**Modelos de “Machine Learning” aplicados a “churn” de clientes - estudo de caso**

Emerson Donizeti Batista¹\*;Mateus Modesto2

1 Analista de Sistemas. Rua Alberto Segalla, 1-33 – apto 411 – Jd. Infante Dom Henrique. 17012-634 Bauru, São Paulo, Brasil.

**Modelos de “Machine Learning” aplicados a “churn*”* de clientes - estudo de caso**

**Resumo**

Devido aos constantes avanços nas áreas do conhecimento direcionados à previsão dos comportamentos dos clientes em diversas áreas, a utilização de modelos matemáticos se mostra útil e necessária para se construir modelos preditivos com relação a “churn” de clientes, ou seja, a previsão de que um cliente possa a vir a cancelar um serviço. Este trabalho aplicou o modelo de “Generalized Linear Model” [GLM] e os algoritmos de “Decision Tree” e “Random Forest” numa base de dados de cliente de telecomunicação. Foram analisados indicadores como “Log Likelihood” [LL], “Akaike Information Criterion” [AIC], Sensibilidade, Sensitividade, Acuracidade, ROC (AUC), observando-se que é possível utilizar modelos matemáticos e algoritmos de “machine learning”, obtendo acuracidade em torno de 79% na base estudada.

**Palavras-chave:** Regressão Logística, Decision Tree, Random Forest, Churn

**Machine Learning models applied to customer churn - case study**

**Abstract**

Due to the constant advances in the areas of knowledge aimed at predicting customer behavior in several areas, the use of mathematical models proves to be useful and necessary to build predictive models regarding customer churn, that is, the prediction that a customer may cancel a service. This work applied the “Generalized Linear Model” [GLM] and the “Decision Tree” and “Random Forest” algorithms in a telecommunication customer database. Indicators such as “Log Likelihood” [LL], “Akaike Information Criterion” [AIC], Sensitivity, Sensitivity, Accuracy, ROC (AUC) were analyzed, and it was concluded that it is possible to use mathematical models and “machine learning” algorithms. obtaining accuracy around 79% in the studied base.

**Keywords:** Logistic Regression, Decision Tree, Random Forest, Churn

**Introdução**

É geralmente mais caro conquistar novos clientes do que manter os já existentes. Em muitos setores, o custo de aquisição de novos clientes pode ser cinco vezes superior ao seu custo de retenção (Kurtz e Clow, 1998). Na indústria de telecomunicações, por exemplo, o custo de aquisição foi empiricamente auferido entre cinco e oito vezes superior ao custo de retenção do cliente (AU e outros., 2003).

Conhecer melhor o cliente é o desafio para qualquer organização que busca elevar seu nível de atendimento, bem como seus resultados financeiros. Uma vez indicados os fatores críticos de sucesso, seja por pesquisas ou análise de bases-de-dados, emerge a necessidade de obter predição e inteligência através de modelos matemáticos e estatísticos.

Por exemplo, em vez de focar em todos os “prospects” ou fornecer incentivos iguais para todos, uma empresa pode selecionar apenas aqueles clientes que se enquadram em certos critérios de rentabilidade, baseados nas necessidades individuais e padrões de compra (PriceWaterHouseCoopers, 1999). Neste sentido, reforçam Bounsaythip e Rinta-Runsala (2001) que isso pode ser conseguido através da construção de modelos de predição do valor futuro de um indivíduo baseado em dados demográficos, estilo de vida ou comportamentos anteriores. O modelo produz informações que irão focar tanto a retenção quanto programas de recrutamento, a fim de construir e manter a base de clientes mais rentável. A isso se dá o nome de “customer behavior modeling”[CBM]ou perfil de cliente. A identificação do perfil de cliente é uma ferramenta para focar no melhor entendimento de marketing sobre as características de sua base de dados.

A motivação de longo prazo para a montagem de perfis de consumidores é converter este entendimento numa interação automatizada com esses consumidores, como afirmam Berson & Thearling (2001). Estes reforçam que para estas tarefas os mercados precisam de vasta gama de processos e ferramentas de tecnologia. Estas ferramentas têm sido usadas para coletar dados e simplificar os processos de extrair conhecimento sobre o mercado e planejar campanhas de marketing. As ferramentas de mineração de dados têm sido usadas para identificar grupos significativos em dados históricos. Por exemplo, aplicam-se a critérios de seleção para mala-direta, ou identificar mercados com alto potencial, ou mídias ou estilos de vida que vão ao encontro dos consumidores. Em suma, tais ferramentas possibilitam encontrar padrões humanos de comportamento que descrevem os dados; eles também permitem utilizar algumas variáveis para predizer valores futuros ou desconhecidos de outras variáveis.

Assim, a busca por compreensão dos mecanismos de explicação para tomada de decisões dos clientes, aplicada à questão dos cancelamentos (“churn”), se alinha com esta necessidade premente de se compreender o comportamento do cliente. A proposta será de discutir modelo matemático estatístico para ajudar na compreensão do fenômeno, ainda que existam vários outros, pois, conforme salienta (Dio, 1979) se a verdade é uma só – ainda que, por vezes, vista de ângulos diferentes –, os caminhos que conduzem os pesquisadores a ela podem ser diversos. E a diversidade de métodos, mais do que um inconveniente, é uma vantagem. Sendo assim, quando, por técnicas ou processos diferentes, se chega à mesma conclusão, há maior razão para aceitá-la.

Analisando as plataformas online de publicações acadêmicas (Google Scholar, Elsevier e SciELO), encontramos em torno de 18.000 trabalhos sobre o “churn” de clientes da área de telecom desde 2017. Eis alguns exemplos: "A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector" (Ullah e outros, 2019). Foi utilizada também a técnica de boosting no trabalho "A Customer Churn Prediction Model in Telecom Industry Using Boosting" (LU, 2012). Inúmeros outros trabalhos tiveram outras abordagens, reforçando a importância do tema e da atenção que tem merecido entre os pesquisadores.

Especificamente a utilização de GLM [Generalized Linear Models] para análise de "churn" já foi aplicada em trabalhos acadêmicos por (McGowan, Donal et al., 2011), (Dash, Singh, Kunwar & Ravendra) e por (Stripling e outros, 2015), oferecendo informações significativas de incentivo à continuidade de investigação do tema.

Da mesma família dos GLM, a técnica multinível também já foi aplicada na análise de “churn” por (Ahmed & Linen, 2017) e no trabalho realizado por (Seo, Ranganathan, Babad, 2008) que focaram na análise de “churn” em área de “telecom”. Portanto, essa linha de análise com utilização de métodos da família GLM já possui importantes trabalhos anteriores realizados. No caso específico deste trabalho, faremos avaliação da melhor assertividade e acurácia dos resultados entre os dois modelos. Neste sentido, o resultado deste trabalho poderá reforçar, ou refutar, a aplicabilidade destas técnicas neste tipo de base de dados, pois algum dado presente pode induzir a vieses não detectadas em trabalhos anteriores.

Por fim, cumpre ressaltar que em ambientes corporativos altamente digitalizados, com informações coletadas em volume e diversidade, sabe-se que as empresas possuem um grande tesouro em suas mãos, que são os dados. Obter, analisar e compreender esses dados tem se mostrado a nova corrida do ouro, em que os mais eficazes obterão os melhores resultados e geração de riqueza. Neste sentido, concordam (Tjaden & Cohen, 2006) pois, em que pesem a enorme gama de fontes de informação, observa-se que se torna inviável, para uma grande quantidade de dados, a extração de informações sem o uso de ferramentas computacionais. Refinar e enriquecer os critérios de escolha de métodos de análises de dados reforçam a necessidade de mais trabalhos nesta linha de pesquisa.

**Materiais e Métodos**

**Coleta de Dados**

Os dados do problema foram obtidos da plataforma online Kaggle (“site”) de uma empresa no ramo de telecomunicações (*Telco*). Todos os códigos fontes utilizados nesta implementação se encontram disponíveis no repositório do “Github” através do link disponível no Anexo 1.

A análise exploratória, bem como os testes foram feitos através da linguagem R, utilizando diversos pacotes auxiliares. As variáveis do problema foram essas:

Tabela 1 – Lista das variáveis presentes na base (continua)

|  |  |
| --- | --- |
| **Variável** | **Categoria** |
| **CustomerId** | Identificação da linha |
| **Gender** (Gênero) | dicotômica (*male* / *female*) |
| **SeniorCitzen** (Idoso) | dicotômica (0-1) |
| **Partner** (possui parceiro) | dicotômica (0-1) |
| **Dependents** (possui dependentes) | dicotômica (0-1) |
| **Tenure** (meses de uso do serviço) | quantitativa discreta |
| **Phone Service**(possui serviço de telefone) | dicotômica (yes-no) |
| **Multiple Lines** (possui múltiplas linhas) | policotômica |
| **InternetService**  (tipos de serviços de internet) | policotômica |
| **Online Security**  (possui serviço de segurança online) | policotômica |
| **OnlineBackup**  (possui backup online) | policotômica |
| **DeviceProtection**  (possui dispositivo de proteção) | policotômica |
| **TechSupport**  (possui suporte técnico) | policotômica |
| **StreamingTV**  (possui streaming de TV) | policotômica |
| **StreamingMovies**  (possui streaming de filmes) | policotômica |
| **Contract**  (periodicidade do contrato) | policotômica |
| Tabela 1 – Lista das variáveis presentes na base (continuação) | |
| **Variável** | Categoria |
| **PaperlessBilling**  (fatura eletrônica) | dicotômica |
| **PaymentMethod**  (forma de pagamento) | policotômica |
| **MonthlyCharges**  (valor da fatura mensal) | quantitativa contínua |
| **TotalCharges**  (total faturado) | quantitativa contínua |
| **Churn** | dicotômica |

Fonte: Resultados originais da pesquisa.

Para avaliar o evento estudado, foram escolhidos 3 (três) modelos de “Machine Learning” para avaliar e comparar seus resultados:

**Generalized Linear Models [GLM]**

Desde a década de 70, foi possível o desenvolvimento computacional de modelos que exigiam iteração na estimação de parâmetros. Assim, (McCullagh & Nelder, 1989) propuseram os “Generalized Linear Models” [GLM] que consistia na ampliação da distribuição da variável de resposta, de forma que pertencesse à família exponencial de distribuições, dando flexibilidade para a relação funcional entre a média da variável de resposta (μ) e o preditor linear (η). O resultado destes estudos permitiu aplicar, por exemplo, transformação na variável de resposta Y para normalizá-la utilizando a distribuição Poisson em dados de contagem, fazendo a relação funcional entre a média de Y e o preditor linear η = log μ. Isso permitiu melhoria na estimação, já que garantiria um valor positivo para quaisquer valores dos parâmetros do preditor linear.

Segundo (Fávero & Belfiore, 2017) a Regressão Logística Binária e Multinomial consiste no modelo em que há mais de duas variáveis dependentes, categóricas, binárias e variáveis independentes métricas ou não métricas. Além disso, esta relação expressa a probabilidade de ocorrência do evento diante das variáveis às quais os objetos ou sujeitos estão expostos, sendo possível uma taxa de risco de cada variável explicativa do modelo. O objetivo principal aqui é encontrar a função logística, obtida por meio da ponderação das variáveis, permitindo dar peso a estas ocorrências.

Para estimação dos parâmetros, foi aplicado o processo iterativo chamado Método da Máxima Verossimilhança, ou “Maximum Likelihood Estimation”. Em seguida, foram avaliados o teste de χ2 e os resultados dos testes z para os parâmetros estimados das variáveis preditoras.

Na equação abaixo está o modelo geral da regressão Logística Binária e Multinomial, a probabilidade de ocorrência do evento, temos que:

(1)

Probabilidade de ocorrência da categoria 0 (referência):

(2)

Probabilidade de ocorrência da categoria 1:

(3)

Probabilidade de ocorrência da categoria 2:

(4)

Nas fórmulas (1), (2), (3) e (4), o é a probabilidade de ocorrer o evento e é o *logito*.

A performance final geral aferida no modelo GLM foi medida através do “Log Likelihood” conforme descrito em Woolf (1957).

Foi analisada também curva “Receiver Operating Characteristic” [ROC]. A curva ROC é uma métrica para avaliação de modelos, que permite estudar a variação para as medidas de sensibilidade e especificidade do modelo, para diferentes valores de ponto de corte (Fawcett, 2006).

Além destes utilizamos AIC e BIC para analisar modelos. Modelo com AIC menor do que outro, o BIC vai ser menor também. Estes conceitos estão discutidos em Penny (2012) em que discute a avaliação de performance de algoritmos através de “Bayesian information criterion” [BIC] e “Akaike Information Criterion” [AIC].

A fim de analisar multicolinearidade das variáveis, foi analisado o Fator de Inflação da Variância (VIF), conforme descrito por (Montgomery; Peck; Vining, 2006), cuja fórmula é:

(5)

Nesta fórmula, p é o número das variáveis preditoras; é o coeficiente de correlação múltipla, resultante da regressão de Xj nos outros p-1 regressores. A matriz de variâncias e covariâncias para as estimativas dos coeficientes de regressão padronizados é:

(6)

Se o valor de o valor for próximo a um, isto significa que existe uma alta correlação entre a variável Xj e as demais variáveis, então 1- estará próximo de zero e consequentemente, o VIF assumirá um valor grande, apontando para o envolvimento dessa covariável em colinearidades. Um VIF máximo acima de 10 indica que a multicolinearidade pode estar influenciando as estimativas de mínimos quadrados (Montgomery; Peck; Vining, 2006).

Para verificar quais variáveis foram úteis, utilizamos utilizar o fator de inflação (VIF – Variance Inflation Factor) para eliminar as preditoras redundantes, ou variáveis que possuem multicolinearidade entre elas. Sabemos que multicolinearidade existe quando duas ou mais preditoras estão altamente relacionados entre si o que torna difícil de se identificar o impacto de uma variável independente na variável dependente.

O fator de inflação de variância é usado para medir colinearidade entre o preditor e as variáveis em um modelo. O preditor tendo VIF de 2 (dois) ou menos geralmente é considerado seguro e pode ser assumido que não é correlacionado com outra variável preditora. Quanto maior o VIF, maior é a correlação da variável preditora com relação a outras variáveis preditoras. No entanto, preditoras com alto VIF podem ter um alto p-value, então nós precisamos ver se a significância da variável preditora antes de removê-la do nosso modelo.

**Árvores de Decisão (*Decision Tree*) - Algoritmo CART**

Serão avaliados também com o algoritmo descrito por Breiman et. Al (1984). Árvores de decisão são métodos supervisionados não paramétricos usados para classificação e regressão. O algoritmo CART é abreviação de “Classification and Regression Trees*”* e foi introduzido por Breiman (1984). Com ele é possível construir tanto classificação quanto regressões nas árvores. A árvore de classificação pelo CART é baseada na divisão binária dos atributos. É também baseado no algoritmo de Hunt e pode ser implementado de forma serial. Utiliza o índice de GINI como medida de divisão e seleção do atributo de divisão.

Este algoritmo é único com relação a outros algoritmos de Hung já que também pode ser usado para análises de regressão com ajuda das árvores de regressão (S. Anupama et al, 2011). A análise de regressão é utilizada para prever variáveis dependentes dados um conjunto de variáveis preditoras para um período.

Utiliza muitas variáveis únicas como critério de divisão como índice de GINI, etc., e outra multivariável para determinar o melhor ponto de quebra e os dados são guardados a cada nó para determinar o melhor ponto de corte. A combinação linear de critérios de quebra é utilizada na regressão linear. SALFORD SYSTEMS implementou uma versão do CART, chamado CART usando o código original de Breimen (1984). O CART reforçou as características principais e tratou as deficiências, dando origem a um classificador de árvores com alta acurácia na predição. (Priyam, A., 2013)

Conforme também explanado por Yajing et al. (2018), neste algoritmo, os dados são divididos em duas amostras de forma que as subsequentes são mais homogêneas que as predecessoras. No algoritmo CART utiliza o índice de Gini para fazer as divisões. O índice de Gini no nó numa árvore CART é dado por:

(7)

Onde é a proporção de que pertence à classe (padrão ou não-padrão). Portanto, quando os clientes em um nó não possuem diferença no status padrão, o índice de Gini atinge seu máximo valor. Quando todos os clientes estão em um nó que pertence à mesma classe, o índice de Gini se reduz a 0.

Para o critério de Gini na função , temos que é o número de clientes em enviados para o lado direito do nó-filho, e é o número de clientes em enviados para o nó-filho esquerdo, e é o número de clientes no nó S, e é o índice de Gini da sub árvore esquerda, é o índice de Gini na sub árvore direita.

(8)

Esta fórm. 8 responde pelo decréscimo da impureza relativa à impureza do nó que está sendo dividido.

Seguindo o exemplo dado por Yajing et al. (2018), abaixo serão demonstrados de forma resumida todos os passos a serem executados na aplicação do algoritmo de Árvores de Decisão. Assim, em vez de implementar esses comandos em alguma linguagem específica, optou-se por ilustrar o algoritmo através de pseudocódigo:

**Pseudocódigo algoritmo Árvore de Decisão**:

**Encontrar a melhor quebra de uma característica**:

Identificar o melhor ponto de quebra t em um ponto no nó que

maximiza

na fórm. 8.

**Encontrar a melhor quebra para o nó**:

Selecionar a característica cuja melhor quebra fornece o decréscimo

mais significante na impureza para o nó

Utilizar a melhor quebra da característica como melhor quebra geral

para o nó.

**Checar regras de parada e recursividade**. Checar o nível máximo de

profundidade da árvore e o tamanho de cada nó-pai e cada nó-filho. Se

nenhuma regra de parada foi atingida, implementar o algoritmo de novo em

cada nó-filho (Yajing et al.,2018).

**“Random Forest”**

Random Forest é um método supervisionado de machine learning. E pode ser usado tanto para classificação como para regressão, sendo também um algoritmo flexível e de fácil utilização. Uma floresta é composta por árvores. Diz-se que quanto mais árvores ali estão, mais robusta é a floresta. Random Forest cria árvores de decisão em dados selecionados aleatoriamente, obtendo a predição para cada árvore e seleciona a melhor solução através de mecanismos de votação. É também um bom indicador de importância das características (Shivapuje, P., & Manwade, K. B. (2019).

Tendo como base as árvores de decisão, o algoritmo “Random Forest” também descrito por Breiman, L (2002), faz parte dos chamados métodos “ensamble”, que são combinações de diferentes modelos para um resultado unificado. Como nome diz, as Árvores Aleatórias serão criadas com base nos dados analisados para que sejam consolidados e seu resultado seja otimizado. Nas *“*Random Forest*”*, modelos de regressão representarão as médias dos resultados, e os modelos classificatórios trarão o resultado que mais se repete dentre as árvores aleatórias.

No intuito de explanar o funcionamento deste algoritmo que, na verdade é uma combinação de várias execuções do Decision Tree, (Devi e Sumanjani, 2015) demonstraram através de pseudocódigo o fluxo de tratamento das informações e testes a serem verificados para implementação deste modelo:

**Pseudocódigo para “Random Forest”:**

**Passo 1: Para gerar a floresta**

Para cada j no número de árvores

Para cada i no número de nós

Aleatoriamente selecionar n características para o conjunto de dados.

Para cada característica no conjunto de características aleatórias

Calcular o ganho de informação.

Fim para cada.

Criar um nó para a característica com o maior ganho de informação.

Fim para cada.

Fim para cada.

**Passo 2: Encontrar o rótulo da classe**

Para cada rótulo de classe

Fazer média das probabilidades obtidas por todas as árvores da Floresta.

Fim para cada.

Definir que o rótulo de classe com a maior média para a instância.

**Resultados e Discussão**

Foi observado que havia apenas 11 (onze) valores faltantes nas variáveis TotalCharges. E estas linhas foram eliminadas pare efeito do estudo.

Foram 4 (quatro) variáveis contínuas Tenure, MonthlyCharges, TotalCharges e SeniorCitizen, esta última foi convertida em variável categórica. A coluna “Churn” representa o número de clientes que deixaram o plano no último mês. Pudemos observar que 26% dos clientes deixaram a plataforma no último mês.

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figura 1. Percentual de “churn” de clientes no último mês

Na análise, pudemos verificar que “Gender” (gênero) é praticamente igual para masculino e feminino. O percentual é mais alto em pessoas mais velhas “SeniorCitizen”. Os consumidores com parceiros “Partner” e dependentes tem “Churn” menor comparado àqueles que não possuem parceiros ou dependentes.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Fig. a | Fig. b | Fig. c |
|  |  |  |
| Fig. d | Fig. e | Fig. f |

Figura 2. Percentual de “Churn” de clientes agrupados pelas variáveis “gender” (a), “SeniorCitizen” (b), “Partner” (c), “Dependents: (d), “PhoneService: (e) e “MultipleLines” (f)

O “Churn” foi muito maior em casos de Fibra Ótica (“FiberOptic”) e serviços de internet (“InternetServices”). Clientes que não possuíam serviços de segurança online (“OnlineSecurity”), backup online (“OnlineBackup”) e suporte técnico (“TechSupport”) deixaram a plataforma no último mês.

A maioria dos clientes com assinatura mensal deixou o plano no último mês comparado com aqueles que tem plano de um ou dois anos de fidelidade. Observou-se também que aqueles que fazem pagamento de forma eletrônica tem maior “churn” do que os que não utilizam esta opção.

|  |  |  |
| --- | --- | --- |
|  |  |  |
| Fig. a | Fig. b | Fig. c |
|  |  |  |
| Fig. d | Fig. e | Fig. f |

Figura 3. Percentual de “churn” de clientes agrupados pelas variáveis “InternetService”, “Online Security”, “OnlineBackup”, “DeviceProtection”, “TechSupport” e “StreamingTV”. Fonte: Resultados originais da pesquisa

Observamos a prevalência para não haver “churn” em clientes que não possuíam suporte (“TechSupport”) e leve aumento em casos de clientes que não possuíam proteção de dispositivos (“DeviceProtection”).

|  |  |
| --- | --- |
|  |  |
| Fig. a | Fig. b |
|  |  |
| Fig. c | Fig. d |

Figura 4. Percentual de “Churn” de clientes agrupados pelas variáveis StreamingMovies, “Contract”, “PaperlessBilling” e “PaymentMethod”

Fonte: Resultados originais da pesquisa

Convém notar que é significativamente maior o “churn” clientes com pagamento eletrônicos (Fig. d), em contratos mensais (Fig. b), especialmente os que pagam boletos eletrônicos (Fig. c).

Gráfico, Gráfico de caixa estreita

Descrição gerada automaticamente

Figura 5. Box plot de Tenure: A mediana do tempo que os clientes saíram do plano (“Tenure”) foi de 10 meses. Fonte: Resultados originais da pesquisa.

Observou-se que o tempo de permanência no plano contratado de telecomunicações teve interferência no evento analisado (Fig. 5), uma vez que os que cancelam os planos permanecem em torno de 10 (dez) meses com o contrato ativo.

|  |  |
| --- | --- |
| Gráfico, Gráfico de caixa estreita  Descrição gerada automaticamente  Figura 6. Box plot de MonthlyCharges: Clientes que saíram do plano tiveram maior média mensal, acima de 75 dólares  Fonte: Resultados originais da pesquisa | Gráfico, Gráfico de caixa estreita  Descrição gerada automaticamente  Figura 7. Box plot de TotalCharges: A média de total pago (TotalCharges) dos clientes que saíram do plano é mais baixa do que os que ficaram no plano  Fonte: Resultados originais da pesquisa |

Conforme verificado abaixo na Figura 8, o total pago (TotalCharges) tem correlação positiva com pagamentos mensais (MonthlyCharges) e com o tempo de permanência do cliente (“Tenure”).

Gráfico

Descrição gerada automaticamente

Figura 8. Gráfico da correlação entre as variáveis quantitativas

Fonte: Resultados originais da pesquisa

No gráfico abaixo foram avaliadas as variáveis quantitativas para identificação de possíveis “outliers”. No caso, não foram identificados “outliers”, ficando os quartis distribuídos conforme os gráficos de “box plot” abaixo:

|  |  |  |
| --- | --- | --- |
| Gráfico, Gráfico de caixa estreita  Descrição gerada automaticamenteTenure  Fig. a | Gráfico, Gráfico de caixa estreita  Descrição gerada automaticamente  MonthlyCharges  Fig. b | Gráfico, Gráfico de caixa estreita  Descrição gerada automaticamenteTotalCharges  Fig. c |

Figura 9. Verificação de possíveis outliers nas variáveis quantitativas

Fonte: Resultados originais da pesquisa

Para transformar a variável “Tenure” em categórica, foram criadas faixas de tempo: 0-1 ano, 1-2 anos, 2-3 anos, 3-4 anos, 4-5 anos e 5-6 anos. Após o procedimento podemos temos uma visão do tempo de permanência dos clientes através destas faixas:

Gráfico, Gráfico de barras

Descrição gerada automaticamente

Figura 10. Gráfico da distribuição do churn de clientes conforme os tempos de permanência no plano (“Tenure”)

Fonte: Resultados originais da pesquisa

A base foi dividida entre treino e teste. O treino ficou com 80% das observações, sendo reservados os 20% para verificação posterior e avaliação de “overfitting”. A seguir, foi feita a padronização das variáveis contínuas através do processo “zscore”, que consiste em subtrair a média ( ) do valor e dividir pelo desvio padrão (), ou seja:

(9)

O próximo passo foi utilizar transformar as variáveis categóricas em “dummy” para preparar para a aplicação a regressão logística. No caso dos demais algoritmos este procedimento não é necessário, uma vez que possuem outros mecanismos.

Foi executado o modelo GLM em função de todas as variáveis para obter um primeiro resultado. No primeiro modelo, utilizando o comando glm (formula = Churn ~., family = “binomial”, data = telco), já verificamos que algumas variáveis não são significativas, como: “gender”, “SeniorCitizen”, “Partner”, “Dependents”, “PhoneService”, “TotalCharges”, “PhoneService”, “MultipleLines” cujo p-value não atinge os 5% de significância.

O modelo executado em função de todas as variáveis, obtendo AIC 4651,89 e um LL (Loglik) de -2880.817 (df=29). Assim, para avaliar o efeito, aplicamos o procedimento “stepwise” que eliminou variáveis e devolveu o modelo otimizado, mais parcimonioso. Após o “stepwise”, observemos que neste caso o LL (loglik) melhorou para -2072.641 (df=17). Temos abaixo a tabela comparativa entre os modelos antes e após o procedimento “stepwise”.

Tabela 2. Coeficientes e erro-padrão comparativo após procedimento de stepwise.

|  |  |  |
| --- | --- | --- |
|  | Sem “StepWise” | Após “StepWise” (continua) |
| (Intercept) | -3.96 \*\* | -5.34 \*\*\* |
|  | (1.50) | (0.51) |
| tenure | -2.44 \*\*\* | -2.00 \*\*\* |
|  | (0.30) | (0.17) |
| MonthlyCharges | -1.08 | -2.08 \*\*\* |
|  | (1.09) | (0.38) |
| Tabela 2. Coeficientes e erro-padrão comparativo após procedimento de stepwise. | | |
|  | Sem “StepWise” | Após “StepWise” (continua) |
| TotalCharges | 0.33 |  |
|  | (0.19) |  |
| gender | -0.01 |  |
|  | (0.07) |  |
| SeniorCitizen | 0.17 | 0.20 \* |
|  | (0.10) | (0.09) |
| Partner | 0.07 |  |
|  | (0.09) |  |
| Dependents | -0.15 |  |
|  | (0.10) |  |
| PhoneService | 0.06 | 0.79 \*\* |
|  | (0.74) | (0.30) |
| MultipleLines | 0.57 \*\* | 0.77 \*\*\* |
|  | (0.20) | (0.11) |
| InternetService.xFiber.optic | 1.75 | 2.68 \*\*\* |
|  | (0.91) | (0.32) |
| InternetService.xNo | -1.57 | -2.41 \*\*\* |
|  | (0.92) | (0.38) |
| OnlineSecurity | -0.28 |  |
|  | (0.20) |  |
| OnlineBackup | 0.08 | 0.27 \* |
|  | (0.20) | (0.11) |
| DeviceProtection | 0.14 | 0.34 \*\* |
|  | (0.20) | (0.11) |
| TechSupport | -0.15 |  |
|  | (0.21) |  |
| StreamingTV | 0.56 | 0.94 \*\*\* |
|  | (0.37) | (0.16) |
| StreamingMovies | 0.66 | 1.04 \*\*\* |
|  | (0.37) | (0.16) |
| Contract.xOne.year | -0.71 \*\*\* | -0.73 \*\*\* |
|  | (0.12) | (0.12) |
| Contract.xTwo.year | -1.35 \*\*\* | -1.41 \*\*\* |
|  | (0.20) | (0.20) |
| PaperlessBilling | 0.39 \*\*\* | 0.39 \*\*\* |
|  | (0.08) | (0.08) |
| PaymentMethod.xCredit. card.automatic. | -0.14 |  |
|  | (0.13) |  |
| PaymentMethod.xElectro nic.check | 0.26 \* | 0.36 \*\*\* |
|  | (0.11) | (0.08) |
| Tabela 2. Coeficientes e erro-padrão comparativo após procedimento de stepwise. | | |
|  | Sem “StepWise” | Após “StepWise” (conclusão) |
| PaymentMethod.xMailed. check | -0.15 |  |
|  | (0.13) |  |
| tenure\_bin.x1.2.anos | 0.20 |  |
|  | (0.18) |  |
| tenure\_bin.x2.3.anos | 0.91 \*\* | 0.68 \*\*\* |
|  | (0.29) | (0.18) |
| tenure\_bin.x3.4.anos | 1.98 \*\*\* | 1.68 \*\*\* |
|  | (0.42) | (0.25) |
| tenure\_bin.x4.5.anos | 2.59 \*\*\* | 2.25 \*\*\* |
|  | (0.54) | (0.32) |
| tenure\_bin.x5.6.anos | 3.29 \*\*\* | 2.95 \*\*\* |
|  | (0.67) | (0.40) |
|  |  |  |
| N | 5625 | 5625 |
| AIC | 4651,89 | 4644,38 |
| BIC | 4844,30 | 4777,08 |
| Pseudo R2 | 0.42 | 0.42 |

Fonte: Resultados originais da pesquisa.

É importante observar que houve reflexo da remoção de variáveis menos significativas também no AIC e BIC. A seguir, precisamos avaliar se existe colinearidade de variáveis para prosseguimento na análise.

Tabela 3. Valores VIF para as variáveis após “stepwise”.

|  |  |
| --- | --- |
| variável | “Variance Inflation Factor” [VIF] (continua) |
| tenure | 2,355329 |
| MonthlyCharges | 1,456172 |
| Dependents | 1,030633 |
| PhoneService | 1,484161 |
| MultipleLines | 1,362878 |
| InternetService.xFiber.optic | 2,066139 |
| InternetService.xNo | 1,833716 |
| OnlineSecurity | 1,092687 |
| OnlineBackup | 1,15825 |
| DeviceProtection | 1,256749 |
| TechSupport | 1,163156 |
| StreamingTV | 1,48944 |
| StreamingMovies | 1,480605 |
| Contract.xOne.year | 1,207092 |
| Tabela 3. Valores VIF para as variáveis após “stepwise”. | |
|  | “Variance Inflation Factor” [VIF] (conclusão) |
| Contract.xTwo.year | 1,26164 |
| PaperlessBilling | 1,122183 |
| PaymentMethod.xCredit.card..automatic. | 1,241474 |
| PaymentMethod.xElectronic.check | 1,363924 |
| tenure\_bin.x1.2.anos | 1,055156 |
| tenure\_bin.x5.6.anos | 2,036837 |

Fonte: Resultados originais da pesquisa.

Na tabela 3, acima, observemos que diversas variáveis não significativas foram removidas pelo “stepwise”, como “DeviceBackup” e “DeviceProtection”, o que nos resultou em novo modelo para foram feitas as análises de matriz de confusão, eficácia e área da curva ROC (AUC – “Area Under Curve”).

Tabela 4. Análise de sensibilidade, especificidade e acurácia do modelo glm na base de teste.

|  |  |  |  |
| --- | --- | --- | --- |
|  | 50% | 70% | 30% |
| Sensitividade | 0,6321 | 0,6933 | 0,5265 |
| Especificidade | 0,8411 | 0,7582 | 0,9000 |
| Acurácia | 0,7939 | 0,7548 | 0,7548 |

Fonte: Resultados originais da pesquisa.

Sabendo que o modelo GLM devolveu os percentuais de probabilidade de ocorrer o evento Churn devemos definir agora qual percentual irá significar Churn ou não-Churn. Agora vamos avaliar usando dados de validação, supondo alguns “cutoffs”.

Percebemos que um cutoff de 50% temos boa acurácia e especificidade, porém baixa sensitividade. No gráfico abaixo, podemos avaliar visualmente a evolução dos indicadores em função da variação do cutoff e verificamos que em torno de 30% seria o cruzamento das três linhas (sensitividade, especificidade e acurácia). No nosso caso, vamos focar na acurácia por representar indicador geral do modelo:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figura 11. Gráfico de sensitividade, especificidade e acurácia em função do “cutoff”

Fonte: Resultados originais da pesquisa

Devemos analisar a área debaixo da curva ROC e coeficiente de GINI, pois necessitamos de mais indicadores de qualidade para o nosso modelo. No modelo abaixo, foi mantida a variável resposta como devolvida pelo GLM, com percentuais de probabilidade, e não 0 ou 1, e não como variável binária:

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figura 12 – Curva ROC do modelo glm na base de treino

Fonte: Resultados originais da pesquisa

Para que seja comprovada a acurácia de um modelo, devemos agora aplicar os mesmos parâmetros na base de testes, pois tal ação permitiu assegurar a taxa de predição em dados desconhecidos. Observemos que a área debaixo da curva cai de 0,838 para 0,758, o que é natural, em se tratando de base de teste.

Gráfico, Gráfico de linhas

Descrição gerada automaticamente

Figura 13 – Curva ROC do modelo GLM na base de teste

Fonte: Resultados originais da pesquisa

Gráfico, Gráfico de pizza

Descrição gerada automaticamente

Figura 14. Matriz de Confusão modelo GLM cutoff de 0,32

Além da matriz de confusão, foi observado uma queda na área sob a Curva ROC na base de teste. A curva das características operacionais do receptor “Receiver Operating Characteristic” [ROC] mostra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos para diferentes limites.

Tendo como base o modelo de Generalized Linear Model (GLM), que analisamos até agora, iremos comparar a capacidade classificatória com dois outros modelos de “machine learning”. A seguir, temos a matriz de confusão do algoritmo “Decision Tree”, seguido pela matriz de confusão do algoritmo “Random Forest”, ambos também treinados com a mesma amostra utilizada no GLM.

Temos abaixo a matriz de confusão do algoritmo “Decision Tree”. Observa-se que nele houve grande ganho na predição dos positivos (921 x 785), porém perda na comparação de predição dos negativos com o GLM (283 x 181).

Gráfico, Gráfico de pizza

Descrição gerada automaticamente

Figura 15. Matriz de Confusão modelo “Decision Tree”

Na aplicação do modelo “Random Forest”, foi parametrizado para que fossem utilizadas 2000 (duas mil) árvores randômicas para que houvesse o resultado médio. Isso demandou processamento maior do que os modelos anteriores. Observa-se que é o modelo que mais acerta os positivos (947), porém é o que menos acerta os negativos (168).

Gráfico, Gráfico de pizza

Descrição gerada automaticamente

Figura 15. Matriz de Confusão modelo Random Forest

Tabela 6. Comparativo de indicadores dos modelos (base teste).

|  |  |  |  |
| --- | --- | --- | --- |
| medida | glm | Decision Tree | Random forest |
| Acurácia | 0,7591 | 0,7832 | 0,7932 |
| Curva ROC (AUC) | 76% | 67% | 70% |
| GINI | 0,5170 | 0,3390 | 0,3640 |
| Sensitividade | 0,5330 | 0,6177 | 0,6602 |
| Especificidade | 0,8961 | 0,8268 | 0,8232 |
| Precisão | 0,7567 | 0,4840 | 0,4572 |
| Recall | 0,5330 | 0,6177 | 0,6602 |

Fonte: Resultados originais da pesquisa.

**Gráfico, Gráfico de linhas

Descrição gerada automaticamente**

Figura 17. Gráfico comparativo de área sob curva ROC

Fonte: Resultados originais da pesquisa

Com base no gráfico Figura 17, observamos que a área debaixo da curva ROC foi maior no modelo GLM, em comparação com modelos derivados da árvore de decisão. Cabe observar que o “cutoff” adotado no modelo GLM foi de 32%, ou seja, clientes cuja probabilidade de “churn” fosse maior ou igual a 32% foram classificados como positivos, e abaixo disso foi considerado negativo.

Quanto à acurácia, se isolarmos como fator de ranking, o modelo “Random Forest” é o melhor, uma vez que possui acurácia de 79,32% comparado com 75,91% do modelo GLM e 78,32% do modelo Decision Tree. Se entendermos apenas acurácia como fator de decisão, o glm não performa tão bem quanto estes últimos dois algoritmos. Porém, sabe-se que estes dois modelos não possuem parâmetro de “cutoff”, perdendo assim a capacidade de convergir no modelo final as curvas de sensitividade e especificidade. Caso adotássemos cutoff de 50% (em vez de 32%), a 79,67%, bem próximo ao “Random Forest”.

Quanto às medidas de impureza, como coeficiente de GINI, o “Random Forest” obteve 0,4572 como menor e melhor índice para os modelos avaliados, sendo que o modelo GLM obteve índice de GINI mais elevado, de 0,5170.

Cabe observar nos indicadores da matriz de confusão o grande nível de precisão atingido pelo modelo GLM, que foi de 75,67%, comparado a 48,40% e 45,72% de Decision Tree e Random Forest.

**Conclusões**

Com a análise dos dados através das modelagens realizadas, conclui-se que é possível construir modelos matemáticos e utilizar algoritmos de “machine learning” para previsão de “churn” de clientes. Como demonstrado, os modelos mantiveram acurácia consistente entre 75% e 79% com a divisão das amostras entre 80% para treino e 20% para teste. Um ponto importante observado é que o modelo menos demandante computacionalmente, o GLM, performou satisfatoriamente diante dos modelos baseados em árvore de decisão, que exigem maior poder computacional.

Convém notar que a análise realizada ficou restrita às variáveis fornecidas apenas por uma base de dados, da empresa Telco, o que limita o modelo à aplicação geral. No entanto, é válido observar que o fenômeno estudado responde bem a inferências matemáticas e deve ter seu estudo ampliado para ajudar mais empresas a compreenderem seus clientes e atuarem na sua retenção. É oportuno recomendar a ampliação destas análises com a utilização de algoritmos baseados em Redes Neurais, uma vez que avançam na captação de modelos não-lineares.

**Agradecimento**

Agradeço à minha família, aos meus professores do MBA USP-Esalq e ao meu orientador Mateus pela paciência e apoio durante o desenvolvimento deste trabalho.

**Referências**

1. Tjaden, B.; Cohen, J. A survey of computational methods used in microarray data interpretation. In: Applied Mycology and Biotechnology. [S.l.]: Elsevier, 2006. v. 6, p. 161–178.
2. Chen, Y.; Wong, M. L. An ant colony optimization approach for stacking ensemble. In: IEEE. Nature and Biologically Inspired Computing (NaBIC), 2010 Second World Congress on. [S.l.], 2010. p. 146–151.
3. Gil, A. C. (2002) Como elaborar projetos de pesquisa. 4a. ed. São Paulo: Atlas S/A.
4. Fávero, L. P., & Belfiore, P. (2017). Manual de Análise de Dados Estatística Modelagem Multivariada Com Excel, SPSS E Stata. In Elsevier.
5. Bryk A. S, & Raudenbush, S. W. (1992). Hierarchical linear models: applications and data analysis methods. 2. ed. Thousand Oaks: Sage Publications.
6. Tabachnick, B. G. & Fidell, L. S. (2013). Using multivariate statistics. 6. ed. Boston: Pearson.
7. Shapiro, S. S. & Francia, R. S. (1972). Approximate analysis of variance test for normality. Am. Statist. Assoc. 67, 215-25.
8. Breusch, T., & Pagan, A. (1980). The Lagrange multiplier test and its applications to model specification in econometrics, The Review of Economic Studies, 47(1):239-254
9. Dio, R. A. T. D. (1979). Prefácio à edição brasileira. In: CAMPBELL, D. T.; STANLEY, J. C. Delineamentos experimentais e quase-experimentais de pesquisa. São Paulo: EPU.
10. Bounsaythip, C. & Rinta-Runsala, E. (2001) Overview of Data Mining for Customer Behavior Modeling. Louhi. Research Report TTE1-2001-18.
11. PriceWaterHouseCoopers. (1999) *The CRM Handbook: from Group to multiindividual*.
12. Berson, A. & Thearling, K. (2001), "Building Data mining Applications for CRM". McGraw-Hill, 2000.
13. McGowan, Donal et al. (2011) Churn prediction in mobile telecommunications. In: Proceedings of the 22nd Irish Conference on Artificial Intelligence and Cognitive Science.
14. E. Stripling, S. vanden Broucke, K. Antonio, B. Baesens and M. Snoeck, 2015. "Profit maximizing logistic regression modeling for customer churn prediction," 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), pp. 1-10, doi: 10.1109/DSAA.2015.7344874.
15. A. Ahmed and D. M. Linen, 2017, "A review and analysis of churn prediction methods for customer retention in telecom industries," 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pp. 1-7, doi: 10.1109/ICACCS.2017.8014605.
16. Seo, DongBack; Ranganathan, C.; Babad, Yair. Two-level model of customer retention in the US mobile telecommunications service market. Telecommunications policy, v. 32, n. 3-4, p. 182-196, 2008.
17. Dash, Manoj; Singh, Kunwar Ravendra. Churn Prediction in Telecom Industry Using R.
18. Woolf, Barnet. The log likelihood ratio test (the G‐test). Annals of human genetics, v. 21, n. 4, p. 397-409, 1957.
19. Penny, William D. Comparing dynamic causal models using AIC, BIC and free energy. Neuroimage, v. 59, n. 1, p. 319-330, 2012.
20. Au, W; Chan, C. C; Yao, X. A novel evolutionary data mining algorithm with applications to churn prediction. IEEE Transactions on Evolutionary Computation, v. 7, n. 6, p. 532-545, 2003.
21. Kurtz, D. L; Clow, K. E. Services marketing New York: John Wiley & Sons, 1998.
22. Fawcett, Tom. ROC graphs with instance-varying costs. Pattern Recognition Letters, v. 27, n. 8, p. 882-891, 2006.
23. I. Ullah, B. Raza, A. K. Malik, M. Imran, S. U. Islam and S. W. Kim, "A Churn Prediction Model Using Random Forest: Analysis of Machine Learning Techniques for Churn Prediction and Factor Identification in Telecom Sector," in IEEE Access, vol. 7, pp. 60134-60149, 2019, doi: 10.1109/ACCESS.2019.2914999.
24. Lu, Ning et al. A customer churn prediction model in telecom industry using boosting. IEEE Transactions on Industrial Informatics, v. 10, n. 2, p. 1659-1665, 2012.
25. Santos, C. A. S. T. et al. Modelagem multinível. Sitientibus, Feira de Santana, v. 22, p. 89-98, 2000.
26. Montgomery, D. C.; Peck, E. A.; Vining, G. G. Introduction to linear regression analysis. John, Wiley and Sons, Inc., New York, 612p, 2006.
27. Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). Classification and regression trees. Routledge.
28. Breiman, L. (2001), Random Forests, Machine Learning 45(1), 5-32.
29. Breiman, L (2002), “Manual on Setting Up, Using, And Understanding Random Forests V3.1
30. Breiman L., Friedman J. H., Olshen R. A., and Stone, C. J. (1984) *Classification and Regression Trees.* Wadsworth.
31. S. Anupama Kumar and Dr. Vijayalakshmi M.N. (2011) “Efficiency of decision trees in predicting student’s academic performance”, D.C. Wyld, et al. (Eds): CCSEA 2011, CS & IT 02, pp. 335-343, 2011.
32. Priyam, A., Abhijeeta, G. R., Rathee, A., & Srivastava, S. (2013). Comparative analysis of decision tree classification algorithms. *International Journal of current engineering and technology*, *3*(2), 334-337.
33. Yajing, Zhang & Chi, Guotai & Zhang, Zhipeng. (2018). Decision tree for credit scoring and discovery of significant features. Filomat. 32. 10.2298/FIL1805513Z.
34. R. G. Devi and P. Sumanjani, "Improved classification techniques by combining KNN and Random Forest with Naive Bayesian classifier," 2015 IEEE International Conference on Engineering and Technology (ICETECH), 2015, pp. 1-4, doi: 10.1109/ICETECH.2015.7274997.
35. Shivapuje, P., & Manwade, K. B. (2019). Detecting Users Stress by using Social Network. International Journal of Scientific Research in Computer Science, Engineering and Information Technology, 493–496. https://doi.org/10.32628/CSEIT183821

Anexos

Anexo 1. Repositório dos códigos fontes utilizados neste trabalho:

https://github.com/emebatista/telco\_churn\_tcc