INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS - *CAMPUS* IBIRITÉ
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Filipe Augusto Valentins Araújo

APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL NA PREDIÇÃO DE HORAS GASTAS NO PROCESSO PRODUTIVO DO NÚCLEO DA PARTE ATIVA DE TRANSFORMADORES DE GRANDE PORTE

Ibirité - MG
2023

FILIPE AUGUSTO VALENTINS ARAÚJO

APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL NA PREDIÇÃO DE HORAS GASTAS NO PROCESSO PRODUTIVO DO NÚCLEO DA PARTE ATIVA DE TRANSFORMADORES DE GRANDE PORTE

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus* Ibirité para a obtenção do título de Engenheiro de Controle e Automação.

Orientador: Prof. Dr. Thiago Henrique Barbosa de Carvalho Tavares

Ibirité - MG
2023

A663a Araújo, Filipe Augusto Valentins.

Aplicação de inteligência artificial na predição de horas gastas no processo produtivo do núcleo da parte ativa de transformadores de grande porte. [recurso eletrônico] / Filipe Augusto Valentins Araújo. – Ibirité, MG, 2023.

60 p. : il. color.

Orientador: Prof. Dr. Thiago Henrique Barbosa de Carvalho Tavares.

Trabalho de Conclusão de Curso (Graduação) – Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais, *Campus* Ibirité, Bacharelado em Engenharia de Controle e Automação, 2023.

1. Inteligência artificial 2. Python (Linguagem de programação de computador). 3. Teoria da previsão 4. Aprendizado do computador. I. Tavares, Thiago Henrique Barbosa de Carvalho. II. Instituto Federal de Minas Gerais. *Campus* Ibirité. III. Título.

CDD 006.3

Catalogação: Viviane Barbosa Andrade - Bibliotecária - CRB-6/2819

Filipe Augusto Valentins Araújo

APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL NA PREDIÇÃO DE HORAS GASTAS NO PROCESSO PRODUTIVO DO NÚCLEO DA PARTE ATIVA DE TRANSFORMADORES DE GRANDE PORTE

Trabalho de conclusão de curso apresentado ao Curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais - *Campus* Ibirité para a obtenção do título de Engenheiro de Controle e Automação.

Aprovado em: 05/ 12/ 2023 pela banca examinadora:

gov.br Documento assinado digitalmente
THIAGO HENRIQUE BARBOSA DE CARVALHO TA
Data: 08/01/2024 15:34:10-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Thiago Henrique Barbosa de Carvalho Tavares - IFMG (Orientador)

gov.br Documento assinado digitalmente
EDUARDO VALADAO MELLO DE RESENDE
Data: 08/01/2024 16:05:02-0300
Verifique em https://validar.iti.gov.br

Eduardo Valadão Mello de Resende - Engenheiro de Produção

gov.br Documento assinado digitalmente
IVAN REINALDO MENEGHINI
Data: 08/01/2024 15:41:16-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Ivan Reinaldo Meneghini - IFMG

Dedico esta monografia aos meus amados pais, maiores incentivadores e fontes inesgotáveis de apoio, amor e compreensão.

AGRADECIMENTOS

Agradeço a toda à minha família, meus pais pelo incentivo constante na realização deste trabalho.

Agradeço ao meu orientador, supervisor de estágio, professores e a todos que contribuíram de alguma forma para a realização deste trabalho.

“Education is not preparation for life; education is life it self."

John Dewey

RESUMO

O desenvolvimento industrial desempenha um papel fundamental no crescimento das empresas e no atendimento às demandas sociais. Nesse contexto, este trabalho apresenta uma proposta concebida para abordar a necessidade urgente de otimização de processos enfrentada por empresas que evoluem para se tornarem multinacionais. O foco desse projeto é auxiliar o setor de Planejamento, Programação e Controle da Produção (PPCP) de uma empresa de transformadores de grande porte na previsão do tempo necessário para a fabricação de núcleos de grande dimensão.

Para alcançar esse objetivo, foi desenvolvido um algoritmo em Python que emprega Inteligência Artificial (IA) para realizar previsões. Além disso, ferramentas como o Excel e o VBA foram utilizadas para criar uma interface gráfica de fácil acesso para os usuários. Durante a execução do projeto, foram avaliados três métodos de regressão - Random Forest, Support Vector Machine e Rede Neural Artificial - a fim de selecionar o modelo mais adequado para ser empregado na interface gráfica.

Os resultados obtidos indicaram que o modelo Random Forest apresentou o melhor desempenho, o que levou à sua escolha final. Com a conclusão bem-sucedida do projeto, foi possível propor ao setor de PPCP a automação do processo de cálculo das horas gastas do setor de fabricação do núcleo.

Palavras-chave: Inteligência Artificial. Python. Teoria da previsão. Aprendizado do computador.

ABSTRACT

Industrial development plays a crucial role in the growth of companies and in meeting societal demands. In this context, this work presents a proposal designed to address the urgent need for process optimization faced by companies that evolve into multinational corporations. The focus of this project is to assist the Planning, Programming, and Production Control (PPCP) department of a large-scale transformer manufacturing company in predicting the time required for the production of large cores.

To achieve this objective, a Python algorithm was developed, utilizing Artificial Intelligence (AI) for making predictions. Additionally, tools like Excel and VBA were used to create a user-friendly graphical interface. During the project execution, three regression methods - Random Forest, Support Vector Machine, and Artificial Neural Network - were assessed to select the most suitable model for integration into the graphical interface.

The results obtained indicated that the Random Forest model exhibited the best performance, leading to its final selection. With the successful completion of the project, it became feasible to propose the automation of the hours spent calculation process to the PPCP department.

Keywords: Artificial Intelligence. Python. Prediction theory. Computer learning.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

IA	Inteligência Artificial
RNA	Rede Neural Artificial
SVM	*Support Vector Machine*
MLP	*Multi-Layer Perceptron*
RF	*Random Forest*
IDE	Ambiente de desenvolvimento integrado
IBM	*International Business Machines*
MAE	Erro absoluto médio
MSE	Erro quadrático médio
KNN	*K-Neighbor Nearest*
ARIMA	*Auto Regressive Integrated Moving Averages*
MAPE	*Mean Absolut Percentual Error*
ANN	*Artificial Neural Network*
LR	*Logistic Regressio*
LDA	*Linear Discriminant Analysis*
GBDT	*Gradient-Boosting Decision Trees*
LSTM	*Long Short Term Memory*
LST	*Land surface temperature* /Temperatura da Superfície Terrestre
GDM	*Gradient Descent with Momentum*
GD	*Gradient Descent*
AUC	*Area Under the Curve*
PPCP	Planejamento, Processo e Controle da Produção
SQL	*Structured Query Language*
TR	Trafo Série
RT	Reator Auxiliar

TP	Tipo do Núcleo Principal
LNP	Largura do Núcleo Principal
PNP	Peso do Núcleo Principal
FC	Espessura Aço-Silicio
VNNP	VNocht do Núcleo Principal
SLNP	Step-Lap do Núcleo Principal
TIF	Tipo do Trafo Série
LTF	Largura do Trafo Série
ECTF	Espessura da chapa do Trafo Série
VNTF	VNocht do Trafo Série
SLTF	Step-lap do Trafo Série
PTF	Peso do Trafo Série
HRS	Horas Gastas

LISTA DE SÍMBOLOS

Σ	Letra grega Sigma
N	Letra grega Ni
η	Letra grega Eta

SUMÁRIO

1 INTRODUÇÃO

A humanidade, em seus primórdios, já tinha maneiras de se resguardar de informações, mesmo sem ter desenvolvido a escrita ou a pronúncia, talvez apenas por meio de grunhidos e gestos. O farmacêutico, botânico e arqueólogo espanhol Marcelino Sanz De Sautuola, há aproximadamente 150 anos, encontrou registros conhecidos como Arte Rupestre na Caverna de Altamira. Esses registros são datados da Pré-História, segundo Madariaga de la Campa et al. (2000).

Presumimos que é praticamente instintivo a importância dos dados para os seres humanos. O acesso à informação foi o fator principal para a evolução humana, pois enquanto as pessoas se vão com o passar do tempo, muitos dos dados são repassados de gerações em gerações e isso é o que dá a total diferença para a garantia da sobrevivência e por consequência o aperfeiçoamento dessas informações, contribuindo de maneira significativa para a evolução.

Assim que a importância dos dados foi notada de maneira consciente pela sociedade, surgiram diversas soluções para armazena-los. No século XV, houve grandes avanços no armazenamento dos dados. Tudo começou com a Prensa de Impressão, inventada por Johannes Gutenberg em 1450, a prensa permitia a produção em massa de livros, segundo Rees (2006). Com isso, a disseminação de conhecimento e informação tiveram um crescimento em larga escala.

No entanto, esse foi apenas o primeiro passo para o que estava por vir. Após a invenção dos livros impressos, houve criações como a mídia analógica, criada por Renée Dragon, que consistia em um rolo de filme fotográfico. Essa tecnologia teve sua primeira utilização na Guerra Franco-Prussiana, para expor mapas microfilmados das tropas inimigas, conforme mencionado por Domingos et al. (2021).

Além dessa invenção, foram criados: Tambor Magnético por Gustav Tauscheck, em 1932; Disco Rígido pela IBM, em 1956; Disquete flexível também pela IBM, em 1968; Discos a Laser pela MCA, em 1978; Pen Drive por Dov Moran, em 2000; Serviço baseados em Nuvem pela Amazon, em 2006.

Apesar do Pen Drive e do Armazenamento em Nuvem terem sido criados a mais de uma década, as duas soluções sofreram modificações conforme o avanço tecnológico e ainda são usadas nos dias de hoje, fazendo parte das principais formas de armazenamento da atualidade.

Assim que os dados se tornaram mais acessíveis e abundantes diante de tantas formas de armazenamento, foram criadas maneiras de organiza-los. A principal forma de fazer isso foi por meio dos bancos de dados, criados pela empresa IBM.

Os bancos de dados são caracterizados pelo armazenamento de informações em massa e pela utilização de linguagens como Structured Query Language (SQL). Essa, por sua vez, é conhecida como Linguagem de Consulta Estruturada, essa é uma linguagem especificamente utilizada em bancos de dados SQL. Porém, existem também os bancos de dados NoSQL que são bancos de dados não relacionais. A principal diferença entre os dois tipos de bancos de

dados, se dá pela estrutura de armazenamento, onde no SQL todos os dados são armazenados em tabelas. Enquanto, no NoSQL os dados são armazenados separadamente em chaves, onde cada informação é acessada através de sua respectiva chave, segundo Chamberlin (2012).

Porém, mesmo com a criação dos bancos de dados e uma melhora no controle das informações, ainda sim haviam desafios pela frente, uma vez que, nem todas as informações existentes em um banco de dados podem ser consideradas confiáveis. Muitas vezes em uma massa de dados, acaba havendo a existência de dados equivocados, dados faltantes, dentre outros, que por fim geram um resultado não satisfatório nas análises realizadas como um todo.

Portanto, uma maneira de solucionar esse problema, seria aplicar tratamentos em cima desses dados. Desse modo, o procedimento padrão é utilizar a linguagem de programação como ferramenta, no qual uma das mais utilizadas é o Python.

As linguagens de programação são métodos padronizados, que consistem em um conjunto de regras semânticas e sintáticas, para implementação de um código fonte. Já o Python, especificamente, é uma linguagem de alto nível e fácil utilização por ter uma semântica muito próxima da linguagem natural.

Além disso, ela possui diversos pacotes e bibliotecas dedicadas à manipulação e tratamento de dados estruturados, a linguagem conta também com bibliotecas direcionadas a criação de gráficos para auxiliar na análise dos dados. Essa linguagem, foi criada por Guido van Rossum, em 1991, segundo o próprio Van Rossum et al. (2007).

Com a utilização das Bibliotecas em Python é possível encontrar, retirar e/ou substituir os dados faltantes, além de visualizar e realizar cálculos com os conjuntos de dados. A principal finalidade desses procedimentos é melhorar as amostras de dados para análises futuras.

Atualmente, existem inúmeros métodos de cálculos estatísticos e probabilísticos. No entanto, quando o contexto envolve um grande volume de dados, torna-se inviável realizar esses cálculos manualmente, mesmo com o auxílio de calculadoras. Nesse cenário, torna-se crucial a utilização de linguagens de programação, aliadas à Inteligência Artificial, para realizar essas tarefas de forma eficiente e escalável.

A inteligência artificial emerge como um campo de estudo desde os anos 50, com Allen Newell e Hebert Simon pioneiros na criação do primeiro laboratório de inteligência artificial na Universidade Carnegie Mellon. Paralelamente, McCarty e Marvin Minsky estabeleceram o MIT AI Lab em 1959. Este campo fundamenta-se na análise do comportamento de aprendizado da mente humana, conectado ao processamento de máquinas e à ciência de dados.

No entanto, mesmo com a disponibilidade de diversas ferramentas, muitas empresas ainda enfrentam desafios ao armazenar e utilizar eficazmente seus dados. Embora reconheçam a importância desse hábito, nem sempre conseguem extrair o potencial máximo de contribuição que esses dados oferecem. Em diversas situações, como previsões de vendas, demanda de materiais ou tempo de produção, empresas podem necessitar de informações futuras. Apesar de possuírem

uma base de dados sólida, algumas ainda enfrentam dificuldades em adquirir ou utilizam métodos de cálculo manual para obter essas informações.

Nesse cenário, a aplicação da inteligência artificial torna-se crucial no contexto industrial. A capacidade de prever tendências e comportamentos com base em dados históricos pode proporcionar uma vantagem competitiva significativa. A utilização de algoritmos avançados e modelos preditivos permite que as empresas otimizem processos, tomem decisões embasadas e alcancem eficiência operacional. Dessa forma, a integração da inteligência artificial na predição de dados emerge como um catalisador essencial para o avanço e sucesso no cenário industrial contemporâneo.

Os transformadores de grande porte desempenham um papel crucial no fornecimento de energia elétrica eficiente, sendo peças fundamentais em sistemas de transmissão e distribuição. Esses equipamentos são responsáveis por elevar ou reduzir a tensão elétrica, permitindo a transferência eficaz de energia ao longo de longas distâncias. A importância desses dispositivos é incontestável, pois impactam diretamente a confiabilidade e estabilidade dos sistemas elétricos, essenciais para o funcionamento de indústrias, empresas e residências.

A crescente demanda por transformadores de grande porte é impulsionada pela expansão das infraestruturas elétricas globais e pela transição para fontes de energia renovável. Com o aumento da geração de energia eólica, solar e outras formas de energia limpa, a necessidade de transformadores capazes de lidar com variações complexas de carga torna-se mais evidente. Nesse contexto, as empresas do setor de transformadores enfrentam desafios significativos para atender a essa demanda em constante crescimento.

A precisão no cálculo do tempo de produção torna-se um fator crítico para a eficiência operacional dessas empresas. Aqui, a integração de tecnologias avançadas, como a inteligência artificial (IA), emerge como um diferencial estratégico. A IA oferece a capacidade de analisar grandes volumes de dados históricos e em tempo real, identificando padrões e otimizando processos de produção. No contexto da fabricação de transformadores, a aplicação da IA pode ser particularmente benéfica para a previsão precisa do tempo de produção de cada peça, desde a bobina até os sistemas de isolamento.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo geral desse projeto é prever de maneira eficiente e otimizada as horas gastas no processo de fabricação do núcleo da parte ativa de um transformador. Para isso, será utilizado a linguagem de programação Python para desenvolver o algoritmo que filtra os dados de um data frame com as informações específicas dos núcleos dos transformadores, além de treinar e utilizar a Inteligência Artificial que realizará as predições das horas gastas. Por fim, unir tudo

isso e aplicar em uma IDE, criada no Excel, dedicada ao usuário, com o objetivo de criar um ambiente intuitivo e de fácil utilização.

1.1.2 Objetivos específicos

I Definir as informações do banco de dados geral do transformador a serem utilizadas para a realização de todo o processo de limpeza e tratamento dos dados;

II Escolher a linguagem de programação a ser utilizada para a realização do código que irá produzir a Inteligência artificial e realizar o processamento dos dados;

III Escolher os interpretadores para utilizar a linguagem definida, como por exemplo o Pycharm, Visual Code, Jupyter Notebook, Colab, dentre outros;

IV Importar a base de dados das características desejadas do transformador e bibliotecas a serem utilizadas para auxiliar no desenvolvimento de cálculos estatísticos, processamento e refinamento dos dados;

V Escolher os métodos de tratamento dos dados, conforme a necessidade e os tipos de dados apresentados na base de dados referente aos transformadores;

VI Definir as entradas e saídas da IA, de acordo com a base de dados selecionada e com a variável que se deseja predizer;

VII Separar dados em Treino e Teste de maneira aleatória e com a melhor proporção possível, tendo como base a quantidade de dados fornecidos;

VIII Definir o melhor método (Classificação ou Regressão) e técnica (Árvore de Decisão, Random Forest, SVM, MLP, KNN) a ser utilizado;

IX Parametrizar a IA, uma vez que as técnicas de predição dispõe de diversos parâmetros de configuração, o que ocasiona resultados diferentes a cada parametrização distinta, sendo assim necessário avaliar a base de dados e entender os melhores parâmetros para o modelo selecionado;

X Treinar a IA com os dados já processados, filtrados, convertidos no formato ideal e separados de maneira aleatória;

XI Aplicar os dados de teste na IA treinada, que tenham sofrido o mesmo processo de filtragem e conversão realizados nos dados de treino;

XII Calcular eficiência da IA utilizando cálculos estatísticos como Variância, Desvio Padrão, MAE, R2 ajustado, MSE;

XIII Verificar resultados gerados pelos cálculos estatísticos com o intuito de avaliar o desempenho da IA e avaliar se ela é confiável para ser utilizada;

XIV Criar um algoritmo que avalia as informações do núcleo do transformador de entrada e compara com as informações na base de dados, com o intuito de selecionar apenas os transformadores que tenham núcleos mais semelhantes. Ao final, serão realizados cálculos estatísticos com os dados desses transformadores. Desse modo, expõe-se ao usuário, facilitando a análise de confiança do resultado gerado pela IA;

XV Criar uma IDE utilizando o Excel ou alguma biblioteca direcionada a objeto, que possa ser intuitiva para que o usuário final consiga fazer uma requisição e possa visualizar o resultado da IA;

XVI Aplicar a IA na IDE, caso a IDE não seja feita com a mesma linguagem de programação utilizada no código do processo de filtragem, treinamento e teste da IA, será necessário a criação de um código na linguagem utilizada para criar a IDE que consiga executar o código onde tenha a IA;

XVII Testar funcionamento, utilizando os dados de um transformador qualquer e aplicando na IA já treinada e ver se o resultado está sendo exibido na IDE para o usuário final;

XVIII Utilizar o programa para fins práticos, como prever as horas gastas na fabricação do núcleo da parte ativa de um transformador que deseja ser fabricado.

1.2 Justificativa

Mesmo com toda a evolução tecnológica e maior acessibilidade à informação, ainda existem diversos problemas cuja solução não é tão trivial e fácil de ser implantada. Por isso, muitas empresas ainda tendem a utilizar técnicas ultrapassadas para realizar as atividades diárias. Um exemplo disso é a execução manual de cálculos preditivos de horas gastas nos processos produtivos, uma prática comum entre os engenheiros de produção. Esses profissionais utilizam esses cálculos para programar a fábrica e determinar quando ocorrerá a entrega dos produtos finais.

Entretanto, em empresas onde os produtos não seguem um padrão em série, esse processo manual pode se tornar confuso, maçante, demandar muito tempo e resultar em informações imprecisas em certos momentos. Por essa razão, criar um algoritmo que possa automatizar esse cálculo com uma boa taxa de precisão e eficiência em apenas um clique pode ajudar os setores de Vendas e de Planejamento e Controle da Produção nas empresas, otimizando e melhorando o processo de cálculos preditivos.

Por essas e outras razões, torna-se totalmente viável utilizar a Inteligência Artificial para ajudar no processo de automatização de tarefas repetitivas que exigem recursos probabilísticos.

Existem diversas maneiras de se aplicar a inteligência artificial, seja por métodos como a classificação ou a regressão. Dentro desses métodos, são utilizadas técnicas como *K-Nearest*

Neighbors (KNN), Árvore de decisão, *Random Forest*, *Support Vector Machine* (SVM), Rede Neural, dentre outras. Entretanto, em cada técnica, existem diversas variações de parâmetros que podem ser realizadas.

Ademais, para algumas aplicações um método é mais adequado que o outro. Portanto, para melhorar ainda mais a eficiência do método escolhido pode-se conciliar o mesmo com técnicas diferentes e com parametrizações distintas diante da técnica escolhida. Nesse caso, esse projeto vem com o intuito de descobrir qual a melhor combinação de métodos, técnicas e parametrizações da Inteligência Artificial para alcançar a maior eficiência possível na predição de horas gastas no setor de produção do Núcleo da Parte Ativa de uma empresa de transformadores.

1.3 Organização do Texto

Este trabalho está organizado da seguinte forma:

- Capítulo 1 – Introdução
  No primeiro capítulo, é apresentado uma breve contextualização sobre os temas ao entorno do trabalho em formato de texto corrido, além de apresentar os problemas existentes que motivaram o desenvolvimento do mesmo. Ademais, são encontrados o objetivo geral, os objetivos específicos, a justificativa e a estrutura do trabalho.

- Capítulo 2 – Revisão Bibliográfica
  Neste capítulo, são apresentadas diversas literaturas que possuem temas semelhantes com alguma área que englobe os assuntos abordados neste trabalho, além de serem base de estudo para o desenvolvimento do mesmo.

- Capítulo 3 – Metodologia
  No terceiro capítulo, são descritos os métodos de seleção, tratamento e filtragem dos dados, também são descritos os modelos de regressão selecionados para o treinamento da IA, os parâmetros selecionados, os métodos de avaliação e os softwares escolhidos para cada aplicação.

- Capítulo 4 – Resultados
  No quarto capítulo, são mostrados os gráficos, tabelas e imagens referentes aos resultados adquiridos com o treinamento da IA, avaliação de desempenho e designer da IDE. No mais, foram analisados os desempenhos dos modelos, possíveis contrapontos, as limitações e as possíveis contribuições do trabalho.

- Capítulo 5 – Conclusão
  No último capítulo, foram apresentadas as conclusões finais do trabalho, um resumo e uma visão geral sobre os modelos de regressão, foi destacado a importância do tema e os possíveis trabalhos futuros.

- Referência Bibliográfica
  Ao final, foram incluídas citações, conforme as normas, de diversas fontes de artigos, livros, documentos, etc., que contribuíram de maneira significativa para o desenvolvimento deste trabalho.

- Apêndice e Anexos
  Caso haja arquivos, textos, imagens, tabelas, dentre outros materiais que complementem o texto, eles serão adicionadas ao apêndice, após as referências bibliográficas.

2 REVISÃO BIBLIOGRÁFICA

A aplicação da inteligência artificial abrange a previsão de informações em diversas esferas do conhecimento, revelando-se uma ferramenta amplamente valorizada e adotada por analistas, cientistas e entusiastas. Esse reconhecimento é particularmente evidente quando se lida com grandes volumes de dados, exigindo a habilidade de identificar padrões em uma extensa gama de variáveis e argumentos. No entanto, é imperativo destacar que, para que a IA alcance um desempenho adequado diante do problema proposto, torna-se essencial a execução de processos preliminares, tais como mineração e tratamento de dados.

A mineração de dados geralmente é realizada pelos Engenheiros de dados, o processo consiste em extrair os dados de uma fonte de dados não relacional (No SQL) e converte-los para um banco de dados relacional (SQL), como introduzido pelos autores de Castro (2016). Com isso, podendo haver a utilização dos dados para criação de tabelas, gráficos e algoritmos de treinamentos para IA.

De outra forma, o tratamento de dados é geralmente realizado pelos cientistas de dados, o que consiste no tratamento das bases de dados relacionais, aplicando métodos estatísticos e métodos de filtragem dos dados, como preenchimento dos dados faltantes, exclusão/substituição de dados equivocados, exclusão de outliers, reparo e balanceamento dos dados, dentre outros. Exemplo desses métodos, foram abordados pelo autor Shaikh (2018), em sua publicação o autor demonstra e explica dois métodos de tratamento dos dados, denominados Label Encoder e One Hot Encoder. Ambos os métodos são utilizados para conversão dos dados qualitativos em dados quantitativos. Ao longo do artigo, o autor utiliza o RMSE para avaliar o desempenho de ambos os métodos. Como conclusão, o método One Hot Encoder se sobressaiu diante do método Label Encoder, ou seja, obteve um erro quadrático menor.

Além disso, diversos outros autores também empregaram métodos de mineração e tratamento de dados. O autor da Rocha et al. (2019), em seu trabalho, aborda o tema da ciência de dados e aprendizado de máquina para realizar previsões por meio de séries temporais financeiras. Como método de extração dos dados, foram aplicados filtros na massa de dados para extrair apenas informações específicas das ações na B3, como código do ativo e preço de fechamento.

Após a extração dos dados, foi realizado o processo de tratamento, no qual foram removidos valores nulos, incorretos ou inconsistentes. Em seguida, todos os dados restantes passaram pelo processo de normalização, utilizado para evitar que os dados transmitam informações equivocadas durante o treinamento da IA, como receber um grau de relevância ou peso irreal.

Em relação ao contexto da mineração de dados, o autor Souza (2021) em seu TCC usou como tema a mineração de dados aplicada a previsão de preços de ações utilizando WEKA. Durante o trabalho foram aplicadas técnicas de mineração aos dados históricos das ações da Petrobras no ano de 2019. No entanto, o processo de mineração teve o auxílio do software WEKA. O autor após realizar a mineração dos dados também aplicou técnicas de tratamento e aplicação

de Classificadores e Regressores como: Árvore de decisão, Análise Bayesiana, KNN e RNA. Com o intuito de realizar a previsão das informações das ações. Como resultado o método RNA foi o que obteve a melhor performance, após a mineração e tratamento dos dados.

Contudo, para facilitar e melhorar o desenvolvimento dos algoritmos de tratamento e treinamento das Inteligências Artificiais, existem diversos Ambientes de Desenvolvimento Integrados (*Integrated Development Environment* - IDE) e Linguagens de programação. Com isso, uma das linguagens que mais vem sendo utilizadas para a área da ciência de dados é a linguagem Python. O autor Tatsat et al. (2020) defende a utilização dessa ferramenta para a criação dos algoritmos de *machine learning*, pelo fato da ampla cobertura de bibliotecas e pacotes dedicados ao *machine learning* fornecidos pela linguagem.

A aplicação dos métodos baseado em IA, para previsão de dados, tem sido utilizado em diversos setores. Exemplo disso, seriam trabalhos envolvendo a aplicação da IA para previsão da inflação, previsão de demandas para materiais nas indústrias, previsão de preço de ações, previsão de preço de mercadorias, previsão de possíveis desastres naturais, predição da suscetibilidade a doenças, etc. Existem inúmeros estudos aplicados nesse contexto.

O autor Borsato and Corso (2019) realizou um trabalho voltado para a análise de desempenho de um modelo de Rede Neural Artificial (RNA) onde ele buscava comparar o modelo com o método Médias Móveis Integradas Auto-regressivas (*Auto Regressive Integrated Moving Averages* - ARIMA). Para realizar a comparação ele usou os dois métodos para prever a demanda de uma empresa que atual no setor metal mecânico. Como critério de avaliação de desempenho, foi-se utilizado o Erro Percentual Médio Absoluto (*Mean Absolute Percentual Error* - MAPE) e o Erro Absoluto Médio (*Mean Absolute Error* - MAE). Ao fim das analises feitas com as comparações estatísticas entre os resultados do método ARIMA e o método RNA, foi evidenciado que o método com melhor desempenho é do RNA.

Com foco na área da economia, a autora Zaniol et al. (2021) construiu uma frente de trabalho para a previsão da inflação. Foi-se utilizado como objeto de estudo o núcleo de inflações baseados em *wavelets*, já para a previsão das informações foi-se utilizado o método de Rede Neural Artificial. Os dados aplicados à RNA foram os núcleos de Inflação denominados IPCA-EX2, IPCA-MS, IPCA-DP, db4, db6, db8, db10. Para análise de desempenho foi-se utilizado a métrica $R2$ e o Erro Quadrático Médio. No caso do Erro Quadrático Médio, foram-se utilizados dados de entradas para a previsão de 5 horizontes temporais, 1 mês, 3 meses, 6 meses, 9 meses e 12 meses. Onde o melhor desempenho, no EQM, foi para os dados de entrada de 6 meses e 9 meses para os núcleos de inflação IPCA-MS e IPCA-EX2. Já na métrica $R2$, o melhor desempenho foi no núcleo de inflação db6. Como conclusão, a autora afirma que é viável a utilização da IA pelo método de RNA para previsão da inflação, porém conciliada a núcleos de inflação oficiais, que são intervalos de confiança, e por esse motivo agregam maior robustez aos resultados.

O autor Nascimento (2023), aplicou métodos de aprendizagem de maquina como ARIMA, PROPHET e LSTM na base de valores de fechamento diário de duas ações e uma ETF negociadas

na Bovespa. Com um período de 60 dias todos os modelos obtiveram resultados satisfatórios. Nos primeiros 30 dias apenas o ARIMA e PROPHET tiveram uma precisão satisfatória, já no período de 90 dias o modelo LSTM obteve a melhor métrica RMSE e MAPE.

No entanto, o autor Santos (2021) utilizou diversos métodos de machine learning para verificar qual método tem o melhor desempenho para a classificação de crédito bancário. Como métodos, foram selecionados *Logistic Regression* (LR), *Linear Discriminant Analysis* (LDA), *Artificial Neural Network* (ANN), AdaBoost, *Gradient-Boosting Decision Trees* (GBDT), *Xtreme Gradient Boosting* (XGBoost), *Light Gradient Boosting Machine* (LightGBM) e CatBoost. Como resultado final o autor identificou que os melhores métodos de avaliação são os modelos de classificadores heterogêneos, como por exemplo o XGBoost, CatBoost e LightGBM.

Assim como o trabalho do Borsato and Corso (2019), o autor Adebiyi et al. (2014) realizou a comparação entre o método ARIMA e o Método RNA. Contudo, alinhado ao tema relacionado a previsão de ações no mercado financeiro. Ao fim do trabalho, o autor chegou a mesma conclusão que o Borsato and Corso (2019), sobre o desempenho dos métodos, onde o método RNA se sobre saiu diante do método ARIMA, uma vez que o maior modulo de erro dos dados previstos para o método ARIMA foi de 0,038 e para o método RNA foi de 0,032.

Ademais, há muitos trabalhos, envolvendo Inteligência artificial, voltados para áreas naturais e contextos rurais. Em relação às áreas rurais, o autor Marujo et al. (2021) realizou um estudo comparativo entre o modelo ARIMA e de rede LSTM para previsão do preço do café no Brasil. Os resultados da comparação implicaram que o modelo de melhor desempenho foi o modelo ARIMA, porém a curto prazo, em períodos maiores, a longo prazo, o LSTM teve um melhor desempenho.

Voltado para estudos das áreas naturais, dois outros autores desenvolveram grandes pesquisas, mas dessa vez utilizando um método diferente denominado *Random Forest*, método que randomiza e tende a expandir a eficiência do método *DecisionTrees* Rigatti (2017).

O primeiro autor Sun et al. (2021) fez um estudo comparativo entre os métodos *Logistic Regression* (LR) e *Random Forest* (RF). Em seu trabalho, o autor usufruiu dos dois métodos para predizer a suscetibilidade do solo em sofrer um deslizamento. Afim de obter uma melhor acurácia, o autor aplicou uma hiper parametrização bayesiana otimizada nos métodos propostos. Ao fim de suas análises, foi-se encontrado um aumento de desempenho de 4% no método LF e um aumento de 10% no desempenho do método RF. Portanto, em relação aos resultados da predição, ambos os métodos demonstraram performances razoáveis. Porém, o método Random Forest, quando baseado na hiper parametrização otimizada, teve uma melhor estabilidade e capacidade de previsão no caso de área.

Por outro lado, o segundo autor Zhao et al. (2019) realizou um estudo prático para redução dos efeitos no terreno, gerados pela Temperatura da Superfície Terrestre (*Land surface temperature* - LST), utilizando *Random Forest Regression*. O autor utilizou como base de dados

diversas variáveis de superfície que tivessem ligação com a LST, como índice de vegetação por diferença normalizada (NDVI), índice de vegetação melhorado (EVI), índice de área foliar (LAI), albedo de superfície (ALB), incidente cumulativo radiação solar (CSR), índice de diferença hídrica normalizada (NDWI), elevação da superfície (ELV) e inclinação da superfície (SLP). Foi realizada a aplicação do RF nos dados, onde o coeficiente de determinação ($R2$) resultou em um valor acima de 92%, o que indica uma boa relação entre o LST e as demais variáveis. Com isso, o autor alterou algumas variáveis que são afetadas pela topografia como CSR, ELV e SLP. Desta forma, por meio da validação cruzada ele verificou uma significativa mudança na Temperatura da Superfície Terrestre (LST), havendo uma diminuição de 9,51 Kelvin.

Existem trabalhos que também envolvem objetos de estudo derivados da natureza, mas tem grande impacto tanto na área urbana, quanto nas zonas rurais. O autor Silva (2019) realizou uma dissertação onde o tema propunha a utilização da Inteligência Artificial para avaliação da qualidade da água. No trabalho foi-se utilizado o método de RNA e Random Forest para a predição da concentração de clorofila-a, que é um dos principais indicadores do processo de eutrofização. Para que houvesse a realização da predição foram utilizados bancos de dados da Companhia Ambiental do Estado de São Paulo (CETESP) e da United State Geological Service (USGS). Para avaliação dos desempenhos foram utilizados o MSE e o RMSE. A Rede Neural Artificial foi utilizada para realizar a predição da clorofila-a e o desempenho do método teve concordância tanto nos dados de treino quanto nos dados de teste. Já o método Random Forest, foi utilizado para realizar as predições da clorofila-a em reservatórios do estado de Sergipe, porém a base de dados era menor do que o necessário. Com isso, o resultado se manteve satisfatório. No entanto, haveria uma melhora no desempenho, se a base de dados fosse maior.

Com a mesma perspectiva de manter água como fonte de estudo, o autor Mustafa et al. (2012) desenvolveu um artigo com o objetivo de prever, usando Multilayer Perceptron, a descarga de sedimento suspenso em rios na Península da Malásia. Como algoritmos de treinamento, foram usados: Gradient Descent (GD), Gradient Descent with Momentum (GDM), Scaled Conjugate Gradient (SCG) e Levenberg Marquardt (LM). Os desempenhos dos algoritmos foram baseados no tempo de convergência e número de épocas. Como resultado, os algoritmos LM e SCG obtiveram o melhor desempenho, porém o LM conseguiu atingir o tempo de convergência antes do SCG (Cerda de 1/7 do tempo gasto pelo SCG).

A área da medicina é fundamental para solucionar os problemas diários dos seres humanos e em concordância a esse ponto, as Inteligências Artificiais, através de outra perspectiva, têm como intenção solucionar diversos problemas encontrados pelos humanos. Desta forma, a IA é totalmente sinérgica com a área da medicina. Conforme a autora Braga et al. (2019), a Inteligência Artificial ser utilizada na medicina pode viabilizar a melhora da precisão nas previsões da evolução de doenças, na manutenção do desempenho dos tratamentos e nos menores riscos para pacientes. Em sua pesquisa a autora concluiu que o Machine Learning possibilita uma significativa melhora na precisão e confiabilidade das modalidades diagnósticas e tem o potencial de contribuir com o

objetivo da medicina de precisão.

A partir do contexto da IA na medicina, a autora Takáo (2023) e o autor Yang et al. (2020) realizaram trabalhos envolvendo esse tema. A autora Takáo (2023) fez um estudo a cerca da Inteligência Artificial na área da Alergologia e da Imunologia, foram realizados o desenvolvimento de modelos de predição de risco para Erros Inatos da Imunidade (EII). Durante o processo de predição foram usados três modelos de Machine Learning e um modelo baseado em Regressão Logística. As variáveis utilizadas como dados de treino foram anemia, leucopenia, neutropenia e linfopenia, baixos níveis séricos de imunoglobulinas A/G/M e níveis séricos aumentados de imunoglobulina E. Com o algoritmo aplicado, foi possível identificar diferentes preditores de risco. De acordo com os resultados, os modelos de Machine Learning obtiveram uma performance melhor em relação ao modelo de Regressão Logística. O modelo de Machine Learning que mais se sobre saiu em relação ao modelo de RL, foi o modelo denominado Random Forest.

De forma distinta, o autor Yang et al. (2020) utilizou como base o modelo de predição Random Forest para realizar estudos sobre doenças cardiovasculares (CVD) na região leste da China. O trabalho realizado utilizou como base de dados 30 possíveis condições que estão relacionadas a doenças cardiovasculares, algumas destas são: velhice, indivíduos masculinos, renda familiar, tabagismo, consumo de álcool, obesidade, circunferência da cintura excessiva, colesterol anormal, lipoproteína de baixa densidade anormal, glicemia de jejum anormal e muito mais. Para realizar a predição foram utilizados 6 modelos de Machine Learning, multivariate regression model, classification and regression tree (CART), Naïve Bayes, Bagged trees, Ada Boost and Random Forest. Como método de avalição do desempenho foi utilizado o AUC (Area Under the Curve) o modelo de referência foi o de Regressão Multivariada (AUC = 0.714). Como conclusão, o melhor método apresentado foi o do Random Forest (AUC = 0.787) que obteve uma performance significativamente maior do que o método de referência.

A IA não necessariamente está atrelada apenas a predição dos dados, como números e textos, advindos de planilhas e banco de dados, ela também pode ser usada, com eficiência, para reconhecer padrões e anomalias em imagens. O autor Olivo et al. (2020) aplicou o uso da inteligência artificial para auxiliar no reconhecimento de objetos nas redes elétricas, com o intuito de facilitar a manutenção. As imagens utilizadas para treinar a IA foram capturadas por um drone. Foi-se utilizado para o projeto uma Rede Neural Artificial (RNA) Darknet-53 do YOLOv3 no framework Tensorflow, o que possibilitou a classificação e detecção das anomalias e componentes presentes nas linhas de transmissões, em tempo real. Como atributos alvo, foram selecionadas 7 classes de objetos. Obteve-se 84,16% de precisão da variação média (mAP).

Além disso, o autor Santos et al. (2020) realizou um trabalho onde parte de suas fontes de dados foram retiradas de bases de dado e a outra de um banco de imagens. O trabalho envolve a criação de um algoritmo de machine learning para a previsão dos dados das ações da B3. Na dissertação foi proposta a comparação da predição dos métodos: Random Forest, Redes Neurais Artificiais e Support Vector Machine. No mais, foi-se utilizado uma função denominada

Kernel-SVM utilizada para reconhecimento de imagens. Os argumentos de entradas para o treinamento da IA foram: Índice de Força Relativa, o oscilador estocástico, entre outros. Para medir o desempenho foram avaliados acurácia, precisão, recall e especificidade. Ao fim dos testes, o modelo com a melhor performance em todos os períodos e ações selecionadas foi do Random Forest (média da Acurácia: 94%) e o pior foi o método RNA (média da Acurácia: 79%).

Focado no âmbito elétrico no desenvolvimento e estudo de tecnologias voltadas para transformadores, o autor Alexandre et al. (2017), destaca a importância vital desses equipamentos em subestações, especialmente em um contexto em que o sistema elétrico brasileiro opera próximo de seus limites. A falta de disponibilidade de transformadores pode resultar em desligamentos e transtornos significativos. Costa enfatiza a necessidade crucial de implementar sistemas de monitoramento eficazes para antecipar e agendar intervalos de manutenção, prevenindo paradas e falhas. Ele ressalta que um sistema de monitoramento eficiente deve abranger a aquisição, armazenamento e tratamento de variáveis medidas, proporcionando diagnósticos e prognósticos precisos do estado dos transformadores. Assim, o monitoramento não apenas fornece acesso rápido a informações seguras, mas também se torna uma ferramenta indispensável para reduzir custos, aumentar a confiabilidade e a disponibilidade do equipamento, além de minimizar drasticamente o tempo necessário para a manutenção. Costa destaca a relevância desse enfoque na otimização da eficiência operacional e na garantia da integridade dos transformadores de potência.

Por outro lado, o autor Calil (2009), por meio deste estudo, apresenta uma proposta inovadora para o cálculo da correção do fator de construção de núcleos de transformadores de potência, destacando a influência das perdas na região das juntas magnéticas. A relevância desse fator reside na capacidade de corrigir as perdas em vazio no núcleo, originalmente obtidas a partir das curvas do fabricante das chapas de aço-silício. Calil observa que a determinação desse fator, até o momento, tem sido predominantemente baseada em estimativas empíricas e estatísticas. A abordagem proposta por Calil se fundamenta no cálculo das perdas magnéticas na região das juntas, sendo esse cálculo derivado de simulações computacionais do transformador pelo Método de Elementos Finitos. O uso desse fator de correção resultou em melhorias significativas na precisão do cálculo das perdas em vazio, aproximando os resultados aos valores experimentais. O estudo investigou diferentes tipos de juntas magnéticas, com e sem step-lap, além de três dimensões de entreferro, analisando a influência desses parâmetros no fator de construção. As simulações foram conduzidas por meio de um programa comercial que emprega o Método de Elementos Finitos em duas dimensões, ressaltando a abordagem avançada proposta por Calil para aprimorar a compreensão e o cálculo preciso das perdas em transformadores de potência.

3 METODOLOGIA

3.1 Base de Dados

O processo de previsão das informações só conseguiu se tornar efetivo, perante a sociedade, quando de fato os dados e informações se tornaram abundantes, a ponto de os padrões serem evidenciados e estudados. Portanto, com isso foi criada o estudo da estatística e probabilística.

Nos dias de hoje se tornou fundamental lidar e saber como gerenciar os dados coletados, principalmente quando o produto vendido pela empresa é completamente personalizável e feito sob a necessidade do cliente, em outra palavras, um produto que não é fabricado em série. Por esse e outros motivos surgiram a necessidade da criação desse trabalho.

Dito isso, para realizar a coleta dos dados foram necessários o preenchimento, por parte da equipe do Planejamento, Programação e Controle da Produção (PPCP), de planilhas no Excel, conforme a fabricação e a demanda dos Transformadores de grande porte, vistos na figura 1.

O desenvolvimento das planilhas consistia em adicionar as características dos transformadores que eram fabricados dividindo-os em duas partes principais como parte mecânica, que envolvia a parte da fabricação do tanque e da pintura dos transformadores e a parte elétrica, que consistia no desenvolvimento do núcleo e das bobinas do transformador. Em seguida, assim que os transformadores eram finalizados em cada área, eram contabilizada as horas que foram demandadas para a fabricação e essas horas eram adicionadas às planilhas.

Figura 1 – Transformador de grande porte.

Fonte: https://www.revistamundoeletrico.com.br/tecnologia/tsea-energia-coloca-em-operacao-nova-unidade-de-reforma-de-transformadores-de-potencia/.

Deste modo, como o trabalho em questão utilizou dados dos transformadores diretamente relacionados a área específica no desenvolvimento do núcleo, apresentado na figura 2, foram necessários separar apenas informações que de fato tinham impacto nas horas de fabricação desse Centro de Custo, em específico, e adiciona-los em uma Base de dados, em Excel, que será utilizada posteriormente.

Figura 2 – Núcleo de transformadores de grande porte.

Fonte: https://www.engineeringworldchannel.com/transformer/.

Além disso, as variáveis empregadas na construção da base de dados destinada ao desenvolvimento integral do projeto foram: TR, RT, TP, LNP, PNP, FC, VNNP, SLNP, TIF, LTF, ECTF, VNTF, SLTF, PTF e HRS. Todas as abreviações mencionadas estão devidamente listadas no glossário de siglas.

No entanto, algumas informações provenientes de determinadas colunas não podem ser detalhadas neste trabalho devido a questões de sigilo. Contudo, há dados que dispensam essa cautela, como, por exemplo, a espessura do Aço Silício utilizado, ou a presença de Trafo Série ou Reator Auxiliar no transformador fabricado. Esses elementos, juntamente com outras informações, compõem a base de dados que será processada e empregada para o treinamento do modelo de IA. Destaca-se que o Aço Silício é um material amplamente empregado na fabricação do núcleo de transformadores, justificando a viabilidade de expor os dados associados a esse componente, conforme ilustrado na figura 3.

Figura 3 – Bobina de aço-silício.

Fonte: https://www.wssa.com.hk/produto/A%C3%A7o%20Galvanizado/5.

3.2 Tratamento dos Dados

3.2.1 Manipulação

Com intuito de obter uma melhor amostra de dados, foi-se utilizado como principal ferramenta de desenvolvimento a linguagem de programação Python. Para facilitar e otimizar esse processo, foi-se utilizado o interpretador Jupyter Notebook, do ambiente virtual Anaconda.

Dentro do Jupyter Notebook, foi possível utilizar a linguagem Python de maneira organizada e estruturada, uma vez que, dentro da ferramenta é possível executar diferentes blocos de códigos na ordem de sua preferência, sendo perfeito para testes, visualizações gráficas e tratamento de dataframes.

Figura 4 – Trecho do código em Jupyter Notebook.

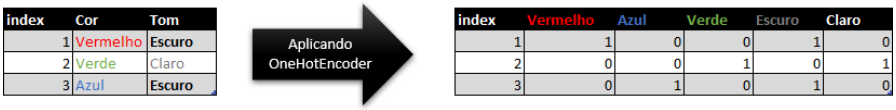
Fonte: Elaborado pelo autor, 2023.

Entretanto, criar algoritmos totalmente do zero para atividades de tratamentos de dados e visualizações gráficas não é uma tarefa trivial. Por esse motivo, o Python é tão recomendado para essa finalidade. Essa linguagem oferece diversos suportes por meio da utilização de bibliotecas, que otimizam de maneira significativa diversos processos de criação de algoritmo.

A biblioteca Python utilizada nesse trabalho, com a finalidade de importar os arquivos, realizar checagem e manipular as tabelas, foi a biblioteca Pandas. Essa biblioteca foi essencial para verificação de dados nulos/faltantes, visualização de métricas básicas e checagem de outliers.

A base de dados utilizada, continha em si dados qualitativos e dados quantitativos, por esse motivo foi-se necessário a utilização da biblioteca OneHotEncoder. Essa biblioteca, tem o propósito de converter os dados qualitativos em dados quantitativos. Tendo em vista, que as equações dos métodos de regressão, necessitam de atributos quantitativos para realizar os cálculos de maneira devida.

Figura 5 – Processo OneHotEncoder.



Fonte: Elaborado pelo autor, 2023.

Além disso, foi-se utilizada a biblioteca MinMaxScaler, com o intuito de reformular os dados de cada coluna da base de dados. Essa biblioteca, utiliza cálculos de normalização para cada dado da tabela. Desta forma, a base de dados tende a ter informações menos tendenciosas devido à escala de cada atributo, fazendo com que o método de regressão não adicione peso a dados de uma coluna por ser muito maiores que os dados de outra coluna. Entretanto, apesar de aplicar a normalização, que de fato modifica o valor dos dados, não há alteração nos parâmetros estatísticos das informações gerais das colunas.

É empregada a fórmula 3.1 para calcular o MinMaxScaler. Nesse processo, cada dado é normalizado ao ser subtraído pelo valor mínimo da coluna correspondente e dividido pela diferença entre o valor máximo e mínimo dessa mesma coluna. O processo é exemplificado na figura 6.

$$x' = \frac{x - min(x)}{max(x) - min(x)}. \tag{3.1}$$

Figura 6 – MinMaxScaler.

Fonte: Elaborado pelo autor, 2023.

Por outro lado, optou-se por incorporar outra técnica conhecida como escore padrão, ou escore z, ao longo do projeto. Essa abordagem demanda a utilização do desvio padrão e da média dos dados de cada atributo na base de dados. A fórmula para realizar a padronização é

explicitada na equação 3.2. Esse cálculo foi predominantemente aplicado nas planilhas de Excel, especialmente nos dados sensíveis das bases de dados, com o intuito de mascará-los.

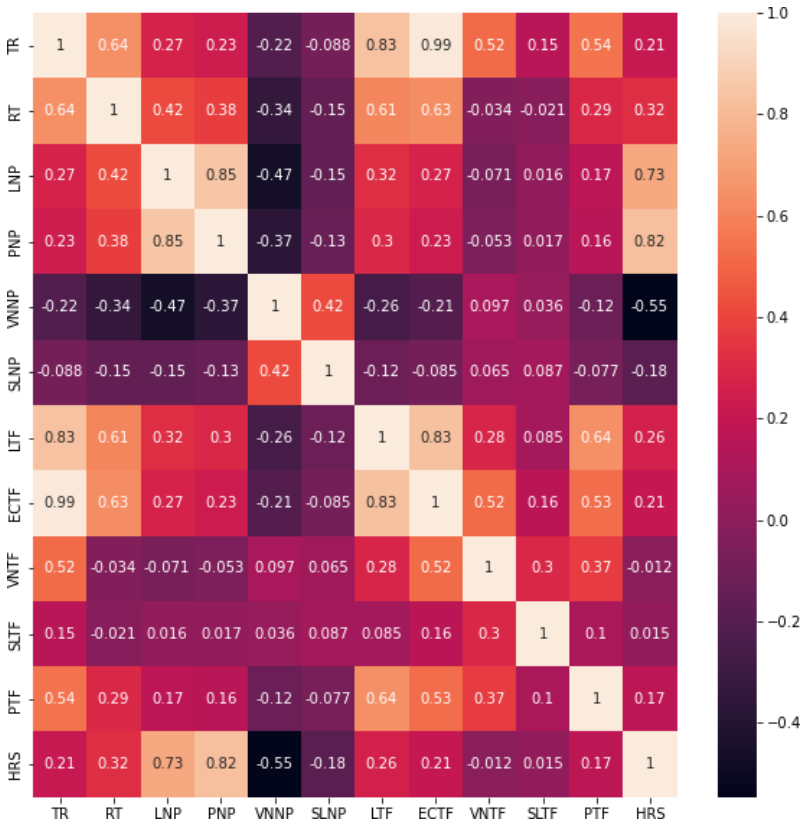
$$z = \frac{x - \mu}{\sigma}. \tag{3.2}$$

3.2.2 Verificação

Com o tratamento dos dados quase concluídos, foi necessário apenas certificar o grau de relevância entre os dados, utilizando a função HeatMap da biblioteca Seaborn.

Com essa biblioteca, é possível realizar uma matriz comparativa, para verificar o grau de relação, entre cada variável do dataset, sendo delimitado de -1 a 1, onde -1 é correlação negativa e 1 correlação positiva. 0 é a ausência de correlação.

Figura 7 – HeatMap correlação das variáveis.



Fonte: Elaborado pelo autor, 2023.

Após analisar os atributos das tabelas, por meio do gráfico gerado, é possível definir se de fato todas as colunas existentes devem se manter como dados de entrada. Para isso, foi-se necessário verificar a relação de cada coluna, relativa as características do núcleo do transformador, com a coluna designada como alvo da predição (horas de fabricação).

3.3 Variáveis de entrada e saída

3.3.1 Criação

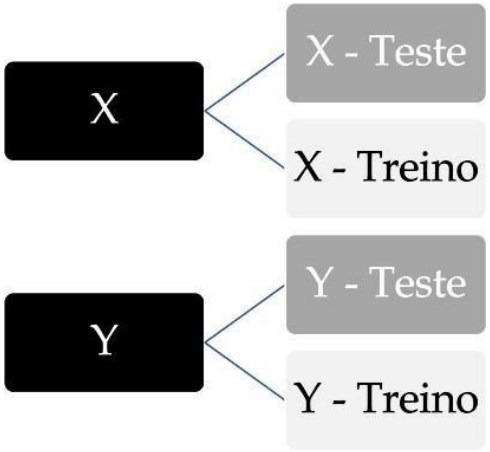
Em seguida, foi-se realizada a criação de duas variáveis, uma delas carrega consigo os dados de entrada (variável *X*), referentes as características do núcleo do transformador, e a outra carrega consigo os dados definidos como saída (variável *Y*), no caso, as horas gastas para a fabricação de cada núcleo do transformador. A biblioteca utilizada para a criação das variáveis citadas, foi a biblioteca Pandas.

3.3.2 Separação em Teste e Treino

Assim que ambas a variáveis (*X* e *Y*) são criadas, é necessário aplica-las a função train_test_split, da biblioteca sklearn.model_selection, que realiza o processo de mistura dos dados de forma aleatória e os aplica nas variáveis de treino e teste, para que possam ser utilizadas no treinamento e nos testes de desempenho dos métodos de predição.

Ao utilizar a função train_test_split, da biblioteca sklearn.model_selection, fez-se necessário a aplicação de dois outros parâmetros. O primeiro, refere-se à proporção de dados que serão utilizados para treino e dos dados que serão utilizados para teste, no algoritmo utilizado nesse trabalho foi-se utilizado a proporção de 80% dos dados para treino e 20% para teste. O segundo parâmetro, se refere ao valor conhecido como semente, que define como os dados serão aleatorizados.

Figura 8 – Divisão das variáveis.



Fonte: Elaborado pelo autor, 2023.

Após as variáveis de treino e teste serem definidas, foram selecionados os métodos de regressão utilizados na execução dos cálculos de predição. Com a intenção de adquirir o melhor resultado possível, foram testados o desempenho de três métodos diferentes conhecidos como

Random Forest Regressor, Support Vector Machine (SVM) e Rede Neural Multi-Layer Perceptron (RNA MLP).

3.4 Métodos de regressão

3.4.1 Random Forest Regressor

O método *Random Forest Regressor* consiste em realizar a síntese de diversas árvores de decisão durante o processo do treinamento, como mostrado na figura 9. Esse método é derivado do método denominado Árvore de Decisão, tendo em vista que o princípio de estruturação das árvores é semelhante. Porém, o *Random Forest* se diverge do método das Árvores de Decisão por não utilizar a base de dados por inteiro e apenas amostras de forma aleatória. A criação aleatória de cada árvore de decisão é um processo importante, pois em cada uma haverão ramos com nós nas extremidades compostos pelas melhores variáveis, definidas pelo cálculo de ganho da amostragem atual, fazendo com que cada árvore tenha divergências das demais, evitando possíveis *overfitting* (sobreajuste dos dados, que pode levar pouca robustez do modelo). Após a criação de todas as árvores de decisão, o algoritmo está pronto, ao utiliza-lo para previsão dos dados, cada árvore dará um resultado diferente e como resultado final será dado a média desses resultados.

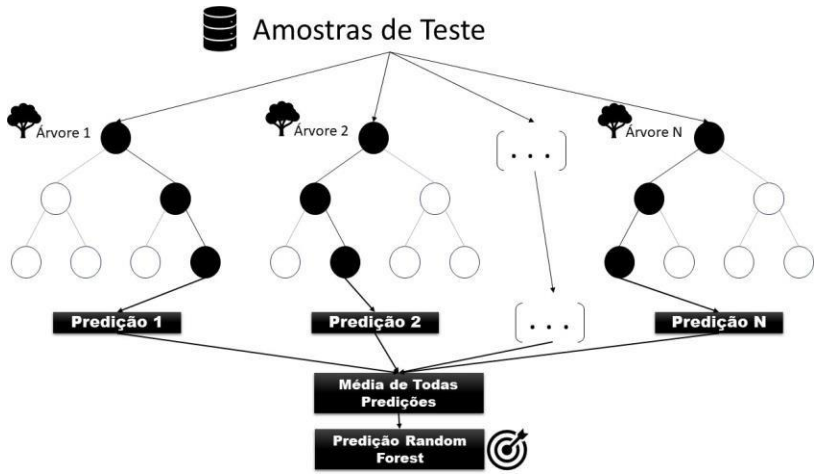
As fórmulas 3.3 e 3.4 desempenham funções cruciais no cálculo realizado durante a construção de árvores de decisão. A Equação 3.3 representa a entropia de um conjunto de dados, medindo a impureza do conjunto por meio da probabilidade de ocorrência de eventos específicos. Por outro lado, a Equação 3.4 descreve o cálculo do ganho de informação em um nó de decisão, sendo utilizado para determinar a eficácia de divisões nos dados ao avaliar a entropia do nó pai e a soma ponderada das entropias dos nós filhos.

Essas formulações são ferramentas essenciais para a construção de árvores de decisão, contribuindo significativamente para a avaliação da qualidade das divisões nos dados e facilitando a tomada de decisões no processo. As referências associadas a cada equação fornecem os detalhes completos para consultas mais aprofundadas sobre os cálculos específicos.

$$H(x) = -\sum_{i=1}^{n} p(x_i) log_2 p(x_i), \tag{3.3}$$

$$ganho = —Entropia(pai) - \sum_{i=1}^{n} peso(filho_i) * entropia(filho_i). \tag{3.4}$$

Figura 9 – Arvores de decisão.

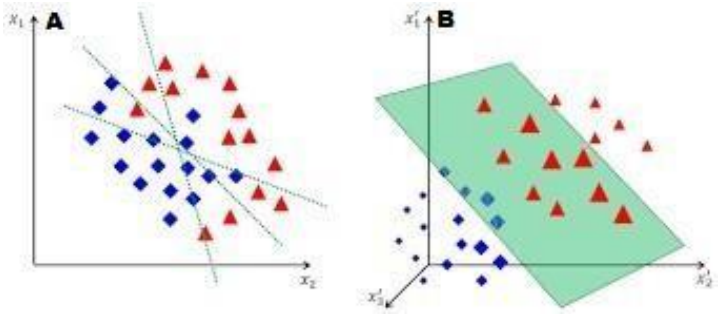


Fonte: Elaborado pelo autor, 2023.

3.4.2 Support Vector Machine

O *Support Vector Machine* (SVM) tem como principal modo operante a criação de um hiperplano que faz a separação dos dados distribuindo-os em um espaço contínuo de um plano cartesiano. O diferencial encontrado nesse método, é que ele não se limita em traçar uma reta diante da distribuição de dados em um plano comum, como é o caso de uma regressão linear padrão. Na Figura 10 é possível ver que no gráfico A não era possível separar os dados com uma reta, mas adicionando um hiperplano como no gráfico B essa ação se tornou trivial.

Figura 10 – Separação dos dados por um hiperplano



Fonte: (Maselli and Negri, 2019).

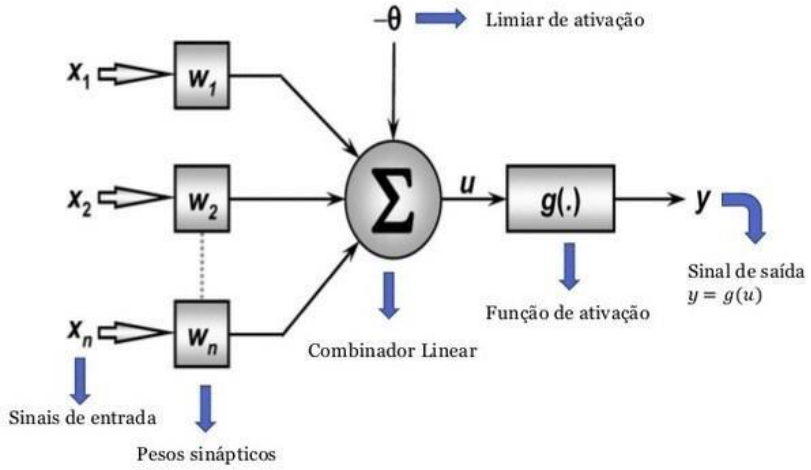
3.4.3 Rede Neural Artificial - *Multi-Layer Perceptron*

O Multi-Layer Perceptron (MLP) é uma rede neural que possui um número indeterminado de neurônios e pode conter de uma a várias camadas ocultas. Esse método é derivado do Perceptron, que de forma distinta, utiliza apenas uma camada oculta. As camadas ocultas da rede neural, são utilizadas apenas como parte do processo de previsão dos dados, tendo em vista, que não é possível obter o resultado definitivo por meio delas, essa função é destinada a camada de

saída. De maneira geral, as redes neurais MLP tem como estrutura base a camada de entrada, as camadas ocultas e a camada de saída.

A equação 3.5 descreve o cálculo do potencial de ativação em um contexto de Rede Neural Artificial (RNA), especificamente em uma camada de um Perceptron de Múltiplas Camadas (MLP). Nessa equação, u representa o potencial de ativação, w_i denota os pesos associados às entradas x_i e θ é o limiar de ativação.

Figura 11 – Perceptron.



Fonte: https://embarcados.com.br/rede-perceptron-de-uma-unica-camada/.

$$u = \sum_{i=1}^{n} w_i * x_i - \theta. \tag{3.5}$$

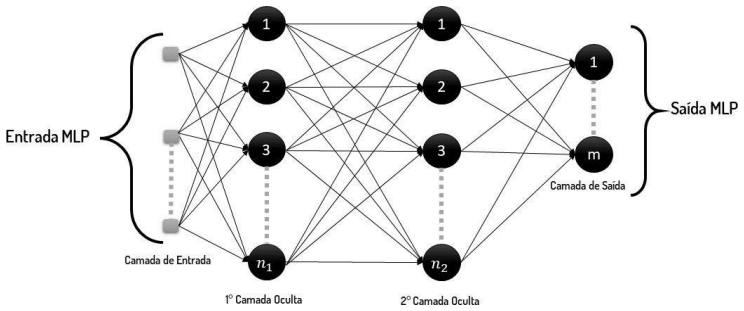
A Figura 12 ilustra o esquema de funcionamento das camadas ocultas de uma Rede Neural Artificial (RNA). De maneira geral, n_1 refere-se à primeira camada oculta, n_2 à segunda camada oculta, e m à camada de saída.

Para uma compreensão mais detalhada do funcionamento da RNA, é essencial destacar que cada camada oculta (n_1 e n_2) contém neurônios que processam informações a partir das entradas da rede. Esses neurônios são conectados por pesos sinápticos, e cada conexão representa a influência de uma entrada sobre um neurônio específico.

A camada de saída (m), por sua vez, representa a parte final da rede onde os resultados são gerados. Cada neurônio nesta camada está associado a uma classe ou valor de saída. O processo de treinamento da rede visa ajustar os pesos sinápticos de modo a otimizar a predição ou classificação realizada pela camada de saída.

Assim, a Figura 12 proporciona uma visão esquemática da estrutura da RNA, destacando as camadas ocultas e de saída, fundamentais para o processamento e a geração de resultados da rede.

Figura 12 – Diagrama MLP.



Fonte: Elaborado pelo autor, 2023.

O algoritmo subjacente ao MLP, conhecido como *backpropagation*, consiste em quatro passos principais. No primeiro passo, ocorre a Inicialização, onde os pesos da rede são atribuídos valores iniciais. No segundo passo, a Ativação, as entradas são propagadas através da rede, passando pelas camadas ocultas até a camada de saída, com a aplicação de funções de ativação nos neurônios. No terceiro passo, ocorre o Treinamento dos Pesos, onde é utilizado um conjunto de dados de treinamento para ajustar iterativamente os pesos sinápticos, visando minimizar a diferença entre as saídas da rede e os resultados desejados. Por fim, no quarto passo, a Iteração, o processo de treinamento é repetido até que a rede alcance um desempenho aceitável, medido por critérios pré-definidos.

Assim, o algoritmo de *backpropagation* realiza uma série de iterações, ajustando continuamente os pesos da rede para otimizar o aprendizado e a capacidade preditiva do modelo. Essa abordagem é fundamental para a eficácia do MLP em tarefas de aprendizado supervisionado.

3.5 Utilização dos métodos de regressão

Com o intuito de utilizar os métodos de IA, supracitados, foi utilizada a biblioteca *Scikit-Learn*. Dentro dessa biblioteca, existem inúmeros módulos que podem ser importados. Como por exemplo, o método *Random Forest Regressor* é referente ao módulo sklearn.ensemble, já o módulo sklearn.svm é referente ao método *Support Vector Machine* e por fim o método de RNA MLP é utilizado através do módulo sklearn.neural_network.

As vantagens de utilizar os módulos dessa biblioteca está ligada a otimização e parametrização das funções, é possível testar diversos parâmetros diferentes de forma rápida. Desse modo, encontrando os parâmetros ideais para cada situação proposta. Tal como, no método *Random Forest* foram modificados os parâmetros “n_estimators” que é utilizado para definir o número de árvores que serão usadas para calcular os resultados, “min_samples_split” que define o número mínimo de amostras necessárias para dividir os nós internos e “n_jobs” que é usado para selecionar o número de CPUs para execução. Já no caso do método SVM, foi selecionado

apenas o parâmetro do método “Kernel” que serve para definir qual a função a ser utilizada. Por fim, os parâmetros passados no Método de Rede Neural MLP, foram o número de camadas internas “hidden_layer_sizes” e o “max_iter” que serve para definir o número de iterações que devem ser realizadas.

3.6 Hiper parametrização

Assim que os parâmetros de interesse são identificados, é possível realizar um processo denominado de hiper parametrização, que consiste em testar diversas combinações de parâmetros em busca de um resultado ótimo. Existem bibliotecas em Python que podem ser utilizadas para realizar esse processo, porém dependendo das versões das bibliotecas os resultados podem ser incompatíveis. Portanto, foi necessário a criação de um pequeno algoritmo, composto por dois laços e uma condicional para realizar a hiperparametrização.

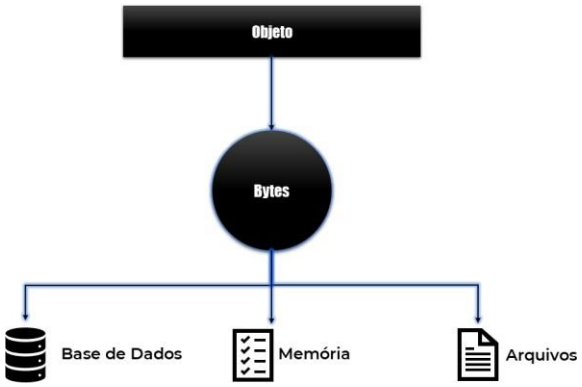
Após a hiperparametrização, o melhor método de regressão entre os três citados foi o *Random Forest Regressor*. O método escolhido.

3.7 Desenvolvimento e estruturação da IDE

3.7.1 Arquivo salvo da IA treinada

Com todos os parâmetros definidos e o método escolhido, foi necessário utilizar a biblioteca Joblib. Essa biblioteca serve para salvar as IAs treinadas, auxiliando e facilitando nas aplicações futuras. Além de disso, a biblioteca armazena o arquivo na extensão pickle que realiza o processo de serialização, demonstrado na figura 13, transformando-os em sequência de bytes e por fim em arquivo.

Figura 13 – Serialização.

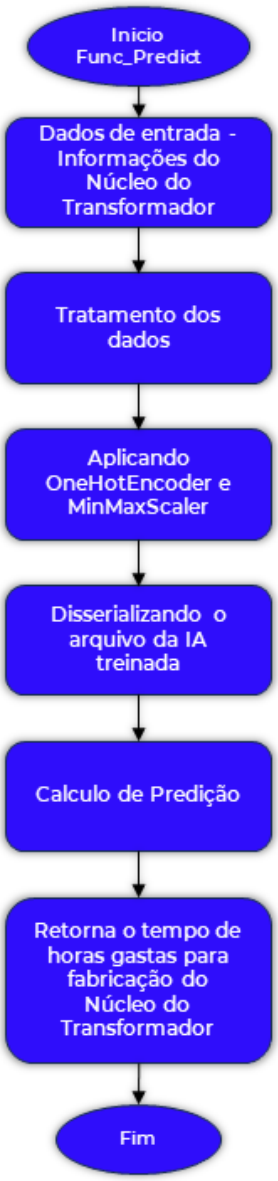


Fonte: Elaborado pelo autor, 2023.

3.7.2 Função de aplicação da IA

Após todo o processo de tratamento dos dados, treinamento, validação e salvamento da IA, foi-se realizado a transferência dos algoritmos em Jupyter Notebook (.ipynb) para a IDE Pycharm, onde os arquivos foram recriados em formato Python (.py). Esse processo foi necessário, pois normalmente os arquivos em Jupyter Notebook são melhores para testar e realizar alterações nos scripts, por outro lado os arquivos Python são melhores para realizar aplicações e interações com outros códigos. Deste modo, todo o script criado em Notebook foi transformado e convertido em uma única função (denominada Func_Predict), com algumas alterações que serviriam para captar novas entradas de dados, como por exemplo um novo núcleo do transformador que fosse fabricado. O funcionamento dessa função pode ser visualizado no fluxograma da figura 14.

Figura 14 – Fluxograma Função "Func_Predict".

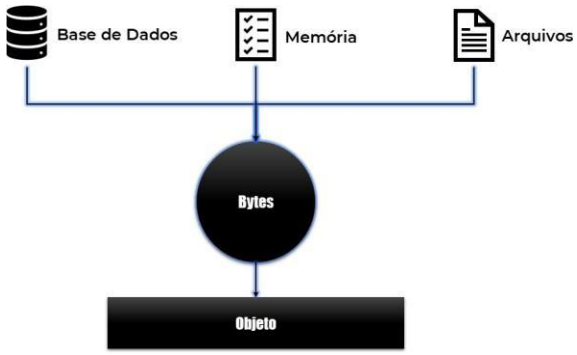


Fonte: Elaborado pelo autor, 2023.

3.7.3 Utilização da IA treinada

Assim que o código ficou pronto, foi possível receber novos dados e aplica-los no arquivo salvo da IA treinada. Porém, para utilizar a IA treinada foi necessário aplicar a biblioteca Joblib, para que houvesse a desserialização, exemplificado no esquema da figura 15, e conversão do arquivo em objeto novamente.

Figura 15 – Desserialização.



Fonte: Elaborado pelo autor, 2023.

3.7.4 Criação de uma função de conferência

Após a função Func_Predict ser criada, foi necessário testar os valores retornados pela IA. Desta forma, criou-se uma função, denominada Func_State, que comparava os valores de entrada com a base de dados e logo em seguida essa mesma função retornava uma lista com os dados de todos os transformadores que continham o núcleo semelhante aos do transformador referente ao dado de entrada. Por fim, a função retornava a média, o valor máximo, o valor mínimo e o desvio padrão da hora gasta de fabricação do núcleo de todos os transformadores da lista. O fluxograma da função Func_State pode ser visualizado na figura 16 . Essas saídas, foram geradas com o intuito de auxiliar os usuários na conferência dos dados retornados pela IA. Com isso, ajudando também na tomada de decisão.

Figura 16 – Fluxograma Função "Func_State".

Fonte: Elaborado pelo autor, 2023.

3.7.5 Função Final

Por fim, criou-se um último algoritmo, denominado IA_predict, que serviria como base de funcionamento para a criação da IDE feita através do Excel. Esse algoritmo, reunia as funções supracitadas (Func_Predict e Func_Mean) e retornava o valor retornado por elas direto em um arquivo Excel. Esse processo, só foi possível devido a utilização da Biblioteca Openpyxl que serve principalmente para realizar ações diretamente em arquivos Excel utilizando comandos em

Python. O processo realizado pelo algoritmo IA_predict, pode ser visualizado no fluxograma da figura 17.

Figura 17 – Fluxograma Função "IA_Predict".

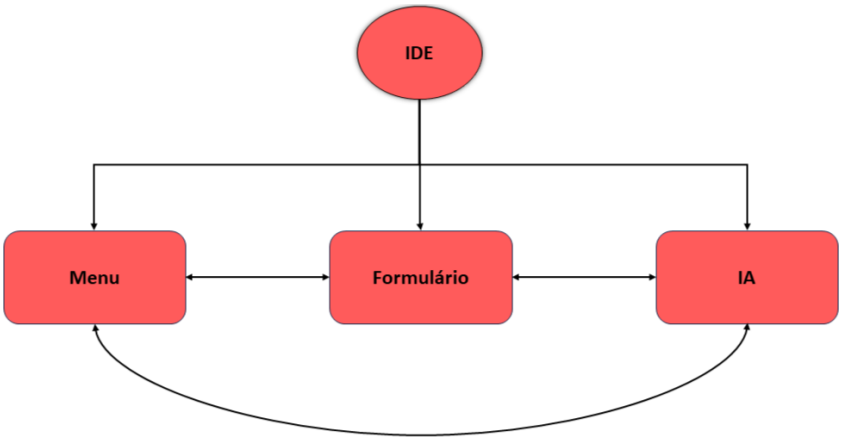
Fonte: Elaborado pelo autor, 2023.

3.7.6 Criação da IDE

Logo que o algoritmo IA_predict foi desenvolvido, tornou-se possível criar a IDE de acesso do usuário. Primeiramente, começou-se a criação das macros utilizando VBA para o envio das informações até a planilha que armazena os dados de entrada. Em seguida, criou-se um novo script em VBA que possibilitava a execução dos códigos em Python. Com ambas as

macros criadas, bastou adiciona-las em botões, criados dentro do Excel, para que houvesse a ativação. Ao final, foram desenvolvidos os designers e configurações finais da IDE. A estrutura final da IDE contou com 3 abas, o menu, a aba de preenchimento das informações e a aba de visualização do resultado gerado. A estrutura está exemplificada na figura 18.

Figura 18 – Fluxograma IDE.



Fonte: Elaborado pelo autor, 2023.

4 RESULTADOS

4.1 Avaliação da base de dados

Como ponto inicial, procedeu-se à coleta e organização dos dados para a síntese da base, a qual foi objeto de estudo no decorrer do projeto. Esse procedimento frequentemente requer um investimento de tempo considerável, dada a necessidade de localizar fontes apropriadas e organizá-las de maneira eficaz. No intuito de garantir a qualidade dessas atividades, conforme mencionado na seção metodológica, utilizou-se a ferramenta Excel. Essa abordagem resultou na seguinte estruturação da base de dados.

Tabela 1 – Estrutura Base de Dados

Colunas	Tipo	Finalidade
Transformador	Varchar(15)	Chave Primaria
TR	Bool	Dados de Entrada
RT	Bool	Dados de Entrada
TP	Varchar(5)	Dados de Entrada
LNP	Float	Dados de Entrada
PNP	Int	Dados de Entrada
FC	Varchar(5)	Dados de Entrada
VNNP	Bool	Dados de Entrada
SLNP	Bool	Dados de Entrada
TIF	Varchar(5)	Dados de Entrada
LTF	Float	Dados de Entrada
ECTF	Float	Dados de Entrada
VNTF	Bool	Dados de Entrada
SLTF	Bool	Dados de Entrada
PTF	Float	Dados de Entrada
HRS	Float	Dado Alvo

O resultado da tabela 1 foi obtido a partir de uma base de dados com mais de 232 colunas. Foram necessários alguns estudos em conjunto com os funcionários mais experientes da empresa, para identificação e seleção das variáveis, que representavam as características do transformador, que mais tinham influência sobre o tempo de confecção dos diversos núcleos fabricados.

Com os dados organizados, foi necessário a aplicação de técnicas de tratamento de dados para verificar possíveis problemas. Desta forma, foi usado como ferramentas as bibliotecas disponíveis na linguagem Python, utilizando como interpretador o Jupyter Notebook. Após a aplicação dos tratamentos, identificou-se que a base estava pronta para ser utilizada.

Após a análise do quadro 1, que foi gerado durante a execução do código, identificou-se que não haviam linhas contendo dados nulos entre as colunas da base de dados e que os dados que possuíam o tipo “Bool” e “Varchar” foram redefinidos para “int64” e “object”, respectivamente. Esse resultado, serviu demonstrou que todas as linhas que possuíam dados nulos foram preenchidas com zero, ou apagadas conforme a situação e que os tipos de dados foram alterados

com o intuito de serem otimizados, passando de 5 tipos de variáveis para 3 tipos.

Quadro 1 – Valores nulos

Coluna	Tipo
Transformador	object
TR	int64
RT	int64
TP	object
LNP	float64
PNP	float64
FC	object
VNNP	int64
SLNP	int64
TIF	object
LTF	float64
ECTF	float64
VNTF	int64
SLTF	int64
PTF	float64
HRS	float64

Além de apresentar as informações referentes à quantidade de dados nulos, foi possível realizar uma análise da natureza do conjunto de dados em cada coluna. Assim, identificamos a presença de dados qualitativos e quantitativos. Os dados quantitativos são reconhecíveis quando o tipo da coluna é indicado como "float64"(valores reais) ou "int64"(valores inteiros), enquanto os dados qualitativos são identificados pelo tipo "object".

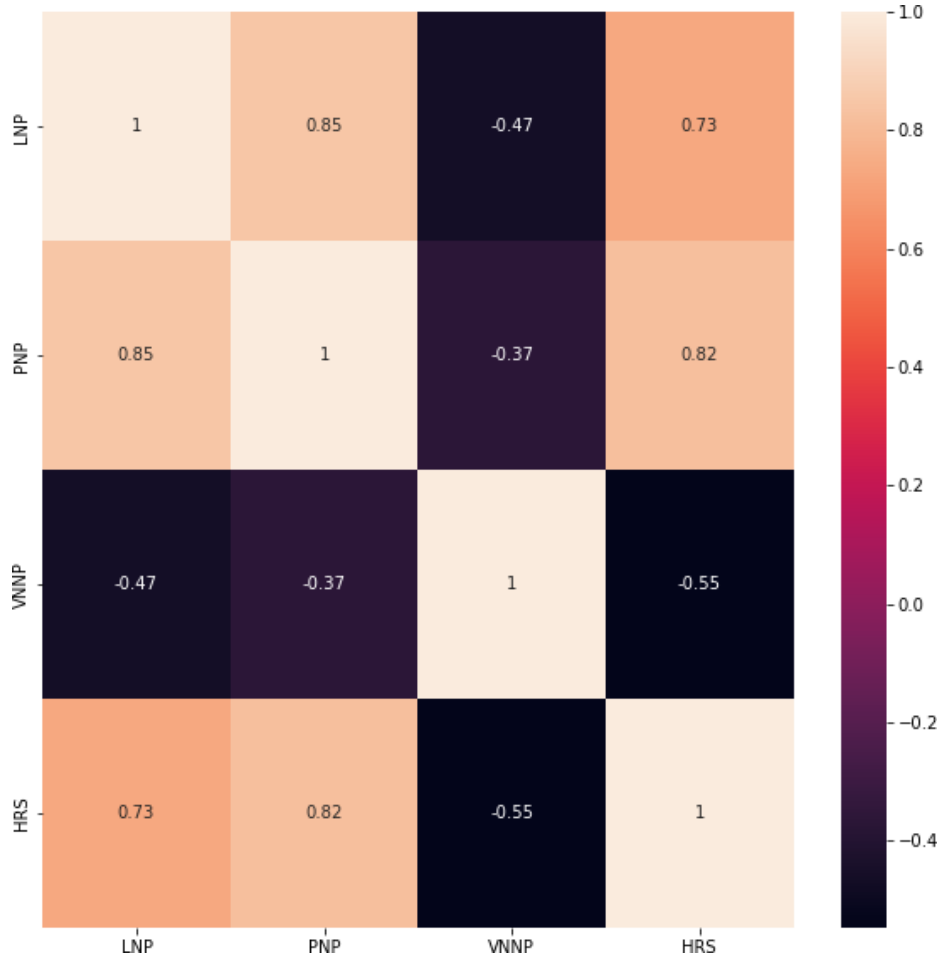
Por motivo de sigilo, é importante ressaltar que todos os dados foram normalizados, resultando em valores que variam de -1 a 1. Adicionalmente, elaborou-se uma segunda tabela que oferece informações estatísticas específicas para os dados quantitativos. Nessa Tabela 2, encontram-se detalhes como média, mediana, valor mínimo, valor máximo, entre outros, proporcionando uma visão mais aprofundada das características desses dados.

A forma de avaliação dos dados citada na metodologia foi por meio do *Heatmap* de correlação. Analisando o gráfico da figura 7, foi possível observar que os dados com maior correlação com a variável alvo foram as variáveis: LNP, PNP e VNNP. Essas informações podem ser vistas no gráfico da figura 20.

Tabela 2 – Sobre valores quantitativos

Colunas	Qtd.	Média	Desv.Pad.	Min	1° Quartil	Mediana	3° Quartil	Max
TR	905	0.120	0.326	0.000	0.000	0.000	0.000	1.000
RT	905	0.117	0.322	0.000	0.000	0.000	0.000	1.000
LNP	905	0.000	1.000	-2.746	-0.613	-0.225	0.613	3.750
PNP	905	-0.000	1.000	-1.061	-0.596	-0.364	0.200	5.239
VNNP	905	0.715	0.452	0.000	0.000	1.000	1.000	1.000
SLNP	905	0.305	0.461	0.000	0.000	0.000	1.000	1.000
LTF	905	0.000	1.000	-0.307	-0.307	-0.307	-0.307	9.941
ECTF	905	-0.000	1.000	-0.365	-0.365	-0.365	-0.365	3.293
VNTF	905	0.036	0.188	0.000	0.000	0.000	0.000	1.000
SLTF	905	0.003	0.056	0.000	0.000	0.000	0.000	1.000
PTF	905	-0.000	1.000	-0.199	-0.199	-0.199	-0.199	14.716
HRS	905	-0.000	1.000	-1.080	-0.610	-0.317	0.424	5.819

Figura 19 – *Heatmap*com melhores correlações.



Fonte: Elaborado pelo autor, 2023.

Por outro lado, elas se diferem no modo de se correlacionar com a variável alvo, como apresentado na figura 19. No caso das variáveis PNP e VNNP, o tipo de correlação pode ser

definido como positivamente correlacionado, pelo fato dos valores se aproximarem de um. Entretanto, a variável LNP possui o valor mais próximo de menos um, o que a caracteriza como negativamente correlacionado. Essa informação, pode ser melhor visualizada no quadro 2.

Quadro 2 – Correlação

Coluna	Correlação com HRS	Nível	Correlação
LNP	-0,55	Razoavelmente forte	Negativa
PNP	0,82	Muito forte	Positiva
VNNP	0,73	Forte	Positiva

Após o tratamento dos dados, a análise de correlação e verificação dos tipos das variáveis, foi realizada a conversão dos dados qualitativos para quantitativos, utilizando o método OneHotEncoder dentro do Python, como mencionado na metodologia. Desse modo, obteve-se uma modificação na representação dos dados presentes nas colunas TP, FC e TIP. Como a base contém quase 1000 linhas, foi retirada apenas uma amostra dos dados da tabela, afim de representar o processo. O resultado pode ser visualizado na tabela 3.

Como representado na tabela 3, é possível perceber que as colunas que possuíam dados qualitativos foram dividas em mais colunas que representam o número de dados distintos contidos nela. Como por exemplo, a coluna TP foi dividida em TP_1, TP_2, TP_3, TP_4, TP_5, TP_6, TP_7 e TP_8, pois nela haviam 8 tipos de dados diferentes. Um exemplo desse processo pode ser visto na figura 5.

Como processo final de preparação da base dados, bastou apenas retirar a coluna de transformadores, que obtinha a função de index na tabela, uma vez que não é utilizada.

4.2 Variáveis de entrada e saída

Com a base de dados pronta para utilização, foi realizado o processo de separação dos dados, onde os dados de entradas foram atribuídos à variável X e os dados de saída foram atribuídos a variável Y. Como pode ser visualizado no quadro 3.

Assim que as variáveis são definidas, é possível segmenta-las em dados de teste e dados de treino. Esse processo pode ser visualizado por meio da Figura 8. Após a separação das variáveis de teste e treino, conforme os parâmetros do quadro 4, foi possível prosseguir para o processo de seleção do método de regressão, que foi utilizado posteriormente para a predição do valor da variável alvo.

4.3 Definição dos parâmetros

Para definir o melhor método de regressão, foram testados três métodos, conhecidos como SVM, RNA e Random Forest, citados e explicados brevemente no capítulo 3. Com o intuito de avaliar os métodos, foi necessário aplicar os dados de treino em cada um deles. Os parâmetros

Tabela 3 – Dados colunas por transformadores

	T194	T195	T196	T197	T198	T199	T200
TR	1	0	0	0	0	1	0
RT	0	0	0	0	0	1	0
TP_1	0	1	1	1	1	0	1
TP_2	0	0	0	0	0	1	0
TP_3	1	0	0	0	0	0	0
TP_4	0	0	0	0	0	0	0
TP_5	0	0	0	0	0	0	0
TP_6	0	0	0	0	0	0	0
TP_7	0	0	0	0	0	0	0
TP_8	0	0	0	0	0	0	0
LNP	0.802700	-0.418825	0.744532	-0.127986	-0.709665	0.744532	0.822089
PNP	-0.186327	-0.409725	0.777495	0.111058	-0.686177	1.193.936	0.711724
FC_1	1	1	1	1	1	0	1
FC_2	0	0	0	0	0	0	0
FC_3	0	0	0	0	0	0	0
FC_4	0	0	0	0	0	1	0
VNNP	0	1	1	1	1	0	1
SLNP	0	0	1	0	0	0	0
TIF_1	0	1	1	1	1	0	1
TIF_2	0	0	0	0	0	0	0
TIF_3	0	0	0	0	0	1	0
TIF_4	1	0	0	0	0	0	0
TIF_5	0	0	0	0	0	0	0
LTF	2.104.537	-0.306724	-0.306724	-0.306724	-0.306724	1.682.566	-0.306724
ECTF	2.457.248	-0.365071	-0.365071	-0.365071	-0.365071	2.039.127	-0.365071
VNTF	0	0	0	0	0	0	0
SLTF	0	0	0	0	0	0	0
PTF	0.431827	-0.198927	-0.198927	-0.198927	-0.198927	0.489168	-0.198927
HRS	0.168322	-0.337338	0.602651	0.029622	-0.473660	1.163.791	0.630391

alterados e passados em cada um deles durante o processo de treinamento, podem ser vistos no quadro 5.

A partir dos parâmetros passados para a execução do treinamento de cada método de regressão, foi necessário definir as técnicas de avaliação de desempenho para que pudesse ser mensurado a qualidade dos métodos treinados.

4.4 Avaliação dos métodos de regressão

4.4.1 Desempenho

Portanto, normalmente para avaliação dos métodos de regressão, com algoritmos de IA, são utilizados a métrica *R*, *Mean Absolut Error* (MAE), *Mean Squared Error* (MSE) e como proposta de melhora na avaliação da métrica *R* é usada a *R* ajustado [O Kramer, 2016].

Quadro 3 – Dados *X* e *Y*

Variável	Coluna
X	TR
X	RT
X	TP_1
X	TP_2
X	TP_3
X	TP_4
X	TP_5
X	TP_6
X	TP_7
X	TP_8
X	LNP
X	PNP
X	FC_1
X	FC_2
X	FC_3
X	FC_4
X	VNNP
X	SLNP
X	TIF_1
X	TIF_2
X	TIF_3
X	TIF_4
X	TIF_5
X	LTF
X	ECTF
X	VNTF
X	SLTF
X	PTF
Y	HRS

Quadro 4 – Parâmetros

Parâmetros	Valores
Dados de entrada	Variável *X*
Dados de Saída	Variável *Y*
Distribuição	20% Teste / 80% Treino
Estado Aleatório	579

A *R*, também conhecido como coeficiente de determinação, serve para avaliar o percentual de variância dos dados que são explicados pelo modelo. Em outras palavras, ele mede o quanto a variabilidade nos dados de resposta é explicada pelo modelo de regressão. Essa métrica, é medida de forma adimensional de 0 a 1, onde 1 representa que o modelo se ajusta muito bem aos dados, enquanto o 0 significa que o modelo não se ajusta aos dados. A formula da métrica $R2$ é mostrada na equação (4.1).

Quadro 5 – Parâmetros dos métodos

Métodos de Regressão	Parâmetros	Valores
Random Forest	N° de Arvores de Decisão	1000
Random Forest	Folhas de amostra mínimos	1
Random Forest	N° de Núcleos	-1 (Todos)
Random Forest	Estado Aleatório	579
Support Vector Machine	Kernel	RBF (Radial Basis Function)
Rede Neural Artificial	Máximo de Iterações	1000
Rede Neural Artificial	Camada ocultas	30

1. Y: valores reais
2. $\hat{Y}$: valores previstos
3. $\bar{Y}$: média dos valores reais

$$R^2_{(y,\hat{y})} = 1 - \frac{\sum^{n}_{i=1}(y_i - \hat{y}_i)^2}{\sum^{n}_{i=1}(y_i - \bar{y})^2}. \quad (4.1)$$

O MAE que tem a sua função de avaliar a média da diferença entre o valor predito e o valor real. Essa métrica não é afetada quando existe valores discrepantes na massa de dados de treino, o que significa que ela não acusa quando há *outliers*. A formula do MAE pode ser vista na equação (4.2).

$$MAE_{(y,\hat{y})} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \quad (4.2)$$

O MSE é usado, bem como o MAE, para verificar a diferença média entre o valore real e o valor predito. No entanto, o MSE consegue punir os valores que são discrepantes na massa de dados, gerando uma alteração nos resultados quando esses "outliers" estão presentes nos conjuntos de dados. Isso ocorre, pois o MSE eleva ao quadrado a diferença entre os dados, ao invés de utilizar o módulo do resultado entre Y e $\hat{Y}$, como é o caso do MAE. A formula do MSE pode ser vista na equação (4.3).

$$MSE_{(y,\hat{y})} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i) \quad (4.3)$$

A R2 ajustado é basicamente um método de avaliação baseado na R2, porém ele tem como forma de ajuste a penalização das features (colunas com dados de entrada) presentes na base de dados que não contribuam para os cálculos de predição realizados pelo modelo. A fórmula da R2 ajustado pode ser visualizada na equação (4.4).

$$R^2_a = 1 - \frac{(1 - R^2)(N - 1)}{N - p - 1}. \quad (4.4)$$

1. p: representa o número de features (dados de entrada do modelo) da equação 4.4
2. N: representa o número de amostras da equação 4.4
3. a: representa as amostras da equação 4.4

Com as técnicas de avaliação definidas, houve a comparação do desempenho de cada método. Todas as performances dos métodos podem ser acompanhadas pelo tabela 4, que mostra os resultados do R2, pelo tabela 5 que mostra os resultados do MSE, pelo tabela 6 que mostra os resultados do MAE e pelo tabela 7 que mostra o resultado do R2 ajustado.

Tabela 4 – Valores obtidos pela métrica R^2 para:

Método	R2
Random Forest	92.0%
SVM	91.0%
RNA	89.0%

Tabela 5 – Valores obtidos pela métrica *MSE* para:

Método	MSE
Random Forest	0.082
SVM	0.091
RNA	0.105

Tabela 6 – Valores obtidos pela métrica *MAE* para:

Método	MAE
Random Forest	0.201
SVM	0.195
RNA	0.210

Tabela 7 – Valores obtidos pela métrica R^2 ajustado para:

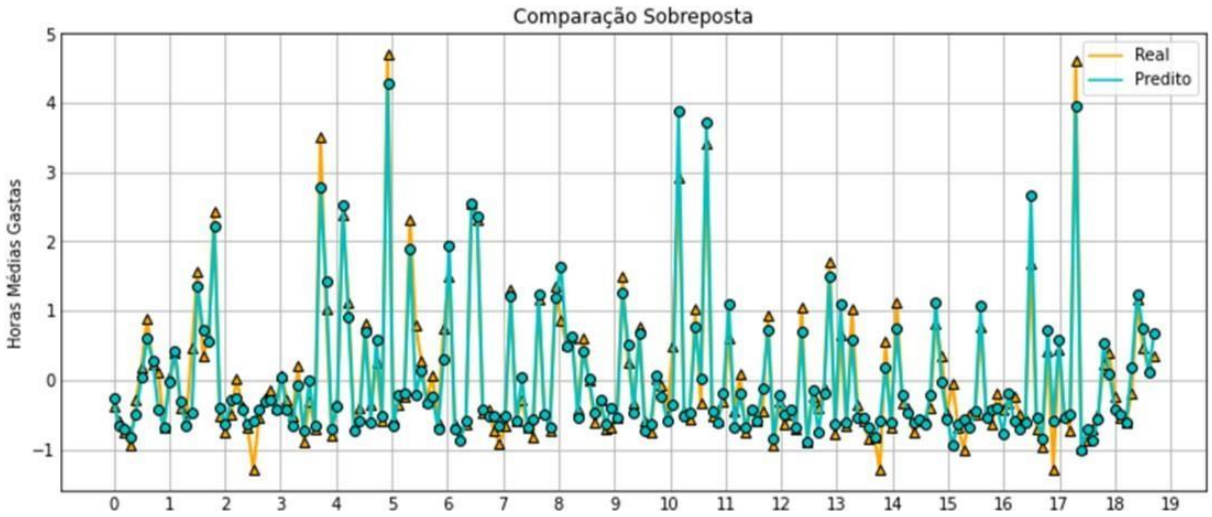
Método	R2 ajustado
Random Forest	90.24%
SVM	89.17%
RNA	87.57%

Após a análise dos resultados gerados pelos cálculos de desempenho, é possível afirmar que o método Random Forest se sobressai na maioria das avaliações, ficando apenas atrás do SVM no cálculo do Erro Absoluto Médio. O método que obteve os resultados piores foi o método de Rede Neural Artificial utilizando o *Multi-Layer Perceptron*. Entretanto, todos os métodos de maneira geral obtiveram resultados bons diante das métricas de desempenho.

4.4.2 Visualização gráfica

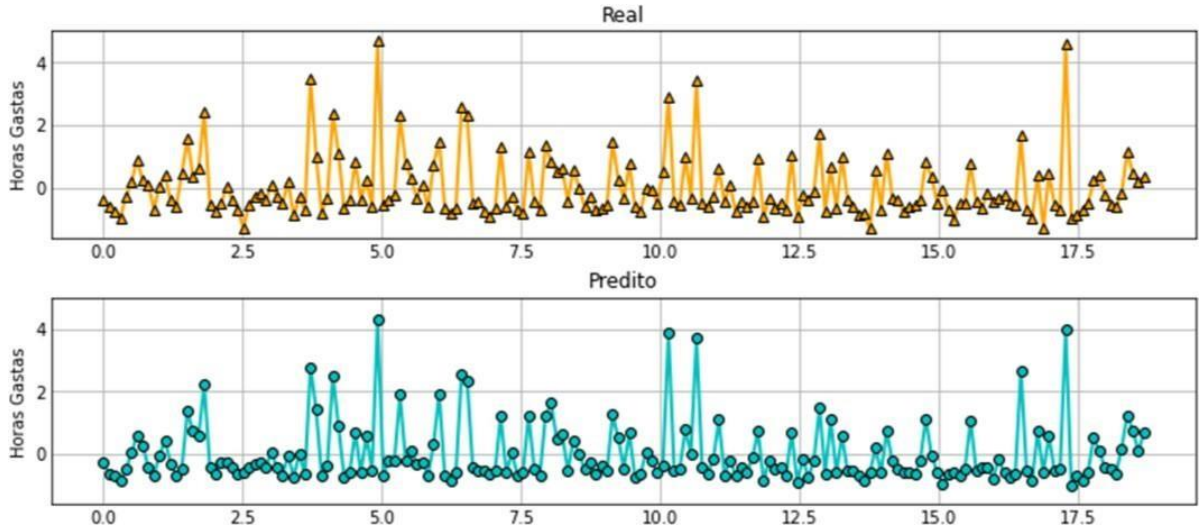
Para auxiliar na avaliação dos resultados das predições realizadas por cada método de regressão, foram criados gráficos com a utilização da biblioteca Matplotlib, por meio do Python, com os dados de saída (da amostra de teste) reais e preditos. Os gráficos podem ser visualizados pelas figuras 20, 21, 22, 23, 24 e 25. Todos os valores do eixo x representam a quantidade de dados e estão na ordem de $x10^{-1}$.

Figura 20 – Dados de saída do real e previsto - Random Forest.



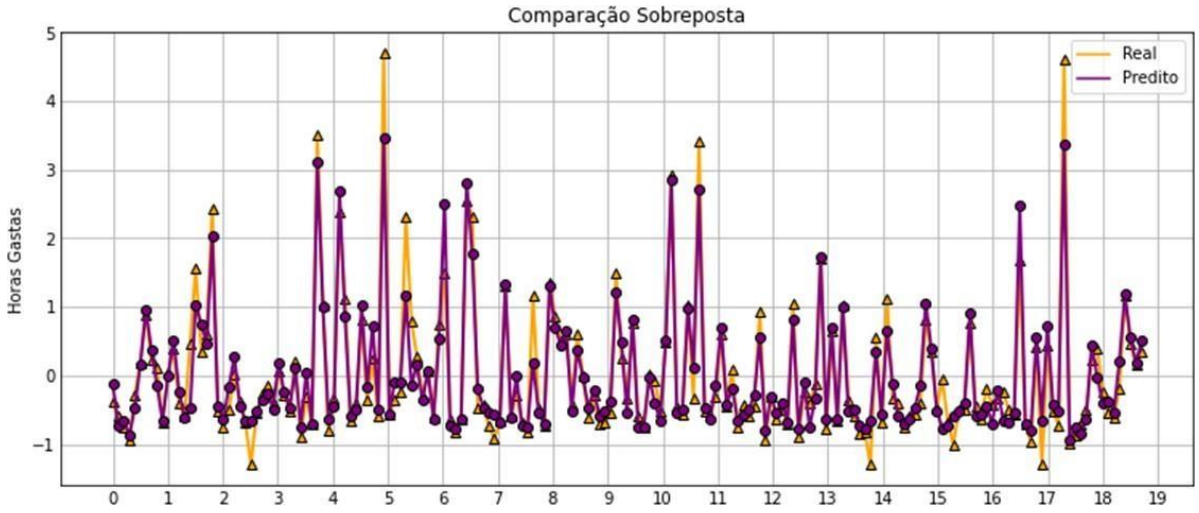
Fonte: Elaborado pelo autor, 2023.

Figura 21 – Dados de saída do real e previsto separados - Random Forest.



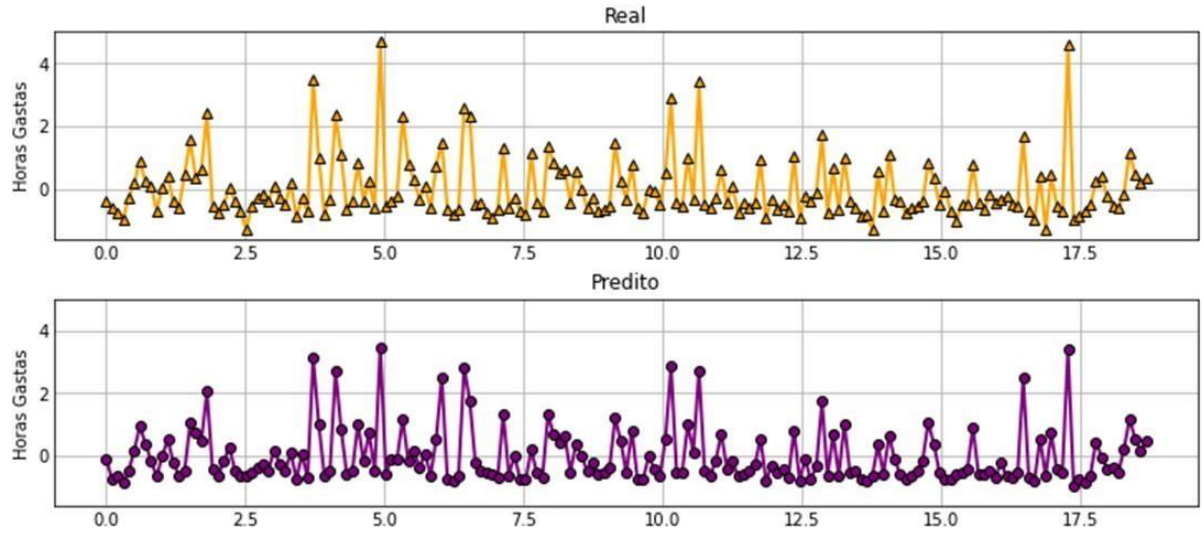
Fonte: Elaborado pelo autor, 2023.

Figura 22 – Dados de saída do real e previsto - SVM.



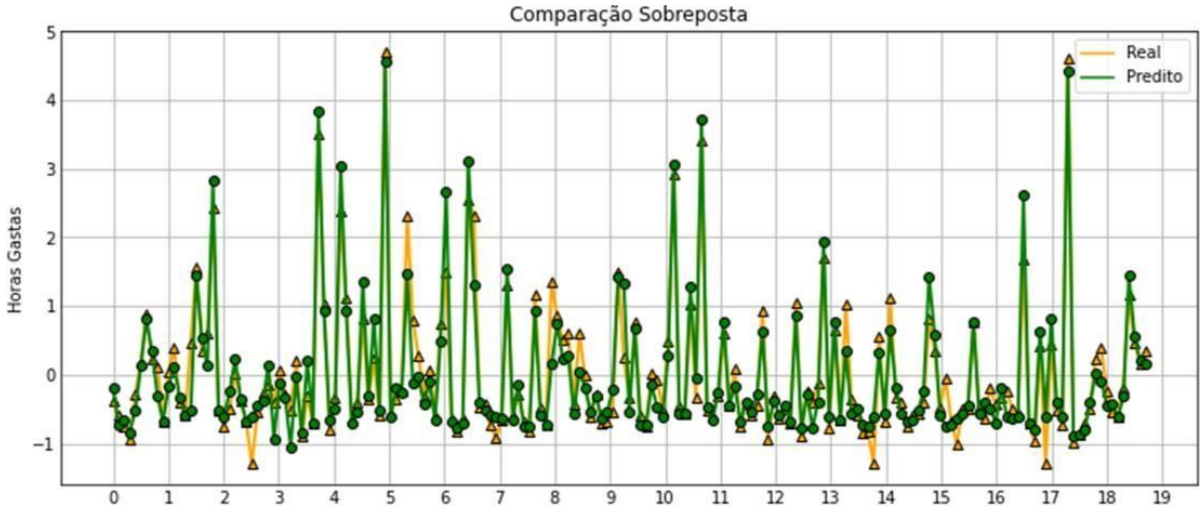
Fonte: Elaborado pelo autor, 2023.

Figura 23 – Dados de saída do real e previsto separados - SVM.



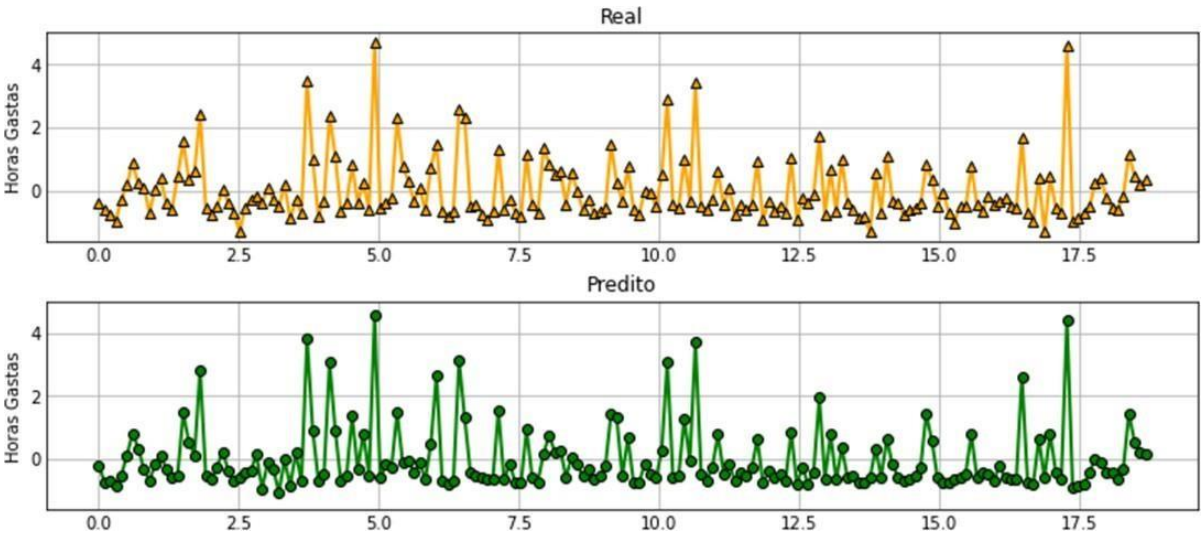
Fonte: Elaborado pelo autor, 2023.

Figura 24 – Dados de saída do real e previsto - RNA.



Fonte: Elaborado pelo autor, 2023.

Figura 25 – Dados de saída do real e previsto separados - RNA.



Fonte: Elaborado pelo autor, 2023.

A partir de uma análise descritiva dos gráficos gerados, notou-se que, de fato, visualizando pico a pico, é possível perceber que as predições realizadas utilizando o método Random Forest foram as que mais se aproximaram dos dados de saída reais da amostra de teste.

Com os resultados adquiridos, o método de Random Forest foi escolhido como o método de regressão que daria prosseguimento ao trabalho de predição. Portanto, com o método de regressão escolhido, Random Forest, foi possível aplica-lo aos algoritmos citados no Capitulo 3 (Func_Predict, Func_State e IA_predict), criadas para prever futuros dados de entrada e integrar todos os resultados com a IDE criada em Excel.

4.5 Interface para interação do usuário

Com os algoritmos prontos para integração, foi necessário criar alguns códigos em VBA para que a integração se concretizasse e o funcionamento da IDE ocorresse de maneira dinâmica. A IDE tem como interface as páginas mostradas na figura 19.

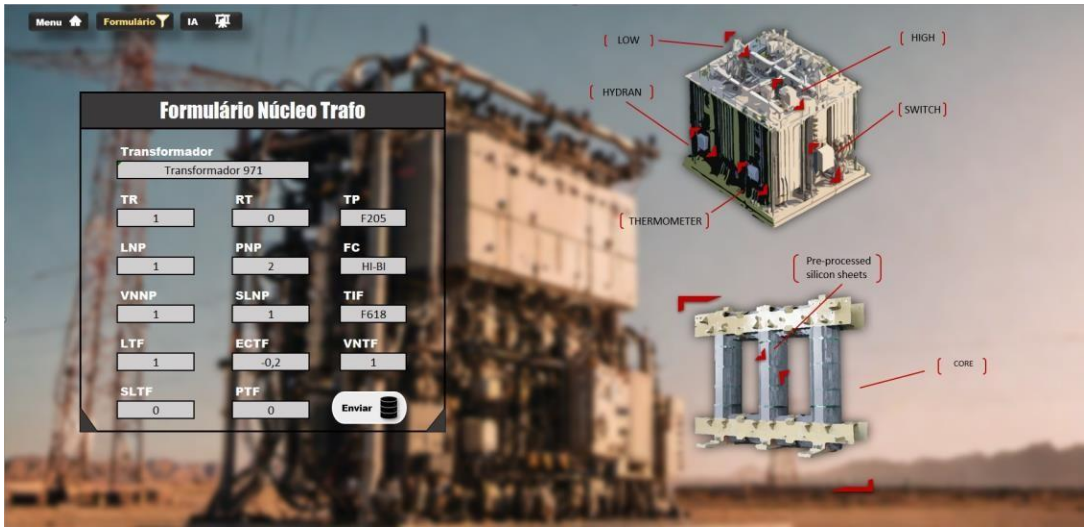
O Menu, possui apenas algumas macros feitas para haver acessibilidade entre as páginas por meio de botões. Como pode ser visualizado na figura 26.

Figura 26 – Menu temático interativo utilizado como página principal do aplicativo.

Fonte: Elaborado pelo autor, 2023.

O Formulário, por sua vez, possui o código em VBA que realiza o processo de envio dos novos dados para a função Func_Predict, que tem como objetivo aplicar o método Random Forest. Na figura 27, é possível visualizar os campos de preenchimento e o botão “Enviar” que ativa o código em VBA.

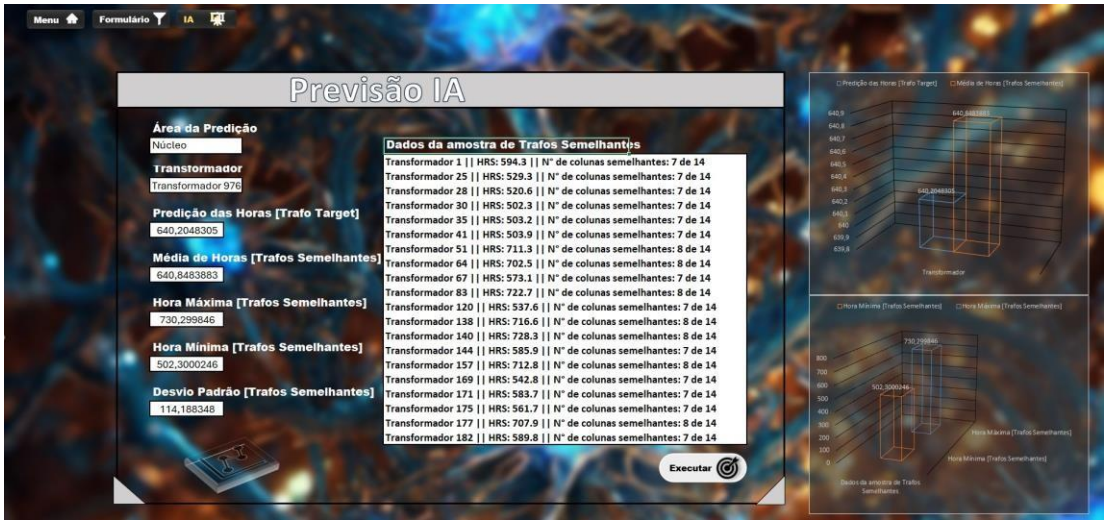
Figura 27 – Formulário de preenchimento das informações referetes ao núcleo do transformador com imagem ilustrativa de um transformador gerada por IA.



Fonte: Elaborado pelo autor, 2023.

A última página é intitulada como IA, nessa aba é possível visualizar o resultado da predição da hora gasta, bem como os resultados estatísticos como: média, desvio padrão, valor máximo e mínimo, dos transformadores mais semelhantes em relação as informações dos dados de entrada. Além disso, é possível visualizar os dados dos transformadores mais semelhantes, na lista da caixa de texto. O botão “Executar”, serve para acionar o código em VBA que executa o script, em Python, IA_predict que serve para acionar as funções Func_Predict e Func_State, como também retornar os valores de saída das funções dentro de um arquivo Excel que manda os dados para a página IA, mostrada na figura 28.

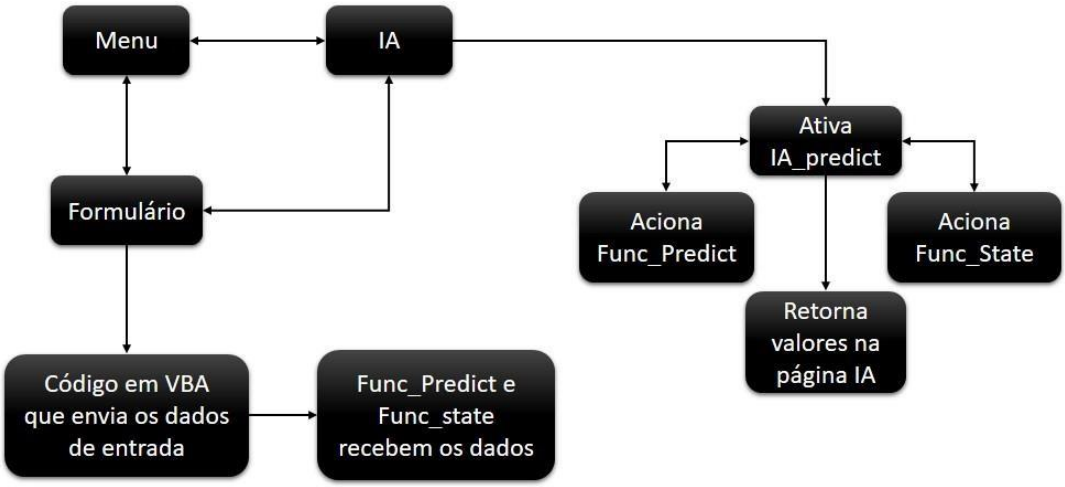
Figura 28 – Página de exibição das informações retornadas pela IA, numericamente e graficamente.



Fonte: Elaborado pelo autor, 2023.

Uma outra explicação didática da cadeia de funcionamento das interfaces da IDE, pode ser visualizada na figura 29.

Figura 29 – Fluxo de atividade da IDE.



Fonte: Elaborado pelo autor, 2023.

5 CONCLUSÃO E TRABALHOS FUTUROS

Ao longo do trabalho realizado, foi possível evidenciar várias maneiras de tratar e analisar os dados em função de prepará-los para o treinamento e teste dos modelos de predição. Deste modo, afirma-se que nem todos os dados de fato têm uma influência significativa diante da variável que se deseja prever, e uma das formas de auxiliar na definição dos melhores dados de entrada é utilizando ferramentas de análise de correlação, fornecidas pelas bibliotecas em Python. Outro ponto a se considerar é a importância do pré-tratamento dos dados. Para que as bibliotecas em Python funcionem de maneira efetiva, é necessário que os dados estejam configurados e tratados conforme as restrições pré-definidas pelas mesmas.

O método de regressão que se sobressaiu dos demais foi o *Random Forest*, conforme as métricas de desempenho $R2$, $R2$ ajustado e MSE. Embora, na métrica MAE, tenha ficado apenas atrás do método SVM. Portanto, por mais que desta vez o *Random Forest* tenha sido o método com a melhor performance, não significa que ele sempre será a melhor opção. Uma vez que, para cada massa de dados distintos, um método de regressão diferente pode se sair melhor. A base de dados deste projeto tinha muitos dados de entrada; quando isso acontece, o *Random Forest* tende a ser mais eficiente. Além disso, os demais métodos avaliados, SVM e RNA, tiveram um bom desempenho diante das métricas de avaliação. Dentre os dois, o SVM obteve melhores resultados que o método RNA.

Com o trabalho proposto, foi desenvolvida uma IA que retorna dados preditos do número de horas gastas para a fabricação do núcleo de transformadores. A partir deste projeto, houve ganhos na performance da elaboração de propostas por parte do setor de PPCP, que antes realizava esse processo de forma manual, utilizando como método a análise de curvas de tempo e comparação em planilhas.

5.1 Trabalhos Futuros

No projeto atual, foi proposto o desenvolvimento de uma IDE, que foi totalmente desenvolvida utilizando o Excel em conjunto com o VBA. Portanto, como sugestão de melhorias para os projetos futuros, seria a utilização de linguagens de programação no desenvolvimento da IDE, que possibilitassem o acesso via aplicativos Android, iOS e página web. Como, por exemplo, linguagens como C#, ASP.NET e XAML.

Além disso, os resultados decorrentes das previsões efetuadas pelos três modelos de regressão poderiam ser submetidos a um teste de hipótese. Este processo visa avaliar se as diferenças observadas nos resultados são estatisticamente significativas, proporcionando uma análise rigorosa da validade das conclusões obtidas. O teste de hipótese consiste em formular uma hipótese nula e uma hipótese alternativa, aplicando técnicas estatísticas para determinar se há evidências suficientes para rejeitar a hipótese nula em favor da hipótese alternativa. Este processo adiciona robustez à interpretação dos resultados, fortalecendo a confiança nas conclusões obtidas.

REFERÊNCIAS

A. A. Adebiyi, A. O. Adewumi, C. K. Ayo, et al. Comparison of arima and artificial neural networks models for stock price prediction. *Journal of Applied Mathematics*, 2014, 2014. Citado na página 23.

M. Alexandre, D. Carrijo, and A. J. AL. Arquiteturas de monitoramento e diagnóstico de transformadores de potência. *XVIII Seminário Nacional de Distribuição de Energia Elétrica*, 18, 2017. Citado na página 26.

R. Borsato and L. L. Corso. Aplicação de inteligência artificial e arima na previsão de demanda no setor metal mecânico. *Scientia cum Industria*, 7(2):165–176, 2019. Citado 2 vezes nas páginas 22 e 23.

A. V. Braga, A. F. Lins, L. S. Soares, L. G. Fleury, J. C. Carvalho, and R. S. do Prado. Machine learning: O uso da inteligência artificial na medicina. *Brazilian Journal of Development*, 5(9): 16407–16413, 2019. Citado na página 24.

W. V. Calil. *Determinação do fator de correção para cálculo de perdas magnéticas em núcleos de transformadores de potência pelo método de elementos finitos.* PhD thesis, Universidade de São Paulo, 2009. Citado na página 26.

D. D. Chamberlin. Early history of sql. *IEEE Annals of the History of Computing*, 34(4):78–82, 2012. Citado na página 15.

C. M. H. S. da Rocha et al. Ciência de dados e aprendizado de máquina para predição em séries temporais financeiras. 2019. Citado na página 21.

D. de Castro, LN e Ferrari. Introdução à mineração de dados: Conceitos básicos. *Algoritmos e Aplicações, Saraiva*, 2016. Citado na página 21.

A. Domingos, A. Joaquim, D. Dauta, and O. A. Formiga. Microfilmagem e digitalização de documentos. 2021. Citado na página 14.

B. Madariaga de la Campa et al. Sanz de sautuola y el descubrimiento de altamira: consideraciones sobre las pinturas. 2000. Citado na página 14.

L. Marujo et al. Estudo comparativo entre métodos estatísticos e de inteligência artificial para previsão de preço de café no brasil. Master's thesis, Universidade Tecnológica Federal do Paraná, 2021. Citado na página 23.

L. Z. Maselli and R. G. Negri. Integração entre estratégias multiclasses e diferentes funções kernel em máquinas de vetores suporte para classificação de imagens de sensoriamento remoto. *Rev. Bras. Cartogr*, 71(1):149–175, 2019. Citado na página 35.

M. Mustafa, R. Rezaur, S. Saiedi, and M. Isa. River suspended sediment prediction using various multilayer perceptron neural network training algorithms—a case study in malaysia. *Water resources management*, 26:1879–1897, 2012. Citado na página 24.

O. d. S. Nascimento. Previsão de preços de ações utilizando inteligência artificial. 2023. Citado na página 22.

R. Y. Olivo, P. L. de Paula Filho, and A. C. Junior. Uma abordagem neural na identificação de objetos em imagens para auxilio na manutenção de rede elétrica. In *Anais Estendidos do XXXIII Conference on Graphics, Patterns and Images*, pages 179–182. SBC, 2020. Citado na página 25.

F. Rees. *Johannes Gutenberg: Inventor of the printing press*. Capstone, 2006. Citado na página 14.

S. J. Rigatti. Random forest. *Journal of Insurance Medicine*, 47(1):31–39, 2017. Citado na página 23.

G. C. Santos et al. Algoritmos de machine learning para previsão de ações da b3. 2020. Citado na página 25.

J. A. V. Santos. *Inteligência artificial aplicada à avaliação de crédito bancário*. PhD thesis, 2021. Citado na página 23.

R. Shaikh. Choosing the right encoding method-label vs onehot encoder. *Towards Data Science*, 9:48, 2018. Citado na página 21.

I. S. Silva. Inteligência artificial para avaliação da qualidade da água. 2019. Citado na página 24.

W. B. C. d. Souza. Mineração de dados aplicada a previsão de preços de ações utilizando weka. 2021. Citado na página 21.

D. Sun, J. Xu, H. Wen, and D. Wang. Assessment of landslide susceptibility mapping based on bayesian hyperparameter optimization: A comparison between logistic regression and random forest. *Engineering Geology*, 281:105972, 2021. Citado na página 23.

M. M. V. Takáo. *Inteligência artificial em alergologia e imunologia: desenvolvimento de modelos de predição de risco para erros inatos da imunidade*. PhD thesis, [sn], 2023. Citado na página 25.

H. Tatsat, S. Puri, and B. Lookabaugh. *Machine Learning and Data Science Blueprints for Finance*. O'Reilly Media, 2020. Citado na página 22.

G. Van Rossum et al. Python programming language. In *USENIX annual technical conference*, volume 41, pages 1–36. Santa Clara, CA, 2007. Citado na página 15.

L. Yang, H. Wu, X. Jin, P. Zheng, S. Hu, X. Xu, W. Yu, and J. Yan. Study of cardiovascular disease prediction model based on random forest in eastern china. *Scientific reports*, 10(1): 5245, 2020. Citado na página 25.

C. Zaniol, C. Pazinatto, A. P. S. Schiller, and J. C. P. de Moraes. Previsão de inflação com o uso de inteligência artificial. *Revista Brasileira de Computação Aplicada*, 13(2):96–104, 2021. Citado na página 22.

W. Zhao, S.-B. Duan, A. Li, and G. Yin. A practical method for reducing terrain effect on land surface temperature using random forest regression. *Remote sensing of environment*, 221: 635–649, 2019. Citado na página 23.