JONAS ARRUDA NOVAES NETO

MODELO PREDITIVO DE CAPACIDADE DE PAGAMENTO PARA PROSPECÇÃO PF:

Atraindo e Fidelizando clientes no cenário de *Open Finance*

Brasília

2022

JONAS ARRUDA NOVAES NETO

MODELO PREDITIVO DE CAPACIDADE DE PAGAMENTO PARA PROSPECÇÃO PF: Atraindo e Fidelizando clientes no cenário de *Open Finance*

Dissertação apresentada ao Curso de Mestrado Profissional em Economia, Universidade de Brasília, como requisito parcial para a obtenção do título de Mestre em Economia

Universidade de Brasília - UnB

Faculdade de Administração Contabilidade e Economia - FACE

Departamento de Economia - ECO

Programa de Pós-Graduação

Orientador: Dr. Herbert Kimura

Coorientador: Dr. Daniel Oliveira Cajueiro

Brasília

2022

JONAS ARRUDA NOVAES NETO
MODELO PREDITIVO DE CAPACIDADE DE PAGAMENTO PARA PROSPECÇÃO PF: Atraindo e Fidelizando clientes no cenário de *Open Finance*/ JONAS ARRUDA NOVAES NETO. – Brasília, 2022-
128p. : il. (algumas color.) ; 30 cm.

Orientador: Dr. Herbert Kimura

Dissertação (Mestrado) – Universidade de Brasília - UnB
Faculdade de Administração Contabilidade e Economia - FACE
Departamento de Economia - ECO
Programa de Pós-Graduação, 2022.

1. Risco de Crédito. 2. Capacidade de Pagamento. 3. Aprendizado de Máquina. II. Universidade de Brasília. III. Faculdade de Administração, Contabilidade e Economia - FACE. IV. Departamento de Economia MODELO PREDITIVO DE CAPACIDADE DE PAGAMENTO PARA PROSPECÇÃO PF: Atraindo e Fidelizando clientes no cenário de *Open Finance*

JONAS ARRUDA NOVAES NETO

MODELO PREDITIVO DE CAPACIDADE DE PAGAMENTO PARA PROSPECÇÃO PF: Atraindo e Fidelizando clientes no cenário de *Open Finance*

Dissertação apresentada ao Curso de Mestrado Profissional em Economia, Universidade de Brasília, como requisito parcial para a obtenção do título de Mestre em Economia

Trabalho aprovado. Brasília, 15 de junho de 2022:

Dr. Herbert Kimura
Orientador

Dra. Marina Delmondes de Carvalho Rossi
Convidado

Dr. Pablo Jose Campos de Carvalho
Convidado

Brasília
2022

Agradecimentos

"Os homens agem sobre o mundo, modificam-no, e são modificados pelas consequências de suas ações"
(B. F. Skinner, 1957)

Acima de tudo, agradeço a Deus, nosso pai e Jesus Cristo, nosso senhor por toda força que recebi para superar todos os obstáculos que se colocaram no meu caminho durante essa jornada.

Agradeço a minha esposa e aos meus pais, por todo carinho, compreensão e suporte que recebi.

Dedico essa dissertação a minha filha, que com tão tenra idade teve que lidar com situações nas quais eu estava presente, porém mentalmente focado na realização deste trabalho, sem muito tempo para lhe dar a devida atenção.

Agradeço aos amigos e colegas, do trabalho e do mestrado, que me auxiliaram e me incentivaram a não desistir da jornada na reta final, momento em que o desafio enfrentado parecia intransponível.

Agradeço aos meus orientadores e ao monitor de python por todo o suporte recebido quando precisei.

Agradeço ao banco pelo fornecimento dos dados para estudo e a todos os empregados que participaram da sua confecção.

Agradeço à Universidade de Brasília, todos aqueles que foram meus professores e aos funcionários do mestrado profissionalizante.

Por fim, a todos aqueles que contribuíram de alguma forma para esta dissertação, deixo meu muito obrigado.

"Tudo na vida é gerenciamento de riscos, não sua eliminação"
(Walter Wriston, ex-presidente do Citicorp)

Resumo

As grandes instituições financeiras tradicionais estão direcionando suas estratégias para a expansão dos serviços financeiros no mundo digital, passando a concorrer nesse mercado com as *fintechs* e bancos digitais, empresas inovadoras, com menor custo operacional e que fazem uso intenso de tecnologia. A pandemia do coronavírus acelerou os hábitos digitais da população e aumentou a busca por créditos nos canais digitais, aumentando a concorrência entre as instituições, principalmente após a implantação do compartilhamento de dados através do *Open Finance*. Este trabalho teve como objetivo construir um modelo preditivo de capacidade de pagamento nos produtos comerciais para prospecção de pessoa física, com o uso de algoritmos supervisionados de aprendizagem de máquina para regressão. Para isso, utilizamos um conjunto de dados anonimizado proveniente de uma grande instituição financeira brasileira do segmento S1, contendo 350.953 registros e 61 variáveis, sendo que a variável dependente é o valor que a instituição deseja ofertar estrategicamente para esses clientes. Aplicamos quatro ferramentas para a seleção de variáveis (*Boruta, FeatureWiz, SelectKBest e RFE*) com diferentes parâmetros que resultaram em 39 variáveis únicas (Selecionadas), além de filtrar as 10 mais comuns entre todos os modelos (Top 10) e realizamos a otimização de hiperparâmetros com o *RandomizedSearchCV* e *Optuna* integrado com Neptuno para 18 estimadores (*Linear Regression, Ridge, Lasso, Elastic-Net, Huber Regressor, Passive Aggressive Regressor, Linear SVR, Nu SVR, K-Neighbors Regressor, PLS Regression, Decision Tree Regressor, Extra Trees Regressor, Random Florest Regressor, Gradient Boosting Regressor, Histogram Gradient Boosting Regressor, LightGBM Regressor, XGBoost Regressor e CatBoost Regressor*). Comparamos o resultado de 162 modelos construídos pela combinação do estimador (18 opções), seleção de variáveis (Todas variáveis, Selecionadas e Top 10) e otimização de hiperparâmetros (Sem otimização (*Default*), *RandomizedSearchCV* e *Optuna*), sendo escolhidos os melhores modelos na avaliação por diferentes métricas (*MAE, MSE, MAPE, RMSE, MedAE,* R^2*, Variância Explicada e Erro Residual Máximo*) aplicadas na base de Teste, *Out-of-Time* e *Out-of-Sample*. Os que apresentaram melhor resultado foram o *Gradiente Boosting Regressor Optuna* e o *LightGBM Regressor Optuna*, ambos modelos considerados "caixas preta" com complexa interpretação e explicação. Para extrair as regras do modelo de previsão, tornando-as interpretáveis utilizamos o LIME. Esse trabalho mostrou que diferentes técnicas de aprendizado de máquina, com excelente performance em relação aos modelos lineares tradicionais, podem ser aplicadas para a predição da capacidade de pagamento do cliente no ambiente bancário, altamente regulamentado.

Palavras-chave: Risco de Crédito; Capacidade de Pagamento; Aprendizado de Máquina.

Abstract

The large traditional financial institutions are directing their strategies towards the expansion of financial services in the digital world, starting to compete in this market with fintechs and digital banks, innovative companies, with lower operating costs and that make intense use of technology. The coronavirus pandemic accelerated the population's digital habits and increased the search for credit on digital channels, increasing competition between institutions, especially after the implementation of data sharing through Open Finance. This work aimed to build a predictive model of ability to pay (affordability) in commercial products for prospecting individuals, using supervised machine learning algorithms for regression. For this, we used an anonymized dataset from a large Brazilian financial institution in the S1 segment, containing 350,953 records and 61 variables, with the dependent variable being the value that the institution wants to offer strategically to these customers. We applied four tools for the selection of variables (Boruta, FeatureWiz, SelectKBest and RFE) with different parameters that resulted in 39 unique ("selected") variables, in addition to filtering the 10 most common among all models ("Top 10") and we performed hyperparameter optimization with RandomizedSearchCV and Optuna integrated with Neptune for 18 estimators (Linear Regression, Ridge, Lasso, Elastic-Net, Huber Regressor, Passive Aggressive Regressor, Linear SVR, Nu SVR, K-Neighbors Regressor, PLS Regression, Decision Tree Regressor, Extra Trees Regressor, Random Forest Regressor, Gradient Boosting Regressor, Histogram Gradient Boosting Regressor, LightGBM Regressor, XGBoost Regressor, and CatBoost Regressor). We compared the results of 162 models built by combining the estimator (18 options), variable selection (all variables, "selected" and "top 10") and hyperparameter optimization (without optimization, RandomizedSearchCV and Optuna), and the best models were chosen. in the evaluation by different metrics (MAE, MSE, MAPE, RMSE, MedAE, R^2, Explained Variance and Maximum Residual Error) applied to the Test, Out-of-Time and Out-of-Sample basis. The ones that presented the best results were the Gradient Boosting Regressor (Optuna) and the LightGBM Regressor (Optuna), both models considered "black boxes" with complex interpretation and explanation. To extract the rules from the prediction model, making them interpretable, we use LIME. This work showed that different machine learning techniques, with excellent performance compared to traditional linear models, can be applied to predict the customer's ability to pay in the highly regulated banking environment.

Keywords: Credit Risk; Payment Capacity (Affordability); Machine Learning.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

BCB	Banco Central do Brasil
CBR	CatBoost Regressor
CMN	Conselho Monetário Nacional
CRISP-DM	*CRoss Industry Standard Process for Data Mining*
CS	*Credit Scoring*
CV	Validação Cruzada
DTI	*Debt-to-income ratio*
DTR	Decision Tree Regressor
EDA	Análise Exploratória de Dados
ETR	Extra Trees Regressor
FEBRABAN	Federação Brasileira de Bancos
GBR	Gradiente Boosting Regressor
GIGO	Entra lixo, sai lixo
HGBR	Histogram Gradiente Boosting Regressor
IA	Inteligência Artificial
IF	Instituição Financeira
KNN	K-Neighbors Regressor
LGBMR	LightGBM Regressor
LIME	Explicações Agnósticas do Modelo Interpretável Local
LR	Regressão Logística
MAE	Erro Médio Absoluto
MAPE	Erro Médio Absoluto Percentual
MAX ERROR	Erro Residual Máximo

MEDAE	Erro Mediano Absoluto
ML	*Machine Learning*
MPCP	Modelo Preditivo de Capacidade de Pagamento
MSE	Erro Quadrático Médio
NN	Rede Neural Artificial
OOS	*Out-of-Sample*
OOT	*Out-of-Time*
PD	Probabilidade de Inadimplência
PLDFT	Prevenção a Lavagem de Dinheiro e Financiamento ao Terrorismo
PLS	PLS Regression
R^2	Coeficiente de Determinação
RFR	Random Forest Regressor
RMSE	Raiz do Erro Quadrático Médio
SFN	Sistema Financeiro Nacional
SVM	Máquina de vetores de suporte
VAR EXP	Variância Explicada
XGBOOST	Extreme Gradient Boosting

Lista de símbolos

λ	Letra grega minúscula Lambda
ν	Letra grega minúscula Nu
$\in$	Pertence

Sumário

1 Introdução

As grandes instituições financeiras tradicionais (Itaú, Banco do Brasil, Caixa, Bradesco e Santander) estão direcionando suas estratégias para a expansão dos serviços financeiros no mundo digital, passando a concorrer nesse mercado com as *fintechs* e bancos digitais, empresas inovadoras, com menor custo operacional e que fazem uso intenso de tecnologia.

Esse processo foi agravado pela expansão do coronavírus no país e o forte impacto econômico - social que a doença trouxe (campanhas de isolamento social; fechamento de empresas; desemprego em massa; redução no funcionamento das agências bancárias; fornecimento em massa de benefícios sociais para as empresas e pessoas; entre outras), que ocasionaram uma aceleração nos hábitos digitais da população.

De acordo com a Pesquisa FEBRABAN[1] de Tecnológica Bancária 2021 realizada pela DELOITTE (2021), em 2020 as transações bancárias realizadas através do *Mobile Banking* alcançaram 51% de todas as transações realizadas nos diferentes canais de atendimento[2] se tornando pela primeira vez o principal canal de relacionamento do cliente com o banco, inclusive para a solicitação e contratação de crédito, conforme tabela 1.

Tabela 1 – Canais de solicitação e contratação de Crédito

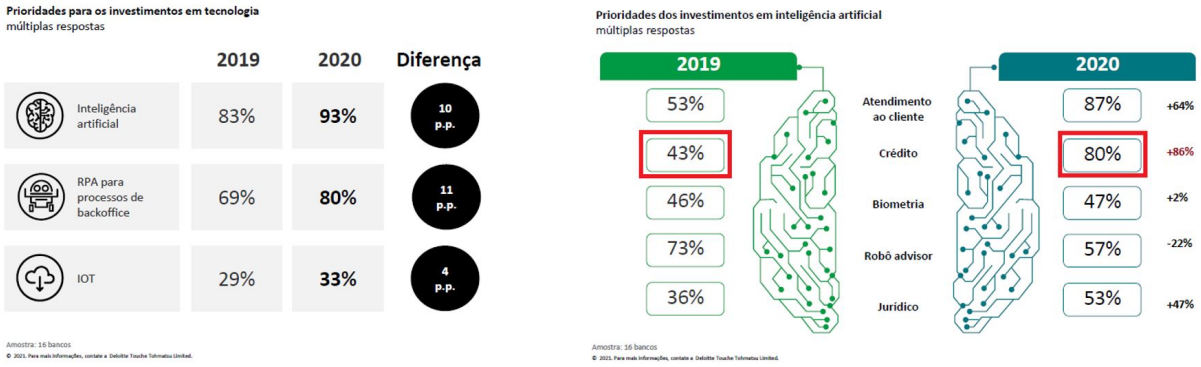
Canal	**2019***	**2020***	**Diferença**
Mobile banking	528,3	761,7	44%
Internet banking	165,6	147	-11%
Agências e PAB's	40,7	34,4	-15%
ATMs	62,2	44,1	-29%
Total	**796,8**	**987,2**	**24%**

** (em milhões de transações)*
Fonte: Adaptado de DELOITTE (2021)

Se por um lado os clientes optaram por utilizar o celular ao invés de outros canais, por outro lado, isso somente foi possível graças aos elevados investimentos por parte dos bancos em inteligência artificial aplicada ao atendimento ao cliente e na disponibilização de crédito, conforme figura 1. Ressalta-se que inteligência artificial, análise de dados e *machine learning* (ML) estão entre as principais tecnologias usadas pelas *fintechs* no segmento de crédito para entender quem é o cliente e oferecer o que ele precisa a taxas menores que o dos bancos tradicionais.

[1] Federação Brasileira de Bancos
[2] *Mobile Banking, Internet Banking*, Pontos de venda no comércio (POS), Autoatendimento (ATM), Correspondentes Bancários, Agências Bancárias, *Contact Centers*

Figura 1 – Prioridade de investimentos em Tecnologia Bancária



Fonte: DELOITTE (2021)

As plataformas digitais dos bancos tradicionais (Next do Bradesco; Superdigital do Santander; Iti do Itau; Caixa TEM da Caixa; e BB Digital do Banco do Brasil) fornecem crédito (crédito pessoal, cartão de crédito, cheque especial, financiamento de veículos, financiamento imobiliário) através de um processo mais simplificado quando comparado com o canal tradicional, se tornando uma excelente plataforma para a prospecção de clientes em um ambiente altamente concorrido, como o proporcionado pelo *Open Finance*[3].

No canal tradicional presencial, quando um cliente procura uma unidade bancária (agência ou correspondente), o atendente (empregado, terceirizado ou parceiro) faz uma entrevista com o cliente para entender a sua necessidade de crédito e solicita uma lista de documentos (ex: de identificação e de renda - ocupação), atuando como filtro para o banco: observa o estado emocional do cliente *(feeling)* e confere a documentação (analisa, solicita complementar e desconsidera as de característica estranha ou fraudulenta). Depois, insere os dados no sistema conforme o padrão e informa ao cliente o resultado obtido, aproveitando para oferecer outros produtos agregados ao crédito, como seguros, capitalização ou previdência.

Como vantagem, destacamos que é um processo com menor risco para o banco, em termos operacionais e de fraude; os dados são fidedignos aos documentos apresentados, fornecendo uma capacidade de pagamento adequada ao cliente; proporciona a venda de outros produtos não creditícios (seguro, capitalização e previdência) através da venda cruzada (*cross selling*). Como desvantagem, tem-se o maior custo envolvido, devido ao uso de mão de obra e espaço físico; menor volume de crédito diário a ser concedido, limitado

[3] Através do *Open Finance*, que é uma evolução do *Open Banking* realizada pela resolução conjunta BCB e CMN nº 4 de 24/03/2022, o cliente pode compartilhar de forma padronizada e segura suas informações bancárias entre diferentes instituições financeiras (bancos, corretoras, seguradoras, *fintechs*, cooperativas de crédito, corretoras de câmbio, distribuidoras, entre outras), tendo oportunidade de acessar ao mesmo tempo diferentes ofertas de crédito, optando pela mais vantajosa. Entre as informações que podem ser compartilhadas, destacamos: dados cadastrais; de conta; de aplicação financeira; de previdência; de seguro; de empréstimo; de capitalização; de câmbio; entre outros. Maiores informações estão disponíveis em https://openbankingbrasil.org.br (mantido pelo BCB) e https://noomis.febraban.org.br/temas/open-banking (mantido pela FEBRABAN).

a capacidade de atendimento da equipe; pré-análise subjetiva ao atendente, que pode rejeitar bons clientes sem chegar a inserir as informações para análise do sistema; pouco competitivo no cenário de *Open Finance* e para prospecção.

No canal digital o cliente pode ser atendido através de um *chatbot*, um robô que possui parâmetros já definidos, conversando com um atendente humano via *chat* somente em último caso, declarando as informações no *App* conforme o seu entendimento de cada pergunta, inclusive a renda e a ocupação, para solicitar o crédito desejado. Após análise pelo sistema, o cliente é avisado do resultado e pode contratar o crédito sem a necessidade de adquirir outros produtos não creditícios.

Como vantagem, podemos destacar que possui um menor custo envolvido, com o uso mínimo de mão de obra e economia de escala; maior volume de crédito diário a ser concedido, limitado a capacidade de processamento dos sistemas; menor tempo de resposta ao cliente e maior comodidade na contratação do crédito; maior competitividade em cenários altamente concorridos, como no *Open Finance* e para prospecção. Como desvantagem, destacamos o maior risco operacional, de fraude, e de PLDFT[4] para a IF, visto que as informações inseridas podem não ser fidedignas (erro de digitação ou má-fé), e ainda podem acabar sendo compartilhadas entre todo o sistema financeiro nacional através do *Open Finance*; uso de regras para evitar alimentar o sistema com ”lixo”, no conceito de *Garbage in, garbage out (GIGO)*, visto que é o próprio cliente quem preenche as informações sem o uso de qualquer padrão ou apresentação de documentos para uma segunda conferência, além daquelas de identificação obrigatórias por lei[5]; menor receita proveniente de produtos agregados (seguros, capitalização e previdência).

Através de parcerias com diferentes instituições, os bancos recebem informações de pessoas físicas para efetuarem estratégias de prospecção de novos clientes, concorrência que foi agravada com a expansão das plataformas digitais e depois do início do compartilhamento de dados no sistema financeiro aberto (*Open Finance*), já que o próprio cliente pode optar por compartilhar os seus dados que estão nos bancos em que possui relacionamento com os demais *players* financeiros. Diante desse novo cenário, este trabalho levanta o seguinte problema: como a instituição financeira pode mensurar adequadamente a capacidade de pagamento dos clientes e oferecer crédito através de estratégias de prospecção, como no ambiente competitivo de *Open Finance*, sem extrapolar seu apetite a riscos?

Uma hipótese para a solução desse problema vem com o desenvolvimento de um modelo específico para esse cenário que leve em consideração suas particularidades. Assim, o objetivo geral deste trabalho é construir um modelo preditivo para aferir a capacidade de pagamento nos produtos comerciais para prospecção da pessoa física por meio de

[4] Prevenção a Lavagem de Dinheiro e Financiamento ao Terrorismo

[5] Os documentos obrigatórios para apresentação são documento de identidade, CPF e comprovante de endereço

técnicas estatísticas e de *machine learning* (ML). Sendo o objetivo específico identificar entre as variáveis disponíveis para análise (demográfica, comportamental e de mercado), aquelas que melhor explicam a capacidade de pagamento do cliente (*Target*), identificando o modelo que apresenta a melhor performance para a base de dados entre as diferentes metodologias existentes.

2 Revisão da Literatura

2.1 Risco de Crédito

O Banco Central no artigo 21º da resolução 4.557/2017 (BCB, 2017), define o risco de crédito como:

- a possibilidade de ocorrência de perdas associadas ao não cumprimento pela contraparte (tomador de recursos, garantidor ou emissor de título) de suas obrigações pactuadas;
- a redução dos ganhos esperados em instrumento financeiro (ou renegociação que implique em concessão de vantagens) para a contraparte, o interveniente ou o instrumento mitigador, que sejam ocasionados pela deterioração da qualidade creditícia;
- custos de recuperação dos ativos problemáticos.

Segundo Anderson (2022), Joseph (2013), de forma simplificada, o conceito de risco de crédito está relacionado a possibilidade de o tomador de crédito não cumprir de forma parcial ou total com as condições acordadas com o concessor, também conhecido como risco de *default*. Esse descumprimento pode estar relacionado a diversos aspectos, como o risco da operação, da administração de crédito, da carteira de crédito, do cliente, entre outros, podendo ser minimizado ou evitado com a utilização de modelos de risco de crédito.

Os modelos de risco de crédito podem ser utilizados em todo o ciclo do crédito, como por exemplo, na concessão do crédito, para identificação da probabilidade de *default* e do valor a ser disponibilizado ao cliente; na manutenção periódica do risco da operação, com a extinção ou a expansão das linhas de crédito já fornecidas; na cobrança e na recuperação, para definição de estratégias de recebimento do crédito em atraso ou cessão do crédito em prejuízo (ANDERSON, 2022).

De acordo com Joseph (2013), a análise do risco de crédito, quando realizada de forma eficiente, fornece segurança à instituição financeira concessora, devido à mensuração adequada dos riscos do tomador, podendo ser realizada mediante a análise subjetiva e a análise objetiva.

A análise subjetiva utiliza modelos especialistas alimentados por critérios qualitativos para mensurar o risco do tomador e sua capacidade de pagamento, como a experiência do analista na avaliação daquele tipo de cliente, a qualidade da garantia oferecida e as

informações constantes na documentação analisada, como balanços e demonstrativos de resultados. Esse tipo de análise realiza-se de forma detalhada e individualizada, exigindo o envolvimento pessoal do analista em todo o processo e uso de suas habilidades para identificar todos os riscos envolvidos na operação, sendo utilizada principalmente para a análise de crédito de grandes corporações, conglomerados e entes públicos. As operações de crédito envolvem grandes somas financeiras (JOSEPH, 2013).

A análise objetiva utiliza modelos matemáticos, técnicas estatísticas e de *machine learning* para determinar a probabilidade de descumprimento do tomador ou o limite máximo de crédito, utilizando-se de regras claras e previamente estabelecidas. Nesse tipo de análise, as avaliações são realizadas de forma massificada por *softwares* especializados em análise de risco de crédito, de forma ágil e padronizada, podendo ser avaliados milhares de clientes ao mesmo tempo, de acordo com a capacidade de processamento dos computadores da IF. A função do analista está relacionada ao desenvolvimento de modelos e políticas de risco, além de sua respectiva manutenção. É utilizada para o fornecimento de créditos para pessoa física, microempreendedores individuais e empresas de micro, pequeno e médio porte (JOSEPH, 2013; ANDERSON, 2022).

Antes da utilização de modelos especialistas ou matemáticos pelas instituições financeiras, a mensuração do risco de crédito ocorria através dos 5Cs do Crédito (HAPSILA; ASTARINA, 2020; JOSEPH, 2013):

- Caráter, que analisa se o cliente estava honrando com os compromissos financeiros assumidos anteriormente (histórico de restrição cadastral, idoneidade, honestidade);
- Capital, que analisa o patrimônio e as reservas financeiras do cliente (aplicações financeiras, previdência, bens móveis e imóveis, dinheiro em caixa, endividamento, solidez, entre outros);
- Colateral, que analisa as garantias oferecidas na concessão do crédito (aplicações, imóveis, recebíveis, estoque, equipamentos, veículos, itens de valor, avalistas, fiadores, entre outros);
- Capacidade, que analisa os meios para pagar o empréstimo, as fontes de renda e o poder de compra do cliente (salário, aposentadoria, rendimentos, pensão, aluguéis, entre outros);
- Condições, que analisa a situação na qual o cliente está inserido, inclusive fatores internos e externos a ele (profissão, estabilidade da profissão e da renda, facilidade de recolocação no mercado de trabalho, taxa de desemprego, endividamento das famílias, entre outros).

Com o aumento no número de clientes Pessoa Física solicitando crédito, a metodologia qualitativa dos 5Cs do Crédito foi sendo substituída pelos modelos quantitativos, que tomam decisões de forma padronizada através de modelos empíricos, que utilizam diferentes técnicas para determinar a probabilidade de descumprimento do tomador e calcular sua capacidade de pagamento (JOSEPH, 2013).

2.1.1 *Credit Scoring*

Os bancos estimam a credibilidade do tomador de crédito em honrar com os pagamentos acordados, calculando a probabilidade de ocorrer atraso nos pagamentos (*Probability of default* - PD), que podem migrar para uma inadimplência (90 dias de atraso) e prejuízo (180 dias de atraso), conforme artigo 8º da resolução 2.682/1999 (BCB, 1999). Através dessa estimativa, calculada pelo modelo de Pontuação de Crédito (*Credit Scoring* - CS) a instituição pode agir rapidamente em caso de necessidade, assim que for identificada uma piora no risco do tomador.

Segundo Anderson (2022), existem diferentes modelos de CS para o cálculo do *score* de crédito (*scorecard*), destacando-se os modelos de Análise de Perfil (*Application Scoring* - AS), utilizados processo de tomada de decisão de concessão de produtos a um novo cliente e Análise de Comportamento (*Behaviour Scoring* - BS), que são utilizados na avaliação de risco das operações existentes.

Conforme Basiléia II a Probabilidade de Inadimplência (PD) deve ser associada a exposição financeira no momento do descumprimento (*Exposure at default* - EAD) e ao percentual de Perda financeira do valor que entrou em descumprimento (*Loss given default* - LGD) para o cálculo da perda esperada (*Expected Loss* - EL) (BCB, 2013; BIS, 2006; ANDERSON, 2022).

2.1.2 Capacidade de Pagamento

Após a mensuração do *score* de crédito por um modelo de pontuação de crédito, a instituição mensura o valor do crédito a ser oferecido ao cliente através do Modelo Preditivo de Capacidade de Pagamento (MPCP).

De acordo com Turkson, Baagyere e Wenya (2016), os bancos consideram vários fatores para mensurar o risco do cliente e calcular uma pontuação automatizada de crédito, porém a maioria dessas variáveis apresentam pouco efeito na previsão dos valores a serem concedidos aos clientes.

Isso ocorre porque a pontuação de crédito avalia a qualidade de crédito e propensão do cliente honrar com suas dívidas ou se tornar inadimplente (PD), independente dos motivos e sem se preocupar com a sustentabilidade do crédito no longo prazo. Porém, a propensão a reembolsar um empréstimo é separada da capacidade de reembolsar esse

empréstimo, pois clientes classificados como baixo risco podem não conseguir pagar o crédito tomado (BIJAK et al., 2015).

Segundo Bijak (2013), Anderson (2022), a avaliação da capacidade de pagamento (*affordability*) também se preocupa com a inadimplência, mas com o enfoque na capacidade do cliente em reembolsar o crédito assumido considerando seus outros compromissos financeiros e despesas regulares; na sustentabilidade desse crédito a longo prazo com a possibilidade de perda de fluxo financeiro (renda) ou o enfrentamento de situações financeiras inesperadas, como ocasionadas pela pandemia do coronavírus; na situação de superendividamento ao ocupar grande parte da sua renda com o pagamento do crédito ou tomar crédito em diferentes instituições financeiras.

No estabelecimento do valor do crédito a ser oferecido ao cliente deve ser levado em consideração as condições para que a dívida seja cumprida integralmente, sem que ocorra atrasos frequentes que obriguem o cliente a recorrer a repactuações (quando a dívida pode ser regularizada apenas pagando as parcelas em atraso) ou renegociações (quando a dívida deve ser liquidada por completo, sem a possibilidade de pagar apenas as parcelas em atraso). Essa preocupação com o crédito responsável evita que seja fornecido ao cliente um valor de crédito muito além da sua capacidade de arcar com o compromisso a longo prazo, mesmo que pontualmente o cliente esteja necessitando daquele valor (BIJAK, 2013).

De acordo com Bijak et al. (2015), Bijak (2013) para avaliar a capacidade de pagamento devem ser consideradas as informações cadastrais, como escolaridade, ocupação, região, idade, sexo, renda; as provenientes de birôs de crédito e de modelos de pontuação de crédito (*rating*); as relativas a endividamento bancário, gastos e despesas, que podem ser estimadas através de informações de pesquisas relacionadas ao custo de vida.

Em relação a renda, ressalta que o fato dos clientes exagerarem no valor da renda informada afeta todas as avaliações que se baseiam na renda, como a relação dívida e renda (*Debt-to-Income (DTI) Ratio*) e a relação serviço, renda e dívida (*Debt Service-To-Income (DSTI) Ratio*). Esse problema pode ser minimizado utilizando formas de validar essa renda, mesmo que parcialmente, como utilizar comprovantes, comparar com valores declarados em solicitações anteriores de crédito e comparar com os valores de renda presumida oferecidas pelos birôs de crédito (BIJAK et al., 2015).

Também pode ser considerado como renda mensal os rendimentos de poupanças e os recebíveis, visto serem uma capacidade de fluxo de caixa futuro, de acordo com Sharma (2009). A relação entre esses valores e as despesas mensais do cliente (hipoteca, aluguéis, outras dívidas, etc.) permite identificar a propensão a pagar, como ocorre na DTI.

Outras variáveis para a avaliação da capacidade de pagamento são: combinação de ciclo de vida do empréstimo, renda líquida, endividamento e depósitos; combinação de renda líquida, depósitos e despesas; taxa de pagamento de empréstimos em locais com

alto índice de desemprego (acima da média) e combinação de *score* de birôs de crédito, taxa de falência e fluxo de caixa (SHARMA, 2009).

O valor do crédito a ser oferecido ao cliente não deve ser calculado de forma isolada, considerando apenas o produto solicitado, mas levando em consideração todos os demais produtos de crédito que o cliente já possui, podendo chegar em até 25% da renda para empréstimos sem garantia, de forma a evitar o superendividamento. (BIJAK et al., 2015).

Para modelar a capacidade de pagamento Sharma (2009) utilizou Random Florest e Turkson, Baagyere e Wenya (2016) Regressão Linear.

2.1.3 Legislação

Os serviços financeiros disponibilizados pelos canais digitais possibilitam as instituições financeiras tradicionais expandirem suas carteiras de crédito, prospectarem novos clientes e fidelizarem os clientes já existentes, obtendo bons lucros devido aos menores custos associados, porém essa expansão tem que estar associada a uma seleção adequada dos clientes que vão receber o crédito e a mensuração adequada da capacidade de pagamento.

O Banco Central do Brasil (BCB), que por meio dos artigos 6º e 7º da resolução 4.557/2017, estabeleceu que a instituição financeira (IF) deve identificar, mensurar, controlar e mitigar o risco de crédito na qual esteja sujeita de forma relevante, principalmente quando envolver mudanças ou a criação de novos serviços, produtos e processos, com o objetivo de manter a sua exposição aos riscos dentro dos limites estabelecidos pela declaração de apetite ao risco (BCB, 2017).

Os artigos 7º, 9º e 23º da 4.557/2017 informam que a IF deve adotar ações para mitigar o risco de crédito e avaliar a sua eficácia, monitorando periodicamente o seu desempenho, inclusive com a comparação entre as perdas estimadas, as perdas observadas e os valores necessários para o seu provisionamento (BCB, 2017).

O artigo 2º da 2.682/1999 estabelece que a mensuração do risco deve ser efetuada com base em critérios verificáveis, consistentes, amparadas por informações internas e externas, como situação econômico-financeira (renda e patrimônio); grau de endividamento; pontualidade e atraso nos pagamentos (restrições cadastrais); limite de crédito; garantia; entre outras (BCB, 1999).

O BCB inclusive determina quais são os requisitos e critérios que devem ser considerados nos cálculos de provisão para devedores duvidosos e de perda esperada, conforme resolução 2.682/1999 e na circular 3.648/2013 (BCB, 1999; BCB, 2013), porém não estabelece nenhum critério para a mensuração da capacidade de pagamento.

Para os produtos consignados (crédito consignado e cartão consignado) existem legislações que definem a metodologia de cálculo e o valor do teto para comprometimento

de renda do cliente, por cada um dos entes federativos (União, Estados, Distrito Federal e Municípios).

A União, por exemplo, normatiza o valor do comprometimento máximo de salário/remuneração/soldo/benefício disponível a ser utilizado no crédito consignado e cartão consignado para os empregados regidos pela CLT[1], servidores públicos ativos e inativos federais, aposentados e pensionistas do INSS[2] através das leis: 10.820/2003, 8.213/1991 e 8.112/1990 (BRASIL, 2003; BRASIL, 1991; BRASIL, 1990).

Para esse público, até dezembro/2021 vigorou a lei 14.131/2021 que permitia consignar até 40% da renda, sendo 5% destinados exclusivamente para cartão consignado, como forma de combate a pandemia do coronavírus (BRASIL, 2021a). Atualmente os trabalhadores celetistas e os servidos públicos voltaram a ser regidos pela lei 13.172/2015 (BRASIL, 2015), com margem consignável de até 35% e os aposentados e pensionistas pela Medida Provisória 1.106/2022 com limite de até 40% (BRASIL, 2022).

Servidores estaduais, distritais e municipais, assim como militares das Forças Armadas (Exército, Marinha e Aeronáutica) possuem suas próprias legislações. O exército, por exemplo, através da portaria 124/2021 estabelece que o militar não poderá receber proventos menores que 30%, podendo consignar todo o valor restante (até 70%) após efetuar os descontos obrigatórios e as despesas médicos-hospitalares (EXÉRCITO, 2021). O estado do Paraná, por exemplo, através do decreto 9.220/2021, estabelece uma margem consignável de até 50%, sendo 10% exclusivos para cartão consignado (PARANÁ, 2021). A prefeitura de Belo Horizonte/MG, no decreto 15.537/2014 estabelece que poderá ser consignado até 43% da remuneração do servidor, após os descontos compulsórios, podendo utilizar 40% desse valor para o pagamento de empréstimo pessoal, imóvel residencial e cartão de crédito (BELO HORIZONTE, 2014).

Para habitação, a lei 8.692/1993 estabelece que o mutuário poderá destinar até 30% da renda bruta para o pagamento de encargos mensais (amortização e juros) no financiamento imobiliário. Durante a vigência do contrato, caso o valor dos encargos supere esse teto estabelecido, o cliente poderá solicitar a instituição financeira seu reenquadramento, em atendimento ao plano de comprometimento de renda (BRASIL, 1993).

A lei 14.181/2021 estabelece no artigo 6º que as instituições financeiras devem adotar práticas de crédito responsável, educação financeira, prevenção e tratamento de situações de superendividamento. Ao definir um valor de crédito a ser concedido ao cliente, deve-se levar em consideração o valor que seria mínimo para garantir a sua existência com dignidade (mínimo existencial) (BRASIL, 2021b), porém até o momento não está definido quanto será esse mínimo ou a sua forma de cálculo[3].

[1] Consolidação das Leis do Trabalho (CLT) regido pelo Decreto Lei nº 5.452/1943, que regulamenta as relações trabalhistas urbanas e rurais

[2] Instituto Nacional do Seguro Social (INSS)

[3] Estimativas da FEBRABAN informam que a adoção desse mínimo existencial poderá ocasionar uma

Nessa mesma lei, no artigo 54º-A informa que o cliente que não tiver condições de arcar com todos os compromissos financeiros assumidos (vincendas e vencidas) decorrentes de relação de consumo, incluindo operações de crédito, sem comprometer o seu mínimo existencial estará enquadrado na situação de superendividamento e passa a ser amparado por esta lei, possuindo assim condições favoráveis para renegociar suas dívidas (BRASIL, 2021b).

Diferentemente dos produtos consignados e da habitação, não existe amparo legal que determine qual o valor de crédito a ser disponibilizado para os diferentes produtos comerciais, como crédito pessoal e cartão de crédito, ou quais fatores devem ser considerados no seu cálculo. Dessa forma, cada instituição financeira possui liberdade para desenvolver sua própria metodologia de cálculo da capacidade de pagamento e disponibilizar ao cliente um limite de crédito que julgar adequado, de acordo com o seu apetito a risco, desde que não exponha o cliente a uma situação de superendividamento.

2.2 *Machine Learning*

Os algoritmos de aprendizado de máquina são ferramentas importantes para a construção de modelos preditivos, sendo utilizados para reconhecer, extrair padrões, fazer associações e construir modelos de aprendizagem a partir da observação de grande volume de dados, sendo que a cada iteração o modelo apresenta resultados mais precisos. Dependendo do tipo de problema e das características dos dados analisados, podem ser utilizadas diferentes abordagens: aprendizagem por reforço; aprendizagem supervisionada; aprendizagem não supervisionada e aprendizagem profunda *(Deep Leaning)* (GUPTA; SEHGAL, 2021).

2.2.1 Problema da "Caixa Preta"

A criação de modelos de risco de crédito através de técnicas de ML tornou-se um assunto amplamente investigado por pesquisadores e pelas instituições financeiras. Diversos modelos com excelente acurácia foram propostos, porém sua utilização prática pelas IF ainda é limitada, visto que a maioria dos algoritmos são considerados verdadeiras "caixas pretas", de difícil interpretação e explicação, como no caso da metodologia ensemble, que combina os resultados de múltiplos modelos para alcançar um maior poder preditivo (ALA'RAJ; ABBOD; MAJDALAWIEH, 2021; DUMITRESCU et al., 2022).

O Comitê de Supervisão Bancária da Basiléia e os órgãos reguladores estabelecem normas para que os critérios utilizados na avaliação de risco de crédito possam ser

redução de 22% no tamanho da carteira de crédito de pessoa física (no mínimo R$545 bilhões), provocando uma redução de 2 pontos percentuais no PIB do Brasil previsto para 2022 (FEBRABAN, 2021)

transparentes. A resolução BCB 2.682/99 em seu artigo 2º estabelece que a classificação da operação de risco deve ser efetuada com base em critérios consistentes e verificáveis e a resolução BCB 4.557/17 em seu artigo 23º estabelece que o risco de crédito deve ser estimado através de critérios consistentes e passíveis de verificação (BCB, 1999; BCB, 2017).

A interpretabilidade da pontuação de crédito é uma das preocupações dos reguladores (Comissão Europeia, Banco Central Francês, Banco da Inglaterra, entre outros) em relação a utilização dos modelos de ML.

De acordo com (BRACKE et al., 2019) do Banco da Inglaterra, o problema de explicabilidade da IA é a possibilidade de estudar as entradas e as saídas de um modelo de ML, mas não seu funcionamento interno. Mesmo nos casos em que esteja disponível para inspeção, seu tamanho e complexidade tornam difícil explicar sua operação aos humanos, assim como validar seus resultados, por esta razão esses modelos algumas vezes são considerados como verdadeiras ”caixa pretas”.

A Comissão Europeia também mostra preocupação com a complexidade e a opacidade que existem para os sistemas que utilizam ML, inclusive se apresentando como um dificultador para os reguladores verificarem eficazmente a conformidade de todo processo e o cumprimento das regras aplicáveis (EUROPEAN et al., 2020).

> ”O resultado do sistema de IA torna-se imediatamente efetivo, mas a intervenção humana é assegurada posteriormente (por exemplo, a rejeição de um pedido de cartão de crédito pode ser processada por um sistema de IA, mas a análise humana deve ser possível posteriormente) (EUROPEAN et al., 2020, página 23)”

Para Dupont, Fliche e Yang (2020) da ACPR[4] do Banco da França, os métodos de avaliação empírica devem ser considerados na fase de projeto do algoritmo de IA e incluídos no processo que garante a qualidade dos modelos resultantes. Caso seja necessário utilizar algum método explicativo, como em modelos ”caixas pretas”, estes podem ser implementados na fase de projeto ou operar em modelos previamente treinados, sendo que a escolha do método deve levar em consideração o tipo de algoritmo, o público-alvo das explicações e o risco associado ao processo. Os modelos que seriam diretamente interpretáveis sem a necessidade do uso de métodos explicativos são: Regressão (Logística e Linear), Árvore de Decisão, K-Vizinhos Próximos e Floresta Aleatória (com profundidade e volumes limitados). Para facilitar o trabalho da auditoria interna e da supervisão bancária no acompanhamento desses modelos, recomenda-se que os algoritmos e os modelos de dados sejam o mais modular e bem documentado, quanto possível.

De forma geral, os órgãos reguladores estabelecem normas para que os critérios utilizados na avaliação de risco de crédito possam ser transparentes. O artigo 2º da

[4] Autorité de Contrôle Prudentiel et de Résolution (ACPR)

resolução BCB 2.682/99 e artigo 23º da resolução BCB 4.557/17, estabelecem que o risco de crédito deve ser estimado através de critérios consistentes e passíveis de verificação (BCB, 1999; BCB, 2017).

Apesar de enfrentar problemas de transparência e de dificuldades para explicar as previsões, os algoritmos de ML estão ganhando importância devido a adoção de diferentes métodos para tornar suas regras de previsão interpretáveis e explicáveis às diferentes partes interessadas (IF, reguladores, clientes, magistrados, entre outros), sem fazer com que o modelo perca acurácia. De acordo com Dastile, Celik e Potsane (2020), esses métodos envolvem a extração de regras e racionalização (justificativas) do porquê uma determinada decisão foi tomada, podendo citar como exemplo: NeuroRule; Trepan; Local Interpretable Model Agnostic Explanations (LIME); Nefclass; entre outros.

2.2.2 Algoritmos para modelos supervisionados

A Regressão Logística (*Logistic Regression* - RL) é o algoritmo mais frequentemente utilizado em CS e a abordagem padrão utilizada pelas instituições financeiras, especialmente para fins regulatórios, devido à sua simplicidade, transparência, interpretabilidade intrínseca, estabilidade e robustez. Por isso, a maioria dos bancos nacionais e internacionais utiliza a RL para estimar a probabilidade de inadimplência para fins de provisão (Basiléia II) e para requisitos de capital (Basileia III), além de estimativas pontuais de perdas de crédito esperadas (DUMITRESCU et al., 2022). No Brasil, a resolução BCB 2.682/99 estabelece os critérios para cálculo de provisão e a resolução BCB 4.557/17 para os requisitos de capital.

Além de ser o modelo de pontuação de referência no setor de crédito, é considerada uma técnica estatística popular para resolver problemas de classificação, enquanto a regressão linear é utilizada para problemas de regressão. Segundo Dastile, Celik e Potsane (2020) uma limitação da RL é o pressuposto de linearidade entre as entradas e as chances de log, que nem sempre são válidas, visto que existem casos em que a relação entre as variáveis independentes e as probabilidades de log não é linear.

Para vencer essa limitação, os modeladores de risco de crédito introduzem efeitos não lineares na RL utilizando diferentes métodos, como tornar discretas as variáveis contínuas, fusão de categorias e identificação de efeitos não lineares com cruzamento de variáveis (DUMITRESCU et al., 2022).

A Máquina de vetores de suporte (*Support Vector Machine*) utiliza a ideia de um hiperplano para separar perfeitamente as classes em um espaço de recursos de alta dimensão, para efetuar a análise de dados e reconhecer padrões (DASTILE; CELIK; POTSANE, 2020). Devido aos resultados excepcionais, o algoritmo que pode ser utilizado tanto para regressão quanto classificação, é considerado um dos melhores algoritmos

para a classificação de padrões (ALA'RAJ; ABBOD; MAJDALAWIEH, 2021; TRIVEDI, 2020). Também pode ser utilizado como uma técnica de classificação não linear, sendo seu desempenho sensível ao algoritmo para resolver o problema de programação quadrática e aos parâmetros na aprendizagem (parâmetro kernel, erros toleráveis de classificação incorreta e parâmetro regularizado balanceando a margem de classificação) (DASTILE; CELIK; POTSANE, 2020).

A floresta aleatória (*Random Florest*) é um algoritmo que emprega muitas árvores de decisão para treinar diferentes partes do mesmo conjunto de dados com o objetivo de reduzir a variância, podendo ser utilizado para regressão e para classificação. Para Dumitrescu et al. (2022) e Trivedi (2020) essa técnica fornece um desempenho de classificação melhor do que os modelos de regressão logística padrão, sendo considerado um dos melhores classificadores de ML. O desempenho superior está relacionado as regras não lineares "se-então-senão" subjacentes às árvores de decisão.

O K-vizinhos mais próximos (*K-Nearest Neighbors*) é um algoritmo que pode ser utilizado para regressão e classificação. De acordo com Dastile, Celik e Potsane (2020), tem como objetivo identificar a classe majoritária a que pertence o conjunto de dados analisado, através da distância da classe com os seus k vizinhos mais próximos. O k determina a quantidade de vizinhos que vai ser analisada e a distância pode ser calculada por diferentes medidas estatísticas (Euclidiana, Manhattan, Cosseno, Hamming, Minkowski, etc.). Uma limitação para o seu uso em bases maiores é a exigência de um maior poder de processamento, visto que o algoritmo calcula a distância para cada registro de dados armazenado, requerendo mais cálculos quanto maior for a base (DASTILE; CELIK; POTSANE, 2020).

Os algoritmos de Gradientes de Impulso (*Gradient Boosting*) funcionam estimando vários modelos de forma consecutiva, com a atribuição de pesos às instâncias de dados. Em cada iteração particular, um novo modelo fraco e básico, sendo geralmente uma árvore de decisão rasa, é treinado em relação ao erro do conjunto completo aprendido até a última iteração. Dessa forma, o desenvolvimento do modelo posterior aborda os erros do modelo anterior, sendo por isso melhor do que ele. As instâncias que foram classificadas incorretamente pelo modelo anterior receberão pesos maiores (DASTILE; CELIK; POTSANE, 2020; ALA'RAJ; ABBOD; MAJDALAWIEH, 2021).

O Gradiente de Impulso Extremo (XGBOOST) se diferencia dos demais algoritmos de gradiente pela construção das árvores de decisão em paralelo (ao invés de construir em série), sendo muito utilizado devido ao seu desempenho e sua velocidade de processamento (DASTILE; CELIK; POTSANE, 2020).

A Rede Neural Artificial (*Artificial Neural Networks*) é um sistema de aprendizado de máquina que simula a rede neural biológica do cérebro humano, ou seja, a forma como o cérebro processa as informações através das conexões complexas entre os neurônios

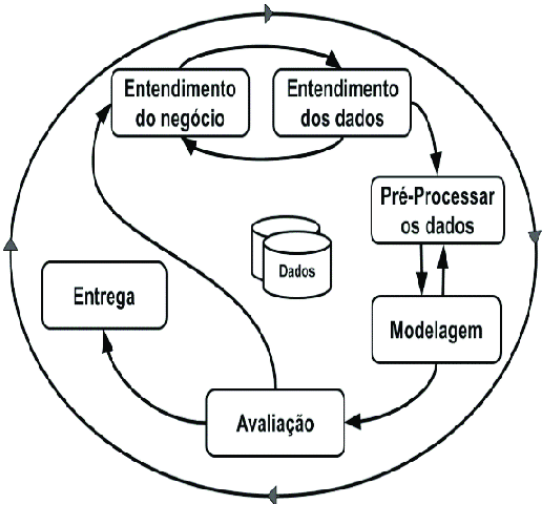
interconectados. Um dos modelos mais utilizados é o de multicamadas, em que são utilizadas no mínimo três camadas (uma de entrada, pelo menos uma oculta, e uma de saída), sendo que o treinamento do modelo envolve encontrar os pesos ótimos para cada camada via retro propagação (ALA'RAJ; ABBOD; MAJDALAWIEH, 2021; DASTILE; CELIK; POTSANE, 2020).

De acordo com Dastile, Celik e Potsane (2020), a rede neural apresenta melhor acurácia em comparação com outras técnicas, porém tem a limitação de não permitir a interpretabilidade dos resultados. Uma forma de resolver essa pendência, sem comprometer a sua acurácia, envolve a utilização de técnicas de extração das regras da rede neural, como por exemplo, Neurorule, Trepan e Nefclass. Como os modelos são flexíveis, podem ser utilizados para classificação e regressão.

2.3 *Cross Industry Standard Process for Data Mining*

A metodologia Processo Padrão Inter-Indústrias para a Mineração de Dados (CRISP-DM) foi desenvolvida em 1996 por um consórcio de empresas (Daimler Chryslerm, AG, SPSS, NCR, e OHRA) com o objetivo de padronizar o desenvolvimento de projetos em Ciências de Dados. Esse modelo de processo hierárquico, ágil e flexível, é composto por seis etapas em forma de ciclo, conforme figura 2 (WIRTH; HIPP, 2000).

Figura 2 – Etapas do CRISP-DM



Fonte: Adaptado de Wirth e Hipp (2000)

A primeira etapa é o "Entendimento de Negócio", que envolve a compreensão de qual é o problema enfrentado pela instituição financeira e de que forma esse problema pode ser resolvido (SCHRÖER; KRUSE; GÓMEZ, 2021). Entre as suas atividades destacam-se: planejamento, entendimento das necessidades e expectativas dos demandantes, levantamentos dos equipamentos, *softwares* e base de dados necessários ao desenvolvimento do

projeto. O entendimento incorreto do problema pode gerar uma solução que não atenda as necessidades dos demandantes, inviabilizando a entrega. Dessa forma, o ciclo prevê a revalidação desse entendimento nas etapas de ”entendimento dos dados” e de ”avaliação” (WIRTH; HIPP, 2000).

Após a identificação do problema, busca-se nas bases de dados disponíveis aquelas informações que contribuem para a sua resolução. De acordo com Wirth e Hipp (2000), Schröer, Kruse e Gómez (2021), na segunda etapa de ”Entendimento dos Dados”, as bases devem ser verificadas de forma cuidadosa para identificar: a qual informação aquele campo se refere; qual a sua descrição ou significado; quando ela foi originalmente gerada e quando foi armazenada no banco (hora, dia, mês, ano); qual a sua periodicidade; qual o sistema de origem; a informação é bruta (sem tratamento), tratada (categorizada, transformada, combinada) ou é uma constante; qual o formato do dado (numérico, texto, data); qual o universo de abrangência; qual o período disponível da informação; existem problemas nessa base de dados; entre outras. Após esse mapeamento, que envolve o conhecimento, avaliação e exploração dos dados disponíveis, de sua qualidade e volumetria, efetua-se a extração das informações importantes para a resolução do problema.

A terceira etapa de ”Pré-Processar os dados” envolve a preparação da base de dados que antecede o processo de modelagem. Os diferentes conjuntos de dados obtidos devem ser agregados com o estabelecimento de uma relação entre eles. Em seguida, deve-se efetuar a exploração e seleção de dados que serão utilizados no modelo, considerando questões como a qualidade dos dados, relevância da informação para a modelagem, restrições técnicas, tipos de dados e uso de *outliers*. Nessa nova base efetua-se a limpeza dos dados conforme a necessidade para evitar inconsistências, efetuando conversões de tipos de dados, combinação de variáveis, transformações de valores e enriquecimento de dados, deixando a base pronta para a modelagem (WIRTH; HIPP, 2000; SCHRÖER; KRUSE; GÓMEZ, 2021).

Segundo Wirth e Hipp (2000), com o conjunto de dados pronto inicia-se a quarta etapa de ”Modelagem”, em que serão selecionados as técnicas de *Data Mining* (regressão, árvore de decisão, gradiente boosting, etc.) mais apropriadas para resolver o problema definido na primeira etapa, efetuando caso necessário a calibragem de parâmetros dos algoritmos. Dependendo das técnicas escolhidas, pode ser necessário voltar para a terceira etapa e efetuar uma nova preparação da base de dados. Nesse momento devem ser selecionados diferentes modelos para que seja possível efetuar uma comparação entre os seus resultados (poder preditivo e performance computacional).

Na quinta etapa é realizada a ”Avaliação” de qual modelo desenvolvido apresentou o melhor resultado na resolução do problema proposto na primeira etapa. Essa solução é apresentada a todas as partes envolvidas na definição do problema para identificar se as expectativas foram atendidas. Caso não tenham sido atendidas, o projeto deve voltar para

a primeira etapa, sendo revisto e efetuado os ajustes necessários (SCHRÖER; KRUSE; GÓMEZ, 2021; WIRTH; HIPP, 2000).

De acordo com Schröer, Kruse e Gómez (2021), quando atende as expectativas inicia-se a sexta etapa de ”entrega” da solução, envolvendo a implantação de acordo com o cronograma estabelecido. O modelo a ser implantado precisa estar aderente as necessidades da organização; estar alinhado com as suas capacidades operacionais e tecnológicas e ser passível de interpretação, de acordo com as normas do regulador.

3 Análise Exploratória de Dados

Recebemos de uma grande instituição financeira brasileira do segmento S1, para construção do Modelo Preditivo de Capacidade de Pagamento (MPCP), uma base de dados anonimizada[1] com informações de clientes pessoa física que tiveram seu risco de crédito avaliado e contrataram pelo menos um produto comercial parcelado ou rotativo no período 10/2018 a 03/2020.

A base de dados está composta de 350.953 linhas e 61 colunas, sem linhas de registros duplicados, distribuídas em uma variável dependente (*Target*); uma coluna referente a data da informação (mês e ano no formato MMYYYY), uma variável sequencial (*TAG*) e 58 variáveis independentes referentes a informações demográficas, comportamentais e de mercado.

As informações demográficas são aquelas fornecidas pelo cliente ao banco no início do relacionamento, no momento da solicitação de crédito ou através da atualização cadastral. Nesse grupo temos 8 variáveis referentes a informações de Renda; Natureza da Ocupação; Idade; CEP; Escolaridade e Estado Civil.

As informações comportamentais são aquelas extraídas dos sistemas internos da IF[2], composto pelas 19 variáveis referentes a informações de Aplicação Financeira, Conta Poupança e Conta Corrente; recebimento de Salário; Relacionamento com a instituição; Empréstimos; *Rating* do cliente; Eventos Negativos relacionados a Conta, Empréstimo e Restrição Cadastral.

As informações de mercado são aquelas disponibilizadas por birôs de crédito e pelo sistema financeiro nacional, sendo que nesse grupo estão 31 variáveis, referentes a Empréstimos e Relacionamento em diferentes instituições; *Rating*, Renda e Capacidade de Pagamento; Eventos Negativos relacionados aos Empréstimos e às Restrições Cadastrais.

Para a análise dos dados e desenvolvimento do modelo de capacidade de pagamento utilizamos *framework* para Python Google Colab PRO+[3], a metodologia de projetos de ciências de dados CRISP-DM (capítulo 2.3) e o GitHub[4].

---

[1] Conforme informado pela IF: **"Em atendimento a Lei Geral de Proteção de Dados Pessoais nº 13.709/2018 e a Resolução CMN nº 4.893/2021, os valores constantes na base de dados foram deflacionados, dessensibilizados, mascarados e qualquer informação pessoal do cliente foi removida de forma a impossibilitar a sua identificação"**.
Ressaltamos que o uso do conjunto de dados fornecido é apenas para a pesquisa acadêmica e não representa situação real de negócios, tendo sido efetuado uma proteção de dados para evitar o risco de vazamento de qualquer informação recebida, por mais descaracterizada que esteja.

[2] Devido a implantação recente (agosto/2021), a IF não disponibilizou dados compartilhados pelo cliente via *Open Finance*.

[3] Google Colab: maiores informações em https://colab.research.google.com

[4] Os principais códigos em Python utilizados nesta dissertação estão disponíveis em:

Efetuamos a análise exploratória de dados (*Exploratory Data Analyses* - EDA) com o objetivo de identificar padrões gerais nas variáveis, detectar anomalias, eliminar inconsistências, identificar correlações, testar hipóteses, selecionar as variáveis que mais discriminam o *Target*, entre outras. Nesse processo, utilizamos a análise univariada, análise bivariada e análise multivariada.

Na análise univariada busca-se o conhecimento prévio das variáveis disponíveis, bem como sua distribuição e características, sendo efetuada uma análise independente de cada variável da base com o objetivo de identificar as que apresentam bom potencial de discriminação, identificar *outliers*, variáveis inconsistentes, variáveis com muitas informações ausentes *(missing, NaN ou NULL)* e problemas de base. As técnicas utilizadas nessa análise envolvem as medidas de tendência central (média, moda e mediana) e as medidas de dispersão (intervalo, variância, percentis e desvio padrão) (BRUCE; BRUCE; GEDECK, 2020).

A análise bivariada consiste em separar as variáveis em categorias de comportamento semelhante quanto ao risco e que fazem sentido para o negócio, com o objetivo de verificar a relação entre cada variável independente e a variável dependente para analisar seu potencial discriminador (MUKHIYA; AHMED, 2020); identificar comportamentos estranhos ou inesperados de uma variável; tratar os problemas de base que influenciam na modelagem (outliers, missing, etc).

A análise multivariada consiste em selecionar as variáveis que serão utilizadas para a construção do modelo, podendo ser utilizadas diferentes análises, como o uso de *dummies* para transformar em binária variáveis qualitativas nominais ou categóricas (estado civil, escolaridade, etc); a de matriz de correlação para identificar as preditoras que apresentam elevada correlação com outras preditoras; a multicolinariedade para tratar as preditoras que são correlacionadas com outras preditoras e com a variável resposta; entre outras (MUKHIYA; AHMED, 2020).

Para realizar a EDA utilizamos as bibliotecas do Python: Numpy (*Numerical Python*)[5] para realizar diversas funções matemáticas; Pandas[6] para análise e manipulação da base de dados; Scipy[7] para funções estatísticas; Scikit-learn[8] para análise preditiva dos dados e de *machine learning*; Statistics[9] para funções estatísticas; Matplotlib[10] para geração de gráficos; Seaborn[11] para gráficos estatísticos; e IPython[12] para uso de HTML.

https://github.com/janovaes/dissertacao_mestrado

[5] NumPy: maiores informações em https://numpy.org/doc/stable/

[6] Pandas: maiores informações em https://pandas.pydata.org/docs/

[7] Scipy: maiores informações em https://scipy.github.io/devdocs/index.html/

[8] Scikit-learn:maiores informações em https://scikit-learn.org/stable/user_guide.html

[9] Statistics: maiores informações em https://docs.python.org/3/library/statistics.html

[10] Matplotlib: maiores informações em https://matplotlib.org/

[11] Seaborn: maiores informações em https://seaborn.pydata.org/index.html

[12] IPython: maiores informações em https://ipython.readthedocs.io/en/stable/index.html

3.1 *Out-of-Time* e *Out-of-Sample*

Conforme informado pela IF, as safras das bases de dados abrangem o período de 10/2018 a 03/2020, informação que foi confirmada através da variável independente Ano_mês.

Com o objetivo de desenvolver modelos desafiadores segregamos parte da base original para realizar a validação fora do tempo (*Out-of-time* - OOT) e a validação fora da amostra (*Out-of-sample* - OOS), com o restante da base sendo utilizada para o desenvolvimento do modelo (Treino, Validação e Teste). A validação OOS tem como princípio verificar o comportamento do modelo na análise de novos dados, diferente daqueles utilizados no seu desenvolvimento, enquanto a validação OOT verifica a estabilidade do modelo na análise de dados de um período de tempo diferente daquele utilizado no processo de modelagem.

Da base originalmente recebida para o desenvolvimento do modelo, segregamos as safras 01/2020 a 03/2020 para construir a base OOT. O restante da base, contendo as safras de 10/2018 a 12/2019, foram divididas em 20% para a base OOS e os 80% a realização do treino, validação e teste, de forma aleatória e embaralhada através da função **Train Test Split** da biblioteca Scikit-Learn. A figura 3 resume as quebras efetuadas na base de dados.

3.2 Análise das variáveis Dependente e Independentes

Na base de "Treino-Validação-Teste" realizamos a análise exploratória e estatística da variável dependente e das variáveis independentes através do uso da ferramenta **Pandas Profiling**[13]; do uso de métricas pontuais (assimetria, curtose de Fisher, média harmônica, etc.) e do uso de análises gráficas (histograma, *boxplot*, máxima verossimilhança gaussiana, probabilidade quantil-quantil, entre outras).

A análise da variável TAG indicou a presença de registros únicos e sequenciais, equivalente ao número de linhas como uma forma de índice, sendo desta forma excluída da análise por não agregar valor ao desenvolvimento do modelo.

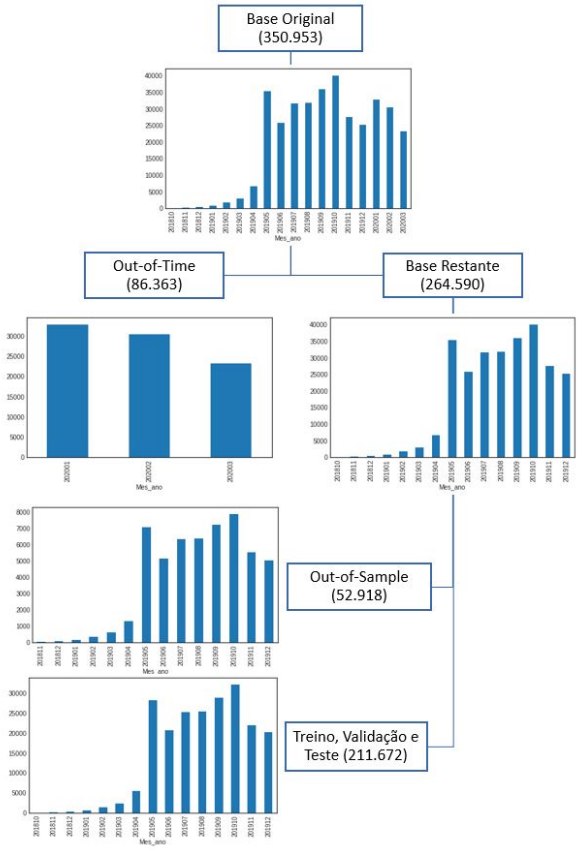
A variável Mês_ano, que informa o período de safras dos dados, foi excluída após a segregação da base, conforme capítulo 3.1 e figura 3.

A variável *Target*[14] é do tipo numérica contínua, com 22,8% de valores únicos e não

[13] Pandas_profiling.ProfileReport: maiores informações em https://pandas-profiling.ydata.ai/docs/master/index.html

[14] A variável *Target* é a Capacidade de Pagamento, que representa o valor para crédito que a instituição financeira deseja ofertar estrategicamente para atrair e fidelizar clientes no cenário de *Open Finance*, de acordo com o seu apetite a risco. Devido a questões de confidencialidade, não podem ser fornecidas maiores informações a respeito dessa variável.

Figura 3 – Divisão da Base de Dados em Treino, Validação, Teste, OOS e OOT)



Fonte: Elaborada pelo autor (2022)

apresenta qualquer valor negativo, zerado ou *missing*, sendo sua distribuição assimétrica positiva e leptocúrtica com uma cauda levemente longa a direita, conforme tabela 2 e gráfico 4.

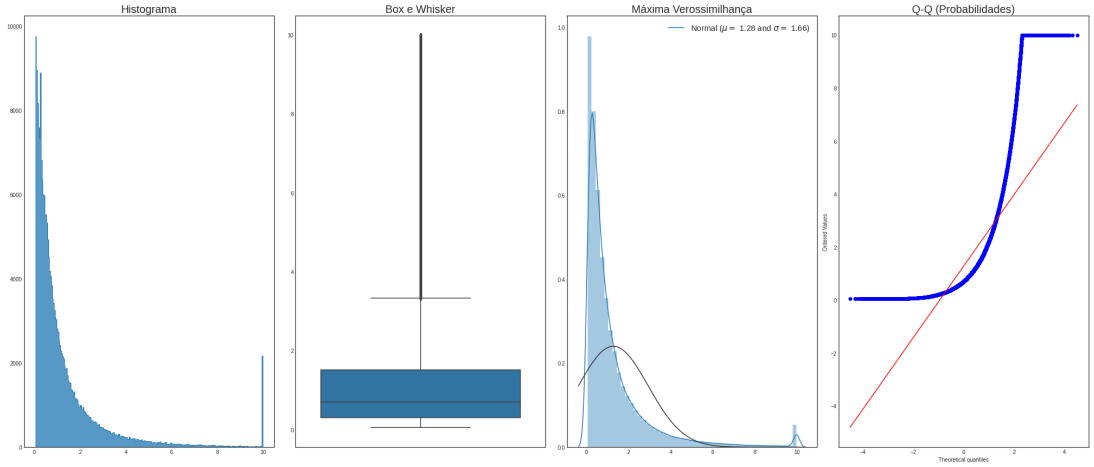
Tabela 2 – Análise Univariada da Variável Dependente

1º Perc./ Mínimo	0,0501	Média Aritmética	1,2761
5º Percentil	0,0945	Desvio Padrão	1,6611
25º Perc./ 1º Quartil	0,3056	Variância	2,7591
50º Perc./ Mediana	0,6975	Média Harmônica	0,3631
75º Perc./ 3º Quartil	1,5166	Coeficiente de variação	1,3016
95º Percentil	4,5285	Assimetria	2,9230
100º Perc./ Máximo	9,9814	Curtose	10,1733
Intervalo Interquartil	1,2110	Desvio Absoluto Mediano	0,4755
Amplitude	9,9313	Moda	9,9814

Fonte: Elaborada pelo autor (2022)

Das 58 variáveis independentes, 7 variáveis são do tipo categóricas; 16 são numéricas com resultados binários (0 ou 1), indicando tratar-se de situação possui/não possui; e os

Figura 4 – Distribuição da Variável Dependente



Fonte: Elaborada pelo autor (2022)

restantes das 35 variáveis são numéricas contínuas ou discretas. Não existem variáveis com valores negativos, infinitos ou constantes; os valores *missing* estão presentes em 7 variáveis, sendo apenas significativo para a "CP para mercado" com 22,80%; e os valores zerados estão presentes na maioria das variáveis numéricas, principalmente as binárias, com exceção das variáveis para as quais os valores zerados não fazem nenhum sentido, como por exemplo, idade e renda, conforme tabela 3.

Tabela 3 – Análise das Variáveis Independentes

Grupo	Tipo de dados	Quantidade	Valores Missing		Valores Zerados		
			Sim	Não	<25%	>25% e <75%	>75%
Demográfica	Numérica	5	3	2	5	0	0
	Categórica	3	0	3	3	0	0
Comportamental	Numérica	10	0	10	0	5	5
	Binária	7	0	7	0	1	6
	Categórica	2	1	1	2	0	0
Mercado	Numérica	20	2	18	11	4	5
	Binária	9	0	9	0	0	9
	Categórica	2	1	1	2	0	0

Fonte: Elaborada pelo autor (2022)

Para cada uma das variáveis independentes realizamos uma análise semelhante a efetuada para a variável *Target*, de forma gráfica[15] e não gráfica. Através das análises, observamos que a distribuição dos dados na maior parte das variáveis numéricas contínuas

[15] No apêndice B estão disponíveis os gráficos: 5 - Histograma; 6 - Histograma pela Máxima Verossimilhança Gaussiana; 7 - Probabilidade Quantil-Quantil e 8 - Diagrama de Caixa. No histograma, o eixo x e y para as variáveis binárias foi invertido em relação ao das variáveis contínuas para facilitar a visualização.

apresentam assimetria positiva, com curtose leptocúrtica de cauda longa, devido a elevada concentração dos valores próximo ao eixo y. Isso indica a necessidade de transformação dessas variáveis por logaritmo ou Box-Cox para torná-las mais simétrica.

Analisando a relação entre o *Target* e cada variável independente[16], observamos que as que apresentam maior correlação estão associadas aos Empréstimos no SFN; Renda do cliente; Renda e Capacidade de Pagamento para o mercado; e Empréstimos na instituição.

O mapa de calor da figura 12 (apêndice B) mostra a correlação entre as diferentes variáveis numéricas. No mapa, observa-se que a maioria das variáveis apresentam correlação próxima a zero e apenas algumas poucas variáveis apresentam uma forte correlação positiva, não tendo nenhuma variável que se destaque com a correlação negativa.

3.3 Tratamento de variáveis categóricas e campos sem informação

A transformação das variáveis categóricas em numérica e do tratamento dos campos sem informação (*missing*) é um procedimento necessário porque não são todos os algoritmos de *machine learning* que conseguem lidar essas questões.

Para a tratamento dos campos de *Rating* e *Rating* no SFN, em que a ordem das informações é importante, utilizamos o **Ordinal Encoder**[17] da biblioteca Category Encoders, que associa um valor número a ordem das informações fornecidas pelo dicionário. Os campos com valores faltantes receberam a menor pontuação.

Para o tratamento das demais variáveis categóricas utilizamos o **CatBoost Enconder**[18] da mesma biblioteca, que substitui os valores categóricos e faltantes por uma combinação do valor esperado da variável dependente em relação a própria variável e a toda a base de treinamento. Difere-se do *Target Encoder* por excluir a variável dependente quando calcula seu valor médio de referência, permitindo aos valores flutuar sem adicionar ruído.

Esse tratamento foi inicialmente aplicado a base "Treino-Validação-Teste" e depois replicado para as bases de OOT e OOS.

3.4 Seleção de variáveis

Após o tratamento das variáveis categóricas e dos campos sem informação (capítulo 3.3), executamos diferentes metodologias com o objetivo de reduzir a quantidade de

[16] No apêndice B estão disponíveis os gráficos: 9 - Estimativa de Densidade por Kernel e 10 - Dispersão; 11 - Dispersão das variáveis categóricas.

[17] Category_encoders.ordinal.OrdinalEncoder: maiores informações em https://contrib.scikit-learn.org/category_encoders/ordinal.html

[18] Category_encoders.cat_boost.CatBoostEncoder: maiores informações disponíveis em: https://contrib.scikit-learn.org/category_encoders/catboost.html

variáveis e eliminar variáveis redundantes e correlacionadas.

Para efetuar essa seleção utilizamos as ferramentas Boruta, FeatureWiz, SelectKBest e RFE na base "Treino-Validação-Teste".

- Na **Boruta**[19] utilizamos os estimadores Random Florest Regressor e Light GBM Regressor, sendo que cada um deles selecionou 17 variáveis.
- Na **FeatureWiz**[20] utilizamos as opções *Regular XGBoost, Regular XGBoost Groupby, Regular XGBoost Target, Dask XGBoost, Dask XGBoost Groupby e Dask XGBoost Target*. A quantidade de variáveis selecionadas foi respectivamente: 10, 13, 11, 12, 13 e 12.
- Na **SelectKBest**[21] utilizamos as funções de *score* F-Regression e Mutual Info Regression para selecionar as 15 variáveis com maior nota em cada função.
- Na **Eliminação Recursiva da Variáveis** (Recursive Feature Elimination - RFE)[22] utilizamos os estimadores Ridge, Linear SVR, Decision Tree Regressor e Gradient Boosting Regressor para manter apenas as 15 variáveis mais importantes para cada estimador.

A união de todas as variáveis selecionadas pelas diferentes técnicas resultou em 39 variáveis únicas e dessas filtramos as 10 mais comuns entre todos os modelos, conforme figura 13 do apêndice B.

Dessa forma, construímos os três grupos de variáveis independentes descritas a seguir para realizar o treinamento dos modelos de ML sem ocasionar viés, visto que cada algoritmo pode performar melhor com um conjunto de variáveis.

- Todas: composto por todas as 58 variáveis independentes da base de dados, sem nenhum filtro;
- Selecionadas: composto pelas 39 variáveis independentes selecionadas por pelo menos uma das quatro técnicas;
- Top 10: composto pelas 10 variáveis independentes mais comuns entre todas as quatro técnicas.

[19] Boruta.BorutaPy: maiores informações em https://github.com/scikit-learn-contrib/boruta_py
[20] Featurewiz.FeatureWiz: maiores informações em https://github.com/AutoViML/featurewiz
[21] Sklearn.feature_selection.SelectKBest: maiores informações em https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html
[22] Sklearn.feature_selection.RFE: maiores informações em https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html

3.5 Transformação de variáveis com Box-Cox

Construir um modelo com variáveis numéricas que se afastam de uma distribuição normal é possível, porém o modelo de regressão vai apresentar uma performance pior que o esperado devido a influência dos valores extremos (*outliers*).

Dessa forma, com o objetivo de efetuar o mínimo de manipulação na base "Treino-Validação-Teste" e possibilitar a reversão dos valores transformados aos originais optamos por realizar a transformação de potência de **Box-Cox (1+x)**[23] para eliminar assimetria, curtose e outliers. Essa função, que apresenta retorno equivalente ao Log(1+x) quando o λ é zero, possibilita retorno aos valores originais com o uso da sua **Inversa** [24].

Para encontrar o λ mais adequado para normalizar a variável, utilizamos o **Box-Cox Normality**[25]. Esse valor de λ foi utilizado na Boxcox1p para a base de "Treino-Validação-Teste" e replicados nas bases de OOT e OOS.

Com a base transformada, realizamos novas análises das variáveis para verificar o seu comportamento (histograma pela máxima verossimilhança gaussiana; probabilidade quantil-quantil e o mapa de calor), conforme gráficos (14, 15, 16, 17) disponíveis no apêndice B.

[23] Scipy.special.boxcox1p: maiores informações disponível em https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.boxcox1p.html
[24] Scipy.special.inv_boxcox: maiores informações em https://docs.scipy.org/doc/scipy/reference/generated/scipy.special.inv_boxcox.html
[25] Scipy.stats.boxcox_normmax: maiores informações em https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.boxcox_normmax.html

4 Metodologia

4.1 Estimadores Regressivos

Após a preparação da base de dados, realizamos a construção do Modelo Preditivo de Capacidade de Pagamento através de algoritmos supervisionados de aprendizagem de máquina para regressão.

Dentre os diferentes estimadores disponíveis para a regressão, que são algoritmos e modelos integrados de aprendizado de máquina, utilizados para prever uma variável dependente numérica contínua, selecionamos aqueles que apresentaram aderência ao conjunto de dados.

- Estimadores de Modelos Lineares:

    1. **Linear Regression**[1] (Regressão Linear) da biblioteca Scikit-Learn que utiliza o método dos mínimos quadrados ordinários para minimizar a soma residual de quadrados entre os valores previstos na aproximação linear e os observados no conjunto de dados.
    2. **Ridge Regression**[2] (Regressão do Cume, Crista ou Regularização de Tikhonov) da biblioteca Scikit-Learn que utiliza o método dos mínimos quadrados ordinários com a imposição da norma L2 como uma penalidade (regularização L2), que limita o tamanho do vetor de coeficiente.
    3. **Lasso Regression**[3] (Regressão do Laço) da biblioteca Scikit-Learn que utiliza o método dos mínimos quadrados ordinários com a imposição da norma L1 como uma penalidade (regularização L1), que impõe esparsidade entre os coeficientes.
    4. **Elastic-Net Regression**[4] (Regressão da Rede Elástica) da biblioteca Scikit-Learn que utiliza o método dos mínimos quadrados ordinários com a regularização por ambas as normas (L1 e L2).
    5. **Huber Regressor**[5] (Regressão Huber) da biblioteca Scikit-Learn que utiliza o método dos mínimos quadrados ordinários com aplicação de penalidade aos

[1] Sklearn.linear_model.LinearRegression: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html
[2] Sklearn.linear_model.Ridge: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html
[3] Sklearn.linear_model.Lasso: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html
[4] Sklearn.linear_model.ElasticNet: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.ElasticNet.html
[5] Sklearn.linear_model.HuberRegressor: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.HuberRegressor.html

valores extremos (*outliers*), atribuindo um peso menor com o uso da função de perda linear.

6. **Passive-Aggressive Regressor**[6] (Regressão Passivo-Agressiva) da biblioteca Scikit-Learn, um algoritmo de aprendizagem online que recebe os dados de forma sequencial, treinando o modelo passo a passo com o algoritmo passivo (que mantém o modelo do jeito que está em caso de previsão correta) e o algoritmo agressivo (que ajusta o modelo em caso de previsão incorreta).

- Estimadores de Máquinas de Vetor de Suporte:

1. **Linear SVR**[7] (Regressão Vetorial de Suporte Linear) da biblioteca Scikit-Learn, que utiliza a máquina vetorial de suporte escalável com kernel linear para ter mais flexibilidade na escolha de penalidades e de funções de perda (L1 e L2).
2. **Nu SVR**[8] (Regressão Vetorial de Suporte Nu) da biblioteca Scikit-Learn utiliza o parâmetro ν para controlar o número de vetores de suporte existentes em relação ao número total de amostras no conjunto de dados.

- Vizinhos mais próximos:

1. **K-Neighbors Regressor**[9] (Regressão de Vizinhos Mais Próximos - KNN) da biblioteca Scikit-Learn implementa o aprendizado com base nos vizinhos mais próximos de cada ponto de consulta.

- Decomposição cruzada:

1. **PLS Regression**[10] (Regressão por Quadrados Mínimos Parciais - PLS) da biblioteca Scikit-Learn utiliza a regressão linear regularizada para reduzir a quantidade de coeficientes a um conjunto menor de coeficientes não correlacionados, efetuando a regressão destes no lugar dos dados originais.

- Árvores de Decisão:

[6] Sklearn.linear_model.PassiveAggressiveRegressor: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.PassiveAggressiveRegressor.html
[7] Sklearn.svm.LinearSVR: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVR.html
[8] sklearn.svm.Nusvr: maiores informações em https://scikit-learn.org/stable/modules/generated/sklearn.svm.NuSVR.html
[9] sklearn.vizinhos. KNeighborsRegressor: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsRegressor.html
[10] Sklearn.cross_decomposition.PLSRegression: maiores informações disponíveis em https://scikit-learn.org/stable/modules/cross_decomposition.html

1. **Decision Tree Regressor**[11] (Regressão de Árvore de Decisão - DTR) da biblioteca Scikit-Learn é uma versão otimizada do algoritmo CART e muito semelhante ao C4.5, que não suporta variáveis categóricas.

- Métodos de Ensemble:

1. **Extra Trees Regressor**[12] (Regressão de Árvores Extremamente Aleatórias - ETR) da biblioteca Scikit-Learn utiliza toda a base de dados para construir um grande número de árvores de decisão, dividindo as árvore de maneira aleatória.
2. **Random Forest Regressor**[13] (Regressão de Floresta Aleatória - RFR) da biblioteca Scikit-Learn utiliza a reamostragem boostrap para construir um grande número de árvores de decisão, dividindo as árvores de maneira determinística.
3. **Gradient Boosting Regressor**[14] (Regressão de Gradiente de Impulso - GBR) da biblioteca Scikit-Learn é um algoritmo baseado em árvores de decisão que combina preditores com baixa precisão para reduzir o erro e desenvolver um modelo com forte precisão, após algumas iterações.
4. **Histogram Gradient Boosting Regressor**[15] (Regressão de Gradiente de Impulso de Histograma - HGBR) da biblioteca Scikit-Learn é um algoritmo de gradiente boosting baseado em histogramas, com suporte para valores ausentes.
5. **Light Gradient Boosting Machine (LigthGBM) Regressor**[16] (Regressão de Máquina de Gradiente de Impulso Leve - LGBMR) da Microsoft é um algoritmo de gradiente boosting baseado em histogramas, que utiliza a amostragem unilateral baseada em gradiente (GOSS) e agrupamento de recursos exclusivos (EFB), suporta GPU e variáveis categóricas.
6. **Extreme Gradiente Boosting (XGBoost) Regressor**[17] (Regressão de Gradiente de Impulso Extremo - XGBR) é um algoritmo de gradiente boosting pré-classificado e baseado em histograma, que utiliza Newton Boosting, penalização inteligente das árvores, encolhimento proporcional de nódulos de folhas e suporta GPU.

[11] Sklearn.tree.DecisãoTreeRegressor: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html
[12] Sklearn.ensemble.ExtraTreesRegressor: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html
[13] Sklearn.ensemble.RandomForestRegressor: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html
[14] Sklearn.ensemble.GradientBoostingRegressor: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingRegressor.html
[15] sklearn.ensemble.HistGradientBoostingRegressor: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.HistGradientBoostingRegressor.html
[16] Lightgbm.LGBMRegressor: maiores informações em https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMRegressor.html?highlight=regressor
[17] Xgboost.XGBRegressor: maiores informações em https://xgboost.readthedocs.io/en/latest/python/python_api.html?highlight=regressor

7. **Category Boosting (CatBoost) Regressor**[18] (Regressão de Impulso Categórico) da Yandex é um algoritmo de gradiente boosting que trabalha eficazmente com variáveis categóricas, utiliza arvores simétricas e suporta GPU.

4.2 Otimização de Hiperparâmetros

Após a seleção dos estimadores, buscou-se otimizar os seus hiperparâmetros. Os hiperparâmetros são diferentes valores de parâmetros que não são aprendidos pelos estimadores durante o processo de aprendizado de máquina, sendo utilizados para configurar o modelo (ex: profundidade de uma árvore) ou suas funções (ex: taxa de aprendizagem), tendo dessa forma um efeito significativo no desempenho do modelo.

A otimização de hiperparâmetros é o processo de encontrar a combinação certa de cada estimador em relação ao conjunto de dados com vistas a obter o melhor desempenho possível. Para encontrar esse valor ótimo, cada estimador é alimentado com um certo número de parâmetros e a faixa de valores possíveis para esse parâmetro, sendo verificado somente os valores explicitamente fornecidos.

Devido ao elevado tamanho da base de dados (211.672 linhas e 59 variáveis) e a quantidade de parâmetros buscados (até 8 com faixa de possíveis valores podendo chegar a 14 mil), a otimização dos algoritmos mais complexos estava sendo finalizada de forma inconclusiva (queda na conexão de internet, desconexão do ambiente do Colab, travamentos, entre outros) após dezenas de horas de processamento.

Para contornar esses dificultadores, geramos uma amostra estratificada proporcional de 5% da base de dados (10.584 linhas), sem reposição e com fixação da semente, através da função **Sample**[19] do Pandas para efetuar a busca pelos hiperparâmetros. E submetemos os mesmos parâmetros de cada estimador a diferentes metodologias.

O método mais tradicional é utilizar o **GridSearchCV**[20] da Scikit-Learn para realizar uma pesquisa exaustiva em uma grade cartesiana com validação cruzada, verificando todos os parâmetros fornecidos para o modelo a um custo computacional extremamente elevado, conhecido como maldição da dimensionalidade.

Um exemplo seria a busca do valor ótimo de um parâmetro em uma faixa de 14 mil valores possíveis, que quando combinada a outros parâmetros faz com que o número de combinações a serem testadas (iterações) cresça exponencialmente. Dessa forma, utilizamos esse método apenas para a Regressão Linear com Validação Cruzada (*Cross Validation* -

[18] Catboost.CatBoostRegressor: maiores informações em https://catboost.ai/en/docs/concepts/python-reference_catboostregressor

[19] Pandas.DataFrame.sample: maiores informações disponíveis em https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.sample.html

[20] Sklearn.model_selection.GridSearchCV: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

CV) igual a 5.

Outro método da biblioteca Scikit-Learn é o **RandomizedSearchCV**[21], que realiza as buscas pelos parâmetros ótimos através de uma pesquisa aleatória com validação cruzada. Possui um custo computacional menor do que o GridSearchCV porque permite estabelecer o número máximo de combinações aleatórias que serão testadas.

Uma desvantagem é que somente algumas combinações serão verificadas, podendo perder parâmetros importantes. Com exceção da regressão linear, utilizamos essa metodologia para os demais estimadores com 200 iterações e validação cruzada igual a 5, obtendo 1000 verificações por modelo.

Além desses dois, utilizamos o Optuna combinado com o Neptuno para buscar outros hiperparâmetros. O Optuna[22] é um framework de otimização automática de hiperparâmetros que utiliza o método Bayesiano, o Neptune[23] é plataforma que permite rastrear todas as métricas e resultados do projeto de aprendizado de máquina, sendo que a utilização integrada de ambos permite acompanhar e registrar a busca por cada hiperparâmetro.

Uma desvantagem desse método é que apenas algumas combinações são verificadas, a exemplo do que ocorre com o RandomizedSearchCV. Utilizamos o framework com todos os estimadores, limitando a otimização a 200 interações sem validação cruzada. A base foi dividida, aleatoriamente e de forma embaralhada, em treino (70%) e teste (30%) com função **Train Test Split**[24] do Scikit-Learn.

Os melhores hiperparâmetros encontrados para cada estimador estão descritos no apêndice D, na tabela 23 para GridSearchCV e RandomizedSearchCV; na tabela 24 para Optuna combinado com Neptuno. Para a Optuna, as figuras de 18 até 35 apresentam o processo de otimização de hiperparâmetros, com a análise de cada um dos parâmetros avaliados; as diferentes tentativas de combinação entre esses parâmetros; o momento em que a melhor combinação foi alcançada; e o peso que cada parâmetro apresenta para o processo de otimização.

4.3 Treinamento e avaliação do modelo

Com os hiperparâmetros otimizados, utilizamos as técnicas de ML para treinar o estimador para fazer previsão do conjunto de dados, validar o desempenho do modelo com o objetivo de otimizá-lo e testar o desempenho generalizado do modelo. Em seguida,

[21] Sklearn.model_selection.RandomizedSearchCV: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html
[22] Optuna: maiores informações disponíveis em https://optuna.org
[23] Neptune: maiores informações disponíveis em https://neptune.ai
[24] Sklearn.model_selection.train_test_split: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

verificamos o poder preditivo do modelo ao analisar um novo conjunto de dados: *Out-of-Time* (OOT) e *Out-of-Sample* (OOS), que foram construídos conforme descrito no capítulo 3.1.

Para isso, através do **Train Test Split** dividimos aleatoriamente e de forma embaralhada a base de "Treino-Validação-Teste", com 211.672 registros, em 70% para "Treino-Validação" e 30% para "Teste". Em seguida, dividimos a base "Treino-Validação" em 10 partes embaralhadas com a Validação Cruzada (CV) K-Fold[25]. O K-Fold CV vai dividir nosso conjunto "treino - validação" em 10 partes iguais, reservando 9 partes para o treinamento e 1 parte para validação, repetindo esse processo por 10 vezes.

A validação cruzada serve para avaliar a capacidade de generalização do modelo para um novo conjunto de dados, de forma a evitar a subestimação (*underfitting*) e a superestimação (*overfitting*), na busca da boa qualidade (*good fit*). A subestimação (*underfitting*) ocorre quando o modelo não consegue discriminar corretamente a variável dependente na base de desenvolvimento (alto viés) e deve ser descartado. A superestimação (*overfitting*) ocorre quando o modelo consegue discriminar a variável resposta com excelente precisão na base de desenvolvimento (alta variância), porém não consegue generalizar o resultado para novos dados na base de validação, devendo ser ajustado ou descartado (MÜLLER; GUIDO, 2016).

O ajuste de boa qualidade (good fit) ocorre quando os resultados obtidos na base de desenvolvimento e na de validação apresentam alguns desvios causados por erros de medição ou fatores aleatórios, estando dentro dos parâmetros desejados (SAMMUT; WEBB, 2017), por isso o modelo está adequado para continuar na análise e ser submetido as métricas de validação. Com o modelo treinado, avaliamos quão bem ele se ajustou aos conjuntos de dados (Teste, OOT e OOS), através de métricas que avaliam a sua qualidade e os valores de resíduo (erros), comparando os valores previstos e os valores observados, através das métricas descritas a seguir.

- Erro Médio Absoluto (*Mean Absolute Error* - MAE) representa a diferença média absoluta entre os valores previstos e os valores verdadeiros sem penalizar de forma desigual os pontos mais distantes, sendo dependente de escala. Essa medida estatística de erro é calculada através da equação 4.1 e no python utilizamos a função **sklearn.metrics.mean_absolute_error**[26] da biblioteca Scikit-Learn, sendo que quanto menor esse valor (mais próximo de zero), mais ajustado está o modelo.

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=0}^{n-1} |y_i - \hat{y}_i| \tag{4.1}$$

[25] Sklearn.model_selection.KFold: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.KFold.html

[26] Metrics.mean_absolute_error: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_error.html

onde: $\widehat{y_i}$ é o valor previsto, y_i é o valor verdadeiro, n é o total de amostras

- Erro Quadrático Médio (*Mean Squared Error* - MSE) representa a diferença média ao quadrado entre os valores previstos e os valores verdadeiros, penalizando os pontos mais distantes, principalmente os valores extremos (outliers). Essa medida estatística de erro é calculada através da equação 4.2 e no python utilizamos a função **sklearn.metrics.mean_squared_error**[27] da biblioteca Scikit-Learn, sendo que quanto mais próximo de zero, melhor é o ajuste do modelo ao conjunto de dados.

$$MSE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2 \tag{4.2}$$

onde: $\widehat{y_i}$ é o valor previsto, y_i é o valor verdadeiro, n é o total de amostras

- Erro Médio Absoluto Percentual (*Mean Absolute Percentage Error* - MAPE) representa o percentual da diferença média absoluta entre os valores previstos e os valores verdadeiros, calculados em relação aos valores verdadeiros. É uma medida estatística sensível a erros relativos, sem penalizar de forma desigual os pontos mais distantes, diferenciando-se do MAE por não ser dependente de escala, o que facilita a comparação de modelos pelo valor percentual do erro. A fórmula de cálculo está descrita em 4.3, sendo que retornos mais próximos de zero indicam melhor ajustamento do modelo ao valores reais. No python utilizamos a função **sklearn.metrics.mean_absolute_percentage_error** [28] da biblioteca Scikit-Learn.

$$MAPE(y, \hat{y}) = \frac{1}{n}\sum_{i=0}^{n-1}\frac{|y_i - \hat{y}_i|}{max(\in, |y_i|)} \tag{4.3}$$

onde: $\widehat{y_i}$ é o valor previsto, y_i é o valor verdadeiro, n é o total de amostras

- Raiz do Erro Quadrático Médio (*Root Mean Squared Error* - RMSE) é a taxa de erro calculada pela raiz quadrada do MSE, minizando o efeito quadrático e facilitando a interpretação ao ser medido nas mesmas unidades que a variável dependente, sendo por isso uma das métricas preferidas na avaliação do modelo. Utilizamos a equação 4.4 para essa medida estatística de erro e no python aplicamos a raiz quadrada no retorno da função **sklearn.metrics.mean_squared_error**[29] da biblioteca Scikit-Learn, sendo melhor o ajuste do modelo quanto menor for o valor de retorno.

[27] Metrics.mean_squared_error: maiores informações em https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

[28] Metrics.mean_absolute_percentage_error: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_absolute_percentage_error.html

[29] Metrics.mean_squared_error: maiores informações em https://scikit-learn.org/stable/modules/generated/sklearn.metrics.mean_squared_error.html

$$RMSE(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2} \tag{4.4}$$

onde: $\widehat{y_i}$ é o valor previsto, y_i é o valor verdadeiro, n é o total de amostras

- Erro Mediano Absoluto (*Median Absolute Error* - MedAE) representa a mediana de todas as diferenças absolutas entre os valores previstos e os valores verdadeiros, sendo robusto para outliers. Sua fórmula de cálculo está descrita em 4.5 e no python utilizamos a função da biblioteca Scikit-Learn **sklearn.metrics.median_absolute_error**[30]. Como as demais métricas de erro, quanto menor o valor, melhor.

$$MedAE(y, \hat{y}) = mediana(|y_i - \hat{y}_i|, ..., |y_n - \hat{y_n}|) \tag{4.5}$$

onde: $\widehat{y_i}$ é o valor previsto, y_i é o valor verdadeiro, n é o total de amostras

- Coeficiente de Determinação (R-Squared - R^2) representa a proporção da variância de dados (y) que é explicada pelas variáveis independentes do modelo, indicando o quão próximo os dados estão da linha de regressão ajustada. Essa medida estatística de qualidade do modelo, calculada através da equação 4.6, possui retorno entre 0 a 1, sendo que quanto mais próximo de 1, melhor. Para efetuar esse cálculo no python utilizamos a função **metrics.r2_score**[31] da biblioteca sciki-learn.

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i, \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i, \overline{y})^2} \tag{4.6}$$

onde: $\widehat{y_i}$ é o valor previsto, y_i é o valor verdadeiro, n é o total de amostras

- Variância Explicada (Explained Variance - Var Exp) calcula a proporção em que o modelo explica a dispersão dos dados sem considerar os deslocamentos sistemáticos na previsão. Acrescida a variância residual ou inexplicada, forma a variância total. Quando o resíduo da previsão tem média zero, o retorno da equação 4.7 é igual ao da equação 4.6 do R^2. Possui retorno entre 0 e 1, sendo que quanto mais próximo de 1, maior a qualidade do modelo. Para efetuar esse cálculo no python utilizamos a função **metrics.explained_variance_score**[32] da biblioteca sciki-learn.

$$Var\_Exp(y, \hat{y}) = 1 - \frac{Var(y - \hat{y})}{Var(y)} \tag{4.7}$$

[30] Metrics.median_absolute_error: maiores informações em https://scikit-learn.org/stable/modules/generated/sklearn.metrics.median_absolute_error.html

[31] Metrics.r2_score: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html

[32] Metrics.explained_variance_score: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.metrics.explained_variance_score.html

onde: $\widehat{y_i}$ é o valor previsto, y_i é o valor verdadeiro, Var é a variância

- Erro Residual Máximo (Max Error) indica o maior valor absoluto de erro existente entre o valor previsto e o verdadeiro. É calculado através da equação 4.8, sendo que quanto menor é esse valor, mais ajustado está o modelo, sendo zero quando o modelo se ajusta perfeitamente aos dados. No python utilizamos a função **metrics.max_error**[33] da biblioteca sciki-learn.

$$Max\_Error(y, \hat{y}) = max(|y_i - \hat{y}_i|) \tag{4.8}$$

onde: $\widehat{y_i}$ é o valor previsto; y_i é o valor verdadeiro

[33] Metrics.max_error: maiores informações disponíveis em https://scikit-learn.org/stable/modules/generated/sklearn.metrics.max_error.html

5 Resultados

5.1 Performance dos Estimadores Regressivos

Para comparar o resultado dos diferentes estimadores regressivos, identificando o impacto do uso de hiperparâmetros otimizados e da seleção de variáveis, realizamos o treinamento de nove modelos por cada estimador e avaliamos seu poder preditivo (Validação, Teste, OOT e OOS), com o objetivo de encontrar o melhor modelo por estimador.

Dessa forma, para cada estimador efetuamos a comparação dos seguintes modelos[1]:

- Modelo *Default*: modelo sem otimização de hiperparâmetros e composto por todas as 58 variáveis independentes da base de dados, sem nenhum filtro;
- Modelo Random: modelo com a otimização de hiperparâmetros realizado pelas metodologias GridSearchCV / RandomizedSearchCV e composto por todas as 58 variáveis independentes da base de dados, sem nenhum filtro;
- Modelo Optuna: modelo com a otimização de hiperparâmetros realizado pela metodologia Optuna (integrada ao Neptune) e composto por todas as 58 variáveis independentes da base de dados, sem nenhum filtro;
- Modelo *Default* Select: modelo sem otimização de hiperparâmetros e composto pelas 39 variáveis independentes selecionadas por pelo menos uma técnica (Boruta, FeatureWiz, KBest e RFE);
- Modelo Random Select: modelo com a otimização de hiperparâmetros realizado pelas metodologias GridSearchCV / RandomizedSearchCV e composto pelas 39 variáveis independentes selecionadas por pelo menos uma técnicas (Boruta, FeatureWiz, KBest e RFE);
- Modelo Optuna Select: modelo com a otimização de hiperparâmetros realizado pela metodologia Optuna (integrada ao Neptune) e composto pelas 39 variáveis independentes selecionadas por pelo menos uma técnicas (Boruta, FeatureWiz, KBest e RFE);
- Modelo *Default* Top 10: modelo sem otimização de hiperparâmetros e composto pelas 10 variáveis independentes mais selecionadas entre todas as técnicas (Boruta, FeatureWiz, KBest e RFE);

[1] O arquivo analítico com os resultados de cada um dos modelos por conjunto de dados (treino, validação, teste, OOT e OOS) está disponível em: https://github.com/janovaes/dissertacao_mestrado

- Modelo Random Top 10: modelo com a otimização de hiperparâmetros realizado pelas metodologias GridSearchCV / RandomizedSearchCV e composto pelas 10 variáveis independentes mais selecionadas entre as diferentes técnicas (Boruta, FeatureWiz, KBest e RFE);
- Modelo Optuna Top 10: modelo com a otimização de hiperparâmetros realizado pela metodologia Optuna (integrada ao Neptune) e composto pelas 10 variáveis independentes mais selecionadas entre as diferentes técnicas (Boruta, FeatureWiz, KBest e RFE);

Para comparar os resultados dos nove modelos (seis no caso da Linear Regression) desenvolvemos o sistema de pontuação descrito a seguir, em que um modelo poderá receber até 30 pontos por item analisado e 270 pontos no total, sendo considerado melhor aquele com a maior pontuação total, e em caso de empate, o com menor tempo de processamento.

- Runtime: Valoriza os modelos com menor tempo de execução (*runtime*). Pontuação: 1º: 30; 2º: 20 pontos; 3º a 9º: 0;
- MAE, MSE, MAPE, RMSE, MedAE e Max Error: Para cada conjunto de dados (Teste, OOT e OOS), valoriza os modelos que apresentam o menor erro (próximo de 0) avaliado em cada uma das diferentes métricas. Pontuação (por métrica e por conjunto de dados): 1º: 10; 2º: 5; 3º a 9º: 0. A pontuação máxima por métrica é de 30 pontos (considerando 10 no Teste, 10 no OOT e 10 no OOS).
- : R^2 e Var Exp: Para cada conjunto de dados (Teste, OOT e OOS), valoriza os modelos que apresentam melhor qualidade (próxima de 1) avaliado em cada uma das métricas. Pontuação (por métrica e por conjunto de dados): 1º: 10; 2º: 5; 3º a 9º: 0. A pontuação máxima por métrica também é de 30 pontos (vide item anterior).

Para Linear Regression não verificamos perda de performance na avaliação específica de um conjunto de dados, mas relacionado ao uso reduzido de variáveis, sendo os piores resultados para os modelos Top 10. O tempo total de execução foi de 16 segundos, distribuídos conforme tabela 25 disponível no apêndice D. O modelo que apresentou a melhor pontuação foi o Linear Regression *Default*, conforme tabela 4 e gráfico 36 das métricas de avaliação disponível no apêndice D.

No caso da Ridge Regression, como ocorreu com a Linear Regression, os piores resultados estão relacionados aos modelos Top 10. O tempo total de execução foi de 24 segundos, distribuídos conforme tabela 26 disponível no apêndice D. Os dois modelos que apresentaram melhor pontuação foram Ridge *Default* e Ridge Optuna, conforme tabela 5, sendo escolhido esse último devido ao menor tempo de exceção. O gráfico 37 com resultado por métrica de avaliação está disponível no apêndice D.

Tabela 4 – Análise da pontuação para os modelos de Linear Regression

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
LinearReg *Default*	-	25	25	10	25	15	25	30	10	165	MELHOR
LinearReg Grid/Optuna	-	20	20	30	20	30	20	15	-	155	-
LinearReg *Default* Select	-	-	-	-	-	-	-	-	5	5	-
LinearReg Grid/Optuna Select	-	-	-	5	-	-	-	-	-	5	-
LinearReg *Default* Top 10	20	-	-	-	-	-	-	-	10	30	-
LinearReg Grid/Optuna Top 10	30	-	-	-	-	-	-	-	20	50	-

Fonte: Elaborada pelo autor (2022)

Tabela 5 – Análise da pontuação para os modelos de Ridge Regression

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
Ridge *Default*	-	20	25	25	25	25	25	25	10	180	MELHOR
Ridge Random	-	20	20	20	20	10	20	20	-	130	-
Ridge Optuna	-	20	25	25	25	25	25	25	10	180	MELHOR
Ridge *Default* Select	-	-	-	-	-	-	-	-	-	-	-
Ridge Random Select	-	-	-	-	-	-	-	-	-	-	-
Ridge Optuna Select	-	-	-	-	-	-	-	-	-	-	-
Ridge *Default* Top 10	30	-	-	-	-	-	-	-	15	45	-
Ridge Random Top 10	-	-	-	-	-	-	-	-	20	20	-
Ridge Optuna Top 10	20	-	-	-	-	-	-	-	15	35	-

Fonte: Elaborada pelo autor (2022)

Na análise do Lasso Regression, os piores resultados estão relacionados aos modelos sem otimização, indicando a importância dos hiperparâmetros. O tempo total de execução foi de 12 minutos e 27 segundos, sendo maior o tempo para os modelos otimizados, conforme disponível na tabela 27 do apêndice D. O modelo que apresentou a melhor pontuação foi o Lasso Random, conforme tabela 6. O gráfico 38 com os resultados por métrica de avaliação está disponível no apêndice D.

Tabela 6 – Análise da pontuação para os modelos de Lasso Regression

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
Lasso *Default*	-	-	-	-	-	-	-	-	15	15	-
Lasso Random	-	30	30	30	30	20	30	30	-	200	MELHOR
Lasso Optuna	-	15	15	15	15	25	15	15	-	115	-
Lasso *Default* Select	-	-	-	-	-	-	-	-	15	15	-
Lasso Random Select	-	-	-	-	-	-	-	-	-	-	-
Lasso Optuna Select	-	-	-	-	-	-	-	-	-	-	-
Lasso *Default* Top 10	30	-	-	-	-	-	-	-	30	60	-
Lasso Random Top 10	-	-	-	-	-	-	-	-	-	-	-
Lasso Optuna Top 10	20	-	-	-	-	-	-	-	-	20	-

Fonte: Elaborada pelo autor (2022)

Para Elastic Net Regression, assim como no Lasso, os piores resultados foram para os modelos não otimizados. O tempo total de execução foi de 15 minutos e 34 segundos, sendo maior o tempo para os modelos otimizados (tabela 28 no apêndice D). O modelo que

apresentou a melhor pontuação foi o Elastic Net Optuna, conforme tabela 7. No apêndice D está disponível o gráfico 39 das métricas de avaliação.

Tabela 7 – Análise da pontuação para os modelos de Elastic Net Regression

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
ElasticNet *Default*	-	-	-	-	-	-	-	-	5	5	-
ElasticNet Random	-	15	15	30	15	30	15	15	-	135	-
ElasticNet Optuna	-	30	30	15	30	15	30	30	-	180	MELHOR
ElasticNet *Default* Select	20	-	-	-	-	-	-	-	10	30	-
ElasticNet Random Select	-	-	-	-	-	-	-	-	-	-	-
ElasticNet Optuna Select	-	-	-	-	-	-	-	-	-	-	-
ElasticNet *Default* Top 10	30	-	-	-	-	-	-	-	30	60	-
ElasticNet Random Top 10	-	-	-	-	-	-	-	-	-	-	-
ElasticNet Optuna Top 10	-	-	-	-	-	-	-	-	-	-	-

Fonte: Elaborada pelo autor (2022)

No caso do Huber Regressor, assim como no Lasso e na Elastic Net, a otimização melhorou a performance dos modelos. O tempo total de execução foi de 6 horas, 22 minutos e 21 segundos, com 98% desse tempo direcionado aos modelos otimizados (vide tabela 29 no apêndice D). O Huber Regressor Random foi o melhor modelo, de acordo com a tabela 8, estando disponível no apêndice D o gráfico 40 com os resultados por métrica.

Tabela 8 – Análise da pontuação para os modelos de Huber Regressor

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
Huber *Default*	-	-	-	-	-	-	-	-	-	-	-
Huber Random	-	15	30	10	30	15	30	30	5	165	MELHOR
Huber Optuna	-	30	15	30	15	30	15	15	-	150	-
Huber *Default* Select	20	-	-	-	-	-	-	-	-	20	-
Huber Random Select	-	-	-	-	-	-	-	-	10	10	-
Huber Optuna Select	-	-	-	5	-	-	-	-	-	5	-
Huber *Default* Top 10	30	-	-	-	-	-	-	-	-	30	-
Huber Random Top 10	-	-	-	-	-	-	-	-	15	15	-
Huber Optuna Top 10	-	-	-	-	-	-	-	-	15	15	-

Fonte: Elaborada pelo autor (2022)

No estimador Passive-Aggressive Regressor, diferentes do que ocorreu para outros estimadores até o momento, o uso das variáveis Select e Top 10 proporcionaram melhora na performance dos modelos, principalmente quando combinado com o Random. O tempo total de execução foi de 1 minuto e 22 segundos, distribuídos conforme tabela 30 no apêndice D. O modelo com melhor resultado foi Passive-Aggressive Regressor Random Select, conforme tabela 9. O gráfico 41 referente a cada métrica analisada está disponível no apêndice D.

Para o Linear SVR, assim como no Lasso, Elastic Net e Huber, os modelos sem otimização tiveram as piores performances, porém o tempo de execução foi bem menor (4,9% do total de 11 horas, 41 minutos e 15 segundos), conforme pode ser conferido na

Tabela 9 – Análise da pontuação para os modelos de Passive-Aggressive Regressor

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
Pas-Aggr *Default*	-	-	-	-	-	-	-	-	-	-	-
Pas-Aggr Random	-	-	-	-	-	-	-	5	-	5	-
Pas-Aggr Optuna	-	-	-	-	-	-	-	-	-	-	-
Pas-Aggr *Default* Select	-	-	-	10	-	-	-	-	10	20	-
Pas-Aggr Random Select	-	25	25	-	25	25	25	20	-	145	MELHOR
Pas-Aggr Optuna Select	-	-	-	-	-	-	-	-	-	-	-
Pas-Aggr *Default* Top 10	30	-	-	20	-	-	-	5	10	65	-
Pas-Aggr Random Top 10	-	20	20	5	20	20	20	15	5	125	-
Pas-Aggr Optuna Top 10	20	-	-	10	-	-	-	-	20	50	-

Fonte: Elaborada pelo autor (2022)

tabela 31 no apêndice D. A melhor performance foi do Linear SVR Random, que se destacou com uma pontuação bem maior que os demais (52% do total), conforme tabela 10. O gráfico 42 do resultado de cada métrica analisada esta disponível no apêndice D.

Tabela 10 – Análise da pontuação para os modelos de Linear SVR

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
LinerSVR *Default*	-	-	-	-	-	-	-	-	-	-	-
LinerSVR Random	-	30	30	30	30	30	30	30	5	215	MELHOR
LinerSVR Optuna	-	-	-	-	-	-	-	-	10	10	-
LinerSVR *Default* Select	20	-	-	-	-	-	-	-	-	20	-
LinerSVR Random Select	-	15	15	15	15	15	15	15	-	105	-
LinerSVR Optuna Select	-	-	-	-	-	-	-	-	-	-	-
LinerSVR *Default* Top 10	30	-	-	-	-	-	-	-	5	35	-
LinerSVR Random Top 10	-	-	-	-	-	-	-	-	15	15	-
LinerSVR Optuna Top 10	-	-	-	-	-	-	-	-	10	10	-

Fonte: Elaborada pelo autor (2022)

No caso do Nu SVR, o destaque foi para o Modelos Optuna. O tempo total de execução foi de apenas 6 horas e 50 segundos (vide tabela 32 no apêndice D) devido ao uso da extensão Intel(R) para Scikit-learn[2], que acelerou o treinamento dos modelos. O melhor resultado foi alcançado pelo Nu SVR Optuna Top 10, conforme tabela 11 e gráfico 43 das métricas de avaliação disponíveis no apêndice D.

Para K-Neighbors Regressor (KNN), quanto menor a quantidade de variáveis, melhor o resultado e menor o tempo de execução, indicando a importância da seleção de variáveis para esse estimador. A otimização de hiperparâmetros ajudou a buscar resultados melhores, porém onerou os modelos. O tempo de execução total foi de 17 horas, 10 minutos e 28 segundos (vide tabela 33 no apêndice D). O grande destaque desse estimador foi o K-Neighbors Regressor Optuna Top 10 com runtime de apenas 0,67% do total e 51% da pontuação total, conforme tabela 12 e gráfico 44 no apêndice D.

Para o estimador PLS Regression, o tempo total de execução dos modelos foi de 25 segundos, distribuídos conforme tabela 34 no apêndice D. O diferencial foi o uso

[2] Scikit-learn-intelex: maiores informações em https://github.com/intel/scikit-learn-intelex

Tabela 11 – Análise da pontuação para os modelos de Nu SVR

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
NuSVR *Default*	-	-	-	-	-	-	-	-	10	10	-
NuSVR Random	-	-	-	-	-	-	-	-	-	-	-
NuSVR Optuna	-	5	-	-	-	-	-	-	-	5	-
NuSVR *Default* Select	-	-	-	-	-	-	-	-	5	5	-
NuSVR Random Select	-	-	-	-	-	-	-	-	-	-	-
NuSVR Optuna Select	-	25	20	15	20	25	20	15	-	140	-
NuSVR *Default* Top 10	30	-	-	-	-	-	-	-	15	45	-
NuSVR Random Top 10	20	-	-	-	-	-	-	-	5	25	-
NuSVR Optuna Top 10	-	15	25	30	25	20	25	30	10	180	MELHOR

Fonte: Elaborada pelo autor (2022)

Tabela 12 – Análise da pontuação para os modelos de K-Neighbors Regressor (KNN)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
KNN *Default*	-	-	-	-	-	-	-	-	-	-	-
KNN Random	-	-	-	-	-	-	-	-	15	15	-
KNN Optuna	-	-	-	-	-	-	-	-	15	15	-
KNN *Default* Select	-	-	-	-	-	-	-	-	10	10	-
KNN Random Select	-	-	-	-	-	-	-	-	5	5	-
KNN Optuna Select	-	-	-	-	-	-	-	-	-	-	-
KNN *Default* Top 10	30	-	-	-	-	-	-	-	-	30	-
KNN Random Top 10	20	15	15	15	15	15	15	15	-	125	-
KNN Optuna Top 10	-	30	30	30	30	30	30	30	-	210	MELHOR

Fonte: Elaborada pelo autor (2022)

das variáveis Top 10 ou do otimizador Optuna, que quando combinados gerou o melhor modelo (PLS Regression Optuna Top 10), com vantagem frente aos demais, como pode ser verificado na tabela 13 e no gráfico 45 no apêndice D.

Tabela 13 – Análise da pontuação para os modelos de PLS Regression (PLS)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
PLS *Default*	-	-	-	-	-	-	-	-	20	20	-
PLS Random	-	-	-	-	-	-	-	-	20	20	-
PLS Optuna	-	-	-	5	-	-	-	-	-	5	-
PLS *Default* Select	-	-	-	-	-	-	-	-	-	-	-
PLS Random Select	-	-	-	-	-	-	-	-	-	-	-
PLS Optuna Select	-	10	15	-	15	10	15	15	-	80	-
PLS *Default* Top 10	30	5	-	10	-	5	-	-	10	60	-
PLS Random Top 10	20	5	-	10	-	5	-	-	10	50	-
PLS Optuna Top 10	-	30	30	30	30	30	30	30	-	210	MELHOR

Fonte: Elaborada pelo autor (2022)

Na análise da Decision Tree Regressor (DTR), os piores resultados ocorreram para os modelos sem otimização, mostrando a importância da busca pelos hiperparâmetros no caso deste estimador. Os modelos foram executados em 5 minutos e 55 segundos (vide tabela 35 no apêndice D), sendo que os modelos Top 10 tiveram o processamento mais

rápido. Os modelos otimizados pelo Optuna tiveram os melhores resultados, se destacando o Decision Tree Regressor Optuna, vide tabela 14 e gráfico 46 no apêndice D.

Tabela 14 – Análise da pontuação para os modelos de Decision Tree Regressor (DTR)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
DTR *Default*	-	-	-	-	-	-	-	-	5	5	-
DTR Random	-	-	-	30	-	25	-	-	5	60	-
DTR Optuna	-	30	20	15	20	5	20	25	5	140	MELHOR
DTR *Default* Select	-	-	-	-	-	-	-	-	-	-	-
DTR Random Select	-	-	-	-	-	15	-	-	5	20	-
DTR Optuna Select	-	15	20	-	20	-	20	15	5	95	-
DTR *Default* Top 10	-	-	-	-	-	-	-	-	-	-	-
DTR Random Top 10	20	-	-	-	-	-	-	-	10	30	-
DTR Optuna Top 10	30	-	5	-	5	-	5	5	20	70	-

Fonte: Elaborada pelo autor (2022)

No caso do Extra Trees Regressor (ETR), os modelos Top 10 apresentaram os piores resultados. O tempo total de execução foi de 20 horas, 13 minutos e 14 segundos, sendo 81,2% referente aos modelos Optuna (vide tabela 36 no apêndice D). O modelo com melhor resultado foi Extra Trees Regressor Random, conforme tabela 15 e gráfico 47 referentes a cada métrica analisada disponíveis no apêndice D.

Tabela 15 – Análise da pontuação para os modelos de Extra Tree Regressor (ETR)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
ETR *Default*	-	-	-	-	-	-	-	-	10	10	-
ETR Random	-	25	25	25	25	25	25	25	-	175	MELHOR
ETR Optuna	-	15	15	20	15	15	15	15	5	115	-
ETR *Default* Select	-	-	-	-	-	-	-	-	5	5	-
ETR Random Select	-	5	5	-	5	5	5	5	-	30	-
ETR Optuna Select	-	-	-	-	-	-	-	-	-	-	-
ETR *Default* Top 10	30	-	-	-	-	-	-	-	25	55	-
ETR Random Top 10	20	-	-	-	-	-	-	-	-	20	-
ETR Optuna Top 10	-	-	-	-	-	-	-	-	-	-	-

Fonte: Elaborada pelo autor (2022)

No Random Forest Regressor (RFR), as piores performances foram dos modelos Top 10. O tempo total de execução foi de apenas 11 horas, 4 minutos e 42 segundos graças ao treinamento por GPU proporcionado pela biblioteca Rapids[3] no lugar da Scikit-Learn, que não oferece suporte para GPU[4]. O maior tempo de execução ocorreu para os modelos Optuna (88,5%), conforme tabela 37 no apêndice D, mas esse tempo gasto em treinamento ocasionou melhor performance, com o modelo Random Forest Regressor Optuna obtendo melhor resultado, conforme tabela 16 e gráfico 48 no apêndice D.

[3] Cuml.ensemble.RandomForestRegressor: maiores informações em https://docs.rapids.ai/api/cuml/stable/api.html#random-forest

[4] https://scikit-learn.org/stable/faq.html#why-is-there-no-support-for-deep-or-reinforcement-learning-will-there-be-support-for-deep-or-reinforcement-learning-in-scikit-learn

Tabela 16 – Análise da pontuação para os modelos de Random Forest Regressor (RFR)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
RFR *Default*	20	-	-	-	-	-	-	-	5	25	-
RFR Random	-	20	15	30	15	25	15	15	-	135	-
RFR Optuna	-	25	30	15	30	20	30	30	5	185	MELHOR
RFR *Default* Select	-	-	-	-	-	-	-	-	10	10	-
RFR Random Select	-	-	-	-	-	-	-	-	-	-	-
RFR Optuna Select	-	-	-	-	-	-	-	-	10	10	-
RFR *Default* Top 10	30	-	-	-	-	-	-	-	5	35	-
RFR Random Top 10	-	-	-	-	-	-	-	-	10	10	-
RFR Optuna Top 10	-	-	-	-	-	-	-	-	-	-	-

Fonte: Elaborada pelo autor (2022)

Na análise do Gradiente Boosting Regressor (GBR), os piores resultados foram para os modelos não otimizados e para os Top 10. O tempo de execução total foi de 11 horas, 40 minutos e 12 segundos, vide tabela 38 no apêndice D, com os modelos não otimizados ocupando apenas 7,90% do tempo total. O melhor modelo foi o Gradiente Boosting Regressor Optuna, conforme tabela 17. O gráfico 49 referente a cada métrica analisada disponível no apêndice D.

Tabela 17 – Análise da pontuação para os modelos de Gradiente Boosting Regressor (GBR)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
GBR *Default*	-	-	-	-	-	-	-	-	25	25	-
GBR Random	-	15	25	20	25	5	25	25	-	140	-
GBR Optuna	-	30	20	25	20	30	20	20	-	165	MELHOR
GBR *Default* Select	20	-	-	-	-	-	-	-	20	40	-
GBR Random Select	-	-	-	-	-	5	-	-	-	5	-
GBR Optuna Select	-	-	-	-	-	5	-	-	-	5	-
GBR *Default* Top 10	30	-	-	-	-	-	-	-	-	30	-
GBR Random Top 10	-	-	-	-	-	-	-	-	-	-	-
GBR Optuna Top 10	-	-	-	-	-	-	-	-	-	-	-

Fonte: Elaborada pelo autor (2022)

Para o Histogram Gradiente Boosting Regressor (HGBR), os modelos Top 10 tiveram os piores resultados. Os modelos foram executados em 25 minutos e 57 segundos sem o uso de GPU ou de aceleradores (vide tabela 39 no apêndice D), um tempo baixo comparado com os demais modelos de *ensemble*[5] vistos até o momento, cumprindo a promessa de ser um modelo rápido para bases grandes. O modelo com melhor resultado foi o Histogram Gradiente Boosting Regressor Random (vide tabela 18 e gráfico 50 disponíveis no apêndice D).

No caso do LightGBM Regressor (LGBMR), as melhores performances foram obtidas pelos modelos otimizados. O tempo total de execução foi de 1 hora e 8 minutos utilizando o GPU, distribuídos conforme tabela 40 no apêndice D, sendo que 97,4% desse

[5] Métodos Ensemble: https://scikit-learn.org/stable/modules/ensemble.html

Tabela 18 – Análise da pontuação para os modelos de Histogram Gradiente Boosting Regressor (HGBR)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
HGBR *Default*	-	-	-	-	-	-	-	-	10	10	-
HGBR Random	-	25	20	30	20	30	20	20	10	175	MELHOR
HGBR Optuna	-	10	5	15	5	5	5	5	10	60	-
HGBR *Default* Select	20	-	-	-	-	-	-	-	-	20	-
HGBR Random Select	-	10	15	-	15	10	15	15	10	90	-
HGBR Optuna Select	-	-	5	-	5	-	5	5	5	25	-
HGBR *Default* Top 10	30	-	-	-	-	-	-	-	-	30	-
HGBR Random Top 10	-	-	-	-	-	-	-	-	-	-	-
HGBR Optuna Top 10	-	-	-	-	-	-	-	-	-	-	-

Fonte: Elaborada pelo autor (2022)

tempo foi gasto com o treinamento dos modelos Optuna. Esse treinamento surtiu efeito, visto que o modelo LightGBM Regressor Optuna obteve o melhor resultado, vide tabela 19. O gráfico 51 com as métricas analisadas está disponível no apêndice D.

Tabela 19 – Análise da pontuação para os modelos de LightGBM Regressor (LGBMR)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
LGBMR *Default*	-	-	-	-	-	-	-	-	30	30	-
LGBMR Random	-	10	10	5	10	10	10	10	-	65	-
LGBMR Optuna	-	20	20	20	20	20	20	20	-	140	MELHOR
LGBMR *Default* Select	20	-	-	-	-	-	-	-	15	35	-
LGBMR Random Select	-	5	5	10	5	5	5	5	-	40	-
LGBMR Optuna Select	-	10	10	10	10	10	10	10	-	70	-
LGBMR *Default* Top 10	30	-	-	-	-	-	-	-	-	30	-
LGBMR Random Top 10	-	-	-	-	-	-	-	-	-	-	-
LGBMR Optuna Top 10	-	-	-	-	-	-	-	-	-	-	-

Fonte: Elaborada pelo autor (2022)

Para o XGBoost Regressor (XGBR) os piores resultados ocorreram para os modelos sem otimização e para os Top 10, sendo observada perda de performance na avaliação dos conjuntos OOT e OOS. Através do uso da GPU, o tempo total de execução dos modelos foi de apenas 5 minutos (vide tabela 41 no apêndice D), um tempo extremamente baixo comparado com os demais modelos de ensemble. A alta velocidade de execução e o baixo tempo de execução torna esse algoritmo bem popular nas competições do Kaggle[6]. O melhor resultado foi obtido pelo XGBoost Regressor Random, conforme tabela 20. O gráfico 52 com as métricas regressivas está disponível no apêndice D.

Na análise do CatBoost Regressor (CBR) os piores resultados foram ocasionados pelos modelos Top 10. O tempo total de execução foi de 22 minutos e 1 segundo sem uso de GPU ou aceleradores, distribuídos conforme tabela 42 no apêndice D, sendo segundo menor tempo entre os métodos de ensemble. O modelo CatBoost Regressor *Default* obteve o melhor resultado (tabela 21 e gráfico 53 do apêndice D).

[6] Maiores informações em https://www.kaggle.com/kaggle-survey-2021

Tabela 20 – Análise da pontuação para os modelos de XGBoost Regressor (XGBR)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
XGBR *Default*	-	-	-	5	-	-	-	-	-	5	-
XGBR Random	-	20	20	20	20	15	20	20	30	165	MELHOR
XGBR Optuna	-	10	10	10	10	10	10	10	-	70	-
XGBR *Default* Select	-	-	-	-	-	-	-	-	-	-	-
XGBR Random Select	20	10	10	10	10	15	10	10	15	110	-
XGBR Optuna Select	-	5	5	-	5	5	5	5	-	30	-
XGBR *Default* Top 10	-	-	-	-	-	-	-	-	-	-	-
XGBR Random Top 10	30	-	-	-	-	-	-	-	-	30	-
XGBR Optuna Top 10	-	-	-	-	-	-	-	-	-	-	-

Fonte: Elaborada pelo autor (2022)

Tabela 21 – Análise da pontuação para os modelos de CatBoost Regressor (CBR)

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Runtime	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
CBR *Default*	-	25	25	30	25	25	25	25	-	180	MELHOR
CBR Random	-	-	-	-	-	-	-	-	10	10	-
CBR Optuna	-	20	15	10	15	15	15	15	-	105	-
CBR *Default* Select	-	-	-	5	-	5	-	-	-	10	-
CBR Random Select	20	-	5	-	5	-	5	5	10	50	-
CBR Optuna Select	-	-	-	-	-	-	-	-	10	10	-
CBR *Default* Top 10	-	-	-	-	-	-	-	-	10	10	-
CBR Random Top 10	30	-	-	-	-	-	-	-	-	30	-
CBR Optuna Top 10	-	-	-	-	-	-	-	-	5	5	-

Fonte: Elaborada pelo autor (2022)

5.2 Modelo Preditivo de Capacidade de Pagamento

Após analisar 162 modelos para selecionar o melhor modelo por estimador regressivo, totalizando 18 modelos, efetuamos uma análise com estes finalistas para selecionar o 1º e 2º melhores modelos entre todos, denominados Campeão e o Vice, respectivamente.

Utilizando o mesmo sistema de pontuação descrito anteriormente (capítulo 5.1), escolhemos como Campeão (1º lugar) o Gradiente Boosting Regressor Optuna e como Vice (2º lugar) o LightGBM Regressor Optuna, conforme resultado da pontuação disponível na tabela 22. Os gráficos 54 e 55 com os resultados de cada uma das métricas analisadas e a tabela 43 com o tempo de execução dos modelos analisados estão disponíveis no apêndice D.

Na análise dos 18 melhores modelos, verificamos que:

- Os modelos Passive-Aggressive Regressor e PLS Regression tiveram os piores resultados para a maioria das métricas regressivas;
- O XGBoost Regressor apresentou resultados não satisfatórios para a maioria das métricas nos conjuntos de dados OOT e OOS, considerando o resultado dos demais modelos Ensemble;

Tabela 22 – Análise da pontuação dos melhores estimadores regressivos

Modelo	PONTUAÇÃO NOS DIFERENTES INDICADORES										Resultado
	Processamento	MAE	MSE	MAPE	RMSE	MedAE	R2	Var Exp	Max Error	Total	
LinearReg Default	-	-	-	-	-	-	-	-	-	-	
Ridge Optuna	20	-	-	-	-	-	-	-	-	20	
Lasso Random	-	-	-	-	-	-	-	-	-	-	
ElasticNet Optuna	-	-	-	-	-	-	-	-	-	-	
Huber Random	-	-	-	-	-	-	-	-	-	-	
Pas-Aggr Random Select	-	-	-	-	-	-	-	-	-	-	
LinerSVR Random	-	-	-	-	-	-	-	-	-	-	
NuSVR Optuna Top 10	-	-	-	-	-	-	-	-	-	-	
KNN Optuna Top 10	-	-	-	-	-	-	-	-	-	-	
PLS Optuna Top 10	30	-	-	-	-	-	-	-	5	35	
DTR Optuna	-	-	-	-	-	-	-	-	-	-	
ETR Random	-	5	5	5	5	-	5	5	10	40	
RFR Optuna	-	-	-	-	-	-	-	-	-	-	
GBR Optuna	-	20	25	10	25	20	25	25	-	150	CAMPEÃO
HGBR Random	-	-	-	-	-	-	-	-	10	10	
LGBMR Optuna	-	20	-	30	-	25	-	-	-	75	VICE
XGBR Random	-	-	-	-	-	-	-	-	20	20	
CBR Default	-	-	15	-	15	-	15	15	-	60	

Fonte: Elaborada pelo autor (2022)

- Os melhores resultados para a maioria das métricas foram dos modelos de Ensemble (com exceção do XGboost Regressor), da Decision Tree Regressor e do K-Neighbors Regressor;
- Com relação a distribuição de modelos por hiperparâmetros, temos 2 modelos sem otimização (*default*), 7 otimizados pelo RandomizedSearchCV (Random) e 9 pelo Optuna com Neptuno (Optuna);
- Com relação ao conjunto de variáveis, temos 14 com 58 variáveis (todas), 1 com 39 variáveis (Select) e 3 com 10 variáveis (Top 10);
- Com relação ao tempo de execução, 50,6% do tempo total são referentes a dois modelos: Linear SVR e Random Forest Regressor;
- O XGBoost Regressor apresentou 4º menor tempo, o LightGBM Regressor 11º menor e o Gradiente Boosting Regressor 16º menor.

5.3 Abertura da "Caixa Preta" dos modelos

Um dos dificultadores para a utilização dos algoritmos complexos de *machine learning* para o desenvolvimento de modelos de risco de crédito é a dificuldade de interpretar e explicar o resultado obtido (vide capítulo 2.2.1). Uma das formas de resolver essa questão é a adoção de métodos para a extração das regras e justificativas do porquê uma decisão foi tomada pelo modelo.

Uma das ferramentas disponíveis para python para explicar o funcionamento dos modelos "caixa preta" é o *Local Interpretable Model-Agnostic Explanations* (Explicações

Agnósticas do Modelo Interpretável Local - LIME)[7]. De acordo com Ribeiro, Singh e Guestrin (2016), com o objetivo de entender a tomada de decisão por um modelo "caixa preta", o LIME perturba a entrada de dados fornecida ao modelo para verificar seu comportamento e identificar como as previsões mudam.

Em seguida, através um modelo de fácil interpretação desenvolvido a partir da análise das perturbações, como por exemplo um modelo linear sem coeficientes zerados, explica como localmente o modelo "caixa preta" tomou determinada decisão. É importante ressaltar que essa explicação é válida apenas para esse ponto em específico (modelo simples local), ou seja, na vizinhança da previsão que se deseja explicar e não representa como o modelo se comporta para todo o conjunto de dados (RIBEIRO; SINGH; GUESTRIN, 2016).

Neste trabalho utilizamos o LIME para analisar localmente o mesmo cliente pelos modelos Gradiente Boosting Regressor Optuna (modelo Campeão) e o LightGBM Regressor Optuna (Vice). Para essa análise configuramos o LIME para mostrar apenas as 12 variáveis mais importantes, de forma a facilitar a visualização, comparamos os resultados obtidos em cada modelo para esse cliente com as variáveis mais importantes do modelo de forma a ter clareza de como o valor da capacidade de pagamento foi atribuída ao cliente. Os resultados fornecidos pelo LIME para os modelos campeão e vice estão disponíveis respectivamente nas figuras 56 e 57 do apêndice D.

[7] LIME: maiores informações disponíveis em https://github.com/marcotcr/lime

6 Conclusão e estudos futuros

Este trabalho, que teve inicio na etapa de ”Entendimento de Negócio” da metodologia CRISP-DM (capítulo 2.3), envolveu a construção de diferentes modelos de capacidade de pagamento (MPCP) utilizando algoritmos supervisionados de aprendizagem de máquina para regressão, com a intenção de encontrar o modelo que melhor se aproxima dos valores para crédito que a instituição financeira deseja ofertar estrategicamente para atrair e fidelizar clientes no cenário de *Open Finance* (capítulo 1).

Após a análise do resultado de 162 modelos construídos pela combinação de 18 estimadores (capítulo 4.1), três grupos de seleção de variáveis: Todas, Selecionadas e Top 10 (capítulo 3.4) e três grupos de otimização de hiperparâmetros: sem otimização (Default), RandomizedSearchCV e Optuna (capítulo 4.2), escolhemos os dois melhores modelos a partir de um sistema de pontuação (capítulo 5.1) que considerou a performance computacional e o poder de previsibilidade diante de diferentes métricas regressivas nos conjuntos Teste, OOT e OOS (capítulo 4.3).

Os que apresentaram melhor resultado foram os modelos considerados ”caixa preta” Gradiente Boosting Regressor Optuna e o LightGBM Regressor Optuna, com valores médios[1] respectivamente de 22,36% e 21,40% na MAPE e 84,86% e 83,05% no R^2, resultados bem superiores[2] quando comparados com o melhor modelo do estimador mais simples analisado, o Linear Regression Default[3] que obteve 27,64% na MAPE e 79,94% R^2.

A utilização de modelos considerados ”caixa preta” na análise do risco de crédito pelas instituições financeiras é uma preocupação para as autoridades de supervisão bancária de diferentes países (capítulo 2.2.1), visto que a entrada e a saída do modelo é facilmente verificada, porém o funcionamento interno do modelo, quando disponível para inspeção, é de difícil interpretação.

No caso do Brasil essa preocupação relaciona-se principalmente aos modelos de Pontuação de Crédito (capítulo 2.1.3), que impactam em provisão (Basiléia II), requisitos de capital (Basiléia III) e são responsáveis diretos pela aprovação ou negação do pedido de crédito do cliente, assim como a sua atribuição de *rating*. No caso específico dos modelos de capacidade de pagamento, a legislações aborda apenas os valores máximos de crédito a serem oferecidos ao cliente nos produtos consignados (crédito parcelado consignado e

[1] Média entre os valores obtidos no conjunto de Teste, OOT e OOS.

[2] Durante a revisão da literatura não encontramos um referencial de valores para MAPE e R^2 em modelos preditivos de capacidade de pagamento.

[3] Para efeito de comparação, para esse estimador o modelo que obteve a menor pontuação geral foi o Linear Regression Grid/Optuna Select com MAPE de 27,58% e R^2 de 79,79%.

cartão consignado) e habitação, não abordando a questão da metodologia para se chegar a esse valor e não existindo portanto objeções legais a implantação de um modelo de Ensemble (capítulo 4.1).

Porém, a dificuldade em interpretar as decisões tomadas por MPCP de Ensemble dificulta a "venda" da solução para os stakeholders na etapa de "Avaliação" (capítulo 2.3).

Mesmo após a aprovação da solução e sua implantação na etapa de "Entrega", a falta de clareza de como o modelo calculou o valor a ser de crédito a ser disponibilizado ao cliente deixa a IF em uma situação vulnerável, principalmente no atendimento a questionamentos provenientes de SAC, Ouvidoria, Reclame Aqui, Processos Judiciais, entre outros.

Para evitar essa situação utilizamos o LIME (capítulo 5.3) para explicar como os modelos selecionados (Gradiente Boosting Regressor Optuna e o LightGBM Regressor Optuna) decidiram qual valor de crédito seria disponibilizado ao cliente em específico. O LIME utiliza um modelo linear simples para extrair as regras de qualquer modelo complexo para um determinado cliente, tornando-as facilmente interpretáveis, resolvendo o problema da "caixa preta".

Esse trabalho mostrou que diferentes técnicas de aprendizado de máquina, com excelente performance em relação aos modelos lineares tradicionais, podem ser aplicadas para a predição da capacidade de pagamento do cliente no ambiente bancário, altamente regulamentado.

Como esse trabalho focou apenas em alguns pontos específicos, sugerimos a realização de estudos futuros com o objetivo de:

- explorar por quais motivos os modelos com hiperparâmetros otimizados não tiveram sempre os melhores resultados em comparação aos modelos sem otimização;
- explorar por quais motivos a seleção de variáveis realizada com diferentes técnicas (Boruta, FeatureWiz, Select KBest e RFE) não apresentam a melhor performance;
- explorar os impactos do uso da redução de dimensionalidade em comparação com a seleção de variáveis na performance final dos modelos;
- testar outros algoritmos de ML supervisionados aplicados a regressão e comparar seus resultados com os estimadores utilizados, como a Rede Neural, Bagging, Stacking, entre outras;
- construir um modelo não supervisionado utilizando essa base de dados.

Referências

ALA'RAJ, M.; ABBOD, M. F.; MAJDALAWIEH, M. Modelling customers credit card behaviour using bidirectional lstm neural networks. *Journal of Big Data*, SpringerOpen, v. 8, n. 69, p. 1–27, May 2021. Citado 3 vezes nas páginas 33, 36 e 37.

ANDERSON, R. A. *Credit Intelligence & Modelling: Many Paths Through the Forest of Credit Rating and Scoring*. 2nd. ed. New York/USA: Oxford University Press, 2022. 944 p. Citado 4 vezes nas páginas 27, 28, 29 e 30.

BCB. *Resolução nº 2.682, de 21 de dezembro de 1999. Dispõe sobre critérios de classificação das operações de crédito e regras para constituição de provisão para créditos de liquidação duvidosa*. 1999. Disponível em: <https://www.bcb.gov.br/pre/normativos/res/1999/pdf/res_2682_v2_L.pdf>. Acesso em: 13 de agosto de 2021. Citado 4 vezes nas páginas 29, 31, 34 e 35.

BCB. *Circular nº 3.648, de 4 de março de 2013. Estabelece os requisitos mínimos para o cálculo da parcela relativa às exposições ao risco de crédito sujeitas ao cálculo do requerimento de capital mediante sistemas internos de classificação do risco de crédito (abordagens IRB) (RWACIRB), de que trata a Resolução nº 4.193, de 1º de março de 2013*. 2013. Disponível em: <https://www.bcb.gov.br/htms/Normativ/CIRCULAR3648.pdf>. Acesso em: 13 de agosto de 2021. Citado 2 vezes nas páginas 29 e 31.

BCB. *Resolução nº 4.557, de 23 de fevereiro de 2017. Dispõe sobre a estrutura de gerenciamento de riscos e a estrutura de gerenciamento de capital*. 2017. Disponível em: <https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/20471202/do1-2017-03-01-resolucao-n-4-557-de-23-de-fevereiro-de-2017-20471020>. Acesso em: 13 de agosto de 2021. Citado 4 vezes nas páginas 27, 31, 34 e 35.

BELO HORIZONTE. *Decreto nº 15.573, de 23 de maio de 2014. Estabelece normas para consignações em folha de pagamento dos servidores públicos ativos, aposentados, pensionistas e empregados públicos da administração direta, autárquica e fundacional do poder executivo do município de Belo Horizonte/MG*. 2014. Disponível em: <https://leismunicipais.com.br/a/mg/b/belo-horizonte/decreto/2014/1558/15573/decreto-n-15573-2014-estabelece-normas-para-consignacoes-em-folha-de-pagamento-dos-servidores-pu Acesso em: 12 de janeiro de 2022. Citado na página 32.

BIJAK, K. *Selected modelling problems in Credit Scoring*. 180 p. Tese (Doutorado) — University of Southampton, Southampton/United Kingdom, Aug. 2013. Citado na página 30.

BIJAK, K. et al. Credit card market literature review: Affordability and repayment. *FCA Market studies*, Financial Conduct Authority, p. 1–62, Nov. 2015. Citado 2 vezes nas páginas 30 e 31.

BIS. *Basel II: International Convergence of Capital Measurement and Capital Standards: A Revised Framework—Comprehensive Version*. 2006. Disponível em: <https://www.bis.org/publ/bcbs128.pdf>. Acesso em: 13 de agosto de 2021. Citado na página 29.

BRACKE, P. et al. Machine learning explainability in finance: an application to default risk analysis. *Staff Working Paper*, Bank of England, n. 816, p. 1–44, Aug. 2019. Citado na página 34.

BRASIL. *Lei nº 8.112, de 11 de dezembro de 1990. Dispõe sobre o regime jurídico dos servidores públicos civis da União, das autarquias e das fundações públicas federais*. 1990. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/l8213cons.htm>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

BRASIL. *Lei nº 8.213, de 24 de julho de 1991. Dispõe sobre os Planos de Benefícios da Previdência Social e dá outras providências*. 1991. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/l8213cons.htm>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

BRASIL. *Lei nº 8.692, de 28 de julho de 1993. Define planos de reajustamento dos encargos mensais e dos saldos devedores nos contratos de financiamentos habitacionais no âmbito do Sistema Financeiro da Habitação e dá outras providências*. 1993. Disponível em: <https://www.planalto.gov.br/ccivil_03/LEIS/L8692compilado.htm>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

BRASIL. *Lei nº 10.820, de 17 de dezembro de 2003. Dispõe sobre a autorização para desconto de prestações em folha de pagamento, e dá outras providências*. 2003. Disponível em: <http://www.planalto.gov.br/ccivil_03/leis/2003/L10.820Compilado.htm>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

BRASIL. *Lei nº 13.172, de 21 de outubro de 2015. Altera as Leis nº 10.820, de 17 de dezembro de 2003, nº 8.213, de 24 de julho de 1991, e nº 8.112, de 11 de dezembro de 1990, para dispor sobre desconto em folha de pagamento de valores destinados ao pagamento de cartão de crédito*. 2015. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13172.htm>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

BRASIL. *Lei nº 14.131, de 30 de março de 2021. Dispõe sobre o acréscimo de 5% (cinco por cento) ao percentual máximo para a contratação de operações de crédito com desconto automático em folha de pagamento até 31 de dezembro de 2021; e altera a Lei nº 8.213, de 24 de julho de 1991*. 2021. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2019-2022/2021/Lei/L14131.htm>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

BRASIL. *Lei nº 14.181, de 1 de julho de 2021. Altera a Lei nº 8.078, de 11 de setembro de 1990 (Código de Defesa do Consumidor), e a Lei nº 10.741, de 1º de outubro de 2003 (Estatuto do Idoso), para aperfeiçoar a disciplina do crédito ao consumidor e dispor sobre a prevenção e o tratamento do superendividamento*. 2021. Disponível em: <http://www.planalto.gov.br/ccivil_03/_Ato2019-2022/2021/Lei/L14181.htm>. Acesso em: 15 de janeiro de 2022. Citado 2 vezes nas páginas 32 e 33.

BRASIL. *Medida Provisória nº 1.106, de 17 de março de 2022. Altera a Lei nº 10.820, de 17 de dezembro de 2003, para ampliar a margem de crédito consignado aos segurados do Regime Geral de Previdência Social e para autorizar a realização de empréstimos e financiamentos mediante crédito consignado para beneficiários do Benefício de Prestação Continuada e de programas federais de transferência de renda, e a Lei nº 13.846, de 18 de*

julho de 2019, para dispor sobre a restituição de valores aos cofres públicos. 2022. Disponível em: <https://www.planalto.gov.br/ccivil_03/_Ato2019-2022/2022/Mpv/mpv1106.htm>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

BRUCE, P.; BRUCE, A.; GEDECK, P. *Practical statistics for data scientists: 50+ essential concepts using R and Python*. 2nd. ed. Sebastopol/USA: O'Reilly Media, 2020. 368 p. Citado na página 42.

DASTILE, X.; CELIK, T.; POTSANE, M. Statistical and machine learning models in credit scoring: A systematic literature survey. *Applied Soft Computing*, Elsevier, v. 91, n. 106263, p. 1–21, Mar. 2020. Citado 3 vezes nas páginas 35, 36 e 37.

DELOITTE. *Pesquisa FEBRABAN de Tecnologia Bancária 2021*. 2021. Disponível em: <https://portal.febraban.org.br/pagina/3106/48/pt-br/pesquisa>. Acesso em: 27 de agosto de 2021. Citado 2 vezes nas páginas 23 e 24.

DUMITRESCU, E. et al. Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, Elsevier, v. 297, n. 3, p. 1178–1192, Mar. 2022. Citado 3 vezes nas páginas 33, 35 e 36.

DUPONT, L.; FLICHE, O.; YANG, S. Governance of artificial intelligence in finance. *ACPR Discussion document*, Banque de France, p. 1–82, June 2020. Citado na página 34.

EUROPEAN, C. et al. White paper on artificial intelligence: A european approach to excellence and trust. *COM*, European Commission Brussels, v. 65 Final, p. 1–27, Feb. 2020. Citado na página 34.

EXÉRCITO. *Portaria nº 124 SEF/C Ex, de 18 de fevereiro de 2021. Aprova as instruções reguladoras para consignação de descontos em folha de pagamento (EB90-IR- 02.001), 1ª Edição, 2021*. 2021. Disponível em: <http://www.sef.eb.mil.br/images/ass2/portarias/port_n_124_sef_18fev2021.html>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

FEBRABAN. *Regulamentação Lei do Superendividamento*. 2021. Disponível em: <https://cmsarquivos.febraban.org.br/Arquivos/documentos/PDF/SEMARC2021-Regulamentaç~aoSuperendividamento-AmauryOliva.pdf>. Acesso em: 15 de abril de 2022. Citado na página 33.

GUPTA, P.; SEHGAL, N. K. *Introduction to machine learning in the cloud with python: Concepts and practices*. Cham/Switzerland: Springer Nature, 2021. 303 p. Citado na página 33.

HAPSILA, A.; ASTARINA, I. The effect of character, capacity, capital, collateral and condition of economy on giving credit. *Jurnal Manajemen dan Bisnis*, v. 9, n. 1, p. 41–50, June 2020. Citado na página 28.

JOSEPH, C. *Advanced Credit Risk Analysis and Management*. Chichester/United Kingdom: John Wiley & Sons, 2013. 448 p. (The Wiley Finance Series). Citado 3 vezes nas páginas 27, 28 e 29.

MUKHIYA, S. K.; AHMED, U. *Hands-On Exploratory Data Analysis with Python: Perform EDA techniques to understand, summarize, and investigate your data*. Birmingham/United Kingdom: Packt Publishing Ltd, 2020. 352 p. Citado na página 42.

MÜLLER, A. C.; GUIDO, S. *Introduction to machine learning with Python: a guide for data scientists*. Sebastopol/USA: O'Reilly Media, 2016. 398 p. Citado na página 54.

PARANÁ. *Decreto nº 9220, de 28 de outubro de 2021. Regulamenta a Lei nº 20.740, de 05 de outubro de 2021, que dispõe sobre as normas pertinentes aos descontos e consignação em folhas de pagamento de servidores civis e militares, ativos e inativos, assim como de Pensionistas de geradores de pensão do Estado do Paraná, e dá outras providências*. 2021. Disponível em: <https://leisestaduais.com.br/pr/decreto-n-9220-2021-parana-regulamenta-a-lei-no-20-740-de-05-de-outubro-de-2021>. Acesso em: 12 de janeiro de 2022. Citado na página 32.

RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. "Why Should I Trust You?": Explaining the predictions of any classifier. In: *Proceedings of the 22º ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco/USA: KDD, 2016. p. 1135–1144. Citado na página 70.

SAMMUT, C.; WEBB, G. I. *Encyclopedia of Machine Learning and Data Mining*. 2nd. ed. New York/USA: Springer Science+Business Media, 2017. 1352 p. Citado na página 54.

SCHRÖER, C.; KRUSE, F.; GÓMEZ, J. M. A systematic literature review on applying CRISP-DM process model. *Procedia Computer Science*, Elsevier, v. 181, p. 526–534, Jan. 2021. Citado 3 vezes nas páginas 37, 38 e 39.

SHARMA, D. Not if affordability data adds value but how to add real value by leveraging affordability data: Enhancing predictive capability of credit scoring using affordability data. *SSRN Electronic Journal*, Elsevier, n. 1801346, p. 1–52, Aug. 2009. Citado 2 vezes nas páginas 30 e 31.

TRIVEDI, S. K. A study on credit scoring modeling with different feature selection and machine learning approaches. *Technology in Society*, Elsevier, v. 63, n. 101413, p. 1–9, Nov. 2020. Citado na página 36.

TURKSON, R. E.; BAAGYERE, E. Y.; WENYA, G. E. A machine learning approach for predicting bank credit worthiness. In: *2016 Third International Conference on Artificial Intelligence and Pattern Recognition (AIPR)*. Lodz/Poland: IEEE, 2016. p. 1–7. Citado 2 vezes nas páginas 29 e 31.

WIRTH, R.; HIPP, J. CRISP-DM: Towards a standard process model for data mining. In: *Proceedings of the Fourth international conference on the practical applications of knowledge discovery and data mining*. Manchester/United Kingdom: Practical Application Company, 2000. v. 1, p. 29–40. Citado 3 vezes nas páginas 37, 38 e 39.

Apêndices

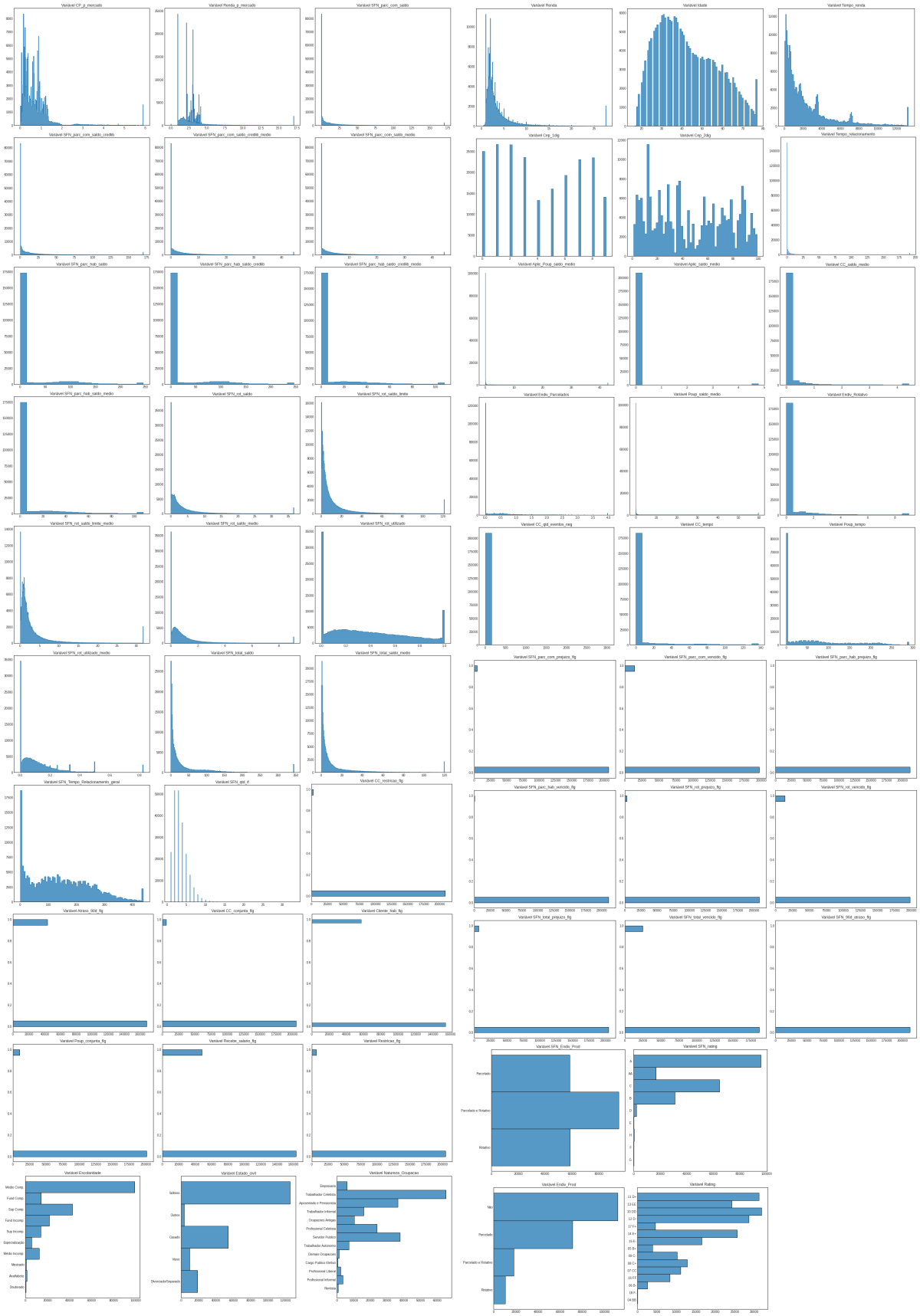
APÊNDICE A – Aspectos Metodológicos

Em relação ao método de pesquisa, a dissertação é considerada:

- Empírica, de acordo com o nível, porque tem como objetivo efetuar uma comprovação prática de fundamentos teóricos, ao aplicar os conhecimentos relacionados a aprendizagem de máquina e risco de crédito na construção de um modelo preditivo;
- Dedutiva, de acordo com o método científico, que é caminho sistemático para se chegar a uma conclusão, porque parte de uma análise geral a respeito dos modelos de risco de crédito (premissa verdadeira) para se chegar a uma conclusão específica de um modelo preditivo para prospecção PF (resultado), através do uso da dedução e do raciocínio lógico.
- Quantitativa, de acordo com a abordagem, que é a forma de estudar o objeto, porque a investigação e a tomada de decisão são baseadas na análise quantificada dos dados coletados (dos números, tabelas e gráficos gerados a partir da base de dados);
- Aplicada e de uso prático, de acordo com a natureza, que está relacionada a finalidade e a contribuição da pesquisa, pois visa construir um modelo através do uso de teorias e métodos validados pela comunidade acadêmica, permitindo uma aplicação imediata dos resultados obtidos pela pesquisa após a sua finalização.
- Explicativa, de acordo com os objetivos, que está relacionado ao grau de aprofundamento da pesquisa, porque se propõem determinar quais são as variáveis independentes (demográficas, comportamentais ou de mercado) que mensuram adequadamente o risco do cliente, estabelecendo uma relação entre essas variáveis independentes e a variável dependente (desempenho) para construção do modelo preditivo.
- *Ex-Post-Facto*, de acordo com os procedimentos técnicos, visto que investiga a base de dados para descobrir a relação existente entre as variáveis independentes (demográficas, comportamentais e de mercado) que estavam presentes no momento da contratação do crédito (passado) e o desempenho do cliente no pagamento desse contrato (evento posterior), não sendo possível efetuar manipulações dessas variáveis que já ocorreram para se obter resultados futuros diferentes.

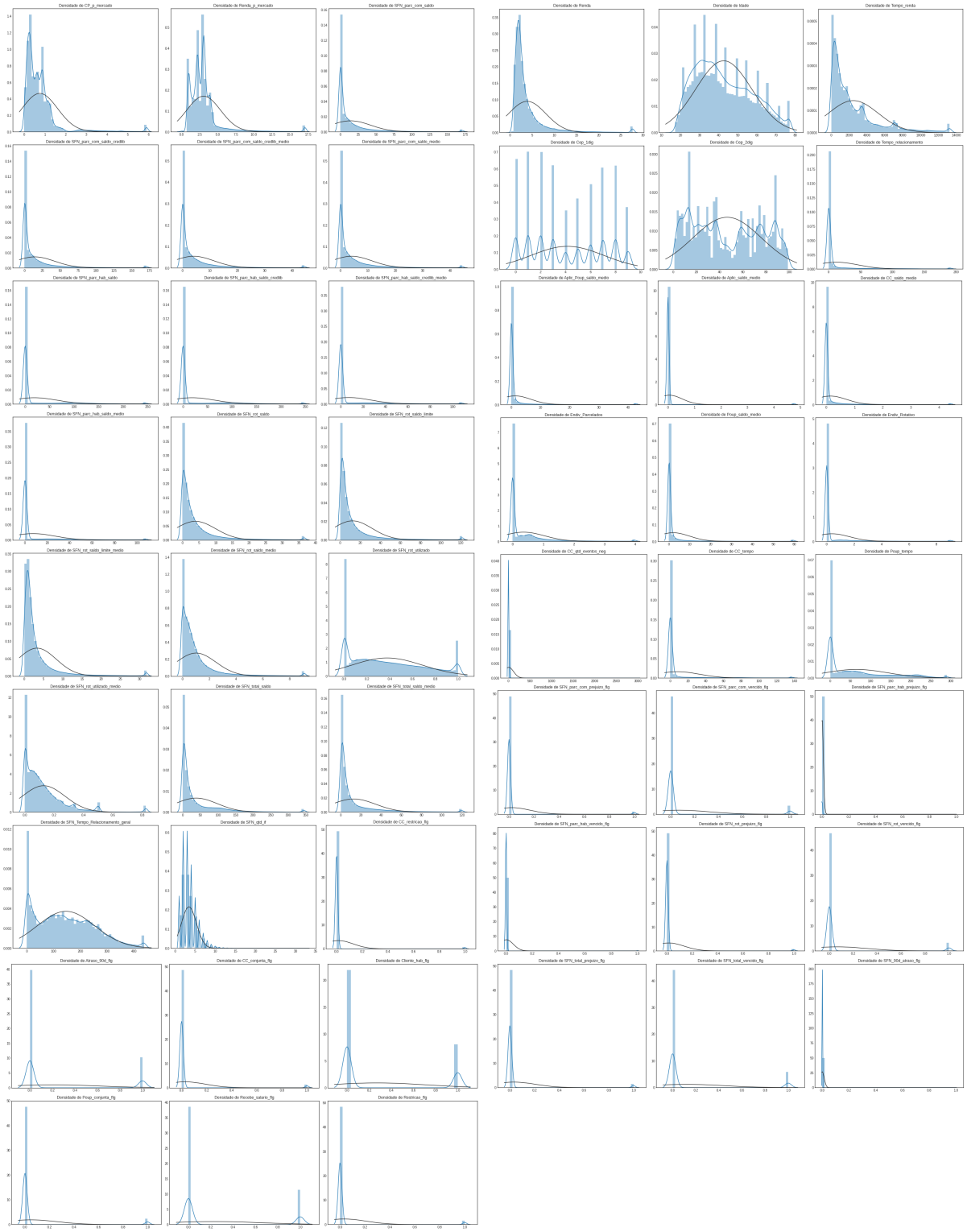
APÊNDICE B – EDA: Tabelas e Gráficos Complementares

Figura 5 – Histograma da distribuição das Variáveis Independentes



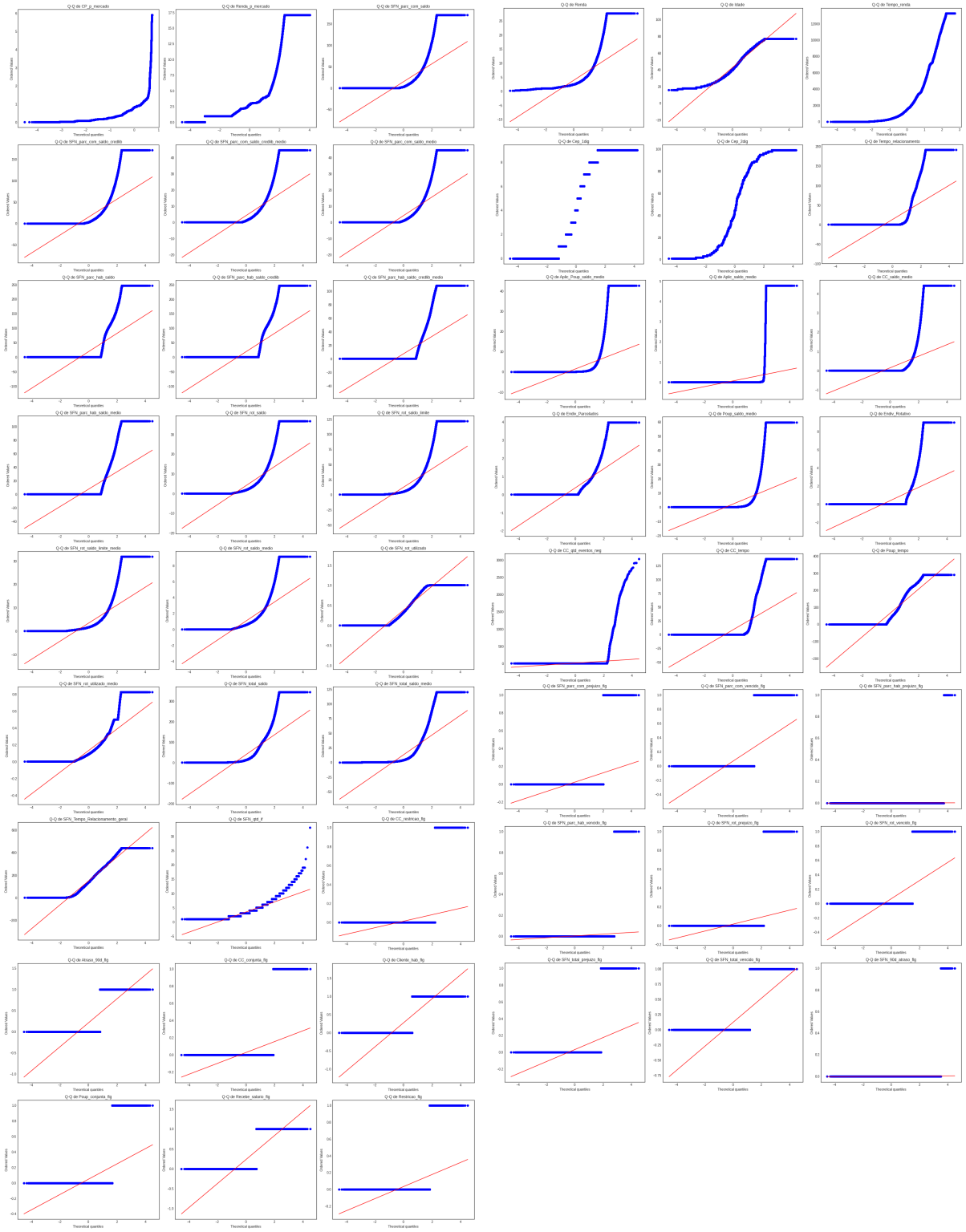
Fonte: Elaborada pelo autor (2022)

Figura 6 – Histograma pela Máxima Verossimilhança Gaussiana das Variáveis Independentes



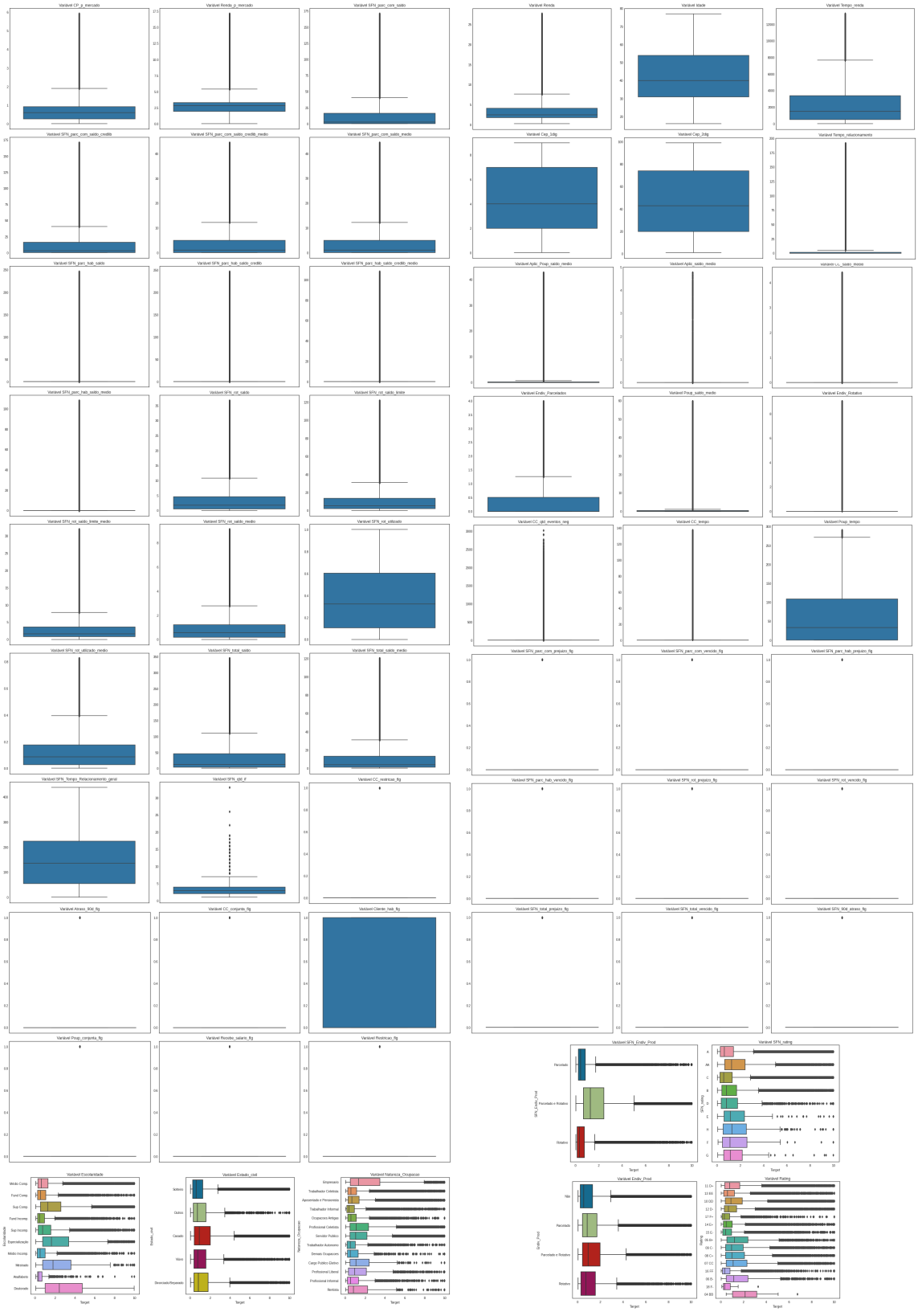
Fonte: Elaborada pelo autor (2022)

Figura 7 – Probabilidade Quantil-Quantil das Variáveis Independentes



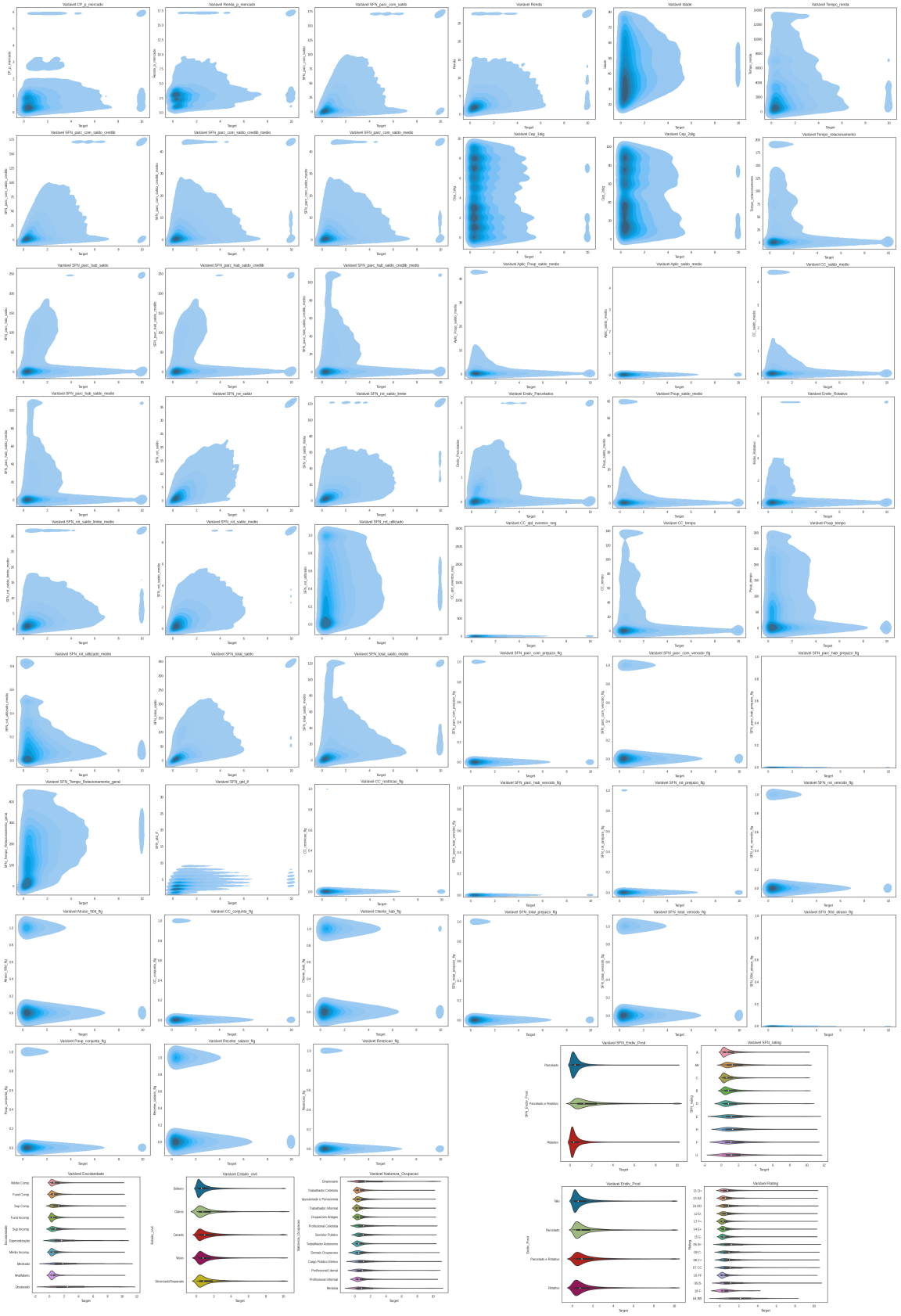
Fonte: Elaborada pelo autor (2022)

Figura 8 – Diagrama de Caixa das Variáveis Independentes



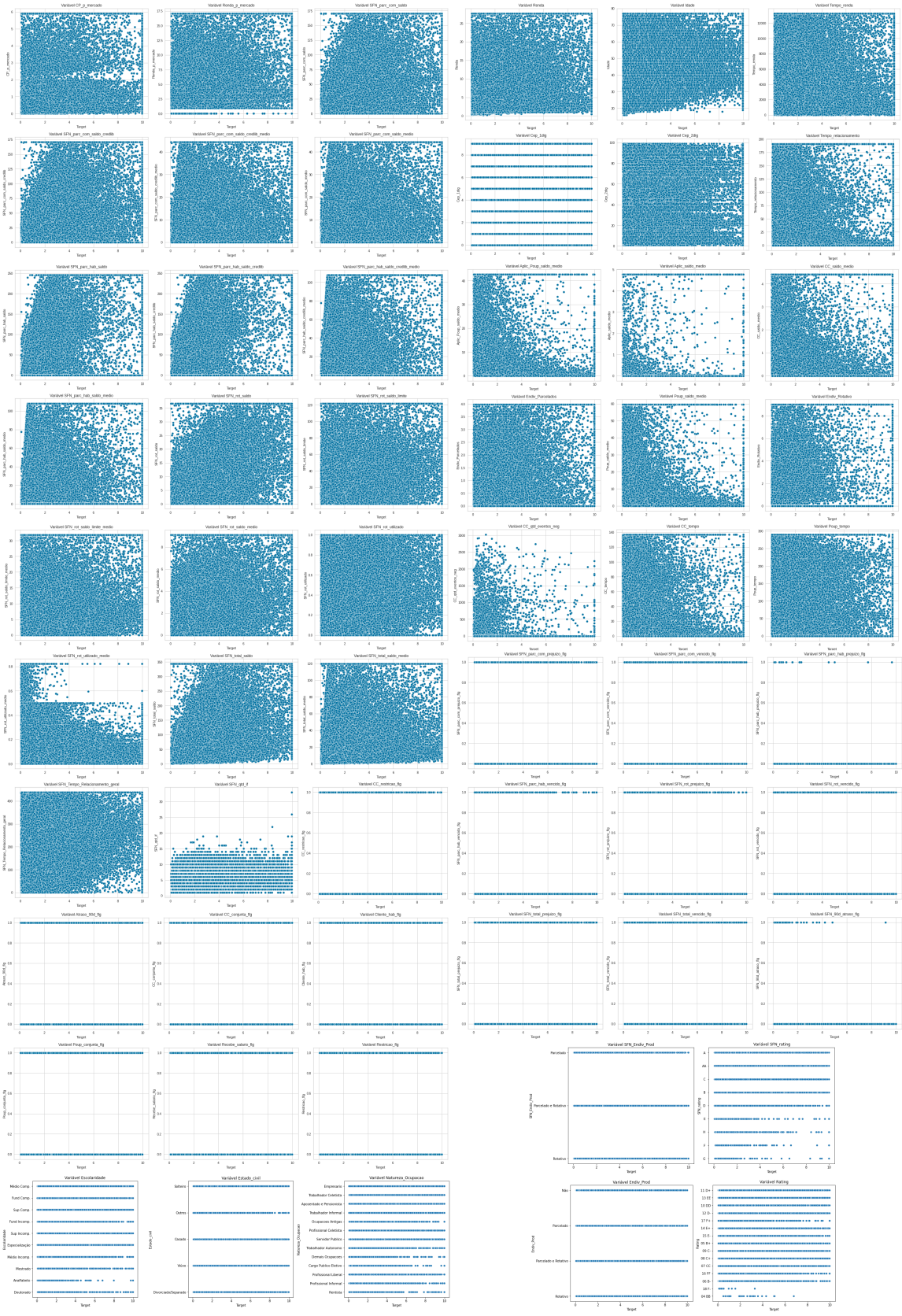
Fonte: Elaborada pelo autor (2022)

Figura 9 – Estimativa de Densidade por Kernel das Variáveis Independentes pelo Target



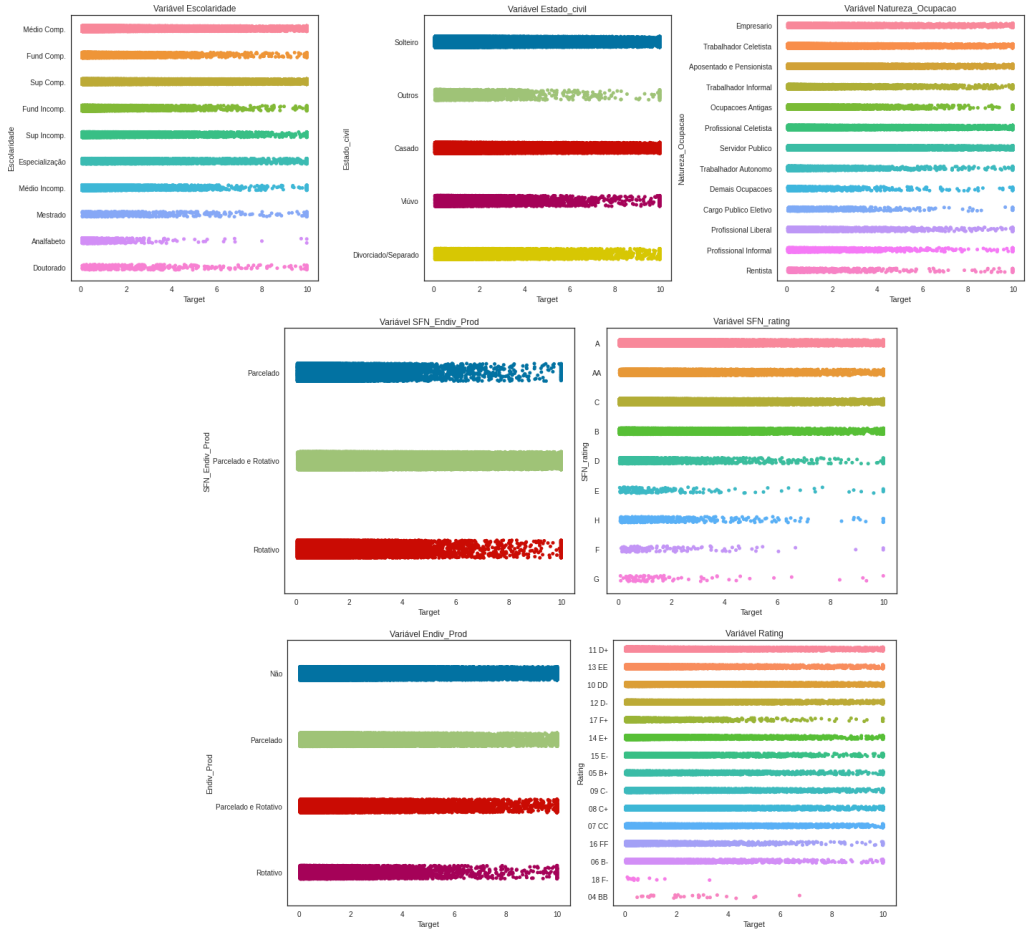
Fonte: Elaborada pelo autor (2022)

Figura 10 – Gráfico de Dispersão das Variáveis Independentes pelo Target



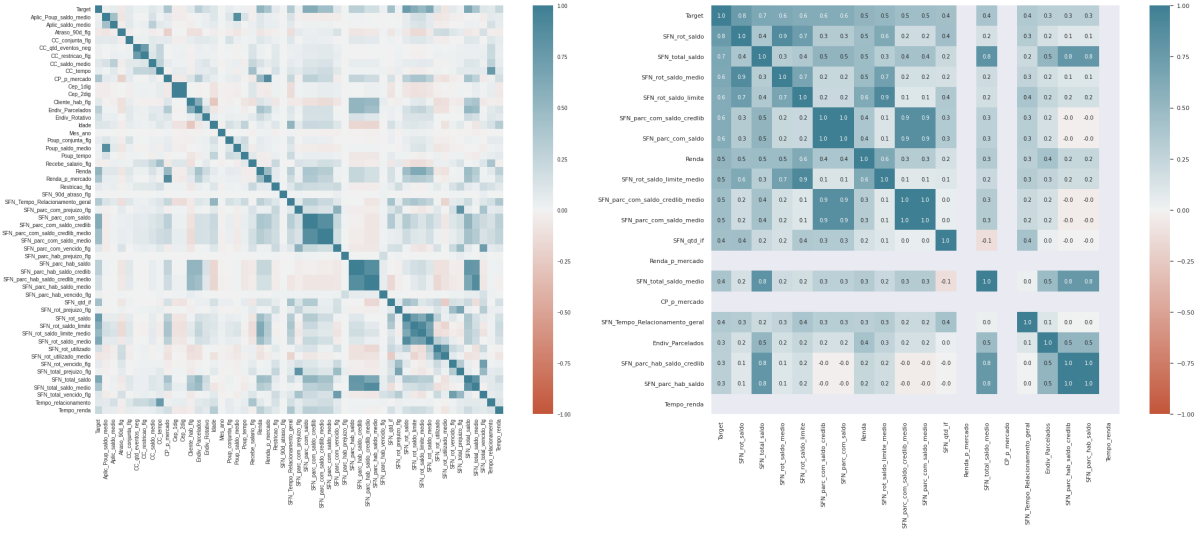
Fonte: Elaborada pelo autor (2022)

Figura 11 – Gráfico de Dispersão das Variáveis Independentes Categóricas pelo Target



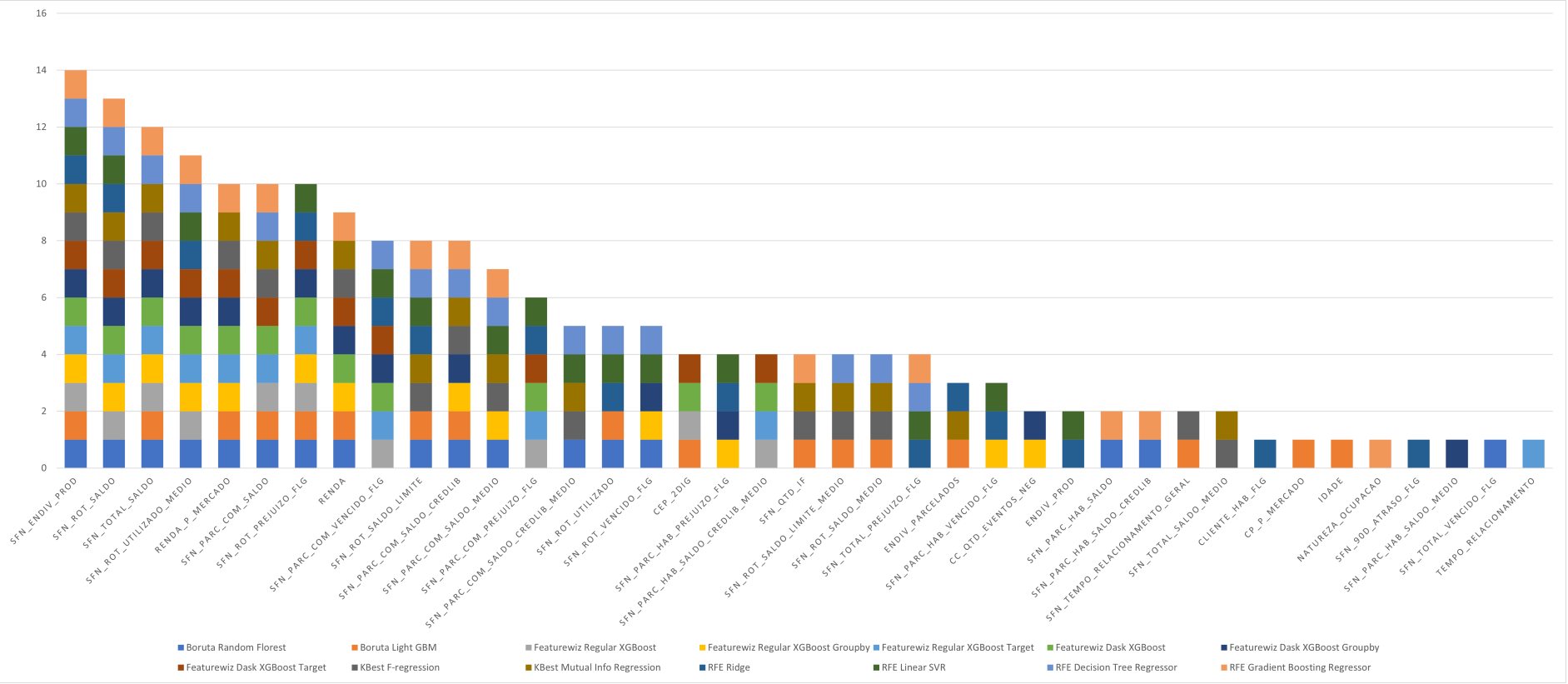
Fonte: Elaborada pelo autor (2022)

Figura 12 – Mapa de Calor - Matriz de Correlação



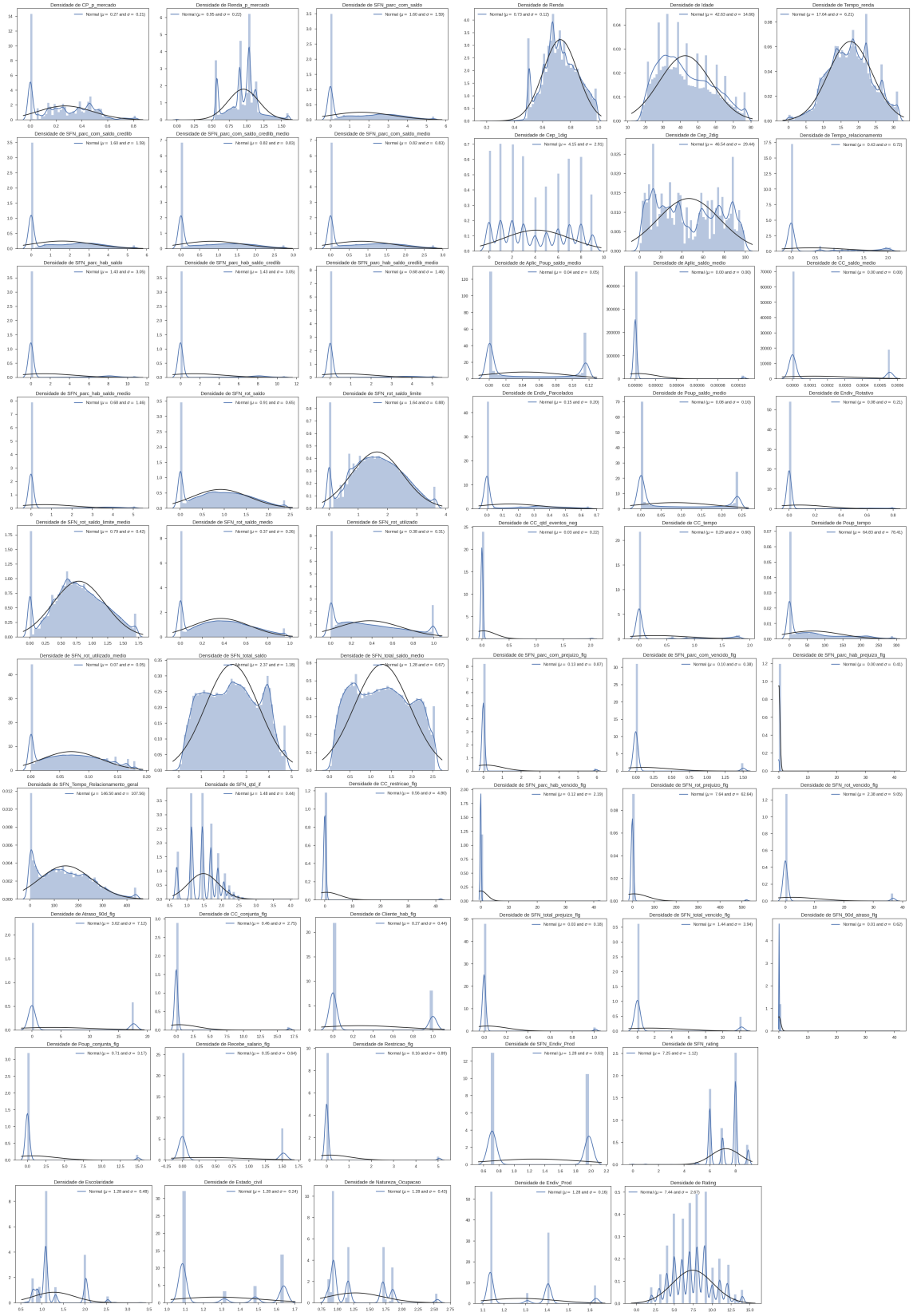
Fonte: Elaborada pelo autor (2022)

Figura 13 – Seleção de Variáveis



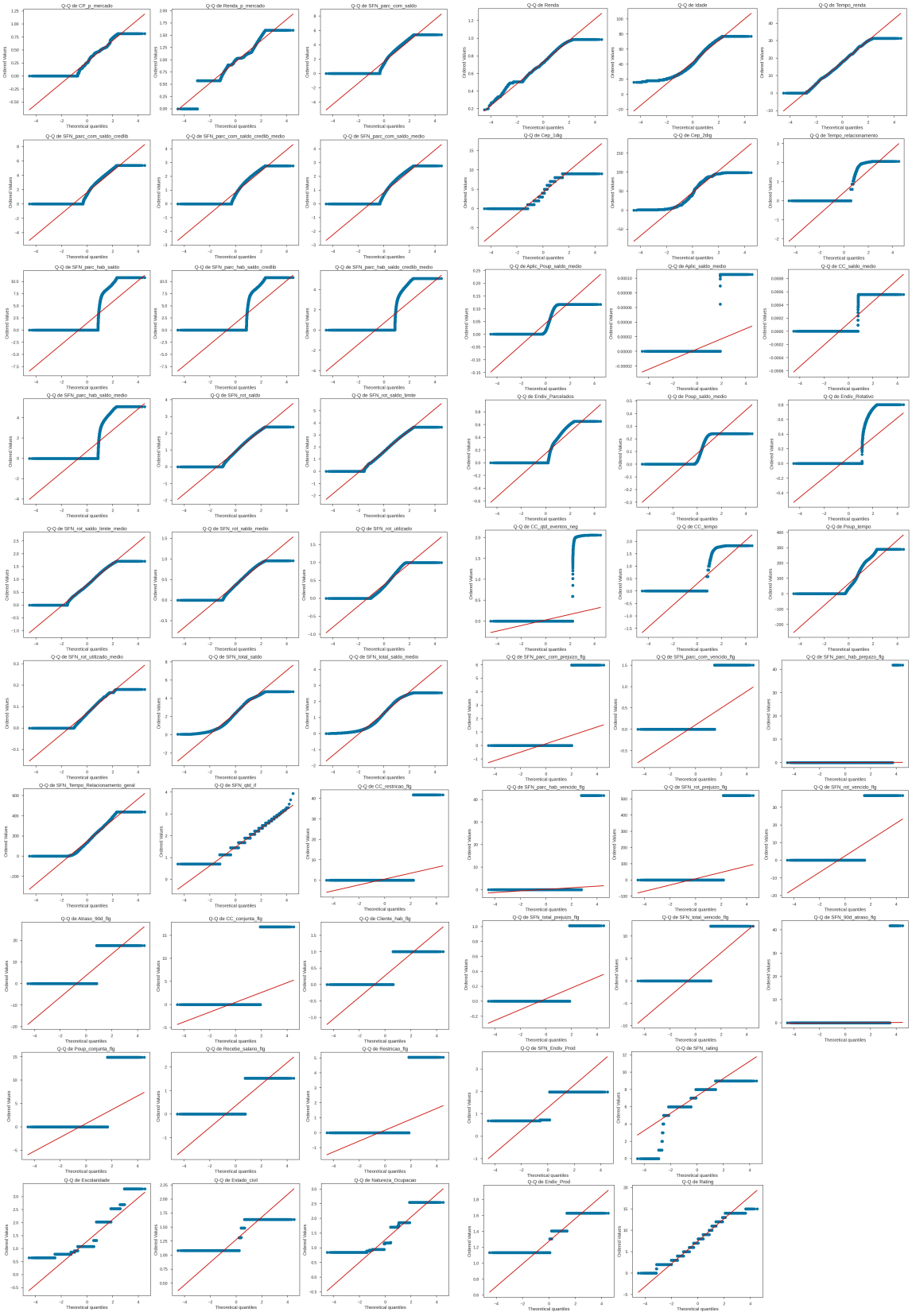
Fonte: Elaborada pelo autor (2022)

Figura 14 – Histograma pela Máxima Verossimilhança Gaussiana das Variáveis Independentes (tratadas e transformadas)



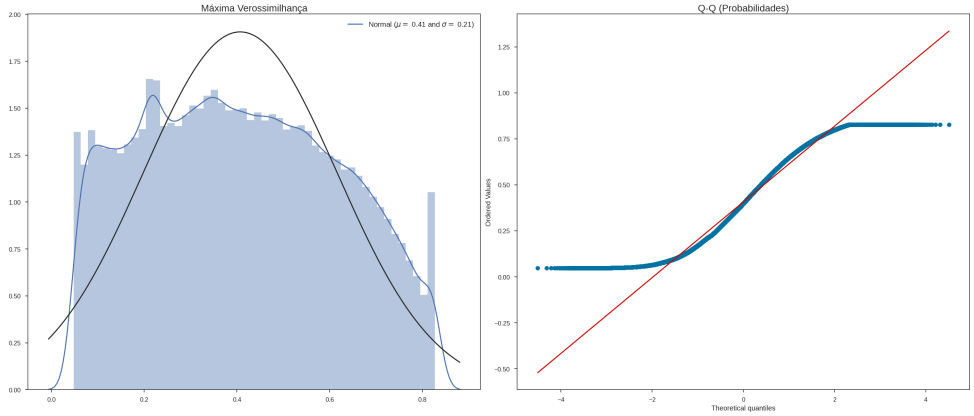
Fonte: Elaborada pelo autor (2022)

Figura 15 – Probabilidade Quantil-Quantil das Variáveis Independentes (tratadas e transformadas)



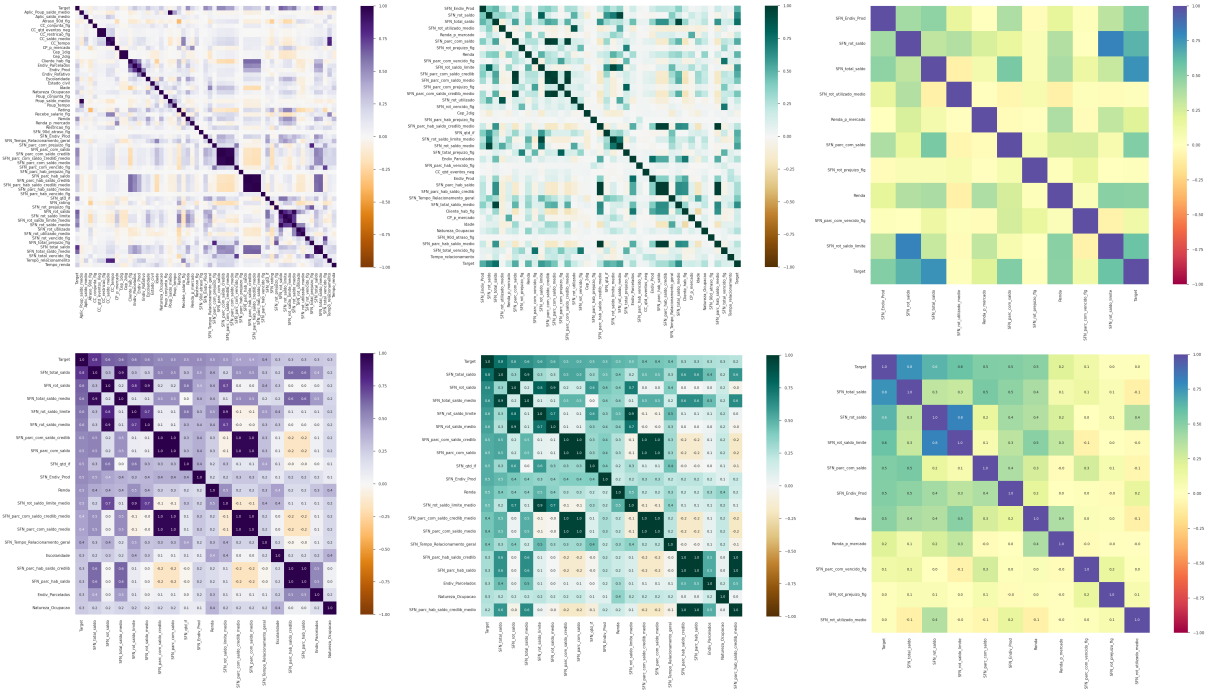
Fonte: Elaborada pelo autor (2022)

Figura 16 – Distribuição da Variável Dependente (tratada e transformada)



Fonte: Elaborada pelo autor (2022)

Figura 17 – Mapa de Calor - Matriz de Correlação por seleção de variável (tratadas e transformadas)



Fonte: Elaborada pelo autor (2022)

APÊNDICE C – Metodologia: Tabelas e Gráficos Complementares

Tabela 23 – Melhores Hiperparâmetros do GridSearchCV - RandomizedSearchCV

Estimadores	GridSearchCV[1] ou RandomizedSearchCV[2]
Linear Regression[1]	{"fit_intercept": False, "positive": False}
Ridge Regression[2]	{"tol": 0.0005, "solver": "svd", "max_iter": 9881, "alpha": 1}
Lasso Regression[2]	{"tol": 1e-05, "selection": "random", "positive": False, "max_iter": 3642, "alpha": 1e-05}
Elastic Net Regression[2]	{"tol": 0.01, "selection": "cyclic", "positive": False, "max_iter": 2316, "l1_ratio": 0, "alpha": 5e-05}
Huber Regressor[2]	{"tol": 0.05, "max_iter": 10207, "epsilon": 1.4, "alpha": 4}
Passive-Aggressive Regressor[2]	{"tol": 1e-05, "max_iter": 5220, "loss": "epsilon_insensitive", "fit_intercept": True, "C": 1e-05}
Linear SVR[2]	{"tol": 0.0005, "max_iter": 14296, "loss": "squared_epsilon_insensitive", "epsilon": 0, "C": 0.05}
Nu SVR[2]	{"tol": 1e-05, "nu": 0.9, "max_iter": 10831, "kernel": "linear", "degree": 2, "coef0": 1, "C": 5e-05}
K-Neighbors Regressor[2]	{"weights": "distance", "p": 1, "n_neighbors": 11, "leaf_size": 30}
PLS Regression[2]	{"tol": 0.05, "scale": True, "n_components": 2, "max_iter": 4075}
Decision Tree Regressor[2]	{"min_samples_split": 73, "min_samples_leaf": 32, "max_features": "auto", "max_depth": 15}
Extra Tree Regressor[2]	{"n_estimators": 317, "min_samples_split": 14, "min_samples_leaf": 6, "max_features": "auto", "max_depth": 22}
Random Forest Regressor[2]	{"n_estimators": 317, "min_samples_split": 14, "min_samples_leaf": 6, "max_features": "auto", "max_depth": 22}
Gradiente Boosting Regressor[2]	{"n_estimators": 1283, "min_samples_split": 98, "min_samples_leaf": 38, "max_features": "sqrt", "max_depth": 9, "learning_rate": 0.01}
Hist. Gradiente Boosting Regressor[2]	{"min_samples_leaf": 41, "max_iter": 1269, "max_depth": 20, "learning_rate": 0.01}
LightGBM Regressor[2]	{"reg_lambda": 0, "reg_alpha": 0.1, "objective": "regression_l1", "n_estimators": 651, "min_child_samples": 27, "max_depth": 19, "learning_rate": 0.05, "boosting_type": "gbdt"}
XGBoost Regressor[2]	{"n_estimators": 193, "max_depth": 4, "learning_rate": 0.05, "feature_selector": "shuffle", "booster": "gbtree"}
CatBoost Regressor[2]	{"n_estimators": 483, "max_depth": 6, "learning_rate": 0.05}

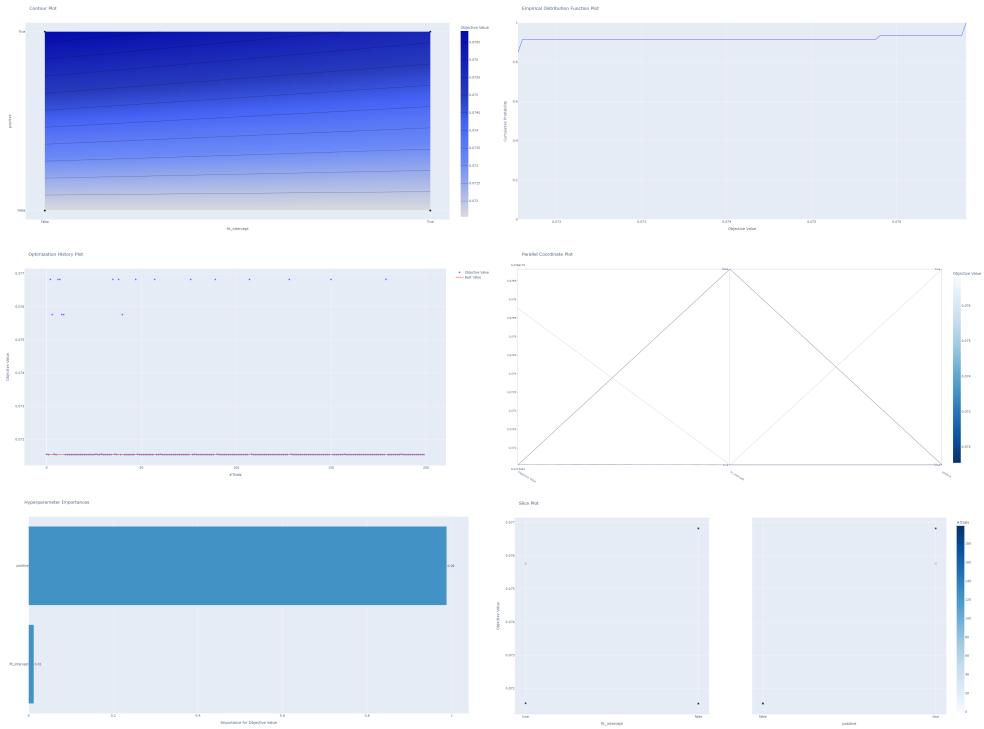
Fonte: Elaborada pelo autor (2022)

Tabela 24 – Melhores Hiperparâmetros do Optuna e Neptuno

Estimadores	Optuna com Neptuno
Linear Regression	{"fit_intercept": False, "positive": False}
Ridge Regression	{"alpha": 1, "max_iter": 7016, "tol": 0.05, "solver": "cholesky"}
Lasso Regression	{"alpha": 1e-05, "max_iter": 6611, "tol": 0.05, "selection": "cyclic", "positive": False}
Elastic Net Regression	{"alpha": 1e-05, "l1_ratio": 0.19008764566104874, "max_iter": 14734, "tol": 0.05, "selection": "random", "positive": False}
Huber Regressor	{"epsilon": 1.2091537446837244, "max_iter": 12599, "alpha": 3, "tol": 0.01}
Passive-Aggressive Regressor	{"C": 5, "tol": 0.0001, "max_iter": 14970, "loss": "squared_epsilon_insensitive", "fit_intercept": False}
Linear SVR	{"epsilon": 0.10002023752183387, "tol": 5e-05, "C": 0.05, "loss": "epsilon_insensitive", "max_iter": 7522}
Nu SVR	{"nu": 0.9459743180916032, "tol": 0.1, "C": 10, "max_iter": 14480, "kernel": "rbf", "degree": 9, "coef0": 7}
K-Neighbors Regressor	{"n_neighbors": 13, "weights": "distance", "leaf_size": 32, "p": 1}
PLS Regression	{"n_components": 3, "scale": True, "max_iter": 10580, "tol": 0.1}
Decision Tree Regressor	{"max_features": "auto", "max_depth": 11, "min_samples_split": 62, "min_samples_leaf": 37}
Extra Tree Regressor	{"max_features": "auto", "max_depth": 29, "n_estimators": 1936, "min_samples_split": 8, "min_samples_leaf": 3}
Random Forest Regressor	{"max_features": "auto", "max_depth": 30, "n_estimators": 2075, "min_samples_split": 8, "min_samples_leaf": 4}
Gradiente Boosting Regressor	{"max_features": "sqrt", "max_depth": 19, "n_estimators": 764, "min_samples_split": 50, "min_samples_leaf": 48, "learning_rate": 0.01}
Hist. Gradiente Boosting Regressor	{"max_depth": 29, "max_iter": 916, "min_samples_leaf": 62, "learning_rate": 0.01}
LightGBM Regressor	{"objective": "regression_l1", "max_depth": 29, "n_estimators": 1425, "min_samples_leaf": 63, "learning_rate": 0.3, "boosting_type": "dart", "reg_alpha": 0.05, "reg_lambda": 0}
XGBoost Regressor	{"feature_selector": "shuffle", "max_depth": 7, "n_estimators": 917, "learning_rate": 0.01, "booster": "gbtree"}
CatBoost Regressor	{"max_depth": 7, "n_estimators": 898, "learning_rate": 0.05}

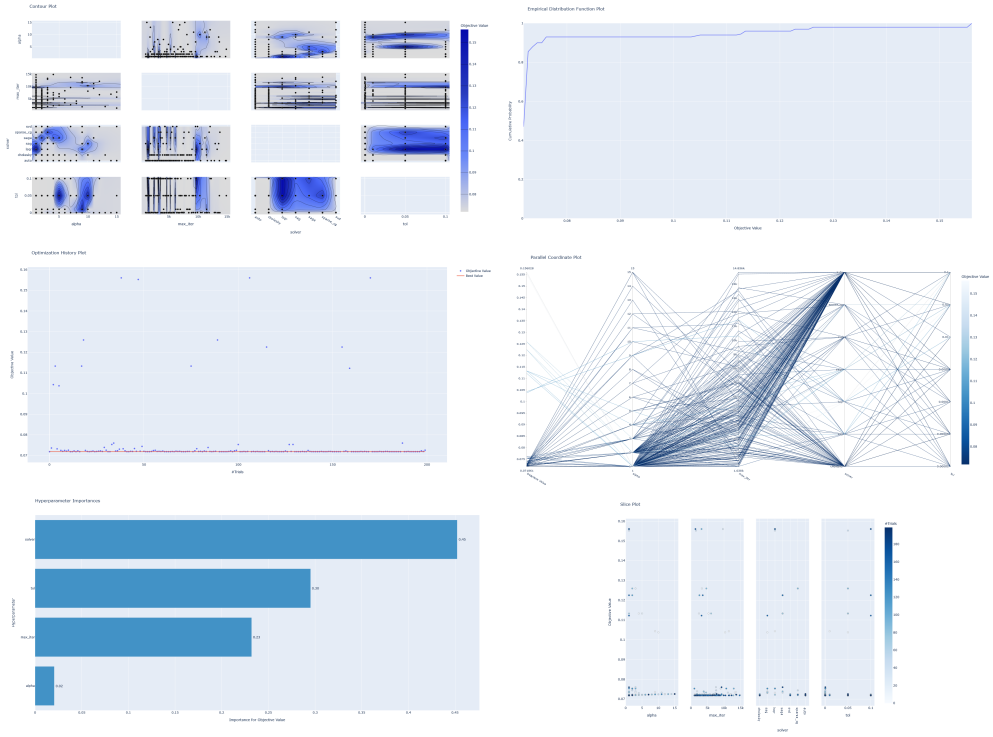
Fonte: Elaborada pelo autor (2022)

Figura 18 – Otimização pelo Optuna - Liner Regression



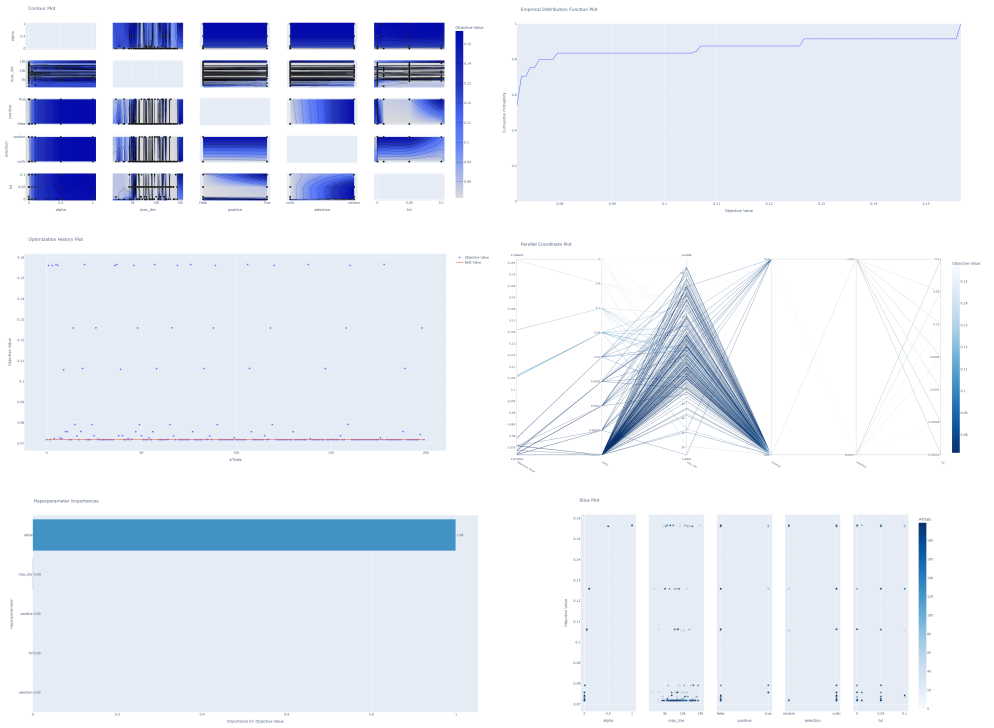
Fonte: Elaborada pelo autor (2022)

Figura 19 – Otimização pelo Optuna - Ridge Regression



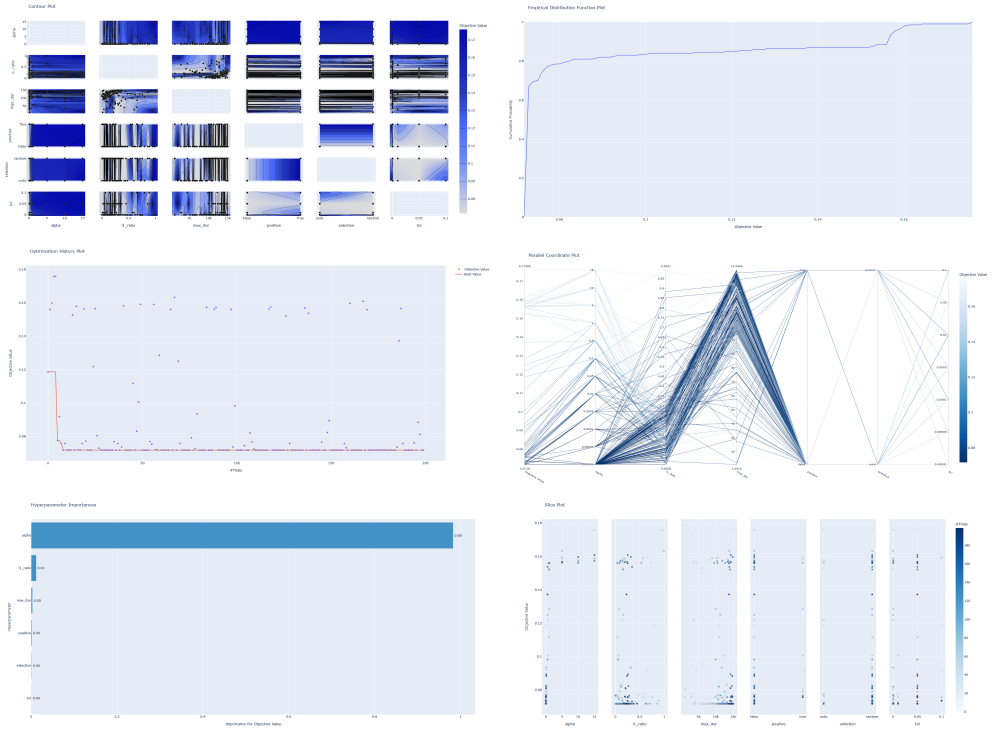
Fonte: Elaborada pelo autor (2022)

Figura 20 – Otimização pelo Optuna - Lasso Regression



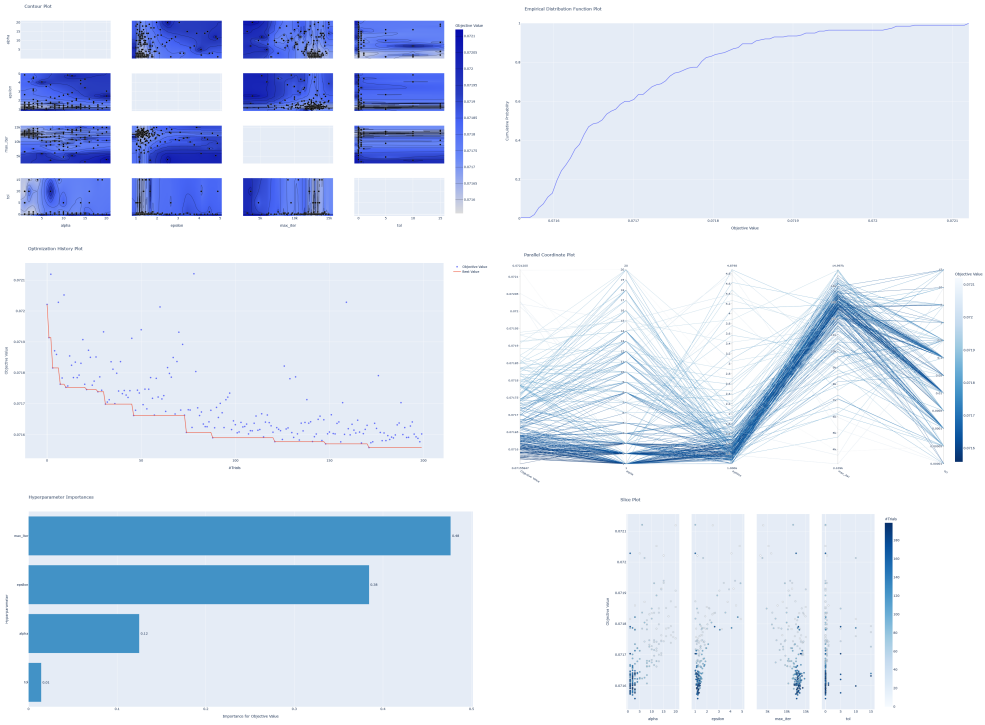
Fonte: Elaborada pelo autor (2022)

Figura 21 – Otimização pelo Optuna - Elastic Net Regression



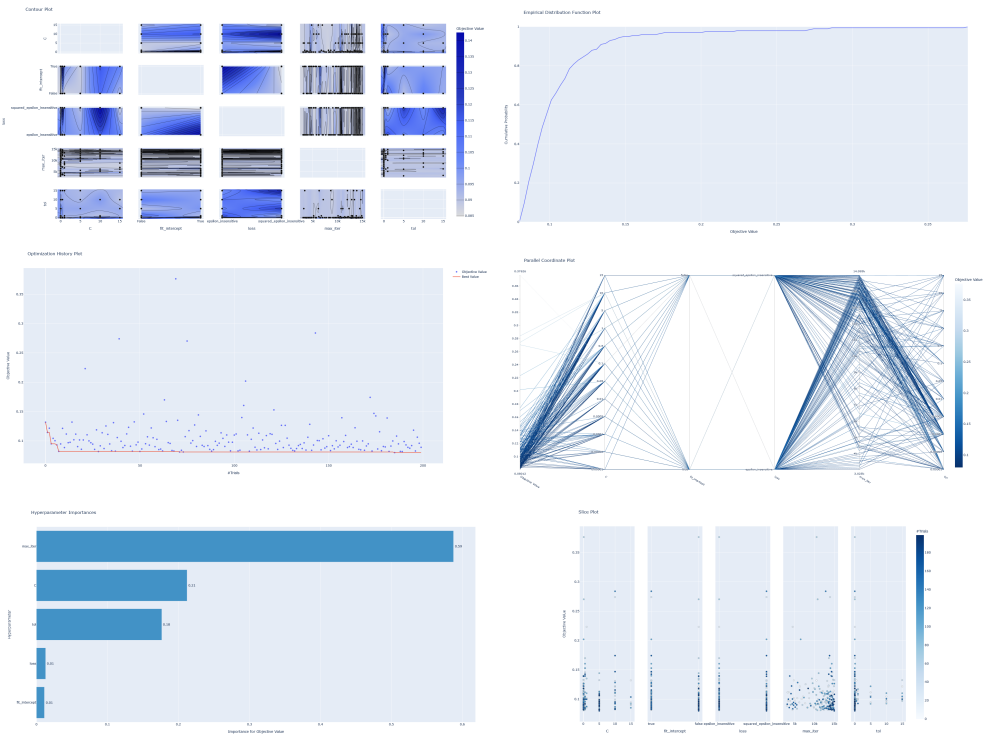
Fonte: Elaborada pelo autor (2022)

Figura 22 – Otimização pelo Optuna - Huber Regressor



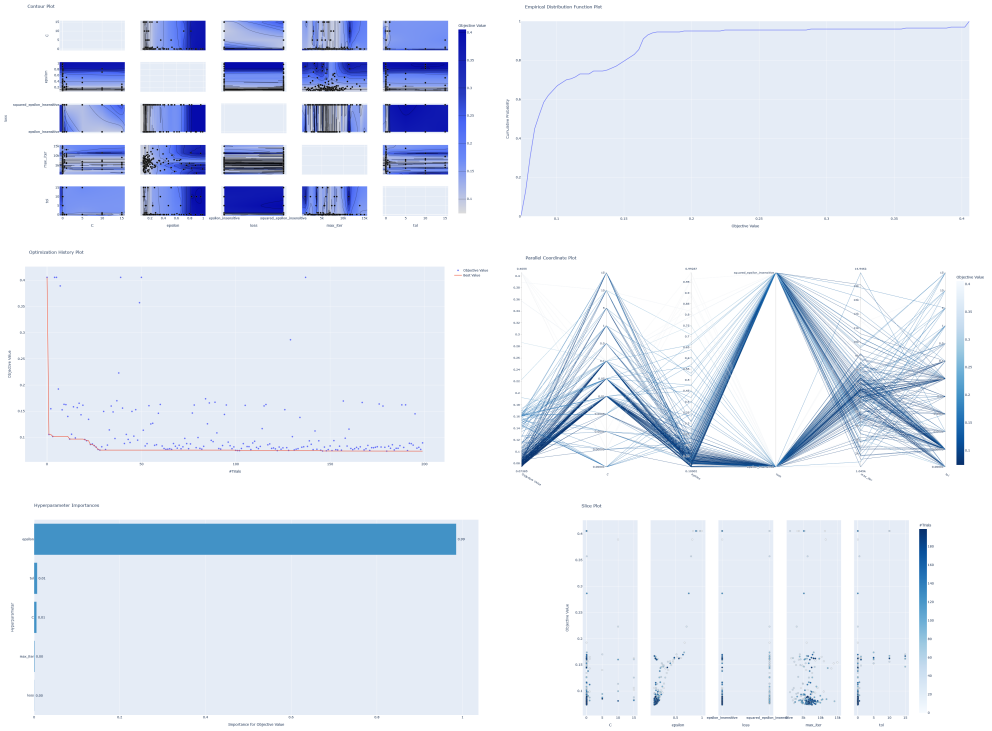
Fonte: Elaborada pelo autor (2022)

Figura 23 – Otimização pelo Optuna - Passive-Aggressive Regressor



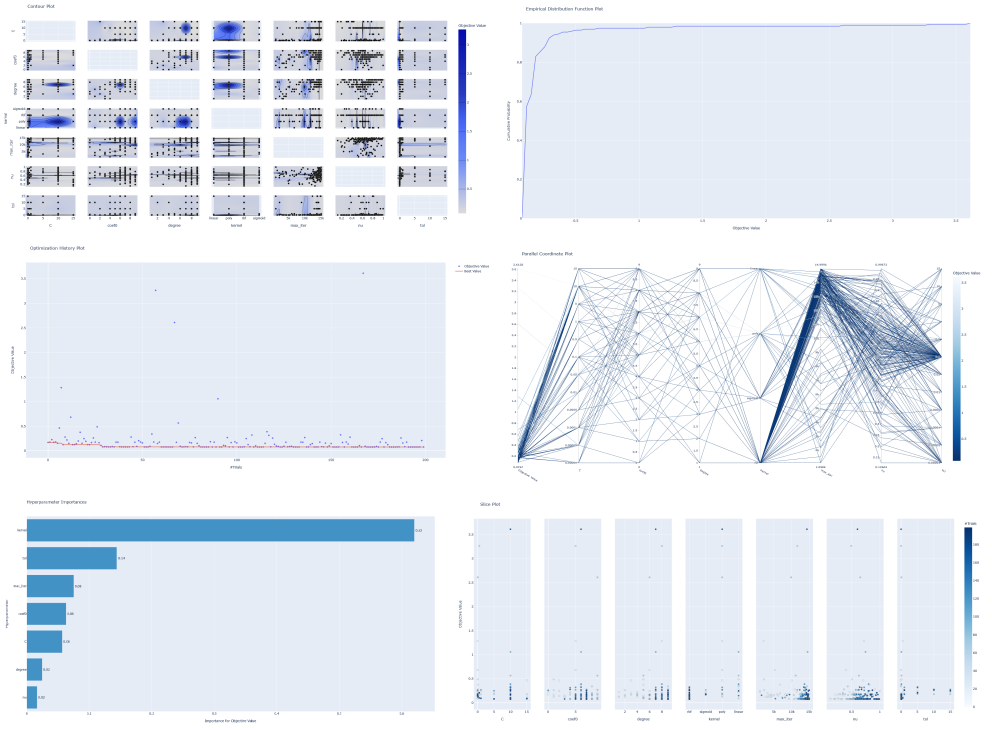
Fonte: Elaborada pelo autor (2022)

Figura 24 – Otimização pelo Optuna - Linear SVR



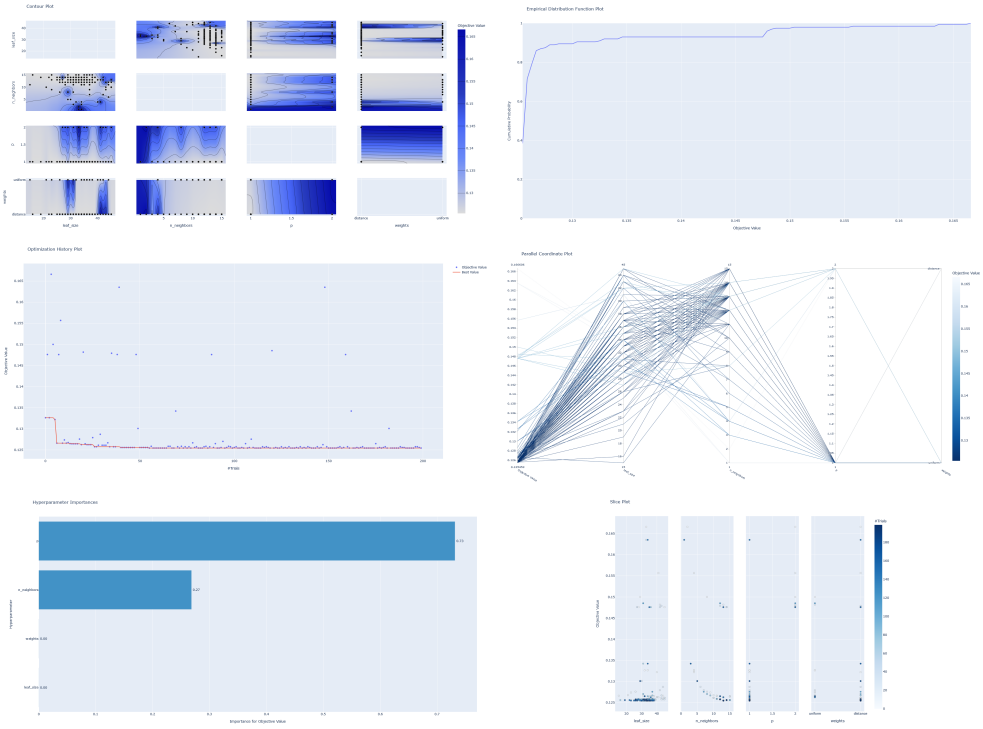
Fonte: Elaborada pelo autor (2022)

Figura 25 – Otimização pelo Optuna - Nu SVR



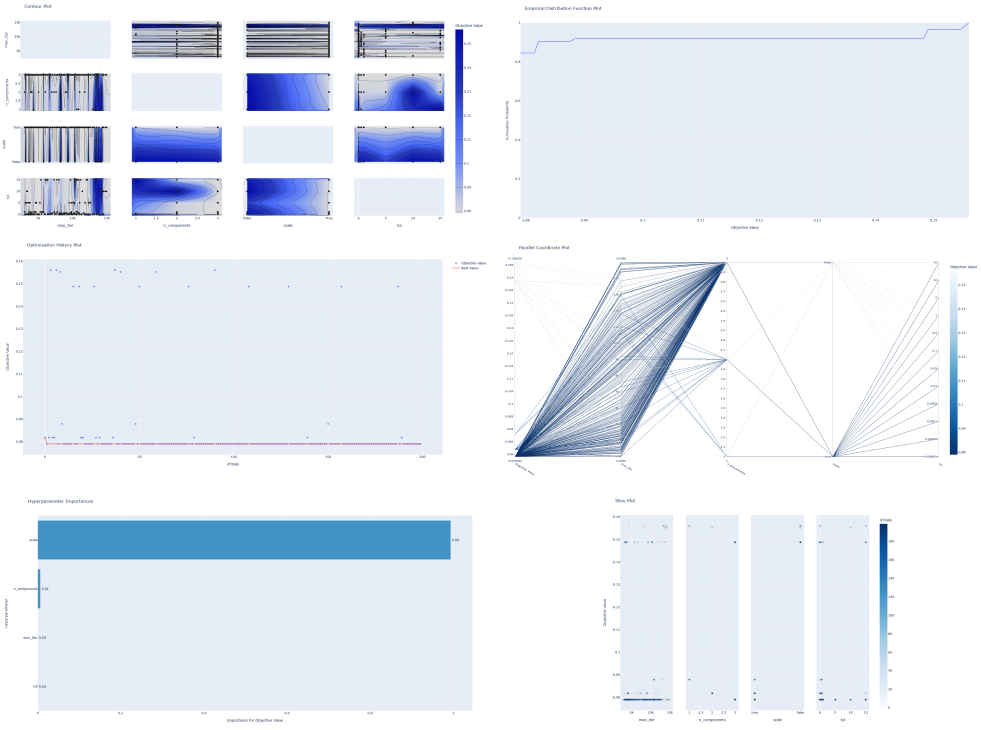
Fonte: Elaborada pelo autor (2022)

Figura 26 – Otimização pelo Optuna - K-Neighbors Regressor



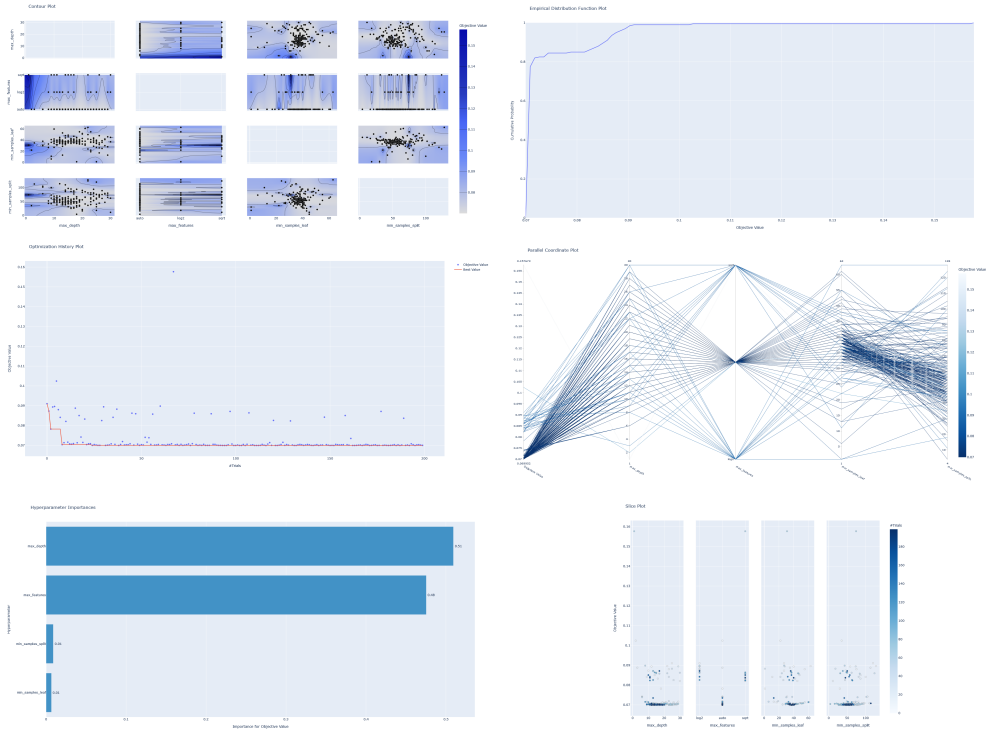
Fonte: Elaborada pelo autor (2022)

Figura 27 – Otimização pelo Optuna - PLS Regression



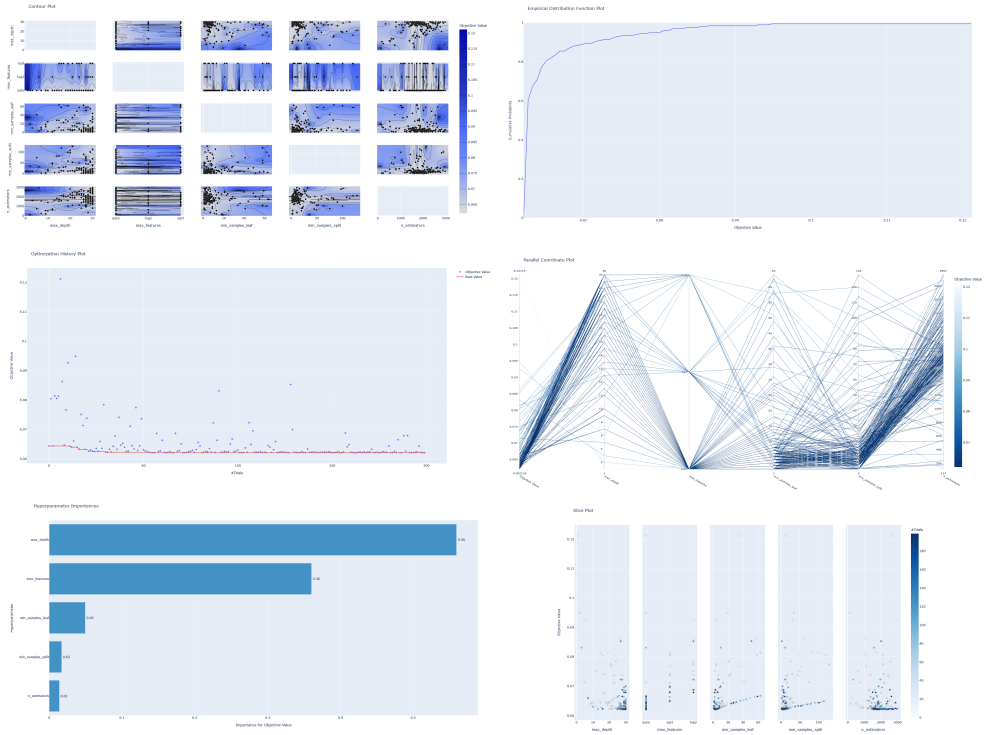
Fonte: Elaborada pelo autor (2022)

Figura 28 – Otimização pelo Optuna - Decision Tree Regressor



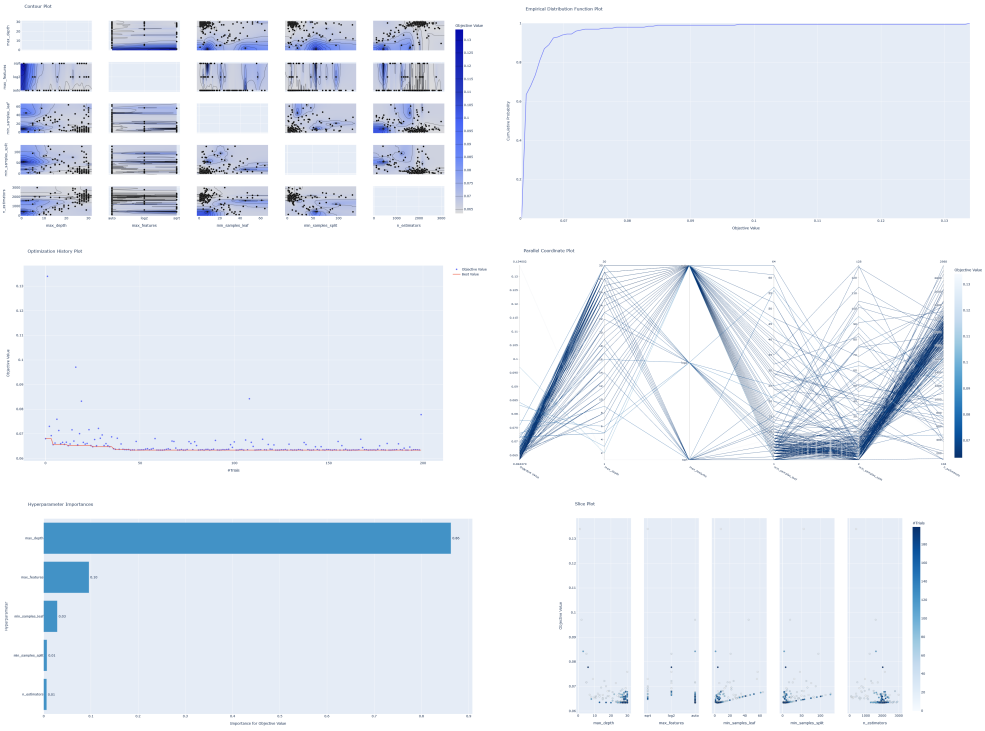
Fonte: Elaborada pelo autor (2022)

Figura 29 – Otimização pelo Optuna - Extra Tree Regressor



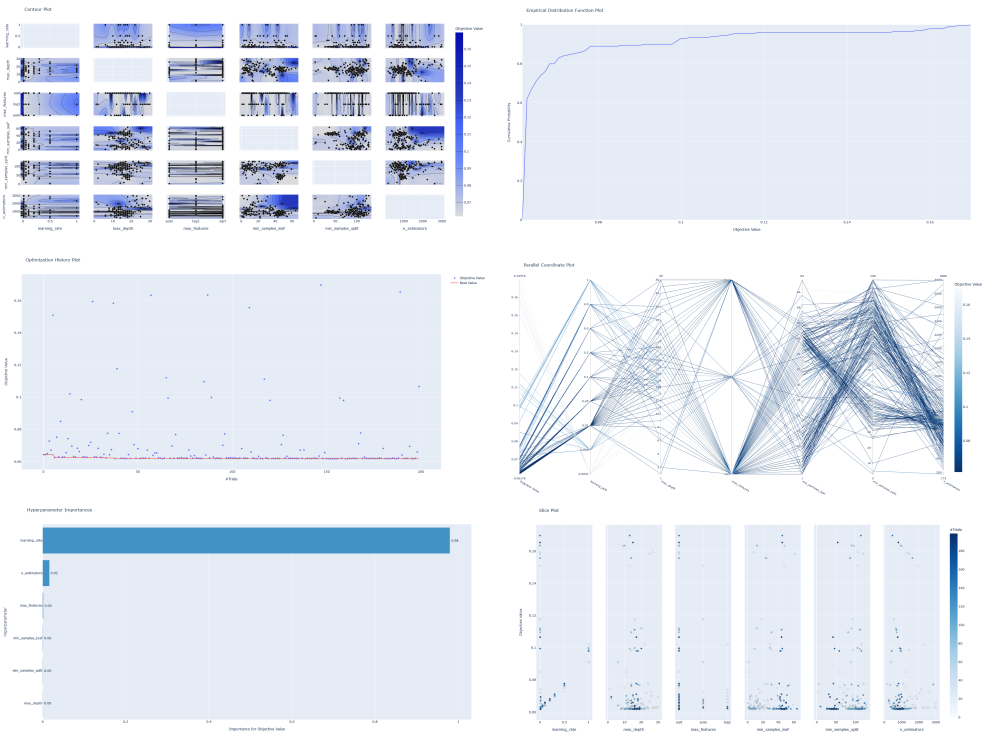
Fonte: Elaborada pelo autor (2022)

Figura 30 – Otimização pelo Optuna - Random Forest Regressor



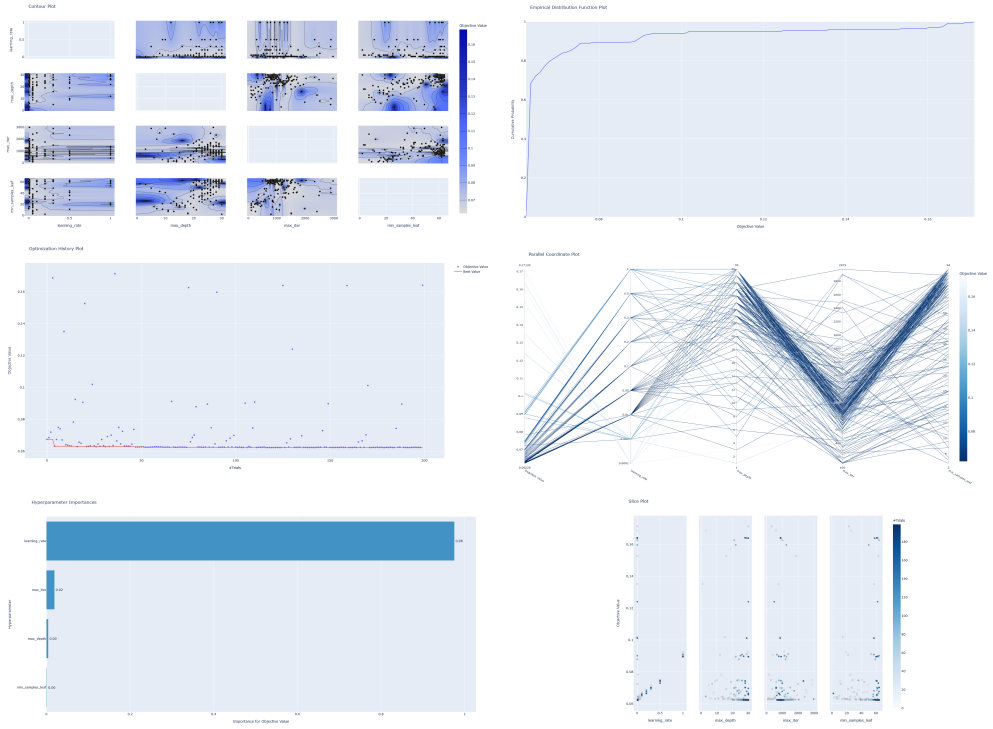
Fonte: Elaborada pelo autor (2022)

Figura 31 – Otimização pelo Optuna - Gradiente Boosting Regressor



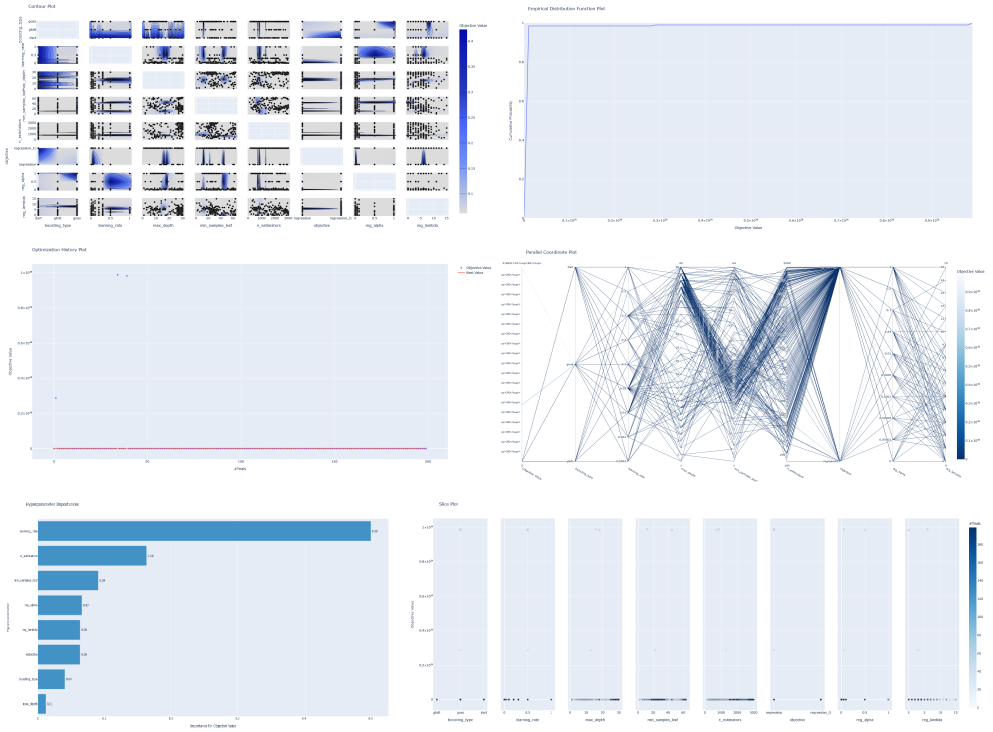
Fonte: Elaborada pelo autor (2022)

Figura 32 – Otimização pelo Optuna - Histogram Gradiente Boosting Regressor



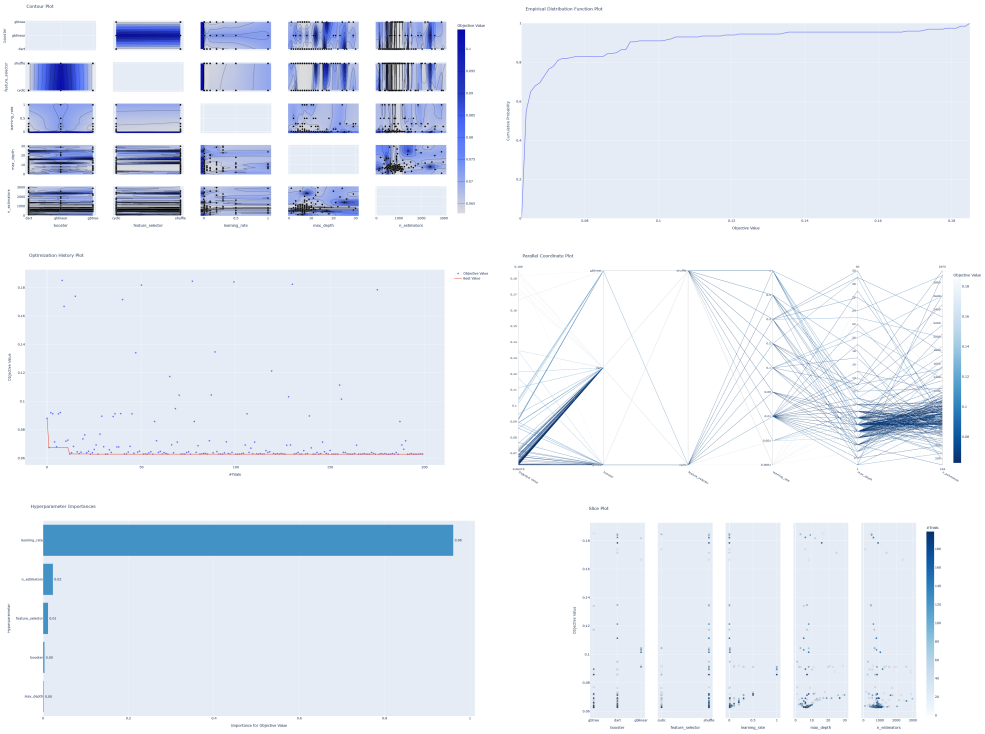
Fonte: Elaborada pelo autor (2022)

Figura 33 – Otimização pelo Optuna - LightGBM Regressor



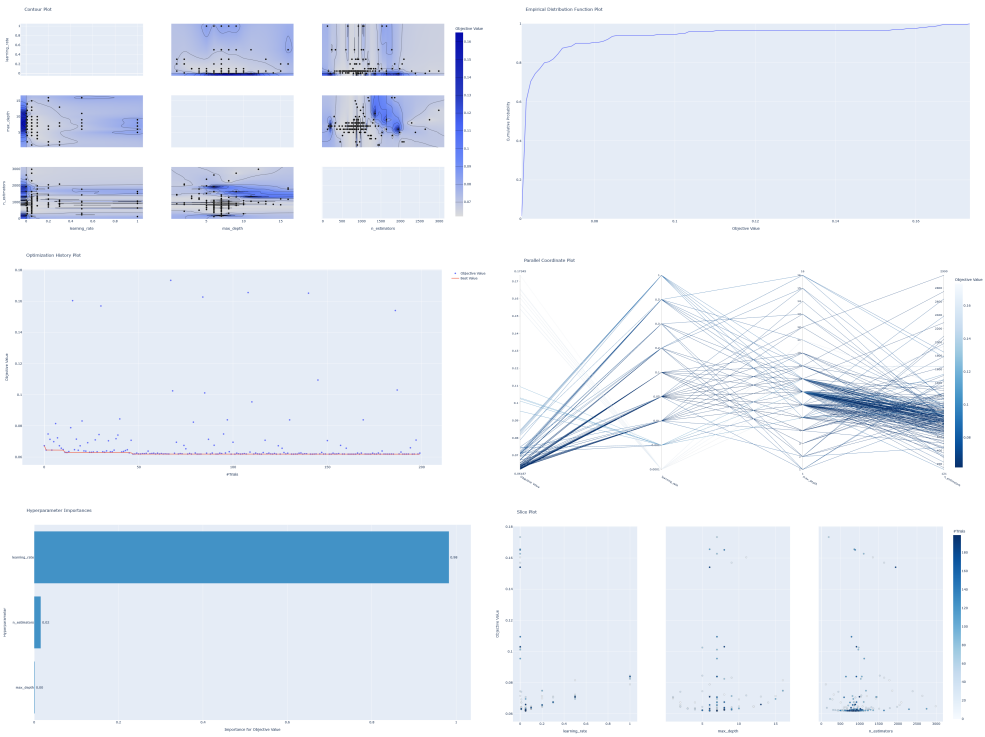
Fonte: Elaborada pelo autor (2022)

Figura 34 – Otimização pelo Optuna - XGBoost Regressor



Fonte: Elaborada pelo autor (2022)

Figura 35 – Otimização pelo Optuna - CatBoost Regressor



Fonte: Elaborada pelo autor (2022)

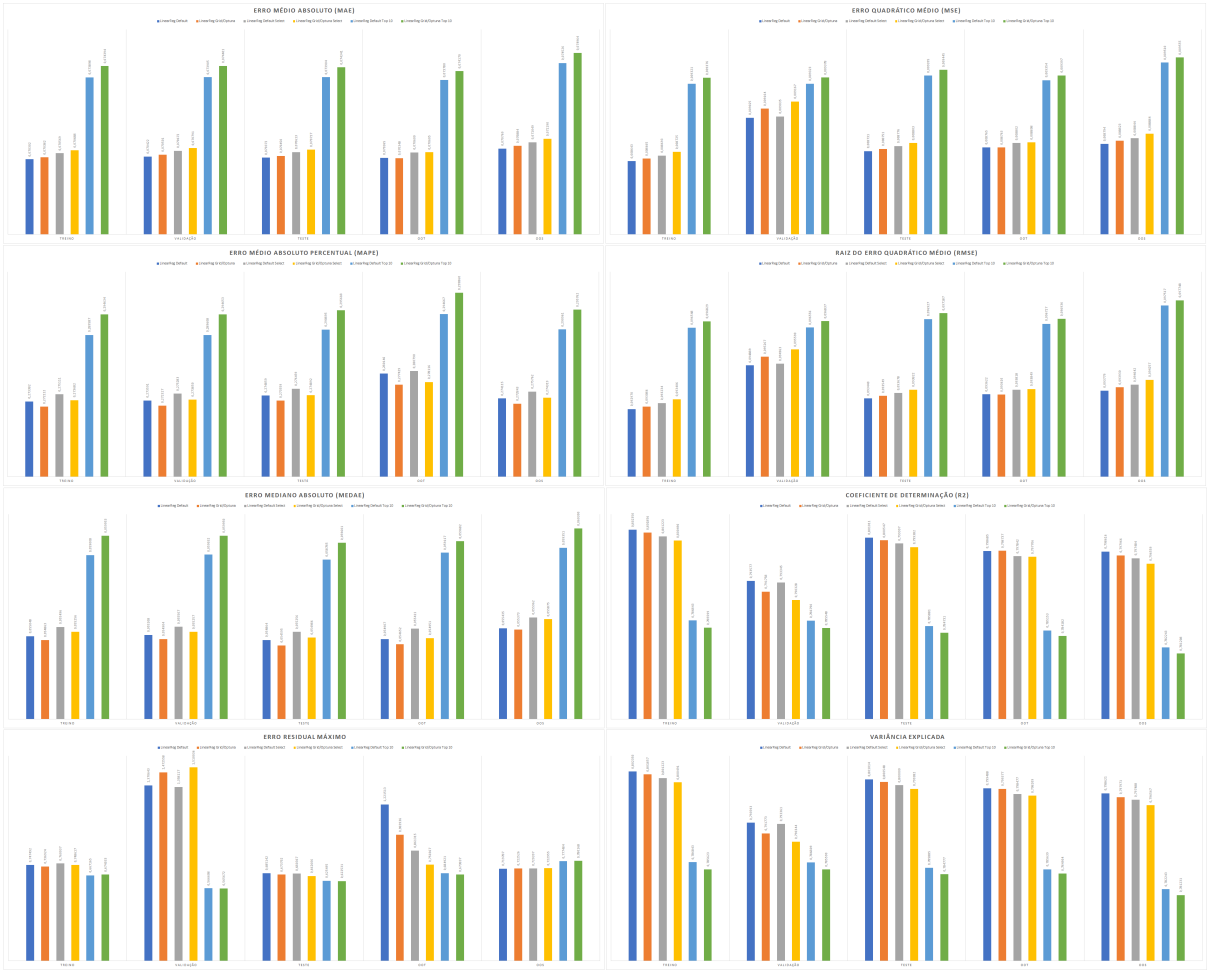
APÊNDICE D – Resultados: Tabelas e Gráficos Complementares

Tabela 25 – Resultado do Tempo de Execução dos modelos de Linear Regression

Modelo	Tempo de Processamento	Representatividade
LinearReg Default	00:00:03,972	25,480%
LinearReg Grid/Optuna	00:00:03,853	24,716%
LinearReg Default Select	00:00:02,810	18,026%
LinearReg Grid/Optuna Select	00:00:02,749	17,634%
LinearReg Default Top 10	00:00:01,156	7,415%
LinearReg Grid/Optuna Top 10	00:00:01,049	6,729%
Total de tempo de processamento	**00:00:15,589**	**100,000%**

Fonte: Elaborada pelo autor (2022)

Figura 36 – Resultado das Métricas dos modelos de Linear Regression

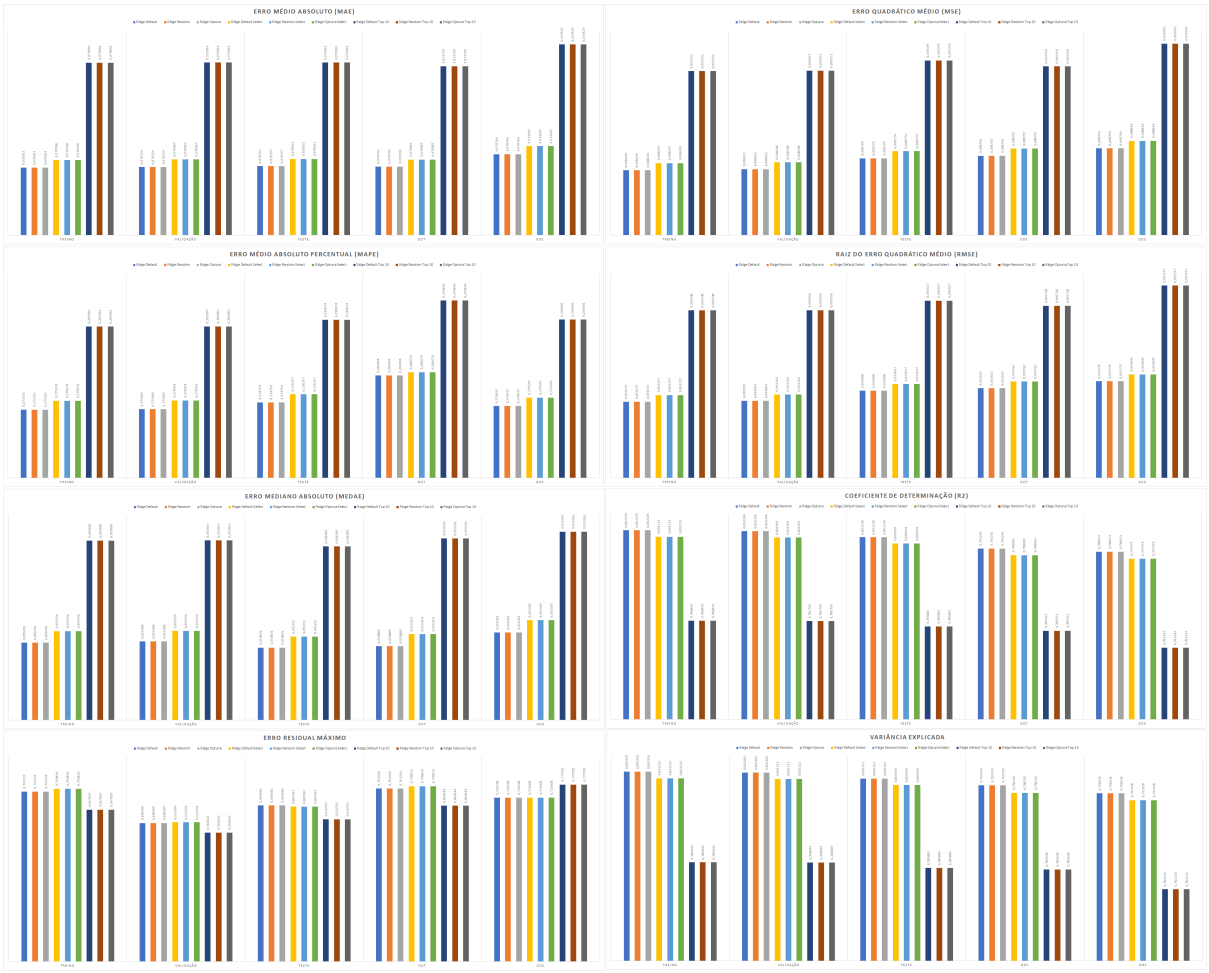


Fonte: Elaborada pelo autor (2022)

Tabela 26 – Resultado do Tempo de Execução dos modelos de Ridge Regression

Modelo	Tempo de Processamento	Representatividade
Ridge Default	00:00:02,592	10,852%
Ridge Random	00:00:06,163	25,804%
Ridge Optuna	00:00:02,569	10,756%
Ridge Default Select	00:00:02,419	10,128%
Ridge Random Select	00:00:04,166	17,443%
Ridge Optuna Select	00:00:01,977	8,278%
Ridge Default Top 10	00:00:00,910	3,810%
Ridge Random Top 10	00:00:02,116	8,859%
Ridge Optuna Top 10	00:00:00,972	4,070%
Total de tempo de processamento	**00:00:23,884**	**100,000%**

Figura 37 – Resultado das Métricas dos modelos de Ridge Regression

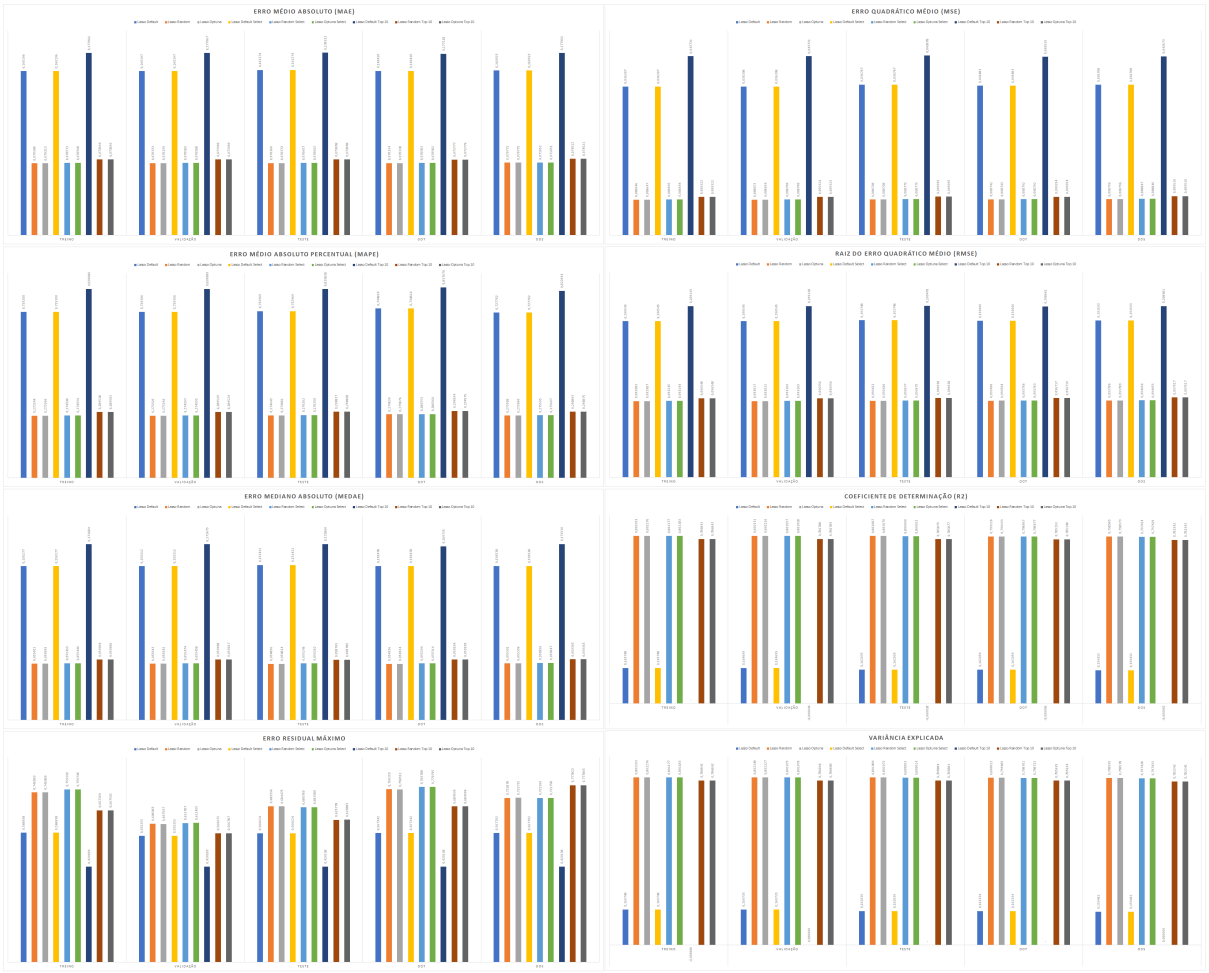


Fonte: Elaborado pelo autor (2022)

Tabela 27 – Resultado do Tempo de Execução dos modelos de Lasso Regression

Modelo	Tempo de Processamento	Representatividade
Lasso Default	00:00:02,306	0,309%
Lasso Random	00:03:28,673	27,940%
Lasso Optuna	00:01:19,345	10,624%
Lasso Default Select	00:00:01,987	0,266%
Lasso Random Select	00:02:18,428	18,535%
Lasso Optuna Select	00:05:10,213	41,535%
Lasso Default Top 10	00:00:01,410	0,189%
Lasso Random Top 10	00:00:02,591	0,347%
Lasso Optuna Top 10	00:00:01,913	0,256%
Total de tempo de processamento	**00:12:26,866**	**100,000%**

Figura 38 – Resultado das Métricas dos modelos de Lasso Regression

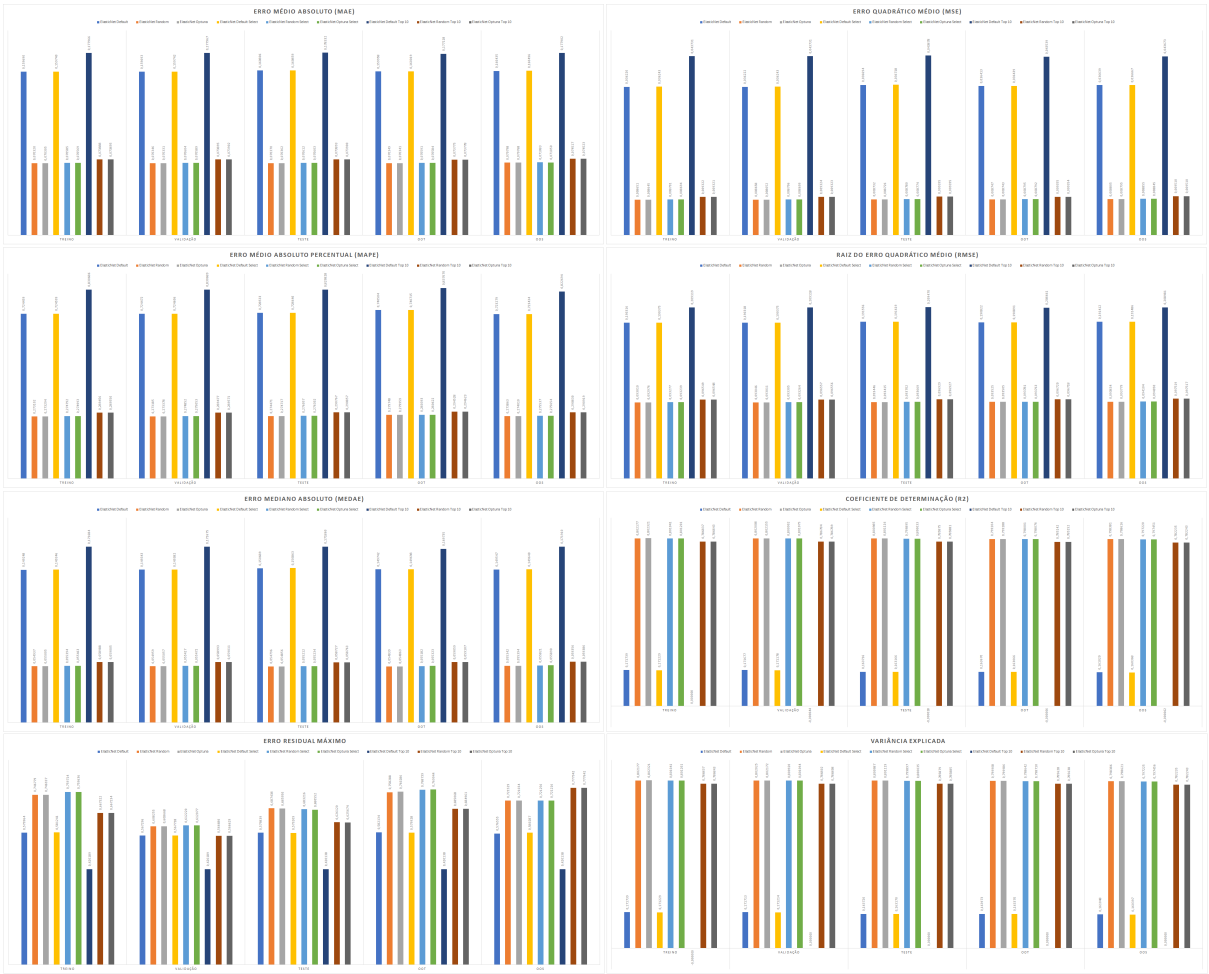


Fonte: Elaborada pelo autor (2022)

Tabela 28 – Resultado do Tempo de Execução dos modelos de Elastic Net Regression

Modelo	Tempo de Processamento	Representatividade
ElasticNet Default	00:00:02,531	0,271%
ElasticNet Random	00:03:36,014	23,131%
ElasticNet Optuna	00:05:23,825	34,675%
ElasticNet Default Select	00:00:02,210	0,237%
ElasticNet Random Select	00:02:16,542	14,621%
ElasticNet Optuna Select	00:03:33,013	22,809%
ElasticNet Default Top 10	00:00:01,573	0,168%
ElasticNet Random Top 10	00:00:35,389	3,789%
ElasticNet Optuna Top 10	00:00:02,795	0,299%
Total de tempo de processamento	**00:15:33,892**	**100,000%**

Figura 39 – Resultado das Métricas dos modelos de Elastic Net Regression

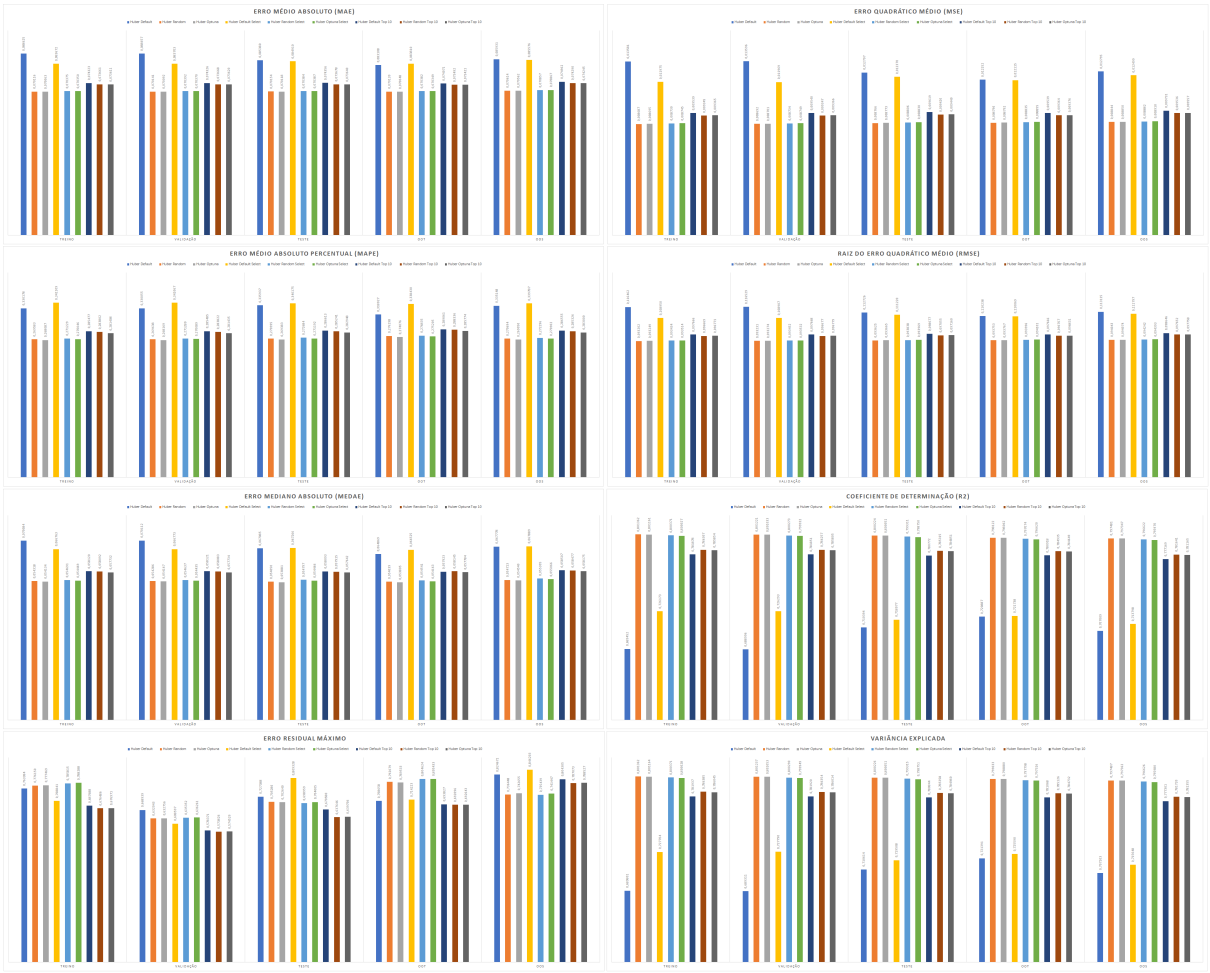


Fonte: Elaborada pelo autor (2022)

Tabela 29 – Resultado do Tempo de Execução dos modelos de Huber Regressor

Modelo	Tempo de Processamento	Representatividade
Huber Default	00:01:10,986	0,309%
Huber Random	01:54:27,778	29,936%
Huber Optuna	01:58:31,022	30,997%
Huber Default Select	00:00:46,707	0,204%
Huber Random Select	01:09:36,628	18,206%
Huber Optuna Select	01:12:00,623	18,833%
Huber Default Top 10	00:00:32,435	0,141%
Huber Random Top 10	00:02:36,884	0,684%
Huber Optuna Top 10	00:02:38,170	0,689%
Total de tempo de processamento	**06:22:21,233**	**100,000%**

Figura 40 – Resultado das Métricas dos modelos de Huber Regressor

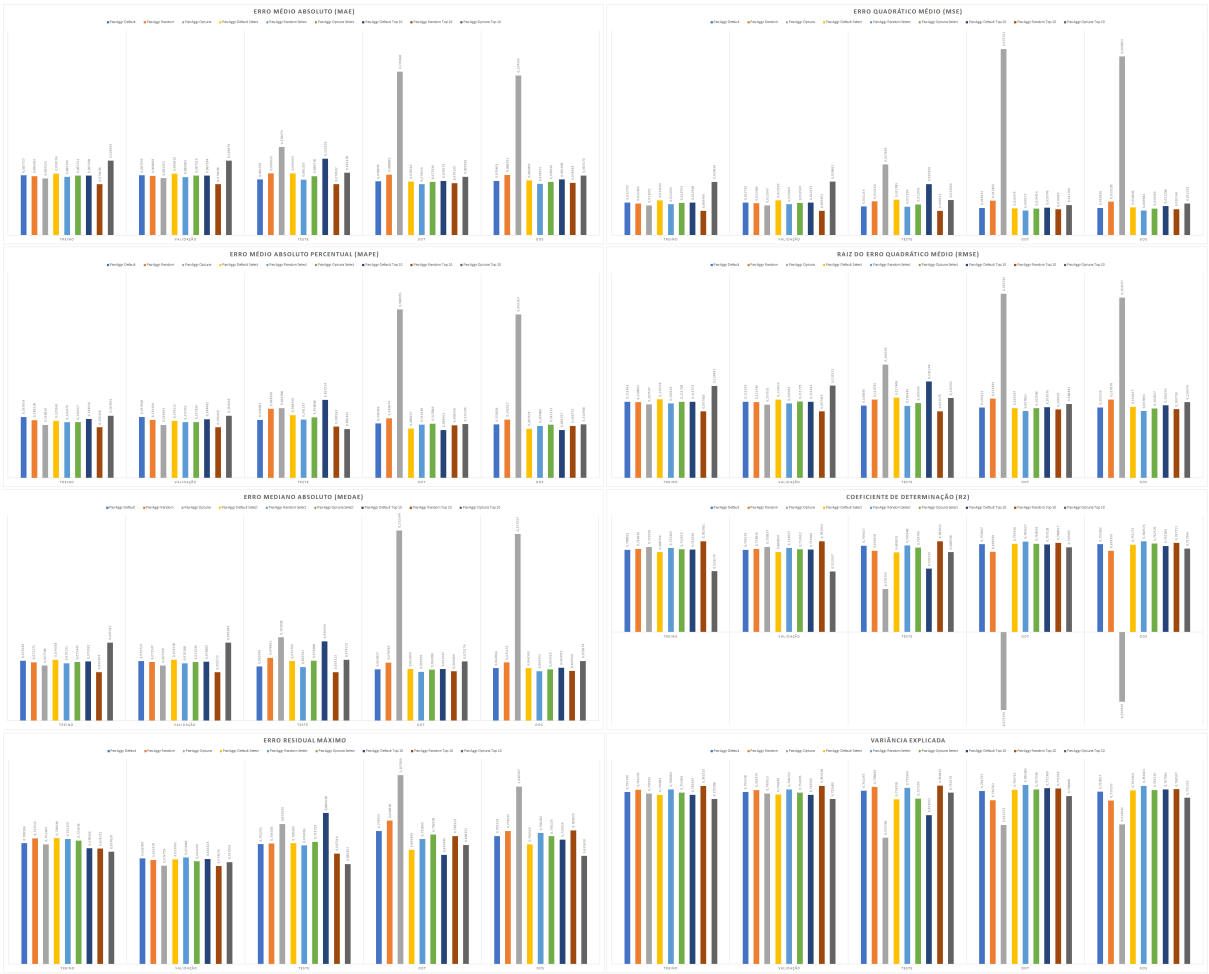


Fonte: Elaborada pelo autor (2022)

Tabela 30 – Resultado do Tempo de Execução dos modelos de Passive-Aggressive Regressor

Modelo	Tempo de Processamento	Representatividade
Pas-Aggr Default	00:00:05,125	6,231%
Pas-Aggr Random	00:00:22,455	27,299%
Pas-Aggr Optuna	00:00:14,632	17,789%
Pas-Aggr Default Select	00:00:04,127	5,017%
Pas-Aggr Random Select	00:00:15,559	18,916%
Pas-Aggr Optuna Select	00:00:09,396	11,423%
Pas-Aggr Default Top 10	00:00:01,957	2,379%
Pas-Aggr Random Top 10	00:00:06,907	8,397%
Pas-Aggr Optuna Top 10	00:00:02,097	2,549%
Total de tempo de processamento	**00:01:22,255**	**100,000%**

Figura 41 – Resultado das Métricas dos modelos de Passive-Aggressive Regressor

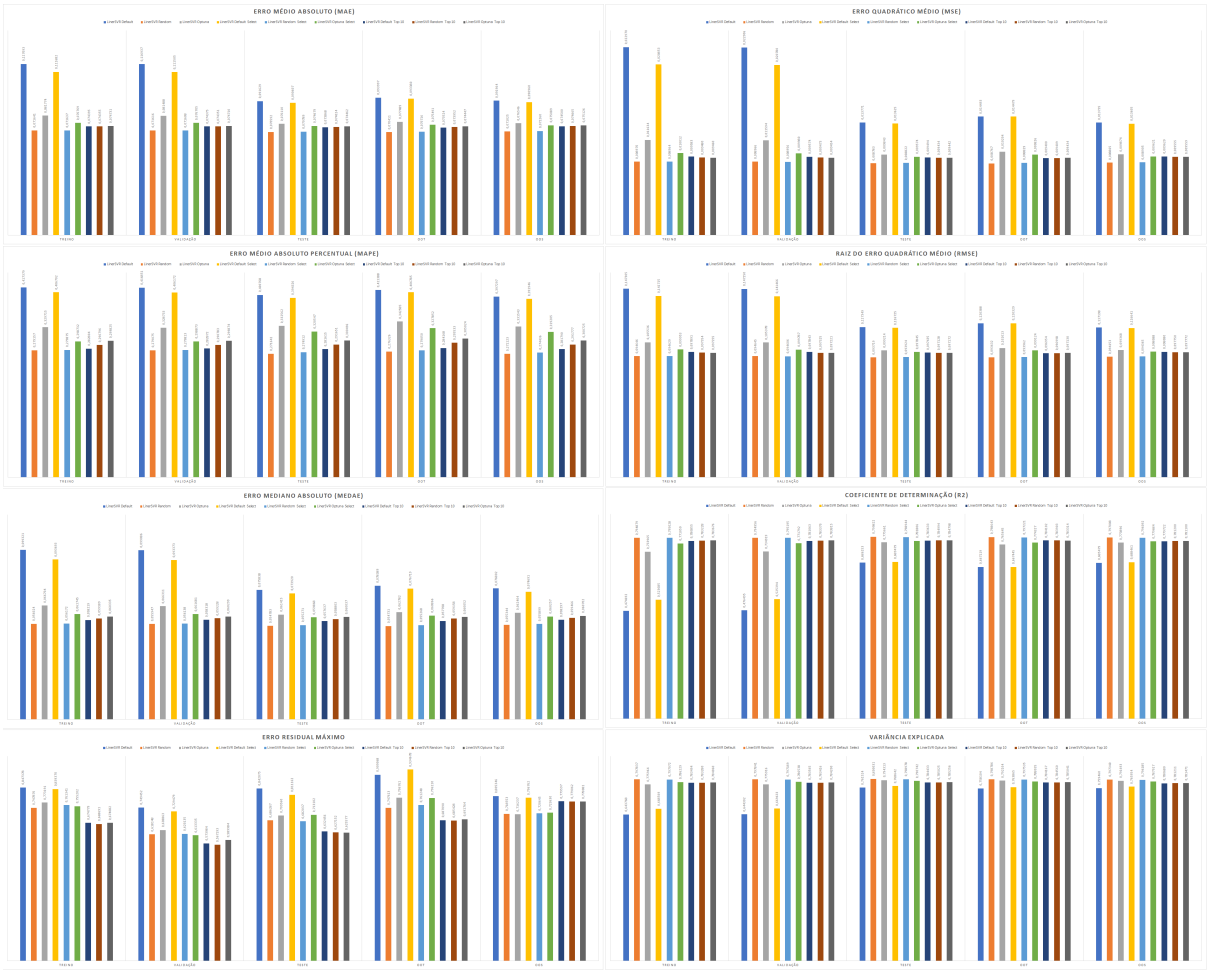


Fonte: Elaborada pelo autor (2022)

Tabela 31 – Resultado do Tempo de Execução dos modelos de Linear SVR

Modelo	Tempo de Processamento	Representatividade
LinerSVR Default	00:15:33,478	2,219%
LinerSVR Random	03:09:03,549	26,961%
LinerSVR Optuna	01:29:14,602	12,726%
LinerSVR Default Select	00:12:19,700	1,758%
LinerSVR Random Select	02:50:21,658	24,294%
LinerSVR Optuna Select	01:22:48,250	11,808%
LinerSVR Default Top 10	00:06:18,082	0,899%
LinerSVR Random Top 10	01:36:52,922	13,816%
LinerSVR Optuna Top 10	00:38:42,420	5,520%
Total de tempo de processamento	**11:41:14,661**	**100,000%**

Figura 42 – Resultado das Métricas dos modelos de Linear SVR

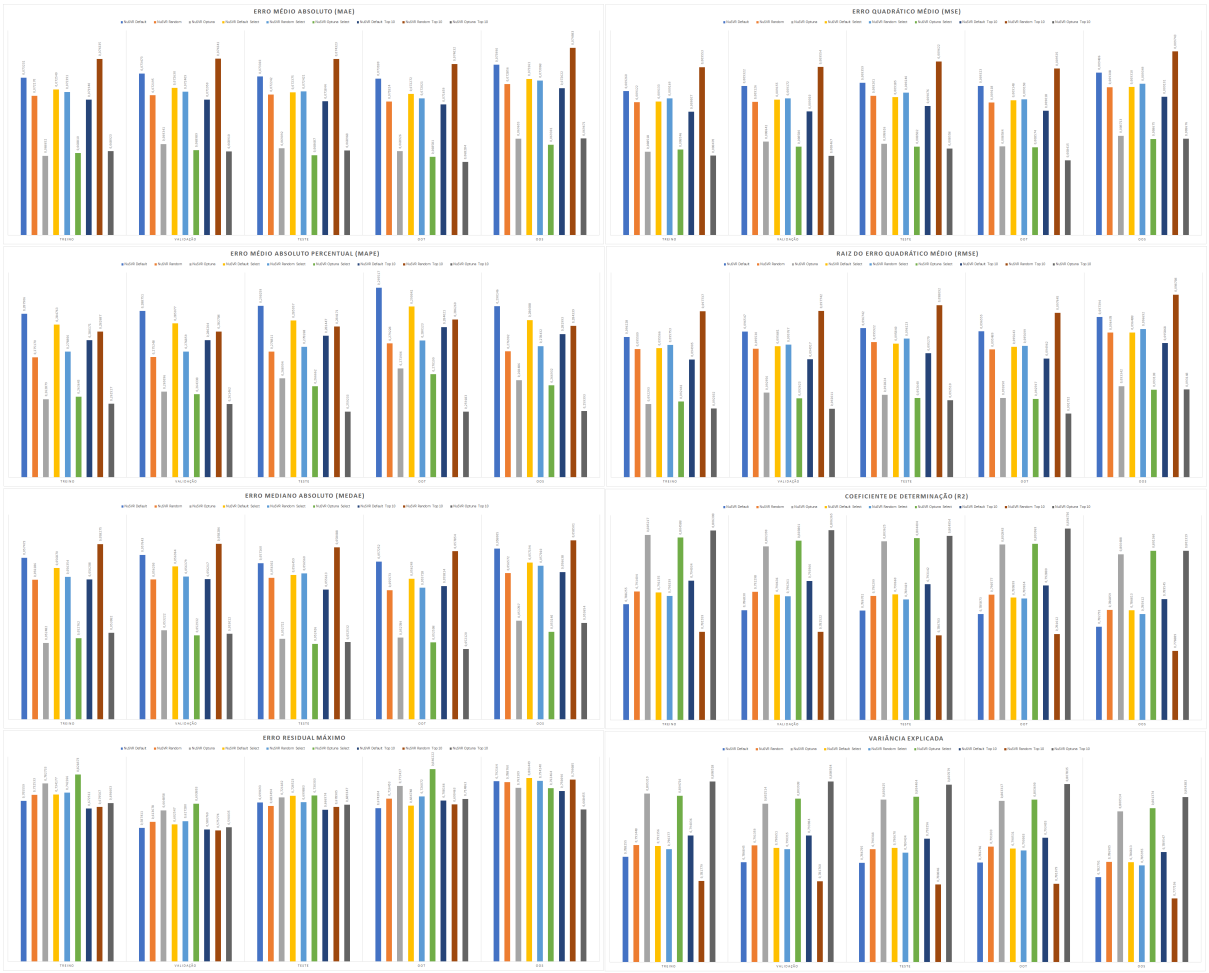


Fonte: Elaborada pelo autor (2022)

Tabela 32 – Resultado do Tempo de Execução dos modelos de Nu SVR

Modelo	Tempo de Processamento	Representatividade
NuSVR Default	00:33:14,655	9,213%
NuSVR Random	00:59:44,005	16,554%
NuSVR Optuna	01:01:44,292	17,110%
NuSVR Default Select	00:31:06,890	8,623%
NuSVR Random Select	00:49:31,308	13,724%
NuSVR Optuna Select	00:54:13,918	15,030%
NuSVR Default Top 10	00:17:16,089	4,786%
NuSVR Random Top 10	00:26:43,617	7,407%
NuSVR Optuna Top 10	00:27:15,064	7,552%
Total de tempo de processamento	**06:00:49,838**	**100,000%**

Figura 43 – Resultado das Métricas dos modelos de Nu SVR

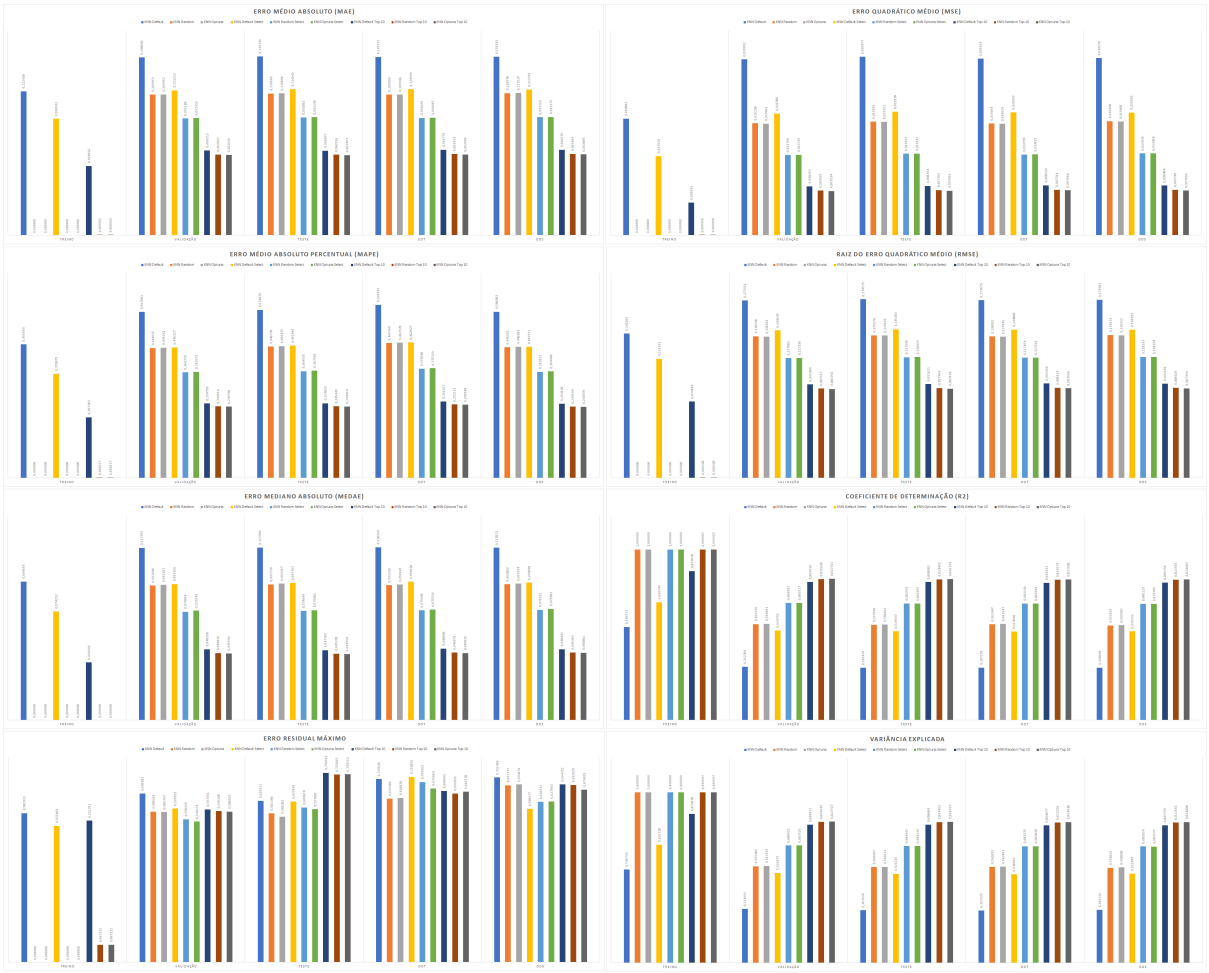


Fonte: Elaborada pelo autor (2022)

Tabela 33 – Resultado do Tempo de Execução dos modelos de K-Neighbors Regressor (KNN)

Modelo	Tempo de Processamento	Representatividade
KNN Default	01:04:46,786	6,286%
KNN Random	04:15:37,860	24,807%
KNN Optuna	04:14:45,263	24,722%
KNN Default Select	01:05:18,738	6,338%
KNN Random Select	03:07:00,556	18,148%
KNN Optuna Select	03:06:03,398	18,056%
KNN Default Top 10	00:03:29,761	0,339%
KNN Random Top 10	00:06:28,608	0,629%
KNN Optuna Top 10	00:06:56,747	0,674%
Total de tempo de processamento	**17:10:27,717**	**100,000%**

Figura 44 – Resultado das Métricas dos modelos de K-Neighbors Regressor (KNN)

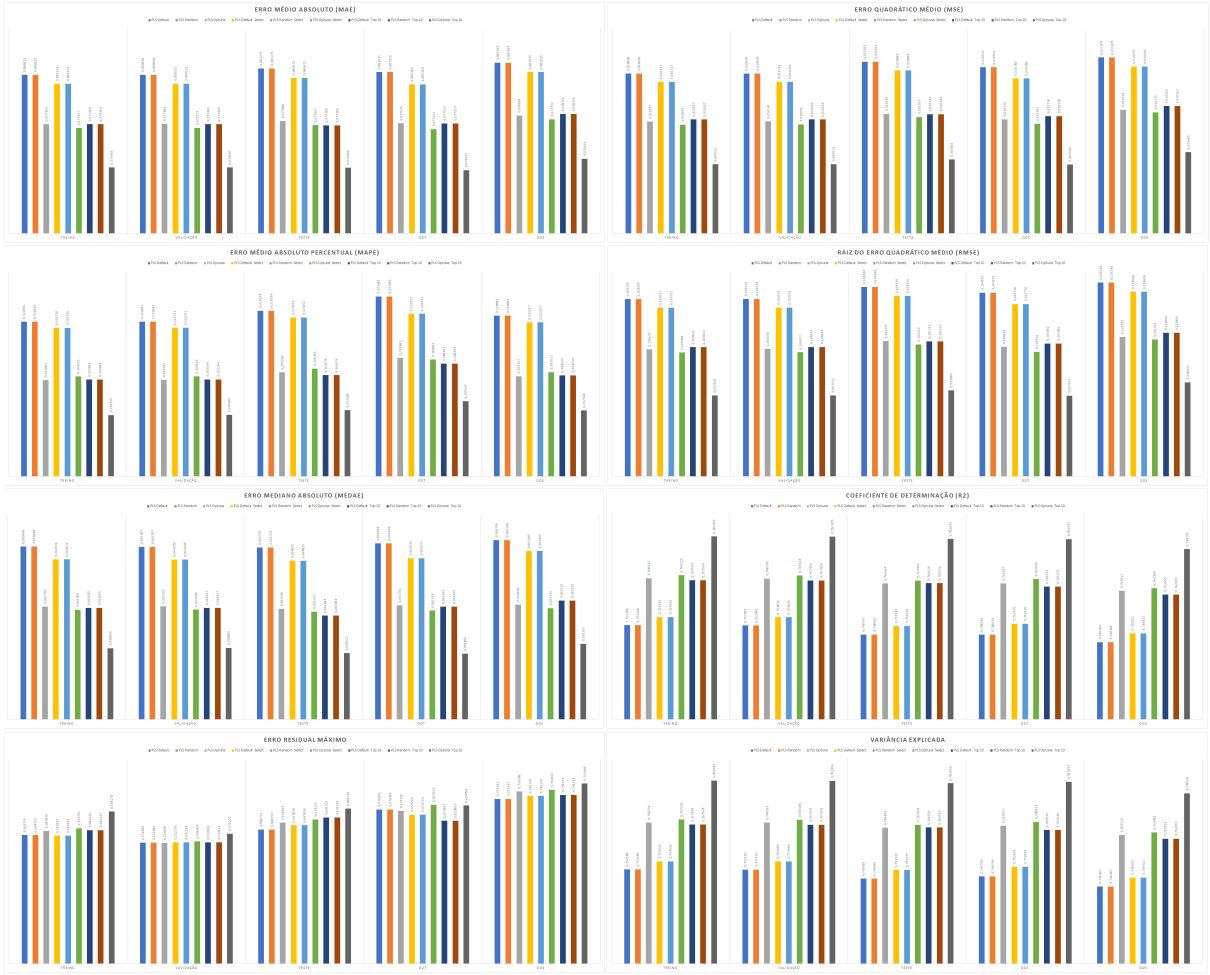


Fonte: Elaborada pelo autor (2022)

Tabela 34 – Resultado do Tempo de Execução dos modelos de PLS Regression (PLS)

Modelo	Tempo de Processamento	Representatividade
PLS Default	00:00:03,644	14,443%
PLS Random	00:00:03,578	14,181%
PLS Optuna	00:00:04,306	17,066%
PLS Default Select	00:00:02,931	11,617%
PLS Random Select	00:00:02,915	11,553%
PLS Optuna Select	00:00:03,287	13,028%
PLS Default Top 10	00:00:01,487	5,894%
PLS Random Top 10	00:00:01,516	6,008%
PLS Optuna Top 10	00:00:01,567	6,211%
Total de tempo de processamento	**00:00:25,231**	**100,000%**

Figura 45 – Resultado das Métricas dos modelos de PLS Regression (PLS)

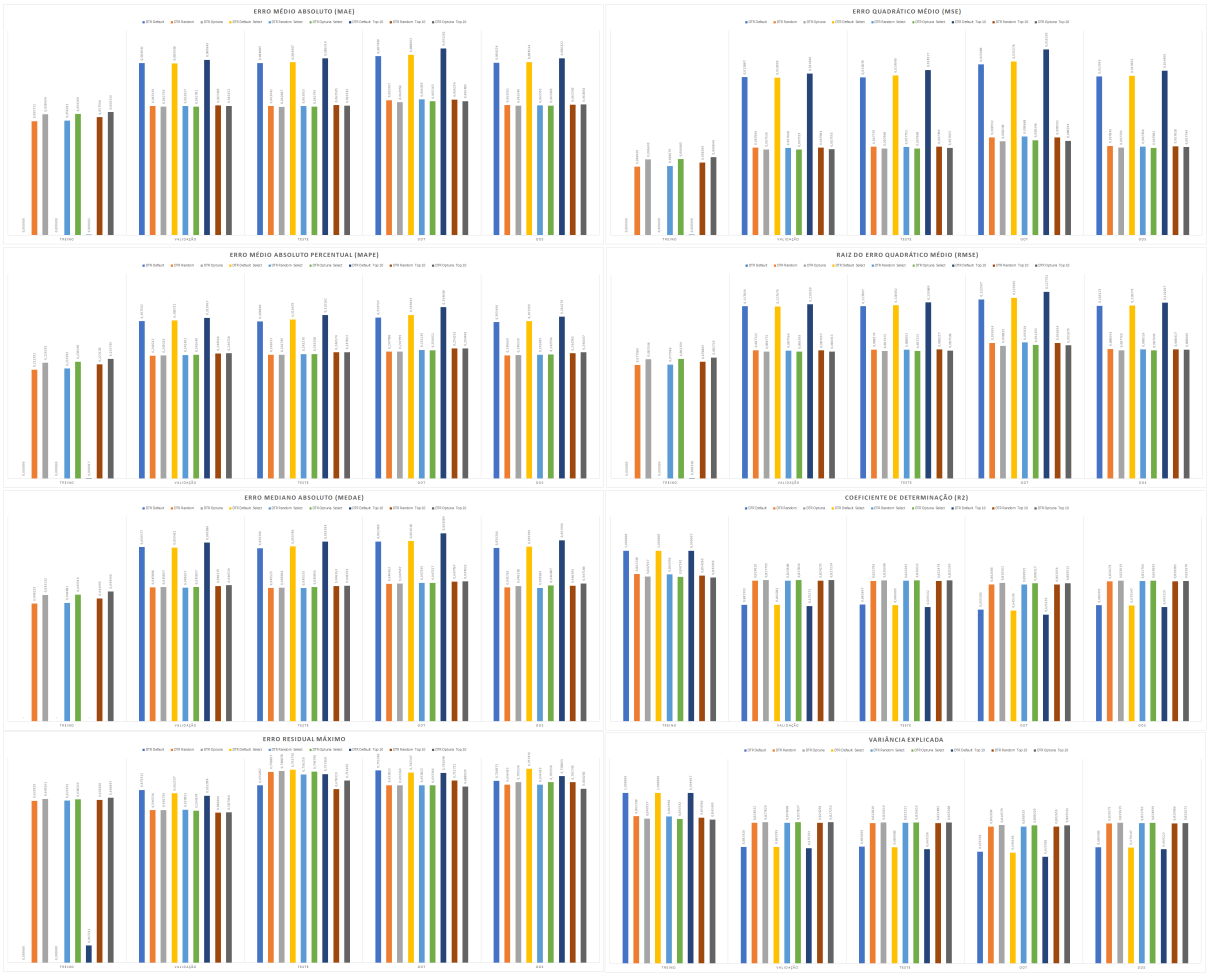


Fonte: Elaborada pelo autor (2022)

Tabela 35 – Resultado do Tempo de Execução dos modelos de Decision Tree Regressor (DTR)

Modelo	Tempo de Processamento	Representatividade
DTR Default	00:01:19,390	22,366%
DTR Random	00:00:48,979	13,799%
DTR Optuna	00:00:43,038	12,125%
DTR Default Select	00:01:00,595	17,071%
DTR Random Select	00:00:38,915	10,963%
DTR Optuna Select	00:00:34,194	9,633%
DTR Default Top 10	00:00:22,885	6,447%
DTR Random Top 10	00:00:14,343	4,041%
DTR Optuna Top 10	00:00:12,617	3,555%
Total de tempo de processamento	**00:05:54,956**	**100,000%**

Figura 46 – Resultado das Métricas dos modelos de Decision Tree Regressor (DTR)

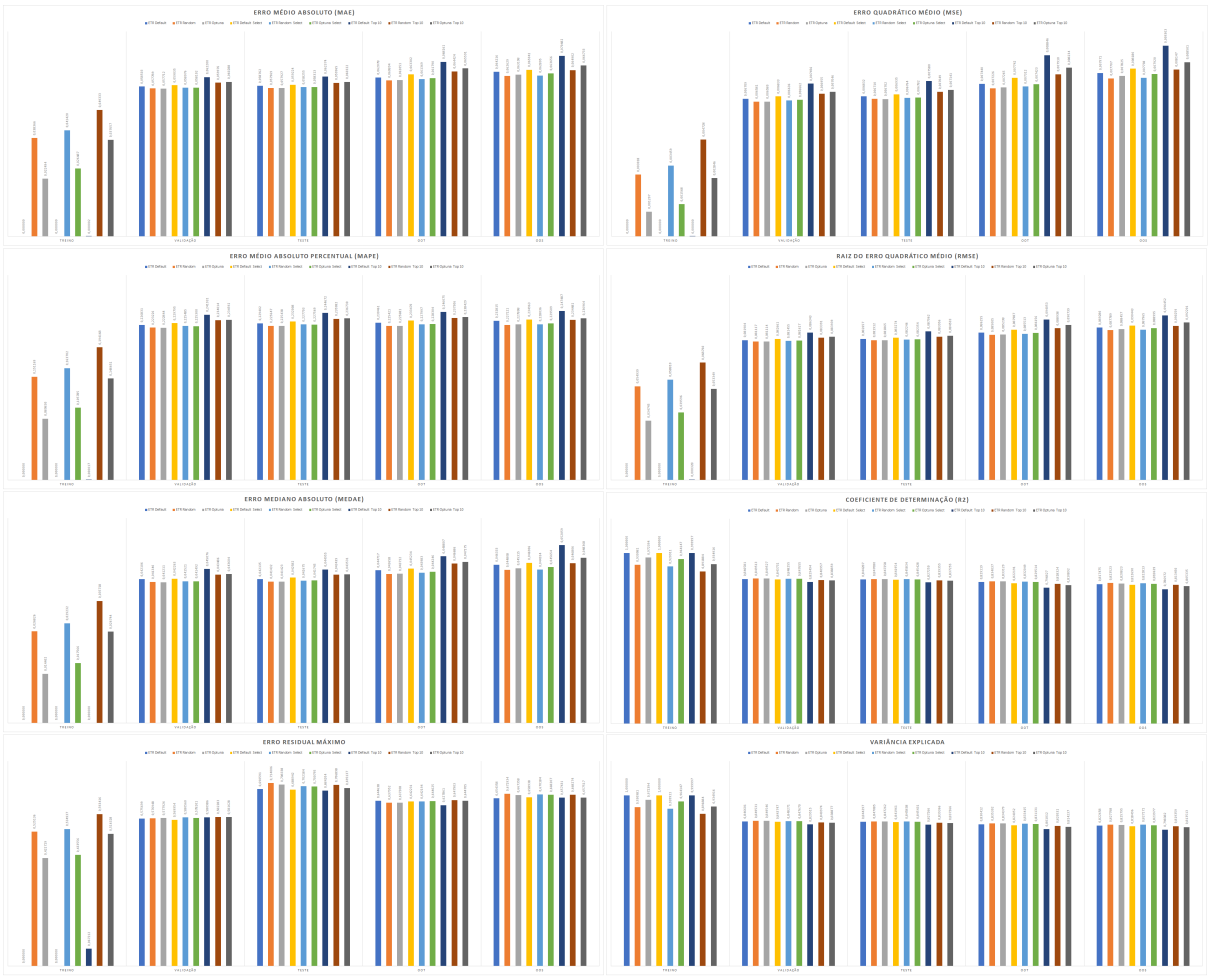


Fonte: Elaborada pelo autor (2022)

Tabela 36 – Resultado do Tempo de Execução dos modelos de Extra Tree Regressor (ETR)

Modelo	Tempo de Processamento	Representatividade
ETR Default	00:37:35,719	3,099%
ETR Random	01:22:45,381	6,821%
ETR Optuna	08:44:31,269	43,233%
ETR Default Select	00:27:23,730	2,258%
ETR Random Select	00:56:08,203	4,627%
ETR Optuna Select	05:53:34,352	29,143%
ETR Default Top 10	00:09:18,291	0,767%
ETR Random Top 10	00:14:45,009	1,216%
ETR Optuna Top 10	01:47:12,271	8,836%
Total de tempo de processamento	**20:13:14,225**	**100,000%**

Figura 47 – Resultado das Métricas dos modelos de Extra Tree Regressor (ETR)

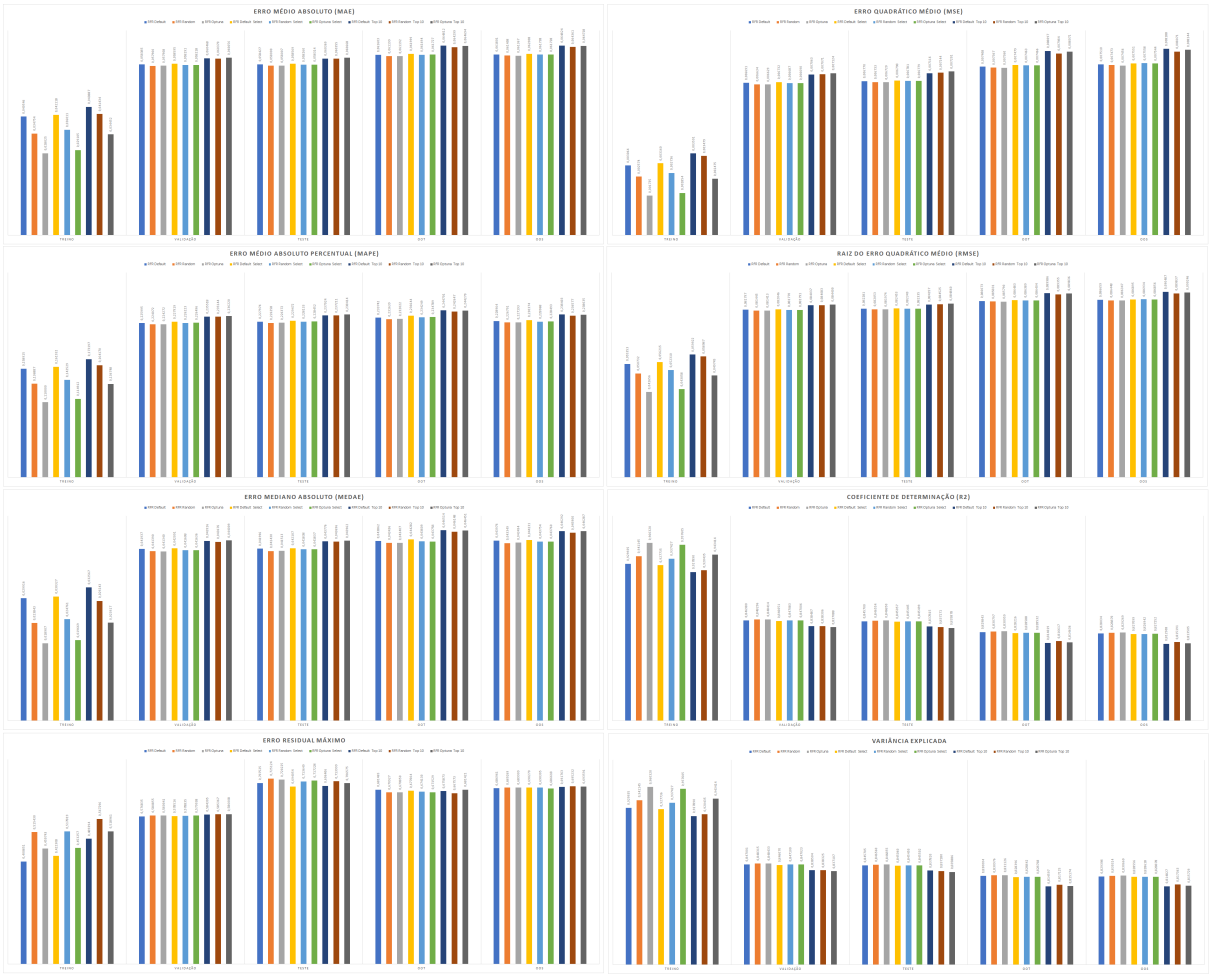


Fonte: Elaborada pelo autor (2022)

Tabela 37 – Resultado do Tempo de Execução dos modelos de Random Forest Regressor (RFR)

Modelo	Tempo de Processamento	Representatividade
RFR Default	00:04:55,622	0,741%
RFR Random	00:20:14,673	3,046%
RFR Optuna	03:29:29,706	31,517%
RFR Default Select	00:05:15,748	0,792%
RFR Random Select	00:21:16,559	3,201%
RFR Optuna Select	03:28:01,230	31,295%
RFR Default Top 10	00:04:48,491	0,723%
RFR Random Top 10	00:19:40,522	2,960%
RFR Optuna Top 10	02:50:59,751	25,725%
Total de tempo de processamento	**11:04:42,302**	**100,000%**

Figura 48 – Resultado das Métricas dos modelos de Random Forest Regressor (RFR)

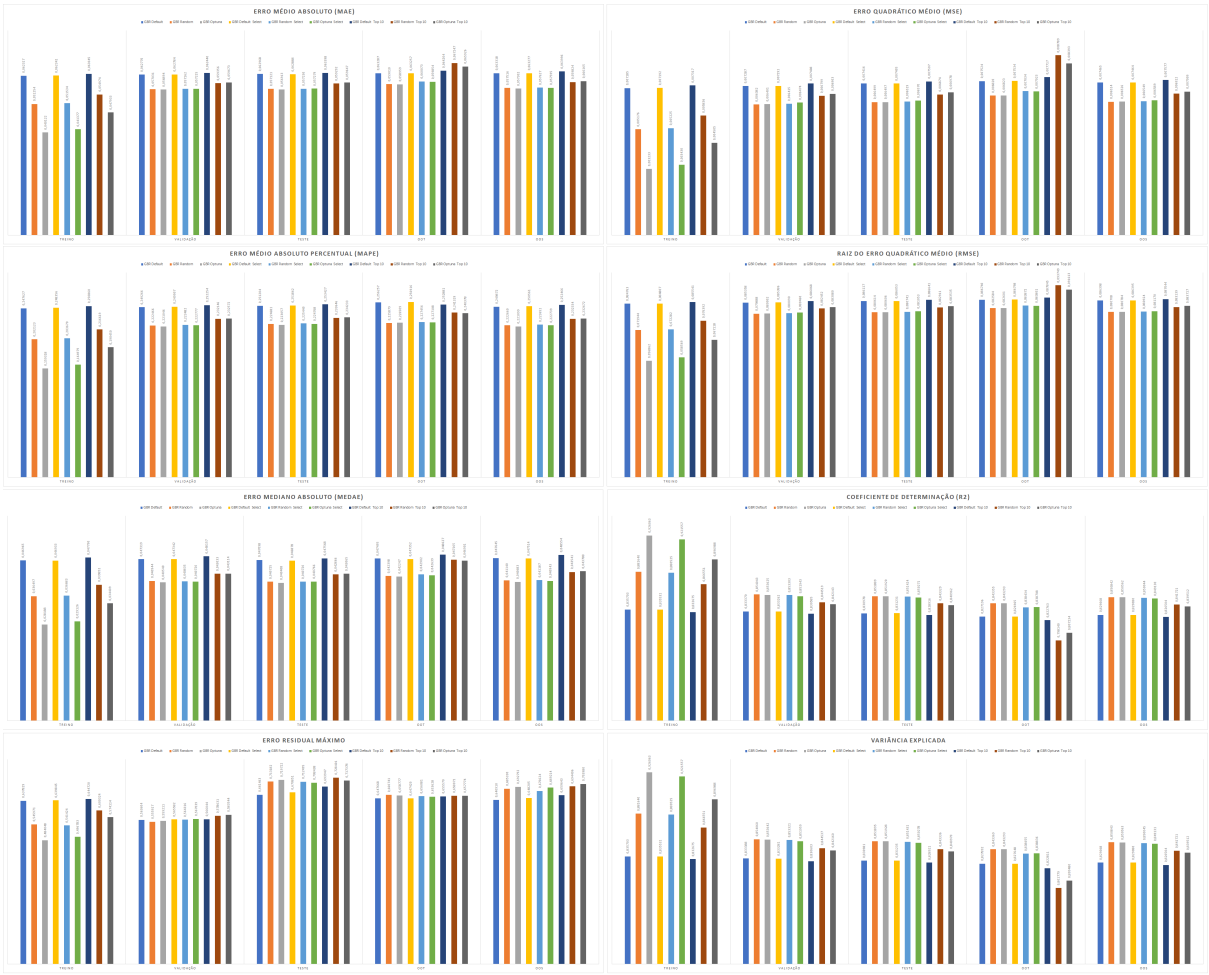


Fonte: Elaborada pelo autor (2022)

Tabela 38 – Resultado do Tempo de Execução dos modelos de Gradiente Boosting Regressor (GBR)

Modelo	Tempo de Processamento	Representatividade
GBR Default	00:27:00,300	3,857%
GBR Random	02:03:12,580	17,596%
GBR Optuna	01:55:17,655	16,466%
GBR Default Select	00:20:52,157	2,980%
GBR Random Select	02:01:39,603	17,375%
GBR Optuna Select	01:53:29,615	16,209%
GBR Default Top 10	00:07:25,404	1,060%
GBR Random Top 10	01:26:11,005	12,308%
GBR Optuna Top 10	01:25:03,911	12,149%
Total de tempo de processamento	**11:40:12,230**	**100,000%**

Figura 49 – Resultado das Métricas dos modelos de Gradiente Boosting Regressor (GBR)

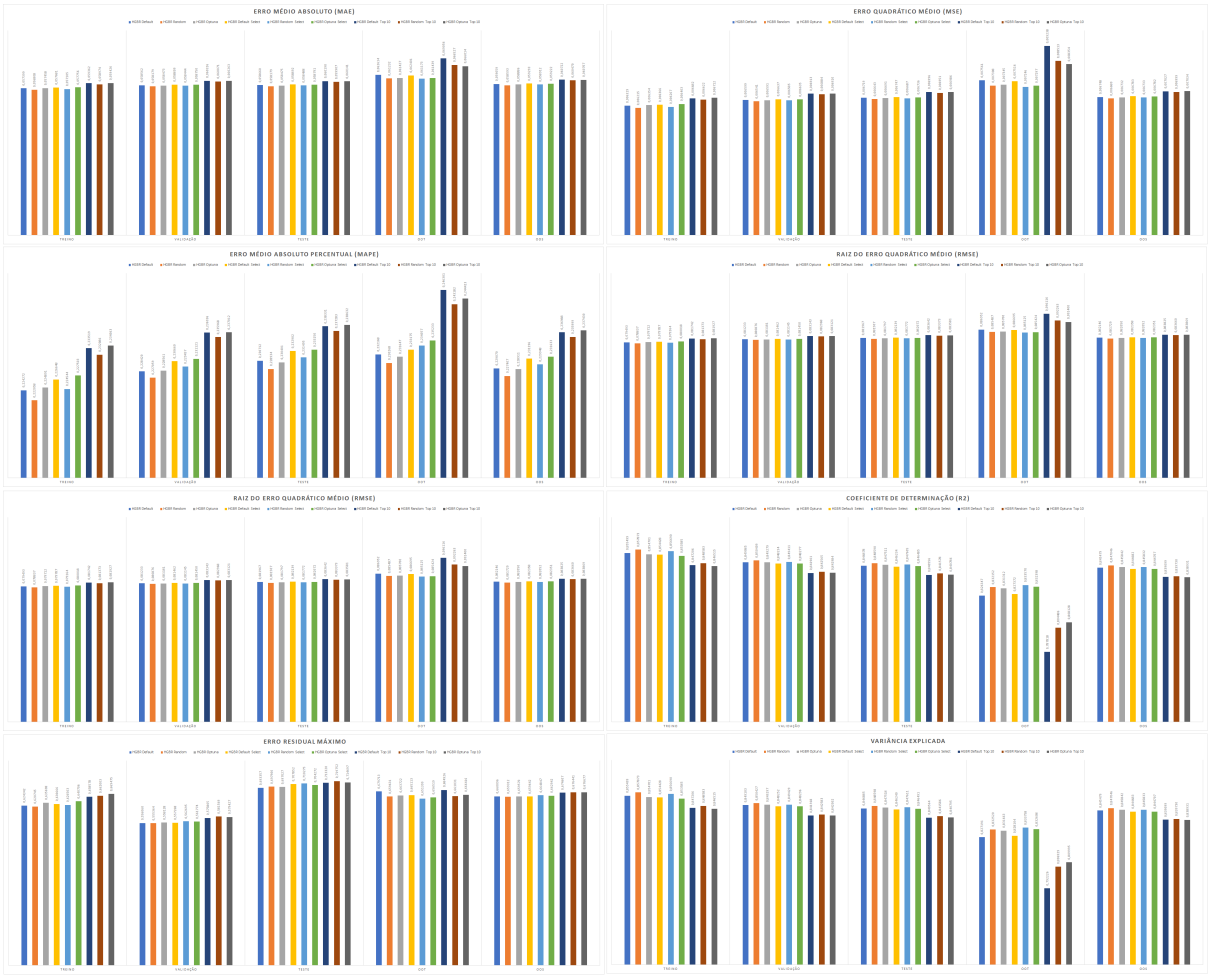


Fonte: Elaborada pelo autor (2022)

Tabela 39 – Resultado do Tempo de Execução dos modelos de Histogram Gradiente Boosting Regressor (HGBR)

Modelo	Tempo de Processamento	Representatividade
HGBR Default	00:00:40,393	2,594%
HGBR Random	00:06:10,378	23,787%
HGBR Optuna	00:04:51,363	18,713%
HGBR Default Select	00:00:29,315	1,883%
HGBR Random Select	00:04:32,960	17,531%
HGBR Optuna Select	00:03:30,152	13,497%
HGBR Default Top 10	00:00:17,643	1,133%
HGBR Random Top 10	00:03:03,965	11,815%
HGBR Optuna Top 10	00:02:20,872	9,047%
Total de tempo de processamento	**00:25:57,041**	**100,000%**

Figura 50 – Resultado das Métricas dos modelos de Histogram Gradiente Boosting Regressor (HGBR)

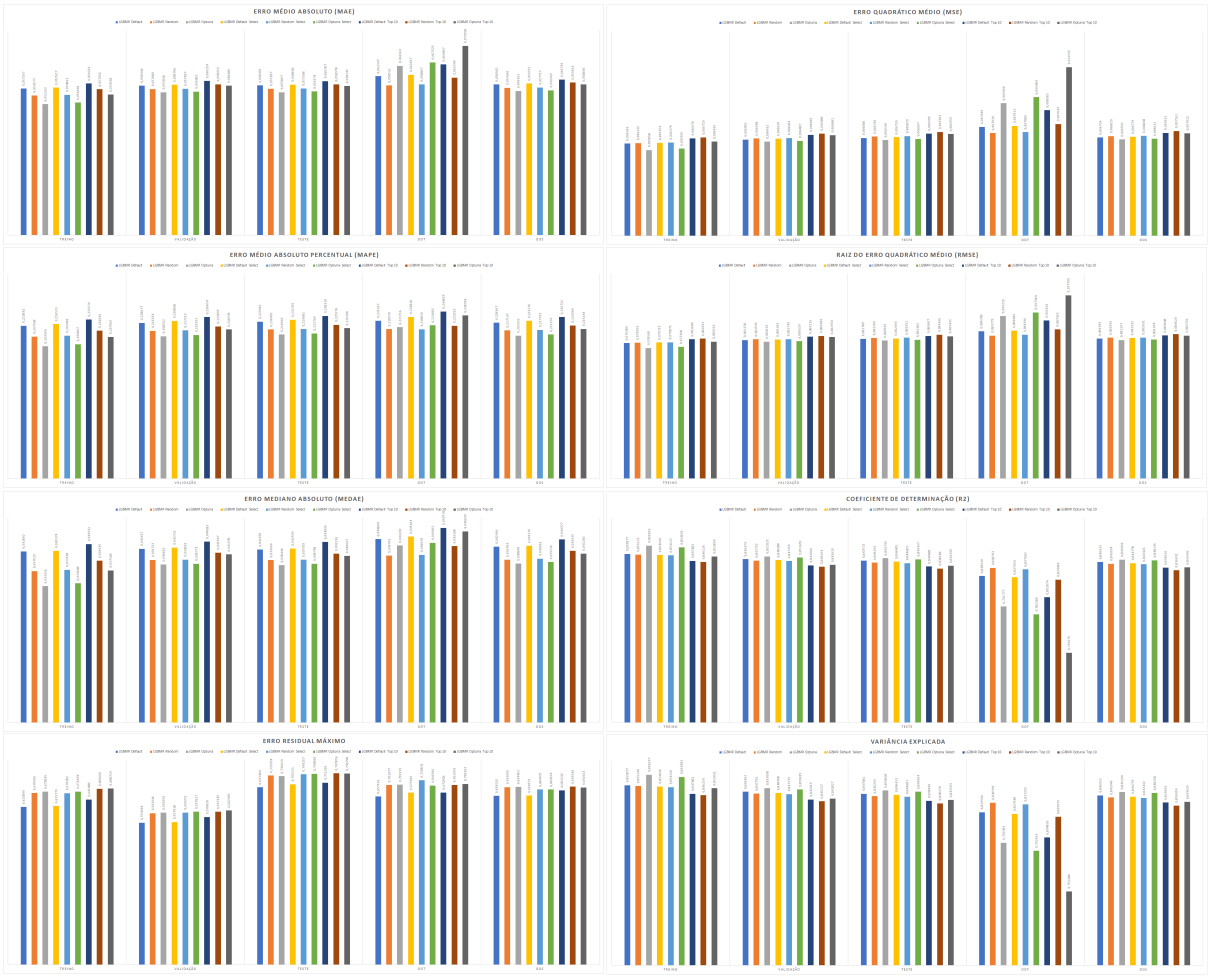


Fonte: Elaborada pelo autor (2022)

Tabela 40 – Resultado do Tempo de Execução dos modelos de LightGBM Regressor (LGBMR)

Modelo	Tempo de Processamento	Representatividade
LGBMR Default	00:00:21,226	0,520%
LGBMR Random	00:01:21,401	1,995%
LGBMR Optuna	00:22:36,734	33,253%
LGBMR Default Select	00:00:13,516	0,331%
LGBMR Random Select	00:01:12,578	1,779%
LGBMR Optuna Select	00:21:55,490	32,242%
LGBMR Default Top 10	00:00:09,361	0,229%
LGBMR Random Top 10	00:00:57,301	1,404%
LGBMR Optuna Top 10	00:19:12,410	28,245%
Total de tempo de processamento	**01:08:00,017**	**100,000%**

Figura 51 – Resultado das Métricas dos modelos de LightGBM Regressor (LGBMR)

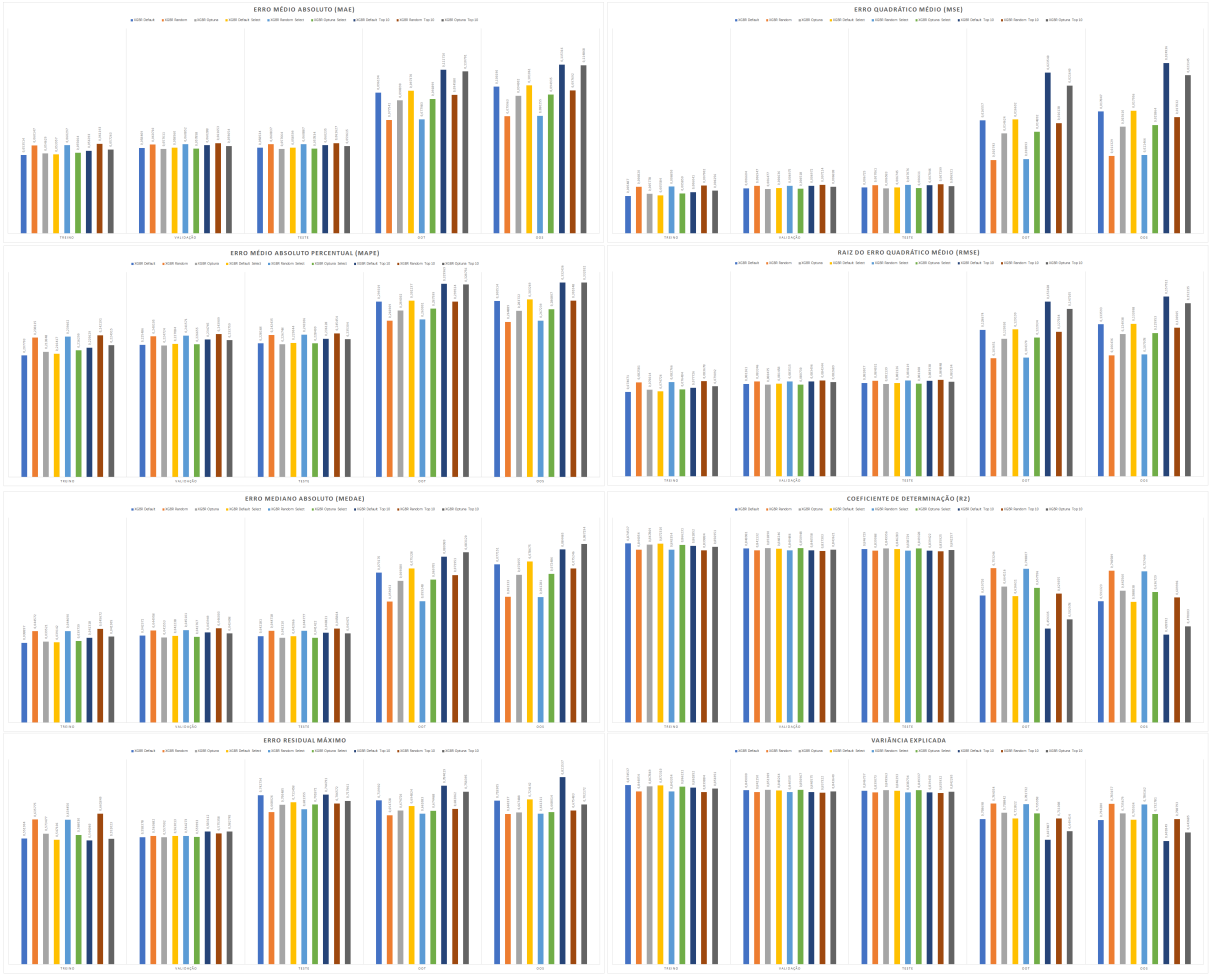


Fonte: Elaborada pelo autor (2022)

Tabela 41 – Resultado do Tempo de Execução dos modelos de XGBoost Regressor (XGBR)

Modelo	Tempo de Processamento	Representatividade
XGBR Default	00:00:08,273	2,760%
XGBR Random	00:00:07,111	2,372%
XGBR Optuna	00:01:35,914	32,000%
XGBR Default Select	00:00:07,410	2,472%
XGBR Random Select	00:00:06,066	2,024%
XGBR Optuna Select	00:01:25,639	28,572%
XGBR Default Top 10	00:00:06,075	2,027%
XGBR Random Top 10	00:00:04,928	1,644%
XGBR Optuna Top 10	00:01:18,313	26,128%
Total de tempo de processamento	**00:04:59,729**	**100,000%**

Figura 52 – Resultado das Métricas dos modelos de XGBoost Regressor (XGBR)

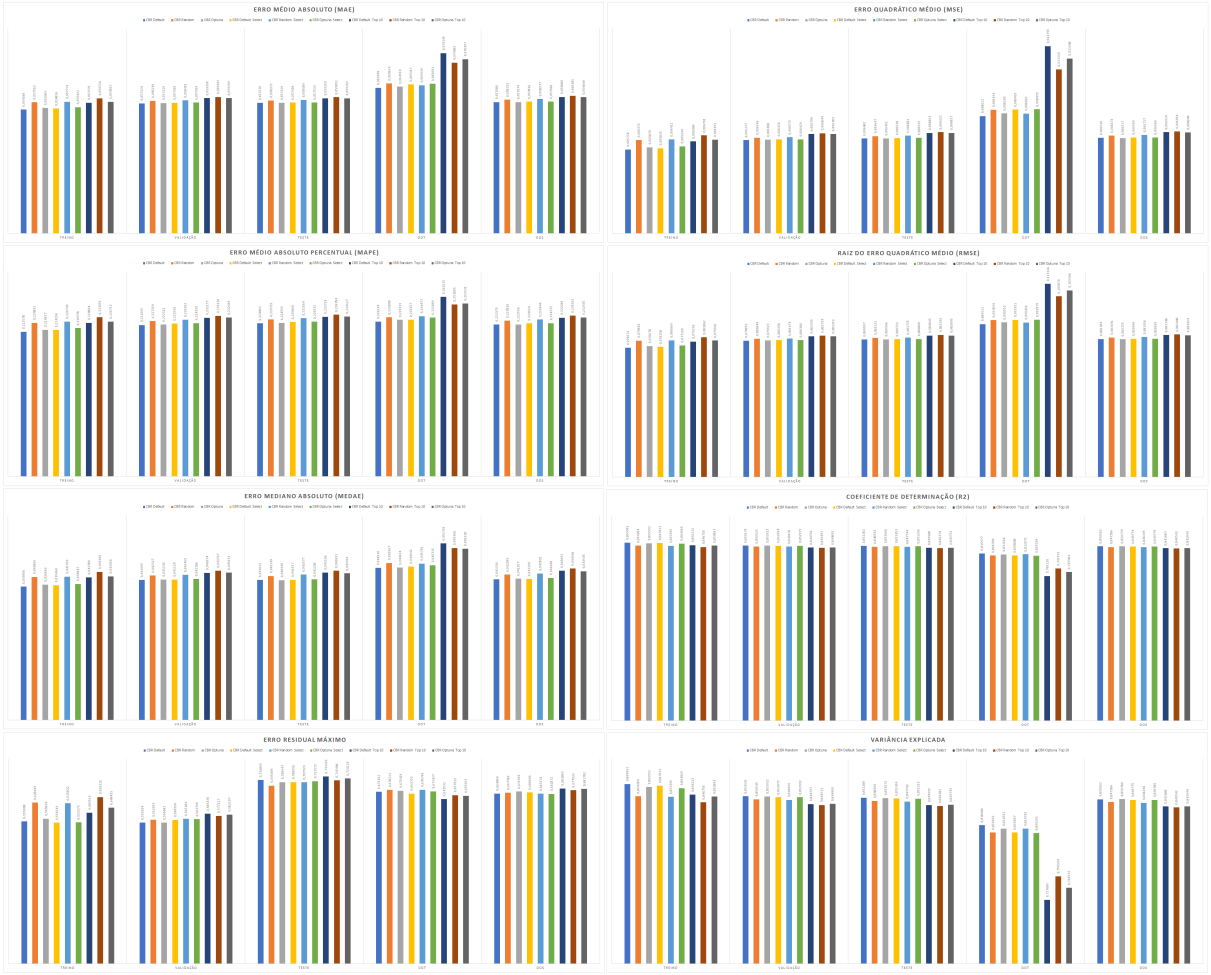


Fonte: Elaborada pelo autor (2022)

Tabela 42 – Resultado do Tempo de Execução dos modelos de CatBoost Regressor (CBR)

Modelo	Tempo de Processamento	Representatividade
CBR Default	00:03:33,001	16,124%
CBR Random	00:02:08,713	9,743%
CBR Optuna	00:03:52,339	17,587%
CBR Default Select	00:02:55,233	13,265%
CBR Random Select	00:01:40,591	7,614%
CBR Optuna Select	00:03:11,355	14,485%
CBR Default Top 10	00:01:47,054	8,104%
CBR Random Top 10	00:00:58,790	4,450%
CBR Optuna Top 10	00:01:53,982	8,628%
Total de tempo de processamento	**00:22:01,058**	**100,000%**

Figura 53 – Resultado das Métricas dos modelos de CatBoost Regressor (CBR)



Fonte: Elaborada pelo autor (2022)

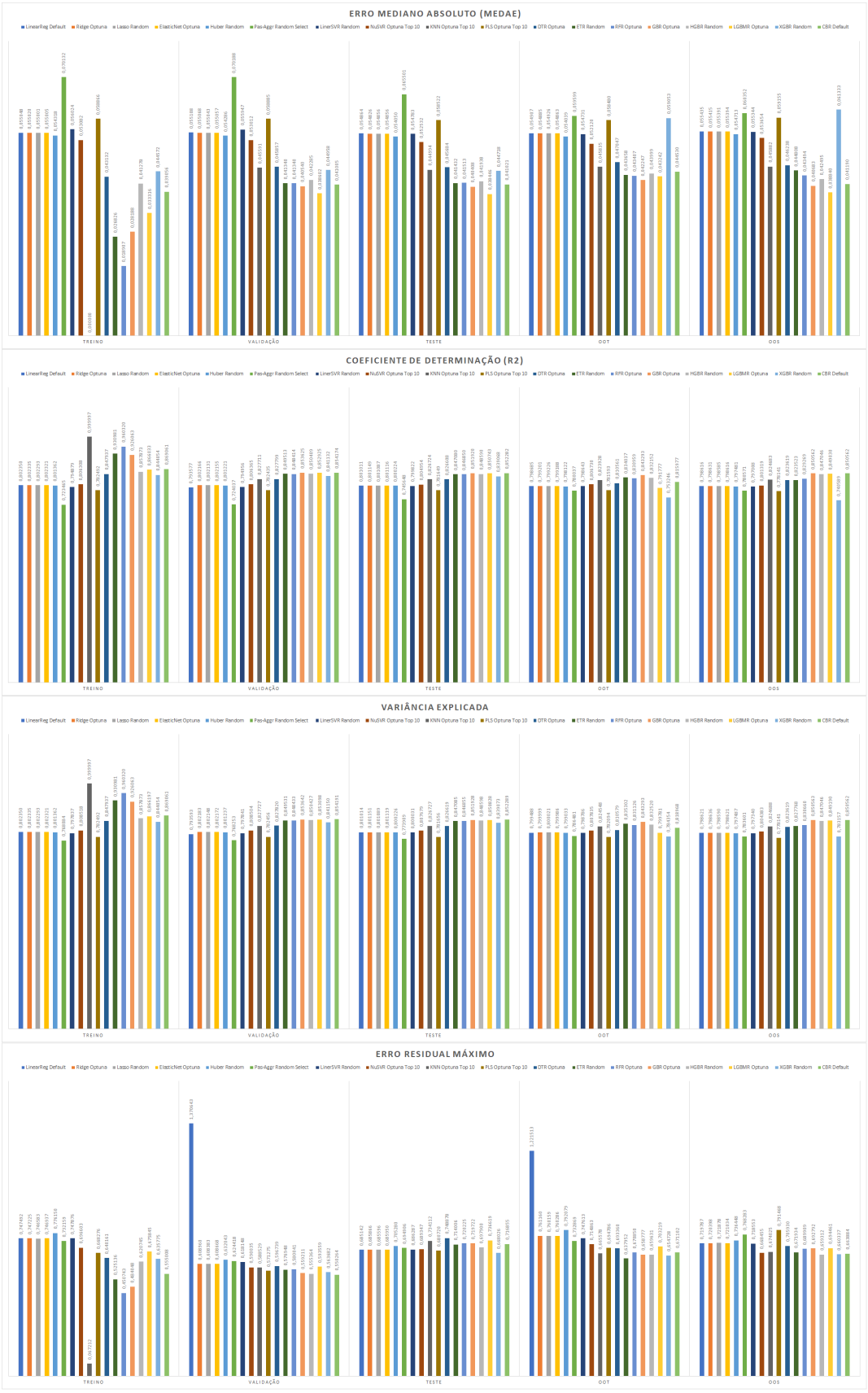
Tabela 43 – Resultado do Tempo de Execução dos melhores estimadores regressivos

Modelo	Tempo de Processamento	Representatividade
LinearReg Default	00:00:03,972	0,008%
Ridge Optuna	00:00:02,569	0,005%
Lasso Random	00:03:28,673	0,442%
ElasticNet Optuna	00:05:23,825	0,685%
Huber Random	01:54:27,778	14,531%
Pas-Aggr Random Select	00:00:15,559	0,033%
LinerSVR Random	03:09:03,549	24,001%
NuSVR Optuna Top 10	00:27:15,064	3,460%
KNN Optuna Top 10	00:06:56,747	0,882%
PLS Optuna Top 10	00:00:01,567	0,003%
DTR Optuna	00:00:43,038	0,091%
ETR Random	01:22:45,381	10,506%
RFR Optuna	03:29:29,706	26,596%
GBR Optuna	01:55:17,655	14,637%
HGBR Random	00:06:10,378	0,784%
LGBMR Optuna	00:22:36,734	2,871%
XGBR Random	00:00:07,111	0,015%
CBR Default	00:03:33,001	0,451%
Total de tempo de processamento	**13:07:42,307**	**100,000%**

Figura 54 – Resultado das Métricas dos melhores estimadores regressivos (parte 1)

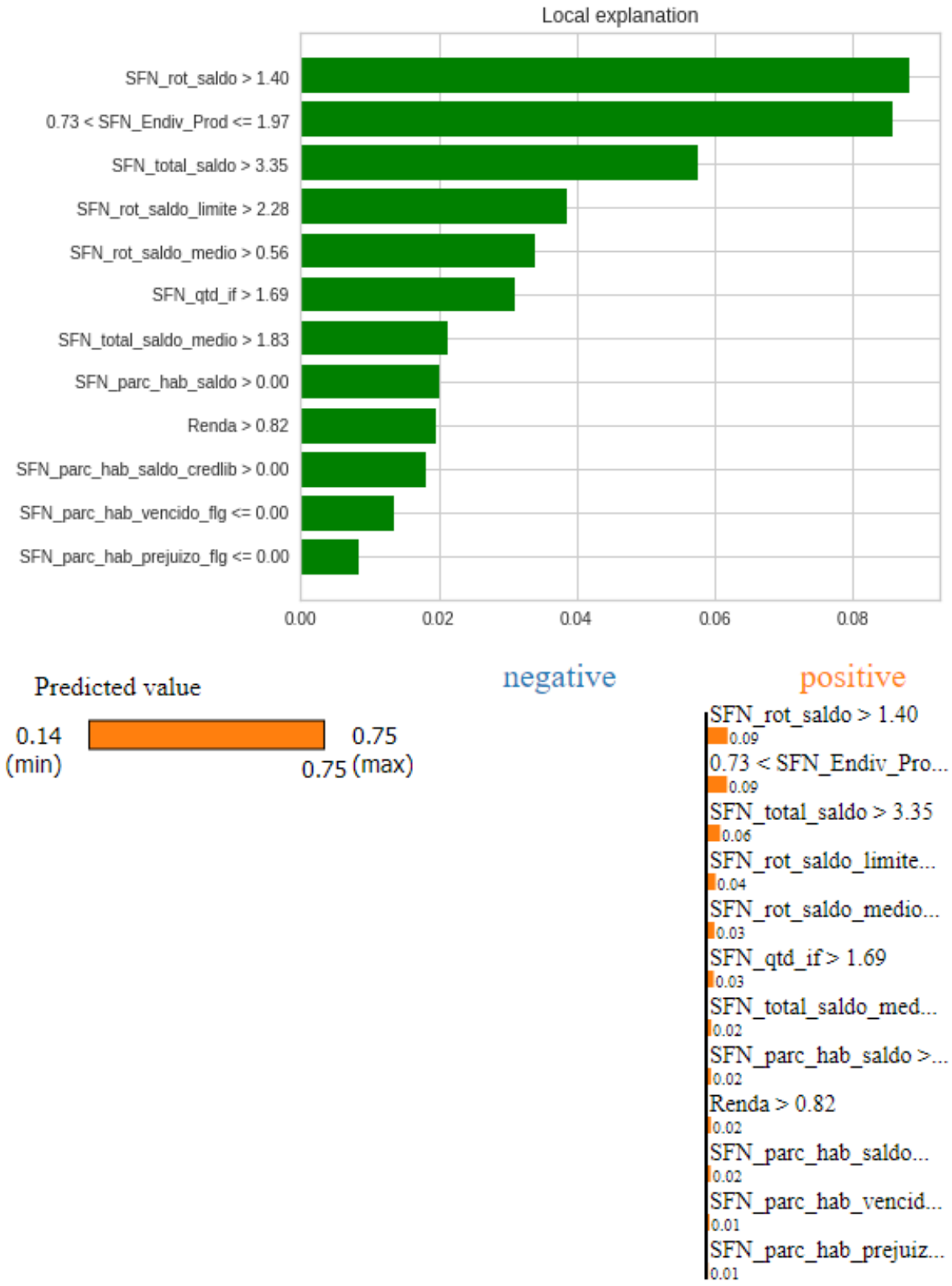
Fonte: Elaborada pelo autor (2022)

Figura 55 – Resultado das Métricas dos melhores estimadores regressivos (parte 2)



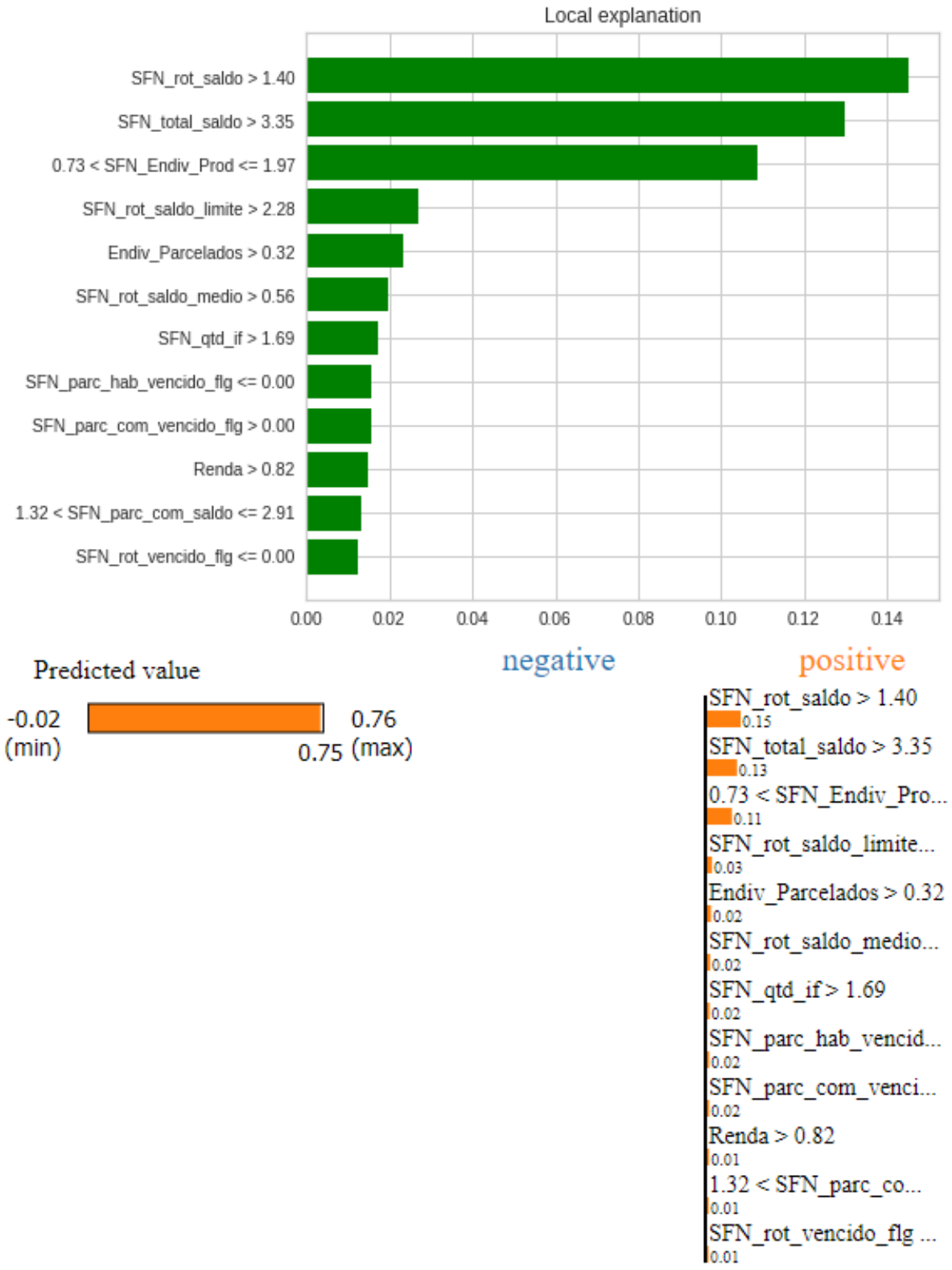
Fonte: Elaborada pelo autor (2022)

Figura 56 – LIME - Gradiente Boosting Regressor Optuna



Fonte: Elaborada pelo autor (2022)

Figura 57 – LIME - LightGBM Regressor Optuna



Fonte: Elaborada pelo autor (2022)