UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

DOMINGOS PEREGO JUNIOR

PREVISÃO DE POTÊNCIA GERADA POR SISTEMA FOTOVOLTAICO NA CIDADE DE FOZ DO IGUAÇU DE ACORDO COM DADOS METEOROLÓGICOS

MEDIANEIRA

2022

DOMINGOS PEREGO JUNIOR

PREVISÃO DE POTÊNCIA GERADA POR SISTEMA FOTOVOLTAICO NA CIDADE DE FOZ DO IGUAÇU DE ACORDO COM DADOS METEOROLÓGICOS

Power prediction of a photovoltaic system located in Foz do Iguaçu city according to meteorological data

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador: Evandro André Konopatzki.

Coorientador: Leandro Antônio Pasa.

MEDIANEIRA

2022

DOMINGOS PEREGO JUNIOR

PREVISÃO DE POTÊNCIA GERADA POR SISTEMA FOTOVOLTAICO NA CIDADE DE FOZ DO IGUAÇU DE ACORDO COM DADOS METEOROLÓGICOS

Trabalho de Conclusão de Curso de Graduação apresentado como requisito para obtenção do título de Bacharel em Engenharia Elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 03/novembro/2022

_____

Evandro André Konopatzki
Doutorado
Universidade Tecnológica Federal do Paraná

_____

Leandro Antônio Pasa
Doutorado
Universidade Tecnológica Federal do Paraná

_____

Tatiane Tambarussi Thomaz
Doutorado
Universidade Tecnológica Federal do Paraná

_____

José Airton Azevedo dos Santos
Doutorado
Universidade Tecnológica Federal do Paraná

MEDIANEIRA

2022

RESUMO

A energia solar vem ganhando espaço nos últimos anos como uma alternativa renovável à utilização de combustíveis fósseis. No entanto, sua geração está atrelada às condições meteorológicas. Dessa forma, é proposta a utilização dos algoritmos de regressão *Support Vector Machines*, Árvores de Decisão, *Nearest Neighbors*, e Rede Neural Artificial para prever a potência gerada por um sistema fotovoltaico localizado na cidade de Foz do Iguaçu com base em dados meteorológicos. Os dados obtidos foram tratados, e realizou-se a seleção de variáveis por meio de uma análise exploratória, onde foi avaliado o coeficiente de correlação e gráficos de dispersão. Nessa análise determinou-se que as variáveis que mais influenciam na potência gerada são: radiação solar, temperatura, umidade, ponto de orvalho, chuva, hora e mês. Posteriormente os modelos de regressão foram treinados e validados por meio de validação cruzada. Utilizou-se 10% do conjunto de dados para teste do modelo por meio das métricas RMSE, MAPE e R^2, sendo que, de acordo com essas métricas, os melhores algoritmos são o SVR e RNA. O Teste de Dunnett com intervalo de confiança de 95% foi utilizado para comparar as médias obtidas por cada um dos algoritmos com os valores reais, e seu resultado indicou que não há diferença significativa entre elas, sugerindo que todos os algoritmos propostos são adequados para a realização da previsão.

Palavras-chave: geração de energia fotovoltaica; aprendizado de máquina; análise de regressão.

ABSTRACT

Photovoltaic power is getting popular lately as a reneweable alternative to fossil fuel consumption. However, its power generation is linked to meteorological factors. The use of the regression algorithms Support Vector Machines, Decision Trees, Nearest Neighbors, and an Artificial Neural Network are proposed to predict the amount of power generated by a solar power plant located in Foz do Iguaçu city, based on meteorological data. The data was treated, and a feature selection was performed after an exploratory data analysis, evaluating the correlation coefficient and scatter plots. In this context, the selected variables were: solar radiation, temperature, humidity, dew point, rain, hour and month. After that, the regression models were trained and validated using cross validation. 10% of the dataset was designated to be used as a test set, using RMSE, MAPE and R^2 as evaluation metrics. According to these metrics, the best algorithms are SVR and ANN. A Dunnett's Test with a Confidence Interval of 95% was performed to compare the means obtained by each one of the algorithms and the real data, pointing out that there is not a significant difference between the algorithms and the real data, suggesting that all of them are fitted for the prediction.

Keywords: photovoltaic power generation; machine learning; regression analysis.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

Adam	Adaptive Moment Estimation
ANOVA	Análise de Variância
CSV	Comma Separated Values
DMS	Diferença Mínima Significativa
EPE	Empresa de Pesquisa Energética
INMET	Instituto Nacional de Meteorologia
KNR	K Neighbors Regressor
MAPE	Erro Médio Absoluto Percentual
ML	Machine Learning
MLP	Rede Perceptron Multicamadas
MSE	Erro Quadrático Médio
QM	Quadrado Médio dos Resíduos
RBF	Radial Basis Function
RMSE	Raiz Quadrada do Erro Quadrado Médio
RNA	Rede Neural Artificial
SVM	Support Vector Machines
SVR	Support Vector Regression
UTC	Tempo Universal Coordenado
XLSX	Excel Spreadsheet

SUMÁRIO

TOC \o "2-5" \h \t "Heading 2,2,Heading 3,3,Heading 4,4,Heading 5,5,Titulo 1;1;Título REFERÊNCIAS;6;Pós-Textuais - APÊNDICES;

1 INTRODUÇÃO

A crescente preocupação com a emissão de poluentes gerada pelas usinas termoelétricas fez surgir um interesse em fontes renováveis de geração de energia, como a eólica e a solar.

A potência gerada pelas fontes de energia supracitadas dependem das condições meteorológicas, e, portanto, tratam-se de formas de geração de energia intermitentes. Dessa forma, diferentemente das fontes de energia despacháveis, é difícil prever exatamente a quantidade de energia que será gerada. Portanto, é de extrema importância que se possa modelar a dependência entre energia gerada e as condições meteorológicas.

Segundo Cardoza, Uribe e Palacios (2018), a previsão de cenários operacionais é um dos fatores mais importantes para o bom funcionamento das empresas do setor energético, sendo crucial na gestão de projetos e influenciando na tomada de decisão acerca da viabilidade econômica e operacional de um projeto.

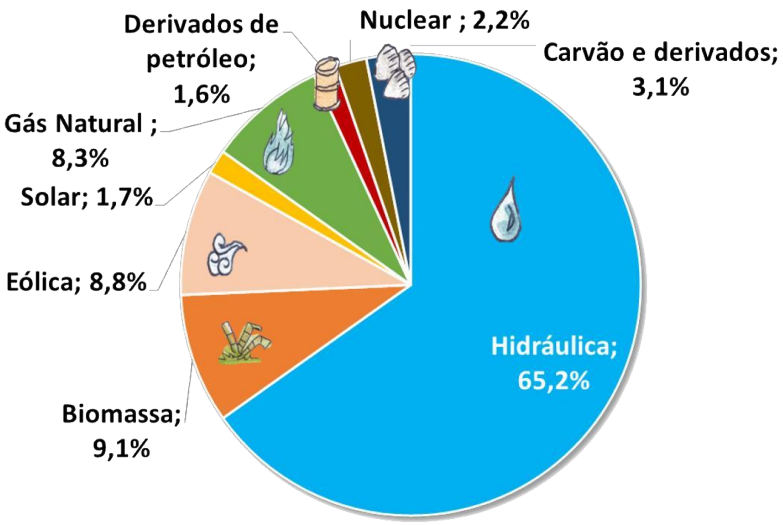
Para Francisco *et al*. (2019), conhecer a influência de parâmetros meteorológicos é importante como forma de determinar a viabilidade de implementação de painéis fotovoltaicos em diferentes regiões do país.

Já Wang *et al*. (2018) afirmam que a inconstância e aleatoriedade da geração fotovoltaica causada pelos fatores meteorológicos à ela atrelados, acabam dificultando a sua disseminação, sendo de extrema importância o conhecimento acerca da influência desses parâmetros.

1.1 Delimitação do tema

Ao longo dos anos, diante da necessidade de se expandir a matriz energética para superar a dependência de combustíveis fósseis, a energia fotovoltaica passou a ser vista como uma alternativa, se tornando fonte de interesse econômico e acadêmico por se tratar de uma energia renovável (LANA *et al.,* 2015). No entanto, segundo a Empresa de Pesquisa Energética (EPE), a energia solar representa apenas 1,7% da matriz energética brasileira, conforme visto na Figura 1:

Figura 1 – Matriz Elétrica Brasileira em 2020.



Fonte: EPE (2022)

Segundo Tiepolo *et al.* (2018), a região oeste do Paraná, onde está localizada a cidade de Foz do Iguaçu, possui grande potencial de geração fotovoltaica, sendo que a média de irradiação global horizontal é de 1.744$ kWh.m^2.

A relação entre energia gerada por um painel fotovoltaico e os dados meteorológicos pode ser modelada por meio da utilização de algoritmos de regressão.

1.2 Problemas e hipóteses

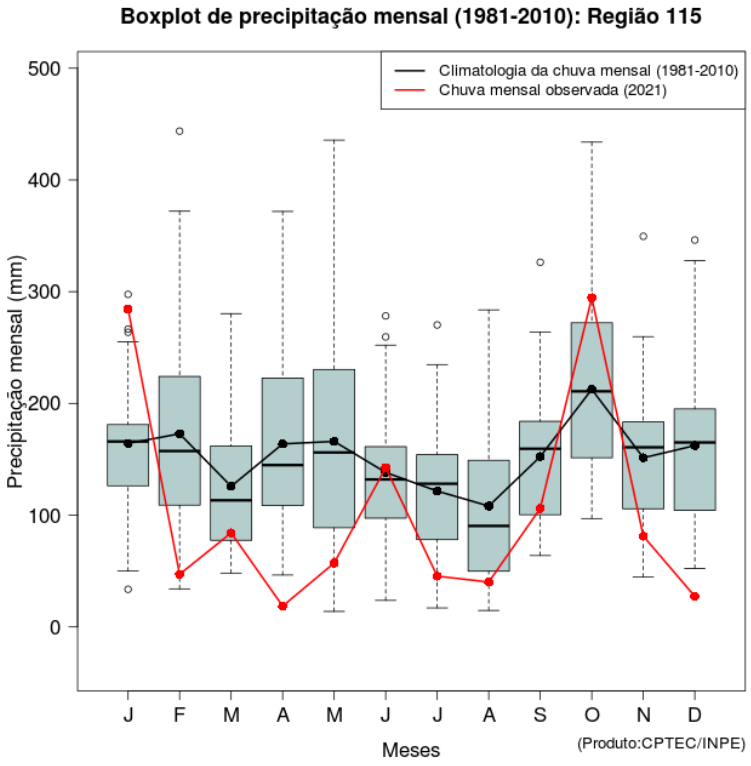
É de grande importância conseguir estimar a potência gerada por um sistema de geração fotovoltaico, para que se possa dimensiona-lo mais adequadamente. Ademais, conseguir prever a potência gerada é interessante para as usinas de geração fotovoltaica, de modo que seja possível predizer se a utilização de um sistema complementar de geração será necessário ou não. Isso é de particular utilidade caso esse sistema complementar seja uma termoelétrica, por exemplo, já que em geral as mesmas demoram algumas horas para entrar em operação.

Para isso, é possível realizar análises estatísticas com base em dados meteorológicos e dados de geração solar, e desenvolver modelos de regressão com o intuito de prever a potência gerada.

1.3 Justificativa

A demanda por energia elétrica cresce cada vez mais no Brasil, sendo cada vez mais necessária a ampliação da capacidade de geração. No entanto, a preocupação com o meio ambiente faz com que exista grande dificuldade em explorar o potencial hidroelétrico ainda não utilizado no país (TIEPOLO *et al.*, 2018), e, pelo mesmo motivo, há um desincentivo ao uso de termoelétricas. Dessa forma, é de extrema importância a utilização de mais fontes de energia renovável, como a solar. Além disso, a crise hídrica vivida pelo país nos últimos anos evidencia a necessidade de se adotar meios alternativos de geração de energia, de modo a diversificar a matriz energética brasileira aumentando o volume de água nos reservatórios. Na Figura 2 é possível ver a comparação dos níveis de chuva na região oeste do Paraná:

Figura 2 – Comparação de precipitação mensal entre 1981-2010 e 2021.



Fonte: INPE (2022)

Na Figura 2, os *boxplots* referem-se à distribuição das médias mensais de chuva entre o ano de 1981 e 2010, sendo que a linha preta representa a média de chuva para cada mês. A linha vermelha, por sua vez, representa a média de chuva no ano de 2021.

Para possibilitar uma maior disseminação da energia fotovoltaica, é necessário que os fatores que influenciam na sua geração sejam determinados, de modo a solucionar em parte a incerteza causada pela sua intermitência. Portanto, o desenvolvimento de modelos capazes de prever a potência gerada é de extrema importância.

1.4 Objetivos

Comparar os modelos de regressão: *Support Vector Regression*, Árvores de Decisão, *Nearest Neighbors* e Rede Neural Artificial na previsão da potência gerada por um sistema fotovoltaico de acordo com dados como temperatura, umidade, pressão, ponto de orvalho, vento, irradiação e precipitação.

1.4.1 Objetivos específicos

- Tratar os dados (quanto à sua formatação, ausência e normalização) de potência de uma unidade consumidora do Grupo B (residencial) instalada na cidade de Foz do Iguaçu-PR; e de temperatura, umidade, pressão, ponto de orvalho, vento, irradiação e precipitação obtidos do Instituto Nacional de Meteorologia (INMET). Ambos os conjuntos correspondentes ao período entre 06/01/2019 e 05/01/2020, e entre 30/10/2021 e 29/07/2022.
- Realizar análise exploratória dos dados por meio de ferramentas estatísticas: média, coeficiente de variação, desvio padrão, correlação e gráficos de dispersão e realizar a seleção de variáveis.
- Validar diferentes algoritmos de regressão por meio da raiz quadrada do erro quadrático médio, coeficiente de determinação e erro médio absoluto percentual, e comparar as médias obtidas com os dados reais por meio do Teste de Dunnett com intervalo de confiança de 95%.

2 REFERENCIAL TEÓRICO

2.1 Fundamentos de Machine Learning

Machine Learning (ML) traduzido para o português como aprendizado de máquina e por vezes chamado de aprendizado estatístico, consiste em predizer dados de saída (valores-alvo) de acordo com dados de entrada. As técnicas de ML podem ser utilizadas para resolver problemas de classificação, onde os dados de saída são variáveis qualitativas, e problemas de regressão, nos quais os dados de saída são variáveis quantitativas, ou contínuas. No caso deste trabalho, como se quer prever valores de potência gerada, tem-se um problema de regressão (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Isso pode ser feito por meio de algoritmos que analisam dados de treino, encontrando relação entre as entradas e as saídas, abordagem essa que recebe o nome de aprendizagem supervisionada (JAMES *et al*., 2014).

A implementação de um algoritmo de aprendizagem supervisionada ocorre por meio de um conjunto de dados com entradas e saídas já conhecidas. Esse conjunto de dados é dividido entre conjunto de treino e de validação. O conjunto de treino é utilizado para que o algoritmo entenda a relação entre as entradas e saídas, gerando um modelo capaz de prever os valores-alvo.

Depois disso, os valores de entrada de validação são submetidos ao modelo, que prevê as saídas. As saídas de previsão são então comparadas com as saídas de validação, sendo possível analisar o quão bem o modelo consegue prever os resultados.

2.1.1 Seleção de variáveis

Em grande parte dos problemas de *Machine Learning*, existem muitos dados de entrada, sendo que alguns são redundantes ou irrelevantes. Esses dados atrapalham no processamento do modelo, tornando-o mais lento, e em alguns casos podem torna-lo menos preciso. Por esse motivo, é necessário realizar uma seleção dos dados de entrada (LI *et al*., 2017).

Uma das abordagens mais utilizadas para realizar a seleção de variáveis é a abordagem de filtro, onde deve ser especificada uma medida de avaliação, sendo que algumas das medidas de avaliação mais utilizadas são o coeficiente de correlação e o coeficiente de variação. O algoritmo então compara cada subconjunto de variáveis gerado com o subconjunto anterior, selecionando o melhor com base no critério de avaliação. O processo continua até que um critério de parada seja satisfeito. Essa abordagem de filtro é muito eficiente computacionalmente. No entanto, ela corre o risco de deixar de fora da seleção algumas variáveis que não são relevantes diretamente, mas sim quando combinadas com outras, e por isso é necessário ter cuidado ao descartar variáveis por terem um baixo nível de correlação (KUMAR; MINZ, 2014).

No entanto, segundo Daoud (2017), o processo de seleção de variáveis envolve ainda várias outras características, como dados históricos e análises empíricas, de modo que sua determinação é subjetiva.

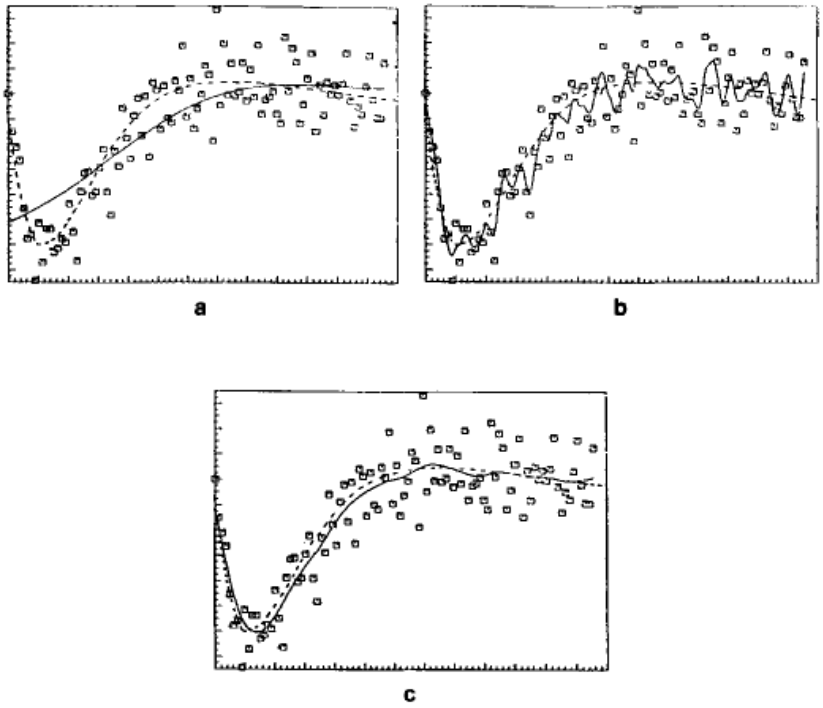
2.1.2 Balanço viés-variância

Em problemas de *Machine Learning*, a proporção entre os dados de treino e de validação, e o método utilizado influenciam na qualidade do modelo obtido.

Um modelo que se adéqua demasiadamente aos dados de treino, sofre de variância muito grande, perdendo a capacidade de generalizar. Diz-se então, que o modelo está sobre-ajustado. Caso o modelo não tenha aprendido suficientemente sobre os dados de treino, acabará tendo um alto viés. Diz-se então que o modelo está sub-ajustado. Portanto, para que um modelo de previsão seja bom, deve-se buscar um equilíbrio entre viés e variância (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Na Figura 3(a) é possível ver uma curva em sub-ajuste; (b) uma curva com sobre-ajuste; e (c) uma curva ideal:

Figura 3 – Exemplo de curvas com sub-ajuste, sobre-ajuste e ideal.



Fonte: Geman, Bienenstock e Doursat (1992)

Um meio de avaliar o nível de viés e de variância de um modelo é analisar os erros de treino e de validação para o modelo. Caso o erro de treino seja pequeno e o erro de validação grande, isso significa que o modelo está sobre-ajustado. Outra possibilidade é de que o erro de treino e de validação sejam grandes, o que indica um alto viés.

O balanço pode ser ajustado através da mudança da proporção entre dados de treino e de validação, bem como outras adequações específicas de cada algoritmo.

2.2 Análises estatísticas e tratamento de dados

A implementação de um modelo de ML depende de algumas análises estatísticas que devem ser feitas previamente para que seja possível entender melhor o conjunto de dados.

2.2.1 Tendências centrais

Durante a implementação de um modelo de ML, é geralmente importante que se tenha uma noção de qual é a tendência central dos dados. Para isso, pode-se utilizar a média, que é a soma dos dados dividido pela sua quantidade. No entanto, a média é muito sensível à valores discrepantes que possam existir no conjunto de dados. Por esse motivo, em alguns casos a métrica de maior interesse é a mediana, que é o valor central da amostra. Além disso, também pode-se utilizar o quantil, onde a análise é feita de acordo com o valor abaixo no qual estão uma certa porcentagem do número total de dados (GRUS, 2021).

2.2.2 Dispersão

Descrever um conjunto de dados por meio de sua tendência central, faz com que não se tenha conhecimento acerca de sua variabilidade. Portanto, é necessário adotar métricas que quantifiquem o quão dispersos da média os dados estão. Para isso, costuma-se usar o desvio padrão, que indica qual é o erro médio cometido ao tentar definir cada elemento do conjunto de dados como a média (MORETTIN, BUSSAB, 2017).

2.2.3 Associação entre variáveis

O Coeficiente de Correlação de Pearson é uma métrica que quantifica, por meio de um único número, o grau de associação linear entre duas variáveis, variando geralmente entre -1 e 1, onde 1 indica uma correlação direta perfeita e -1, uma correlação inversa perfeita. Quanto mais próximo de 0, menor a correlação (MORETTIN, BUSSAB, 2017). A interpretação de cada valor de correlação pode ser vista no Quadro 1:

Quadro 1 – Interpretação dos valores de correlação.

Coeficiente	Intepretação
0,90 a 1,00	Muito Alta
0,70 a 0,89	Alta
0,50 a 0,69	Moderada
0,30 a 0,39	Baixa
0,00 a 0,29	Muito Baixa

Fonte: Asuero, Sayago e Gonzalez (2006)

Além disso, outra ferramenta útil são os gráficos de dispersão, que permitem avaliar visualmente a associação entre duas variáveis.

2.3 Métodos de regressão

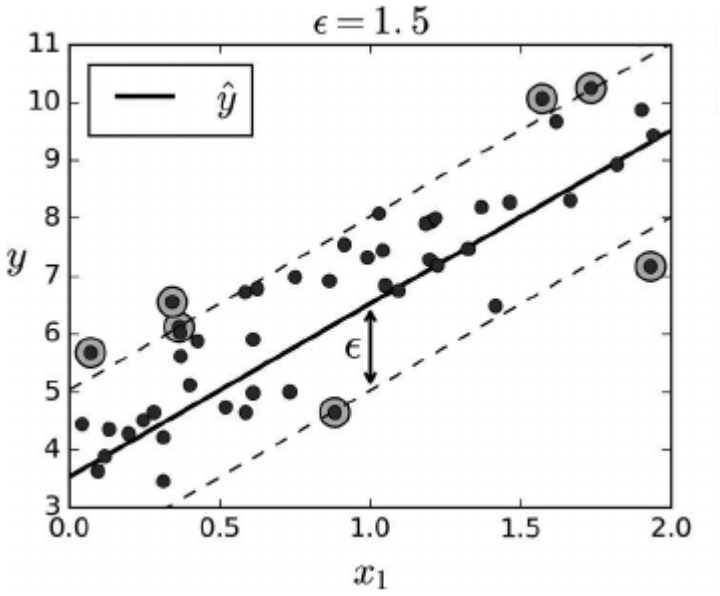
Para prever os valores de potência gerada, são utilizados quatro algoritmos de regressão, sendo eles: *Support Vector Regression*, Árvores de Decisão, *Nearest Neighbors* e Rede Neural Artificial.

2.3.1 Support Vector Machines

Support Vector Machines (SVM) é um algoritmo desenvolvido por Cortes e Vapnik em 1995, e é utilizado para resolver problemas de classificação e de regressão (AWAD; KHANNA, 2015).

O algoritmo SVM utilizado para regressão é chamado de *Support Vector Regression* (SVR). Seu funcionamento consiste em traçar duas margens que englobem toda ou maior parte dos dados, de modo que a curva de regressão fica no ponto médio entre as duas curvas, como pode ser visto na Figura 4:

Figura 4 – Representação gráfica do algoritmo SVR.



Fonte: Geron (2019)

Também é necessário realizar a distinção entre os conceitos de margem rígida e margem flexível. Em um modelo de margem rígida, as margens devem obrigatoriamente englobar todos os dados, fazendo com que o modelo seja extremamente sensível a pontos discrepantes.

Portanto, em alguns casos é interessante que o modelo seja mais flexível, permitindo que alguns dados fiquem de fora das margens (GERON, 2019).

Nesses casos, a distância entre as margens e a linha de regressão é determinada pelo fator **ε**, conforme visto na Figura 4. Os pontos que violam as margens (destacados na Figura 4) sofrem uma penalização proporcional à sua distância até a margem. A soma de todas as penalizações não pode exceder o limite de tolerância C, que deve ser previamente escolhido. Um valor de C muito alto faz com que o algoritmo tolere um alto valor de erro, resultando em um alto viés e em um modelo sub-ajustado. Um valor de C pequeno no entanto, tem efeito oposto, aumentando a variância e causando sobre-ajuste como consequência (SMOLA; SCHOLKOPF, 2004).

Para que sejam obtidas curvas de regressão não lineares, é necessário utilizar uma função *kernel*, que serve para aplicar uma solução linear a um problema não-linear (THEODORIDIS; KOUTROUMBAS, 2008).

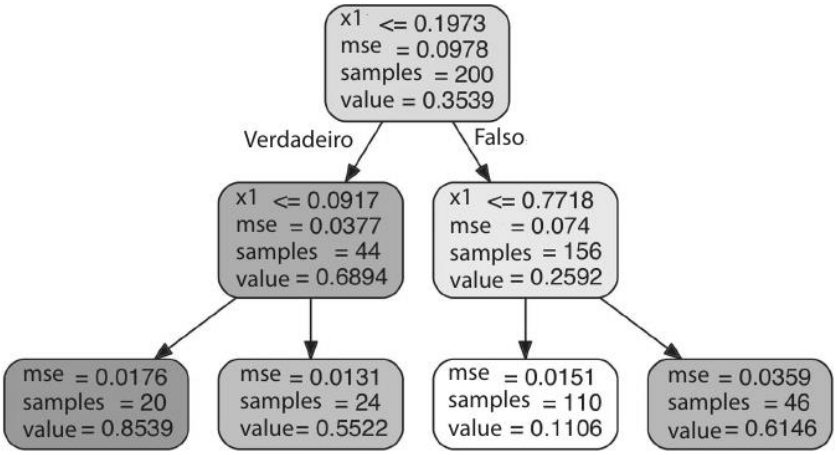
2.3.2 Árvores de decisão

O algoritmo de Árvore de Decisão, como nome sugere, reproduz uma árvore dividida por camadas ou nós, que iniciam na raiz e terminam nas folhas. A quantidade de nós entre a raiz e as folhas depende da profundidade escolhida.

O algoritmo consiste em analisar os dados de treino e separa-los em regiões diferentes em um plano bidimensional, com fronteiras que são sempre ortogonais. A quantidade de regiões corresponde à quantidade de camadas existentes. Dessa forma, o modelo analisa os dados e os coloca em determinada região de acordo com alguma métrica de erro (GERON, 2019).

O exemplo da estrutura de uma árvore de decisão pode ser visto na Figura 5, onde "x1" representa um dado de entrada de treino, "*samples*" representa a quantidade de amostras existentes na região (nó) específico, "mse" é o erro quadrático médio da previsão sobre os valores da amostra e "*value*" é o valor que está sendo previsto pelo algoritmo.

Figura 5 –Exemplo de estrutura de uma árvore de decisão.



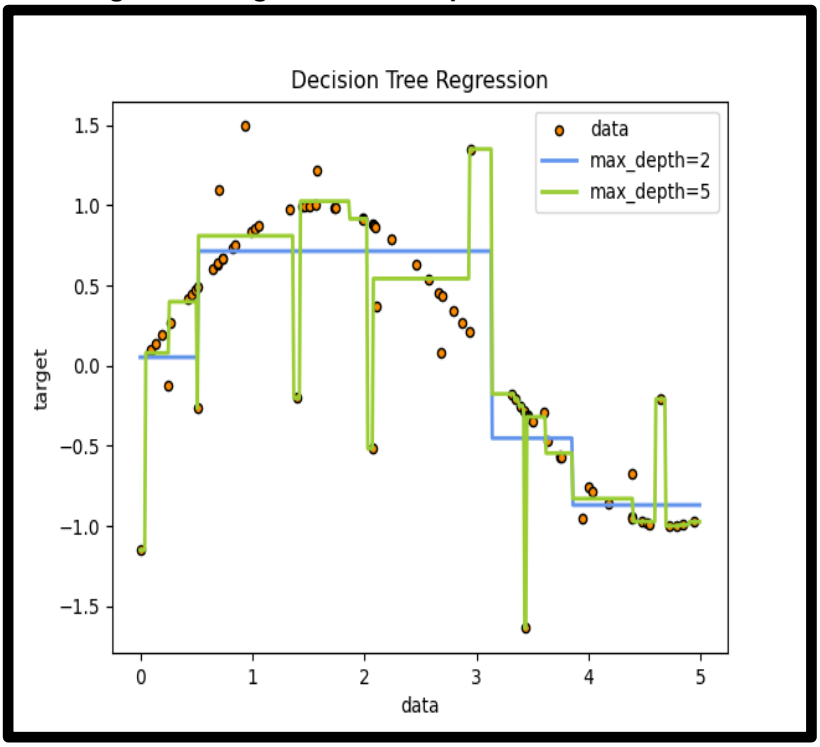
Fonte: Geron (2019)

Portanto, cada nó representa uma decisão a ser tomada pelo algoritmo de acordo com as características dos dados de entrada, de modo que o número de folhas aumenta exponencialmente junto ao número de camadas. Dessa forma, a profundidade escolhida influencia diretamente na performance do modelo, pois a existência de um número excessivo de camadas implica em uma variância muito

grande nos dados de saída, dificultando a generalização e resultando em sobre-ajuste. Em contrapartida, um número insuficiente de camadas resulta em poucas possibilidades de dados de saída, causando sub-ajuste (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

O exemplo de curvas de regressão obtidas por uma Árvore de Decisão, considerando dois valores de profundidade máxima podem ser vistas na Figura 6:

Figura 6 – Regressão obtida por árvore de decisão.



Fonte: SCIKIT-LEARN (2022)

Na Figura 6 é possível ver que na curva azul, onde utiliza-se uma profundidade igual a 2, a resposta é mais ajustada aos dados. Já adotando uma profundidade de 5, representado pela curva verde, modelo fica sobre-ajustado, sendo demasiadamente afetado pelos *outliers*.

2.3.3 Nearest Neighbors

O algoritmo *Nearest Neighbors* (vizinhos próximos) é utilizado para problemas de aprendizagem não-supervisionada, classificação e regressão.

Seu princípio de funcionamento é encontrar um determinado número de pontos de treino com menor distância do ponto de teste, estimando seu valor a partir deles (GERON, 2019).

Em problemas de regressão, o valor de saída para um determinado conjunto de dados é calculado com base na média dos valores de saída de seus vizinhos mais próximos. O número de vizinhos mais próximos pode ser um valor constante e inteiro, ou variável, calculado com base em um raio, que também deve ser pré-determinado (SCIKIT-LEARN, 2022).

Já a distância pode ser calculada por várias métricas, dentre as quais a de maior interesse é a Distância de Minkowski, dada pela fórmula vista na Equação 1:

$$D(X,Y) = (\sum_{i=1}^{n} |X_i - Y_i|)^{\frac{1}{p}} \quad (1)$$

Sendo que para p=1, a distância é equivalente à Distância de Manhattan, e para p=2 é equivalente à Distância Euclidiana.

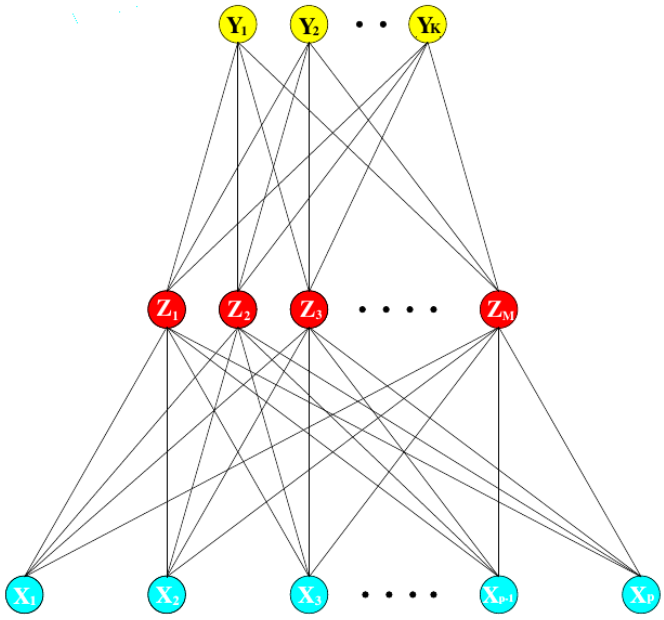
2.3.4 Rede Neural Artificial (RNA)

Uma Rede Neural Artificial é um modelo preditivo inspirado no funcionamento do cérebro humano. Um neurônio recebe uma entrada, realiza um cálculo e, a depender do resultado, dispara ou não (GRUS, 2021).

Em 1957, Frank Rosenblatt criou o *Perceptron*, que é uma das mais simples arquiteturas de RNA, sendo equivalente a um único neurônio com entradas binárias. Ele realiza uma soma ponderada dessas entradas, e caso o resultado seja maior ou igual a zero, sua saída é ativada.

Anos depois, foram criadas as Redes Perceptron Multicamadas (MLP), que possuem várias camadas entre a entrada e a saída, que são chamadas de camadas ocultas. Dessa forma, os neurônios da próxima camada recebem como entrada as saídas do neurônio anterior. Esse modelo recebe o nome de *feed-forward*. A ilustração de uma rede neural MLP pode ser vista na Figura 7:

Figura 7 – Rede neural MLP.



Fonte: Grus (2021)

Na Figura 7 é possível observar as entradas do algoritmo X1, X2,...,Xn), as camadas ocultas (Z1, Z2,...,Zn) e as saídas (Y1, Y2,...,Yn).

Uma RNA funciona atribuindo pesos para cada um dos neurônios. Esse peso é multiplicado pelo valor de entrada, sendo submetido à uma função de ativação, que determina se o neurônio em questão será ativado ou não.

O método de treinamento de MLP chamado de \textit{backpropagation}, funciona apresentando cada valor de treino para a rede, que calcula os valores de saída dos neurônios em cada uma das camadas até a saída. Depois disso, o valor de saída é comparado com o valor-alvo, obtendo assim o erro. É calculado o quanto cada neurônio contribuiu para o valor do erro, fazendo isso para todas as camadas predecessoras até chegar na camada inicial, ajustando os parâmetros de cada neurônio de modo a minimizar o erro. Esse processo ocorre a partir de uma função de otimização, que nesse contexto recebe o nome de solucionador. O processo continua até que o erro seja minimizado até um valor aceitável pré-determinado (GERON, 2019).

É interessante pontuar que as RNA são extremamente sensíveis à distribuição dos dados, sendo de extrema importância que esses sejam

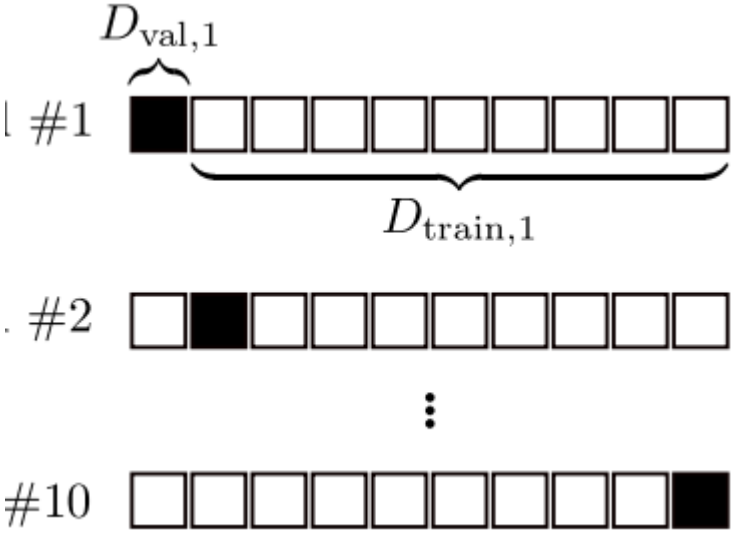
normalizados antes do treino do algoritmo (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

2.4 Validação dos Modelos

2.4.1 Validação cruzada

Para garantir que o modelo desenvolvido tem um bom desempenho independentemente do conjunto de dados, é indicado que o mesmo seja validado com dados diferentes. Uma forma de se fazer isso, é utilizando o processo de validação cruzada, que divide o conjunto de dados em partes iguais, revezando quais serão usadas para treino e validação. O processo de uma validação cruzada pode ser visto na Figura 8:

Figura 8 – Representação gráfica de uma validação cruzada.



Fonte: Berrar (2019)

2.4.2 Raiz quadrada do erro quadrático médio (RMSE)

Uma das medidas mais comuns de serem utilizadas para verificar a acurácia de modelos é o Erro Quadrático Médio (MSE), que é dado pela Equação 2:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

(2)

O MSE compara cada um dos valores previstos $\hat{Y}_i$ com os valores reais Y_i, eleva ao quadrado e divide pelo número de amostras (n).

No entanto, a métrica mais utilizada é a Raiz Quadrada do Erro Quadrático Médio (RMSE), que eleva o MSE ao quadrado, permitindo comparar o valor do erro na mesma dimensão que a variável analisada (JAMES *et al.,* 2014).

2.4.3 Erro médio absoluto percentual (MAPE)

Outra métrica importante é o *Mean Absolute Percentage Error* (MAPE), que permite calcular o erro em porcentagem (MYTTENAERE *et al*., 2016), sendo dado pela Equação 3, onde $\hat{Y}_i$ representa o valor previsto, Y_i, o valor real e n o número de amostras:

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{Y}_i - Y_i}{\hat{Y}_i}\right|$$

(3)

2.4.4 Coeficiente de determinação (R^2)

O coeficiente de determinação busca relacionar duas variáveis por meio de uma equação linear. Ela indica a proporção da variância da variável dependente que pode ser explicada por uma variável independente. Quanto mais próximo de 1, maior a relação entre as duas variáveis.

2.4.5 Teste de Dunnett

O teste de Dunnett é um teste utilizado para comparar médias de testes com uma média de controle, com o objetivo de aferir se há diferença significativa entre elas. O valor da diferença mínima significativa (DMS) é dado pela Equação 4:

$$DMS = t_{Dunnett}\sqrt[2]{\frac{2QM}{n}} \quad (4)$$

Onde o valor de $t_{Dunnett}$ é dado pela Tabela de Dunnett a partir do número de amostras, número de tratamentos e do intervalo de confiança. O valor de n é o número de amostras. Por fim, o termo QM corresponde ao quadrado médio dos resíduos, que é obtido por meio da Análise de Variância (ANOVA).

3 PROCEDIMENTOS METODOLÓGICOS

A pesquisa a ser realizada é do tipo descritiva, onde os dados são obtidos em campo, com caráter experimental.

Os algoritmos serão desenvolvidos em linguagem Python, utilizando as bibliotecas pandas e numpy para manipulação dos dados, matplotlib e seaborn para visualização e scikit-learn para implementação e validação dos modelos de *Machine Learning*.

3.1 Obtenção dos dados

Os dados meteorológicos (temperatura, umidade, pressão, vento, radiação e pluviosidade) são obtidos da base de dados públicos do Instituto Nacional de Meteorologia (INMET), de uma estação meteorológica localizada na cidade de Foz do Iguaçu-PR. Os dados são salvos no formato *comma separated values* (CSV).

Já os dados de geração fotovoltaica são obtidos de uma unidade consumidora do Grupo B (grupo tarifário das unidades consumidoras que possuem tensão de fornecimento inferior a 2,3kV), também localizada em Foz do Iguaçu, com uma potência instalada de 3,58 kWp e Inversor da marca Canadian. Esses dados são salvos em uma planilha no formato *Excel Spreadsheet* (XLSX).

Ambos conjuntos de dados obtidos correspondem aos períodos de 06/01/2019 à 05/01/2020, e 30/10/2021 à 29/07/2022.

3.2 Tratamento dos dados

Os dados coletados devem ser tratados, visto que esses não são obtidos na forma ideal para serem utilizados.

Primeiramente, os dados de geração são fornecidos de 5 em 5 minutos, enquanto os dados meteorológicos são fornecidos de hora em hora. Porém, o conjunto de dados de geração e o dos dados meteorológicos devem possuir o mesmo número de amostras. Por esse motivo, deve-se realizar uma re-amostragem nos dados de geração. Para isso, a média de potência para cada hora é calculada.

Enquanto as medições para os dados meteorológicos são dados no Tempo Universal Coordenado (UTC), os dados de potência são dados no Horário de Brasília (UTC-3). Dessa forma, é necessário ajustar o horário dos dados meteorológicos, deixando-os de acordo com o horário local. Nesse processo, alguns dados residuais (fora do intervalo de tempo determinado) são obtidos, e estes devem ser excluídos.

Durante a noite, a geração é nula devido à ausência de radiação solar. Dessa forma, caso sejam considerados os horários em que não existe geração, há a chance de que as outras variáveis tenham seu efeito subestimado pelo modelo. Por esse motivo, são considerados os dados meteorológicos apenas nos horários em que há geração. Depois disso, os dois conjuntos de dados possuem o mesmo número de amostras, então são mesclados formando um único *dataframe*.

No entanto, existem dados faltantes no conjunto de dados meteorológicos. Por isso, as amostras que possuem valores faltantes precisam ser descartadas ou preenchidas. Toma-se um cuidado especial com os dados de radiação solar, pois esses exercem grande influência sobre o modelo. Por esse motivo, opta-se por descartar as amostras com dados faltantes de radiação solar, por meio da função pandas dropna. Para os dados faltantes de chuva, o valor é substituído pela moda, já que existe variância muito grande nas observações. Para os demais valores realiza-se o preenchimento por meio da função pandas fillna, sendo que o valor é preenchido com o valor da observação anterior.

3.3 Análise exploratória e seleção de variáveis

Após o tratamento dos dados, é possível realizar uma análise exploratória com o objetivo de resumir as características principais do conjunto, identificando tendências e padrões. Esse processo se inicia observando valores como a contagem total de dados, a média, desvio padrão, entre outros. Dessa forma, é possível obter uma breve descrição acerca do conjunto de dados que será utilizado, e formular hipóteses acerca da solução do problema a ser resolvido.

O coeficiente de correlação de Pearson, que mede a relação linear entre cada uma das variáveis de entrada com a variável de saída é obtido por meio da biblioteca pandas. Também são plotados gráficos de dispersão utilizando as

bibliotecas matplotlib e seaborn relacionando cada uma das variáveis, permitindo uma análise visual da correlação. Com base nessas duas análises, as variáveis importantes para o modelo são definidas.

3.4 Aplicação dos algoritmos

Para realizar a aplicação dos algoritmos, será utilizada a biblioteca scikit-learn da linguagem Python.

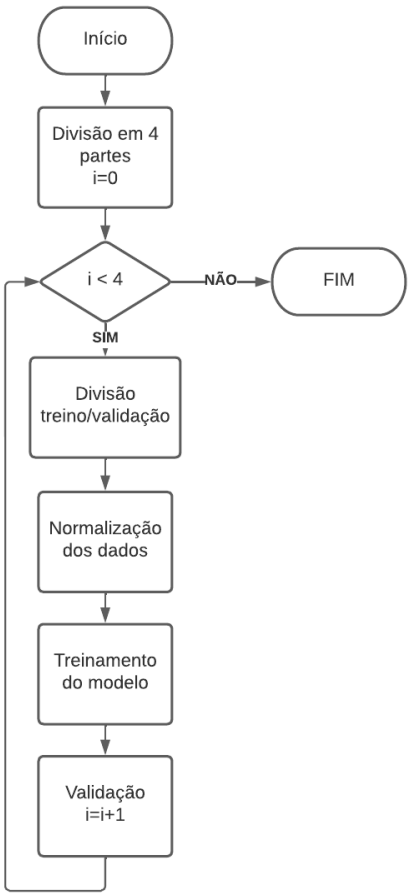
Primeiramente, a função train_test_split do scikit-learn é utilizada para separar aleatoriamente 10% dos dados para posterior teste dos modelos.

Cada um dos algoritmos utilizados possuem hiper-parâmetros que podem ser variados com o objetivo de adequar melhor o modelo ao problema, melhorando sua performance e diminuindo o erro na previsão. Para realizar a escolha dos hiper-parâmetros, utiliza-se Validação Cruzada com a função cross_val_score, utilizando o RMSE como métrica de erro. Um dos parâmetros dessa função é o modelo a ser utilizado. Esse modelo é montado com o auxílio da função Pipeline, que permite criar uma sequência de transformações que serão aplicadas aos dados depois da separação entre treino e validação, e antes do treinamento do modelo. No caso dos algoritmos SVR, KNR e RNA, utiliza-se como transformação a função StandardScaler, que normaliza os dados. Como a Árvore de Decisão não precisa de dados normalizados, a utilização da função Pipeline não é necessária.

A validação dos modelos é feita utilizando a função KFold, que divide o conjunto de dados em K partes iguais. Nesse caso, utiliza-se K=4. Depois disso, a função cross_val_score usa três partes para treino e 1 parte para validação, alternando as parcelas em cada iteração. Dessa forma, o algoritmo será treinado e validado 4 vezes, obtendo ao final a média dos valores de RMSE obtidos em cada uma das iterações.

O fluxograma do algoritmo de Validação Cruzada pode ser vista na Figura 9:

Figura 9 – Algoritmo de validação cruzada.



Fonte: Autoria própria (2022)

Importante notar que para o algoritmo de Árvore de Decisão, a etapa de normalização dos dados é pulada.

Dessa forma, os hiper-parâmetros dos modelos são alterados de forma a analisar quais são os valores ótimos. Cada um dos hiper-parâmetros que devem ser escolhidos para implementação dos algoritmos propostos podem ser vistos a seguir.

3.4.1 Support Vector Regression (SVR)

Para a utilização do algoritmo SVR na biblioteca scikit-learn é necessário determinar o tipo de *kernel* utilizado, sendo que nesse caso, opta-se pelo *kernel* RBF. Para o nível "C" de tolerância à *outliers* e a distância **ε** devem ser testados diferentes valores de modo a maximizar o desempenho do modelo.

3.4.2 Árvores de decisão

No algoritmo de Árvores de Decisão aplicado, é necessário determinar o critério adotado para cálculo do erro, que no caso será o RMSE. Além disso, é possível determinar máxima profundidade da árvore. Caso a profundidade não seja informada ao algoritmo, a mesma é determinada automaticamente com base no número mínimo de amostras possíveis para um nó e para uma folha.

Dessa forma, a abordagem adotada será primeiramente determinar o número mínimo de amostras para uma folha. Com base no valor do erro obtido na validação, o número de amostras é ajustado.

3.4.3 Nearest Neighbors

Na biblioteca scikit-learn há a possibilidade de utilizar o algoritmo *K Neighbors Regressor* (KNR), onde o número de vizinhos é uma constante K. O método utilizado para calcular a distância entre os pontos é a Distância de Minkowski. Nesse método é necessário determinar o parâmetro "p", onde p=1 equivale à distância de Manhattan, p=2 equivale à distância euclidiana. Nesse caso, o valor de p será variado, influenciando nos pontos que serão considerados mais próximos.

Além disso, também é possível determinar a influência que cada vizinho terá por meio do hiper-parâmetro *weights*. Quando *weights*='*distance*', a influência do vizinho é inversamente proporcional à distância. Já quando *weights*='*uniform*', a influência de cada vizinho independe da distância.

3.4.4 Rede neural artificial

Será utilizada uma rede neural *multi-layer perceptron*, com retro-propagação. O número de camadas ocultas será variado de 5 a 25, com incremento de 5. Como função de ativação serão testadas a função sigmoide, tangente hiperbólica e a *rectified linear unit* (Relu). O solucionador utilizado será o Adam (*adaptive moment estimation*), que possui os parâmetros $\boldsymbol{\alpha}$, $\boldsymbol{\beta}_1$, $\boldsymbol{\beta}_2$ e $\boldsymbol{\varepsilon}$. Segundo Kingma e Ba (2014),

os valores padrão adequados para essas variáveis são: $\boldsymbol{\alpha}$=0,001; $\boldsymbol{\beta}_1$=0,9; $\boldsymbol{\beta}_2$=0,999; $\boldsymbol{\varepsilon}$=10^{-8}.

A tolerância será de 10^{-4}, e o algoritmo chega ao final caso esse valor de erro permaneça por 5 iterações seguidas. O número máximo de iterações, independentemente do erro, será de 200.

3.5 Teste dos modelos

Selecionados os melhores hiper-parâmetros, o modelo é aplicado ao conjunto de teste, prevendo seus valores de saída. Os valores de saída previstos são comparados com os valores reais por meio do RMSE, MAPE e R^2, sendo possível assim determinar qual algoritmo conseguiu prever melhor os dados reais.

Depois disso, as médias dos resultados obtidos são comparadas com o valor real por meio do teste de Dunnett, com o objetivo de determinar se há ou não diferenças significativas entre as médias obtidas pelos modelos e a média das variáveis alvo. Para isso, o intervalo de confiança adotado será de 95% ($\boldsymbol{\alpha}$ = 0,05). Dessa forma, o valor do t de Dunnett é obtido pela Tabela de Dunnett de acordo com o número de tratamentos (5) e o número de amostras do conjunto de teste (606).

4 RESULTADOS E DISCUSSÃO

4.1 Obtenção e tratamento dos dados

Os dados meteorológicos foram obtidos da base de dados do INMET. Como os dados estão fora do fuso-horário brasileiro, estes foram convertidos. Depois disso, os dados residuais foram excluídos. Os dados de potência foram obtidos, e integralizados para o intervalo de uma hora, adicionando os horários faltantes no conjunto original. Os dados meteorológicos e os dados de potência foram mesclados em uma única planilha, obtendo-se assim um conjunto de dados com 15312 amostras.

Depois disso, inicia-se o processo de tratamento dos dados. Primeiramente, são excluídas do conjunto todas as observações onde a potência ou a radiação solar são nulas ou faltantes, conforme os critérios estabelecidos previamente.

Para o tratamento dos dados faltantes, primeiramente a proporção de dados faltantes em cada variável foi obtida, de acordo com a Figura 10:

Figura 10 – Porcentagem de dados faltantes.

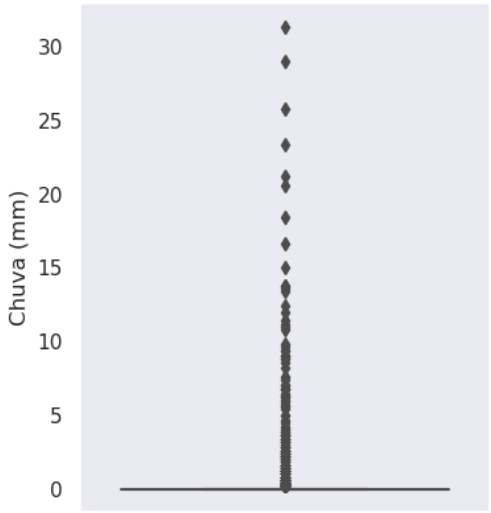
```
Data e Hora             0.000000
Hora                    0.000000
Mês                     0.000000
Temp. Ins. (C)          0.000000
Temp. Max. (C)          0.330033
Temp. Min. (C)          0.330033
Umi. Ins. (%)           0.000000
Umi. Max. (%)           0.330033
Umi. Min. (%)           0.330033
Pto Orvalho Ins. (C)    0.000000
Pto Orvalho Max. (C)    0.330033
Pto Orvalho Min. (C)    0.330033
Pressao Ins. (hPa)      0.000000
Pressao Max. (hPa)      0.330033
Pressao Min. (hPa)      0.330033
Vel. Vento (m/s)        0.066007
Dir. Vento (m/s)        0.066007
Raj. Vento (m/s)        0.429043
Radiacao (KJ/m²)        0.000000
Chuva (mm)              1.732673
Potência CA (W)         0.000000
dtype: float64
```

Fonte: Autoria própria (2022)

Para as variáveis com proporção faltante de menos de 1%, a estratégia adotada para o preenchimento foi substituir o valor ausente pelo valor da observação anterior.

Já para os valores de Chuva, cujo percentual foi de 1,73%, obtém-se um *boxplot* para analisar sua distribuição, conforme visto na Figura 11:

Figura 11 – Distribuição dos valores de chuva.



Fonte: Autoria própria (2022)

Analisando a Figura 11 é possível notar que todos os valores diferentes de 0 são considerados pontos discrepantes, visto que representam uma pequena proporção da amostra. Por esse motivo, o valor utilizado para o preenchimento foi o valor da moda.

Dessa forma, ao final do processo, tem-se um conjunto de dados com 6060 amostras e sem dados ausentes.

4.2 Seleção de variáveis

Para selecionar as variáveis que serão utilizadas no modelo, primeiramente, obtém-se o Coeficiente de Correlação de Pearson, que pode ser visto na Figura 12:

Figura 12 – Coeficiente de correlação de Pearson.

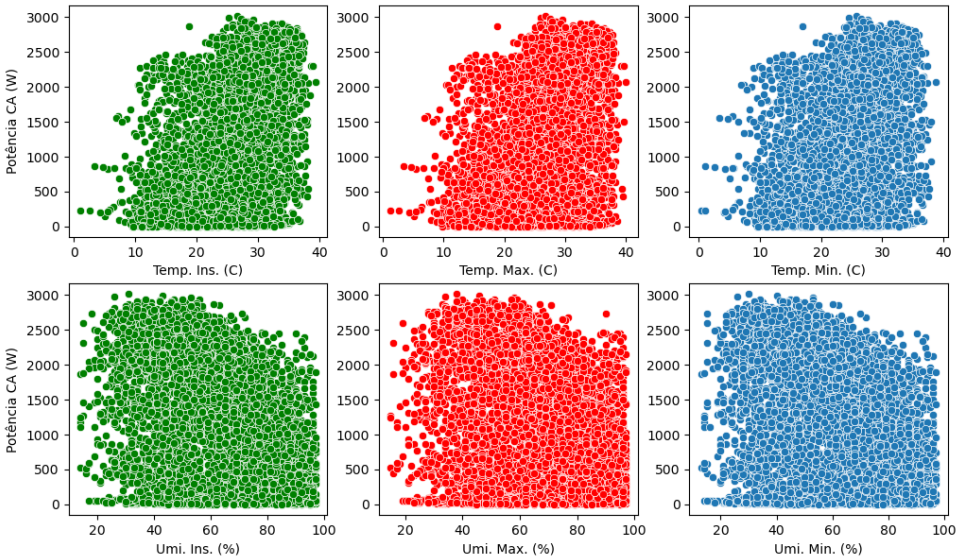
```
Potência CA (W)          1.000000
Radiacao (KJ/m²)         0.819057
Temp. Ins. (C)           0.438668
Temp. Max. (C)           0.397042
Temp. Min. (C)           0.368587
Raj. Vento (m/s)         0.257354
Vel. Vento (m/s)         0.204788
Pressao Max. (hPa)       0.097188
Pressao Min. (hPa)       0.091788
Pto Orvalho Max. (C)     0.083306
Pressao Ins. (hPa)       0.053424
Pto Orvalho Ins. (C)     0.016795
Pto Orvalho Min. (C)    -0.002369
Hora                    -0.010074
Mês                     -0.022528
Chuva (mm)              -0.109281
Dir. Vento (m/s)        -0.145307
Umi. Max. (%)           -0.384140
Umi. Min. (%)           -0.430421
Umi. Ins. (%)           -0.466839
Name: Potência CA (W), dtype: float64
```

Fonte: Autoria própria (2022)

É possível perceber que, como esperado, a Radiação tem um alto nível de correlação com a Potência. Além disso, também é possível destacar os valores de Temperatura instantânea, máxima e mínima, e os valores de umidade instantânea, mínima e máxima.

Além disso, é necessário obter gráficos de dispersão relacionando cada uma das variáveis com a potência, para que seja possível analisar visualmente a correlação, especialmente as que não são lineares. Todos os gráficos de dispersão podem ser vistos nas Figuras 13, 14 e 15:

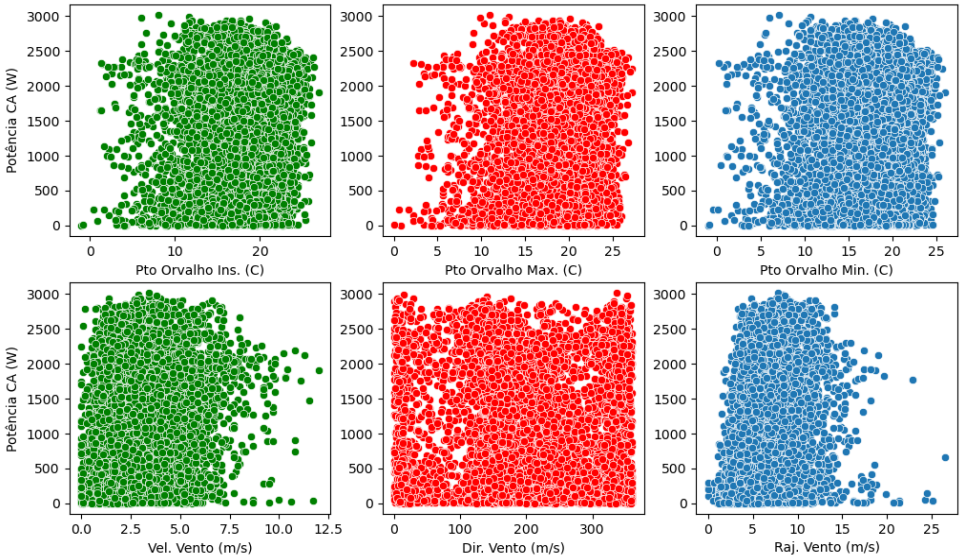
Figura 13 – Gráficos de dispersão: Temperatura e umidade.



Fonte: Autoria própria (2022)

Analisando a Figura 13, é possível visualizar claramente a existência de correlação linear entre Temperatura e Umidade com a Potência gerada, confirmando o que foi observado no coeficiente de correlação de Pearson.

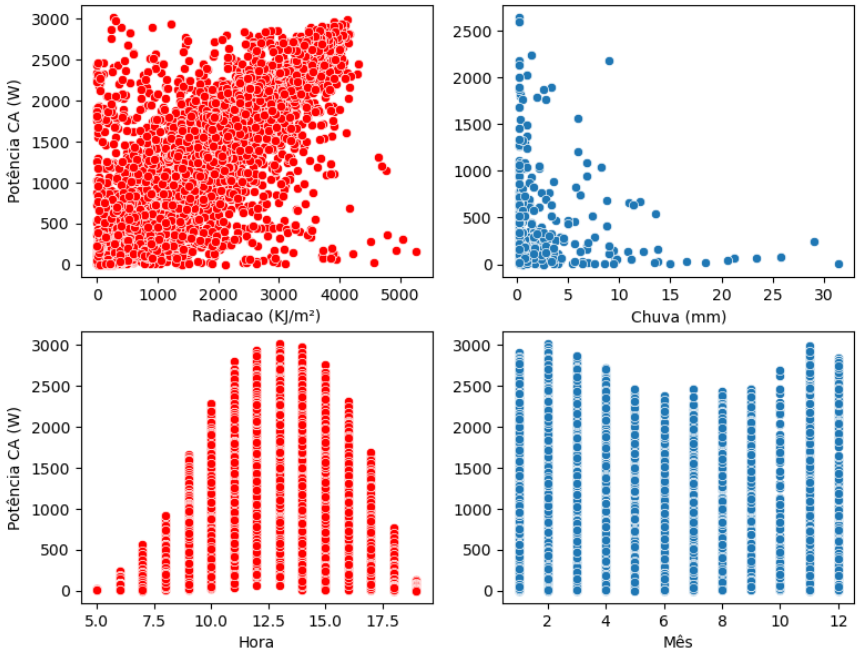
Figura 14 – Gráficos de dispersão: Ponto de orvalho e vento.



Fonte: Autoria própria (2022)

Na Figura 14, nota-se que há uma leve correlação entre o Ponto de Orvalho e a Potência. No que diz respeito aos dados relacionados ao vento, não é observada nenhuma correlação.

Figura 15 – Gráficos de dispersão: Radiação, chuva, hora e mês.



Fonte: Autoria própria (2022)

A Figura 15 torna claro o alto nível de correlação existente entre Radiação e Potência. Nota-se também que a Hora e o Mês possuem uma relação com a variável alvo, sendo que a mesma sofre variação de acordo com o horário e com a época do ano. Em relação à chuva, nota-se uma relação exponencial negativa, onde valores muito altos de chuva causam valores pequenos de potência, e altos valores de potência gerada ocorrem apenas com um baixo índice de chuva.

Dessa forma, determina-se que as seguintes variáveis serão utilizadas no modelo: Radiação, Temperatura instantânea, Temperatura máxima, Temperatura mínima, Umidade instantânea, Umidade máxima, Umidade mínima, Ponto de orvalho instantânea, Ponto de orvalho máximo, Ponto de orvalho mínimo, Chuva, Hora e Mês.

Os resultados são coerentes com os obtidos por (ZHU *et al.*, 2015), FRANCISCO *et al.*, 2019) e (OLIVEIRA; BELCHIOR, 2021) embora estes dois últimos tenham observado uma maior influência da velocidade do vento nos dados de potência, selecionando essa variável para seus modelos.

4.3 Treinamento dos algoritmos e seleção de parâmetros

Primeiramente, foi realizada a separação de 10% dos dados para ser utilizado no teste. Depois disso, realizou-se a Validação Cruzada para cada um dos algoritmos com o objetivo de ajustar seus hiper-parâmetros.

4.3.1 Support Vector Regression (SVR)

Para o algoritmo SVR, utilizou-se o *kernel* RBF, pois em testes preliminares observou-se ser o mais adequado e com menor erro. Além disso, por conta da formulação do algoritmo, existe a possibilidade de ele prever valores negativos. Para contornar esse problema aplica-se uma transformação logarítmica nos dados de potência. Os valores de erro obtido em cada uma das tentativas com seus hiper-parâmetros podem ser vistos na Tabela 1:

Tabela 1 – Validação do algoritmo SVR

ID	Parâmetros		Validação
	C	epsilon	RMSE
1	1	0,1	-0,65
2	1	0,01	-0,66
3	5	0,1	-0,61
4	10	0,1	-0,593
5	20	0,1	-0,58
6	**40**	**0,1**	**-0,5793**
7	40	1	-0,71

Fonte: Autoria própria (2022)

Na Tabela 1 é possível observar que o menor valor de RMSE é atingido quando C=40 e epsilon = 0,1, obtendo um RMSE de -0,5793. É importante notar que o valor de RMSE obtido não está em Watts, por conta da transformada logarítmica aplicada no potência. Devido a uma particularidade do algoritmo de validação cruzada, não é possível fazer a transformada inversa nos dados de saída.

4.3.2 Árvore de decisão (DT)

Para a Árvore de Decisão, o valor do número mínimo de amostras por folha, representado pela variável min_sample_leaf, foi variado com o objetivo de minimizar o valor do RMSE obtido. Os valores de erro obtido em cada uma das tentativas com seus hiper-parâmetros podem ser vistos na Tabela 2:

Tabela 2 – Validação do algoritmo DT

ID	Parâmetros	Validação
	min_samples_leaf	**RMSE**
1	30	378,72
2	10	377,85
3	5	395,86
4	20	370,28
5	**16**	**366,57**

Fonte: Autoria própria (2022)

Pode-se observar na Tabela 2 que o número de ótimo de amostras por folha é de 16, obtendo um RMSE de 366,57W. Um número maior de amostras por folha faz com que o modelo não aprenda suficientemente sobre os dados. Já um número menor, faz com que o modelo perca a capacidade de generalizar.

4.3.3 Nearest Neighbors (KNR)

Para o *K Neighbors Regressor*, o parâmetro *weights* utilizado foi o '*distance*'. Além disso, o número de vizinhos próximos K e o fator da distância de Minkolski p foi variado, obtendo os resultados vistos na Tabela 3:

Tabela 3 – Validação do algoritmo KNR

ID	Parâmetros		Validação
	n_neighbors	p	RMSE
1	3	1	364,9
2	3	2	360,55
3	3	3	362,43
4	2	2	380,73
5	4	2	355,28
6	10	2	346,37
7	**9**	**2**	**345,58**

Fonte: Autoria própria (2022)

Dessa forma, por meio da Tabela 3, nota-se que o número ótimo de vizinhos próximos é 9, e o do fator da Distância de Minkoski é p=2, correspondendo à distância euclidiana. Com esses valores, obteve-se um RMSE de 345,58W.

4.3.4 Rede neural artificial (RNA)

Para a Rede Neural Artificial, o número de camadas ocultas, bem como o número de neurônios em cada camada foi variado pelo parâmetro *layer*. As funções de ativação relu e logística se mostraram mais promissoras do que a sigmoide e a tangente hiperbólica por apresentar um menor erro, e por isso essa última foi excluída. Como *solver* foram testados o adam com os parâmetros recomendados por Kingma e Ba (2014), e o lbfgs. Em relação ao *learning rate*, não foi observada mudança no erro ao variar seu valor. O resultados obtidos podem ser vistos na Tabela 4:

Tabela 4 – Validação do algoritmo RNA

ID	Parâmetros			Validação
	layers	**activation**	**solver**	**RMSE**
1	7x100	relu	lbfgs	322,95
2	7x100	relu	adam	324,43
3	5x100	relu	lbfgs	317,06
4	5x100	logistic	lbfgs	558,96
5	3x100	relu	lbfgs	319,98
6	4x100	relu	lbfgs	317,48
7	**5x50**	**relu**	**lbfgs**	**313,48**
8	5x25	relu	lbfgs	322,61
9	5x75	relu	lbfgs	318,34

Fonte: Autoria própria (2022)

A Tabela 4 mostra que a função de ativação logística apresentou um maior erro do que a função relu. Em relação ao solucionador, o adam se mostrou pior do que o lbfgs, por apresentar um maior valor de erro. Portanto, utilizou-se a função de ativação relu, solucionador lbfgs e uma rede neural com 5 camadas de 50 neurônios em cada uma, obtendo assim um valor de RMSE de 313,48W.

4.4 Teste dos modelos

Selecionados os melhores hiper-parâmetros para cada um dos algoritmos, os valores de teste são previstos e os erros em relação aos valores reais são obtidos, obtendo as grandezas vistas na Tabela 5:

Tabela 5 – Erros obtidos por cada modelo.

Algoritmo	RMSE	MAPE	R^2
SVR	319,94	0,68	0,86
DT	384,72	1,23	0,79
KNR	346,46	1,46	0,83
RNA	312,16	1,23	0,86

Fonte: Autoria própria (2022)

Analisando a Tabela 5, é possível perceber que SVR possui o menor MAPE, no valor de 0,68%, e o melhor R^2 ao lado da RNA, ambas com valor de 0,86, indicando alto nível de correlação linear em relação aos valores reais. Em relação ao RMSE, o algoritmo que apresentou o menor valor foi a RNA, sendo, portanto, o melhor algoritmo de acordo com essa métrica. Dessa forma, pode-se dizer que os dois melhores são o SVR e a RNA. Além disso, o MAPE obtido pela RNA foi menor do que o obtido por (ZHU *et al*., 2015). O RMSE da RNA, por sua vez, representa aproximadamente 8,7% da potência total instalada, ficando muito próximo dos 6% obtido por (AL-DAHIDI *et al*., 2019) utilizando Rede Neural com abordagem *ensemble*. No entanto, é importante ressaltar que (WANG *et al*., 2018) conseguiram resultados melhores utilizando o algoritmo *ensemble Gradient-Boost Decision Tree* utilizando apenas a radiação solar e a temperatura como variáveis independentes.

4.5 Comparação das médias

De acordo com o número de tratamentos (5), o número de amostras (606), as previsões realizadas pelos algoritmos e os valores reais (controle), é possível calcular o Quadrado Médio dos Resíduos, tendo sido obtido um valor de 648.164,54.

Depois disso, o valor de Dunnett é determinado pela Tabela de Dunnett levando em consideração o número de tratamentos, de amostras, e o intervalo de confiança de 95% ($\boldsymbol{\alpha}$=0,05), obtendo um valor de 2,442. O valor do Quadrado Médio dos Resíduos e o valor de Dunnett são submetidos à equação 4 para obter o DMS, obtendo um valor de 112,94.

Por fim, são calculadas as diferenças absolutas entre as médias das previsões de cada um dos algoritmos e a média dos valores reais. Obtendo assim, os valores vistos na Tabela 6:

Tabela 6 – Diferenças absolutas entre as médias.

Tratamento	Média	Diferença
Controle	1097,69	-
SVR	1095,72	1,96
DT	1094,36	3,33
KNR	1068,97	28,72
RNA	1084,44	13,24

Fonte: Autoria própria (2022)

Analisando os resultados obtidos na Tabela 6, é possível notar que todas as diferenças são menores do que o DMS, indicando que não há diferença significativa entre as médias obtidas pelos algoritmos e os valores reais. Portanto, pode-se dizer que todos os algoritmos testados são adequados para a previsão.

5 CONSIDERAÇÕES FINAIS

5.1 Conclusões

Aumentar o uso de energias renováveis é um dos principais desafios da humanidade. A disseminação do uso de energia solar passa por entender a influência que os parâmetros meteorológicos exercem sobre ela.

Nesse estudo, foram avaliados algoritmos de *machine learning* para realizar a previsão de potência gerada por um painel fotovoltaico, sendo eles: *Support Vector Regression*, Árvore de Decisão, *K Neighbors Regressor* e Rede Neural Artificial.

Os resultados obtidos por cada um dos algoritmos foram validados por meio do RMSE, MAPE e R^2, sendo que os melhores algoritmos foram o SVR e a RNA. O Teste de Dunnett realizado com intervalo de confiança de 95% concluiu que não há diferença significativa entre os resultados obtidos pelos algoritmos e os valores reais.

Dessa forma, pode-se afirmar que, de acordo com os resultados obtidos pelo Teste de Dunnett, todos os algoritmos são adequados para realizar a previsão, mas que SVR e RNA obtiveram os resultados mais próximos aos valores reais.

5.2 Recomendação de trabalhos futuros

Para um sistema de previsão ser útil, o mesmo precisa funcionar em tempo real, e para isso é necessário estudar o custo computacional de cada um dos algoritmos, já que alguns dos algoritmos utilizados, como a Rede Neural Artificial possuem custo computacional elevado.

Além disso, o estudo foi realizado levando em conta uma instalação residencial com um conjunto de dados limitado. Para avaliar o uso de um sistema em ampla escala, é necessário avaliar os modelos com dados de uma planta fotovoltaica maior, em um conjunto maior.

REFERÊNCIAS

AL-DAHIDI, S. et al. Ensemble approach of optimized artificial neural networks for solar photovoltaic power prediction. **IEEE Access**, IEEE, v. 7, p. 81741–81758, 2019.

ASUERO, A. G.; SAYAGO, A.; GONZALEZ, A. The correlation coefficient: An overview. **Critical reviews in analytical chemistry**, Taylor & Francis, v. 36, n. 1, p. 41–59, 2006.

AWAD, M.; KHANNA, R. Support vector regression. **Efficient Learning Machines**, Apress, Berkeley, CA, p. 67–80, 2015. Disponível em: <https://link.springer.com/chapter/10.1007/978-1-4302-5990-9 4>.

BERRAR, D. **Cross-Validation**. 2019.

CARDOZA, D.; URIBE, J. M.; PALACIOS, J. Risk analysis using meteorological weather factors in solar energy conversion systems. **Dyna**, 2006, Revista DYNA, v.85, n. 205, p. 98–104, 2018.

DAOUD, J. I. Multicollinearity and regression analysis. In: IOP PUBLISHING. **Journal of Physics: Conference Series**. [S.l.], 2017. v. 949, n. 1, p. 012009.

EPE. **Matriz Energética**. 2022. Disponível em: <https://www.epe.gov.br/pt/abcdenergia/matriz-energetica-e-eletrica>.

FRANCISCO, A. C. C. et al. Influência de parâmetros meteorológicos na geração de energia em painéis fotovoltaicos: um caso de estudo do smart campus facens, SP, Brasil. urbe.**Revista Brasileira de Gestão Urbana**, SciELO Brasil, v. 11, 2019.

GEMAN, S.; BIENENSTOCK, E.; DOURSAT, R. Neural networks and the bias/variance dilemma. **Neural Computation**, MIT Press, v. 4, p. 1–58, 1 1992. ISSN 0899-7667. Disponível em: <https://direct.mit.edu/neco/article/4/1/1/5624/ Neural-Networks-and-the-Bias-Variance-Dilemma>.

GERON, A. **Mãos à Obra: Aprendizado de M áquina com Scikit-Learn & TensorFlow**. [S.l.]: Alta Books, RJ, 2019.

GRUS, J. **Data Science Do Zero: Noções Fundamentais com Python**. [S.l.]: Alta Books, 2021. ISBN 9788550816463.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition**. [S.l.]: Springer New York, 2009. (Springer Series in Statistics). ISBN 9780387848587.

INPE. **Evolução mensal e sazonal das chuvas**. 2022. Disponível em: <http://clima1.cptec.inpe.br/evolucao/pt>.

JAMES, G. et al. **An Introduction to Statistical Learning: with Applications in R**. [S.l.]: Springer New York, 2014. (Springer Texts in Statistics). ISBN 9781461471370.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. **arXiv**, 2014. Disponível em: <https://arxiv.org/abs/1412.6980>.

KUMAR, V.; MINZ, S. Feature selection: a literature review. **SmartCR**, v. 4, n. 3, p. 211–229, 2014.

LANA, L. T. C. et al. Energia solar fotovoltaica: revisão bibliográfica. **Engenharias On-line**, v. 1, n. 2, p. 21–33, 2015.

LI, J. et al. Feature selection: A data perspective. **ACM computing surveys (CSUR)**, ACM New York, NY, USA, v. 50, n. 6, p. 1–45, 2017.

MORETTIN, P.; BUSSAB, W. **Estatística Básica**. [S.l.]: Saraiva Educação S.A., 2017. ISBN 9788502207172.

MYTTENAERE, A. de et al. Mean absolute percentage error for regression models. **Neuro-computing**, Elsevier BV, v. 192, p. 38–48, jun 2016. Disponível em: <https://doi.org/10.1016%2Fj.neucom.2015.12.114>.

OLIVEIRA, J. C. de; BELCHIOR, F. N. Energia elétrica produzida por um sistema fotovoltaico versus dados meteorológicos–uma aplicação da correlação de pearson. **Brazilian Journal of Development**, v. 7, n. 5, 2021.

SCIKIT-LEARN. **Decision Tree Regression**. 2022. Disponível em: <https://scikit-learn.org/stable/auto examples/tree/plot tree regression.html>.

SCIKIT-LEARN. **Nearest Neighbors Regression**. 2022. Disponível em: <https://scikit-learn.org/stable/modules/neighbors.html#nearest-neighbors-regression>.

SMOLA, A. J.; SCH ¨OLKOPF, B. A tutorial on support vector regression. **Statistics and computing**, Springer, v. 14, n. 3, p. 199–222, 2004.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern Recognition**. 4. ed. [S.l.]: Academic Press, 2008. ISBN 9781597492720; 1597492728.

TIEPOLO, G. M. et al. Atlas de energia solar do estado do paraná-resultados. **Revista Brasileira de Energia Solar**, v. 9, n. 1, p. 01–10, 2018.

WANG, J. et al. A short-term photovoltaic power prediction model based on the gradient boost decision tree. **Applied Sciences**, Multidisciplinary Digital Publishing Institute, v. 8, n. 5, p. 689, 2018.

ZHU, H. et al. A power prediction method for photovoltaic power plant based on wavelet decomposition and artificial neural networks. **Energies**, MDPI, v. 9, n. 1, p.11, 2015.