Machine learning aplicado no problema de perdas com créditos de uma distribuidora de energia elétrica

Jelson André Cordeiro * **Marcelo de Oliveira Rosa** **

* *Depto. Acadêmico de Informática, Universidade Tecnológica Federal do Paraná, Curitiba, PR (e-mail: jelsoncordeiro@gmail.com*
** *Depto. Acadêmico de Eletrotécnica, Universidade Tecnológica Federal do Paraná, Curitiba, PR (e-mail: mrosa@utfpr.edu.br)*

Abstract: Estimated Losses on Doubtful Accounts (ELDA) in companies is an attractive field for investigation due to the percentage of profits it represents. The objective of this work is to find a machine learning model to predict on which day the customer will pay the energy bill in order to maximize the company's profit. To evaluate the proposed methodology, experiments were carried out using real data from customer invoices. The results of the models were compared with each other and statistical analysis was performed to verify if there is a significant difference between them. The results achieved indicate that the application of the proposed modeling is promising.

Resumo: As Perdas Estimadas em Créditos de Liquidação Duvidosa (PECLD) nas empresas é um campo atraente para investigação devido ao percentual dos lucros que representa. O objetivo deste trabalho é encontrar um modelo de aprendizagem de máquina para predizer em que dia o cliente irá pagar a fatura visando maximizar o lucro da empresa. Para avaliar a metodologia proposta foram realizados experimentos utilizando dados reais de faturas dos clientes. Os resultados dos modelos foram comparados entre si e realizado a análise estatística para verificar se existe diferença significativa entre eles. Os resultados alcançados indicam que é promissora a aplicação da modelagem proposta.

Keywords: Supervised machine learning; linear regression; credit estimated losses

Palavras-chaves: Aprendizagem de máquina supervisionado; regressão linear; perdas estimadas em créditos

1. INTRODUÇÃO

Todo planejamento financeiro de uma empresa é baseado nos créditos e débitos. Desse cenário, surge a necessidade de se realizar a chamada Perdas Estimadas em Créditos de Liquidação Duvidosa (PECLD) (Padoveze (1996)).

Medidas como a PECLD são fundamentais para a empresa não se endividar por falta de precaução. A PECLD se refere a uma reserva de dinheiro feita pela empresa com foco em casos de inadimplência: quanto maior for o risco de o cliente não pagar o que deve, maior deve ser o montante guardado pela empresa através da PECLD.

Nas empresas de Distribuição de Energia Elétrica no Brasil e no mundo, o problema de PECLD representa um valor significativo no lucro. Em 2019, a PECLD de umas das maiores empresas deste setor foi de R$ 137,75 milhões que representou 20,87% do lucro daquele período [1] . Além disso, junto com a fatura de energia também são gerados impostos e tributos, que são repassados para os entes tributantes, seja ela arrecadada ou não. Se o consumidor não quita a fatura, a empresa precisa pagar os impostos do seu próprio bolso.

[1] `http://sistemas.cvm.gov.br`

Outro custo que a empresa possui quando o consumidor não quita a fatura é o custo de procedimento de desligamento da unidade consumidora. Após a geração da fatura e caso o consumidor não tenha quitado dentro do prazo limite, é iniciada uma série de procedimentos como envio de e-mails/SMS, contato telefônico, inclusão no Serviço de Proteção ao Crédito (SPC), desligamento, cobrança personalizada e cobrança terceirizada. Além destes custos, a escolha de qual cliente cortar primeiro é importante porque os recursos que a empresa tem (equipe em campo disponível, por exemplo) são limitados.

Portanto, para resolver este tipo de problema das Distribuidoras de Energia Elétrica o objetivo geral deste e de outros trabalhos é buscar um modelo de *machine learning* (ML) capaz de predizer o dia em que o cliente pagará a fatura visando maximizar o lucro da empresa.

1.1 Revisão da literatura

Pandey and Srinivasan (2011) utilizaram o aprendizado supervisionado para estimar se o cliente irá pagar ou não a fatura em uma base de 150.000 registros. *Support Vector Machine* (SVM) e *Random Forest* foram utilizados para obter um desempenho de 86% de *score*. Uma validação cruzada foi usada para encontrar os melhores hiperparâ-

metros. Bastos (2010) utilizou *Random Forest* para prever perda de crédito no setor bancário, na qual a avaliação de desempenho foi dividida em horizontes de 12, 24, 36 e 48 meses. Yazdi et al. (2020) usaram SVM para identificação de *Fake news*. Pahwa and Agarwal (2019) usaram SVM para análise do mercado de ações e Lundberg et al. (2018) usaram o XGBoost, Lasso e SVM para prevenir hipoxemia durante uma cirurgia.

Além dos algoritmos citados, entender como um modelo estatístico fez uma previsão específica é um desafio importante no aprendizado de máquina. No entanto, muitos modelos complexos com excelente precisão, fazem previsões que especialistas têm de interpretar. Isso requer uma compensação entre precisão e interpretação. Entender como o modelo de chegou a determinado resultado traz diversos benefícios: Primeiro a necessidade de transparência dos modelos nas organizações cresceu na medida em que usam grandes volumes de informações pessoais e complexas (Datta et al. (2016)). É essencial para empresa identificar possíveis discriminações, por exemplo, introduzidas pela tomada de decisão do modelo de ML (porque o modelo chegou a conclusão que determinado cliente não vai pagar a conta e o outro vai). Em segundo lugar, a transparência pode ajudar detectar erros nos dados entrada que resultaram em uma decisão errada pelo modelo.

O algoritmo *Shapley Additive exPlanations* (SHAP) (Lundberg and Lee (2017a); Lipovetsky and Conklin (2001)) calcula o impacto de cada atributo no modelo. É uma abordagem para explicar a saída de qualquer modelo de aprendizado de máquina que conecta a alocação de crédito ideal com explicações locais usando os valores clássicos de Shapley da teoria dos jogos e suas extensões relacionadas. Este algoritmo utiliza a média das contribuições marginais em todas as permutações. Lundberg et al. (2018) utilizaram-no para explicar para médicos o resultado do modelo encontrado para prever hipoxemia durante uma cirurgia.

Outra abordagem é o *Quantitative Input Influence* (QII) (Datta et al. (2016)), que avalia as correlações entre as entradas para permitir o raciocínio causal e calcula a influência marginal de entradas em situações na qual as entradas não podem afetam apenas os resultados. *DeepLIFT* (Shrikumar et al. (2017)) é um algoritmo de interpretação para modelos de *deep learning* que decompõe a previsão de saída de uma rede neural propagando as contribuições de todos os neurônios da rede para cada recurso da entrada, tendo sido usado na simulação de genoma. E por fim, o *Local Interpretable Model-agnostic Explanations* (LIME) (Ribeiro et al. (2016)) é um algoritmo para explicar as previsões de qualquer modelo em uma forma interpretável.

2. MATERIAL E MÉTODOS

Na resolução proposta, alguns aspectos devem ser definidos. Os dados utilizados (ou dados de origem) são apresentados na seção (2.1). Na seção (2.2) são definidos os algoritmos utilizados e seus respectivos hiperparâmetros. A metodologia de avaliação é apresentada na seção (2.3) e na seção (2.4) como a comparação entre modelos foi realizada.

O trabalho seguiu o processo de mineração de dados CRISP-DM de Provost et al. (2018), que exigiu sucessivas preparações de dados antes da modelagem efetiva.

2.1 Conjunto de dados

Foram usados dados históricos reais de faturas de clientes de uma Distribuidora de energia elétrica. Para satisfazer a Lei Geral de Proteção de Dados (LGPD) atributos que pudessem identificar o cliente foram retirados. O *dataset* inicial continha 220.218 faturas com 9418 clientes distintos e cada um deles com 23 faturas em média. Este *dataset* possui 40 atributos preditores e um atributo alvo ("dias em atraso"). O atributo alvo estava balanceado, com 59% de faturas com atraso.

Dentre os atributos preditores destacam-se alguns com maior relevância como o valor da fatura, se o cliente esta negativado no Serasa, valor total de débito que o cliente possui e quantidade de faturas em aberto para o mesmo cliente. Idade, sexo, classe, se é cliente pessoa física ou jurídica, indicativo de cliente VIP, indicativo de cliente baixa renda, dias que o cliente retornou contato após SMS e cortes gerados também merecem destaques. Outros atributos menos relevantes utilizados foram: tipo da ligação (monofásica, bifásica ou trifásica), tensão e localidade. Os atributos tensão, localidade, sexo, tipo da pessoa e classe foram recodificados usando o procedimento *OneHotEncoder* (Lakshmanan et al. (2020)).

Um total de 176.174 faturas (80% dos dados) foram reservadas para treinamento e 44.044 faturas (20% dos dados) para o teste dos modelos.

2.2 Algoritmos

Considerando o tipo de variável alvo (quantidade de dias até o consumidor pagar sua fatura), buscou-se modelos de regressão (Russell and Norvig (2004)).

Além da regressão linear simples, foram utilizados o Lasso e Ridge (Tan et al. (2009)). O Lasso utiliza o hiperparâmetro L1 para penalizar os coeficientes do modelo, reduzindo-os para zero, e a regressão Ridge utiliza o hiperparâmetro L2 para amortecê-los sem força-los para zero. A diferença básica entre os dois algoritmos é que o Lasso pode retirar do modelo atributos preditores insignificantes enquanto o Ridge mantém todos os atributos já que os coeficientes nunca chegam a zero.

O algoritmo de *Random Forest* (RF) (Breiman et al. (1984); Quinlan (1986)) também foi utilizado com o intuito de garantir uma generalização maior do modelo gerado e evitar o *overfitting*. Os métodos SVM (Ray (2019)) e redes neurais artificiais (RNAs) também produziram modelos a partir dos dados de entrada e de saída - usou-se RNAs com 1 camada oculta contendo 8 neurônios.

O último algoritmo utilizado é o *XBBoost* (Chen and Guestrin (2016)). Um algoritmo recente que utiliza técnicas de *boosting* para tentar ajustar os dados de forma adaptativa. Ele representa uma categoria de algoritmos baseada em árvores de decisão com aumento de gradiente. Aumento de gradiente significa que o algoritmo usa o algoritmo *Gradient Descent* para minimizar a perda. O *XBBoost* repetidamente cria novos modelos e os combina

Tabela 1. Hiperparâmetros a serem otimizados para cada algoritmo deste trabalho

Algoritmos	Hiperparâmetros	Valores testados
XGBoost	booster η $nEstimators$	gbtree, gblinear e dart. 0; 0,3; 0,6; 1 10; 100; 400
Random Forest	$nEstimators$ $maxFeatures$ $minSamplesLeaf$	10; 100; 200 auto, sqrt e log2 1; 5; 10; 100
Ridge	L2	0,0001; 0,001; 0,01; 0,1; 1, 5, 10
Lasso	L1	0,0001; 0,001; 0,01; 0,1; 1, 5, 10
SVM	$loss$	epsilonInsensitiv e squaredEpsilonInsensitive
RNA	$optimizer$ $activation$	Adam e RMSprop ReLU e Softmax
Linear	-	-

um modelo único. A cada ciclo ele constrói o modelo, adiciona-o no modelo agrupado e calcula o erro.

Para otimizar os resultados (e descobrir os melhores hiperparâmetros de cada algoritmo), a técnica de validação cruzada foi empregada, com 5-*fold*. A Tabela 1 mostra o intervalo de valores que cada hiperparâmetro de cada algoritmo foi variado, que foram escolhidos arbitrariamente para viabilizar os estudos (é possível que um ajuste mais refinado hiperparâmetros leve a melhores resultados, mas isso foge do escopo do trabalho).

A variável alvo também sofreu modificações, impondo-se valores ditos de corte: 90, 180, 365 e ∞ (isto é, sem modificação). Por exemplo, para um corte de 90 dias, se a variável alvo "dias em atraso" for superior a 90 dias, ela é alterada para 90 (o mesmo procedimento é feito para 180 e 365 dias - para ∞, a variável alvo não é alterada). Tal modificação objetivou reduzir a influência de *outliers*, considerando a baixa frequência de casos para valores elevados de "dias de atraso". O procedimento pode ser entendido como uma truncagem da variável alvo.

Dado o número de atributos, aplicou-se a técnica de redução de dimensionalidade denominada *Principal Component Analysis* (PCA), reduzindo um espaço de entrada de 40 para 18 atributos, com 98% de cobertura. A alteração no custo computacional com a redução do espaço de entrada não foi avaliado neste trabalho.

O procedimento descrito abaixo foi usado para automatizar os processos de treinamento, validação cruzada e manipulações nos atributos. Nesse procedimento, a variável alvo sofreu cortes de 90 ($corte = 0$), 180 ($corte = 1$) e 365 ($corte = 2$) dias, além modelagens sem corte ($corte = 3$). As condições $pca = 1$ e $pca = 0$ respectivamente significam aplicação e não aplicação de PCA. Reforça-se que os treinamentos foram realizados com os modelos de melhores hiperparâmetros.

Os ensaios foram feitos um computador I9 G10 de 20 núcleos e 32GB de RAM. Para a implementação dos algoritmos foi utilizada a biblioteca *open-source* scikit-learn [2], com exceção do algoritmo de RNA, oriundo do pacote Keras [3].

Algoritmo 1 Automação para obtenção dos modelos

```
for corte = 0 até 3 do
    for pca = 0 até 1 do
        for algoritmo = 0 até 6 do
            Treinar modelo com validação cruzada
            Testar modelo e preparar resultados
        end for
    end for
end for
```

2.3 Metodologia de Avaliação

Para medir o desempenho, além do erro médio quadrático (*Root Mean Squared Error* ou RMSE), foram usadas as métricas erro médio absoluto (*Mean Absolute Error* ou MAE) e o R^2, que é uma métrica padrão para problemas de regressão. Além disso, por fim o R^2 ajustado que penaliza o modelo de acordo com o número de atributos usados (Equação (1)).

$$AdjR^2 = 1 - \frac{(1 - R^2) * (n - 1)}{n - p - 1} \quad (1)$$

na qual p é a quantidade de atributos e n é a quantidade de observações.

Os modelos treinados podem produzir valores negativos para a variável alvo "dias em atraso". Quando isto ocorreu, eles foram ajustados/truncados para zero porque no mundo real o valor negativo representaria o cliente pagando a fatura antes do vencimento.

2.4 Análise Estatística

Os testes estatísticos de Friedman e Nemenyi Granatyr (2018) foram realizados para determinar se existe diferença estatística entre os modelos.

Os experimentos foram executados 30 vezes para cada algoritmo com sementes aleatórias diferentes e análises estatísticas foram executadas no software R [4] com a biblioteca *open-source* TStools [5].

3. RESULTADOS

A Tabela 2 mostra o resultado da otimização dos hiperparâmetros para os algoritmos. Com os modelos otimizados, os resultados dos testes são apresentados nas Tabelas 3 e 4, para modelos com corte de 90 dias (com e sem aplicação de PCA), nas Tabelas 5 e 6, para cortes de 180 dias (com e sem PCA), nas Tabelas 7 e 8, para cortes de 365 dias (com e sem PCA), e nas Tabelas 9 e 10, para modelos sem corte (com e sem PCA). Em todas essas tabelas, os melhores resultados aparecem destacados em negrito.

As funções densidade de probabilidade dos erros de estimativa dos modelos para cada dia de atraso de pagamento de fatura são apresentadas nas Figuras 1, 2, 3, 4, 5, 6 e 7 (a área dos "violinos" representa 100% das respectivas amostras).

As Figuras 8, 9, 10, 11, 12 e 13 mostram o gráfico *boxplot* para visualizar a distribuição da estimativa da

[2] http://scikit-learn.org
[3] https://keras.io/
[4] https://www.r-project.org/
[5] https://github.com/trnnick/TStools

Tabela 2. Valores otimizados para os hiperparâmetros de cada algoritmo deste trabalho

Algoritmo	*Hyperparameters*
XGBoost	*booster*=gbtree, η=0.3 e $nEstimators$=400
Random Forest	$maxFeatures$=auto, $minSamplesLeaf$=5 e $nEstimators$=200
Ridge	L2=10
Lasso	L1=0,01
SVM	*loss*=epsilonInsensitive
RNA	*optimizer*=Adam e *activation*=ReLU
Linear	-

Tabela 3. Resultado dos modelos com corte de 90 dias sem PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,559	0,558	12,609	311,636	17,653
SVM	0,507	0,507	10,959	348,006	18,655
RNA	0,643	0,643	10,589	251,737	15,866
Lasso	0,558	0,558	12,614	311,743	17,656
Ridge	0,559	0,558	12,609	311,640	17,653
RF	**0,728**	**0,727**	**8,320**	**192,326**	**13,868**
XGBoost	0,710	0,710	9,053	204,782	14,310

Tabela 4. Resultado dos modelos com corte de 90 dias com PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,548	0,548	12,889	319,093	17,863
SVM	0,498	0,497	11,153	354,600	18,831
RNA	0,643	0,643	10,218	252,116	15,878
Lasso	0,548	0,548	12,891	319,121	17,864
Ridge	0,548	0,548	12,889	319,097	17,863
RF	**0,714**	**0,714**	**8,487**	**202,018**	**14,213**
XGBoost	0,692	0,692	9,288	217,332	14,742

Tabela 5. Resultado dos modelos com corte de 180 dias sem PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,598	0,597	15,800	727,878	26,979
SVM	0,548	0,548	14,023	817,356	28,589
RNA	0,687	0,686	13,584	567,031	23,812
Lasso	0,598	0,597	15,804	727,982	26,981
Ridge	0,598	0,597	15,800	727,885	26,979
RF	**0,768**	**0,7678**	**10,388**	**419,690**	**20,486**
XGBoost	0,762	0,762	11,204	431,363	20,769

Tabela 6. Resultado dos modelos com corte de 180 dias com PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,588	0,588	16,195	744,506	27,286
SVM	0,542	0,542	14,462	827,667	28,769
RNA	0,677	0,677	13,238	583,709	24,160
Lasso	0,588	0,588	16,197	744,521	27,286
Ridge	0,588	0,588	16,195	744,530	27,286
RF	**0,752**	**0,752**	**10,6793**	**447,909**	**21,164**
XGBoost	0,736	0,736	11,636	478,087	21,865

variável alvo (dias em atraso) pelos modelos RNA, lasso, RF, regressão linear, XGBoost e SVM, respectivamente. Nele aparecem os quartils 1, 2 (mediana) e 3, além dos *outliers*. Nesses gráficos, uma reta indica o resultado ideal de predição dos modelos (dias de atraso previstos = dias de atraso esperados).

Tabela 7. Resultado dos modelos com corte de 365 dias sem PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,636	0,636	21,424	2032,419	45,082
SVM	0,546	0,546	19,364	2536,842	50,367
RNA	0,743	0,743	16,823	1433,983	37,868
Lasso	0,636	0,636	21,425	2032,492	45,083
Ridge	0,636	0,636	21,424	2032,429	45,083
RF	0,826	0,826	**12,899**	973,105	31,195
XGBoost	**0,829**	**0,828**	13,703	**957,952**	**30,951**

Tabela 8. Resultado dos modelos com corte de 365 dias com PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,610	0,610	22,834	2178,255	46,672
SVM	0,508	0,508	21,015	2746,873	52,411
RNA	0,716	0,716	17,678	1586,107	39,826
Lasso	0,610	0,610	22,834	2178,433	46,674
Ridge	0,610	0,610	22,834	2178,292	46,672
RF	**0,808**	**0,8078**	**13,457**	**1073,231**	**32,760**
XGBoost	0,801	0,801	14,445	1112,212	33,350

Tabela 9. Resultado dos modelos sem corte de dias e sem PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,622	0,621	47,457	16473,071	128,348
SVM	0,448	0,447	39,413	24041,997	155,055
RNA	0,756	0,756	30,068	10621,602	103,061
Lasso	0,622	0,621	47,452	16473,580	128,349
Ridge	0,622	0,621	47,457	16473,194	128,348
RF	0,895	0,895	**20,234**	4581,439	67,686
XGBoost	**0,916**	**0,916**	20,288	**3661,788**	**60,513**

Tabela 10. Resultado dos modelos sem corte de dias e com PCA.

Algoritmo	R^2	Adj R^2	MAE	MSE	RMSE
Linear	0,572	0,572	53,861	18630,976	136,495
SVM	0,255	0,255	45,319	32414,692	180,041
RNA	0,717	0,717	33,787	12320,958	111,000
Lasso	0,572	0,572	53,858	18630,898	136,495
Ridge	0,572	0,572	53,861	18631,156	136,496
RF	0,864	0,864	**22,133**	5924,430	76,970
XGBoost	**0,866**	**0,866**	23,344	**5825,528**	**76,325**

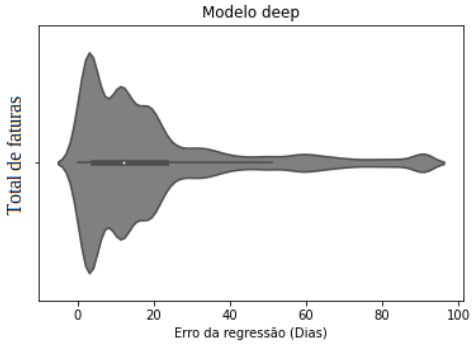

Figura 1. Função densidade de probabilidade dos erros de estimativa do modelo RNA

4. DISCUSSÕES

Verifica-se que modelos que empregam corte no período de atraso de pagamento (Tabelas 3 até 10) têm melhor desempenho, particularmente para menores intervalos de corte (modelos usando corte de 90 dias produziram os

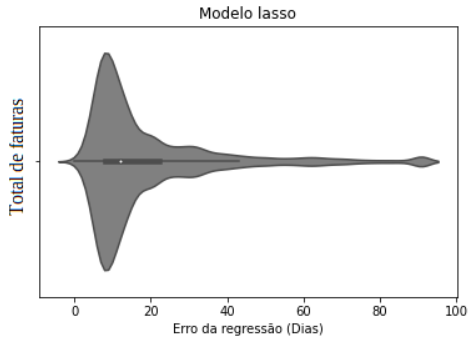

Figura 2. Função densidade de probabilidade dos erros de estimativa do modelo Lasso

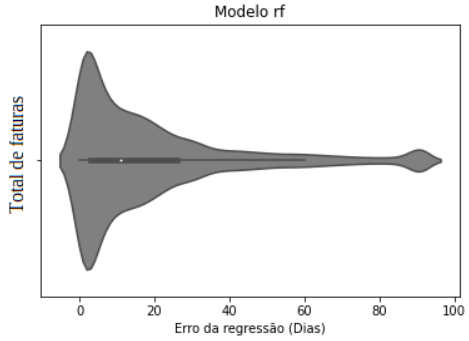

Figura 3. Função densidade de probabilidade dos erros de estimativa do modelo RF

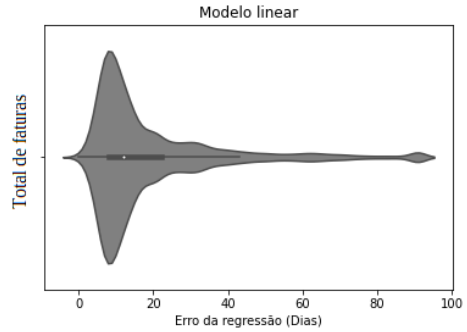

Figura 4. Função densidade de probabilidade dos erros de estimativa do modelo de regressão linear

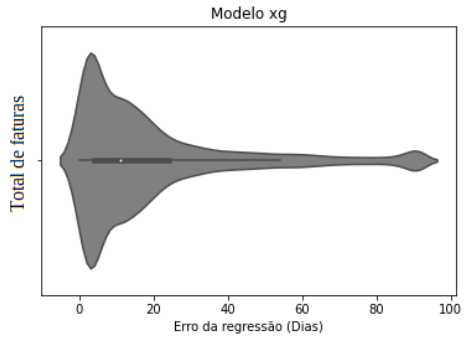

Figura 5. Função densidade de probabilidade dos erros de estimativa do modelo XGBoost

melhores resultados). A origem disso é a existência de

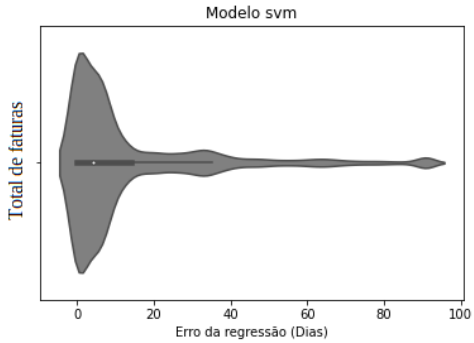

Figura 6. Função densidade de probabilidade dos erros de estimativa do modelo SVM

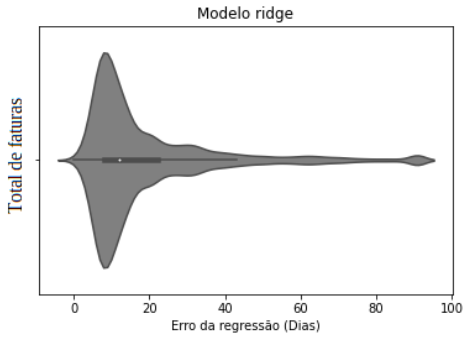

Figura 7. Função densidade de probabilidade dos erros de estimativa do modelo Ridge

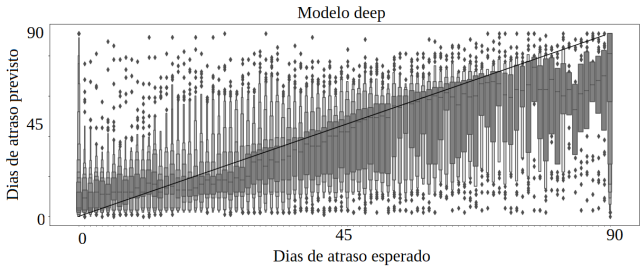

Figura 8. *Boxplot* da estimação da variável alvo (dias de atraso) para modelo RNA.

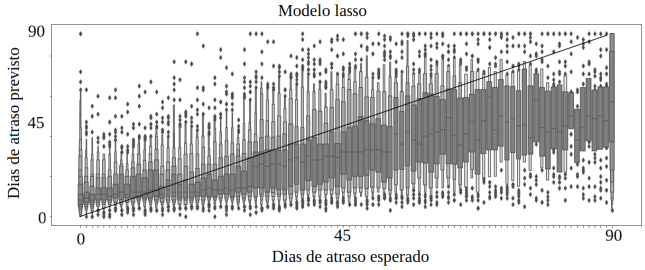

Figura 9. *Boxplot* da estimativa da variável alvo (dias de atraso) para o modelo Lasso.

poucos clientes com atrasos significativos nos pagamentos (houve casos de clientes com mais de 1600 dias sem pagar a fatura). A Tabela 11 evidencia tal resultado.

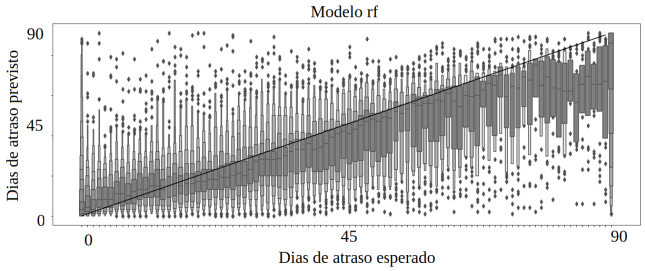

Figura 10. *Boxplot* da estimativa da variável alvo (dias de atraso) para o modelo RF.

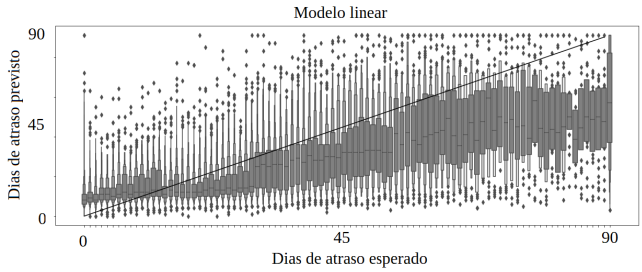

Figura 11. *Boxplot* da estimativa da variável alvo (dias de atraso) para o modelo de regressão linear.

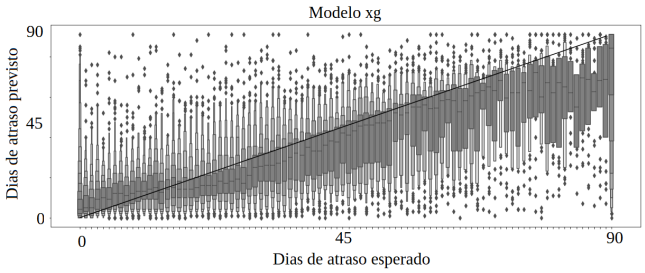

Figura 12. *Boxplot* da estimativa da variável alvo (dias de atraso) para o modelo XGBoost.

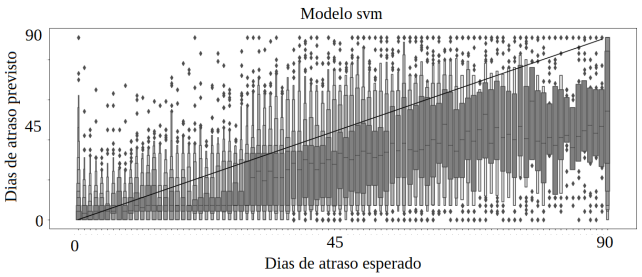

Figura 13. *Boxplot* da estimativa da variável alvo (dias de atraso) para o modelo SVM.

A aplicação de PCA para redução da dimensionalidade (de 40 variáveis de entrada para 18) produziu um resultado ligeiramente inferior.

Reforça-se que todos os resultados consideram uma trava para predições negativas de dias de atraso no pagamento das faturas: valores abaixo de zero são truncados para zero. Não houve diferença estatística nos resultados com ou sem esse truncamento.

Verifica-se pelas Figuras 1 à 7 que os modelos erram nas predição de atraso de pagamento de faturas de até 30 dias.

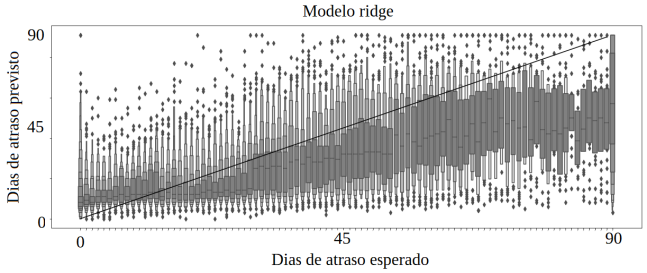

Figura 14. *Boxplot* da estimativa da variável alvo (dias de atraso) para o modelo Ridge.

Tabela 11. Resultado consolidado por grupo de corte sem PCA (∞ = sem corte).

	RMSE						
Corte (dias)	RNA	Lasso	Ridge	RF	SVM	XGBoost	Linear
90	**15,866**	**17,656**	**17,653**	**13,868**	**18,655**	**14,310**	**17,653**
180	23,812	26,981	26,979	20,486	28,589	20,769	26,979
365	37,868	45,083	45,083	31,195	50,367	30,951	45,082
∞	103,061	128,349	128,348	67,686	155,055	60,513	128,348

Tabela 12. Resultado consolidado por grupo de corte com PCA (∞ = sem corte).

	RMSE						
Corte (dias)	RNA	Lasso	Ridge	RF	SVM	XGBoost	Linear
90	**15,878**	**17,864**	**17,863**	**14,213**	**18,831**	**14,742**	**17,863**
180	24,160	27,286	27,286	21,164	28,769	21,865	27,286
365	39,826	46,674	46,672	32,760	52,411	33,350	46,672
∞	111,000	136,495	136,496	76,970	180,041	76,325	136,495

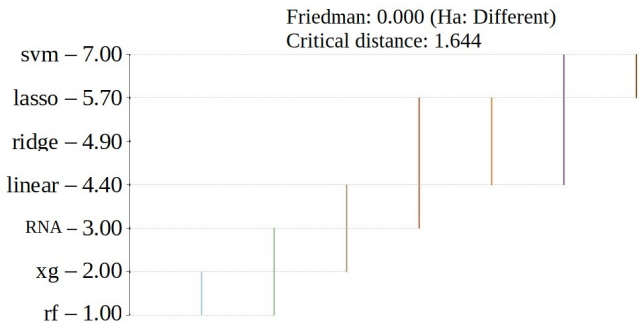

Figura 15. Teste de diferença estatística entre os resultados obtidos dos modelos.

Do ponto de vista econômico (para a empresa) isso é bom por não desencadear processo de cobrança, desligamento e religação do fornecimento de energia elétrica, etc.

A Figura 15 traz os resultados do teste de Friedman e Nemenyi. Dada a *Critical Distance* (CD) de 1.644, os modelos XGBoost e RF não apresentam diferença estatística já que possuem CD $= 1 = (2 - 1) < 1.644$. Com CD $= 2 = (3 - 1) > 1.644$, há diferença estatística entre os modelos RNA e RF. Outras comparações, dois-a-dois, entre os modelos são identificáveis a partir dessa Figura.

A partir destas análises, identica-se que a melhor performance é obtida com os modelos RF e *XGBoost*, que não apresentam diferença estatística entre seus resultados.

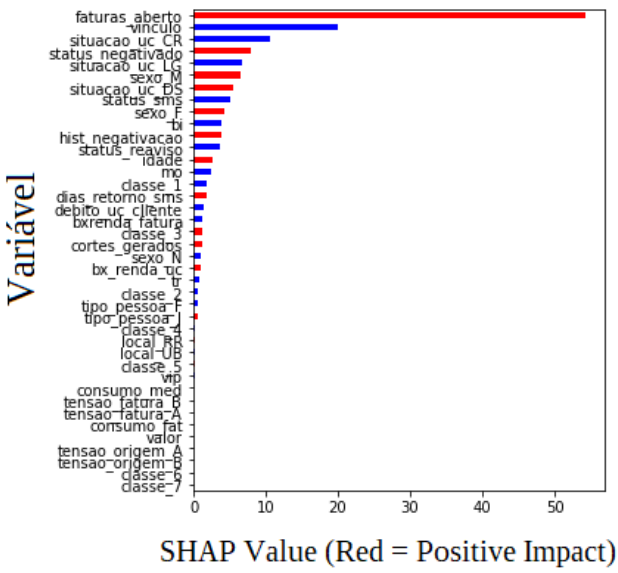

Figura 16. Influência dos atributos de entrada na variável alvo pelo algorithmo SHAP.

Reforça-se que estes e os demais modelos tiveram seus hiperparâmetros otimizados antes de seus treinamentos e testes.

As Figuras 8 à 14 permitem um exame mais detalhado da performance dos modelos em relação a sua previsão dos dias de atraso (variável alvo). Nota-se que os modelos RF e XGBoost produzem resultados mais próximos do esperado, apesar de um viés observável de predizer menos dias de atraso em relação ao valor correto a medida que essa variável alvo é incrementada. Ao mesmo tempo, o pior resultado foi obtido no modelo SVM (Figura 13.

Para avaliar a influência dos atributos de entrada na estimação da variável alvo, este trabalho usou o algoritmo SHAP (Lundberg and Lee (2017b)), cujos resultados aparecem na Figura 16. Quanto maior é o valor do índice SHAP, mais importante é o atributo. A variável alvo é diretamente ou inversamente proporcional à um atributo de entrada de acordo com a coloração das barras nesse gráfico, respectivamente vermelha e azul.

Pelo resultado obtido, a quantidade de faturas em aberto de um cliente impacta diretamente o tempo de atraso no pagamento da sua tarifa, tendo um peso significativo nesse resultado. Atributos como consumo medido ou valor da tarifa têm pequeno impacto na quantidade de dias em atraso das faturas. Tais informações são facilmente traduzidas para *stakeholders* da empresa.

Uma análise individual de cada tarifa também auxilia interpretação dos resultados para *stakeholders*. As Figuras 17, 18, 19 e 20 apresentam valores de atributos de entrada de faturas aleatoriamente escolhidas. Nessas figuras, os atributos em vermelho e azul incrementam e decrementam, respectivamente, os dias de atraso.

A Figura 17 indica uma predição de 11 dias de atraso, em que a inexistência de "faturas em atraso" contribui para a redução do atraso no pagamento, bem como se o cliente mantém vínculo com a unidade consumidora (vive na UC

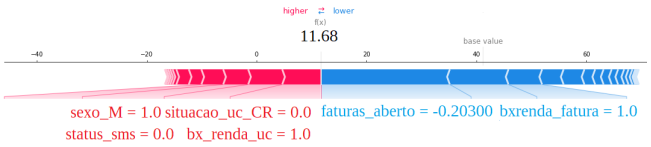

Figura 17. Resultado da interpretação do modelo para os *stakeholders* de uma fatura isolada

Figura 18. Resultado da interpretação do modelo para os *stakeholders* de uma fatura isolada

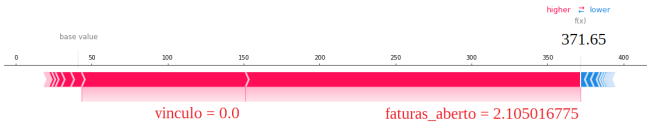

Figura 19. Resultado da interpretação do modelo para os *stakeholders* de uma fatura isolada

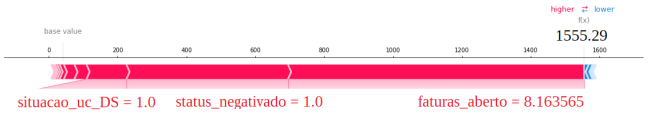

Figura 20. Resultado da interpretação do modelo para os *stakeholders* de uma fatura isolada

ou utiliza-a para trabalho). A Figura 18 confirma que a falta de vínculo do cliente com a UC eleva os dias de atraso.

No caso da falta de vínculo do cliente om a UC e a existência de duas faturas em aberto provocaram uma previsão de 371 dias de atraso, como mostra a Figura 19. No extremo, a Figura 20 mostra que a adição do *status* de negativado (registro no SERASA - orgão brasileiro de controle de crédito e dívidas) e do indicativo de que a UC já está com a energia elétrica cortada indicaram um atraso de pagamento de 1555 dias.

Esta explicabilidade dos modelos é significativa importante para que não especialistas compreendam e até aceitem resultados oriundos de métodos baseado em aprendizado de máquinas.

5. CONCLUSÃO

Este trabalho teve como objetivo encontrar um modelo para ajudar na predição do problema de (PECLD) utilizando diversos algoritmos de aprendizagem de máquina da literatura.

Para avaliar a metodologia proposta, dados reais de uma distribuidora de energia elétrica foram usados na construção de modelos analisados. Tais modelos foram comparados estatisticamente entre si.

De modo geral, os resultados obtidos foram promissores, o que permite concluir que é possível utilizar os conceitos deste trabalho em um ambiente real, substituindo a decisão humana entre qual unidade consumidora cortar por um modelo automatizado. No entanto, cabe ressaltar que para

automatizar este processo em um ambiente real diversos etapas fora do escopo deste trabalho devem ser realizadas (por exemplo, criação de um serviço que executa o modelo automaticamente para assim que uma nova fatura seja gerada, mostrando o resultado para o setor de faturamento e risco de crédito da empresa).

Outro ponto relevante para colocar os conceitos deste trabalho em produção será a necessidade de acompanhamento para verificar se os resultados obtidos nos testes pelos modelos são próximos aos do ambiente real, com um universo maior de dados para aprimoramento desses modelos. Inclusive, treinar o modelo periodicamente com dados mais atualizados torna-se relevante, pois os perfils de pagamento podem ser alterados. Por exemplo, durante o período de *lockdown* devido ao vírus, clientes podem ter deixado de pagar as faturas pela perda dos empregos. Esta informação não está presente nos dados deste trabalho.

A separação dos grupos de corte criado nos experimentos mostrou que o *stakeholder* poderá escolher qual corte utilizar e quanto menor for o corte, maior será o desempenho dos modelos. Ele pode, por exemplo, realizar um tratamento especial para os *outliers* que estão com mais de 365 sem pagar a fatura e utilizar o modelo de corte de 90 dias que retirar estes valores e, consequentemente, obter um melhor desempenho no modelo.

Como trabalhos futuros considera-se o uso da técnica de *reframing* Lakshmanan et al. (2020) para transformar o problema de regressão em um problema de classificação. Outra abordagem complementar a este trabalho que pode ser realizada é que depois que o modelo proposto neste trabalho encontrou o resultado de quando o cliente irá pagar a conta, utilizar estes dados como entrada de um algoritmo de otimização e juntos com outros dados (quantidade de equipes disponíveis, custos de desligamento, valor da fatura, etc.) encontrar uma ordem de desligamento otimizada para maximizar o lucro da empresa.

REFERÊNCIAS

Bastos, J.A. (2010). Forecasting bank loans loss-given-default. *Journal of Banking & Finance*, 2510–2517.

Breiman, L., Friedman, J.H., Olshen, R.A., and Stone, C.J. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. doi:10.1145/2939672.2939785. URL http://arxiv.org/abs/1603.02754. Cite arxiv:1603.02754Comment: KDD'16 changed all figures to type1.

Datta, A., Sen, S., and Zick, Y. (2016). Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. In *2016 IEEE Symposium on Security and Privacy (SP)*, 598–617. doi: 10.1109/SP.2016.42.

Granatyr, J. (2018). *Modelo Afetivo de Reputação utilizando Personalidade e Emoção*. Novas Edicoes Academicas.

Lakshmanan, V., Robinson, S., and Munn, M. (2020). *Machine Learning Design Patterns*. O'Reilly Media, Inc.

Lipovetsky, S. and Conklin, M. (2001). Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*, 17, 319 – 330. doi: 10.1002/asmb.446.

Lundberg, S.M. and Lee, S.I. (2017a). A unified approach to interpreting model predictions. In I. Guyon, U.V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Lundberg, S.M. and Lee, S. (2017b). A unified approach to interpreting model predictions. *CoRR*, abs/1705.07874. URL http://arxiv.org/abs/1705.07874.

Lundberg, S.M., Nair, B., Vavilala, M.S., Horibe, M., Eisses, M.J., Adams, T., Liston, D.E., Low, D.K.W., Newman, S.F., Kim, J., et al. (2018). Explainable machine-learning predictions for the prevention of hypoxaemia during surgery. *Nature Biomedical Engineering*, 2(10), 749.

Padoveze, C. (1996). *Manual de contabilidade básica: uma introdução à prática contábil*. Atlas. URL https://books.google.com.br/books?id=iLU7AAAACAAJ.

Pahwa, K. and Agarwal, N. (2019). Stock market analysis using supervised machine learning. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 197–200. doi: 10.1109/COMITCon.2019.8862225.

Pandey, J.N. and Srinivasan, M. (2011). Predicting probability of loan default. Technical report, Stanford University.

Provost, F., Fawcett, T., and Boscato, M. (2018). *Data Science Para Negócios*. ELSEVIER/ALTA BOOKS. URL https://books.google.com.br/books?id=c4lAvgAACAAJ.

Quinlan, J.R. (1986). Induction of decision trees. *MACH. LEARN*, 1, 81–106.

Ray, S. (2019). A quick review of machine learning algorithms. In *2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon)*, 35–39. doi:10.1109/COMITCon.2019.8862451.

Ribeiro, M.T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier.

Russell, S. and Norvig, P. (2004). *Inteligência Artificial*. CAMPUS - RJ.

Shrikumar, A., Greenside, P., and Kundaje, A. (2017). Learning important features through propagating activation differences. In D. Precup and Y.W. Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 3145–3153. PMLR, International Convention Centre, Sydney, Australia. URL http://proceedings.mlr.press/v70/shrikumar17a.html.

Tan, P., . Steinbach, M., and Kumar, V. (2009). *Introdução ao Data Mining*. Ciência Moderna.

Yazdi, K.M., Yazdi, A.M., Khodayi, S., Hou, J., Zhou, W., and Saedy, S. (2020). Improving fake news detection using k-means and support vector machine approaches. *International Journal of Electronics and Communication Engineering*, 14(2), 38 – 42. URL https://publications.waset.org/vol/158.