Análise de Crédito

Yves Emmanuel
Centro de Informática (UFPE)
Recife, Brasil
yefo@cin.ufpe.br

***Abstract*—Este projeto teve como objetivo analisar a capacidade de diferentes modelos de machine learning em prever a qualidade de pagamentos de clientes. Foram treinados modelos de *Multilayer Perceptron* (MLP), *ensemble MLP*, *Decision Tree*, *Random Forest* e XGBoost, e seus hiperparâmetros foram tunados com o uso do Optuna Framework, visando minimizar a função de custo dos modelos. A métrica principal de análise dos modelos foi o KS-test, onde os resultados indicam que o modelo XGBoost apresentou a melhor performance na classificação de bons e maus pagadores. Este estudo demonstra o potencial do uso de técnicas de *machine learning* em análises de crédito.**

***Index Terms*—Análise de crédito, Multilayer perceptron, Modelos ensemble, problemas de classificação.**

I. Introdução

A análise de crédito é uma etapa fundamental no processo de concessão de empréstimos. Tradicionalmente, essa análise é feita com base em informações financeiras e cadastrais dos clientes, tais como histórico de pagamentos, renda e emprego. No entanto, com o crescente volume de dados disponíveis, tornou-se necessário o uso de técnicas de aprendizado de máquina para automatizar e aprimorar esse processo de análise.

De acordo com um relatório da consultoria Mckinsey (2019), o uso de modelos de machine learning em análises de crédito pode reduzir o tempo necessário para aprovar ou negar um empréstimo, além de aumentar a precisão na classificação de bons e maus pagadores. Esses benefícios são fundamentais para as instituições financeiras, pois reduzem custos e riscos associados à concessão de crédito. [1]

No contexto de uma economia em constante evolução, é fundamental que as instituições financeiras estejam preparadas para lidar com o crescente volume de transações e informações relacionadas à concessão de crédito. A aplicação de técnicas de aprendizado de máquina pode ser um diferencial competitivo para essas instituições, permitindo uma análise mais precisa e eficiente de dados complexos.

II. O conjunto de dados

O conjunto de dados utilizado neste projeto contém informações de clientes de uma instituição financeira, como idade, gênero, renda, histórico de pagamentos e saldo em conta. Esses dados passaram por um processo de pré-processamento e normalização para garantir que os modelos de machine learning pudessem ser treinados adequadamente. No entanto, foi identificado um problema de desequilíbrio (figura 1) entre as classes de bons e maus pagadores, o que pode levar a uma baixa performance do modelo na detecção de clientes que apresentam risco de inadimplência.

Para lidar com esse problema, foi realizado um downsample na classe majoritária (bons pagadores) de forma a igualar a quantidade de exemplos em ambas as classes. Essa técnica visa melhorar a capacidade do modelo de identificar a classe minoritária (maus pagadores), que é geralmente a mais importante para o negócio em questão.

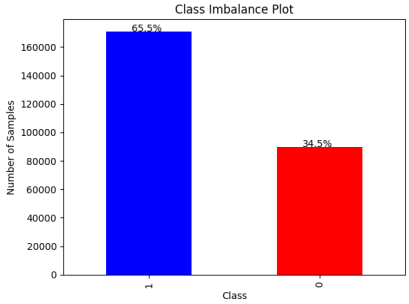

Fig. 1. Plot do desequilíbrio de dados

Modelos

O *pipeline* de treinamento dos modelos consistiu em:

1) Tunagem dos hiperparâmetros do modelo, visando reduzir a função custo;
2) Treinamento com os melhores parâmetros encontrados;
3) Avaliação do modelo.

Resultados

Multilayer Perceptron

A MLP foi o primeiro modelo de machine learning treinado e avaliado nesse projeto. Os hiperparâmetros selecionados para tunagem foram:

	possible values
hidden_layers_units	[32, 128, 256]
hidden_layers	[1.2]
max_iter	[100, 150, 300]
learning_rate	range[0.001, 0.1]
batch_size	[32, 64, 128]
activation	[tanh, relu, sigmoid]
optimizer	[SGD, Adam]
loss_function	[binary_crossentropy, mse]
dropout_rate	range[0.1, 0.5]

O desempenho do modelo na separação das classes de pagadores bons e maus foi demonstrado como relativamente bom pela avaliação (figura 2).

	best values
hidden_layers_units	256
hidden_layers	2
max_iter	100
learning_rate	0.0874
batch_size	32
activation	tanh
optimizer	SGD
loss_function	mse
dropout_rate	0.29

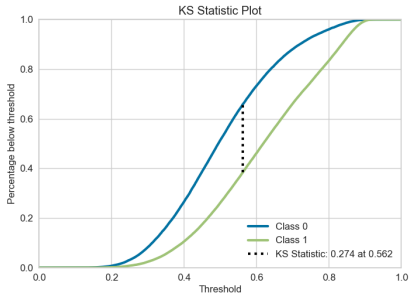

Fig. 2. KS-test da MLP

Ensemble MLP

Depois de treinar a MLP, foi desenvolvido um modelo ensemble usando cinco dessas MLPs em diferentes fases de treinamento. Uma melhoria significativa no modelo pode ser observada em termos de KS (figura 3).

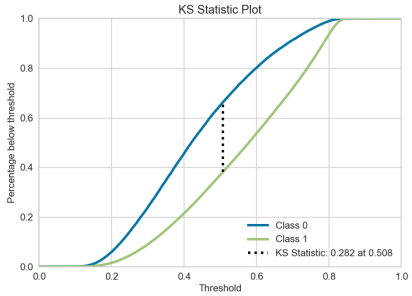

Fig. 3. KS-test do Ensemble

Decision Tree

Posteriormente, uma árvore de decisão foi tunado utilizando os seguintes parâmetros:

	possible values
splitter	[best, random]
max_depth	range[2, 50]
min_samples_split	range[2, 20]
min_samples_leaf	range[1, 10]
criterion	gini

A avaliação da árvore de decisão com os melhores parâmetros (figura 4) resultou em um KS relativamente pior do que a MLP (figura 4), o que é esperado, visto que a árvore de decisão é um modelo mais simples e menos flexível. No entanto, a árvore de decisão pode apresentar um bom desempenho, quando o domínio for conhecido, em termos de interpretabilidade, permitindo a identificação dos atributos mais importantes para a classificação de bons e maus pagadores.

	best values
splitter	best
max_depth	5
min_samples_split	19
min_samples_leaf	10
criterion	gini

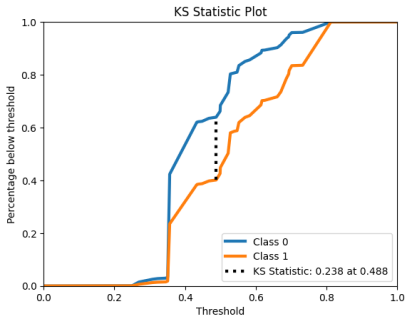

Fig. 4. KS-test da Árvore de Decião

Random Forest

A Random Forest é um ensemble de árvores de decisão, mas optou-se por ajustar os hiperparâmetros das árvores que compõem o classificador. Portanto, os seguintes hiperparâmetros foram tunados:

	possible values
n_estimators	range[50, 200]
max_depth	range[2, 50]
min_samples_split	range[2, 20]
min_samples_leaf	range[1, 10]
criterion	[gini, entropy]
max_features	[sqrt, log2]

Entretanto, com esse modelo tive problemas de instabilidade e interpolação que foram esclarecidos em aula pelo professor Germano Vasconcelos. Dessa forma, ao estudar a importância de cada hiperparâmetro no estudo do Optuna, pude realizar alterações nos parâmetros e obtive um resultado mais estável com os hiperparâmetros (figura 5) testados.

	best values
n_estimators	50
max_depth	10
min_samples_split	2
min_samples_leaf	1
criterion	gini
max_features	None

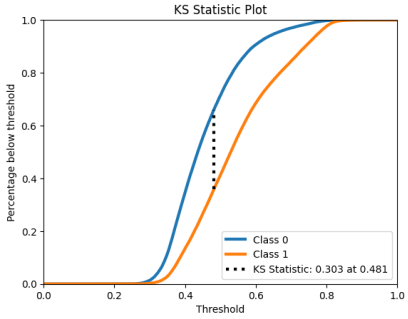

Fig. 5. KS-test da *Random Forest*

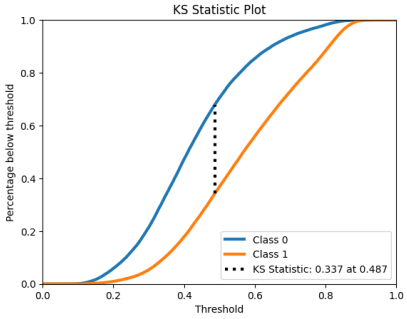

Fig. 7. KS-test do XGBoost

XGBoost

O modelo XGBoost é conhecido por sua capacidade de lidar com conjuntos de dados grandes e complexos, e apresenta um desempenho superior em relação a outros modelos de machine learning, como a MLP e árvore de decisão. Isso se deve ao fato do XGBoost usar uma abordagem baseada em gradient boosting, o que o torna eficiente em reduzir o viés do modelo e melhorar sua precisão.

Os parâmetros tunados foram:

	possible values
n_estimators	range[50, 200]
max_depth	range[3, 10]
learning_rate	range[0.001, 0.1]
min_child_weight	range[1, 10]

De fato, foi o estado da arte do projeto. O modelo conseguiu de forma excepcional distinguir as classes.

Na análise por percentil (população de teste), pode-se observar que nos últimos 3% da população a taxa de acertos é muito boa. Isso indica, que num cenário de tomadas de decisão real, o modelo poderia ser usado com certa segurança para realizar classificações reais.

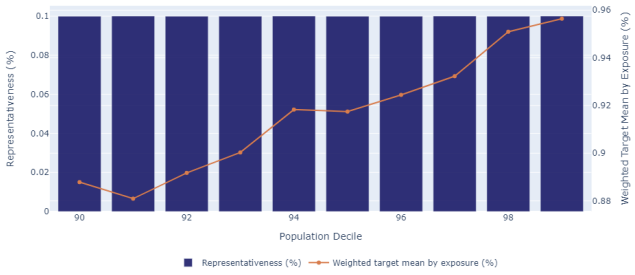

Fig. 6. Análise por percentil do *XGBoost*

Tabela com métricas

	KS	**Accuracy**	**Log-loss**
MLP	0.2740	0.6612	0.6064
Ensemble-MLP	0.2820	0.6373	0.6350
Decision Tree	0.2380	0.5974	0.6505
Random Forest	0.3030	0.6306	0.6324
XGBoost	0.3370	0.6567	0.6091

REFERÊNCIAS

[1] Babel, Oliveira, Buehler K., Pivonka A., Richardson B., Waldron D. (2019). "Derisking machine learning and artificial intelligence." In Mckinsey, Available at: https://www.mckinsey.com/capabilities/risk-and-resilience/our-insights/derisking-machine-learning-and-artificial-intelligence.