Introdução

Esse artigo é a parte 2 do último estudo feito sobre a churn rate do banco. O link para a parte 1, que contém a análise completa dos dados feita em python, pode ser encontrada clicando aqui.

Nessa parte vamos fazer um modelo de machine learning para prever se o cliente vai continuar a ser cliente do banco ou não. Para isso, vamos implementar, em Python, o modelo XGBoost de machine learning.

Objetivo

Temos como objetivo montar um modelo que consiga prever se o banco vai perder um cliente ou não.

Montagem do Modelo

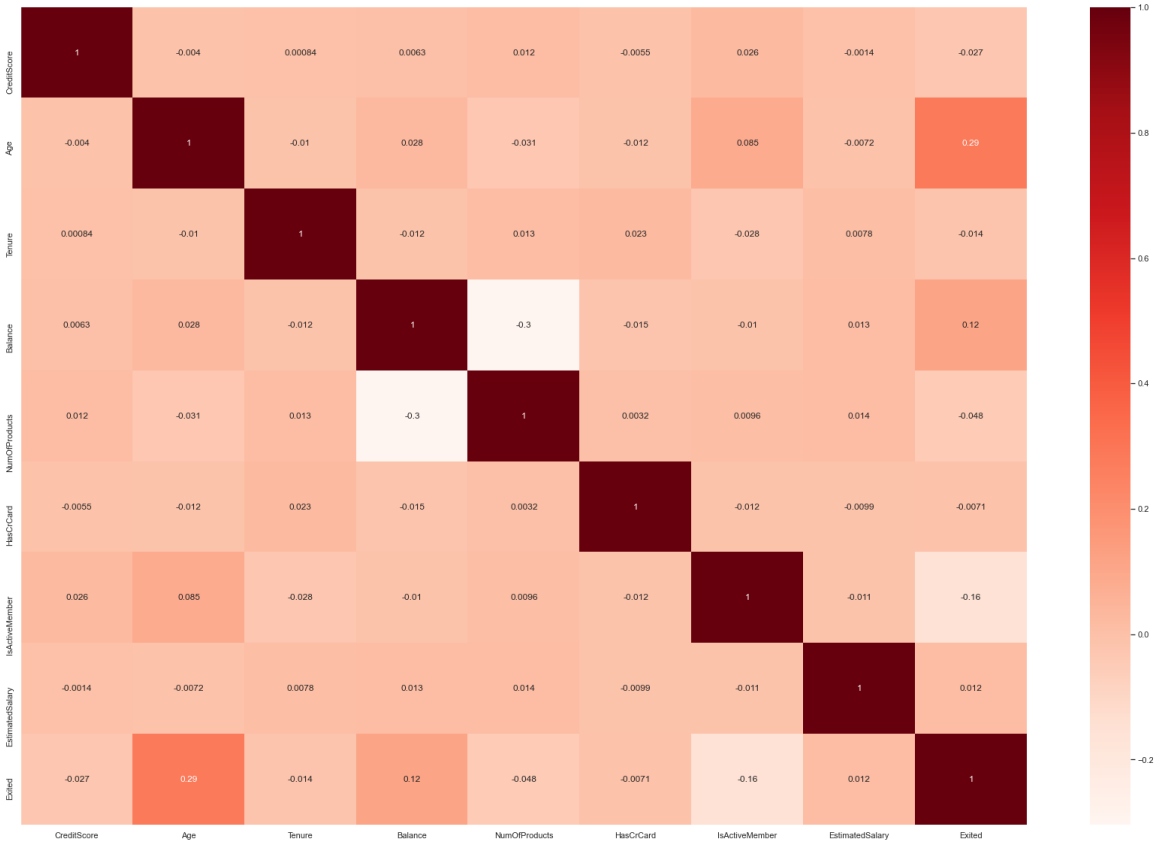
Figura 1 – Dados

vNumber	CustomerId	Surname	CreditScore	Geography	Gender	Age	Tenure	Balance	NumOfProducts	HasCrCard	IsActiveMember	EstimatedSalary	Exited
1	15634602	Hargrave	619	France	Female	42	2	0.00	1	1	1	101348.88	1
2	15647311	Hill	608	Spain	Female	41	1	83807.86	1	0	1	112542.58	0
3	15619304	Onio	502	France	Female	42	8	159660.80	3	1	0	113931.57	1
4	15701354	Boni	699	France	Female	39	1	0.00	2	0	0	93826.63	0
5	15737888	Mitchell	850	Spain	Female	43	2	125510.82	1	1	1	79084.10	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...
9996	15606229	Obijiaku	771	France	Male	39	5	0.00	2	1	0	96270.64	0
9997	15569892	Johnstone	516	France	Male	35	10	57369.61	1	1	1	101699.77	0
9998	15584532	Liu	709	France	Female	36	7	0.00	1	0	1	42085.58	1
9999	15682355	Sabbatini	772	Germany	Male	42	3	75075.31	2	1	0	92888.52	1
10000	15628319	Walker	792	France	Female	28	4	130142.79	1	1	0	38190.78	0

Para começar, vamos relembrar os dados, vistos na figura 1. Nela vemos que precisamos fazer uma limpeza nos dados, por exemplo, retirar algumas colunas como ‘CustomerId’ e ‘Surname’, verificar se em alguma das colunas possuem dados faltando e, por fim, utilizar a técnica de ‘getdummies’ para transformarmos as colunas ‘Geography’ e ‘Gender’ que contém dados categóricos.

Na última coluna da figura 1, temos nosso target, ‘Exited’. Ela contém a informação se a pessoa ainda é cliente do banco (0) ou não (1). Nós estamos interessados em prever se o cliente deixara o banco, ou seja, quando tivermos ‘Exited’ = 1, pois é nesse caso que o banco perde o cliente e, consequentemente, dinheiro. Portanto, não estamos muito preocupados caso a precisão de quanto ‘Exited’ = 0 não for alta.

Figura 2 – Heatmap dos dados



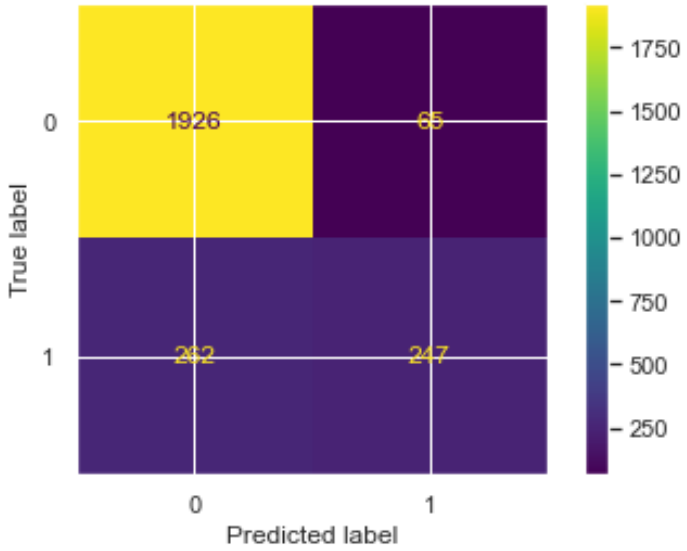
No heatmap, figura 2, temos quão correlacionadas as features estão do target. Vemos que ‘Age’ e ‘Balance’ são as mais correlacionadas positivamente a ‘Exited’ (já sabíamos disso da análise da parte 1), ou seja, contribuem positivamente para o cliente deixar o banco (‘Exited’ = 1).

Com os dados prontos, podemos proceder para montar o modelo. Antes de monta-lo, vamos verificar se o dataset está balanceado. Temos que a coluna ‘Exited possui apenas 20,37 % de de ‘1’, ou seja, não estamos trabalhando com um dataset balanceado e isso pode comprometer os resultados do modelo. Mais a frente, podemos utilizar algumas técnicas para tentar balancear o dataset e melhorar os resultados obtidos.

Resultados e Discussão

Com o modelo feito, a métrica de avaliação foi ‘aucpr’ (area under the curve precision and recall) e, depois de 27 iterações, o resultado foi de 0.72603. Todos os hyperparameters nessa primeira rodada foram deixados com os valores default. De posse dos resultados podemos plotar a confusion matrix para avaliarmos a precision e o recall.

Figura 3 – Confusion matrix com hyperparameters default



Na figura 3, temos que 96,73 % dos clientes que não deixaram o banco foram classificados corretamente, no entanto, apenas 48,52 % dos que não são mais cliente foram classificados corretamente. Como nos mais interessa os classificar os clientes que deixaram o banco, pois nós não queremos perde-los, o resultado de 48,52 % é insatisfatório. Um dos motivos desse resultado é o dataset desbalanceado que temos.

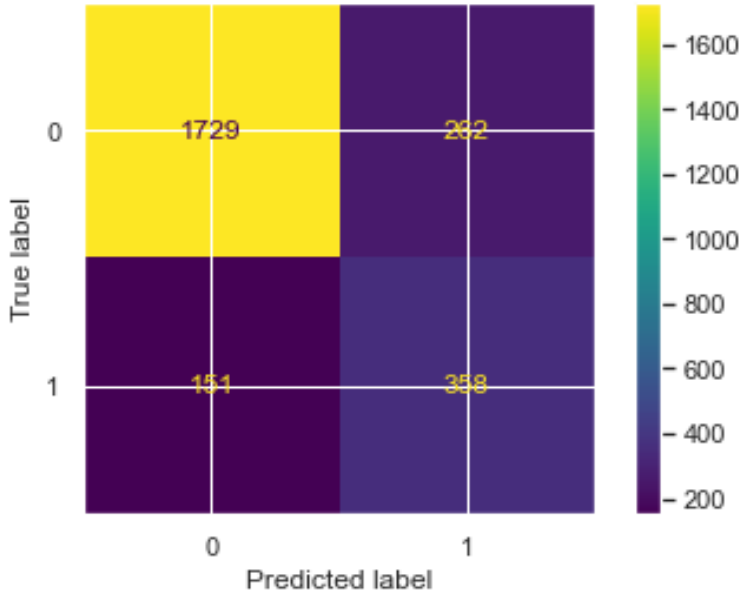
Para tentarmos melhorarmos os resultados, foi feita uma mudança nos hyperparameters, por exemplo, 'learning_rate', 'max_depth', 'reg_lambda' e etc.

Tabela 1 - Hyperparameters

	gamma	learning_rate	max_depth	reg_lambda	sacle_pos_weight
default	0	0.3	6	1	1
otimizados	0.25	0.1	4	0	3

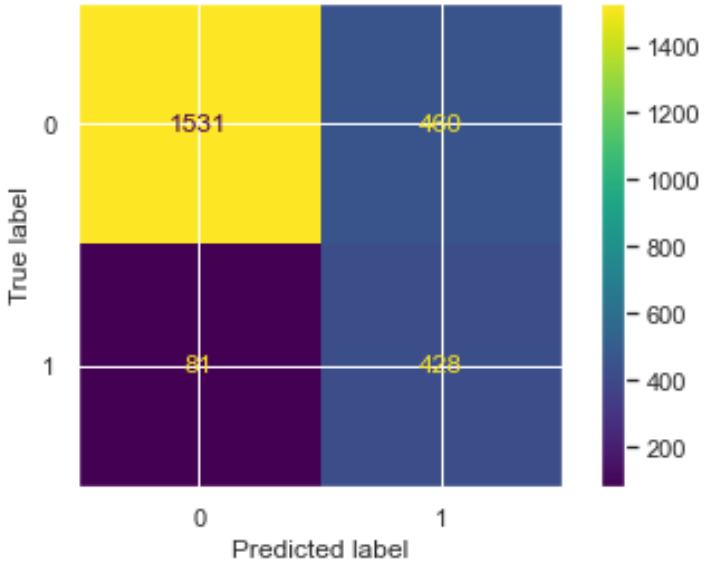
Na tabela 1, vemos as mudanças feitas nos hyperparameters e, em seguidas, elas foram aplicadas no próximo modelo.

Figura 4 – Confusion matrix com mudança nos hyperparameters



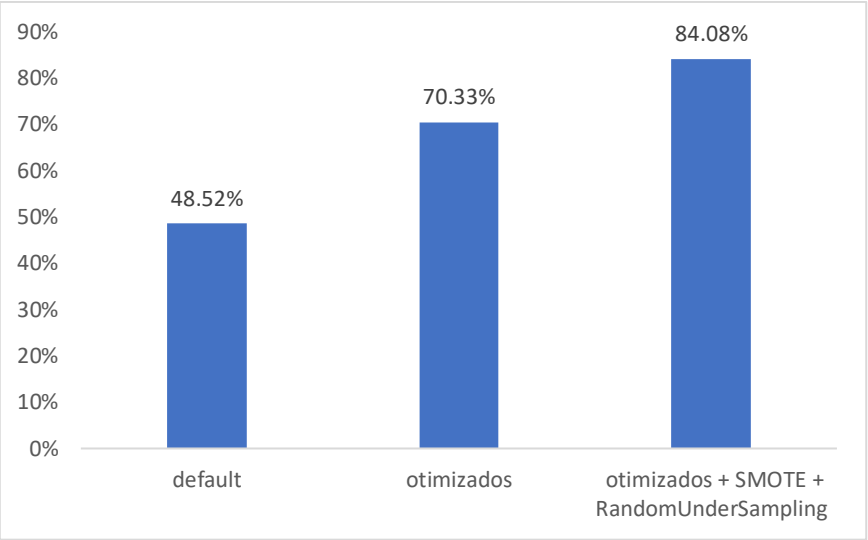
Com esses parâmetros alterados, temos o resultado do 'aucpr' de 0.72748 queé quase o mesmo resultado obtido anteriormente, mas analisando a segunda confusion matrix, figura 4, temos um ganho significativo na identificação dos clientes que deixaram o banco, 70,33 %, o que é um resultado melhor e mais aceitável do que o obtido sem alterar o hyperparameters.

Figura 5 – Confusion matrix hyperparameters otimizados e técnicas de balanceamento do dataset



Podemos ainda melhorar esse resultado utilizando uma técnica de balanceamento do dataset e combina-la com os hyperparameters otimizados. Com a técnica de SMOTE (oversampling) + RandomUnderSampler numa pipeline e combinando com os hyperparameters otimizados conseguimos um resultado superior, visto na figura 5, do que o obtido na figura 4. Agora, conseguimos classificar corretamente os clientes que deixaram o banco 84,08 % das vezes.

Figura 6 – Desempenho do modelo



A comparação dos 3 resultados pode ser vista na figura 6 e, claramente, a ultima iteração do modelo é melhor e que deve ser colocada para produção.

Conclusões

O objetivo do modelo era identificar com precisão as pessoas deixariam de ser clientes do banco e ele o atingiu com uma precisão de 84,08 % dos casos. Esse resultado representa que futuramente será mais fácil identificar quem poderá deixar de ser cliente do banco, avaliar os motivos e trabalhar em cima deles para poder manter o cliente.