COMPARAÇÃO DE MODELOS DE MACHINE LEARNING PARA PREVISÃO DE CHURN EM UMA EMPRESA DE MÍDIA DE PORTO ALEGRE

Filipe Carbonera de Souza[1]

Alessandro Nunes de Souza[2]

RESUMO

Este artigo trata-se de um estudo de caso em que foram elaborados modelos *de machine learning* para previsão de *churn* de um produto digital de uma empresa de mídia de Porto Alegre. O objetivo deste artigo é identificar o melhor modelo de *machine learning* disponível na ferramenta utilizada pela empresa, o BigQuery, e quais dados tem maior relação direta com o cancelamento dos clientes. As métricas de precisão, *recall*, acurácia e matriz de confusão foram utilizadas para medir a performance dos modelos treinados e foi aplicado o método *backward elimination* no modelo de maior acurácia para compreender a relação das variáveis utilizadas para a acurácia do modelo. Os modelos apresentaram resultados semelhantes, com exceção da regressão logística, e as variáveis de maior relevância para os modelos foram acessos via *newsletter*, tempo de base, sistema operacional mais utilizado, produto assinado, plataforma e *browser* mais acessados.

Palavras-chave: Cancelamento; Aprendizado de Máquina; Mídia e Assinatura.

ABSTRACT

This article is a case study where machine learning models were developed to predict the churn of a digital product from a media company in Porto Alegre. The purpose of this article is to identify the best machine learning model available in the tool BigQuery and which feature has the greatest direct relationship with customer churn. The metrics of precision, recall, accuracy and confusion matrix were used to measure the performance of the trained models and the backward elimination method was applied to the model with the highest accuracy to understand the relationship of the variables used for the accuracy of the model. The models had similar results with the exception of logistic regression and the variables with most relevance for the models were hits via newsletter, base time, most used operational system, subscribed product, most accessed platform and browser.

***Keywords**: Churn; Machine Learning; Media; Subscription.*

[1] Graduando do Curso de Administração LF Administração de Empresas da Pontifícia Universidade Católica do Rio Grande do sul. E-mail: filipe.carbonera@edu.pucrs.br.
[1] Orientador Professor da Escola de Negócios da Pontifícia Universidade Católica do Rio Grande do sul. E-mail: ansouza@pucrs.br.

1 INTRODUÇÃO

O avanço tecnológico das últimas décadas transformou drasticamente os setores econômicos e sociais. Nesse contexto, a Ciência de Dados assumiu papel de protagonismo na busca por soluções inovadoras, eficientes e eficazes (ULLAH et al., 2019). Particularmente, na economia digital, o conceito de *'churn'*, que se refere à taxa de cancelamento ou não renovação de um serviço, tornou-se uma métrica essencial para empresas que operam sob modelos de assinatura (KHODABANDEHLOU; RAHMAN, 2017).

A compreensão deste fenômeno, com vistas à minimização de sua ocorrência, é crucial para a sobrevivência e crescimento sustentável de negócios nesse setor. Dessa forma, as empresas têm buscado, cada vez mais, a adoção de estratégias de retenção de clientes baseadas em Modelos de *Machine Learning* (ML) para previsão de *churn* (KUMAR; CHANDRAKALA, 2016). A tarefa é desafiadora, uma vez que envolve a análise de inúmeros dados e comportamentos dos usuários, que são dinâmicos e muitas vezes imprevisíveis. Nessa perspectiva, a escolha correta do modelo de ML a ser adotado é crucial para o sucesso de ações com este objetivo (LALWANI et al., 2022).

Tendo como base este conceito, neste artigo será discutido sobre como ele pode ser aplicado em uma empresa mídia de Porto Alegre, com atuação significativa no setor de assinaturas digitais, pois enfrenta essa realidade diariamente. Em sua base de clientes, composta por milhares de assinantes, a taxa de *churn* representa uma importante métrica a ser monitorada e controlada. Consequentemente, a empresa está em busca de estratégias efetivas de ML para lidar com este fenômeno.

Existem diversos modelos de ML disponíveis para tratar deste problema, cada um com suas características, vantagens e desvantagens. Desde métodos mais simples, como Regressão Logística, até técnicas mais sofisticadas, como Redes Neurais e *Gradient Boosting*, podem ser empregadas para prever o churn (KUMAR; CHANDRAKALA, 2016). A escolha do modelo adequado é uma decisão importante, que deve ser baseada em um entendimento profundo das particularidades do problema, dos dados disponíveis e das capacidades e limitações de cada técnica (AHMAD; JAFAR; ALJOUMAA, 2019).

Nesse sentido, o presente estudo objetiva comparar e avaliar diferentes modelos de ML para previsão de *churn*, considerando seu desempenho e facilidade de implementação, e como esses aspectos se aplicam ao contexto específico da Empresa objeto de estudo. A discussão foi conduzida a partir de uma perspectiva crítica e analítica, visando não apenas entender as

diferenças entre os modelos, mas também elucidar as melhores práticas e estratégias para sua implementação eficaz.

Considerando que o problema de pesquisa foca na identificação dos fatores mais importantes na previsão de *churn*, é apresentado, na sequência, a relação entre *machine learning* e *churn*, e como os modelos de ML e suas métricas auxiliam nesse processo. Então, é descrito o método utilizado e a discussão dos resultados encontrados.

2 MACHINE LEARNING E CHURN

A essência da retenção de clientes reside no conceito de ***'churn'***, uma métrica estratégica que reflete a quantidade de indivíduos que decidem interromper a sua relação com uma empresa durante um período específico (COIMBRA et al., 2023). Derivada do inglês, a palavra *'churn'* significa 'agitador' ou 'borrão', transmitindo a ideia de turbulência e mudança, uma representação adequada para descrever a evasão de clientes (PIRROLAS; CORREIA, 2020). No âmbito das empresas que operam sob um modelo de assinatura, o *churn* se manifesta quando um assinante escolhe não renovar a sua assinatura (ALBUQUERQUE et al., 2021).

Assim sendo, o ***churn* médio mensal** representa a média da quantidade de clientes que cancelaram suas assinaturas ao longo de um mês, em relação ao total de assinantes ativos no início desse período (VAFEIADIS et al., 2015). Para empresas que possuem uma base de clientes bastante variada, com números altos de novas assinaturas e cancelamentos, o *churn* médio mensal se torna uma ferramenta essencial para o monitoramento da saúde do negócio (JADHAV et al., 2021). Ao calcular essa métrica, as empresas podem identificar tendências, sejam elas positivas ou negativas, e ajustar suas estratégias de retenção de clientes de acordo (CHAUDHARY et al., 2022).

A medição do *churn* é essencial em qualquer negócio, mas é especialmente crítica no setor de mídia, onde a concorrência é acirrada e a satisfação do cliente é altamente volátil. As empresas de mídia lutam constantemente para manter seus assinantes engajados e satisfeitos, enquanto tentam atrair novos clientes (LEMOS; SILVA; TABAK, 2022). Portanto, entender o *churn* é fundamental para a sobrevivência e o sucesso de qualquer empresa de mídia (MATUSZELAŃSKI; KOPCZEWSKA, 2022).

No entanto, o entendimento dessas métricas não seria possível sem o apoio de técnicas de análise avançadas, tais como as fornecidas pelo ***Machine Learning*** (ML) (CHAUDHARY et al., 2022). O ML é um ramo da Inteligência Artificial que emprega algoritmos capazes de aprender a partir de dados, melhorando seu desempenho ao longo do tempo sem a necessidade

de programação explícita (KAVITHA et al., 2020). O treinamento de modelos de ML envolve a ingestão de grandes volumes de dados e o ajuste dos parâmetros do modelo para minimizar o erro entre as previsões e os resultados reais (PIRROLAS; CORREIA, 2020).

Existem três principais categorias de ML: **aprendizado supervisionado**, não supervisionado e por reforço. O aprendizado supervisionado é o mais comum e envolve a utilização de um conjunto de dados com respostas conhecidas para treinar o modelo (VAFEIADIS et al., 2015). No **aprendizado não supervisionado**, o modelo aprende a identificar padrões em grandes volumes de dados sem qualquer orientação explícita, o que pode ser útil quando as relações subjacentes aos dados não são conhecidas (KUMAR, CHANDRAKALA, 2016).

Por último, o **aprendizado por reforço** envolve modelos que aprendem através da experimentação, melhorando suas decisões com base no feedback recebido. Cada tipo de aprendizado de máquina tem suas especificidades e é melhor aplicado a diferentes tipos de problemas (PIRROLAS; CORREIA, 2020). Na previsão de *churn*, a abordagem mais comum é tratar o problema como um caso de classificação supervisionada, onde cada cliente é categorizado como 'risco de *churn'* ou 'sem risco de *churn'* (KHODABANDEHLOU; ZIVARI RAHMAN, 2017). Os **modelos de classificação** buscam aprender a partir de um conjunto de dados de treinamento e então aplicar o conhecimento adquirido para classificar novas observações (LEMOS; SILVA; TABAK, 2022).

Na classificação de *churn*, os modelos de ML aprendem com os comportamentos passados dos clientes - aqueles que permaneceram e aqueles que cancelaram suas assinaturas - para identificar padrões que possam indicar uma probabilidade maior de cancelamento no futuro (ALMEIDA et al., 2022). Além disso, esses modelos também podem ajudar a identificar as variáveis mais importantes que contribuem para o *churn* (AHMAD; JAFAR; ALJOUMAA, 2019). Por exemplo, pode ser que a frequência de uso, o tempo desde a última interação ou o tipo de conteúdo consumido estejam fortemente correlacionados com a probabilidade de um cliente cancelar sua assinatura.

Em resumo, o conceito de *churn*, juntamente com a métrica de *churn* médio fornecem um entendimento crucial do comportamento dos clientes e da saúde geral do negócio. Ao mesmo tempo, o *Machine Learning*, com suas diversas modalidades e aplicações, proporciona as ferramentas necessárias para analisar essas métricas de forma profunda e detalhada. Especialmente no setor de mídia, a combinação desses elementos pode fornecer insights que ajudam as empresas a melhorar suas estratégias de retenção de clientes, otimizando a satisfação do cliente e, em última análise, a lucratividade do negócio.

Em relação ao **segmento de mídia**, o *churn* se apresenta como um indicador decisivo, dada a intensa competição do mercado e a dinâmica volátil de consumo dos usuários. Neste setor, os assinantes tendem a alternar entre diferentes serviços de mídia com base em fatores como a qualidade e variedade do conteúdo, preço, experiência do usuário e o serviço ao cliente (COIMBRA et al., 2023). Portanto, para as empresas de mídia, entender e antecipar o *churn* é de suma importância, não apenas para a manutenção dos níveis atuais de rendimento, mas também para o crescimento e expansão dos negócios (PIRROLAS; CORREIA, 2020). Para superar essa instabilidade e construir uma base de assinantes sólida e engajada, as empresas de mídia têm buscado soluções na intersecção da tecnologia e dos dados. É aqui que o ML surge como uma ferramenta poderosa (SRIVASTAVA; EACHEMPATI, 2021).

Por exemplo, um modelo pode ser treinado para prever se um cliente cancelará ou manterá sua assinatura - um problema de classificação binária (SRIVASTAVA; EACHEMPATI, 2021). Neste caso, o modelo de ML é alimentado com dados de clientes, incluindo tanto aqueles que cancelaram suas assinaturas (*churn*) quanto aqueles que permaneceram (não *churn*).

O modelo aprende com esses dados e, em seguida, é capaz de classificar novos clientes com base na probabilidade de *churn* (KHODABANDEHLOU; RAHMAN, 2017). Portanto, em um setor tão volátil como o de mídia, os problemas de classificação, alimentados por ML, tornam-se ferramentas vitais para antecipar o *churn* e, consequentemente, tomar importantes decisões estratégicas (KUMAR, CHANDRAKALA, 2016).

Ao compreender a importância do *churn* no setor de mídia e o papel crucial que o *machine learning* desempenha na previsão e mitigação do *churn*, fica claro que é fundamental adentrar mais profundamente nas técnicas específicas de ML. No próximo capítulo, serão abordados alguns dos modelos de *machine learning* mais comumente usados para prever o *churn*, incluindo o *Boosted Tree Classifier*, *Random Forest Classifier*, *Deep Neural Network Combined Classifier*, *Deep Neural Network Classifier* e *Logistic Regression*.

Cada modelo tem suas peculiaridades e nuances que os tornam apropriados para diferentes tipos de tarefas e dados. Portanto, será detalhado como cada um desses modelos funciona, bem como suas vantagens e limitações. Por exemplo, será abordada a questão da *Logistic Regression* e como essa técnica pode não performar bem com dados não lineares.

2.1 MODELOS DE MACHINE LEARNING PARA PREVISÃO DE CHURN

Dentro do dinâmico ambiente das empresas de mídia, o *churn* representa um desafio constante que requer soluções sofisticadas (VAFEIADIS et al., 2015). Com o avanço da tecnologia, os modelos de ML surgem como poderosos aliados nessa missão, cada um com suas características particulares e aplicações, dando aos profissionais uma gama diversificada de ferramentas para abordar este problema complexo (LALWANI et al., 2022).

Consideremos, inicialmente, *o* ***Boosted Tree Classifier***. Este é um tipo de modelo que explora a estratégia de *boosting*, um método que combina uma série de árvores de decisão fracas para criar um modelo robusto (ULLAH et al., 2019). A ideia é que, ao agregar várias previsões simples, é possível produzir uma previsão final mais precisa. No ambiente volátil das empresas de mídia, onde os padrões de consumo podem mudar rapidamente, esse tipo de modelo pode ser particularmente útil, pois permite captar uma variedade de padrões sutis que podem indicar um risco de *churn* (CHAUDHARY et al., 2022).

Avançando para o ***Random Forest Classifier***, tem-se um modelo que cria um conjunto de árvores de decisão e faz previsões baseadas na média das previsões de todas as árvores (MATUSZELAŃSKI; KOPCZEWSKA, 2022). Este modelo é bem conhecido por sua robustez e capacidade de lidar com uma grande quantidade de recursos, o que pode ser extremamente útil quando se trata de analisar os comportamentos complexos dos assinantes de mídia (KHODABANDEHLOU; RAHMAN, 2017).

Já os modelos baseados em redes neurais profundas, como o ***Deep Neural Network Classifier*** e o ***Deep Neural Network Combined Classifier***, trazem o poder do aprendizado profundo (AL-NAJJAR; AL-ROUSAN; AL-NAJJAR, 2022). Esses modelos imitam o funcionamento do cérebro humano, aprendendo a identificar padrões complexos em grandes volumes de dados (KUMAR, CHANDRAKALA, 2016). No contexto de empresas de mídia, onde os dados sobre o comportamento dos usuários podem ser multifacetados e ricos, esses modelos podem oferecer insights profundos sobre os fatores que levam ao *churn* (AHMAD; JAFAR; ALJOUMAA, 2019).

Ademais, a ***Logistic Regression***, um modelo que, apesar de ser mais simples em comparação com as técnicas anteriormente citadas, ainda é amplamente usado devido à sua interpretabilidade e eficiência (LALWANI et al., 2022). Este modelo estima a probabilidade de um evento ocorrer, tornando-o uma ferramenta natural para prever o *churn*. No entanto, é importante notar que a *Logistic Regression* pode não se sair bem quando se trata de dados não lineares, uma situação comum em empresas de mídia, onde os padrões de comportamento dos

usuários podem ser influenciados por uma multiplicidade de fatores complexamente interligados (COIMBRA et al., 2023). Portanto, é fundamental que as empresas de mídia considerem cuidadosamente a escolha do modelo de ML ao abordar o problema do *churn*.

Dependendo do tipo de dados e das características específicas do ambiente, um modelo pode ser mais ou menos eficaz (LEMOS; SILVA; TABAK, 2022). As empresas de mídia devem reconhecer que a análise de *churn* é uma tarefa complexa, e que a escolha do modelo apropriado é apenas uma peça do quebra-cabeças (ALMEIDA et al., 2022). Para se obter uma imagem completa e precisa, é essencial levar em conta uma variedade de fatores, como a qualidade e a granularidade dos dados, o poder de computação disponível e a interpretabilidade e explicabilidade do modelo escolhido (ALBUQUERQUE et al., 2021).

A eficácia do *Boosted Tree Classifier*, por exemplo, depende da qualidade dos dados de entrada. Uma vez que se baseia na combinação de previsões de várias árvores de decisão, esse modelo pode sofrer de *overfitting* se os dados de entrada forem ruidosos ou se houver muitos outliers (ÇELIK; OSMANOGLU, 2019). Por outro lado, o *Random Forest Classifier* tem a vantagem de ser robusto a outliers e de poder lidar com muitos recursos, mas também pode ser computacionalmente intensivo, o que pode ser uma limitação para empresas com recursos computacionais limitados (JADHAV et al., 2021).

Os modelos de redes neurais profundas, como o *Deep Neural Network Classifier* e o *Deep Neural Network Combined Classifier*, são poderosos, mas também exigem uma quantidade significativa de dados e poder de computação. Eles são mais adequados para situações em que há uma grande quantidade de dados disponíveis e onde os padrões são complexos e não lineares (KAVITHA et al., 2020).

No entanto, podem ser difíceis de interpretar, o que pode ser um desafio em situações em que é necessário explicar as previsões do modelo. A Regressão Logística é um modelo amplamente utilizado, mas sua simplicidade pode ser uma desvantagem quando se trata de dados não lineares (JAIN; TOMAR; JANA, 2021). Embora seja eficaz para prever o *churn* em muitas situações, pode lutar para capturar a complexidade do comportamento dos usuários de mídia se os fatores que influenciam o *churn* são complexamente interligados de maneira não linear (GEILER; AFFELDT; NADIF, 2022).

Dessa forma, nota-se que embora existam muitos modelos de ML disponíveis para prever o *churn*, a escolha do modelo mais adequado depende de uma variedade de fatores. É essencial que as empresas de mídia considerem todas essas variáveis ao implementar soluções de ML para reduzir o *churn* e melhorar a retenção de clientes (ALMEIDA et al., 2022).

As empresas de mídia operam em um cenário em constante evolução, onde a retenção de usuários é um pilar para a sustentabilidade dos negócios (SRIVASTAVA; EACHEMPATI, 2021). Assim, além de selecionar o modelo de ML mais adequado para prever o *churn*, é crucial entender a lógica subjacente à decisão do cliente de se desligar. As informações extraídas desses modelos podem, portanto, contribuir para a criação de estratégias mais eficazes de engajamento e retenção de usuários (ÇELIK; OSMANOGLU, 2019).

Assim, a implementação bem-sucedida de um modelo de ML depende da compreensão profunda do público-alvo e do segmento de mídia no qual a empresa opera (MATUSZELAŃSKI; KOPCZEWSKA, 2022).

Por exemplo, em uma empresa de streaming de vídeo, os padrões de visualização dos usuários, o tempo gasto na plataforma, o gênero preferido, e a regularidade de uso podem ser indicadores cruciais de *churn* (KAVITHA et al., 2020). Nestes casos, um modelo como o *Deep Neural Network Classifier*, capaz de discernir complexas interações não-lineares entre esses fatores, pode ser particularmente útil (JAIN; TOMAR; JANA, 2021).

Em contrapartida, no ambiente de uma empresa de mídia impressa que faz a transição para o digital, a análise do *churn* pode exigir uma abordagem diferente. Ademais, além da escolha do modelo, o treinamento e a otimização também são partes fundamentais do processo. O treinamento envolve alimentar o modelo com um conjunto de dados históricos, permitindo que o algoritmo aprenda a identificar os padrões associados ao *churn* (ULLAH et al., 2019).

Já a otimização, requer um ajuste fino dos parâmetros do modelo para melhorar seu desempenho. No entanto, o treinamento e a otimização de modelos de ML podem ser tarefas desafiadoras, especialmente quando se lida com grandes volumes de dados e complexidade de padrões, como é comum no ambiente de mídia (ALBUQUERQUE et al., 2021). As empresas de mídia devem considerar o impacto do *churn* nos diferentes segmentos de sua base de usuários. Por exemplo, os usuários mais jovens podem ter padrões de comportamento e preferências diferentes dos usuários mais velhos (COIMBRA et al., 2023). Isso poderia justificar a implementação de diferentes modelos de ML para diferentes segmentos, a fim de fornecer uma visão mais granular e precisa do *churn*. Por outro lado, a implementação bem-sucedida de qualquer modelo de ML também depende da qualidade dos dados de entrada (MATUSZELAŃSKI; KOPCZEWSKA, 2022).

Em uma empresa de mídia, isso pode envolver a coleta e a limpeza de dados de várias fontes, incluindo dados demográficos dos usuários, dados de comportamento do usuário e dados de interação com o conteúdo (ÇELIK; OSMANOGLU, 2019). A presença de dados

faltantes ou imprecisos pode prejudicar o desempenho do modelo e levar a previsões imprecisas (JADHAV et al., 2021).

Além disso, as empresas de mídia podem se beneficiar da integração de suas estratégias de ML com outras tecnologias emergentes. Por exemplo, a inteligência artificial (IA) pode ser usada para automatizar a segmentação de usuários e a personalização do conteúdo, enquanto a análise de big data pode proporcionar insights mais profundos sobre os padrões de comportamento do usuário (JAIN; TOMAR; JANA, 2021).

Isso pode contribuir para a criação de uma experiência do usuário mais personalizada e envolvente, o que pode, por sua vez, reduzir o *churn*. Dessa maneira, a adoção de modelos de ML para prever o *churn* é apenas uma parte da solução (GEILER; AFFELDT; NADIF, 2022). Para maximizar a retenção de usuários, as empresas de mídia devem se esforçar para entender as necessidades e os desejos de seus usuários e trabalhar para criar uma experiência do usuário que seja não apenas atraente, mas também significativa e gratificante (CHAUDHARY et al., 2022).

Cada empresa de mídia deve considerar suas particularidades e necessidades específicas ao selecionar e implementar um modelo de ML (LEMOS; SILVA; TABAK, 2022). Isso envolve não apenas escolher o modelo correto, mas também garantir a qualidade dos dados de entrada e considerar o impacto do *churn* nos diferentes segmentos da base de usuários (JADHAV et al., 2021).

As empresas de mídia também devem se concentrar em entender por que os usuários estão cancelando seus serviços e trabalhar para resolver esses problemas (JAIN; TOMAR; JANA, 2021). Isso pode envolver a realização de pesquisas de satisfação do cliente, a análise das reclamações dos clientes e a implementação de melhorias com base nos feedbacks recebidos. Para concluir, o papel dos modelos de ML na previsão de *churn* é inegável. Eles fornecem às empresas de mídia uma maneira poderosa de identificar os usuários que estão em risco de cancelar seus serviços, permitindo que tomem medidas proativas para reter esses usuários (SRIVASTAVA; EACHEMPATI, 2021).

2.2 MÉTRICAS DE AVALIAÇÃO DE MODELOS E OUTRAS TÉCNICAS DE MACHINE LEARNING

Para o entendimento das métricas de avaliação de ML para a retenção de clientes faz-se necessária a compreensão da Precisão, *Recall* e Acurácia. Estas medidas atuam como a bússola que orienta as estratégias de previsão de *churn*, fornecendo o norte de um algoritmo

eficaz (LEMOS; SILVA; TABAK, 2022). A **Precisão**, por exemplo, é uma métrica que quantifica a confiabilidade do algoritmo. Em termos práticos, é a proporção de previsões corretas entre todas as previsões feitas. Na indústria da mídia, uma alta precisão indica que o modelo tem uma alta probabilidade de identificar corretamente os clientes que estão prestes a cancelar seus serviços (KHODABANDEHLOU; RAHMAN, 2017).

Já o ***Recall***, outro indicador fundamental, enfoca a proporção de verdadeiros positivos identificados corretamente. Para empresas de mídia, um recall elevado indica que o modelo é capaz de identificar a maioria dos clientes que estão em risco de *churn* (GEILER; AFFELDT; NADIF, 2022). Por outro lado, a **Acurácia** é uma métrica mais geral, que considera tanto os verdadeiros positivos quanto os verdadeiros negativos (ALBUQUERQUE et al., 2021). Uma alta acurácia significa que o modelo é bom tanto em identificar clientes que vão cancelar quanto em não alarmar falsamente aqueles que permanecerão leais (ÇELIK; OSMANOGLU, 2019).

A representação visual da **matriz de confusão** oferece uma perspectiva aprofundada e intuitiva dos resultados (AL-NAJJAR; AL-ROUSAN; AL-NAJJAR, 2022). Consequentemente, o trabalho de previsão de *churn* não se limita apenas à criação e treinamento de modelos. Ao contrário, é uma operação contínua que requer um monitoramento constante e ajustes regulares (KAVITHA et al., 2020). Assim, as empresas precisam não apenas identificar os clientes que estão em risco de *churn*, mas também entender os motivos que os levam a cancelar seus serviços. É aí que a exploração dos dados da empresa se torna fundamental (ULLAH et al., 2019).

Também é importante considerar técnicas de engenharia de recursos *(Feature Engineering*) e *Backward Elimination*, que são relevantes para o desempenho dos modelos (AL-NAJJAR; AL-ROUSAN; AL-NAJJAR, 2022). A ***Feature Engineering***, por exemplo, envolve a criação de novos recursos a partir dos dados existentes, ajudando a melhorar a performance do modelo (VAFEIADIS et al., 2015). Já a eliminação para trás (***Backward Elimination***), é uma técnica usada para otimizar a performance dos modelos, removendo variáveis que têm pouco ou nenhum impacto sobre a previsão de *churn*. Para as empresas de mídia, isso pode resultar em um modelo mais eficiente e eficaz (GEILER; AFFELDT; NADIF, 2022).

3 MÉTODO

O tipo de pesquisa utilizado neste trabalho foi **quantitativo**, pois a elaboração dos dados, dos modelos de *machine learning* e demais técnicas aplicadas não contaram com

aspectos subjetivos e sim representações matemáticas. Considera-se a pesquisa como **exploratória** pois além da própria elaboração dos dados utilizados nos modelos, e além da aplicação dos próprios modelos, houve uma análise exploratória em busca de possíveis insights para a empresa objeto de estudo. Além disso, é uma pesquisa **descritiva** por apresentar os dados obtidos com os modelos a fim de obter as respostas a serem analisadas. Por fim, o método utilizado foi o **estudo de caso**, porque foi tratado o efeito do *churn* em um produto digital da empresa objeto desse estudo.

A técnica de coleta de dados foi a **pesquisa documental**, pois foi elaborada uma base de dados com as informações históricas de uso dos assinantes juntamente com a elaboração da variável alvo, que será explicada na sequência. As consultas SQL de elaboração da base de dados final estão presentes nos apêndices A até D. Quanto à variável alvo, foi idealizada junto a colaboradores da empresa uma visão que maximiza os ganhos com o modelo, caso aplicado. O esquema da variável é apresentado a seguir (tabela 1):

Tabela 1 – Descrição dos registros da variável alvo

Registro / Variável Alvo	**Descrição**
0	Caso o assinante tenha se mantido ativo até o momento da análise. (não *churn*)
1	Caso o assinante tenha cancelado até o momento da análise. (*churn*)

Fonte: Elaborado pelo autor da pesquisa.

Para a definição do modelo a ser treinado, foram testados modelos de aprendizado supervisionado da ferramenta BigQuery ML. O link para a documentação dos modelos está no anexo A e os resultados de cada modelo serão apresentados na sessão de análise e discussão dos resultados deste artigo. A técnica de análise de dados foi a **análise estatística preditiva**. Com os dados de interpretabilidade do modelo foi realizada uma *backward elimination* para o modelo treinado que se deu da seguinte forma: um novo modelo foi treinado retirando, respectivamente, as 16 e posteriormente 26 *features* que menos tinham relevância para a acurácia do modelo. Os resultados da análise serão apresentados na sessão de análise e discussão dos resultados deste artigo.

A ferramenta utilizada para a elaboração e tratamento dos dados a serem utilizados nos modelos é o BigQuery que, de acordo com o site da ferramenta “é um *data warehouse*

corporativo completamente sem servidor e econômico, que funciona em várias nuvens e escalona de acordo com os dados usando BI, *machine learning* e IA integrados." Ainda sobre a ferramenta, o BigQuery tem um módulo chamado BigQuery ML, ao qual permite treinamento, teste e avaliação de modelos de aprendizado de máquina (GOOGLE CLOUD, 2023). O quadro 1 apresenta cada modelo utilizado neste artigo. A primeira coluna apresenta o nome do modelo na ferramenta e a segunda coluna o nome personalizado dado ao respectivo modelo treinado.

Quadro 1 – Modelos utilizados.

NOME DO MODELO NO BIGQUERY ML	NOME DADO AO MODELO TREINADO
BOOSTED_TREE_CLASSIFIER	CHURN_BOOSTED_TREE_CLASSIFIER
RANDOM_FOREST_CLASSIFIER	CHURN_RANDOM_FOREST_CLASSIFIER
DNN_LINEAR_COMBINED_CLASSIFIER	CHURN_DNN_LINEAR_COMBINED_CLASSIFIER
DNN_CLASSIFIER	CHURN_DNN_CLASSIFIER
LOGISTIC_REG	CHURN_LOGISTIC_REG

Fonte: Elaborado pelo autor da pesquisa.

Para a avaliação do modelo mais performático, foi utilizada a métrica Acurácia por ser uma métrica mais geral. O modelo que apresentou melhor acurácia com a totalidade dos dados de treino foi submetido a uma técnica chamada *Backward Elimination*. Neste estudo, este processo foi realizado restando apenas 5 variáveis além da variável alvo e os resultados deste processo serão apresentados e discutidos na seção seguinte (GOOGLE CLOUD, 2023).

Alguns modelos de aprendizado de máquina não suportam valores faltantes, ou "nulos", no treinamento e no teste, por isso é necessário realizar tratamento nos dados nulos para que não ocorra a perda de uma linha inteira de informação quando apenas a informação de uma coluna está faltando. No caso do BigQuery ML, o tratamento de dados nulos ocorre automaticamente antes do treinamento do modelo, não sendo necessário tratamento anterior. De acordo com a documentação da ferramenta, o BigQuery ML trata os valores nulos de acordo com o tipo de variável da coluna em questão. O quadro 2 apresenta a forma como cada tipo de variável é tratada.

Quadro 2 – Tratamento das variáveis.

Tipo de Coluna	**Método de tratamento**
Numérico	Tanto no treinamento quanto na previsão, os valores `NULL` em colunas numéricas são substituídos pelo valor médio calculado com os dados originais.
Codificação One-Hot e Multi-Hot	Tanto no treinamento quanto na previsão, os valores `NULL` nas colunas codificadas são mapeados para uma categoria adicional que é adicionada aos dados. Os dados não vistos anteriormente recebem um peso de 0 durante a previsão.
Temporal	As colunas temporais usam uma mistura de métodos de imputação de colunas padronizadas e de codificação one-hot. Para a coluna de tempo gerada, o BigQuery ML substitui os valores pelo tempo médio nas colunas originais. Para outros valores gerados, o BigQuery ML os atribui à respectiva categoria "NULL" para cada recurso extraído.
Struct	Tanto no treinamento quanto na predição, cada campo do STRUCT é imputado de acordo com seu tipo.

Fonte: Google Cloud (2023)

Da mesma forma, as demais colunas de informação precisam de um tratamento, ou “transformação” para que o modelo as interprete corretamente (GOOGLE CLOUD, 2023). O BigQuery ML trata cada tipo de variável da seguinte forma (quadro 3).

Quadro 3 - Pré-processamento automático de atributos

Tipo de dados de entrada	**Método de transformação**	**Detalhes**
INT64 NUMERIC BIGNUMERIC FLOAT64	Padronização	Para todas as colunas numéricas, o BigQuery ML padroniza e centraliza a coluna em zero antes de passá-la para o treinamento de todos os modelos, com exceção dos modelos de árvore aprimorada e floresta aleatória. Ao criar um modelo k-means, a opção STANDARDIZE_FEATURES especifica se os recursos numéricos devem ser padronizados.
BOOL STRING BYTES DATE DATETIME TIME	Codificação One-hot	Para todas as colunas não numéricas que não sejam de matriz, exceto TIMESTAMP, o BigQuery ML realiza uma transformação de codificação one-hot para todos os modelos, com exceção dos modelos de árvore aprimorada e floresta aleatória. Essa transformação gera uma nova coluna separada para cada valor exclusivo na coluna original. A Codificação One-hot é aplicada para treinar modelos de árvore e floresta aleatórios aprimorados para converter cada valor exclusivo em um valor numérico.
ARRAY	Codificação Multi-hot	Para todas as colunas ARRAY não numéricas, o BigQuery ML realiza uma transformação de codificação multi-hot. Essa transformação gera um recurso separado para cada elemento exclusivo no ARRAY.
TIMESTAMP	Transformação Temporal	Quando um modelo de regressão linear ou logística encontra uma coluna TIMESTAMP, ele extrai um conjunto de componentes do TIMESTAMP e executa uma mistura de padronização e codificação one-hot nos componentes extraídos. Para o componente de tempo em segundos, o BigQuery ML usa padronização. Para todos os outros componentes, ele usa a codificação one-hot.
STRUCT	Struct expansion	Quando o BigQuery ML encontra uma coluna STRUCT, ele expande os campos dentro do STRUCT para criar uma única coluna. Requer que todos os campos de STRUCT sejam nomeados. STRUCTs aninhados não são permitidos. Os nomes das colunas após a expansão estão no formato {struct_name}_{field_name}.
ARRAY of STRUCTs	Sem transformação	

Fonte: Google Cloud (2023)

Nesta próxima sessão, serão apresentados os resultados de cada um dos modelos de ML treinados inicialmente com todos os dados disponíveis, bem como o resultado dos modelos que passaram pelo processo de *Backward Elimination*.

4 ANÁLISE E DISCUSSÃO DOS RESULTADOS

Dos modelos testados, é apresentada a tabela 2, com a Precisão, *Recall* e Acurácia de cada um dos modelos, treinados com a totalidade dos dados disponíveis.

Tabela 2 – Métricas dos modelos treinados.

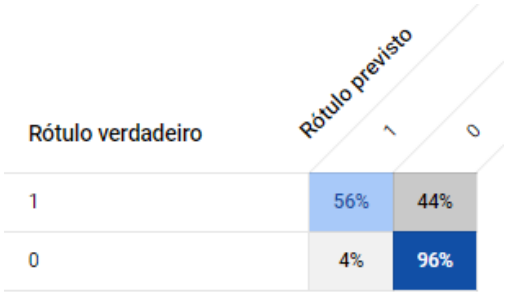
Métricas	BOOSTED TREE CLASSIFIER	RANDOM_FOREST_CLASSIFIER	DNN CLASSIFIER	DNN LINEAR COMBINED CLASSIFIER	LOGISTIC REGRESSION
Precisão	0,5344	0,501	0,5144	0,572	0,3613
Recall	0,564	0,5185	0,5062	0,4938	0,3239
Acurácia	0,9246	0,9276	0,921	0,9288	0,8985

Fonte: Elaborado pelo autor da pesquisa.

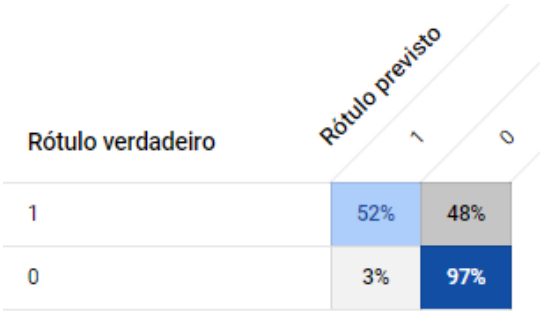
Observa-se que os modelos *Boosted Tree Classifier, Random Forest Classifier, Deep Neural Network Classifier* e *Deep Neural Network Linear Combined Classifier* tiveram resultados semelhantes, com precisões entre 0,5 e 0,57, recall entre 0,49 e 0,56 e acurácia por volta de 0,92. Quanto ao modelo de *logistic regression*, obteve-se resultados interiores, com precisão de 0,36, recall de 0,32 e acurácia de 0,89. Com isso é possível afirmar que apenas o modelo de regressão logística não é recomendado para aplicação de um modelo em produção pela empresa. Abaixo são apresentadas as matrizes de confusão de cada modelo inicialmente treinado (figura 1).

Figura 1 – Matrizes de Confusão dos Modelos treinados.

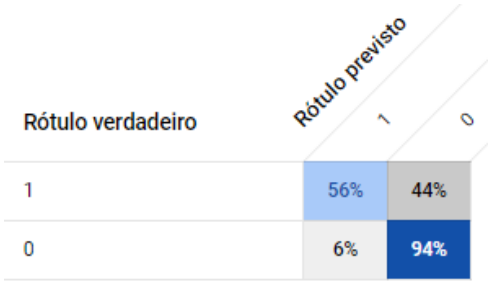
BOOSTED TREE CLASSIFIER



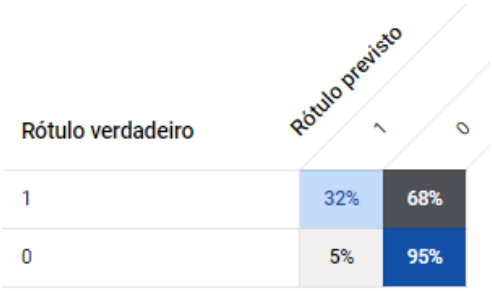
RANDOM_FOREST_CLASSIFIER



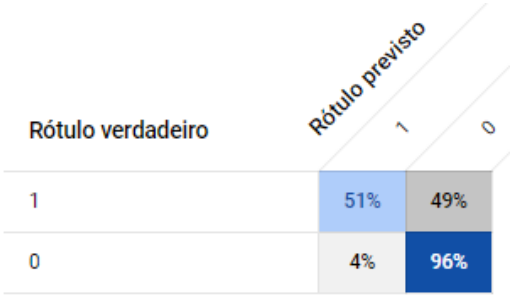
DNN LINEAR COMBINED CLASSIFIER



LOGISTIC REGRESSION



DNN CLASSIFIER

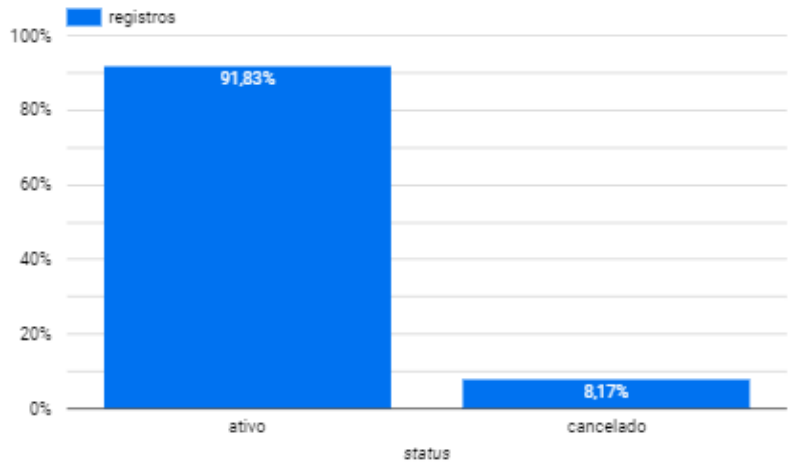


Fonte: Elaborado pelo autor da pesquisa.

Do modelo *Deep Neural Network Linear Combined Classifier* é importante observar que, pela matriz de confusão, ele previu corretamente que 56% iriam cancelar, enquanto previu incorretamente que 6% iriam cancelar. É possível também observar que o modelo previu corretamente a permanência de clientes em 94% das vezes. Isso mostra que o modelo aprendeu mais sobre clientes que permanecem ativos do que com os que cancelam. Isso pode ser explicado pelo desbalanceamento dos dados de treino do modelo, onde, dos 76520 assinantes utilizados no modelo, apenas 6251 haviam cancelado (figura 2). Inclusive, mesmo o modelo de Regressão Logística, que presentou métricas inferiores se comparado aos demais,

apresentou um alto percentual de acerto (95%) em relação aos que permanecem ativos. Em situação semelhante, o modelo de *Random Forest Classifier* apresentou o maior percentual entre os modelos inicialmente treinados (97%), mas nas métricas gerais não se destacou. Ocorre que para o objetivo de estudo, e em geral o da empresa, o fator mais importante é a identificação dos clientes com intenção de cancelamento, o que neste sentido não é o caso.

Figura 2 – Gráfico de colunas do volume % de registros por status de assinatura



Fonte: Elaborado pelo autor da pesquisa.

Interessante observar também que a maior diferença entre o modelo de regressão logística e os demais modelos é ao prever que 68% dos que cancelaram não iriam cancelar, enquanto os demais modelos ficaram abaixo de 50% deste quesito.

Com o objetivo de identificar as variáveis mais relevantes para a previsão de *churn* neste contexto, foi selecionado o modelo com maior acurácia, o *Deep Neural Network Linear Combined Classifier*, para a realização do processo de *backward elimination*. Para esse processo, as variáveis foram ordenadas pela atribuição dada pelo próprio modelo a cada uma delas e a eliminação foi realizada da menor para a maior atribuição. O modelo criado inicialmente apresentou a seguinte interpretabilidade (tabela 3):

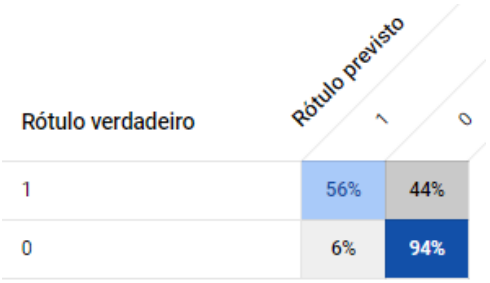
Tabela 3 – Atribuição por *feature* utilizada no modelo (nome do recurso).

Nome do recurso	Atribuição
newsletter	3,165
tempo_base_dias	2,737
top_so	1,272
SiglaModalidade	1,22
top_plataforma	0,876
top_browser	0,651
top_dispositivo	0,464
top_origem	0,417
top_subeditoria	0,282
top_editoria	0,267
dif_editoria	0,154
recencia	0,135
dif_dispositivo	0,125
dif_origem	0,124
dif_plataforma	0,123
dif_subeditoria	0,122
quinta	0,113
segunda	0,105
visitas_capa_app	0,103
manha	0,096
sabado	0,095
visitas_capa_site	0,091
terca	0,09
player_gaucha	0,083
quarta	0,081
noite	0,072
tarde	0,072
domingo	0,071
dias_navegados	0,069
busca	0,068
sexta	0,064
noticias_lidas	0,047

Fonte: Elaborado pelo autor da pesquisa.

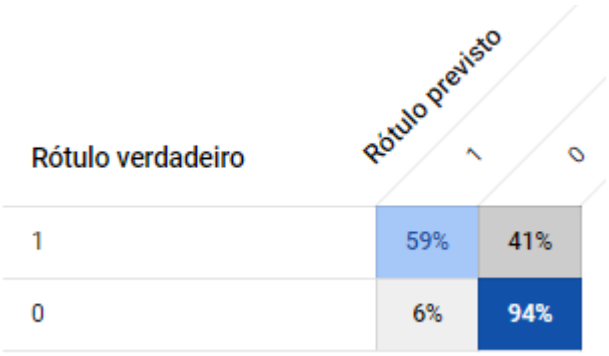
Com as variáveis classificadas de acordo com a interpretabilidade do BigQuery, o processo de *backward elimination* foi realizado e apresentou os resultados mostrados nos gráficos abaixo (figuras 3 a 5).

Figura 3 - Matriz de Confusão inicial do modelo.



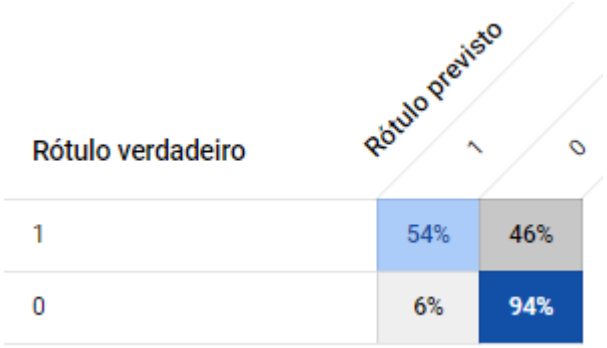
Fonte: Elaborado pelo autor da pesquisa.

Figura 4 - Matriz de Confusão do modelo treinado sem as 16 *features* menos relevantes.



Fonte: Elaborado pelo autor da pesquisa.

Figura 5 - Matriz de Confusão do modelo treinado sem outras 10 *features* menos relevantes.



Fonte: Elaborado pelo autor da pesquisa.

Com isso, observa-se que não houve mudança de performance para os assinantes que não cancelaram, pois tanto o modelo inicial quanto os demais previram corretamente em 94% dos casos. Outro ponto de destaque é o fato de o segundo modelo, treinado com 16 *features*, teve melhor performance, 59% de acerto em cancelados, que o terceiro modelo, treinado com 6 *features*.

5 CONSIDERAÇÕES FINAIS

Com este estudo foi possível identificar, no contexto da empresa objeto de estudo, quais os melhores algoritmos disponíveis na ferramenta BigQuery para a previsão de *churn*, bem como identificar as variáveis mais relevantes para a construção do modelo. Sendo assim, é possível afirmar que o objetivo deste artigo foi atingido.

É importante observar, porém, que os resultados aqui apresentados foram obtidos a partir de dados disponibilizados pela empresa e que técnicas como *feature engineering* poderiam ser melhor aplicadas a fim de aumentar o número de variáveis estudadas e potencialmente os resultados dos modelos.

Além disso, por se tratar de um caso onde fez-se necessária a utilização de uma ferramenta específica, algumas limitações de pré-processamento de dados, tratamento de nulos e interpretabilidade foram impostas durante o estudo. Vale também destacar possíveis limitações devido ao fato de o pesquisador responsável por este artigo não ser um especialista na área de ciência de dados.

O presente artigo traz contribuições acadêmicas no contexto da exploração e mensuração de modelos de *machine learning*, bem como na expansão do estudo sobre retenção de clientes no contexto de serviços de assinatura. Além disso, traz contribuições empresariais, pois elucida para a empresa objeto de estudo os modelos mais eficientes para a previsão de *churn*, bem como as variáveis de comportamento que mais interferem na identificação de potenciais clientes com intenção de assinatura.

Para futuros estudos sugere-se a aplicação de técnicas como *walk forward cross-validation*, que permitirá ao modelo testar apenas com dados posteriores aos dados de treino em cada iteração da validação cruzada.

REFERÊNCIAS

AHMAD, A. K.; JAFAR, A.; ALJOUMAA, K. Customer churn prediction in telecom using machine learning in big data platform. **Journal of Big Data**, v. 6, n. 1, p. 1-24, 2019.

ALBUQUERQUE, I. G. C. de. et al. CHURN RATE: como reduzir em empresas de telecomunicações utilizando aprendizado de máquina. **Revista Interface Tecnológica**, v. 18, n. 2, p. 40-52, 2021.

ALMEIDA, M. et al. A Temporal Approach to Customer Churn Prediction: A Case Study for Financial Services. *In*: Encontro Nacional de Inteligência Artificial e Computacional, 19., 2022, Campinas. **Anais** [...]. Porto Alegre, RS: SBC, 2022. p. 83-94.

AL-NAJJAR, D.; AL-ROUSAN, N.; AL-NAJJAR, H. Machine Learning to develop credit card customer churn prediction. **Journal of Theoretical and Applied Electronic Commerce Research**, v. 17, n. 4, p. 1529-1542, 2022.

COIMBRA, G. T. P. et al. AutoRGNN: um Modelo de Previsão de Churn que preserva a Privacidade com uma Abordagem Híbrida de Aprendizado Profundo. **Anais do Computer on the Beach**, v. 14, p. 062-069, 2023.

ÇELIK, O.; OSMANOGLU, U O. Comparing to techniques used in customer churn analysis. **Journal of Multidisciplinary Developments**, v. 4, n. 1, p. 30-38, 2019.

CHAUDHARY, M. et al. Envisaging employee churn using MCDM and machine learning. **Intelligent Automation & Soft Computing**, v. 33, n. 2, p. 1009-1024, 2022.

GEILER, L.; AFFELDT, S.; NADIF, M. A survey on machine learning methods for churn prediction. **International Journal of Data Science and Analytics**, v. 14, n. 3, p. 217-242, 2022.

JADHAV, A. et al. Churn prediction of employees using machine learning techniques. **Tehnički Glasnik**, v. 15, n. 1, p. 51-59, 2021.

JAIN, N.; TOMAR, A.; JANA, P. K. A novel scheme for employee churn problem using multi-attribute decision making approach and machine learning. **Journal of Intelligent Information Systems**, v. 56, p. 279-302, 2021.

KAVITHA, V. et al. Churn prediction of customer in telecom industry using machine learning algorithms. International **Journal of Engineering Research & Technology (IJERT)**, v. 9, n. 5, p. 181-184, 2020.

KHODABANDEHLOU, S.; ZIVARI RAHMAN, M. Comparison of supervised machine learning techniques for customer churn prediction based on analysis of customer behavior. **Journal of Systems and Information Technology**, v. 19, n. 1/2, p. 65-93, 2017.

LALWANI, P. et al. Customer churn prediction system: a machine learning approach. **Computing**, p. 1-24, 2022.

LEMOS, R. A. de. L.; SILVA, T. C.; TABAK, B. M. Propension to customer churn in a financial institution: a machine learning approach. **Neural Computing and Applications**, v. 34, n. 14, p. 11751-11768, 2022.

MATUSZELAŃSKI, K.; KOPCZEWSKA, K. Customer churn in retail e-Commerce business: spatial and Machine Learning approach. **Journal of Theoretical and Applied Electronic Commerce Research**, v. 17, n. 1, p. 165-198, 2022.

GOOGLE CLOUD. O que é o ML do BigQuery? 2023. Disponível em: https://cloud.google.com/bigquery/docs/bqml-introduction?hl=pt-br. Acesso em 07-jun-2023.

PIRROLAS, O. A. C.; CORREIA, P. M. A. R. O churning aplicado à gestão de recursos humanos: a importância de um modelo de previsão. **Lex Humana**, v. 12, n. 1, p. 59-68, 2020.

GOOGLE CLOUD. Pré-Processamento automático de atributos. 2023. Disponível em: https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-auto-preprocessing. Acesso em 07-jun-2023.

SARAN KUMAR, A.; CHANDRAKALA, D. A survey on customer churn prediction using machine learning techniques. **International Journal of Computer Applications**, v. 154, n. 10, p. 1-4, 2016.

SRIVASTAVA, P. R.; EACHEMPATI, P. Intelligent employee retention system for attrition rate analysis and churn prediction: An ensemble machine learning and multi-criteria decision-making approach. **Journal of Global Information Management (JGIM)**, v. 29, n. 6, p. 1-29, 2021.

ULLAH, I. et al. A churn prediction model using random forest: analysis of machine learning techniques for churn prediction and factor identification in telecom sector. **IEEE Access**, v. 7, p. 60134-60149, 2019.

VAFEIADIS, T. et al. A comparison of machine learning techniques for customer churn prediction. **Simulation Modelling Practice and Theory**, v. 55, p. 1-9, 2015.

APÊNDICE A – Primeira consulta SQL utilizada para a elaboração do *dataset*.

```
create or replace table `company-datalake-dev.tcc.assinantes` as

with

base_canc as
(
select
  carteira.cdass,
  carteira.cdjornal,
  min(carteira.DtUltCancelamento) primcanc
from
  `company-datalake-prd.as_consolidado.carteira_total_ass` carteira
where
  carteira.FlgDegustacao       = 'N'
  and carteira.IdCortesia      = 'N'
  and carteira.Parcerias       = 'N'
  and carteira.PgtoContabil    = 'SIM'
  and carteira.CdJornal        = 1
  and date_trunc(date(carteira.DtVenda), month) >= '2020-11-27'
  and carteira.nometipossituass != 'DESATIVADO PARA SEMPRE'
group by
  1, 2
),

ultperiodo as
(
select
  base_canc.cdass,
  base_canc.cdjornal,
  base_canc.primcanc,
  max(carteira.cdperiodoatual) ultperiodo
from
  base_canc
  inner join `company-datalake-prd.as_consolidado.carteira_total_ass` carteira
    on
```

```
    (
      base_canc.cdass       = carteira.cdass
      and base_canc.cdjornal = carteira.cdjornal
      and ifnull(base_canc.primcanc, current_timestamp()) =
ifnull(carteira.dtultcancelamento, current_timestamp())
    )
group by
  1, 2, 3
),

ultdtcarteira as
(
select
  ultperiodo.cdass,
  ultperiodo.cdjornal,
  ultperiodo.primcanc,
  ultperiodo.ultperiodo,
  max(carteira.dtcarteira) ultdtcarteira
from
  ultperiodo
  inner join `company-datalake-prd.as_consolidado.carteira_total_ass` carteira
    on
    (
      ultperiodo.cdass         = carteira.cdass
      and ultperiodo.cdjornal   = carteira.cdjornal
      and ifnull(ultperiodo.primcanc, current_timestamp()) =
ifnull(carteira.dtultcancelamento, current_timestamp())
      and ultperiodo.ultperiodo = carteira.cdperiodoatual
    )
group by
  1, 2, 3, 4
),

carteira as
(
select
  carteira.*
```

```
from
  ultdtcarteira
  inner join `rbs-datalake-prd.as_consolidado.carteira_total_ass` carteira
    on
    (
      ultdtcarteira.cdass          = carteira.cdass
      and ultdtcarteira.cdjornal   = carteira.cdjornal
      and ifnull(ultdtcarteira.primcanc, current_timestamp()) =
ifnull(carteira.dtultcancelamento, current_timestamp())
      and ultdtcarteira.ultperiodo = carteira.cdperiodoatual
      and ultdtcarteira.ultdtcarteira = carteira.dtcarteira
    )
),

nossa as
(
select
  right(concat('00000000000000', ifnull(nossa.as_key, ifnull(nossa.cpf, nossa.cnpj))), 14)
chave_as,
  nossa.id,
  array_to_string(array_agg(distinct nossa.origem_assinatura order by
nossa.origem_assinatura), ',', '--') origens
from
  `company-datalake-prd.nossa.v_nossa` nossa
group by
  1, 2
),

carteira_final as (
select
  *
from
  carteira
  left join nossa
    on
    (
      right(concat('00000000000000', cast(ifnull(carteira.cpf, carteira.cnpj) as string)), 14)
```

```
    =
    nossa.chave_as
  ))

select
  date(DtVenda)DtVenda,
  produtoImpDig,
  SiglaModalidade,
  NomeCondPg,
  CanalVenda,
  date(DtUltCancelamento)DtUltCancelamento,
  cdass,
  id
from carteira_final
where id is not null
and NomeSituacao not like 'DESATIVADO PARA SEMPRE'
```

APÊNDICE B – Segunda consulta SQL utilizada para a elaboração do *dataset*.

```
create or replace table `rbs-datalake-dev.tcc.consolidado_1` as

with

-- Busca os assinantes a serem avaliados.
assinantes as (select * from `company-datalake-dev.tcc.assinantes`),

-- Frequência: Volume de dias diferentes em que o assinante navegou.
frequencia_assinante as (
  select
    ass.*,
    count(distinct(date(serverTimestamp))) as dias_navegados
  from `company-datalake-prd.zm.events2` zm
    right join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
    and upper(produto) != "product2"
```

```
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- Volume: Volume de notícias lidas pelo assinante.
-- Também a editoria e subeditoria mais acessadas e quantas editorias e subeditorias
diferentes o assinante acessou no período.
volume_assinante as (
  select
    ass.*,
    count(eventType) as noticias_lidas,
    case
      when (approx_top_count(editoria, 1)[offset(0)].value is null and
array_length(approx_top_count(editoria, 2)) >= 2)
      then approx_top_count(editoria, 2)[offset(1)].value
      else approx_top_count(editoria, 1)[offset(0)].value
      end top_editoria,
    count(distinct(editoria)) as dif_editoria,
    case
      when (approx_top_count(subeditoria, 1)[offset(0)].value is null and
array_length(approx_top_count(subeditoria, 2)) >= 2)
      then approx_top_count(subeditoria, 2)[offset(1)].value
      else approx_top_count(subeditoria, 1)[offset(0)].value
      end top_subeditoria,
    count(distinct(subeditoria)) as dif_subeditoria
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
```

```
    and lower(eventType) in ("visitou conteudo")
    and lower(produto) != "product2"
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- Recencia: Número de dias entre o último acesso e a data da assinatura.
recencia_assinante as (
  select
    ass.*,
    date_diff(date(max(serverTimestamp)), date(dtvenda), day) as recencia
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
    and lower(eventType) in ("visitou conteudo")
    and lower(produto) != "product2"
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- Volume de visitas a capa do site.
```

```
visitas_capa_site as (
  select
    ass.*,
    count(eventType) AS visitas_capa_site
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
    and lower(eventType) in ("visitou capa")
    and lower(produto) != "product2"
    and lower(plataforma) = "site"
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- Volume de visitas a capa do app.
visitas_capa_app as (
  select
    ass.*,
    count(eventType) as visitas_capa_app
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
    and lower(eventType) in ("visitou capa")
    and lower(produto) != "product2"
    and lower(plataforma) = "app"
  group by
```

```
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- Volume de buscas realizadas.
buscas as (
  select
    ass.*,
    count(eventType) as busca
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
  where (lower(eventType) in ("usou busca") or lower(eventType) in ("usou a busca"))
  and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda), interval
30 day)
  and lower(produto) != "product2"
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- Outras features diversas de uso avaliando so, browser, dispositivo, plataforma e origem.
uso_diverso as (
  select
    ass.*,
```

```
    case
      when (approx_top_count(so, 1)[offset(0)].value is null and
array_length(approx_top_count(so, 2)) >= 2)
      then approx_top_count(so, 2)[offset(1)].value
      else approx_top_count(so, 1)[offset(0)].value
      end top_so,
    case
      when (approx_top_count(browser, 1)[offset(0)].value is null and
array_length(approx_top_count(browser, 2)) >= 2)
      then approx_top_count(browser, 2)[offset(1)].value
      else approx_top_count(browser, 1)[offset(0)].value
      end top_browser,
    case
      when (approx_top_count(dispositivo, 1)[offset(0)].value is null and
array_length(approx_top_count(dispositivo, 2)) >= 2)
      then approx_top_count(dispositivo, 2)[offset(1)].value
      else approx_top_count(dispositivo, 1)[offset(0)].value
      end top_dispositivo,
    count(distinct(dispositivo)) as dif_dispositivo,
    case
      when (approx_top_count(plataforma, 1)[offset(0)].value is null and
array_length(approx_top_count(plataforma, 2)) >= 2)
      then approx_top_count(plataforma, 2)[offset(1)].value
      else approx_top_count(plataforma, 1)[offset(0)].value
      end top_plataforma,
    count(distinct(plataforma)) as dif_plataforma,
    case
      when DtUltCancelamento is not null
      then date_diff(date(DtUltCancelamento), date(dtvenda), day)
      else date_diff(current_date(), date(dtvenda), day)
      end as tempo_base_dias,
    case
      when (approx_top_count(origem, 1)[offset(0)].value is null and
array_length(APPROX_TOP_COUNT(origem, 2)) >= 2)
      then approx_top_count(origem, 2)[offset(1)].value
      else approx_top_count(origem, 1)[offset(0)].value
      end top_origem,
```

```
    count(distinct(origem)) as dif_origem
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
    and lower(produto) != "product2"
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- Volume de ações realizadas as quais a origem é a newsletter.
newsletter as (
  select
    ass.*,
    count(origem) AS newsletter
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
    and lower(produto) != "product2"
    and lower(origem) = "newsletter"
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
```

```
    cdAss,
    id
),

-- Volume de plays na rádio.
player_radio as (
  select
    ass.*,
    count(eventType) as player_radio
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
    and lower(produto) != "product2"
    and lower(eventType) = "player ao vivo"
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- União entre todas as features elaboradas na query.
consolidado_1 as (
  select
  ass.CdAss,
  ass.SiglaModalidade,
  ass.DtVenda,
  ass.DtUltCancelamento,
  ass.id,
  dias_navegados,
  noticias_lidas,
```

```
  top_editoria,
  dif_editoria,
  top_subeditoria,
  dif_subeditoria,
  recencia,
  visitas_capa_site,
  visitas_capa_app,
  busca,
  top_so,
  top_browser,
  top_dispositivo,
  dif_dispositivo,
  top_plataforma,
  dif_plataforma,
  tempo_base_dias,
  top_origem,
  dif_origem,
  newsletter,
  player_radio
  from assinantes ass
  left join frequencia_assinante f    on ass.id = f.id    and ass.CdAss = f.CdAss
  left join volume_assinante v        on ass.id = v.id    and ass.CdAss = v.CdAss
  left join recencia_assinante r      on ass.id = r.id    and ass.CdAss = r.CdAss
  left join visitas_capa_site vcs     on ass.id = vcs.id  and ass.CdAss = vcs.CdAss
  left join visitas_capa_app vca      on ass.id = vca.id  and ass.CdAss = vca.CdAss
  left join buscas b                on ass.id = b.id    and ass.CdAss = b.CdAss
  left join uso_diverso ud            on ass.id = ud.id   and ass.CdAss = ud.CdAss
  left join newsletter nl             on ass.id = nl.id   and ass.CdAss = nl.CdAss
  left join player_gaucha pg          on ass.id = pg.id   and ass.CdAss = pg.CdAss
)

select distinct * from consolidado_1 c
```

APÊNDICE C – Terceira consulta SQL utilizada para a elaboração do *dataset*.

```
create or replace table `company-datalake-dev.tcc.consolidado_2` as

with

-- Busca os assinantes a serem avaliados.
assinantes as (select * from `company-datalake-dev.tcc.assinantes`),

-- Traz o dia da semana e o horário em que cada evento do assinante ocorreu.
contagem_dias_ass as (
  select
    ass.*,
    extract(dayofweek from date(serverTimestamp)) as dias,
    extract(hour from serverTimestamp) as tempo
  from `company-datalake-prd.zm.events2` zm
    inner join assinantes ass
    on zm.userId = ass.id
    and date(serverTimestamp) between date(DtVenda) and date_add(date(DtVenda),
interval 30 day)
  and lower(produto) != "product2"
),

-- Contagem de vezes em que o assinante realizou ações em cada dia da semana e turno do
dia.
semana_turno_ass as (
  select
    ass.* except(dias,tempo),
    countif(dias=1) as domingo,
    countif(dias=2) as segunda,
    countif(dias=3) as terca,
    countif(dias=4) as quarta,
    countif(dias=5) as quinta,
    countif(dias=6) as sexta,
    countif(dias=7) as sabado,
    countif((tempo >= 20 AND tempo <= 23) or (tempo >= 0 AND tempo <= 5)) noite,
    countif(tempo >= 13  AND tempo <= 19) as tarde,
```

```
    countif(tempo >= 6  AND tempo <= 12) as manha
  from contagem_dias_ass ass
  group by
    DtVenda,
    produtoImpDig,
    SiglaModalidade,
    NomeCondPg,
    CanalVenda,
    DtUltCancelamento,
    cdAss,
    id
),

-- Consolida as features criadas na query.
consolidado_2 as (
select distinct
  ass.CdAss,
  ass.SiglaModalidade,
  ass.DtVenda,
  ass.DtUltCancelamento,
  ass.id,
  manha,
  tarde,
  noite,
  domingo,
  segunda,
  terca,
  quarta,
  quinta,
  sexta,
  sabado
from assinantes ass
left join semana_turno_ass sta
on ass.id = sta.id and
ass.SiglaModalidade = sta.SiglaModalidade)

select distinct * from consolidado_2
```

APÊNDICE D – Quarta consulta SQL utilizada para a elaboração do *dataset*.

```
create or replace table `company-datalake-dev.tcc.consolidado_final` as

with

c1 as (select * from `company-datalake-dev.tcc.consolidado_1`),
c2 as (select * from `company-datalake-dev.tcc.consolidado_2`),

aux_consolidado as (
select
c1.id,
c1.DtUltCancelamento,
c1.SiglaModalidade,
c1.dias_navegados,
c1.noticias_lidas,
c1.top_editoria,
c1.dif_editoria,
c1.top_subeditoria,
c1.dif_subeditoria,
c1.recencia,
c1.visitas_capa_site,
c1.visitas_capa_app,
c1.busca,
c1.top_so,
c1.top_browser,
c1.top_dispositivo,
c1.dif_dispositivo,
c1.top_plataforma,
c1.dif_plataforma,
c1.tempo_base_dias,
c1.top_origem,
c1.dif_origem,
c1.newsletter,
c1.player_radio,
c2.manha,
c2.tarde,
```

```
c2.noite,
c2.domingo,
c2.segunda,
c2.terca,
c2.quarta,
c2.quinta,
c2.sexta,
c2.sabado
from c1
inner join c2
on
  c1.id =                  c2.id and
  c1.SiglaModalidade =       c2.SiglaModalidade
)

select distinct * from aux_consolidado
```

ANEXO A - Lista de links para a documentação do BigQuery ML.

Instrução CREATE MODEL:
https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-create?hl=pt-br

Jornada do usuário completa para cada modelo:
https://cloud.google.com/bigquery/docs/reference/standard-sql/bigqueryml-syntax-e2e-journey?hl=pt-br