INSTITUTO FEDERAL DO ESPÍRITO SANTO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA

RODRIGO SEIDEL

USO DE HEURÍSTICAS PARA O AJUSTE DOS HIPERPARÂMETROS DO *XGBOOST* APLICADO À RECOMENDAÇÃO DE PROPOSTAS DE OPERAÇÃO DE REFINANCIAMENTO

Serra
2020

RODRIGO SEIDEL

USO DE HEURÍSTICAS PARA O AJUSTE DOS HIPERPARÂMETROS DO *XGBOOST* APLICADO À RECOMENDAÇÃO DE PROPOSTAS DE OPERAÇÃO DE REFINANCIAMENTO

Anteprojeto apresentado ao Programa de Pós-Graduação em Computação Aplicada do Instituto Federal do Espírito Santo, como requisito para aprovação da Disciplina de Pesquisa em Computação Aplicada.

Orientador: Profº. Dr· Leandro Colombi Resendo

Orientadora: Profª. Dra Karin Satie Komati

Serra

2020

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

O Sistema Financeiro Nacional (SFN) é formado por um conjunto de entidades e instituições que promovem a intermediação financeira, isto é, o encontro entre credores e tomadores de recursos. É por meio do sistema financeiro que as pessoas, as empresas e o governo circulam a maior parte dos seus ativos, pagam suas dívidas e realizam seus investimentos (BCB, 2020g). Fazem parte do SFN bancos e caixas econômicas, corretoras de câmbio, fintechs, administradora de consórcios, cooperativas de crédito, instituições de pagamento, corretoras e a distribuidoras de títulos e de valores mobiliários e demais instituições não bancárias (BCB, 2020a). As instituições não bancárias recebem esta denominação pois não recebem depósitos à vista, nem podem criar moeda (por meio de operações de crédito) (BCB, 2020c).

As sociedades de crédito, financiamento e investimento (SCFI), conhecidas como "financeiras", são instituições não bancárias e privadas que fornecem empréstimo e financiamento para aquisição de bens, serviços e capital de giro e são supervisionadas pelo Banco Central. Quando é realizado o empréstimo, o dinheiro recebido pelo tomador não tem destinação obrigatória. Já quando é realizado um financiamento o dinheiro recebido pelo tomador está vinculado à aquisição de determinado bem ou serviço como, por exemplo, a aquisição de um veículo ou equipamento (BCB, 2020d).

O empréstimo é uma operação mais simples e rápida, por não haver uma destinação específica para os recursos concedidos. Contudo, por serem mais simples, essas operações podem envolver maiores custos e riscos para a instituição. Por isso, antes da concessão do empréstimo, o cliente é submetido a uma avaliação de sua capacidade de pagamento e de seu histórico de crédito. Os tipos mais comuns de operações de empréstimo são: empréstimos pessoais, empréstimos consignados, cartão de crédito e cheque especial (BCB, 2020b).

O empréstimo consignado é uma modalidade de crédito em que o desconto da prestação é feito diretamente na folha de pagamento ou de benefício previdenciário do tomador. Essa característica leva a uma redução do risco de inadimplência, já que o colateral do empréstimo é parte do salário, o que permite ao credor uma redução na taxa de juros cobrada (BCB, 2018). Também existe a modalidade cartão de crédito consignado, na qual a o valor da fatura pode ser descontado, total ou parcialmente, automaticamente na folha de pagamento do tomador (BCB, 2020e).

Margem consignável é o valor máximo que pode ser descontado do salário, do benefício ou da pensão para pagamento de prestação do empréstimo consignado (BCB, 2013). Atualmente, conforme estabelecido na Lei nº 13.172, de 21 de outubro de 2015 (BRASIL,

2015), esse valor é 30% + 5% = 35%, sendo:

- 30%, referentes a empréstimo consignado convencional.
- 5%, referentes a despesas e saques exclusivamente com cartão de crédito consignado.

Segundo o Relatório de Economia Bancária 2019 (BCB, 2020f), que trata de questões que dizem respeito ao SFN e às relações entre instituições e seus clientes, o crédito de forma geral manteve aceleração do crescimento, que repercutiu na evolução dos empréstimos para pessoas físicas. Especificamente, a modalidade de crédito consignado teve um crescimento de 14,1% frente ao ano de 2018.

A Lei nº 10.820, de 17 de dezembro de 2003 (BRASIL, 2003), define regras gerais para desconto de prestações em folha de pagamento e permite operações chamadas refinanciamento ou renovação de empréstimo, através das quais o tomador pode aumentar o valor financiado, estender o prazo de pagamento ou ambos, sem que se aumente o comprometimento mensal, sempre respeitando a margem consignável. O refinanciamento é a repactuação do saldo devedor de um ou mais contratos existentes, na formalização de um novo contrato (RJ, 2019). Neste trabalho será tratado apenas o crédito consignado e sua possibilidade de refinanciamento.

Na Tabela 1 observamos um exemplo de operação refinanciamento. Uma pessoa possui um empréstimo consignado contratado, no qual o valor do saldo devedor é de R$ 3.500,00 a serem pagos em 12 parcelas restantes, sendo R$ 291,66 por parcela. Caso essa pessoa possua margem disponível e precise renegociar o contrato de forma a liberar um valor para uso dela, a financeira realiza um novo contrato de R$ 5.000,00 que quita o contrato atual, o que deixa para o cliente um "troco" de R$ 1.500,00. Neste exemplo, o cliente optou por manter quase que estável o valor da parcela e estender o prazo de pagamento para 18 meses.

Tabela 1 – Exemplo operação refinanciamento

	Operação Atual	Refinanciamento
Saldo devedor	R$ 3.500,00	R$ 5.000,00
Parcelas a pagar	12	18
Valor parcela	R$ 291,66	R$ 277,77
Troco para o cliente	-	R$ 1.500,00

Fonte: Elaborado pelo autor (2020).

O lucro das instituições financeiras vem do *spread* bancário que é a diferença, em pontos percentuais, entre a taxa de juros pactuada nos empréstimos e financiamentos e a taxa de captação. A taxa de captação é a remuneração paga pelas instituições financeiras em aplicações financeiras - caderneta de poupança, Certificado de Depósito Bancário (CDB),

etc, com o objetivo de captar recursos para conceder empréstimos (BCB, 2016). O lucro da financeira vem a partir das operações de financiamento realizadas, consequentemente, as operações de refinanciamento são uma forma gerar mais *spread* para a financeira, por meio de novas operações e mantendo a relação com o cliente ativa, além de beneficiar o cliente ou liberando mais recursos ou aumentando prazo ou reduzindo taxa de juros.

1.2 PROBLEMA

Esta pesquisa versará sobre o estudo de caso de uma financeira situada no Rio de Janeiro, fundada no início da década de 2000. Atualmente possui 110.000 contratos de empréstimo consignado ativos, aproximadamente. Possui dezenas de lojas de atendimento, distribuídas entre os estados do RJ, MA, MG, PB, PI, PR, RN, SC e SP. A partir daqui será chamada de Financeira A.

A Financeira A adotou como estratégia de mercado vender produtos de crédito de bancos com os quais ela firmou parceria. Os produtos comercializados por ela são, além do crédito consignado e seu refinanciamento, consórcios, seguros e crédito pessoal.

Esta Financeira, visando aumentar sua receita através do aumento do número de contratos de refinanciamento de crédito consignado, celebrou um contrato com um *call center*. Este, por sua vez, realiza contatos diretos com os clientes, ofertando-lhes a oportunidade de refinanciamento.

O contato atualmente é feito a partir de listas de clientes fornecidos pelos bancos parceiros e através de listas de clientes geradas pela própria Financeira. No entanto, cada instituição financeira tem pleno domínio apenas de seu conjunto de dados e clientes, portanto, é comum ocorrer de um cliente estar numa lista de um banco, porém este banco não tem conhecimento se o cliente fez recentemente uma operação de consignação, através da Financeira A ou com outra instituição. Este fato indica que a necessidade do cliente fazer um refinanciamento é baixa, uma vez que ele acabou que receber recursos do empréstimo recém realizado. Nestes casos temos como provável consequência um contato do *call center* sem sucesso com o cliente, uma vez que este tende a não fazer um refinanciamento.

Outra forma de o *call center* estabelecer contato com o cliente é através de listas produzidas e fornecidas pela própria Financeira A, porém estas são preparadas com base em experiências, opiniões e entendimentos de especialistas humanos, possíveis de serem feitas sem nenhum estudo aprofundado.

Nas duas formas de identificação do cliente a ser contactado, ou se tem uma visão parcial ou não há uma análise aprofundada do perfil do cliente. Assim, existe uma grande probabilidade de se estabelecer contato com o cliente e este não ser convertido em um refinanciamento.

Isso não é desejável nem pela Financeira A nem pelo *call center*, pois existe gasto de tempo e recurso para estabelecimento da comunicação e um possível desgaste do relacionamento do cliente com a Financeira, uma vez que o contato pode ser visto de forma negativa pelo cliente. Com isso, temos o aumento dos custos (através da contratação do *call center*) e diminuição da receita (por conta de o número de operações de refinanciamento ser inferior ao esperado), por conta da utilização de uma base de clientes que não se sabe se tem bom potencial para a operação de refinanciamento.

1.3 PROPOSTA

Na literatura, a exploração de conjuntos de dados afim de se realizar uma classificação de forma preditiva é apresentada em diversos trabalhos. No trabalho de Hassan et al. (2019), utilizou-se *dataset* de um banco português com o objetivo de identificar os clientes que teriam maior probabilidade de realização de depósito a longo prazo, através de contato via *call center*. Foi realizada a comparação dos seguintes classificadores: *k-nearest neighbour* (KNN), Regressão Logística (RL), Naive Bayes (NB), *Suport Vector Machine* (SVM), *Decision Tree* e *Neural Network* (NN). Utilizou-se como métricas para comparação, acurácia, precisão, *recall* e *F-Measure*. Os resultados mostraram que a Regressão Logística superou todos os outros modelos de data mining utilizados.

No trabalho de Albiero et al. (2019) foram utilizados os classificadores *XGBoost* e *Logistic Regression* com o objetivo de identificar fraudes no uso de energia elétrica utilizando dados de uma empresa de energia elétrica situada em Campinas, SP. Para realizar a classificação foram utilizados dois *datasets*, um com histórico de inspeção e outro sem o histórico. Como métricas para comparação foram utilizados *F1-score*, precisão e *recall*.

Já no trabalho de Pellicer e Pait (2020) foi realizada a otimização de hiperparâmetros como forma de realizar o *tunning* de modelos de *machine learning*. No trabalho foi proposto o uso do BarySearch e realizada comparação de seus resultados com os resultados de outros algoritmos: *Random Search*, *Simulated Annealing*, CMA-ES, TPE do HyperOpt, TPOT e Hyperband. Os algoritmos foram aplicados para otimização de 7 hiperparâmetros numéricos nos modelos LightGBM e Rede Neural. Os resultados demonstraram que o BarySearch conseguiu ter melhor desempenho em boa parte dos testes que foram realizados em bases da UCI (http://archive.ics.uci.edu/ml), sendo elas, *Ionsphere*, *Credit*, *SPAM*, *Breast-Cancer* e *Dermatology*.

A Financeira A possui um histórico de operações de crédito consignado e dos refinanciamentos realizados, além de informações cadastrais dos clientes. Com esses dados acredita-se que é possível realizar a classificação dos clientes como possíveis contratantes de um refinanciamento ou não. Essa classificação será utilizada na produção da lista de clientes para contato entregue ao *call center*, de maneira que somente estarão presentes os clientes

classificados como possíveis contratantes de refinanciamento.

A proposta deste trabalho é classificar, por meio da utilização do *XGBoost*, os clientes como possíveis contratantes de um refinanciamento. Visando atingir melhores resultados na classificação serão utilizadas metaheurísticas para *tunning* de alguns hiperparâmetros do *XGBoost*.

1.4 OBJETIVO GERAL

O objetivo principal deste trabalho é identificar os possíveis clientes de refinanciamento através do uso do classificador *XGBoost*.

1.5 OBJETIVOS ESPECÍFICOS

Para atingir o objetivo geral deste trabalho, os seguintes objetivos específicos foram traçados:

- Extrair o dataset, baseado nas entrevistas realizadas coma equipe da Financeira A, contendo os dados de contrato de empréstimo consignado, através da construção do processo de extração de dados utilizando a ferramenta Pentaho Data Integration.

- Realizar a exploração dos dados para entender como se comporta sua distribução frente aos atributos existentes, bem como avaliar o balanceamento em relação ao atrituto alvo (Fez refinanciamento), utilizando as bibliotecas para a linguagem Python: pandas, seaborn e matplotlib.

- Entender de maneira aprofundada o funcionamento do XGBoost, visando explorar e confirmar os hiperparâmetros que serão calibrados.

- Implementar as heurísticas para a calibração do classificador *XGBoost* para realização da calibração dos hiperparâmetros selecionados baseando-se nas seguintes metaheurísticas:

  - GRASP
  - *BarySearch*
  - *Simulated Annealing*
  - Genético
  - *Random Search*

2 MATERIAIS E MÉTODOS

Será utilizada uma base de dados contendo somente dados dos contratos de empréstimo consignado realizados no período de 2010/04 a 2020/03 de uma financeira com sede no Rio de Janeiro. A base é composta por 215.672 registros com 23 atributos, como pode ser visto na Tabela 2. Todos os dados que poderiam identificar os contratos ou os tomadores foram retirados antes da disponibilização da base para os experimentos deste trabalho.

Cada registro dessa base de dados possui uma classificação (rótulo) binária, que identifica se foi realizada a operação de refinanciamento: SIM ou NÃO. O primeiro rótulo se refere a todos os contratos que em sua vigência foi realizada uma operação de refinanciamento, já o segundo, se refere a contratos que não tiveram operação de refinanciamento realizada.

É possível agrupar as informações com relação ao:

- Tomador com 8 atributos: (1) Data nascimento, (2) Sexo, (3) Estado civil, (4) Profissão, (5) Salário, (6) Bairro, (7) Cidade, (8) UF.

- Contrato com 15 atributos: (9) ID Parceiro comercial, (10) ID Produto, (11) Tipo produto, (12) Grupo produto, (13) ID Posto atendimento, (14) Valor parcela, (15) Quantidade de parcelas, (16) Valor contrato, (17) Forma pagamento, (18) Flag contrato digital, (19) Flag condição comercial, (20) Data registro contrato, (21) Data fim previsto contrato, (22) Data registro refinanciamento, (23) Fez refinanciamento.

O método que será utilizado para realizar a classificação dos clientes que podem realizar um refinanciamento é o *XGBoost*. Busca-se sempre alcançar o melhor desempenho possível, o que nos leva a necessidade de realizar o *tunning* de um conjunto de hiperparâmetros do modelo utilizado, para que esse obtenha máxima eficiência para dados do problema investigado.

De acordo com Probst (2019), hiperparâmetros são parâmetros que devem ser definidos antes da execução de um algoritmo de aprendizado de máquina, ao contrário dos parâmetros normais de um algoritmo que não são fixos antes da execução, mas otimizados durante o treinamento do algoritmo. Alguns exemplos são, o número de variáveis que são consideradas em cada divisão em uma floresta aleatória (*random forest*), o número de etapas de reforço no aumento de gradiente, o número k no *K-Nearest Neighbor* (KNN), o *kernel* em máquinas de vetor de suporte (*support vector machine* - SVM) e vários outros exemplos. Normalmente, existem hiperparâmetros padrão fornecidos nos pacotes de *software*. Para um determinado problema, eles possivelmente fornecem bons resultados, se forem configurados de forma adequada. Na maioria das vezes, ajustar os hiperparâmetros, buscando um valor ideal

Tabela 2 – Caracterização da base de dados.

Núm.	**Atributo**	**Tipo de Dado**	**Escala de Valores**
1	Data nascimento	Numérico	Intervalar
2	Sexo	Categórico	Feminino
			Masculino
3	Estado civil	Categórico	Casado
			Solteiro
			Viúvo
			Outros
4	Profissão	Simbólico	Nominal
5	Salário	Numérico	Nominal
6	Bairro	Simbólico	Nominal
7	Cidade	Simbólico	Nominal
8	UF	Simbólico	Nominal
9	ID Parceiro comercial	Numérico	Nominal
10	ID Produto	Numérico	Nominal
11	Tipo produto	Simbólico	Fixo "Consignado"
12	Grupo produto	Categórico	Cartão
			Empréstimo
			Portabilidade
			Venda Digital
13	ID Posto atendimento	Numérico	Nominal
14	Valor parcela	Numérico	Nominal
15	Quantidade de parcelas	Numérico	Nominal
16	Valor contrato	Numérico	Nominal
17	Forma pagamento	Categórico	Dinheiro rápido
			DOC
			TED
			Ordem pagamento
18	Flag contrato digital	Categórico	Y
			N
19	Flag condição comercial	Categórico	Y
			N
20	Data registro contrato	Numérico	Intervalar
21	Data fim previsto contrato	Numérico	Intervalar
22	Data registro refinanciamento	Numérico	Intervalar
23	Fez refinanciamento	Categórico	Sim
			Não

Fonte: Elaborado pelo autor (2020).

para eles, pode fornecer melhor desempenho do que usar o valor padrão. O *tunning* dos hiperparâmetros para valores ideais pode fornecer grandes ganhos de desempenho.

No trabalho realizado por Probst, Boulesteix e Bischl (2019) tratou-se o problema de *tunning* de hiperparâmetros de um ponto de vista estatístico e foram sugeridas métricas gerais que quantificam a ajustabilidade de hiperparâmetros de algoritmos, além de fornecer definições teóricas para uma apropriada especificação de espaço de busca no qual o *tunning* deve ser executado. Trabalhou-se com o tunning de hiperparâmetros dos modelos *XGBoost*, *Random Forest*, *Support vector machine*, *K-Nearest Neighbor*, *Decision tree* e *Elastic net*.

A busca por soluções ótimas é inviável para muitos problemas de otimização de importância industrial e científica. Para o problema que será investigado nesse trabalho, a calibração de parâmetros em um classificador, a relação entre os dados de entrada do sistema e as variáveis a serem otimizadas (parâmetros do classificador) não é explícita, tornando inviável a proposição de métodos exatos de otimização. Nesse caso, é necessário a aplicação de métodos heurísticos para a obtenção de soluções de boa qualidade. Tais métodos podem ser criados com base nas relações do problema investigado ou baseados em Metaheurísticas. Metaheurísticas representam uma família de técnicas de otimização aproximada e fornecem soluções aceitáveis em um tempo razoável. Ao contrário dos algoritmos de otimização exatos, as metaheurísticas não garantem a otimização das soluções obtidas (TALBI, 2009).

Para ajustar hiperparâmetros, diferentes estratégias podem ser aplicadas. Espera-se que a estratégia utilizada encontre os melhores valores possíveis de hiperparâmetros e no menor tempo possível. Uma estratégia de ajuste simples que pode ser usada é a pesquisa em grade (*grid search*). Para cada hiperparâmetro, uma quantidade finita de valores possíveis deve ser definida e todas as combinações possíveis de hiperparâmetros são avaliadas. Outra estratégia simples é a busca aleatória (*random search*), onde os valores dos hiperparâmetros são definidos aleatoriamente dentro um determinado espaço definido, por exemplo, usando a distribuição uniforme (PROBST, 2019).

Outras abordagens mais sofisticadas determinam as especificações do hiperparâmetro de forma iterativa. Um exemplo típico disso é a otimização bayesiana, também chamada de otimização baseada em modelo. Na otimização bayesiana, um modelo substituto é treinado com os desempenhos de hiperparâmetros já executados como saída e os hiperparâmetros como entrada. Com a ajuda deste modelo substituto, são propostas novas especificações de hiperparâmetros que atendem a dois requisitos: devem fornecer bons resultados de acordo com o modelo substituto treinado e devem estar em regiões do espaço do hiperparâmetro ainda inexploradas. Na prática, esses dois objetivos são combinados por um critério de preenchimento, também chamado de função de aquisição. Para um dado hiperparâmetro, a média e o desvio padrão do desempenho podem ser estimados por meio do modelo substituto e combinados, por exemplo, com fatores de ponderação na função de aquisição.

A próxima especificação de hiperparâmetro é escolhido como o valor que fornece o melhor critério de preenchimento, o que significa otimizar a função de aquisição (PROBST, 2019).

Ainda no trabalho de Probst (2019) são citados outros procedimentos comuns de *tunning* de hiperparâmetros que são baseados em técnicas de otimização baseadas em gradiente que calculam o gradiente dos hiperparâmetros e, em seguida, usam métodos de pesquisa de descida de gradiente para encontrar o valor ideal. Também são citadas algumas outras estratégias, que incluem algoritmos de enxame, como otimização de enxame de partículas, algoritmos evolutivos, como algoritmos genéticos, *simulated annealing*, entre outros.

Os hiperparâmetros que serão calibrados no XGBoost, utilizando o *booster gbtree* serão (COMMUNITY, 2020):

- n_estimators (int): número de árvores;
- learning_rate (float): taxa de aprendizagem;
- subsample (float): proporção de amostra da instância de treinamento;
- max_depth (int): profundidade máxima da árvore;
- min_child_weight (float): número mínimo de amostras requeridas para formar um nó folha;
- colsample_bytree (float): proporção da amostra de colunas para construir cada árvore.
- colsample_bylevel (float): proporção da amostra de colunas para cada nível.
- reg_alpha (float): termo de regularização L1 sobre os pesos;
- reg_lambda (float): termo de regularização L2 sobre os pesos;

Neste trabalho, para realização do *tunning* serão utilizadas algumas metaheurísticas, sendo elas: GRASP, *BarySearch*, *Simulated Annealing*, Genético e *Random Search*.

O algoritmo GRASP (*Greedy Randomized Adaptive Search Procedure*) foi proposto por Feo e Resende (1995) e de acordo com Resende e Ribeiro (2019) é uma metaheurística iterativa para problemas otimização combinatória, em que cada iteração consiste em duas fases: construção e busca local, como podemos ver na Figura 1. A fase de construção visa produzir uma solução viável, caso a solução não seja viável, então é realizado ajuste para torná-la viável ou realizada uma nova tentativa de construção de uma solução viável.

Uma vez obtida uma solução viável, sua vizinhança é investigada até o mínimo local ser encontrado durante a fase de busca local. A melhor solução geral é mantida no resultado.

Não foi localizado até o momento trabalhos em que se utilizou o GRASP para otimização de hiperparâmetros.

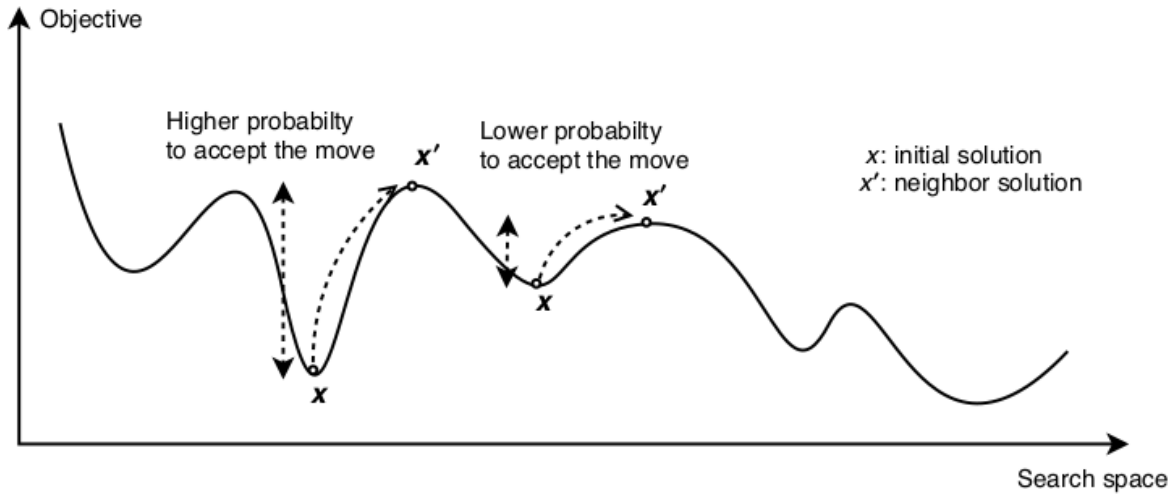
Figura 1 – Pseudo-código GRASP

```
procedure GRASP(Max_Iterations, Seed)
1     Read_Input();
2     for k = 1,...,Max_Iterations do
3           Solution ← Greedy_Randomized_Construction(Seed);
4           if Solution is not feasible then
5                 Solution ← Repair(Solution);
6           end;
7           Solution ← Local_Search(Solution);
8           Update_Solution(Solution, Best_Solution);
9     end;
10    return Best_Solution;
end GRASP.
```

Fonte: Resende e Ribeiro (2019).

O algoritmo Simulated Annealing (SA) é baseado nos princípios da mecânica estatística em que o processo de recozimento requer aquecimento e, em seguida, resfriamento lento de uma substância para obter uma estrutura cristalina forte. A resistência da estrutura depende da taxa de resfriamento dos metais. Se a temperatura inicial não for suficientemente alta ou um resfriamento rápido for aplicado, imperfeições (estados metaestáveis) são obtidas. Cristais fortes são cultivados a partir de um resfriamento lento e cuidadoso. O SA simula as mudanças de energia em um sistema submetido a um processo de resfriamento até que converta para um estado de equilíbrio (TALBI, 2009).

Figura 2 – Simulated Annealing escapando do ótimo local. Quanto mais alta a temperatura, mais significativa é a probabilidade de aceitar o pior movimento.

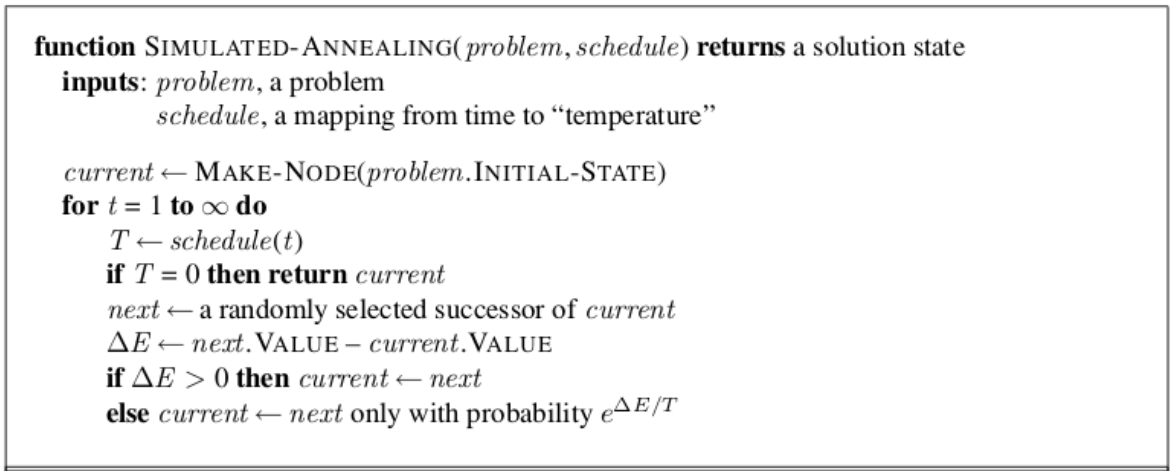


Fonte: Talbi (2009).

Ainda de acordo com Talbi (2009), o objetivo do SA é escapar dos ótimos locais (Figura 2) e, assim, atrasar a convergência e para isso não usa nenhuma informação coletada durante a pesquisa. A partir de uma solução inicial, o SA prossegue em várias iterações. A cada iteração, um vizinho aleatório é gerado. Movimentos que melhoram a função de custo são sempre aceitos. Caso contrário, o vizinho é selecionado com uma dada probabilidade que depende da temperatura atual e da quantidade de ΔE de degradação da função objetivo. ΔE representa a diferença no valor objetivo (energia) entre a solução atual e a solução vizinha gerada. Conforme o algoritmo avança, a probabilidade de que tais movimentos sejam aceitos diminui. A Figura 3 descreve o pseudo-código do *Simulated Annealing*.

No trabalho de Pellicer e Pait (2020) foi utilizado o *Simulated Annealing* para realizar o *tunning* de hiperparâmetros dos modelos *LightGBM* e Rede Neural (*Multi-layer perceptron*). Utilizou-se outros métodos de otimização, sendo eles *Random Search*, *BarySearch*, TPE, CMA-ES, *Hyperband* e TPOT. O *Simulated Annealing* não apresentou os melhores resultados em nenhum dos casos testados, porém também não apresentou os piores resultados.

Figura 3 – Pseudo-código Simulated Annealing



Fonte: Russell e Norvig (2010).

De acordo com Russell e Norvig (2010) algoritmo genético (GA) é uma metaheurística em que estados sucessores são gerados combinando dois estados pais em vez de modificar um único estado. A analogia com a seleção natural é a mesma que na pesquisa de feixe estocástico, exceto que agora estamos lidando com reprodução sexual em vez de assexuada. Conforme podemos ver na Figura 4, o GA começa com um conjunto de estados gerados aleatoriamente, chamados de população. Cada estado é classificado pela função objetivo, que deve retornar valores mais altos para melhores estados. Dois pares são selecionados aleatoriamente para reprodução *crossover*, de acordo com a classificação feita através da função objetivo. Por fim, cada estado está sujeito a mutação aleatória com uma pequena probabilidade independente.

Os algoritmos genéticos combinam uma tendência ascendente com exploração aleatória e troca de informações entre pesquisas paralelas. Uma das principais vantagens dos algoritmos genéticos vem da operação de *crossover*. Intuitivamente, o *crossover* oferece ao algoritmo a capacidade de combinar grandes blocos de estados que evoluíram independentemente para realizar funções úteis, aumentando assim o nível de granularidade em que a pesquisa opera.

No trabalho de Francescomarino et al. (2018) foi utilizado algoritmo genético para otimização de hiperparâmetros dos modelos *Support Vector Machines* (SVM) e Rede Neural. Demonstrou-se que o uso de algoritmos genéticos permitem a redução de tempo de execução para otimização de hiperparâmetros e o sucesso no encontro de valores ótimos para os hiperparâmetros selecionados.

Figura 4 – Pseudo-código Algoritmo Genético

```
function GENETIC-ALGORITHM(population, FITNESS-FN) returns an individual
  inputs: population, a set of individuals
          FITNESS-FN, a function that measures the fitness of an individual

  repeat
      new_population ← empty set
      for i = 1 to SIZE(population) do
          x ← RANDOM-SELECTION(population, FITNESS-FN)
          y ← RANDOM-SELECTION(population, FITNESS-FN)
          child ← REPRODUCE(x, y)
          if (small random probability) then child ← MUTATE(child)
          add child to new_population
      population ← new_population
  until some individual is fit enough, or enough time has elapsed
  return the best individual in population, according to FITNESS-FN

function REPRODUCE(x, y) returns an individual
  inputs: x, y, parent individuals

  n ← LENGTH(x); c ← random number from 1 to n
  return APPEND(SUBSTRING(x, 1, c), SUBSTRING(y, c + 1, n))
```

Fonte: Russell e Norvig (2010).

O método do baricentro é um algoritmo recursivo usado para encontrar o centro de massa ou baricentro, sem utilizar derivação ou ter conhecimento prévio da função objetivo (PAIT, 2018). De acordo com Pellicer e Pait (2020), a fórmula do baricentro é uma ponderação dos resultados de $f(x)$ e os valores de x. O método introduz uma perturbação aleatória para realização da exploração do espaço de busca. A distribuição do espaço de busca pode ser escolhida aleatoriamente, porém o trabalho de Pait (2018) demonstra que o espaço de busca tem uma distribuição gaussiana. Nesse caso o baricentro pode ter direção que converge ao gradiente da função $f(x)$. Dessa forma, o método do baricentro pode

combinar características de gradiente descendente com variáveis aleatórias que introduzem um comportamento mais exploratório.

O algoritmo *BarySearch* se utiliza do método do baricentro para otimização dos parâmetros. Para que o baricentro consiga convergir mais rapidamente, é importante uma boa inicialização do algoritmo. A inicialização do algoritmo pode ser realizada por uma amostragem aleatória de alguns pontos no espaço de busca que costuma apresentar bons resultados ou podemos utilizar uma amostragem de pontos diversos (PELLICER; PAIT, 2020).

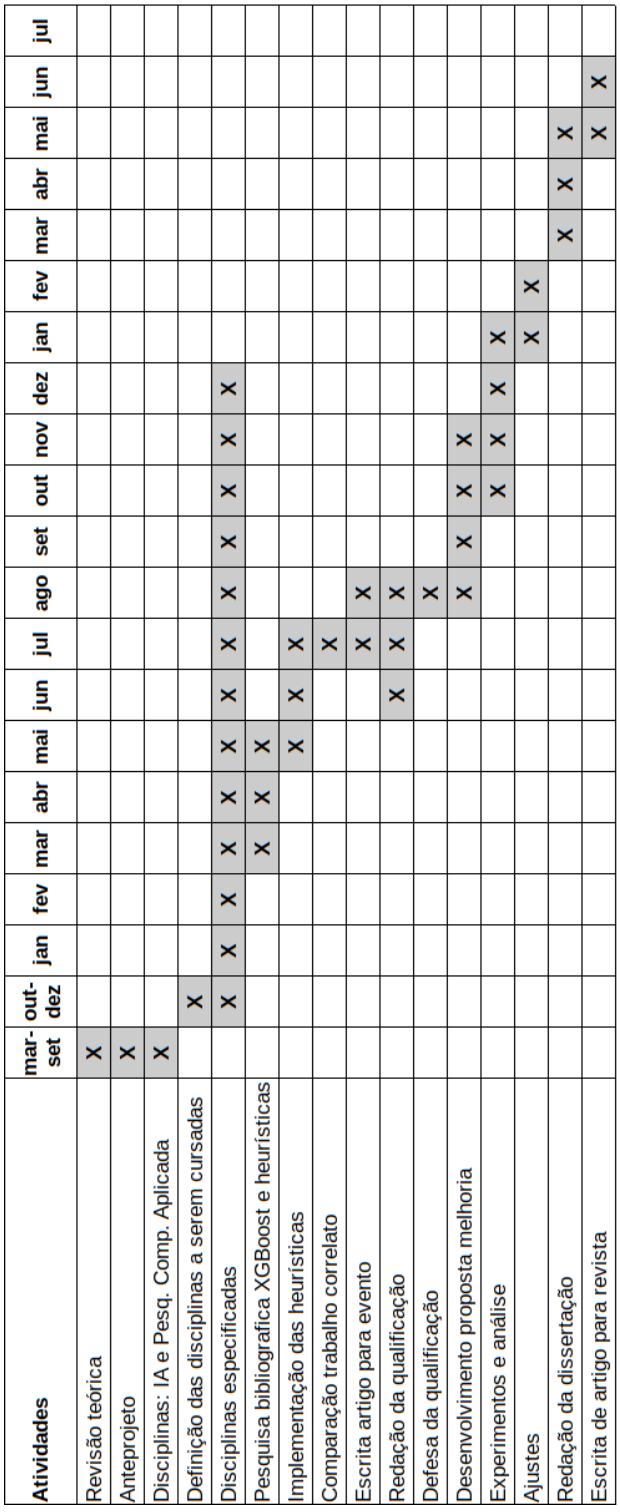
No trabalho de Pellicer e Pait (2020) foi utilizado o *BarySearch* para realizar o *tunning* de hiperparâmetros dos modelos *LightGBM* e Rede Neural (*Multi-layer perceptron*). Utilizou-se outros métodos de otimização, sendo eles *Random Search*, *Simulated Annealing*, TPE, CMA-ES, *Hyperband* e TPOT. Concluiu-se que em grande parte dos experimentos o *BarySearch* apresentou melhores resultados frente aos demais métodos.

O *Random Search* (busca aleatória) é outro método de busca no qual são testadas combinações de valores de hiperparâmetros amostradas aleatoriamente em um espaço de busca, seguindo uma distribuição uniforme, por exemplo. De acordo com Putatunda e Rama (2018) o espaço de configuração é definido usando um processo generativo para o desenho de amostras aleatórias e as atribuições de hiperparâmetros são extraídas deste processo e avaliadas. No trabalho de Bergstra e Bengio (2012) foi demonstrado que a busca aleatória tem vantagens sobre outros métodos, mas demonstra-se mais eficiente em espaços de grande dimensão.

O *Random Search* foi utilizado no trabalho de Sommer, Sarigiannis e Parnell (2019) no qual foi realizado o *tunning* dos seguintes hiperparâmetros do *XGBoost*: lambda, colsample_bytree, max_depth and learning_rate e num_boost_rounds. Demonstrou-se que o uso do algoritmo MeSH (*meta-learning sucessive halving*) - que utiliza *Meta-learning* (aprender a aprender) para melhorar o comportamento de outro algoritmo, o *Successive Halving* - apresentou resultados superiores ao uso dos algoritmos *Successive Halving* e *Random Search*.

3 CRONOGRAMA

Tabela 3 – Cronograma



Atividades	mar-set	out-dez	jan	fev	mar	abr	mai	jun	jul	ago	set	out	nov	dez	jan	fev	mar	abr	mai	jun	jul
Revisão teórica	X																				
Anteprojeto	X																				
Disciplinas: IA e Pesq. Comp. Aplicada	X																				
Definição das disciplinas a serem cursadas		X																			
Disciplinas especificadas		X	X	X	X	X	X	X	X	X	X	X	X	X							
Pesquisa bibliografica XGBoost e heurísticas					X	X	X														
Implementação das heurísticas							X	X	X												
Comparação trabalho correlato									X												
Escrita artigo para evento									X	X											
Redação da qualificação								X	X	X											
Defesa da qualificação										X											
Desenvolvimento proposta melhoria										X	X	X	X								
Experimentos e análise												X	X	X	X						
Ajustes															X	X					
Redação da dissertação																	X	X	X		
Escrita de artigo para revista																			X	X	

Fonte: Elaborado pelo Autor (2020).

REFERÊNCIAS

ALBIERO, Beatriz et al. Employing gradient boosting and anomaly detection for prediction of frauds in energy consumption. In: SBC. *Anais do XVI Encontro Nacional de Inteligência Artificial e Computacional*. [S.l.], 2019. p. 916–925.

BCB, Banco Central do Brasil. *Glossário Simplificado de Termos Financeiros*. 2013. 32 p. Disponível em: <https://www.bcb.gov.br/content/cidadaniafinanceira/documentos_cidadania/biblioteca/glossario_cidadania_financeira.pdf>. Acesso em: 07 agosto 2020.

BCB, Banco Central do Brasil. *Juros e Spread Bancário*. 2016. 10–11 p. Disponível em: <https://www.bcb.gov.br/content/cidadaniafinanceira/Documents/publicacoes/serie_pmf/FAQ%2001-Juros%20e%20Spread%20Banc%C3%A1rio.pdf>. Acesso em: 08 agosto 2020.

BCB, Banco Central do Brasil. *Relatório de Cidadania Financeira*. 2018. 111 p. Disponível em: <https://www.bcb.gov.br/nor/relcidfin/docs/Relatorio_Cidadania_Financeira.pdf>. Acesso em: 07 agosto 2020.

BCB, Banco Central do Brasil. *Composição do SFN*. 2020. Disponível em: <https://www.bcb.gov.br/estabilidadefinanceira/composicaosfn>. Acesso em: 07 agosto 2020.

BCB, Banco Central do Brasil. *Empréstimos, financiamento e arrendamento mercantil (leasing)*. 2020. Disponível em: <https://www.bcb.gov.br/acessoinformacao/perguntasfrequentes-respostas/faq_emprestimosefinanciamentos>. Acesso em: 07 agosto 2020.

BCB, Banco Central do Brasil. *O que são instituições não bancárias?* 2020. Disponível em: <https://www.bcb.gov.br/estabilidadefinanceira/instituicoesnaobancarias>. Acesso em: 07 agosto 2020.

BCB, Banco Central do Brasil. *O que é sociedade de crédito, financiamento e investimento?* 2020. Disponível em: <https://www.bcb.gov.br/estabilidadefinanceira/scfi>. Acesso em: 07 agosto 2020.

BCB, Banco Central do Brasil. *Perguntas frequentes Empréstimos consignados*. 2020. Disponível em: <https://www.bcb.gov.br/acessoinformacao/perguntasfrequentes-respostas/faq_emprestimosconsignados>. Acesso em: 07 agosto 2020.

BCB, Banco Central do Brasil. *Relatório de Economia Bancária, 2019*. 2020. 18 p. Disponível em: <https://www.bcb.gov.br/content/publicacoes/relatorioeconomiabancaria/REB_2019.pdf>. Acesso em: 08 junho 2020.

BCB, Banco Central do Brasil. *Sistema Financeiro Nacional*. 2020. Disponível em: <https://www.bcb.gov.br/estabilidadefinanceira/sfn>. Acesso em: 07 agosto 2020.

BERGSTRA, James; BENGIO, Yoshua. Random search for hyper-parameter optimization. *The Journal of Machine Learning Research*, JMLR. org, v. 13, n. 1, p. 281–305, 2012.

BRASIL. *LEI Nº 10.820, DE 17 DE DEZEMBRO DE 2003*. 2003. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/2003/L10.820.htm>. Acesso em: 08 agosto 2020.

BRASIL. *LEI Nº 13.172, DE 21 DE OUTUBRO DE 2015*. 2015. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2015/Lei/L13172.htm>. Acesso em: 07 agosto 2020.

COMMUNITY, Distributed (Deep) Machine Learning. *XGBoost Documentation*. 2020. Disponível em: <https://xgboost.readthedocs.io/en/latest/python/python_api.html>. Acesso em: 04 setembro 2020.

FEO, Thomas A; RESENDE, Mauricio GC. Greedy randomized adaptive search procedures. *Journal of global optimization*, Springer, v. 6, n. 2, p. 109–133, 1995.

FRANCESCOMARINO, Chiara Di et al. Genetic algorithms for hyperparameter optimization in predictive business process monitoring. *Information Systems*, Elsevier, v. 74, p. 67–83, 2018.

HASSAN, Doha et al. Comparative study of using data mining techniques for bank telemarketing data. In: IEEE. *2019 Sixth HCT Information Technology Trends (ITT)*. [S.l.], 2019. p. 177–181.

PAIT, Felipe M. The barycenter method for direct optimization. *arXiv preprint arXiv:1801.10533*, 2018.

PELLICER, Lucas Francisco Amaral Orosco; PAIT, Felipe Miguel. Barysearch: Algoritmo de tuning de modelos de machine learning com o metodo do baricentro. In: SBC. *Anais do XIV Brazilian e-Science Workshop*. [S.l.], 2020. p. 1–8.

PROBST, Philipp. *Hyperparameters, tuning and meta-learning for random forest and other machine learning algorithms*. 2019. Tese (Doutorado) — lmu, 2019.

PROBST, Philipp; BOULESTEIX, Anne-Laure; BISCHL, Bernd. Tunability: Importance of hyperparameters of machine learning algorithms. *J. Mach. Learn. Res.*, v. 20, n. 53, p. 1–32, 2019.

PUTATUNDA, Sayan; RAMA, Kiran. A comparative analysis of hyperopt as against other approaches for hyper-parameter optimization of xgboost. In: *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning*. [S.l.: s.n.], 2018. p. 6–10.

RESENDE, Mauricio GC; RIBEIRO, Celso C. Greedy randomized adaptive search procedures: advances and extensions. In: *Handbook of metaheuristics*. [S.l.]: Springer, 2019. p. 169–220.

RJ, SECRETARIA DE ESTADO DA CASA CIVIL E GOVERNANÇA. *Crédito Consignado. Tire suas dúvidas*. 2019. 7 p. Disponível em: <http://www.fazenda.rj.gov.br/sefaz/content/conn/UCMServer/uuid/dDocName%3AWCC38931153000>. Acesso em: 08 agosto 2020.

RUSSELL, Stuart J; NORVIG, Peter. *Artificial Intelligence-A Modern Approach, Third International Edition*. [S.l.]: Pearson Education London, 2010.

SOMMER, Johanna; SARIGIANNIS, Dimitrios; PARNELL, Thomas. Learning to tune xgboost with xgboost. *arXiv preprint arXiv:1909.07218*, 2019.

TALBI, El-Ghazali. *Metaheuristics: from design to implementation*. [S.l.]: John Wiley & Sons, 2009. v. 74.