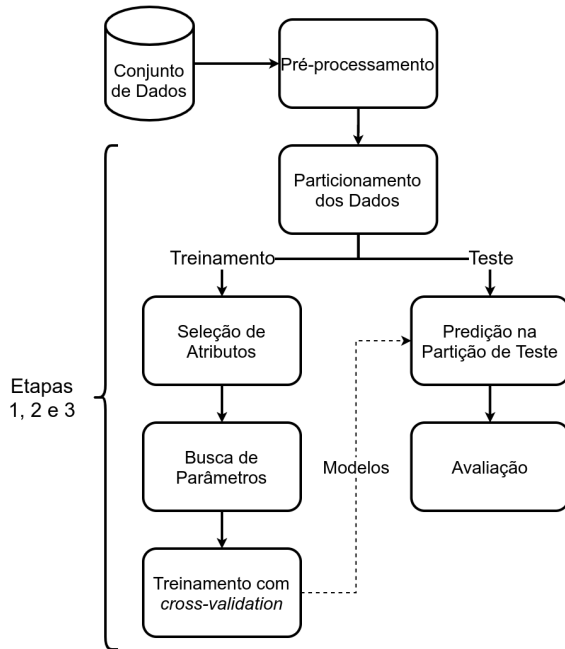


Figura 8: Visão geral da metodologia realizada para o experimento.

4 Análise e Discussão dos Resultados

Nesta seção, os experimentos realizados são apresentados e os resultados obtidos em cada etapa são detalhados nas subseções seguintes. Por fim, os pontos mais relevantes são discutidos.

4.1 Identificação de Pagamentos no Prazo e com Atraso

Na etapa 1, o objetivo consiste em distinguir as faturas com pagamentos no prazo e com atraso.

Assim, além de modelos tradicionais de classificação, também foi utilizado modelos focados em lidar com o desbalanceamento de classes (*Balanced*).

Inicialmente, através do RFE, foi possível remover 3 atributos nesta etapa e ainda manter bons resultados entre os modelos. Estes atributos estavam relacionados à quantidade de faturas pagas e pendentes no dia. Além disso, o *Grid Search* proporcionou as melhores combinações de parâmetros. Assim, foi utilizado o ganho de informação, calculado pela função de qualidade da árvore e seus atributos (entropia). Também foi utilizado, na construção da maioria dos modelos, o valor de 512 estimadores. Para as modelos *Balanced*, a melhor estratégia de reamostragem foi de 70% da classe minoritária sobre a classe majoritária. Nos demais modelos, foi adotado peso de classe balanceado.

Assim, como pode ser observado na Tabela 2, os modelos *XGBoost* e *Random Forest* tiveram os melhores resultados, com *f1-score* de 81,85% e 81,76%, respectivamente. Já o *AdaBoost* obteve o pior resultado, com 79,51% em *f1-score*.

Tabela 2: Resultados obtidos na partição de teste (13.123 dados), com classificação entre faturas pagas no prazo e com atraso.

Modelo	Precision	Recall	F1-score
AdaBoost	79,73%	79,37%	79,51%
Bagging	81,35%	81,53%	81,43%
Balanced Bagging	81,40%	81,76%	81,52%
Balanced Random Forest	81,59%	81,97%	81,71%
Gradient Boosting	81,49%	81,01%	81,20%
Random Forest	81,64%	82,00%	81,76%
RUSBoost	80,03%	80,30%	80,13%
XGBoost	81,75%	81,99%	81,85%

Através dos resultados obtidos, também foi analisada a quantidade dos dados identificados e identificados corretamente. O modelo *XGBoost*, por exemplo, obteve por volta de 82,09% no reconhecimento dos dados, e cerca de 81,20% no reconhecimento de faturas atrasadas. A Figura 8 apresenta a sua matriz de confusão do *XGBoost* na partição de teste, conforme os 13.123 registros.

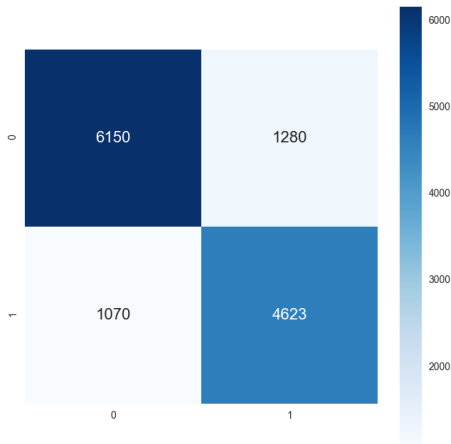


Figura 8: Matriz de confusão do *XGBoost*, no qual "0" são faturas pagas no prazo (7.430 dados); e "1" são faturas pagas com atraso (5.693 dados).

Por fim, vale mencionar o desempenho dos modelos ao longo do mês de teste, especialmente para os dias após os finais de semana, que oferecem maior precisão. A Figura 9 apresenta o desempenho do modelo *XGBoost* ao longo do mês testado.

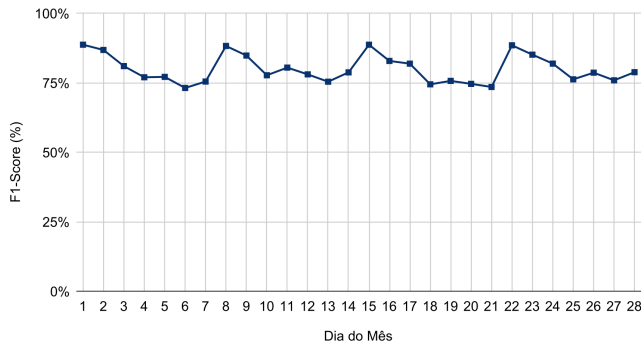


Figura 9: Desempenho do modelo *XGBoost* na etapa 1, ao longo do mês de fevereiro de 2021.

4.2 Identificação de Pagamentos Atrasados Dentro e Além do Mês de Vencimento

Para a etapa 2, é considerado apenas as faturas em atraso para treinamento e teste. Neste cenário, os modelos visam identificar os pagamentos atrasados no mês de vencimento ou posterior.

Como na etapa anterior, o RFE resultou na exclusão de 4 atributos, também relacionados à quantidade de faturas pagas e pendentes no dia. Os parâmetros obtidos pelo *Grid Search*, por sua vez, 11

variaram no número de estimadores utilizados na construção, agora 256, e no valor da reamostragem (métodos *Balanced*), agora com 75% da classe minoritária sobre a majoritária.

Os resultados alcançados pelos modelos foram bem diversos. O *AdaBoost* obteve o melhor resultado em *precision*, com 89,70%. Já no *recall* e *f1-score*, o *Balanced Random Forest* alcançou 85,36% e 85,63%, respectivamente. Como pior resultado, o *RUSBoost* obteve 81,43%. A Tabela 3 apresenta os melhores resultados alcançados por cada modelo.

Tabela 3: Resultados obtidos na partição de teste (5.693 dados), com classificação entre faturas atrasadas no mês de vencimento e posterior.

Modelo	Precision	Recall	F1-score
AdaBoost	89,70%	79,90%	83,22%
Bagging	89,16%	81,42%	84,27%
Balanced Bagging	86,36%	84,93%	85,63%
Balanced Random Forest	85,92%	85,36%	85,63%
Gradient Boosting	87,99%	81,05%	83,66%
Random Forest	87,67%	83,23%	85,09%
RUSBoost	84,65%	79,35%	81,43%
XGBoost	85,90%	81,49%	83,31%

Diante dos resultados dos modelos, o *Balanced Random Forest* obteve cerca de 88,96% no reconhecimento geral dos dados, e cerca de 77,76% no reconhecimento de faturas atrasadas além do mês de vencimento. A Figura 10 apresenta a sua matriz de confusão na partição de teste, conforme os 5.693 registros.

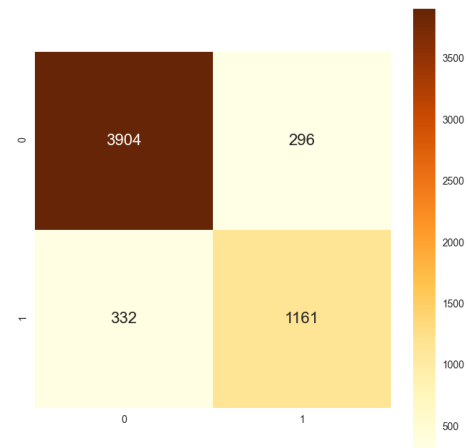


Figura 10: Matriz de confusão do *Balanced Random Forest*, sendo "0" faturas atrasadas pagas no mês de vencimento (4.200 dados); e "1", faturas atrasadas pagas além do mês de vencimento (1.493 dados).

DOI: 10.xxxx/s11468-014-9759-3

Com o objetivo de predição diferente da etapa anterior, os modelos alcançaram uma precisão mediana ao longo do mês testado. Além disso, teve seu pico de desempenho apenas no final do mês, o que coincidentemente corresponde a um final de semana. A Figura 11 apresenta o desempenho do modelo *Balanced Random Forest* ao longo do mês.

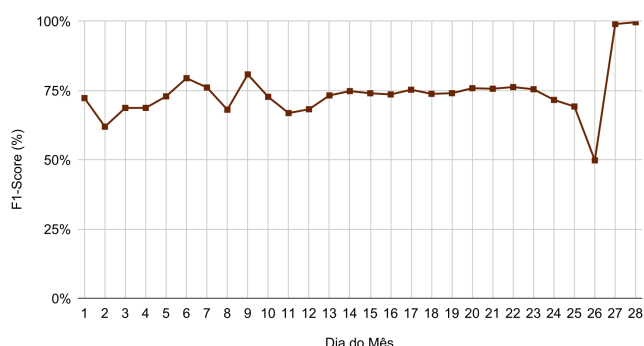


Figura 11: Desempenho do modelo *Balanced Random Forest* na etapa 2, ao longo do mês de fevereiro de 2021.

4.3 Predição dos Dias de Atrasado Além do Mês de Vencimento

Na etapa 3, as faturas atrasadas também são utilizadas para treinamento, sendo a partição de teste composta pelas faturas com atraso além do mês de vencimento.

Diferente das etapas anteriores, através do RFE foi possível excluir cerca de 10 atributos e ainda manter bons resultados. Estes atributos removidos eram relacionados a data e histórico de faturas pagas e pendentes. Além disso, como parâmetros obtidos pelo *Grid Search*, os modelos em geral alcançaram bons resultados com 512 estimadores na construção, $1e-4$ como a fração mínima, da soma dos pesos, necessária para estar em uma folha, e o valor mínimo de 4 amostras para dividir um nó interno.

Nesse contexto, o modelo *Random Forest* obteve o menor valor de RMSE, com cerca de 3,5792 (quanto menor melhor). No entanto, quanto aos resultados de acurácia das predições, considerando os valores de tolerância (± 0 até ± 3 dias), o *AdaBoost* obteve os melhores resultados, indo de 60,23% (sem variação) até 84,16% (com 3 dias de variação). A Tabela 4 sintetiza os resultados obtidos em cada modelo.

Tabela 4: Resultados obtidos na partição de teste (1.493 dados), com a predição do número de dias de atraso posterior ao mês de vencimento.

Modelo	RMSE	Ac. ± 0	Ac. ± 1	Ac. ± 2	Ac. ± 3
AdaBoost	3,6164	60,23%	71,91%	79,62%	84,16%
Bagging	3,5786	54,24%	70,12%	78,46%	84,12%
Gradient Boosting	6,1018	47,53%	61,04%	69,31%	73,70%
Random Forest	3,5792	54,29%	70,10%	78,41%	84,16%
XGBoost	4,7917	53,86%	61,08%	66,43%	71,47%

Para este contexto, a acurácia ao longo do mês testado também foi analisada. Dessa forma, é possível visualizar que a acurácia tende a cair conforme o fim do mês se aproxima. A Figura 12 apresenta os resultados obtidos pelo modelo *AdaBoost* ao longo do mês de fevereiro de 2021.

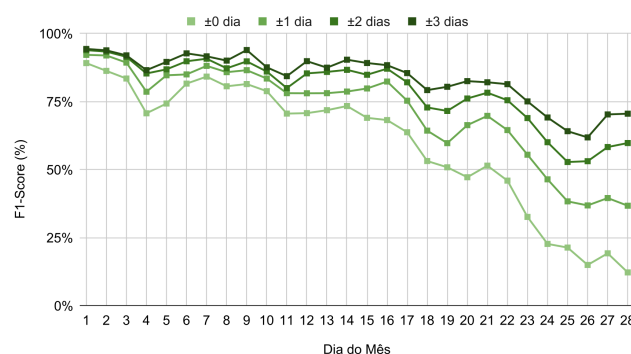


Figura 12: Desempenho do modelo *AdaBoost* na etapa 3, ao longo do mês de fevereiro de 2021 e através da métrica da Acurácia.

4.4 Discussões

Os resultados alcançados por cada modelo e em cada etapa, de certo modo, possibilita o suporte à tomada de decisão na coleta de contas a receber. Comparado aos trabalhos relacionados, a abordagem da modelagem, definida em 3 etapas em vez de 1, traz melhor detalhamento ao processo de decisão.

Nesse contexto, as 3 etapas em conjunto oferecem 81,85% na identificação de faturas com pagamento atrasado, 85,63% na identificação desse pagamento atrasado ser após o mês de vencimento, e por fim, de 60,23% até 84,16% na acurácia de predição desses dias de atraso. Assim, os modelos *XGBoost*, *Balanced Random Forest* e *AdaBoost* compõem cada etapa do processo definido, respectivamente.

Em geral, os modelos utilizaram o ganho de informação através da entropia como medida de qualidade para as divisões dos nós das árvores. Dessa forma, foi possível lidar melhor com os dados desbalanceados e alcançar resultados ligeiramente melhores, quando comparado a utilização do índice de impureza (Gini).

Vale mencionar os diferentes desempenhos dos modelos, conforme o objetivo de cada etapa. Dessa forma, para a etapa 1, foi observado o aumento da precisão no início da semana, seguido pelo seu decaimento. Correspondendo ao intervalo adotado para os atributos históricos. Para a etapa 2, a alta precisão veio no fim do mês, que coincidentemente é foi um final de semana. O que implica novamente o atraso natural da fatura, agora após o mês de vencimento. Por fim, a acurácia da etapa 3, decaiu para todos os modelos, conforme passam os dias do mês testado. Isso indica maior variação dos dias de atraso conforme o fim do mês se aproxima. Como melhor solução, a utilização de intervalos de dias para esta última etapa pode trazer melhores resultados.

5 Conclusões e Trabalhos Futuros

Tendo em vista a otimização no processo de coleta de contas a receber pelo setor financeiro de uma empresa, este trabalho apresenta uma abordagem de predição de pagamentos atrasados através de algoritmos baseados em árvore de decisão. De tal forma, o processo de identificação foi modelado em 3 etapas: (i) faturas pagas com atraso; (ii) faturas pagas com atraso além do mês de vencimento; e (iii) número de dias de atraso além do mês de vencimento.

Para alcançar o objetivo proposto, foi realizado um experimento através da metodologia CRISP-DM, no qual é considerado desde a etapa de coleta de dados até a avaliação dos modelos de aprendizado de máquina. Como proposta, foi utilizada técnicas de pré-processamento para remoção de *outliers* e criação de atributos baseado em histórico, assim como 8 algoritmos de árvore de decisão parametrizados para cada etapa definida.

Através dos resultados obtidos de cada modelo, foi obtido uma precisão média de 80% no decorrer de todo o processo. Em outras palavras, os modelos de melhor desempenho conseguiram identificar cerca de 80% das faturas atrasadas, faturas atrasadas com

pagamento além do mês de vencimento, e por fim a quantidade de dias de atraso. Dessa forma, o trabalho proposto possibilita o auxílio no processo de coleta de contas a receber, oferecendo até 3 etapas de detalhamento para a decisão.

Como trabalhos futuros, o refino das três etapas será aplicado. Ou seja, uma melhor seleção de atributos, busca por parâmetros e até modelos mais robustos, como redes neurais, podem incrementar nos resultados do experimento. Além disso, a predição dos dias de atraso (etapa 3), será modelada através de intervalos de dias (*buckets*).

Referências

- [1] HU, P. Predicting and improving invoice-to-cash collection through machine learning. Thesis: S.M., Massachusetts Institute of Technology, Department of Electrical Engineering and Computer Science, 2015.
- [2] MURPHY, K. P. Machine Learning: A Probabilistic Perspective. Cambridge, MA: MIT Press, 2ª edição, pág. 1-27, 2012.
- [3] ZENG, S.; MELVILLE, P.; LANG, C.A.; BOIER-MARTIN, I.M.; MURPHY, C. Using predictive analysis to improve invoice-to-cash collection. Proceedings of the 14th international conference on Knowledge Discovery and Data mining (KDD'08), 2008.
- [4] NANDA, S. Proactive Collections Management: Using Artificial Intelligence to Predict Invoice Payment Dates. The Credit & Financial Management Review. Credit Research Foundation, 2018.
- [5] YOUNES, B.; BOUFERGUÈNE, A.; AL-HUSSEIN, M.; YU, H. Overdue Invoice Management: Markov Chain Approach. Journal of Construction Engineering and Management, 2015.
- [6] KOCH, B. The e-invoicing journey 2019-2025. E-Invoicing/E-Billing Significant Market Transition Lies Ahead, 2019. Disponível em: <https://compacer.com/wp-content/uploads/2019/05/Billentis_Report_compacer_single-sponsor.pdf>. Acesso em: 18 março, 2021.

- [7] GUNASEKARAN, A.; PATEL, C.; TIRTIROGLU, E. Performance measures and metrics in a supply chain environment. *International journal of operations & production Management*, vol. 21, pág. 71-87, 2001.
- [8] KOUVELIS, P.; ZHAO, W. Supply chain finance. *The Handbook of Integrated Risk Management in Global Supply Chains*, pág. 247-288, 2011.
- [9] JUNIOR, R.; MELO, W. M.; FAGUNDES, R. A.; MACIEL, A. A. Extração de Informação e Mineração de Dados no Diário Oficial de Pernambuco. *Revista de Engenharia e Pesquisa Aplicada*, 2018.
- [10] FAYYAD, U.; PIATETSKY-SHAPIO, G.; SMYTH, P. Knowledge discovery and data mining: Towards a unifying framework. *Proceedings of the 2nd international conference on Knowledge Discovery and Data mining (KDD'96)*, pág. 82-88, 1996.
- [11] SHAH, H. Customer Payment Prediction in Account Receivable. *International Journal of Science and Research (IJSR)*, vol. 8, ed. 1, 2019.
- [12] BANFIELD, R. E.; HALL, L. O.; BOWYER, K. W.; KEGELMEYER, W. P. A Comparison of Decision Tree Ensemble Creation Techniques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, n. 1, 2007.
- [13] BREIMAN, L. Bagging Predictors. *Machine Learning, Statistics Department, University of California, Berkeley*, vol. 24, pág. 123-140, 1996.
- [14] BREIMAN, L. Random Forests. *Machine Learning, Statistics Department, University of California, Berkeley*, vol. 45, pág. 5-32, 2001.
- [15] FREUND, Y.; SCHAPIRE, R. Experiments with a New Boosting Algorithm. *Proceedings of the 13th International Conference on International Conference on Machine Learning (ICML'96)*, pág. 148-156, 1996.
- [16] FRIEDMAN J. Greedy Function Approximation: A Gradient Boosting Machine, *The Annals of Statistics*, vol. 29, n. 5, 2001.
- [17] SEIFFERT, C.; KHOSHGOFTAAR, T. M.; VAN HULSE, J.; NAPOLITANO, A. RUSBoost: A Hybrid Approach to Alleviating Class Imbalance. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, n. 1, pág. 185-197, 2010.
- [18] CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'16)*, pág. 785-794, 2016.
- [19] IGLEWICZ, B.; HOAGLIN, D. Volume 16: How to Detect and Handle Outliers. *The ASQC Basic References in Quality Control: Statistical Techniques*. Edward F. Mykytka, 1993.
- [20] SANTOS, I. H. S.; MACIEL, A. M. A. Desenvolvimento de um Sistema de Apoio à Decisão para priorização de Pedidos de Desembolso no Estado de Pernambuco. *Revista de Engenharia e Pesquisa Aplicada*, 2018.
- [21] WIRTH, R.; HIPPE, J. CRISP-DM: Towards a Standard Process Model for Data Mining. *Proceedings of the Fourth International Conference on the Practical Application of Knowledge Discovery and Data Mining*, 2000.
- [22] SÁ, J. M. C.; RODRIGUES SILVA, I. R.; GOMES DA SILVA, R.; SOUTO, L. G. A.; SANTOS SILVA, P. G. Análise de Crédito Utilizando uma Abordagem de Mineração de Dados. *Revista de Engenharia e Pesquisa Aplicada*, 2018.
- [23] LIPTON, Z. C.; ELKAN, C.; NARYANASWAMY, B. Optimal thresholding of classifiers to maximize F1 measure. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pág. 225-239, 2014.
- [24] CHAI, T.; DRAXLER, R.R. Root Mean Square Error (RMSE) or Mean Absolute Error (MAE)? *Geoscientific Model Development Discussions*, vol. 7, pág. 1247-1250, 2014.