

Análise Temporal da Estabilidade de Técnicas de Aprendizado de Máquina no Gerenciamento de *Churn* em Negócios Não Contratuais

Adriana Leticia dos Reis,¹ Profa. Dra. Roseli A. F. Romero²
ICMC-USP

1 Introdução

As empresas estão inseridas em um cenário cada vez mais competitivo, o que torna a disputa por um novo cliente mais acirrada. De modo que a retenção de um cliente é tão importante quanto adquiri-lo. "Conquistar novos clientes pode custar até cinco vezes mais do que satisfazer e reter os já existentes." [1].

Alguns setores, como o de seguros, telecomunicações, bancos, entre outros, o modelo de negócio é através de um contrato de assinatura recorrente para compra de produto ou serviço. Isso faz com que monitoração da evasão de cliente, comumente conhecida como *churn* seja feita forma simples e precisa, devido ao cliente notificar a empresa quando deseja cancelar o contrato.

De acordo com [2], *churn* pode ser definido como um fenômeno de clientes que descontinuam seu relacionamento com a empresa, independente se o modelo de negócio existente é contratual ou não, como é o caso do varejo, onde modelo de negócio não é contratual.

Sendo assim, aferir a taxa de evasão em negócios não contratuais não é uma tarefa fácil e óbvia. O desafio aumenta quando a empresa pertence ao segmento B2B, devido a carteira de clientes ser menor em comparação ao segmento B2C, o que torna ainda mais relevante a retenção de cliente, de modo que a perda de apenas um cliente pode impactar diretamente a lucratividade da companhia.

Diante deste contexto, a aplicação de técnicas de aprendizado de máquina (AM), contribuem para redução do *churn*, fazendo com que a organização possa antever esse fenômeno e agir de forma ágil para reter o cliente [3].

A proposta deste trabalho é o desenvolvimento de três algoritmos de AM utilizando as técnicas de Regressão Logística (RL), Random Forest (RF) e XGBoost (XGB) para classificar a probabilidade de *churn*. Após o desenvolvimento e treino e teste dos modelos, teremos a fase de acompa-

¹reis.al@usp.br

²rafrance@icmc.usp.br

nhamento da estabilidade ao longo do tempo, assumindo que nesta fase não haverá novos treinos nem ajustes de hiperparâmetros.

O objetivo é identificar qual técnica manteve a eficácia preditiva à medida que o tempo se distancia do momento em que os algoritmos foram treinados.

2 Materiais e Métodos

2.1 Análise da base de dados e pré-processamento

Os dados foram extraídos do sistema de *Data Warehouse* no formato "xlsx" de uma empresa de médio porte do setor de venda de produtos químicos para o agronegócio B2B. Compreendem transações de vendas de 363 clientes ocorridas entre 01 de janeiro de 2020 a 31 de dezembro de 2023, totalizando 16 trimestres, pois a marcação do *churn* é trimestral. Ao total são 2.948 observações e 6 variáveis preditoras.

A proporção das classes da variável *churn* para as observações que são evento (1) representam 15,8% e quem não é evento (0) representam 84,2%. Portanto, há desbalanceamento entre as classes. Além disso, foi constatado que não há valores faltantes dentre as observações.

Ao final, foi aplicada a normalização nas variáveis preditoras, pois estão em escalas diferentes. Para isso foi utilizado o método *MinMaxScaler* [4].

2.2 Modelagem

Com o objetivo de simular o cenário real de uma organização, a divisão dos dados em conjuntos de treino, teste e acompanhamento seguiu a cronologia temporal. Com esta divisão, temos:

- **Treino:** Entre 01/01/2020 a 31/12/2021 - 8 trimestres, totalizando 1.358 observações.
- **Teste:** Entre 01/01/2022 a 31/12/2022 - 4 trimestres, totalizando 800 observações.
- **Acompanhamento:** Entre 01/01/2023 a 31/01/2023 - 4 trimestres, totalizando 790 observações.

A implementação dos modelos foi feita através da linguagem Python, utilizando a biblioteca *scikit-learn*. Para o modelo RL a classe aplicada foi *LogisticRegression* e para o modelo RF foi *BalancedRandomForestClassifier*. A Tabela 1 quais hiperparâmetros foram otimizados.

Após o treino e ajustes dos hiperparâmetros, eles serão salvos no formato *pickle*. Afim de assegurar que na fase de acompanhamento não haverá *tunagem* dos modelos. Isso torna viável analisar qual dos modelos mantém sua capacidade preditiva ao longo do tempo.

2.3 Métricas de qualidade de ajuste dos modelos

A avaliação da qualidade de ajustes dos algoritmos será dada por meio das métricas: Acurácia, Sensibilidade (*Recall*), Precisão (*Precision*), F1 Score e a área formada sob a curva ROC (AUC-ROC) [5]. Elas serão aplicadas no momento do teste dos algoritmos e também no acompanhamento temporal para avaliar qual modelo manteve a capacidade preditiva à medida em que o tempo se distancia do momento em que foram treinados os algoritmos.

Modelo	Hiperparâmetros	Descrição
RL	class_weight = balanced; solver = liblinear	Trata o desbalanceamento das classes na variável churn; Otimiza da função de custo.
RF	n_estimators = 150 max_depth = None	Número ideal de árvores usadas para construir a RF; Profundidade da árvore, afim de ser expandida até que todas as folhas sejam puras.
XGB	A implementar	

3 Resultados

No tocante a modelagem, após o desenvolvimento de cada modelo, o conjunto de teste foi submetido a estes algoritmos para validar a qualidade de ajuste de cada técnica. Conforme observado na Tabela 3, o modelo RF apresenta resultados superiores ao RL em todas as métricas, com exceção ao empate na Sensibilidade.

Tabela 1: Análise da qualidade do ajuste na fase treino-teste

Modelo	Acurácia	Sensibilidade	Precisão	F1 Score	AUC-ROC
Regressão Logística	0,744	0,843	0,354	0,499	0,850
Random Forest	0,794	0,843	0,411	0,553	0,876
XGBoost	A implementar				

Com os modelos salvos no formato *pickle*, foi realizado o acompanhamento da performance preditiva no decorrer de cada trimestre. Neste trabalho demonstraremos apenas as métricas consideradas mais adequadas para tratar o contexto deste estudo. São elas a Sensibilidade que mede a taxa de acerto de quem é evento e a AUC-ROC por ser capaz de medir o desempenho geral do modelo.

Tabela 2: Descrição da implementação dos modelos

Métrica	Trimestre/2023	RL	RF	XGB
Sensibilidade	1º	0,846	0,808	A implementar
	2º	0,733	0,800	
	3º	0,818	0,818	
	4º	0,831	0,708	
AUC-ROC	1º	0,813	0,865	
	2º	0,777	0,869	
	3º	0,842	0,878	
	4º	0,828	0,819	

Com relação a performance ao longo do tempo, embora houve empate na Sensibilidade em ambos modelos no momento do treino. Todavia, no 1º trimestre a RL manteve o mesmo valor,

enquanto RF reduziu. O oposto ocorreu no 2º trimestre, sendo que RL caiu a Sensibilidade enquanto RF cresceu. No 3º trimestre ambas tiveram o mesmo valor, porém, com valores menores em relação ao momento do treino. No 4º trimestre o modelo RL aumentou a performance em relação ao trimestre anterior, todavia, RF apresentou queda significativa.

Já a AUC-ROC o modelo RF foi superior a RL em todos os trimestres, exceto no 4º trimestre, repetindo o mesmo comportamento da métrica Sensibilidade.

4 Conclusão

Neste trabalho foram investigados até o momento dois métodos de AM para o gerenciamento de *churn* em negócios não contratuais do segmento B2B. A principal contribuição deste trabalho consistiu na investigação da técnica mais adequada para resolver o problema colocado, isto é, a técnica que se manteve mais estável ao longo do tempo, visando a redução de custos recorrentes de calibragem de modelos, por exemplo, aluguel de ambientes em *cloud* - AWS, Azure entre outros.

Ambos modelos apresentaram quedas ao longo do tempo, todavia, até o 3º trimestre, o modelo RF apresentou desempenho mais estável em relação à RL. Além disso, as quedas com o passar dos trimestres não apresentam grandes magnitudes, com exceção a Sensibilidade no 4º trimestre.

Desse modo, podemos concluir que é possível escalonar o modelo Random Forest e mantê-lo por nove meses operando sem a necessidade de calibração dos hiperparâmetros. Consequentemente, os ganhos para a empresa serão redução de *churn* ao adotar AM e baixos custos com a manutenção deste modelo.

Referências

- [1] P. Kotler and K. L. Keller. *Administração de Marketing, 15a. edição*. Pearson, São Paulo, 2018.
- [2] J. R. Dias and N. Antonio. Predicting customer churn using machine learning: A case study in the software industry, *Journal of Marketing Analytics*, 2023. DOI: 10.1057/s41270-023-00269-9.
- [3] A. Zadoo, T. Jagtap, N. Khule, A. Kedari and S. Khedkar. A review on Churn Prediction and Customer Segmentation using Machine Learning, *2022 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COM-IT-CON)*, volume 1, 2022. DOI: 10.1109/COM-IT-CON54601.2022.9850924.
- [4] F. Pedregosa et al. Scikit-learn: Machine Learning in Python, *Journal of Machine Learning Research*, volume 12, 2011. URL: <https://scikit-learn.org/1.5/modules/generated/sklearn.preprocessing.MinMaxScaler.html>.
- [5] J. D. Kelleher and B. M. Namee and A. D'Arcy. *Fundamentals of Machine Learning for Predictive Data Analytics - Algorithms, Worked Examples, and Case Studies*. The MIT Press, Cambridge, 2015.