**Título do Projeto**

#### **Otávio Fiori Maginador**

Universidade Federal do ABC (UFABC)

otavio.fiori@aluno.ufabc.edu.br

**Abstract –** This article contains a Data Analysis about the public dataset BankChurners. It will compare different classification models and data preparation trying to classify if a client will leave his bank. After comparing the models, it will be analysed the best decision threshold to classify the client.

**Keywords –** Bank Churners. Unbalanced. Classification. AUC Score. F1-Score.

# 1. Introdução

Um dos trabalhos mais desafiadores de qualquer empresa e mais especificamente times de CRM (*Customer Relationship Management*) é conseguir identificar bem antes quais são aqueles clientes que tem a maior probabilidade de deixar sua empresa, porém, um grande problema que muitas vezes é encontrado é que normalmente há um desequilíbrio de classes, ou seja, há muito mais clientes que continuaram na empresa do que clientes que a deixaram, que aqui chamaremos de *churn*.

Este artigo explora as variáveis deste *dataset* e mostra algumas técnicas que podem ser utilizadas para conseguir contornar este problema e fazer boas previsões.

# 2. Análise Exploratória

Nesta seção, é apresentada a análise exploratória dos dados.

## 2.1 Variável Resposta

A variável resposta de um modelo é aquela à qual desejamos prever os resultados e neste *dataset* será utilizada a variável ‘*Attrition\_Flag*’ que pode assumir os valores ‘*Existing Customer*’ ou ‘*Attrited Customer*’, sendo a segunda a identificação do *churn*. Como pode ser visto em laranja na figura abaixo apenas 16,1% dos clientes deram *churn* no *dataset*.

A blue and orange pie chart

Description automatically generated with medium confidence

**Figura 1. Percentual de Churn.**

Quando a variável resposta não está bem distribuída, ou seja, com uma classe predominante, pode-se dizer então que este é um *dataset* desbalanceado, o que requer uma análise mais profunda dos resultados do modelo para que possa ser avaliado.

## 2.2 Variáveis Contínuas

As variáveis contínuas são aquelas com características mensuráveis que assumem valores em uma escala contínua (na reta real), para as quais valores fracionais fazem sentido, que neste *dataset* são:

* *Customer\_Age*: idade do cliente;
* *Dependent\_count*: número de dependentes do cliente;
* *Months\_on\_book*: tempo de relacionamento com o banco;
* *Total\_Relationship\_Count*: quantidade de produtos contratados no banco;
* *Months\_Inactive\_12\_mon*: meses inativos nos últimos 12 meses;
* *Contacts\_Count\_12\_mon*: quantidade de contatos com o banco nos últimos 12 meses;
* *Credit\_Limit*: limite de crédito no cartão de crédito;
* *Total\_Revolving\_Bal*: saldo rotativo no cartão de crédito;
* *Avg\_Open\_To\_Buy*: média da linha de crédito para compra dos últimos 12 meses;
* *Total\_Amt\_Chng\_Q4\_Q1*: alteração no valor de transação do quarto trimestre dividido pelo primeiro trimestre;
* *Total\_Trans\_Amt*: valor total transacionado nos últimos 12 meses;
* *Total\_Trans\_Ct*: quantidade total de transações nos últimos 12 meses;
* *Total\_Ct\_Chng\_Q4\_Q1*: alteração na contagem de transações do quarto trimestre sobre o primeiro trimestre
* *Avg\_Utilization\_Ratio*: taxa média de utilização do cartão

Para estas variáveis foi feita uma análise de densidade, como pode ser visto na Figura 2 abaixo:

A picture containing text, line, font, plot

Description automatically generated

**Figura 2. Análise de Densidade.**

Como pode ser visto na Figura 2 acima algumas das variáveis não seguem uma distribuição normal e por isso devem existir alguns cuidados na hora de testar uma Regressão Logística, por exemplo. Por ser uma técnica que tem como premissa que as variáveis explicativas seguem uma distribuição normal, o correto seria padronizar os dados antes do treinamento para que a média seja igual a 0 e o desvio padrão igual a 1.

Na Figura 3 abaixo também é possível identificar diversos *outliers* em algumas das variáveis.

A picture containing line, diagram, parallel, plot

Description automatically generated

**Figura 3. Outliers.**

Em estatística, outlier, valor aberrante ou valor atípico, é uma observação que apresenta um grande afastamento das demais da série, ou que é inconsistente. Quando os dados têm muitos *outliers* algoritmos que fazem cálculos de distância, como o KNN (*K-Nearest Neighbors*), podem ter seus cálculos de centroides distorcidos e utilizar mais tempo de processamento.

Outra análise importante sobre as variáveis categóricas é a análise de correlação entre os dados. O Coeficiente de Correlação de Pearson mede o grau da correlação entre duas variáveis de escala métrica. Este coeficiente, normalmente representado por ρ assume apenas valores entre 1 e -1, sendo 1 perfeitamente e diretamente correlatas e -1 perfeitamente e inversamente correlatas. Na Figura 4 há um mapa de calor para identificar as maiores correlações.

A picture containing text, diagram, colorfulness, plan

Description automatically generated

**Figura 4. Mapa de Calor de Correlação.**

Com ela é possível perceber que algumas variáveis têm correlações fortes (maior que 0.5), sendo que as preditoras *Credit\_Limit* e *Avg\_Open\_To\_Buy* tem uma correlação perfeita e igual a 1.

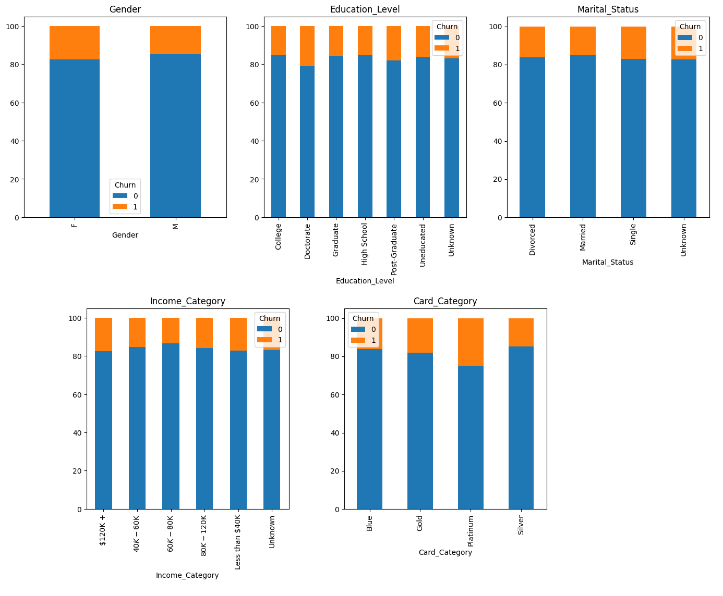
O grande problema da correlação acontece principalmente nos algoritmos de Regressão Logística e *Naive Bayes*, pois ambos assumem que as variáveis preditoras são independentes uma das outras. Em demais algoritmos ter variáveis muito correlatas podem não alterar a performance do modelo, porém podem acarretar uma demora maior na etapa de treinamento.

## 2.3 Variáveis Categóricas

Uma variável categórica é uma variável que pode assumir apenas um número limitado, e geralmente fixo, de valores possíveis, atribuindo cada indivíduo ou outra unidade de observação a um determinado grupo ou categoria nominal com base em alguma propriedade qualitativa. Neste *dataset* temos 5 variáveis deste tipo:

* *Gender*: o gênero do cliente;
* *Education\_Level*: nível educacional;
* *Marital\_Status*: estado civil;
* *Income\_Category*: categoria do salário;
* *Card\_Category*: categoria do cartão.

Para este tipo de variável é importante analisar a proporção da classe resposta dentre cada uma das categorias, para poder assim entender como ela varia.



**Figura 5. Percentual de *Churn* por Categoria.**

Pela Figura 5 é possível perceber que Categoria do Cartão de Crédito, a Categoria de Renda e o Nível Educacional são as variáveis que mais distorcem a distribuição de *churn,* o que indica uma possível relação das categorias com a resposta.

# 3. Preparação dos Dados

Nesta seção, serão apresentadas e discutidas algumas técnicas de preparação dos dados e os primeiros testes de modelos.

## 3.1 Teste de Modelos sem DataPrep

Como pôde ser observado na seção anterior os dados precisam de uma preparação antes que sejam utilizados em modelos, principalmente aqueles que fazem premissas, porém foi testado o desempenho de cada algoritmo sem a preparação para poder compará-los. Os modelos utilizados foram:

* Regressão Logística;
* Naive Bayes;
* KNN (*k-nearest neighbors)*;
* Árvore de Decisão;
* *Random Forest*;
* *SVM* (*Support Vector Machines*);
* *Gradient Boosting*;

Em todos os testes foi utilizada a técnica de validação cruzada com 5 *folds*.

A picture containing text, diagram, screenshot, plan

Description automatically generated

**Figura 6. Performance dos modelos sem *DataPrep*.**

Na Figura 6 é possível notar que os algoritmos *Gradient Boosting*, *Random Forest*, Naive Bayes e Regressão Logísitica tiveram melhor performance e foram mais estáveis tendo um desvio padrão mais baixo.

## 3.2 Técnicas Utilizadas

Para resolver o problema de correlação visto na Figura 4, foi aplicado um filtro de correlação em que após medir todas as correlações entre as variáveis marca aqueles pares onde a correlação é maior que 0,7. Feito isso, é mantido no *dataset* apenas aquela variável com maior correlação com a classe resposta.

Em relação às variáveis que apresentam uma distribuição diferente da normal foi aplicada uma padronização utilizando a função *StandardScalar* do pacote de *machine learning* scikit-learn.

## 3.3 Teste de Modelos com DataPrep

Feita a devida preparação dos dados os mesmos algoritmos anteriores foram treinados novamente com a mesma metodologia.

A picture containing text, diagram, screenshot, plan

Description automatically generated

**Figura 7. Performance dos modelos com *DataPrep*.**

|  |  |  |  |
| --- | --- | --- | --- |
| Modelo | Teste 1 | Teste 2 | Diferença Absoluta |
| NB | 0,9113 | 0,8944 | 0,0169 |
| LR | 0,9254 | 0,9310 | 0,0056 |
| CART | 0,7832 | 0,7797 | 0,0036 |
| KNN | 0,7894 | 0,7907 | 0,0013 |
| SVM | 0,7889 | 0,7877 | 0,0013 |
| RF | 0,9316 | 0,9311 | 0,0005 |
| GB | 0,9595 | 0,9595 | 0,0000 |

**Tabela 1. Comparação de Testes**

Neste segundo teste é notável que os algoritmos que tiveram boa performance no teste anterior continuaram tendo boa performance. As diferenças de fato estão na Regressão Logística e no Naive Bayes, no caso da regressão mesmo utilizando menos variáveis preditoras vemos que teve uma pequena melhora pois no segundo *dataset* criado as variáveis seguem as premissas do algoritmo, já no caso do Naive Bayes mesmo seguindo as premissas necessárias acabou perdendo um pouco de performance.

Outro ponto interessante é que os demais modelos mantiveram seus resultados mesmo descartando 4 variáveis, mostrando que remover variáveis não necessariamente diminui a performance do modelo.

# 4. Modelagem

Nesta seção, serão feitos os modelos finais do projeto utilizando os dados com pré-processamento aplicado.

## 4.1 Modelo Base

Como modelo base vamos utilizar a Regressão Logística e o *Gradient Boosting* por terem alta performance nos testes anteriores para compará-los.

O primeiro modelo treinado foi a Regressão Logística e nas figuras abaixo podemos verificar os resultados.

A picture containing text, screenshot, font, number

Description automatically generated

**Figura 8. Matriz de Confusão Regressão**

A picture containing text, line, plot, screenshot

Description automatically generated

**Figura 9. ROC AUC Score**

Pode-se perceber que a matriz de confusão ficou com uma acurácia baixa neste modelo, porém isso se deve ao fato de haver uma resposta com dados desbalanceados. Para casos como este a curva ROC é a melhor métrica para avaliação da performance.

ROC é uma curva de probabilidade e AUC representa o grau ou medida de separabilidade e diz o quanto o modelo é capaz de distinguir entre as classes. Quanto maior a AUC, melhor será o modelo. Este teste atingiu 0,92 indicando um bom classificador.

O segundo modelo foi o *Gradient Boosting* e podemos verificar os resultados abaixo.

A picture containing text, screenshot, font, number

Description automatically generated

**Figura 10. Matriz de Confusão *Gradient Boosting***

**A picture containing text, screenshot, line, plot

Description automatically generated**

**Figura 11. ROC AUC Score *Gradient Boosting***

Aqui é possível notar que o algoritmo conseguiu resolver relativamente bem o desbalanceamento da base acertando 89% dos clientes com *churn*. A ROC AUC Score do mesmo foi de 0,988, indicando um classificador melhor.

## 4.2 Ajustando Hiperparâmetros

Com os modelos base desenvolvidos é possível agora ajustar alguns hiperparâmetros para aumentar a performance dos modelos e evitar casos de *overfitting*. Neste teste foram alterados os principais hiperparâmetros de cada algoritmo. Para a Regressão Logística:

1. *Penalty*: pode-se aplicar a penalização de variáveis para melhor ajustar os coeficientes do modelo. Melhor encontrado: l2;
2. C: inverso da força de penalização. Melhor encontrado: 0,5.

A picture containing text, screenshot, font, number

Description automatically generated

**Figura 12. Matriz de Confusão Regressão com melhores Hiperparâmetros**

**A picture containing text, line, plot, screenshot

Description automatically generated**

**Figura 13. ROC AUC Score Regressão com melhores Hiperparâmetros**

Para o Gradient Boosting:

1. *learning*\_*rate*: taxa de aprendizado do modelo. Melhor encontrado: 0,1.
2. *n\_estimators*: número de árvores de regressão utilizadas pelo modelo. Melhor encontrado: 300.
3. *min\_samples\_leaf*: quantidade mínima de dados em uma folha. Melhor encontrado: 0,05.
4. *max\_depth*: profundidade máxima das árvores. Melhor encontrado: 5.

A picture containing text, screenshot, font, number

Description automatically generated

**Figura 14. Matriz de Confusão *Gradient Boosting* com melhores Hiperparâmetros**

A picture containing text, screenshot, line, plot

Description automatically generated

**Figura 15. ROC AUC Score *Gradient Boosting* com melhores Hiperparâmetros**

Em ambos os algoritmos podemos verificar que houve avanços significativos tanto no ROC AUC Score quanto nas matrizes de confusão.

# 5. *Threshold*

Nesta seção, serão apresentadas 2 técnicas para decidir o melhor *threshold* de probabilidade.

## 5.1 Histograma de Probabilidade

Até o momento foram apresentados a ROC AUC Score e a matriz de confusão, porém ao se fazer uma estimativa com um dado todos os algoritmos apresentados (com exceção das SVMs) também calculam a probabilidade de determinado registro pertencer à uma classe. Com isso pode-se então gerar um gráfico do histograma dessas probabilidades para identificar se em algum momento as classes se misturam.

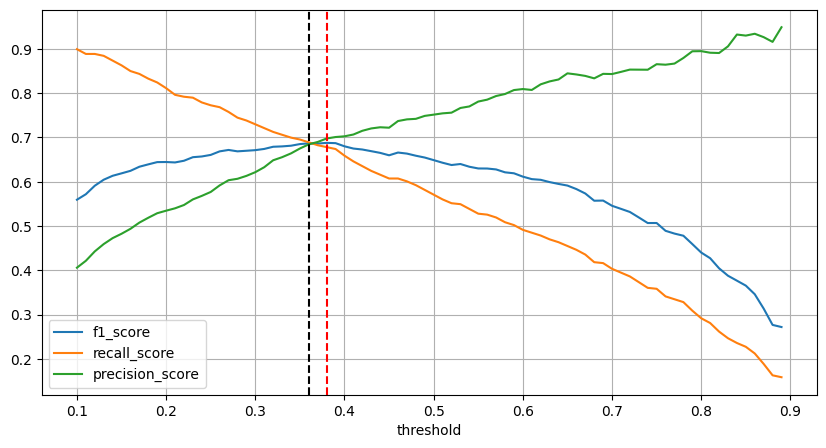
A picture containing text, screenshot, diagram, plot

Description automatically generated

**Figura 16. Histograma de Probabilidade Regressão Logística**

Na figura acima é possível perceber que existem casos de *churn* a partir de probabilidades bem baixas, logo pode-se mudar o ponto de corte de decisão do nosso modelo de 0,5 para outro que melhor se adeque aos dados. Para isso, usa-se o *F1-Score*. O *F1-Score* é a média harmônica da precisão e do *recall*, dessa maneira une ambas as métricas em uma.

Para encontrar o melhor *threshold* temos 2 opções: maximizar o *F1-Score* ou encontrar o ponto em que a precisão e o *recall* se encontram. A figura abaixo mostra o valor das métricas para cada ponto de corte, sendo a linha tracejada preta o ponto onde todas as curvas se encontram e a vermelha onde o *F1-Score* é máximo.



**Figura 17. Melhor *Treshold* Regressão Logística**

Analisando o gráfico acima é possível verificar que o ponto onde as métricas se cruzam é no corte em 0,36. Ao aplicar este ponto de corte é gerada novamente a matriz de confusão abaixo.

A picture containing text, screenshot, font, diagram

Description automatically generated

**Figura 18. Matriz de Confusão Regressão com melhores Hiperparâmetros e *Threshold***

É notável que a matriz de confusão passa a acertar 68,9% dos casos de *churn* sendo bem superior aos 57,1% encontrados anteriormente.

Aplicando os mesmos métodos para o *Gradient Boosting* o melhor ponto de corte foi em 0,43.

A picture containing line, plot, diagram, text

Description automatically generated

**Figura 19. Melhor *Treshold* *Gradient Boosting***

**A picture containing text, screenshot, font, number

Description automatically generated**

**Figura 20. Matriz de Confusão *Gradient Boosting* com melhores Hiperparâmetros e *Threshold***

Novamente é possível notar a melhora do modelo saindo de 85,6% de acerto para 89,1%.

# 6. Conclusões

Ao longo deste artigo foi possível mostrar que mesmo com *datasets* desbalanceados podemos através de técnicas de pré-processamento de dados, aprendizado de máquinas, análises exploratórias entre outros resolver o problema de identificação de clientes que deixaram o banco e ainda melhorar os resultados dos modelos alterando alguns hiperparâmetros e a maneira como utilizamos as saídas do algoritmo.

# Referências

[1] MACHADO, Dalmo. Coeficiente de Correlação de Pearson. Disponível em: < http://ole.uff.br/wp-content/uploads/sites/419/2019/04/Aula\_03\_a\_Pearson.pdf>. Acesso em: 04 de maio de 2023.

[2] Regressão Logística. Disponível em: < https://edisciplinas.usp.br/pluginfile.php/3769787/mod\_resource/content/1/09\_RegressaoLogistica.pdf>. Acesso em: 04 de maio de 2023.

[3] FILHO, Mario. Precisão, Recall e F1 Score Em Machine Learning. Disponível em: < https://mariofilho.com/precisao-recall-e-f1-score-em-machine-learning/>. Acesso em: 04 de maio de 2023.

[4] O que é um outlier. Disponível em: http://ead.pucgoias.edu.br/blog/outlier#:~:text=O%20que%20%C3%A9%20um%20outlier,express%C3%A3o%20do%20dia%20a%20dia.>. Acesso em: 04 de maio de 2023.

[5] BHANDARI, Aniruddha. Guide to AUC ROC Curve in Machine Learning: What Is Specificity?. Disponível em: < https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/#:~:text=The%20ROC%20AUC%20score%20tells,and%20the%20Negative%20class%20points. />. Acesso em: 04 de maio de 2023.