UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Faculdade de Engenharia e Ciências de Guaratinguetá

VINÍCIUS DA LESSANDRO FIORETO

Aplicação de modelos de aprendizado de máquina para a previsão da temperatura de um motor elétrico

Guaratinguetá - SP
2023

Vinícius Da Lessandro Fioreto

Aplicação de modelos de aprendizado de máquina para a previsão da temperatura de um motor elétrico

Trabalho de Graduação apresentado ao Conselho de Curso de Graduação em Engenharia Mecânica da Faculdade de Engenharia e Ciências do Campus de Guaratinguetá, Universidade Estadual Paulista, como parte dos requisitos para obtenção do diploma de Graduação em Engenharia Mecânica.

Orientador: Prof. Dr. José Roberto Dale Luche

Guaratinguetá - SP
2023

F518a

Fioreto, Vinícius Da Lessandro
Aplicação de modelos de aprendizado de máquina para a previsão da temperatura de um motor elétrico / Vinícius Da Lessandro Fioreto - Guaratinguetá, 2023.
55 f : il.
Bibliografia: f. 52-55

Trabalho de Graduação em Engenharia Mecânica – Universidade Estadual Paulista, Faculdade de Engenharia e Ciências de Guaratinguetá, 2023.
Orientador: Prof. Dr. José Roberto Dale Luche

1. Aprendizado de máquinas. 2. Inteligência artificial.
3. Motores elétricos. I. Título.

CDU 621.313.13

Luciana Máximo
Bibliotecária/CRB-8 3595

UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Faculdade de Engenharia e Ciências de Guaratinguetá

VINÍCIUS DA LESSANDRO FIORETO

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO DIPLOMA DE
"**GRADUADO EM ENGENHARIA MECÂNICA**"

APROVADO EM SUA FORMA FINAL PELO CONSELHO DE CURSO DE GRADUAÇÃO EM ENGENHARIA MECÂNICA

Prof. Dr. CELSO TUNA
Coordenador

BANCA EXAMINADORA:

Prof. Dr. JOSÉ ROBERTO DALE LUCHE
Orientador/UNESP-FEG

Prof. Dr. ANEIRSON FRANCISCO DA SILVA
UNESP-FEG

Profa. Dra. CLAUDIA REGINA DE FREITAS
Membro Externo

Jnaeiro de 2023

DADOS CURRICULARES

VINÍCIUS DA LESSANDRO FIORETO

NASCIMENTO	29.08.1997 – Avaré / SP
FILIAÇÃO	José Maria Fioreto Mara Regina Da Lessandro
2016/2022	Graduação em Engenharia Mecânica – Bacharelado Universidade Estadual Paulista "Júlio de Mesquita Filho"

AGRADECIMENTOS

à minha mãe, *Mara Regina Da Lessandro*, que sempre me apoiou, incentivou e me educou para eu ser quem sou hoje.

à minha irmã, *Mayara Da Lessandro Fioreto,* que sempre esteve me incentivando na questão da minha formação como engenheiro.

à minha noiva, *Nathalia Rodrigues Ribeiro*, que esteve ao meu lado em todo o processo de realização do trabalho e me ajudou a manter motivado e focado.

aos meus amigos, que mesmo nos momentos difíceis estiveram comigo e que tornam a vida mais fácil de ser vivida.

ao meu orientador, *Prof. Dr. José Roberto Dale Luche,* pelo apoio dado e por cada aula lecionada durante a graduação.

à empresa IBM, onde estagio, que forneceu todo material necessário para produção deste trabalho.

RESUMO

Os dados vêm moldando a nova fase industrial e empresarial que vivemos, sendo considerado o novo petróleo da era digital pela grande possibilidade de valorização de recursos que ele traz. Com o grande avanço tecnológico e capacidade de armazenamento, surgiram ferramentas sofisticadas para analisar o grande volume de dados, juntamente com algoritmos capazes de identificar padrões e realizar previsões baseados em métodos estatísticos conforme aprendem analisando os dados fornecidos a ele. Esse novo conceito de análise de dados baseia-se no Aprendizado de Máquina, que vem sendo amplamente usado para aumentar os investimentos e nossa qualidade de vida. Neste trabalho investiga-se formas de aplicações de análise de dados e métodos baseados no Aprendizado de Máquina utilizando uma base de dados aplicada a temperaturas de um motor síncrono de ímã permanente. O método utilizado constituiu em passar por todas as etapas da aplicação de aprendizado de máquina, desde a seleção, tratamento de dados, visualizações e por fim, a criação de 5 modelos preditivos utilizando diferentes algoritmos e suas interpretações. O Python foi utilizado como linguagem de programação para implementar o exercício de aprendizado de máquinas. Por fim, foi possível comparar a performance dos diferentes algoritmos aplicados e concluir que o método da Regressão Polinomial de ordem 3 se mostrou o melhor para este estudo de caso, com uma variância explicada de 93%, indicando que este modelo é eficaz para a base de dados em específico.

PALAVRAS-CHAVE: Aprendizado de máquina; motor síncrono de ímã permanente; previsão da temperatura do motor.

ABSTRACT

Data has been shaping the new industrial and business phase that we are experiencing; it being considered the new oil of the digital age due to the great possibility of valuing resources that it brings. With the great technological advancement and storage capacity, sophisticated tools emerged to analyze big data, along with algorithms capable of identify patterns e realize predictions based on statistical methods as they learn analyzing data provided to it. This new concept of analyzing data is based on Machine Learning, that has been used to enhance investments and our life quality. This work will investigate the ways of applying data analysis and methods based on Machine Learning utilizing a database of motor temperatures. The method used consists of going through all stages of machine learning, from selection, data treating, visualizations and finally formation of five predictive models using different algorithms for prediction of engine failures and its results analysis. Python was used as a programming language for implementing Machine Learning. Finally, it was possible to compare the performance of those different algorithms and conclude that Polynomial Regression of third-degree method presented the best result for this specific case, providing a explained variance of 93%, proving to be an algorithm very effective for this database of engine failures.

KEYWORDS: Machine learning; permanent magnet synchronous motor; prediction of motor temperature.

LISTA DE ILUSTRAÇÕES

LISTA DE ABREVIATURAS E SIGLAS

CPU	Unidade central de processamento
KDD	Knowledge Discovery in Databases
MIT	Massachusetts Institute of Technology
MSE	Erro Quadrático Médio
RMSE	Raiz Quadrada do Erro Médio
PMSM	Motor síncrono de ímã permanente

SUMÁRIO

1 INTRODUÇÃO

Como o avanço da globalização, a disputa entre as empresas vem se tornando cada vez mais acirrada, o que tornou a otimização de processos, com o objetivo de maximizar o desempenho com a manutenção da qualidade, a meta para o crescimento de negócios. Uma das principais estratégias nessa busca vem sendo o investimento em novas tecnologias, incluindo o processo de digitalização global e o conceito de empresa *Data Driven*, que são empresas que tomam decisões baseadas na análise de dados (PROVOST; FAWCETT, 2013).

Diante desse cenário, umas das tecnologias que vêm mais se destacando na busca por tal objetivo é do Aprendizado de Máquina, na medida em que captura padrões comportamentais de uma base de dados e os ensina para uma máquina (ZHOU, 2021). Dessa forma, é possível compreender diagnósticos situacionais e realizar predições de eventos futuros, o que auxilia na tomada de decisões.

O Aprendizado de Máquina está gradativamente aumentando sua presença no âmbito empresarial e vem se tornando a ferramenta mais potente e efetiva na análise de dados, que tem como intuito transformar números e relatórios em *insights* para a escolha de estratégias (MAHESH, 2020). Esse método vem influenciando e aprimorando conceitos na gestão de produção, na pesquisa operacional e na engenharia de qualidade dos produtos (POMERLEAU, 1988).

Segundo um levantamento da World Economic Forum (2023), 85 milhões de empregos serão eliminados pela inserção de máquinas e o uso de Inteligência Artificial, porém, concomitantemente, outros 95 milhões surgiram, todos em áreas de tecnologia, com ênfase na ciência e análise de dados. O estudo também revela que as empresas mais bem sucedidas no futuro serão as que investirem em talentos que trabalhem com maior eficiência juntamente com a Inteligência Artificial e o Aprendizado de Máquina.

Além da enorme adesão das empresas, essa área possui um vasto campo de aplicação no mercado de trabalho, desde traduções do Google, sugestões de filmes e produtos para consumo em sites, programação de carros autônomos e até detectar usos fraudulentos de cartões de crédito (MOHAMMED; KHAN; BASHIER, 2016). Onde há dados, o Aprendizado de Máquina pode ser aplicado e gera resultados positivos, tanto que, segundo um relatório da Forrester, empresas orientadas por dados podem crescer mais de 30% ao ano.

Um dos exemplos atuais do uso do Aprendizado de Máquina pode ser visto em estudos de predição de poluentes no ar atmosférico durante a pandemia de COVID-19 e quais as correlações entre ambos. Com isso, foi possível notar a influência que concentrações de poluentes causavam em novos casos da doença e possibilitar melhores formas de agir na questão da saúde pública (CHEN; XU; WANG; ZHANG, 2022).

Segundo a Universidade de Engenharia e Ciências Aplicadas de Virginia (2023), a aplicação do Aprendizado de Máquina pode ajudar a alcançar uma taxa de previsão próxima de 100% em defeitos de manufatura aditiva em tempo real. Essa eliminação de falhas, muito próxima da perfeição, pode transformar o futuro das empresas e indústrias em vários âmbitos, tanto no tempo de produção e lançamento de um produto, quanto na diminuição de desperdício de material e gastos com energia, levando ao aumento de lucros e crescimento.

Pensando na área industrial de engenharia, a base de dados abordada nesse trabalho, consiste em vários dados coletados por sensores de um motor síncrono de ímã permanente (PMSM), que é um motor elétrico no qual a velocidade de rotação é proporcional à frequência da sua alimentação (REZENDE, 1963). Os dados foram coletados pelo departamento de engenharia elétrica da Universidade de Paderborn, Alemanha.

A companhia referência em motores elétricos WEG utiliza o Aprendizado de Máquina para diagnosticar, monitorar e indicar manutenções preditivas em seus motores elétricos, com o uso de sensores que aprendem sobre os padrões e desvios de funcionamento que o motor elétrico apresenta durante sua operação. Com isso, a empresa consegue reduzir as rotas de verificação para a identificação das falhas, diminuindo custos e prazos de entrega (WEG, 2023).

Os principais dados, de uma visão comercial, são o torque e a temperatura do rotor, pois determinando bons estimadores para ambos os elementos é possível ter um maior controle sobre motor, reduzindo as perdas na potência e de uso de material, respectivamente. Portanto, foi escolhido a temperatura do rotor para ser alvo de predição do modelo de Aprendizado de Máquina.

1.1 OBJETIVOS

Este trabalho tem como objetivo realizar a predição do torque e da temperatura do rotor de um motor síncrono de imã permanente usando modelos de Aprendizado de Máquina.

1.1.1 Objetivos específicos

- Fazer o tratamento e a análise dos dados utilizando a linguagem Python;
- Criar e treinar os modelos de Aprendizado de Máquina;
- Indicar o melhor modelo.

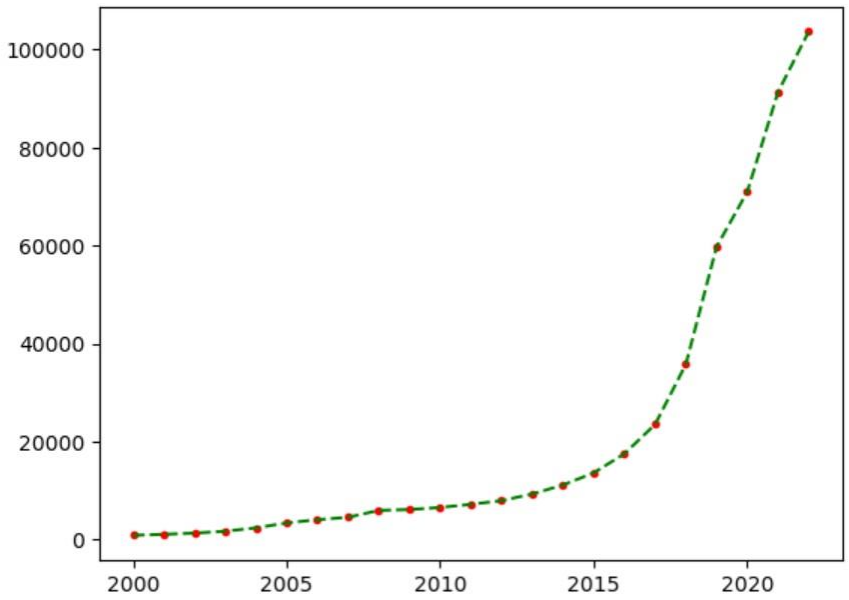
1.2 JUSTIFICATIVAS

O mundo tecnológico vem passando por uma revolução com uma velocidade nunca vista e o protagonista desse processo é o dado. Uma *CPU* moderna consegue executar bilhões de operações por segundo e enquanto na década de 1980 armazenar 1 MB custaria 200 dólares, hoje um 1 GB pode custar menos de 3 centavos de dólares (AMARAL, 2016). Segundo um estudo da International Data Group (IDC), a quantidade de dados que produzimos dobra a cada dois anos, sendo que as empresas geram um total de cerca de 2 zettabytes (ZB) por dia e esses dados valerão cerca de 77 bilhões de dólares até 2023. Além disso, segundo um relatório de 2019 da Vantage, 97.2% das empresas estavam investindo em grandes dados, assim como em inteligência artificial.

Como o Aprendizado de Máquina é uma subcategoria da inteligência artificial e se tornou a forma mais eficiente e ampla de analisar os dados, praticamente todas as empresas atualmente possuem profissionais dessa área. O seu uso traz vantagem competitiva, potencialização dos resultados, processos mais ágeis e redução dos gastos empresariais (DOMINGOS, 2017). Em vista disso, o presente trabalho, assim como tantos outros trabalhos acadêmicos, foca na aplicação desse mecanismo na área da engenharia.

Seguindo essa linha de raciocínio, realizou-se um estudo da quantidade de artigos publicados nas bases de dados Scopus a partir dos anos 2000 que apresentaram o termo Aprendizado de Máquina. Esta data foi escolhida pois o início da década que esta área começou a apresentar um relativo crescimento, que perdura até os dias atuais. O resultado da pesquisa, com todos as publicações que apareceram por ano, pode ser visto na Figura 1.

Figura 1 – Evolução das publicações contendo o termo Aprendizado de Máquina.



Fonte: Scopus (2023).

Como pode se notar na Figura 1, a quantidade de artigos publicados que possuem a palavra "Aprendizado de Máquina" vem crescendo exponencialmente, passando das 100 mil aparições no último registro de 2022. Esse crescimento pode ser associado a maior procura por esse tipo de ferramenta na análise de dados e ao valor que as empresas estão dando aos dados.

1.3 ESTRUTURA DO TRABALHO

O presente trabalho contém 5 capítulos. No Capítulo 2 há o referencial teórico sobre os principais conceitos e considerações abordados dentro da análise de dados, do Aprendizado de

Máquina e da base de dados escolhida. O Capítulo 3 traz os métodos de pesquisa utilizados e qual metodologia e ferramentas aplicadas para o estudo de caso. Já o Capítulo 4 traz toda a aplicação prática realizada, desde tratamentos de dados, gráficos e modelos de Aprendizado de Máquina. Por fim, o Capítulo 5 encerra o trabalho com as conclusões e considerações finais.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo será apresentado o que foi visto da literatura consultada para os principais tópicos que o presente trabalho aborda, englobando o próprio Aprendizado de Máquina, que foi subdividido nos tipos de aprendizado e nos hiperparâmetros, o processo de tratamento de dados que foram utilizados, uma introdução dos modelos de Aprendizado de Máquina utilizados, o conteúdo da base de dados e, por fim, as métricas utilizadas para avaliar os modelos.

2.1 APRENDIZADO DE MÁQUINA

Segundo Mitchell (1997), um programa de computador aprende a partir da experiência E com respeito à algumas classes de tarefas T e medidores de performance P, se a sua performance nas tarefas T, medida por P, aprimore com a experiência E. A esse processo se dá o nome de Aprendizado de Máquina. Ou seja, esse método é aplicado para que o programa busque padrões nos dados e faça previsões com base em todos os dados já recebidos anteriormente.

O primeiro estudo nessa área ocorreu em 1950, quando Alan Turing, pioneiro na ciência computacional, lançou um artigo que contia o teste Turin, no qual dizia que a Inteligência Artificial seria atingida quando uma máquina conseguisse convencer um humano de que também era um humano. Porém, O termo apenas foi usado pela primeira vez em 1959 por Arthur Samuel, engenheiro do *MIT*, que se envolveu no projeto de se criar uma máquina autônoma que pudesse aprender sem ter sido programada para tal (Fradkov, 2020).

Contudo, o Aprendizado de Máquina foi apresentar uma crescente expansão na prática na década de 90, quando novas técnicas de estatística foram incorporadas e mudou-se de uma visão mais orientada ao conhecimento, para uma visão mais orientada aos dados (Fradkov, 2020). Isso ocorreu com o acompanhamento nessa época da crescente exponencial, que perdura até os dias atuais, do volume e da velocidade da geração de dados.

O Aprendizado de Máquina é um subcampo de Inteligência Artificial, sendo um programa no qual o objetivo é acumular experiência através de soluções bem-sucedidas de problemas anteriores para, assim, tomar decisões (Almeida, Carvalho e Menino, 2017). Para isso, usa-se algoritmos matemáticos, baseados em conhecimentos em probabilidade e estatística, engenharia, ciência da computação e até mesmo neurobiologia, no intuito de

aprender e analisar dados. Esse processo envolve a busca por um vasto espaço de possíveis hipóteses para determinar a que melhor se encaixa nos dados observados e qualquer conhecimento previamente detido pelo modelo.

As suas aplicações são das mais variadas, como nas áreas da medicina, biologia, engenharia, informação, logística, entre outros (MOHAMMED; KHAN; BASHIER, 2016). Amplamente utilizado no mercado de trabalho, os exemplos mais comuns são recomendações de conteúdo em serviços de *streaming*, identificação de formas mais eficientes e econômicas de transporte, identificação de ocasiões para transações com base na atividade do mercado no passado, previsão do tempo e detecção e prevenção de câncer a partir de relatórios de evolução médica de um paciente (SHINDE; SHAH, 2018). Porém, cada base de dados é tratada de uma maneira diferente, dependendo do tipo de aprendizado. Pode ser dividido em três grandes grupos: o Aprendizado Supervisionado, o Aprendizado Não Supervisionado e o Aprendizado por Reforço, como mostra a Figura 1.

Figura 2 – Diagrama dos tipos de Aprendizado.

Fonte: Almeida, Carvalho e Menino (2017).

2.1.1 Tipos de Aprendizado

No Aprendizado Supervisionado, os algoritmos aprendem de dados de treinamento rotulados, ou seja, o modelo recebe um conjunto de entradas com suas respectivas saídas e busca encontrar uma função que estabeleça uma relação aproximada entre elas. De uma maneira mais formal, o modelo baseado no Aprendizado Supervisionado busca encontrar uma função h(xi), denominada hipótese, que se aproxime da função f(xi), onde f(xi) é a saída da i-ésima entrada de x (RUSSEL; NORVIG, 2013). Esse tipo de aprendizado está associado aos modelos de regressão e classificação.

Já no Aprendizado Não Supervisionado, os dados não estão rotulados, classificados ou categorizados, não havendo uma variável de saída inicial. São utilizados para tarefas de agrupamento de dados com afinidade, como ocorre em sistemas de recomendação, e para redução de dimensionalidade, ou seja, encontrar um número inferior de variáveis que melhor representam as características dos conjuntos de dados, como ocorre na detecção de bordas no processamento digital de imagens (RUSSEL; NORVIG, 2013).

Por fim, os algoritmos do Aprendizado por Reforço são usados onde grandes quantidades de dados rotulados com os pares de entrada e saída não são explícitos e, também são usados, em cenários onde o mecanismo de recompensa para sucesso e penalização para falha levam à resultados ideais (RUSSEL; NORVIG, 2013). Esse tipo de aprendizado é amplamente aplicado em veículos autônomos, em que a recompensa seria a aproximação do destino e a penalização poderia ser a colisão com algum obstáculo.

2.1.2 Hiperparâmetros

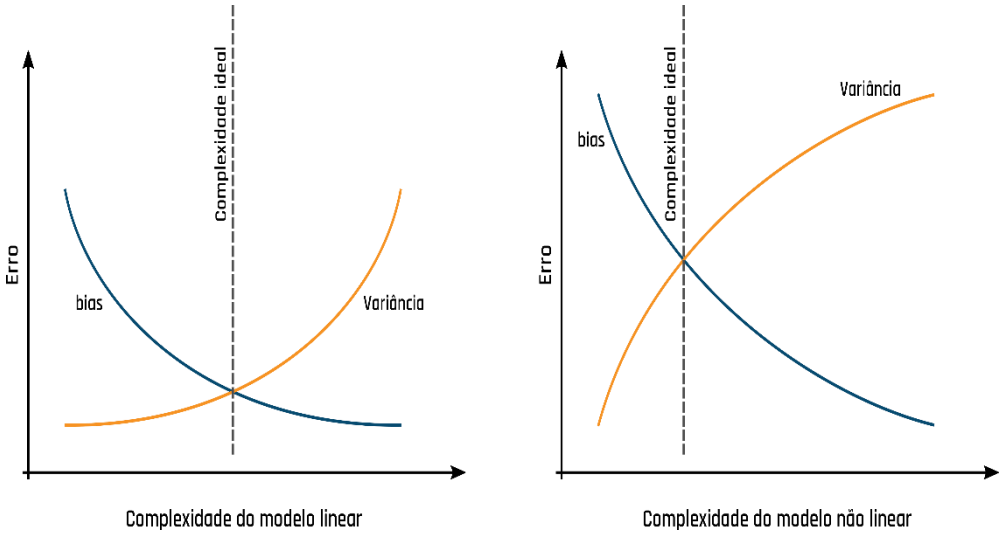
Um programa possui um conjunto de configurações, previamente definidas, que estão associadas à como o modelo aprende, sendo dependentes do tipo do modelo e ao tipo de aprendizado. Esse conjunto de configurações denomina-se hiperparâmetros, que são parâmetros ajustáveis que permitem controlar o processo de treinamento do modelo, sendo ajustados de acordo com os dados (CLAESEN; MOOR, 2019)

O processo de ajuste dos hiperparâmetros buscando melhorar o desempenho do modelo é conhecido como *fine-tuning* (ALMEIDA; CARVALHO; MENINO, 2017). Esse ajuste tem o intuito de calibrar o nível da assertividade e da precisão, sendo características associadas ao *bias* e a variância do modelo. O *bias* se refere à capacidade do modelo se ajustar aos dados aos

quais lhe foram fornecidos, enquanto o variância é a variabilidade das previsões feitas pelo modelo.

Conforme o modelo se ajusta aos dados por meio dos hiperparâmetros, a sua complexidade aumenta, porém isso também acarreta a perda de sua capacidade de generalização, aumentando a variância. Portanto, a configuração dos hiperparâmetros deve equilibrar o *bias* e a variância, no que se é chamado de *trade-off*. Esse equilíbrio depende da linearidade do modelo, sendo que para modelos lineares a complexidade do modelo deve ser ajustada de tal forma que o *bias* e a variância possuam o menor valor possível, enquanto que para modelos não lineares o equilíbrio deve ser onde a complexidade do modelo possui o menor *bias* a maior variância (CLAESEN; MOOR, 2019). Essas configurações são exemplificadas na Figura 3.

Figura 3 – Esquema do *trade-off*.

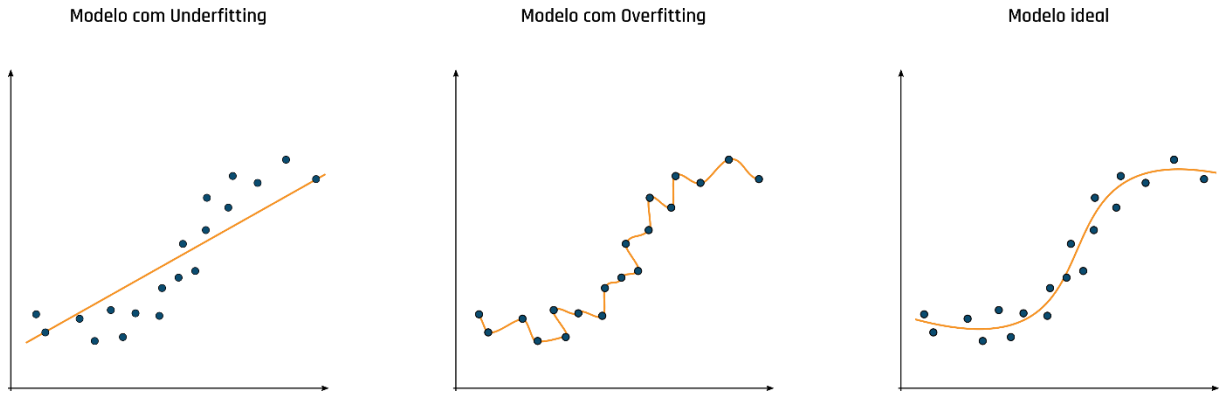


Fonte: Almeida, Carvalho e Menino (2017).

Quando esse ajuste está desbalanceado das complexidades do modelo, podem ocorrer os problemas de *overfitting* ou *underfitting*. No primeiro, o modelo está muito ajustado aos dados de treino, não sendo capaz de fazer a generalização para outros dados nunca vistos antes. Enquanto no segundo, o modelo não conseguiu aprender o suficiente com sobre os dados previamente fornecidos, levando à um erro elevado (ZHOU, 2021).

O *overfitting* pode ocorrer quando o algoritmo é muito complexo para os dados, quando há poucos dados para o treinamento, ou quando há muitos ruídos nos dados (valores extremos ou mesmo incorretos). Já o *underfitting* ocorre quando as características da base dado não são representativas (não tenham relação entre si ou não sejam importantes para o modelo), modelo inflexível com muitos parâmetros de restrição, ou até mesmo quando o algoritmo não é adequado (ZHOU, 2021). A Figura 4 mostra um exemplo desses casos.

Figura 4 – Diferentes ajustes do modelo aos dados.



Fonte: Almeida, Carvalho e Menino (2017).

2.2 TRATAMENTO DE DADOS

Quando se trabalha como uma grande quantidade de dados, se faz necessário o tratamento prévio com o intuito de averiguar a natureza dos dados, sua distribuição e se há algum tipo de anomalia. Porém, aquilo que o investigador procura como ferramenta de tratamento e a própria análise de dados depende muito da natureza e objetivos da investigação que se deseja realizar (RODRIGUES, 2011).

O processo de tratamento de dados consiste basicamente em preparar sua base de dados para a futura análise, podendo englobar a procura e eliminação de valores nulos e valores duplicados, limpeza de *outliers* (dados discrepantes, observações fora do comum), agrupamento de dados, análises das correlações com intuito de selecionar dados mais relevantes, entre outros.

Existem várias formas de fazer essa mineração dos dados, mas as mais comuns são por meio de funções de bibliotecas, como o Pandas e o Numpy, e por meio da visualização através de gráficos. Essas operações, se feitas de maneira correta, levam a uma melhor performance do modelo de Aprendizado de Máquina.

2.3 MODELOS DE APRENDIZADO DE MÁQUINA UTILIZADOS

Os modelos de Aprendizado de Máquina são classificados em supervisionados ou não supervisionados, sendo que os supervisionados ainda são divididos em modelos de regressão ou de classificação. Em modelos de regressão, o *output* é um valor numérico específico, enquanto em modelos de classificação busca-se encontrar uma classe como resultado, dentro das possibilidades limitadas existentes, como por exemplo, marcas de carro ou se um paciente possui uma doença ou não. Como a base de dados escolhida possui a variável *target* numérica e contínua, ou seja, não categórica, o presente trabalho focará nos modelos de regressão (MAHESH, 2020).

2.3.1 Regressão Linear

A análise de regressão define um conjunto vasto de técnicas estatísticas usadas para modelar relações entre variáveis e predizer o valor de uma ou mais variáveis dependentes a partir de um conjunto de variáveis independentes (CHEIN, 2019). A partir desse conceito, o modelo busca estimar a equação de uma reta, descrita por:

$$y = \beta_0 + \beta_1 x + \varepsilon \quad [1]$$

Quando se trata de uma regressão linear simples, em que β_0 é o coeficiente linear da reta, β_1 é o coeficiente angular e ε é o erro do modelo. Em outras palavras, pode-se concluir que o modelo busca estimar a inclinação dessa reta.

Em regressões lineares múltiplas, que são os casos mais comuns no âmbito da análise de dados, existem múltiplas variáveis independentes, diferentemente da regressão linear simples que possui apenas uma (CHEIN, 2019). Sendo n o número de variáveis dependentes, a equação desse tipo de regressão é descrita como:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon \quad [2]$$

Esses modelos levam em consideração alguns pressupostos, sendo eles a linearidade (a relação da variáveis deve ser linear), a homogeneidade de variância (termos do erro da variância constantes), a independência de erros (os erros nas variáveis independentes não devem ser correlacionados), não multicolinearidade (as variáveis independentes não podem ser próximas demais de uma correlação perfeita) e baixa exogeneidade (valores das variáveis independentes não estão contaminados com erros de medida) (CHEIN, 2019).

O quão bom essa reta representará a base de dados, terá forte relação com a correlação entre as variáveis independentes e dependentes, que avalia o grau de associação entre ambas, assim como o desvio padrão, que indica o grau de variação, ou dispersão, do conjunto de dados. Sendo assim, segundo Mitchell (1997), pode-se inferir que a reta de Regressão Linear depende das seguintes estatísticas básicas:

$$\text{Média de x: } \frac{1}{N}\sum_{i=1}^{N} x_i \quad [3]$$

$$\text{Média de y: } \frac{1}{N}\sum_{i=1}^{N} y_i \quad [4]$$

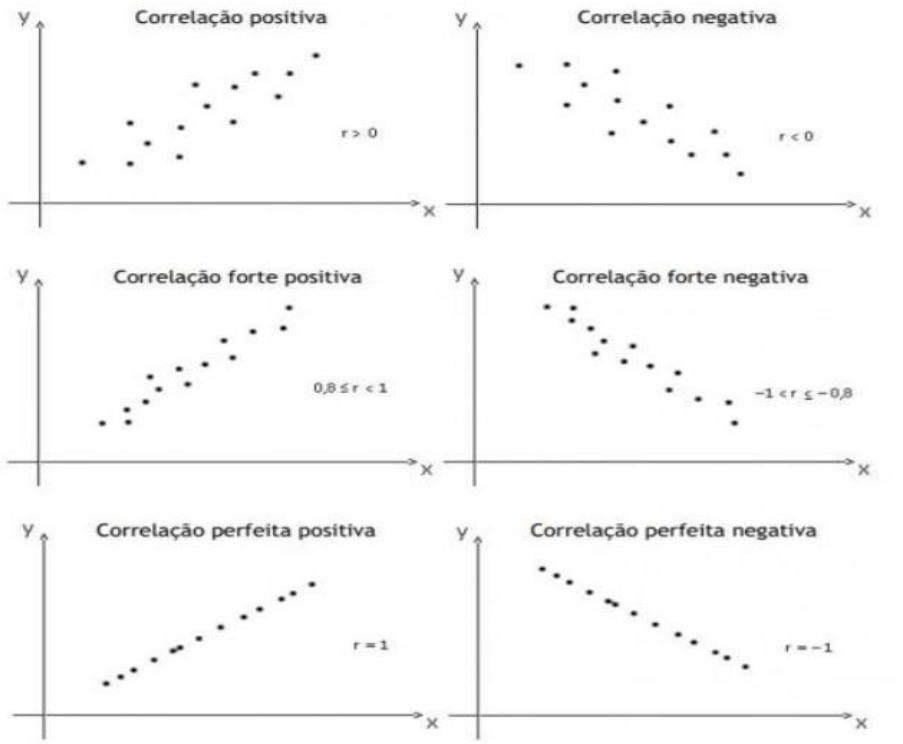
$$\text{Desvio padrão de x: } S_x = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2} \quad [5]$$

$$\text{Desvio padrão de y: } S_y = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2} \quad [6]$$

$$\text{Correlação de x e y: } r = \frac{1}{N}\sum_{i=1}^{N}\frac{(x_i - \bar{x})}{Sx} \cdot \frac{(y_i - \bar{y})}{Sy} \quad [7]$$

Em relação as equações [5] e [6], quanto mais próximo o desvio padrão de 0, mais homogêneo será o conjunto de dados, resultando em uma reta mais representativa. Enquanto na equação [7], correlações próximas de -1 ou 1 demonstram que os dados representam melhor a variável dependente (Mitchell, 1997). A Figura 5 representa como as retas de Regressão Linear variam de acordo com a correlação.

Figura 5 – Classificação da correlação através do diagrama de dispersão.



Fonte: Chein (2019).

2.3.2 Regressão Polinomial

Em muitos casos, a variável resposta (variável dependente) e a variável preditora (variável independente) não possuem uma relação linear, por isso, usa-se a Regressão Polinomial quando efeitos curvilíneos estão presentes na relação entre as duas variáveis (Miguel, 2020). A equação a ser estimada pelo modelo possui a seguinte forma:

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_n x^n + \varepsilon \quad [8]$$

- onde:

β_0 = constante linear

β_i = coeficientes de x^i

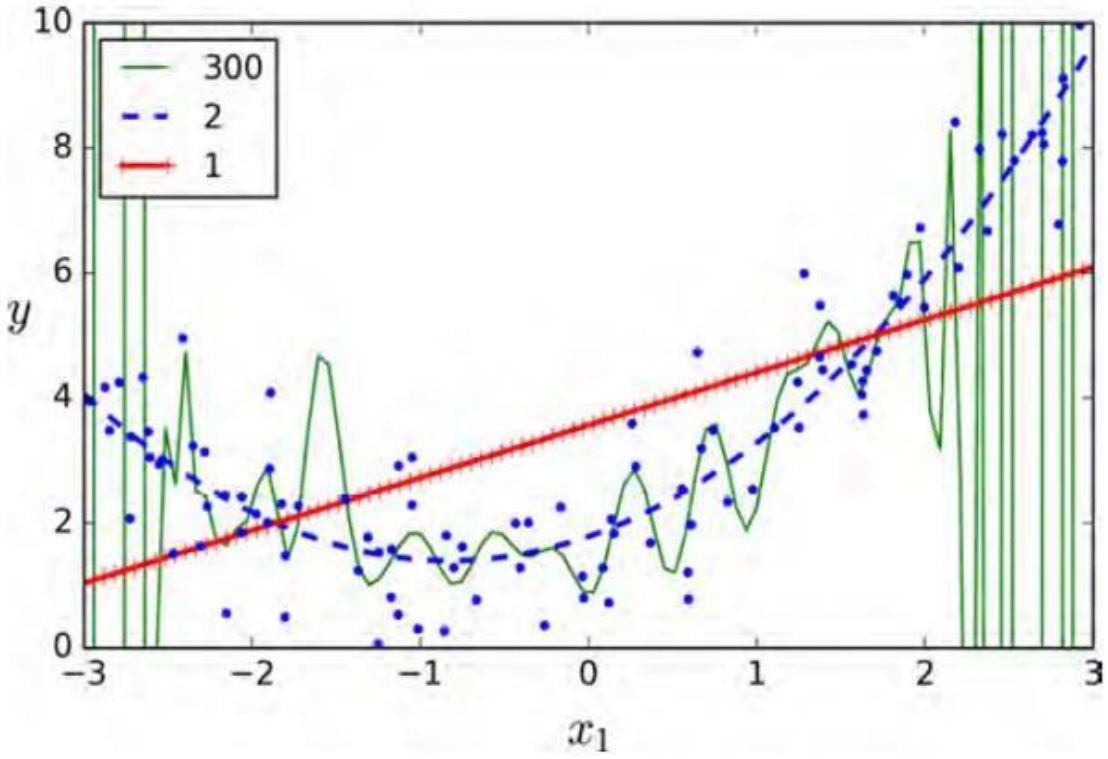
n = ordem do polinômio

ε = erro do modelo

Para o uso desses modelos, é necessário determinar a ordem desejada do polinômio por meio dos hiperparâmetros. A capacidade de ajustamento aos dados é mais forte quanto maior for a ordem, porém valores muito altos comprometem a capacidade de generalização, ou seja, *overfitting* (ZHOU, 2021). Portanto, busca-se manter a ordem a mais baixa possível.

Uma das abordagens possíveis para se escolher a ordem é ajustar sucessivamente os graus do polinômio de forma crescente, analisando as métricas de avaliação escolhidas para cada caso. A Figura 6 exemplifica como o modelo se ajusta aos dados dependendo da ordem do polinômio.

Figura 6 – Comportamento de diferentes ordens de um polinômio.



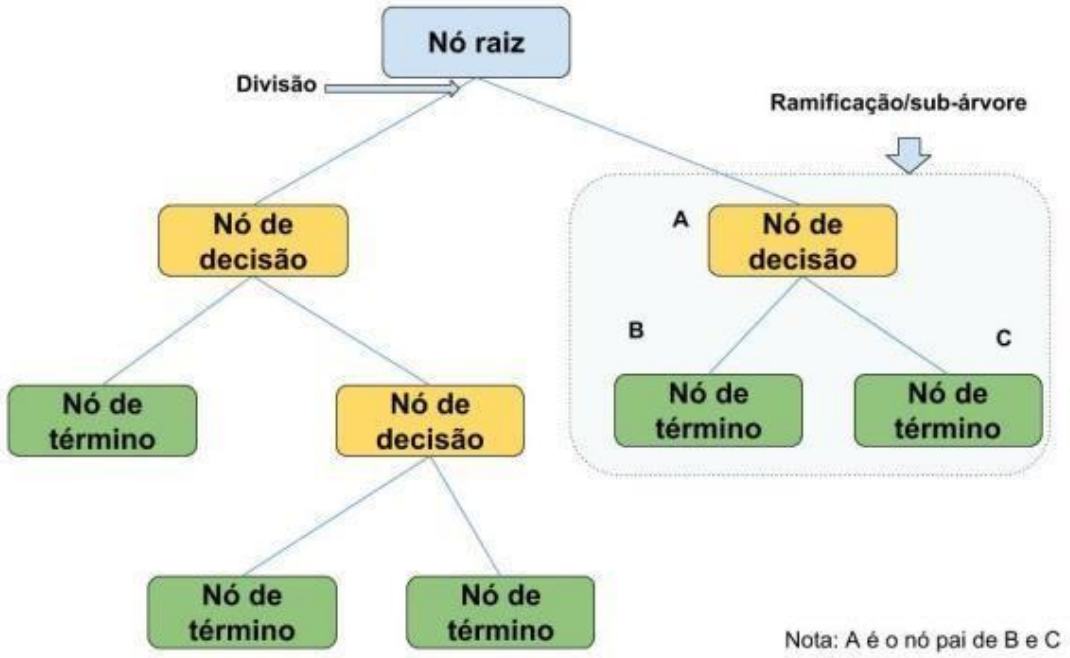
Fonte: Miguel (2020).

2.3.3 Árvore de Decisão

Os modelos de Árvore de Decisão são baseados no sistema de hierarquia, em que a previsão de um resultado é feita por meio de uma sequência de divisões recursivas em passos cada vez menores. Esta árvore é composta por nós de decisão e nós terminais, sendo que em

cada nó de decisão há uma função aplicada que pode gerar os resultados de cada ramo que saem dele (ZHOU, 2021). Dependendo da saída recebida pelo nó, toma-se o caminho correspondente, assim sucessivamente até atingir o nó terminal, onde é atingido um critério de parada. Esse esquema pode ser analisado na Figura 7.

Figura 7 – Esquema de uma Árvore de Decisão.



Fonte: Analytics Vidhya Content Team (2016)

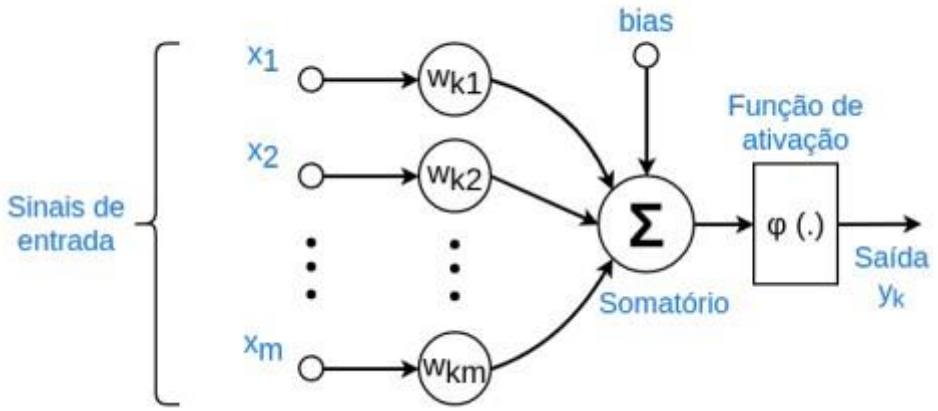
Podem ser usados tanto para classificação quanto para a regressão. Árvores de Regressão particionam um conjunto de dados em sub-árvores menores e então ajustam uma constante para cada observação na sub-árvore. Tal particionamento é gerado a partir de divisões binárias sucessivas com base nos diferente preditores e a constante a prever é baseada nos valores médios de resposta para todas as observações que se enquadram nessa sub-árvore. O tamanho da árvore no modelo é alterado por meio de hiperparâmetros, como o número mínimo de amostras para uma divisão de nós, profundidade máxima da árvore e número máximo de nós de término (ANALYTICS VIDHYA CONTENT TEAM, 2016)

2.3.4 Aprendizado Profundo

O Aprendizado Profundo é um ramo do Aprendizado de Máquina, que utiliza o conceito de redes neurais para prever resultados. Esse conceito surgiu com o propósito de simular estruturas neurais biológicas, em que neurônios, células computacionais simples, compõe uma estrutura massivamente interconectada e paralelamente distribuída capaz de armazenar e processar informações para exercer uma tarefa (GOODFELOW; BENGIO; COURVILLE, 2016).

A estrutura de um neurônio pode ser visualizada na Figura 8. Os componentes x_1, x_2, ..., x_m correspondem às entradas do neurônio que são combinadas em uma soma ponderada, na qual esses valores são escalados utilizando seus respectivos pesos, representados por w_{k1}, w_{k2}, ..., w_{km}. O limiar de ativação inerente (*bias*) é introduzido como um grau de liberdade adicional para manipular a saída do neurônio, permitindo um melhor ajuste. O resultado da soma entre o limiar de ativação e as entradas multiplicadas pelos pesos produz o potencial de ativação υk . A saída, normalmente indicada pelo vetor yk , corresponde ao potencial de ativação transformado pela função de ativação ϕ(.). A função ϕ(.) é normalmente uma função não-linear do tipo f(x) = max(x,0) ou uma função do tipo *softmax*, que mapeia a saída em um intervalo entre 0 e 1 (MAHESH, 2020).

Figura 8 – Estrutura básica de um neurônio



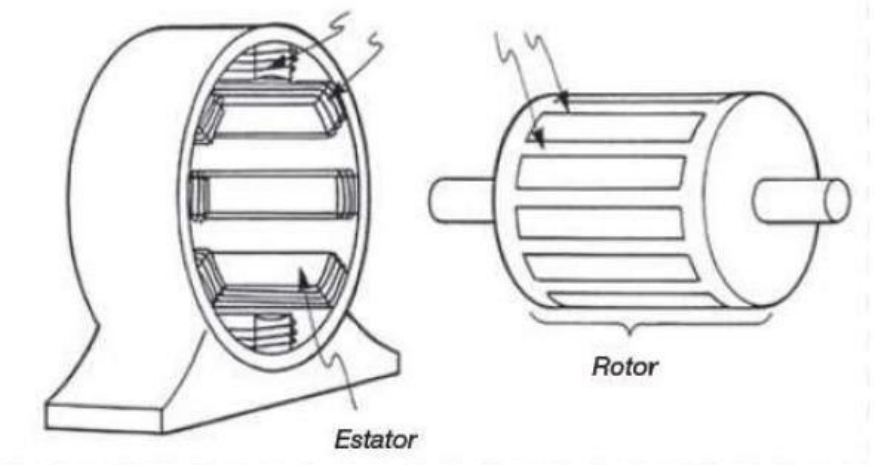
Fonte: Goodfelow; Bengio; Courville (2016)

2.4 CASO DE ESTUDO

O motor síncrono de ímã permanente (PMSM) é uma das melhores escolhas para uma gama completa de aplicações de controle de movimento. Por exemplo, o PMSM é amplamente utilizado em robótica, ferramentas de máquinas, atuadores, aplicações comerciais e residen ciais, e vem sendo considerado em aplicações de alta potência como acionamentos industriais e propulsão veicular. O PMSM é conhecido por ter uma baixa oscilação de torque, desempenho dinâmico superior, alta eficiência e alta densidade de potência (TAVARES, 2018).

Esse tipo de motor utiliza um ímã permanente para gerar o campo magnético ao invés de bobinas ou exigir um componente de magnetização da corrente no estator, o que leva à uma simplificação de sua construção, a redução de perdas e uma melhor eficiência (JONES; LANG, 1989). Os principais componentes de um PMSM podem vistos na Figura 9.

Figura 9 – Estrutura de um PMSM



Fonte: Tavares (2018)

- Estator: Parte estacionária da máquina e inclui os elementos:
  - **o** Lâminas do estator: O conjunto das lâminas forma a pilha do estator, que serve como parâmetro para o dimensionamento eletromagnético.
  - **o** Enrolamento da armadura: Responsável por gerar um campo magnético após ser energizado.

- Entreferro: Situa-se entre o estator e o rotor, sendo preenchido por ar e mantendo uma pequena distância entre os demais elementos para aumentar a eficiência dos ímãs.

- Rotor: Elemento girante da máquina, onde se situa o ímã permanente. Composto pelos elementos:
  - **o** Estrutura de Pólos: Ímãs e componentes magnéticos. **o** Coroa do rotor.
  - **o** Eixo e sistema de rolamento.

2.5 MÉTRICAS DE AVALIAÇÃO DO MODELO

Ao desenvolver modelos de Aprendizado de Máquina, deve-se utilizar métricas apropriadas para cada problema, sendo importante avaliar a performance do modelo o mais preciso possível (MITCHELL, 1997). A partir do valor das métricas é possível determinar a qualidade de um modelo, portanto se forem mal escolhidas, será impossível avaliar se o modelo atende aos requisitos necessários.

Existem várias formas de realizar essa avaliação, desde gráficos a fórmulas previamente definidas. A seguir serão detalhadas as métricas de avaliação utilizadas neste trabalho, tendo em mente que são modelos de regressão.

2.5.1 Erro Quadrático Médio

O Erro Quadrático Médio, MSE (da sigla em inglês *Mean Squared Error*), é usado quando se deseja verificar a acurácia (proximidade entre o valor obtido experimentalmente e o valor verdadeiro da medição) de um modelo, dando um maior peso aos maiores erros, pois cada erro é calculado ao quadrado individualmente e, após isso, a média desses erros quadráticos é calculada (PSAROS; MENG; ZOU; GUO; KARNIADAKIS, 2022). Cada erro é calculado pela diferença entre o valor real y_i e o valor previsto $\hat{y}_i$. A fórmula do MSE é definida da seguinte forma:

$$\text{MSE} = n^{-1} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 \quad [9]$$

2.5.2 Raiz Quadrada do Erro Médio

A Raiz Quadrada do Erro Médio, RMSE (da sigla em inglês *Root Mean Squared Error*), é simplesmente a raiz quadrada do MSE, sendo assim, o erro retorna à unidade de medida do modelo (PSAROS; MENG; ZOU; GUO; KARNIADAKIS, 2022). É mais sensível a erros maiores e retorna um valor mais palpável e instrutivo. A fórmula do RMSE é definida da seguinte forma:

$$\text{RMSE} = \sqrt{\sum_{i=1}^{n} \frac{(y_i - \hat{y}_i)^2}{n}} \quad [10]$$

2.5.3 Variância Explicada

Outra forma de avaliar a qualidade de um modelo é através da análise de variância, que pode ser definida como a dispersão dos erros em torno de valores médios. Nesse caso a Variância Explicada nos diz o quanto o modelo de Aprendizado de Máquina explica dos dados de teste utilizados, sendo que valores próximos de 1 indicam que o modelo explica boa parte dos dados (PSAROS; MENG; ZOU; GUO; KARNIADAKIS, 2022). Sendo Var a variância, a sua fórmula é definida como:

$$\text{Variância Explicada } (y_i,\ \hat{y}_i) = \frac{Var\,(y_i - \hat{y}_i)}{Var(y_i)} \quad [11]$$

3 MÉTODOS

Desejando realizar a classificação desta pesquisa, pode-se classificá-la como de natureza aplicada, pois estuda um fenômeno sem realizar a interferência no mesmo (PRODANOV; FREITAS, 2013). Quanto a seu objetivo, pode ser classificada como descritiva, ao apenas registrar e descrever os fatos observados (PRODANOV; FREITAS, 2013). Em relação aos procedimentos técnicos, foram utilizados a pesquisa bibliográfica, a fim de realizar uma contextualização inicial dos assuntos abordados, e o estudo de caso. Por fim, quanto a abordagem, classifica-se como quantitativa, com o uso de recursos e de técnicas estatísticas (PRODANOV; FREITAS, 2013). A Figura 10 traz a classificação da pesquisa.

Figura 10 – Classificação da Pesquisa

NATUREZA	ABORDAGEM	PROCEDIMENTOS	OBJETIVOS
Básica	Qualitativa	Bibliográfica	Exploratória
Aplicada	Quantitativa	Documental	Explicativa
	Combinada	Experimental	Descritiva
		Estudo de Caso	Normativa
		Levantamento	
		Expost-facto	
		Pesquisa-ação	
		Participante	

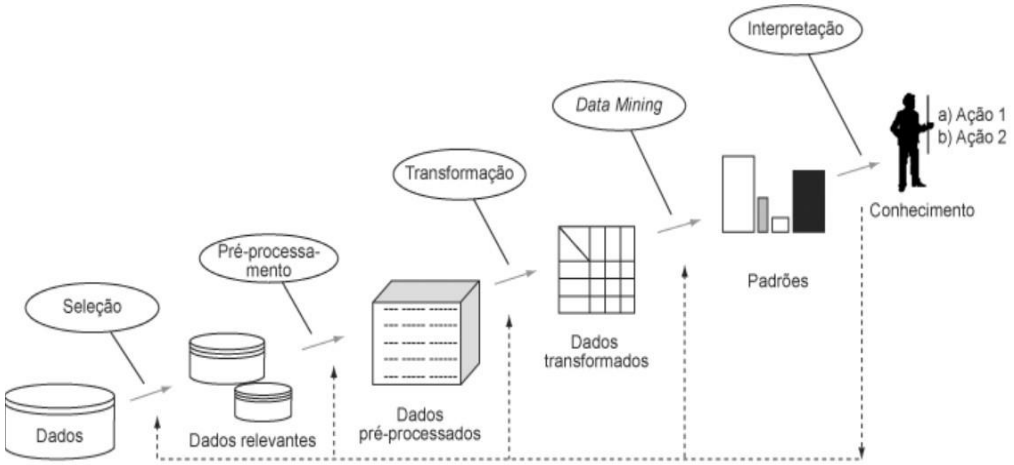
Fonte: Próprio autor.

O presente trabalho baseia-se no Aprendizado de Máquina com o intuito de prever a temperatura do rotor de um motor elétrico do tipo síncrono de ímã permanente, a partir de uma base de dados que apresenta uma série de variáveis que possibilitam que o modelo de Aprendizado de Máquina faça a predição. Para a realização do case foi utilizado a metodologia KDD, do inglês (*Knowledge-Discovery in Databases)*, que é amplamente utilizada no âmbito empresarial, sendo aplicados cinco modelos preditivos que foram posteriormente comparados.

O processo de KDD consiste em um conjunto de atividades contínuas que compartilham o conhecimento encontrado a partir da base de dados. Segundo Fayyad et al. (1996), esse conjunto é composto por cinco passos: seleção dos dados; pré-processamento e limpeza de dados; transformação dos dados; mineração dos dados; e interpretação e avaliação dos

resultados. Esses processos são considerados interativos, podendo ser repetidos quantas vezes o analista precisar. O método pode ser ilustrado pela Figura 11.

Figura 11 – Método KDD



Fonte: Fayyad et al. (1996)

Na fase de seleção dos dados, o analista coleta os dados relevantes que serão utilizados na análise, que podem ser de diferentes fontes e possuírem características distintas, como por exemplo dados estruturados e não estruturados. Seguindo para o pré-processamento e limpeza dos dados, o analista verifica a qualidade dos dados, removendo valores nulos e valores duplicados, identificando *outliers*, corrigindo e removendo outras inconsistências. Já a etapa de transformação de dados consiste em agregações, reduções e sintetizações dos dados, convertendo o conjunto bruto dos dados em uma forma padrão de uso (Fayyad, 1996). Essas três etapas iniciais pertencem a análise exploratória dos dados.

O modelo de Aprendizado de Máquina é somente aplicado na etapa de mineração dos dados, também ocorrendo o reconhecimento de padrões, estatísticas e visualização dos dados, com objetivo de extrair informações relevantes da base de dados. Por fim, o desempenho do modelo é avaliado e interpretado, estando pronto para uso em outras bases de dados.

3.1 BASE DE DADOS

A base de dados foi escolhida na plataforma online Kaggle, um repositório onde são armazenados diversos conjuntos de dados para a análise. A base de dados escolhida não é sintética, ou seja, foi criada baseada na realidade com o objetivo de servir como base para analistas de dados na determinação da temperatura do rotor de motores síncronos de imã permanente. Os dados coletados são oriundos do departamento de engenharia elétrica da Universidade de Paderborn na Alemanha, e podem ser encontrados por meio do link: https://www.kaggle.com/datasets/wkirgsn/electric-motor-temperature.

3.2 FERRAMENTAS UTILIZADAS

A linguagem de programação utilizada para realizar as análises foi Python, que possui várias bibliotecas voltadas para a análise de dados (discutidas no Capítulo 4), permitindo a visualização, tratamento e previsão dos dados. O ambiente de desenvolvimento integrado utilizado foi o Jupyter Notebook, um *software* de código aberto com serviços para computação interativa em diversas linguagens de programação. Esse ambiente permite que usuário execute seus códigos na nuvem, não havendo a necessidade de manter seu código na máquina local.

4 EXPERIMENTO

Neste capítulo serão apresentados todos os passos que foram realizados na base de dados, desde sua escolha, até a seleção do melhor modelo, mostrando todos os resultados obtidos. Os procedimentos descritos neste capítulo são baseados na metodologia KDD que foi discutida no Capítulo 3.

4.1 SELEÇÃO DA BASE DE DADOS

Para a realização do presente experimento, foi utilizada uma base de dados contendo variáveis com o objetivo de aplicar um modelo de Aprendizado de Máquina na previsão da temperatura do rotor de um motor elétrico. Os dados em questão foram obtidos a partir de medições reais de um motor síncrono de ímã permanente realizados pela Universidade de Paderborn na Alemanha, que então foram disponibilizados na plataforma Kaggle.

A base de dados consiste em 185 horas de medição das variáveis do motor, que foram distribuídas em sessões que podem ser distinguidas por diversos perfis dentre as mais de 1 milhão e 300 mil entradas registradas na base. As características do motor são ainda divididas em 13 colunas, sendo uma delas a numeração do perfil, 11 variáveis preditoras e uma variável *target*.

As variáveis presentes na base de dados, com exclusão da numeração do perfil, são:

- Voltagem no eixo q em Volts: Voltagem_q[V]
- Voltagem no eixo d em Volts: Voltagem_d[V]
- Corrente no eixo q em Ampere: Corrente_q[A]
- Corrente no eixo d em Ampere: Corrente_d[A]
- Temperatura ambiente: Temp_Ambiente[°C]
- Temperatura do refrigerante do motor: Temp_Refrigerante[°C]
- Temperatura dos enrolamentos do estator: Temp_Enrolamentos_Estator[°C]
- Temperatura dos dentes do estator: Temp_Dentes_Estator[°C]
- Temperatura da coroa do estator: Temp_Coroa_Estator[°C]

- Velocidade do motor em rotações por minuto: Velo_Motor[rpm]
- Torque do motor em Newton-metro: Torque[Nm]
- Temperatura do rotor (variável *target*): Temp_rotor[°C]

4.1.1 Importação da Base de Dados

A base de dados está presente em uma tabela em formato CSV (do inglês *Comma Separated Values*), então para a importação é utilizada uma das bibliotecas mais utilizadas em Python, a biblioteca Pandas, que possui o método Read para ler os dados de diversas tabelas em diversos formatos e transformá-los em um *data frame*, que possui uma estrutura de dados bidimensional com os dados alinhados de forma tabular em linhas e colunas, permitindo qualquer tipo de alteração e análise dos dados. O código Python que permite a importação da base de dados para visualização e futuras análises é mostrado na Figura 12.

Figura 12 – Importação da base de dados.

```
import pandas as pd

df = pd.read_csv('measures_v2.csv')
```

Fonte: Próprio autor.

4.1.2 Apresentação da Base de Dados

Para conhecer melhor a base de dados com que se está trabalhando, aplica-se alguns métodos da biblioteca Pandas. Primeiramente, usa-se o método *head()* para ter uma primeira visualização da base, retornando apenas as cinco primeiras linhas, juntamente com as colunas, como é mostrado na Figura 13.

Figura 13 – Visualização da base de dados a partir do método *head()*.

	u_q	coolant	stator_winding	u_d	stator_tooth	motor_speed	i_d	i_q	pm	stator_yoke	ambient	torque	profile_id
0	-0.450682	18.805172	19.086670	-0.350055	18.293219	0.002866	0.004419	0.000328	24.554214	18.316547	19.850691	0.187101	17
1	-0.325737	18.818571	19.092390	-0.305803	18.294807	0.000257	0.000606	-0.000785	24.538078	18.314955	19.850672	0.245417	17
2	-0.440864	18.828770	19.089380	-0.372503	18.294094	0.002355	0.001290	0.000386	24.544693	18.326307	19.850657	0.176615	17
3	-0.327026	18.835567	19.083031	-0.316199	18.292542	0.006105	0.000026	0.002046	24.554018	18.330833	19.850647	0.238303	17
4	-0.471150	18.857033	19.082525	-0.332272	18.291428	0.003133	-0.064317	0.037184	24.565397	18.326662	19.850639	0.208197	17

Fonte: Próprio autor.

Nota-se que as variáveis estão em inglês, portanto foram renomeadas para português pelo método *rename*, também do Pandas, e realocadas em seguida para uma melhor compreensão. Outro método utilizado foi *describe()*, que retorna um resumo de todos os dados como mostra Figura 14.

Figura 14 – Retorno da base de dados a partir do método *describe()*.

	profile_id	Voltagem_q[V]	Voltagem_d[V]	Corrente_q[A]	Corrente_d[A]	Temp_Ambiente[°C]	Temp_Refrigerante[°C]	Temp_Enrolamentos_Estator[°C]	Temp_Dentes_Estator[°C]	Temp_Coroa_Estator[°C]	Velo_Motor[rpm]	Torque[Nm]	Temp_rotor[°C]
count	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06	1.330816e+06
mean	4.079306e+01	5.427900e+01	-2.513381e+01	3.741278e+01	-6.871681e+01	2.456526e+01	3.622999e+01	6.634275e+01	5.687858e+01	4.818796e+01	2.202081e+03	3.110603e+01	5.850678e+01
std	2.504549e+01	4.417323e+01	6.309197e+01	9.218188e+01	6.493323e+01	1.929522e+00	2.178615e+01	2.867206e+01	2.295223e+01	1.999100e+01	1.859663e+03	7.713575e+01	1.900150e+01
min	2.000000e+00	-2.529093e+01	-1.315304e+02	-2.934268e+02	-2.780036e+02	8.783478e+00	1.062375e+01	1.858582e+01	1.813398e+01	1.807669e+01	-2.755491e+02	-2.464667e+02	2.085696e+01
25%	1.700000e+01	1.206992e+01	-7.869090e+01	1.095863e+00	-1.154061e+02	2.318480e+01	1.869814e+01	4.278796e+01	3.841601e+01	3.199033e+01	3.171107e+02	-1.374265e-01	4.315158e+01
50%	4.300000e+01	4.893818e+01	-7.429755e+00	1.577401e+01	-5.109376e+01	2.479733e+01	2.690014e+01	6.511013e+01	5.603635e+01	4.562551e+01	1.999977e+03	1.086035e+01	6.026629e+01
75%	6.500000e+01	9.003439e+01	1.470271e+00	1.006121e+02	-2.979688e+00	2.621702e+01	4.985749e+01	8.814114e+01	7.558668e+01	6.146083e+01	3.760639e+03	9.159718e+01	7.200837e+01
max	8.100000e+01	1.330370e+02	1.314698e+02	3.017079e+02	5.189670e-02	3.071420e+01	1.015985e+02	1.413629e+02	1.119464e+02	1.011480e+02	6.000015e+03	2.610057e+02	1.136066e+02

Fonte: Próprio autor.

É possível notar que todas as variáveis possuem o mesmo valor na contagem, indicando que não há valores nulos, o que também pode ser feito pelo somatório dos valores nulos, a partir dos métodos *isnull()* e *sum()*, simultaneamente. Também é possível visualizar um valor médio para cada variável, assim como seus valores mínimos e máximos, podendo se inferir que não há discrepâncias aparentes e, consequentemente, *outliers*.

4.2 PRÉ-PROCESSAMENTO E LIMPEZA DE DADOS

Para a atual etapa, foram analisados a presença de valores nulos, valores duplicados e *outliers*, que comprometem o modelo de Aprendizado de Máquina, portanto devem ser retirados se estiverem presentes na base de dados.

4.2.1 Removendo Valores Nulos

Em Python, existem dois caminhos possíveis na biblioteca Pandas para verificar se existem valores nulos. A primeira consiste em fazer um somatório dos valores nulos e a segunda consiste em um método que retorna quantos valores não são nulos. Escolhendo o primeiro método, são utilizados métodos *isnull()* e *sum()*, retornando os seguintes valores da Figura 15.

Figura 15 – Análise de valores nulos.

```
df.isnull().sum()
```

```
profile_id                      0
Voltagem_q[V]                   0
Voltagem_d[V]                   0
Corrente_q[A]                   0
Corrente_d[A]                   0
Temp_Ambiente[°C]               0
Temp_Refrigerante[°C]           0
Temp_Enrolamentos_Estator[°C]   0
Temp_Dentes_Estator[°C]         0
Temp_Coroa_Estator[°C]          0
Velo_Motor[rpm]                 0
Torque[Nm]                      0
Temp_rotor[°C]                  0
dtype: int64
```

Fonte: Próprio autor.

Nota-se que não há valores nulos, portanto não foi necessário tomar nenhuma medida. Caso houvesse valores nulos, o tratamento desses valores poderia ser feito substituindo-os por valores médios, por exemplo, ou simplesmente removendo as linhas que ocorrem valores nulos, dependendo da necessidade e critério do analista. A ausência de valores nulos pode ser explicada por um filtro e limpeza dos dados feitas já no laboratório.

4.2.2 Removendo Duplicatas

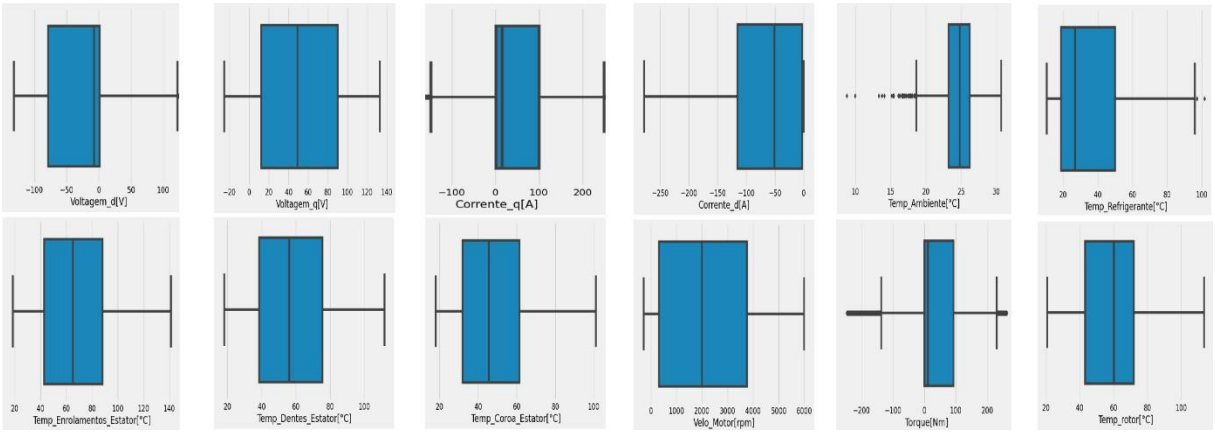
Algumas bases de dados podem apresentar valores duplicados, que podem afetar as propriedades de cada variável. Portanto, faz-se necessário remover esses valores por meio do método *drop_duplicates()*, da biblioteca Pandas. Porém, realizando o método *len()* para verificar o tamanho do *data frame* menos o seu tamanho sem as duplicatas, houve um retorno zero, ou seja, não há duplicatas. A ausência de duplicatas também pode ser explicada por um filtro e limpeza dos dados em laboratório.

4.2.3 Tratamento de Outliers

Por fim, é necessário identificar se há *outliers*, que são valores que diferem drasticamente de todos os outros, ou seja, um dado que foge da normalidade. O analista precisa analisar a existência desses valores juntamente com os dados como um todo, analisando se estes precisam ser realmente retirados, pois afetam o modelo do Aprendizado de Máquina, ou se eles podem ser significativos para base.

O método utilizado para identificar a existência desses valores foi o *Box Plot*, que está na biblioteca de visualização *Seaborn* e permite a visualização gráfica dos valores de cada variável em quartis, assim como os *outliers*, trazendo uma disposição gráfica comparativa. O *Box Plot* é formado pelo valor máximo e mínimo representados pelas linhas extremas, por um retângulo que representa os quartis, sendo que a linha interna do retângulo seria a mediana dos valores, e pelos *outliers* que são representados por círculos ou losangos, dependendo da plataforma utilizada. O método realizado para todas as variáveis está representado pela Figura 16.

Figura 16 – Análise *outliers*.



Fonte: Próprio autor.

Pelos gráficos, é possível notar que a temperatura ambiente e a temperatura refrigerante apresentaram alguns *outliers*, porém isso só indica que houve dias mais frios que o normal e que o refrigerante apenas esquentou em apenas dois casos, como a temperatura do rotor não apresentou valores anormais, não houve tratamento para estes casos. O toque apresentou mais *outliers*, mas foi levado em conta que o motor pode estar operando com grandezas rotacionais mais altas ou mais baixas, sendo uma característica do processo em questão.

4.3 TRANSFORMAÇÃO DOS DADOS

Como todas as variáveis presentes na base de dados possuem valores não categóricos, a única transformação necessária para a utilização da base de dados foi a divisão da base de dados em treino e teste. Os dados de treino são os aqueles utilizados para o treinamento do modelo, representando geralmente dois terços da totalidade de dados, enquanto os dados de teste são aqueles apresentados ao modelo após a sua criação e treinamento, simulando as previsões reais que o modelo realizará, permitindo avaliar seu desempenho.

O primeiro passo para realizar essa divisão começa na separação entre as variáveis preditoras (X) e a variável *target* (y). Em seguida, a partir da biblioteca *Sklearn* (biblioteca mais utilizada no Aprendizado de Máquina) X e y ainda são divididos em treino X, treino y, teste X

e teste y com o método *train_test_split()*, fazendo uma divisão de dois terços dos dados para as variáveis de treino e um terço para as variáveis de teste.

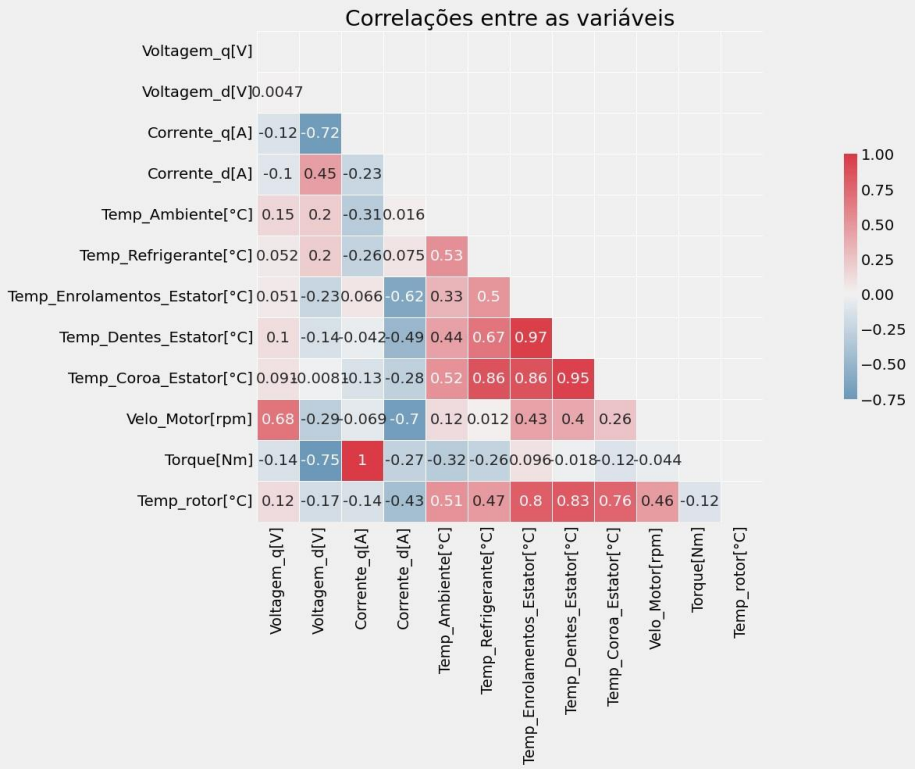
4.4 MINERAÇÃO DOS DADOS

Nessa etapa foram realizados vários gráficos para analisar as correlações entre as variáveis e como a variável *target* se comporta conforme as variações de algumas variáveis principais. Além disso, modelos de Aprendizado de Máquina foram aplicados para a previsão da temperatura do motor.

4.4.1 Correlações

Para analisar as correlações entre as variáveis é usado um *heatmap*, que é um gráfico que utiliza cores como referência para facilitar o entendimento das informações, ou seja, um mapa de calor. Os valores entre as variáveis indicam a linearidade entre as variáveis, sendo que valores próximos de 1 e -1 indicam uma forte correlação entre as variáveis e valores próximos de 0 indicam baixa correlação. O resultado da correlação entre as variáveis pode ser visto na Figura 17.

Figura 17 – Gráfico das correlações.



Fonte: Próprio autor.

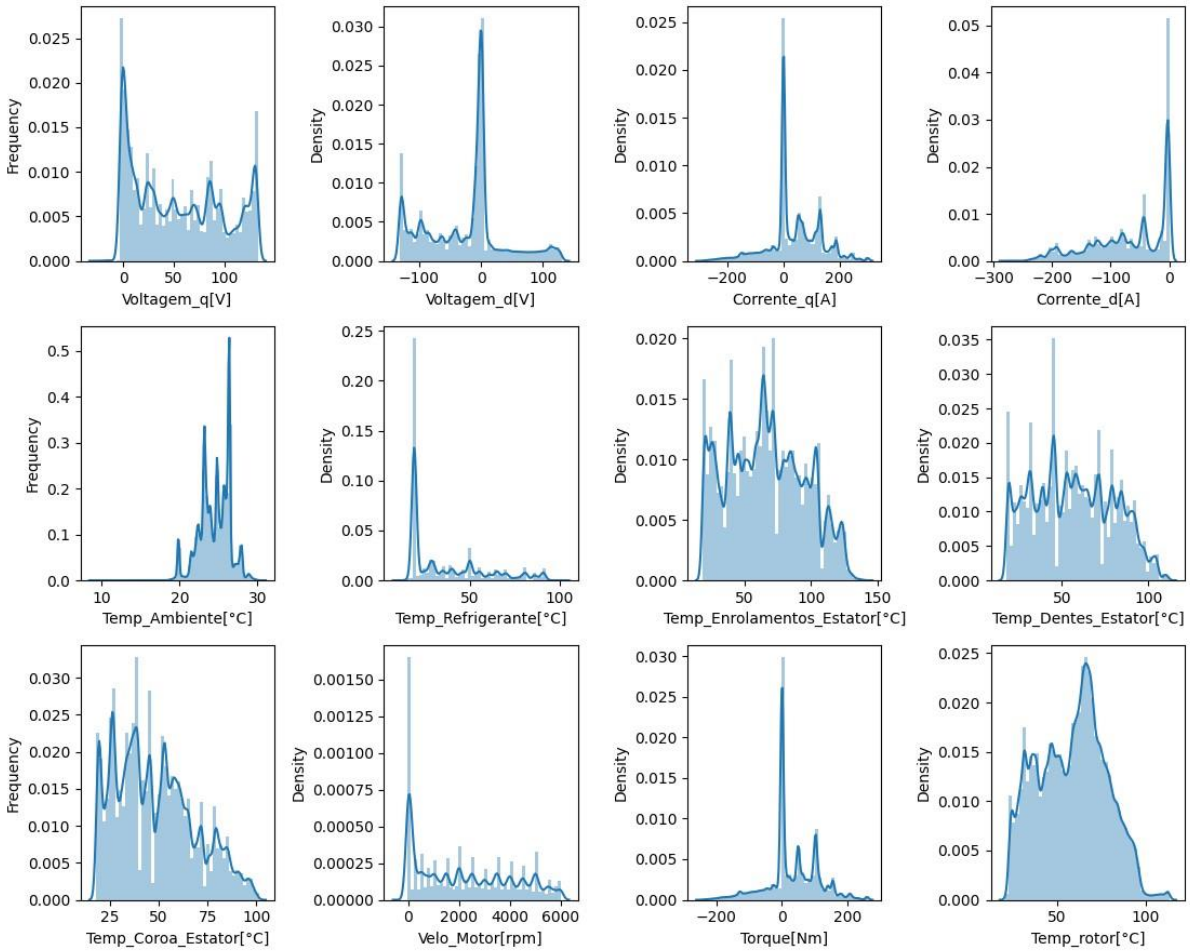
Olhando para as correlações é possível observar que apenas as temperaturas relacionadas ao estator possuem forte correlações com a temperatura do rotor (variável *target*), além de apresentar forte relações entre si. Também é interessante notar que a temperatura ambiente possui uma correlação relativamente alta com a temperatura refrigerante e com as temperaturas do estator (como já era de se esperar). Por fim, nota-se que torque e corrente q apresentaram uma correlação perfeita, indicando uma máxima relação de linearidade, portanto, para evitar a multicolinearidade, retirou-se a variável Torque posteriormente para a aplicação do modelo, apresentando menor correlação com a variável *target* entre ambas.

A partir das percepções retiradas do *heatmap*, pensou-se em gráficos para melhor visualizar como essas variáveis se comportam, escolhendo as variáveis que possuem maior correlação com a variável *target* e que, portanto, interferem mais em seu resultado

4.4.2 Distribuição das Variáveis

Primeiramente, para se ter uma visão mais geral das variáveis, foram realizados gráficos de distribuição que indicam a densidade dos dados por meio de um histograma, a partir do método *distplot()* do *Seaborn*, como é mostrado na Figura 18.

Figura 18 – Distribuição das variáveis.



Fonte: Próprio autor.

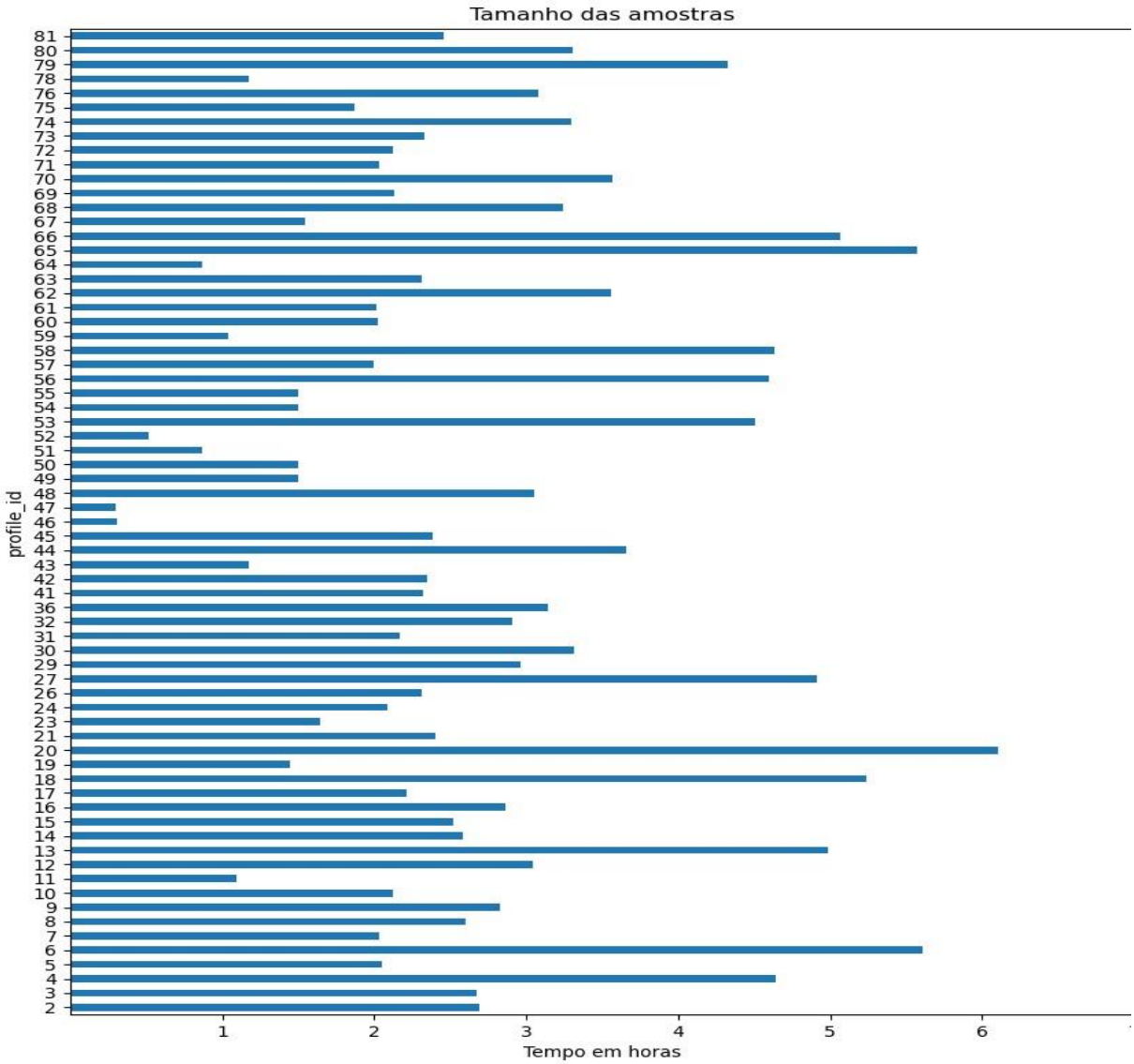
Nota-se que não há distribuições normais dentre as variáveis, indicando que talvez uma Regressão Linear não seja o melhor modelo para a base de dados. Para voltagem e corrente é possível notar que o maior pico entre eles está no zero, indicando que o motor pode estar desligado nesses casos, o que pode ser comprovado pelo pico de velocidade e torque em zero. Também pode se inferir que a maioria das temperaturas ambiente estão em torno de

temperaturas médias para um verão ou primavera na Alemanha (local de medição), e que as temperaturas do estator e do rotor estão bem distribuídas para quase todos os valores.

4.4.3 Análise por Perfil

Cada linha da base de dados representa um instante dos dados do sensor em uma taxa de amostragem de 2 Hz. A partir disso, os perfis foram agrupados para analisar todo o período de amostragem de um determinado perfil e como as variáveis se comportam com o passar do tempo. Cada perfil possui um tamanho em horas diferentes, portanto foram feitas análises para perfis que possuem mais horas de medição. A Figura 19 mostra o tamanho em horas de cada perfil.

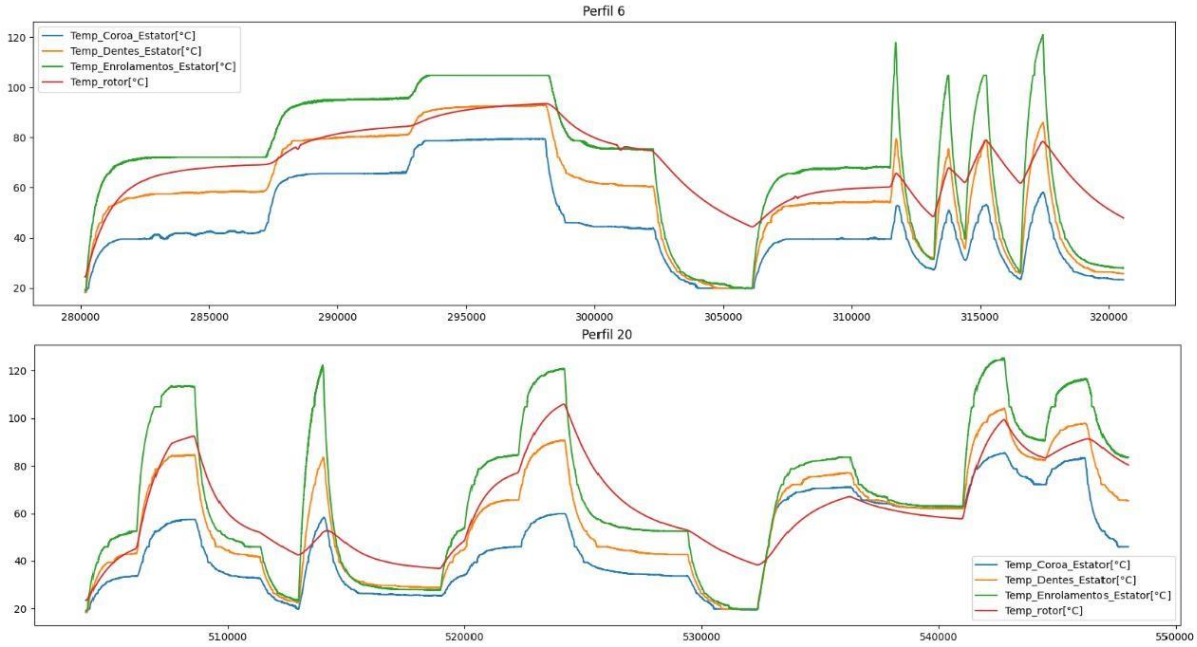
Figura 19 – Perfis agrupados em horas.



Fonte: Próprio autor.

A partir da Figura 19, nota-se que os perfis 6, 20, 65 e 66 são os perfis que possuem maior tempo em horas. Assim, os perfis 6 e 20 foram escolhidos para realizar as análises. Tendo em vista as análises de correlações, foram analisadas as temperaturas do estator em relação à temperatura do rotor, usando os perfis 6 e 20, para visualizar como a temperatura do rotor se comporta com essas variáveis de forte correlação, conforme Figura 20.

Figura 20 – Análise variáveis por perfil.



Fonte: Próprio autor.

Como já se era esperado, pode-se inferir que todas as quatro temperaturas se comportam seguindo o mesmo padrão e que apresentam grandeza semelhante entre ambos os perfis. Outa análise que pode ser feita é que se percebe a correlação mais forte entre as temperaturas do estator, que pode ser explicado pelas proximidades entre os componentes, e que a temperatura do rotor possui uma resposta não tão brusca às variações das temperaturas do estator.

4.4.4 Aplicação dos Modelos de Aprendizado de Máquina

Foram aplicados cinco modelos de Aprendizado de Máquina, sendo eles regressão linear, regressão polinomial, árvore de decisão e aprendizado profundo.

Para todos os modelos são utilizadas metodologias semelhantes e todos com métodos da biblioteca Sklearn. Basicamente, cria-se um objeto do respectivo modelo que é treinado utilizando os dados de treino X e treino y, por meio do método *fit()*. Em seguida, o modelo realiza a predição dos valores de y (variável target) utilizando os dados de teste X, por meio do método *predict()*.

Nos modelos de regressão linear, regressão polinomial e árvore de decisão foram utilizados os parâmetros predefinidos pela biblioteca Sklearn, havendo apenas alteração dos graus do polinômio na regressão polinomial:

- Regressão linear:

class sklearn.linear_model.LinearRegression(*, *fit_intercept=True*, *normalize='deprecated'*, *copy_X=True*, *n_jobs=None*, *positive=False*). (Scikit Learn, 2022)

- Regressão Polinomial:

class sklearn.preprocessing.PolynomialFeatures(*degree=2*, *, *interaction_only=False*, *include_bias=True*, *order='C'*). (Scikit Learn, 2022)

- Árvore de decisão:

class sklearn.tree.DecisionTreeRegressor(*, *criterion='squared_error'*, *splitter='best'*, *max_depth=None*, *min_samples_split=2*, *min_samples_leaf=1*, *min_weight_fraction_leaf=0.0*, *max_features=None*, *random_state=None*, *max_leaf_nodes=None*, *min_impurity_decrease=0.0*, *ccp_alpha=0.0*). (Scikit Learn, 2022)

Já para o modelo de Aprendizado Profundo usa a biblioteca *Tensor Flow*, que é uma biblioteca desenvolvida pela Google, e o *Keras*, que é uma biblioteca de Python para usar o *Tensor Flow*. Usando essas bibliotecas o analista consegue criar a sua própria estrutura de neurônios. Neste trabalho, a estrutura criada foram cinco camadas, sendo as três primeiras compostas por quatro neurônios, a quarta por dois neurônios e a última por apenas um neurônio.

4.5 INTERPRETAÇÃO E AVALIAÇÃO DOS RESULTADOS

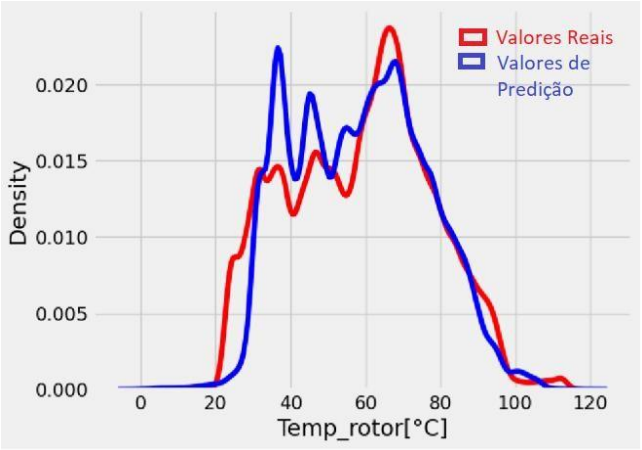
Para cada um dos modelos foi realizado um gráfico comparando os valores reais dos dados de teste y e os valores que foram previstos pelo modelo. Tais gráficos servem para ter uma visualização mais concreta de como os valores de predição se encaixam com os valores

reais da base de dados, analisando os dados a partir da densidade dos valores. Posteriormente, foram realizados o MSE, o RMSE e a variância explicada a partir de métodos do *Sklearn*.

4.5.1 Resultados da Regressão Linear

Após aplicado o modelo de Regressão Linear nos dados de treino da base de dados, realizou-se um gráfico comparando os valores que o modelo previu utilizando as variáveis de predição dos dados de teste e os valores reais da variável *target* dos dados de teste. Os resultados obtidos graficamente e as métricas para o modelo estão na figura 21.

Figura 21 – Resultados para regressão linear.



MSE	52.30
RMSE	7.23
Variância Explicada	0.85

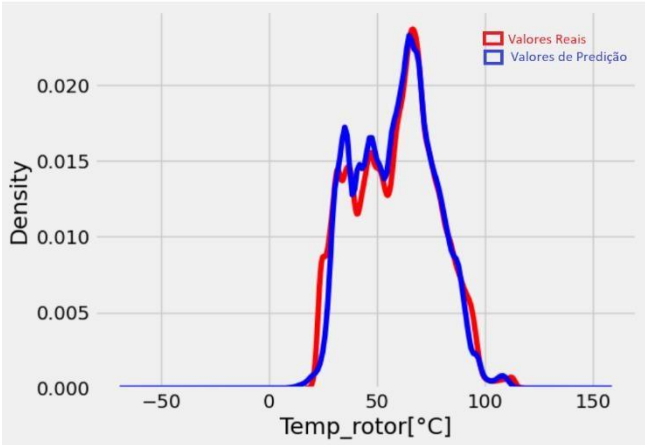
Fonte: Próprio autor.

Os resultados mostram que o modelo até seguiu o mesmo padrão de altos e baixos dos valores reais, porém houve densidades relativamente altas para alguns valores, principalmente por volta de 40°C. Isso pode ser comprovado pela variância explicada que apresentou um valor relativamente bom, mas que precisaria ser um pouco mais alta (em torno de 0.9). Os valores de MSE e RMSE foram um pouco melhores que o esperado, porém um erro de temperatura na ordem de 7°C não seria adequada para uma visão empresarial.

4.5.2 Resultados da Regressão Polinomial

O mesmo procedimento realizado para a Regressão Linear foi utilizado para a Regressão Polinomial, tanto para ordem 2 quanto para ordem 3. Os resultados para a Regressão Polinomial de ordem 2 podem ser vistos na Figura 22.

Figura 22 – Resultados para Regressão Polinomial de ordem 2.

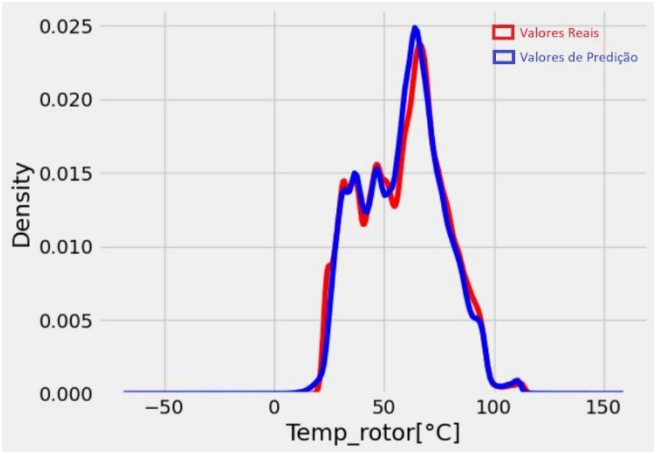


MSE	36.98
RMSE	6.08
Variância Explicada	0.89

Fonte: Próprio autor.

Os resultados demonstraram melhora comparados a Regressão Linear, mostrando que os dados são mais bem descrevidos por uma equação polinomial. Portanto, foi realizado o teste para uma Regressão Polinomial de ordem 3 como mostra a Figura 23.

Figura 23 – Resultados para Regressão Polinomial de ordem 3.



MSE	22.54
RMSE	4.74
Variância Explicada	0.93

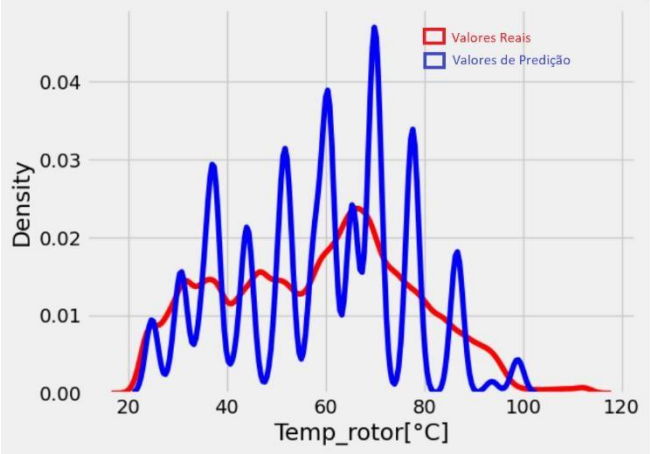
Fonte: Próprio autor.

Nota-se uma melhora significativa dos resultados, com menores erros e um valor ótimo para a variância explicada, mostrando que o modelo explica a maioria dos resultados da base de dados. Seria possível aumentar o grau do polinômio gradativamente, aumentando assim a variância explicando e diminuindo os erros, porém isso levaria a um *overfitting* do modelo, o que não é desejado.

4.5.3 Resultados da Árvore de Decisão

Posteriormente, a mesma metodologia foi aplicada para o modelo de Árvore de Decisão. A Figura 24 mostra como é o comportamento gráfico dos valores reais em comparação com os valores de predição e os resultados obtidos das métricas.

Figura 24 – Resultados para árvore de decisão.



MSE	66.27
RMSE	8.14
Variância Explicada	0.82

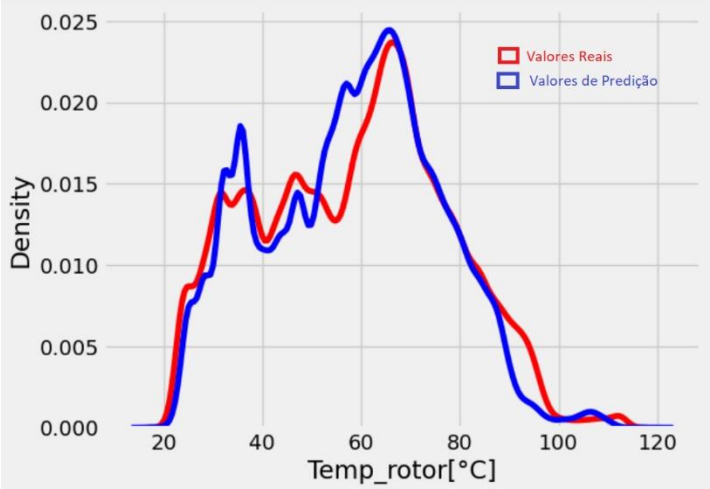
Fonte: Próprio autor.

Nesse caso, o modelo não se encaixou bem aos resultados, mostrando que o modelo de Árvore de Decisão gerou previsões oscilatórias, afetadas pelo esquema de árvore de escolher entre uma decisão ou outra.

4.5.4 Resultados do Aprendizado Profundo

Por fim, aplicando o modelo de Aprendizado Profundo na base de dados, foi possível obter os resultados e as métricas do modelo em questão. A Figura 25 traz o gráfico e os resultados numéricos comparando os valores reais e valores de predição.

Figura 25 – Resultados para Aprendizado Profundo.



MSE	44.15
RMSE	6.64
Variância Explicada	0.87

Fonte: Próprio autor.

Pode-se inferir que os resultados foram relativamente bons, mas ainda abaixo dos valores obtidos para a Regressão Polinomial. Esses valores podem ser otimizados mudando a estrutura dos neurônios, criando mais camadas e mais neurônios por camadas, levando à um resultado superior ou equivalente a Regressão Polinomial. Porém, esse modelo possuiu um processamento muito lento em comparação aos outros e a mudança nos neurônios diminuiria ainda mais a performance, não sendo algo viável em um âmbito empresarial.

4.5.5 Análise dos Resultados

Levando sem consideração todos os resultados dos modelos com seus respectivos gráficos e métricas de avaliação, além do processamento, o modelo de Regressão Polinomial de terceiro grau foi o mais bem avaliado. Esse modelo apresentou os menores valores para o MSE e o RMSE, assim como o melhor resultado para a variância explicada. Isso implica que esse modelo foi o que melhor se encaixou e representou os dados da base, então seria o modelo escolhido para a aplicação futura em uma outra base de dados semelhante.

5 CONCLUSÃO

Conforme exercício realizado neste trabalho, conclui-se que o Aprendizado de Máquina pode ser útil para a previsão da temperatura de um motor do tipo PMSM, o que traria ótimos resultados para a produção e controle desse tipo de motor, já que a base de dados utilizada foi composta por dados reais medidos dentro de um laboratório.

O fato de ter sido obtido um resultado extremamente eficiente, revela que os tratamentos na base aplicados e a forma de validação foi também muito bem selecionada. Nota-se também a importância de aplicar interpretações dos dados do começo ao fim do exercício, que muitas vezes darão *insights* sobre o problema em questão, e norteará as decisões dos tratamentos de dados que precisarão ser feitos para extrair um modelo com boa performance.

Também se nota a importância da compressão do conteúdo da base de dados com que se trabalha, para saber o melhor caminho a se tomar, como fazer o tratamento das variáveis e entender com clareza o que os resultados representam. A partir desse conhecimento pode se ter uma base para escolher quais os melhores modelos de Aprendizado de Máquina a serem empregados.

Por fim, apresentados os modelos de Aprendizado de Máquina utilizados no código, juntamente com as respectivas métricas de avaliação, pode-se concluir que o modelo que melhor representa e melhor faz a predição foi a Regressão Polinomial de terceiro grau, que possui ótimas métricas evitando ao máximo o *overfitting*.

Como um trabalho futuro, seria possível aplicar o mesmo modelo construído em uma base de dados semelhante e real durante os processos de produção de um motor PMSM, o que ajudaria uma indústria a usar menos material e controlar melhor as estratégias de produção e tomada de decisões para que o motor opere em capacidade máxima.

REFERÊNCIAS

ALMEIDA, A.; CARVALHO, F.; MENINO, F.. **Introdução ao machine learning:** de alunos para alunos. *[S.l]*: Dataat, 2017. Disponível em: https://dataat.github.io/introducao-ao-machine-learning/index.html#licen%C3%A7a. Acesso em: 27 set. 2022.

AMARAL, F. **Introdução à ciência de dados**: mineração de dados e big data. *[S.l]*: Alta Books, 2016.

ANALYTICS VIDHYA CONTENT TEAM. **Modelagem baseada em treearvore**, Bombaim: Analytics Vidhya, [2019]. Disponível em: https://www.vooo.pro/insights/um-tutorial-completo-sobre-a-modelagem-baseada-em-treearvore-do-zero-em-r-python/. Acesso em: 11 ago. 2022.

CHEIN, F. **Introdução aos modelos de regressão linear**. 2. ed. Brasília: Enap, 2019.

CHEN, S.; XU, Z.; WANG, X.; ZHANG, C. **Ambient air pollutants concentration prediction during the COVID-19:** a method based on transfer learning. 2022. Tese (Doutorado em Negócios) – Faculdade de Negócios, Universidade de Sichuan, Chengdu, 2021. Disponível em: https://www.sciencedirect.com/science/article/pii/S0950705122010899?via%3Dihub. Acesso em: 13 jan. 2023.

CLAESEN, M.; MOOR, B. Hyperparameter search in machine learning. *In*: MIC 2015: THE XI METAHEURISTIC INTERNATIONAL CONFERENCE, 11., 2019, Agadir. **Proceedings** […]. Leuven: Department of Electrical Engineering (Esat) – Iminds, 2015. p. 1-14. Disponível em: https://arxiv.org/pdf/1502.02127.pdf. Acesso em: 28 dez. 2022.

DOMINGOS, P. **A revolução do algoritmo mestre**. *[S.l]*: Editorial Presença, 2017.

FAYYAD, U. M.; PIATETSKY - SHAPIRO, G.; SMYTH, P. **From data mining to knowledge in databases**. 3rd ed. Menlo Park: AI Magazine, 1996.

FRADKOV, Alexander L. Early history of machine learning. *In*: 21. IFAC WORLD CONGRESS, 21., 2020, Berlin. **Proceedings** [...]. São Petersburgo: Elsevier, 2021. p. 1385-1390.
Disponível em: https://reader.elsevier.com/reader/sd/pii/S2405896320325027?token=6974168BE5E2B092D354C1F4FE69D2C23EA8EEE52E4AC409F78CE6FDAEE5812680221F5E704B058011BC61F524F74383&originRegion=us-east-1&originCreation=20230113170933. Acesso em: 06 jan. 2023.

GOODFELOW, I.; BENGIO, Y.; COURVILLE; A. **Deep learning**: a mit press book. *[S.l]*: Mit Press, 2016. Disponível em: http://www.deeplearningbook.org. Acesso em: 12 nov. 2022.

JONES, L. A.; LANG, J. H. A state observer for the permanent-magnet synchronous motor. ***IEEE Transactions on Industrial Electronics***, Liverpool, v. 36, n. 3, p. 374-382, 1989.

MAHESH, B. Machine learning algorithms: a review. **International Journal of Science and Research**, Raipur, v. 9, n. 1, p. 381-386, 2020. Disponível em: https://www.researchgate.net/profile/Batta-Mahesh/publication/344717762_Machine_Learning_Algorithms_-A_Review/links/5f8b2365299bf1b53e2d243a/Machine-Learning-Algorithms-AReview.pdf?eid=5082902844932096. Acesso em: 12 jan. 2023.

MOHAMMED, M.; KHAN, M.; BASHIER, E. B. M. **Machine learning**: algorithms and applications. Boca Raton: Crc Press, 2016.

MIGUEL, T. **Regressão polinomial**. aprenderdatascience, 2020. Disponível em: https://aprenderdatascience.com/regressao-polinomial/. Acesso em: 25 out. de 2022.

MITCHELL, T. M. **Machine learning**. *[S.l]*: McGraw-Hill Science/Engineering/Math, 1997.

POMERLEAU, D. A. **Neural network simulation at waq speed**: how we got 17 million connections per second. 5th ed. San Diego: Proceedings of Ieee International Conference on Neural Networks, 1988.

PRODANOV, C. C.; FREITAS, E. C. **Metodologia do trabalho científico:** métodos e técnicas da pesquisa e do trabalho acadêmico. 2. ed. Novo Hamburgo: Universidade Feevale, 2013.

PROVOST, F; FAWCETT, T. Data science and its relationship to big data and data-driven decision making. *In*: PROVOST, F; FAWCETT, T. **Big data**. *[S.l]*: Mary Ann Liebert, 2013. p. 51-59.

PSAROS, A. F.; MENG, X.; ZOU, Z.; GUO, L.; KARNIADAKIS, G. **Uncertainty quantification in scientific machine learning**: methods, metrics, and comparisons. 2022. Tese (Doutorado em Matemática Aplicadada) – Departamento de Matemática Aplicada, Universidade de Brown, Providence, 2022.

REZENDE, E. M. **Máquinas síncronas de pólos girantes**. Rio de Janeiro: Serviço Gráfico do IBGE, 1963.

RODRIGUES, M. O tratamento e análise de dados. *In* Silvestre, H. C.; Araújo, J. F. **Metodologia para a investigação social**. Lisboa: Escolar Editora, 2021. p. 179-230.

RUSSEL, S.; NORVIG, P. **Artificial intelligence**: a modern approach. 2nd ed. *[S.l]*: Prentice Hall, 2002.

SCIKIT LEARN. **Sklearn.linear_model.LinearRegression.** Disponível em: https://scikitlearn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. Acesso em: 26 nov. 2022.

SCIKIT LEARN. **Sklearn.tree.DecisionTreeRegressor.** Disponível em: https://scikitlearn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html. Acesso em: 26 nov. 2022.

SCIKIT LEARN. **Sklearn.preprocessing.PolynomialFeatures**. Disponível em: https://scikitlearn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html. Acesso em: 26 nov. 2022.

SHINDE, P. P.; SHAH, S. A review of machine learning and deep learning applications. *In*: 4. INTERNATIONAL CONFERENCE ON COMPUTING COMMUNICATION CONTROL AND AUTOMATION**,** 18617885., 2018, Pune. **Anais** […]. Liverpool: Ieee, 2019. Disponível em: https://ieeexplore.ieee.org/abstract/document/8697857. Acesso em: 28 dez. 2022.

TAVARES, T. A. **Estudo do sistema de acionamento do motor síncrono à ímã permanente com partida direta a rede elétrica**. 2018. Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica) – Departamento de Engenharia Elétrica, Universidade Federal de Campina Grande, Campina Grande, 2018.
Disponível em: http://dspace.sti.ufcg.edu.br:8080/jspui/bitstream/riufcg/18957/1/THALES%20DE%20AGUIAR%20TAVARES%20-%20%20TCC%20ENG.%20EL%c3%89TRICA%202018..pdf. Acesso em: 11 nov. 2022.

ZHOU, Z. **Machine learning**. Nanjing: Springer, 2021.

BIBLIOGRAFIA CONSULTADA

ALPAYDIN, E. **Intorduction to machine learning**. 3rd ed. *[S.l]*: Mit Press, 2020.

MONARD, M. C. Conceitos sobre Aprendizado de Máquina. *In*: REZENDE, S. O. **Sistemas inteligentes**: fundamentos e aplicações. *[S.l]*: Manole, 1994. v. 1, cap. 4. p. 39-56. Disponível em: https://dcm.ffclrp.usp.br/~augusto/publications/2003-sistemasinteligentes-cap4.pdf. Acesso em: 26 set. 2022.

SILVESTRE, A. L. **Análise de dados e estatística descritiva**. *[S.l]*: Escolar Editora, 2007.

SLEMON, G. R. **Electric machines drives**. Reading: Addison Wesley, 1992.