Universidade Federal de Ouro Preto
Escola de Minas
Departamento de Engenharia de Minas

SANDYLLA NAIADE OLIVEIRA

USO DE INTERPOLADORES NÃO LINEARES NA MODELAGEM DA MÉDIA LOCAL EM KRIGAGEM SIMPLES COM MÉDIAS LOCAIS

OURO PRETO

2026

SANDYLLA NAIADE OLIVEIRA

USO DE INTERPOLADORES NÃO LINEARES NA MODELAGEM DA MÉDIA LOCAL EM KRIGAGEM SIMPLES COM MÉDIAS LOCAIS

Monografia apresentada ao Departamento de Engenharia de Minas da Escola de Minas da Universidade Federal de Ouro Preto, como parte integrante dos requisitos para obtenção do título de Engenharia de Minas.

Orientador: Prof. Dr. Allan Erlikhman Medeiros Santos

OURO PRETO

2026

MINISTÉRIO DA EDUCAÇÃO
UNIVERSIDADE FEDERAL DE OURO PRETO
REITORIA
ESCOLA DE MINAS
DEPARTAMENTO DE ENGENHARIA DE MINAS

FOLHA DE APROVAÇÃO

Sandylla Naiade Oliveira

Uso de interpoladores não lineares na modelagem da média local em krigagem simples com médias locais

Monografia apresentada ao Curso de Engenharia de Minas da Universidade Federal
de Ouro Preto como requisito parcial para obtenção do título de bacharel em Engenharia de Minas

Aprovada em 24 de abril de 2026

Membros da banca

Dr. Allan Erlikhman Medeiros Santos - Orientador - Universidade Federal de Ouro Preto
Dr. Felipe Ribeiro Souza - Universidade Federal de Ouro Preto
Eng. José Matheus Vieira Matos - Universidade Federal de Ouro Preto

Allan Erlikhman Medeiros Santos, orientador do trabalho, aprovou a versão final e autorizou seu depósito na Biblioteca Digital de Trabalhos de Conclusão de Curso da UFOP em 30/04/2026

Documento assinado eletronicamente por **Allan Erlikhman Medeiros Santos**, **PROFESSOR DE MAGISTERIO SUPERIOR**, em 30/04/2026, às 14:42, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site http://sei.ufop.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0 , informando o código verificador **1099229** e o código CRC **C1C11BB6**.

R. Diogo de Vasconcelos, 122, - Bairro Pilar Ouro Preto/MG, CEP 35402-163
Telefone: 3135591590 - www.ufop.br

AGRADECIMENTOS

Agradeço primeiramente a Deus, por todas as oportunidades colocadas no momento certo para me fazer chegar até esse momento.

Agradeço a minha mãe, Rosa, por ser exemplo de força, luta e muita luz, por sempre a primeira a confiar em mim e por todo apoio até aqui. Você é fonte de inspiração diária.

Ao meu pai, Moacir, por ser a pessoa mais simples e bondosa que conheço e me ensinar todos os dias a confiar na vida e saber apreciar as coisas simples que ela tem a oferecer.

Ao meu irmão, Elionay, por ser meu melhor amigo desde que nasceu. Nossa conexão não é desse mundo.

Tudo que eu faço é por vocês.

Agradeço ao Igor, por ser meu ponto de apoio nos momentos ruins e a primeira pessoa a comemorar comigo os momentos bons. Sorte a minha ter você, te amo e te admiro muito.

À Dona Zelia e Babaçu, pelo acolhimento e por me permitir me sentir em casa mesmo estando tão longe.

Agradeço aos amigos da Engenharia de Minas que fizeram esse caminho ser mais leve: Guto, Gabi, Lóren, Pedro, Arthur e Rafinha.

Agradeço a minha casa, Paraíso, por me acompanhar em toda essa trajetória, os altos e baixos da vida acadêmica e estar presente em cada um deles, em especial KiuBiu, Prace, Minitauro, Konká, Perguntados, Tiana, Fernanda, Laurivane, Maria e Polly. Vocês tornaram a vida ouropretana inesquecível.

As minhas mascotes: Luna, Agatha e Nano.

Ao professor Allan Erlikhman, pelo cuidado, paciência e boa vontade em ensinar. Você se tornou para mim um exemplo de professor, pesquisador e alguém que enxerga além das dificuldades dos alunos. Obrigada.

Aos professores Tatiana, Hernani, Carlão, Elton e Ivo, por todo conhecimento passado. Foi um privilégio ter aprendido tanto com vocês.

Agradeço a Universidade Federal de Ouro Preto e à Escola de Minas, pelo ensino gratuito e de qualidade e por todas as oportunidades de desenvolvimento oferecida.

"De tanto encher meu coração com a beleza, esqueci de sofrer"

Oração para desaparecer

Socorro Acioli

RESUMO

A estimativa de teores em depósitos minerais constitui uma etapa fundamental para a caracterização geológica e para o planejamento das atividades de lavra. Nesse cenário, a geoestatística oferece ferramentas para a modelagem da continuidade espacial e para a estimativa mais consistente de teores. O presente trabalho teve por objetivo analisar o uso de interpoladores não lineares na modelagem da média local em krigagem simples com médias locais (SKLM), um método geoestatístico que busca estimar uma variável primária por meio da krigagem dos resíduos em relação a uma média local previamente modelada. Dessa forma, o uso de interpoladores não lineares visa contribuir para a estimativa de uma variável primária a partir de sua relação com uma variável secundária. Para isso, foram empregados modelos de regressão linear, Random Forest e Redes Neurais Artificiais na representação da média local, seguidos da aplicação do método geoestatístico para estimativa dos resíduos e composição das estimativas finais. A análise considerou o comportamento espacial dos dados, a continuidade representada por variogramas e o desempenho dos modelos com base em métricas de validação, como RMSE (raiz do erro quadrático médio), MAE (erro médio absoluto) e R^2 (coeficiente de determinação). Os resultados indicaram que os interpoladores não lineares apresentaram potencial por capturar relações mais complexas e, essa conclusão foi evidenciada pelo resultado das métricas de validação: o método de regressão linear teve como resultados RMSE = 0,3947, MAE = 0,2821 e R^2 = 0,7369; Random Forest com RMSE = 0,3592, MAE = 0,2572 e R^2 = 0,7821 e Redes Neurais com resultados RMSE = 0,3641, MAE = 0,2667 e R^2 = 0,7760. O Random Forest demonstrou maior capacidade de representar a média local e, consequentemente, entregar melhores estimativas finais, o que mostra a capacidade dos nterpoladores não lineares em contribuir para o aprimoramento do método SKLM em problemas relacionados a geoestatística multivariada.

Palavras-chave: Geoestatística; Krigagem simples com médias locais; Média local; Random Forest; Redes Neurais Artificiais; Estimativa de Teores.

ABSTRACT

Estimating grades in mineral deposits is a fundamental step in geological characterization and in planning mining activities. In this context, geostatistics provides tools for modeling spatial continuity and for more consistent grade estimation. The objective of this study was to analyze the use of nonlinear interpolators in modeling the local mean in simple kriging with local means (SKLM), a geostatistical method that seeks to estimate a primary variable by kriging the residuals relative to a previously modeled local mean. Thus, the use of nonlinear interpolators aims to contribute to the estimation of a primary variable based on its relationship with a secondary variable. To this end, linear regression models, Random Forest, and Artificial Neural Networks were employed to represent the local mean, followed by the application of the geostatistical method to estimate the residuals and compose the final estimates. The analysis considered the spatial behavior of the data, continuity represented by variograms, and model performance based on validation metrics such as RMSE (root mean square error), MAE (mean absolute error), and R^2 (coefficient of determination). The results indicated that nonlinear interpolators have the potential to capture more complex relationships, and this conclusion was supported by the validation metrics: the linear regression method yielded results of RMSE = 0,3947, MAE = 0,2821 and R^2 = 0,7369; Random Forest with RMSE = 0,3592, MAE = 0,2572 and R^2 = 0,7821; and Neural Networks with results of RMSE = 0,3641, MAE = 0,2667 and R^2 = 0,7760. Random Forest demonstrated a greater ability to represent the local mean and, consequently, to deliver better final estimates, which demonstrates the ability of nonlinear interpolators to contribute to the improvement of the SKLM method in problems related to multivariate geostatistics.

Keywords: Geostatistics; Simple Kriging with Local Means; Local Mean; Random Forest; Artificial Neural Networks; Content Estimation.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1. INTRODUÇÃO

A distribuição de teores em um depósito mineral não ocorre de forma uniforme no espaço, uma vez que o processo de mineralização costuma originar zonas ricas e pobres dentro de uma mesma estrutura geológica. Essa heterogeneidade, no entanto, não impede que exista certo grau de continuidade espacial entre os valores observados. Segundo Matheron (1971), apesar de possuir certo grau de continuidade, um depósito de ouro não é regular o suficiente para ter seus teores estimados de forma determinística. Por essa razão, na caracterização de um depósito devem ser considerados duas características principais: seu fator estrutural e aleatório, inerente a qualquer depósito.

G. Matheron propôs, na década de 1960, o que hoje conhecemos como Geoestatística, que é definido por ele mesmo como "A aplicação do formalismo de funções aleatórias para o reconhecimento e estimativa de fenômenos naturais". Um fenômeno natural pode ser caracterizado como a distribuição no espaço de uma ou mais variáveis, nomeadas de variáveis regionalizadas (Journel e Huijbregts, 1978). A partir dessa base teórica, a Geoestatística passou a oferecer ferramentas, métodos e teorias que ajudam a interpretar a estrutura geológica que está sendo estudada.

Entre as diferentes abordagens geoestatísticas, destaca-se a análise de múltiplas variáveis, que é a ampliação da teoria clássica da Geoestatística para a estimativa de n variáveis que possuem correlação entre si, que é o foco deste trabalho. Essa perspectiva mostra-se muito relevante quando a variável de interesse (ou variável primária) possui um conjunto de dados limitado, mas apresenta relação com uma variável secundária. Nesse contexto, tem-se o uso da Krigagem, mais especificamente o método SKLM, empregado para estimar os teores explorando a relação entre essas duas variáveis. Assim, o problema central do presente trabalho consiste em avaliar de que forma a modelagem da média local pode contribuir para a melhoria das estimativas geoestatística em contextos multivariados.

Para aprimorar ainda mais as estimativas através da utilização da teoria e dos métodos geoestatísticos, e, consequentemente, complementar essa representação das médias locais, foi explorado nesse trabalho o uso de ferramentas de inteligência artificial, mais especificamente os métodos *Random Forest* e Redes Neurais Artificiais. Essas técnicas possuem um grande potencial em sua capacidade de aprender padrões a partir dos dados, de forma a fornecer respostas satisfatórias e, em alguns casos, desempenho comparável ou até mesmo superior aos métodos tradicionais. Essa abordagem busca vincular os aspectos teóricos e clássicos da Geoestatística ao fator de expansão do uso das inteligências Artificiais dentro da ciência.

2. OBJETIVOS

2.1 Objetivo Geral

Avaliar o uso de interpoladores não lineares na modelagem da média local em krigagem simples com médias locais, analisando seu impacto nos resultados de estimativa em comparação com abordagens lineares tradicionais.

2.2 Objetivos Específicos

Os objetivos específicos são:

- Implementar a krigagem simples com médias locais (SKLM) utilizando um modelo linear como referência para a modelagem da média local;
- Desenvolver modelos de média local baseados em interpoladores não lineares, especificamente *Random Forest* e redes neurais artificiais, a partir da relação entre a variável primária e a variável secundária;
- Avaliar e comparar o comportamento dos modelos linear, *Random Forest* e redes neurais artificiais na representação da relação entre variável primária e variável secundária;
- Calcular e analisar os resíduos associados a cada técnica de modelagem da média local, verificando sua distribuição estatística e estrutura espacial;
- Aplicar a krigagem dos resíduos para cada abordagem de média local, integrando os resultados ao modelo SKLM final;
- Validar os modelos SKLM obtidos por meio de validação cruzada *leave-one-out* (LOOCV), utilizando métricas estatísticas como RMSE, MAE e coeficiente de determinação (R^2);
- Comparar os resultados finais dos modelos SKLM linear, SKLM-*Random Forest* e SKLM- redes neurais artificiais por meio de análises quantitativas e qualitativas, incluindo mapas, seções e gráficos comparativos;
- Analisar criticamente os ganhos e limitações da utilização de interpoladores não lineares na modelagem da média local em krigagem simples com médias locais. Comparar os resultados obtidos com aqueles existentes na literatura.

3. REVISÃO BIBLIOGRÁFICA

A revisão bibliográfica teve por finalidade apresentar o estado da arte dos seguintes conteúdos abordados no presente estudo: geoestatística multivariada, uso de variáveis secundárias, krigagem simples com médias locais, técnicas de inteligência artificial, aplicação de inteligência artificial na geoestatística.

3.1 Geoestatística multivariada

Dentro do contexto da engenharia de minas e, principalmente, da estimativa de depósitos, é de extrema importância conhecer a extensão e os teores de um determinado corpo de minério e, para isso, a Geoestatística surge como uma ferramenta apropriada para este fim. A Geoestatística é uma aplicação da Teoria das Variáveis Regionalizadas, idealizada por George Matheron entre 1957 e 1962, que leva em consideração os aspectos aleatório e espacial das amostras em estudo, buscando algum tipo de correlação linear entre elas e como uma amostra é capaz de influenciar outras a uma dada distância.

Com base nisso, a Geoestatística Multivariada se caracteriza como uma extensão da Geoestatística clássica, tratando de uma ou mais variáveis que possuem correlação entre si (como é o caso de diferentes espécies químicas, diferentes elementos, dentre outros). A utilização da Multivariada se dá, principalmente, quando deseja-se estimar, de forma consistente, variáveis primárias que foram amostradas de forma limitada e, para conhecê-las, é preciso utilizar variáveis secundárias para sua estimativa (desde que ela esteja disponível em todos os pontos em que se deseja estimar a variável primária), por meio da correlação existente entre elas.

3.2 Conceitos básicos de geoestatística multivariada

Para que seja possível entender a geoestatística multivariada e como ela é utilizada no contexto da mineração, é importante compreender alguns conceitos básicos que a compõem. A seguir, serão explicados esses conceitos e suas aplicações.

3.2.1 Fenômenos regionalizados multivariados

Como mencionado no tópico 3.1, a Geoestatística é baseada na Teoria das Variáveis Regionalizadas, proposta por George Matheron. Segundo Guerra (1985), as variáveis regionalizadas são aquelas cujos valores são relacionados de algum modo com a posição espacial que ocupam. Vinculado a isso, têm-se duas características importantes acerca dessas

variáveis: seu aspecto aleatório, ou seja, elas variam de um ponto a outro no espaço; e seu caráter espacial, o qual os valores medidos e observados, apesar de possuírem certa variabilidade, não são totalmente independentes, pois possuem alguma relação com o espaço. Por isso, apesar da relação espacial existente entre dois pontos, eles também estão sujeitos a aleatoriedade, o que lhes confere certa margem de erro.

Segundo Journel e Huijbregts (1978), a teoria da Geoestatística se baseia na observação de que as variáveis regionalizadas possuem uma estrutura particular: os teores do metal de interesse *z(x)* e *z(x+h)* nos pontos *x* e *x+h* são auto correlacionados e, essa correlação, depende do vetor *h* (tanto em módulo quanto em direção), que separa esses dois pontos, no contexto da mineralização considerada. Outro aspecto importante tratado por Journel e Huijbregts (1978) é como serão interpretadas as variáveis regionalizadas, que devem levar em consideração as duas características dessa variável (aleatoriedade e estrutura espacial) a fim de fornecer uma representação simples do espaço com uma abordagem que seja consistente, e, uma dessas abordagens, corresponde à interpretação probabilística por meio de funções aleatórias, que se caracteriza como uma realização particular de um conjunto de variáveis regionalizadas, descrito como *Z(x)*.

Dessa forma, os fenômenos regionalizados multivariados consistem na aplicação da Geoestatística clássica para duas ou mais variáveis, assim como descreve Goovaerts (1997, pag 72) "[...] as notações e conceitos utilizados para definir uma função aleatória podem também ser estendidas ao caso multivariado, denotado como Z(u) e chamado de função aleatória multivariada."

3.2.2 Estrutura espacial e continuidade das variáveis

Para que seja possível interpretar a disposição das variáveis regionalizadas no espaço, é preciso compreender a sua continuidade espacial e como cada amostra tem um efeito sobre a outra no espaço a uma determinada distância *h.* De forma geral, dois pontos próximos entre si possuem uma maior possibilidade de serem similares do que dois pontos que se encontram separados a uma distância maior e, nos referimos a este fenômeno como continuidade espacial, que é a condição entre duas ou mais variáveis de possuírem uma relação entre si devido ao espaço em que ocupam (Isaaks & Srivastava, 1989), ou seja, valores baixos tendem a estar próximos a valores mais baixos e o mesmo vale para valores mais altos. Dessa forma, utilizamos ferramentas que sejam capazes de demonstrar qual a relação existente entre essas variáveis que possuem uma localização próxima.

Dentre as formas de medição dessa relação espacial, neste trabalho citaremos o uso de *h-scattergram* e semivariogramas.

O *h-scattergram* consiste em um diagrama de dispersão construído a partir de pares de valores ($z(u_\alpha)$, $z(u_\alpha+h)$) de um mesmo atributo z em locais separados por uma dada distância *h* e direção θ. Em geral, esses pares de dados geralmente são agrupados em classes de distância (ou *lags*) e direção, o que permite avaliar, de forma preliminar, o grau de associação espacial entre os dados. Quando os pontos do diagrama tendem a se concentrar ao redor da reta de 45º, sugere maior continuidade espacial para aquele intervalo de separação; por outro lado, maior dispersão dos pares indica perda de correlação entre os valores. Além disso, o *h-scattergram* auxilia na identificação de valores muito isolados que podem indicar *outliers*, inconsistências amostrais ou mesmo dados que foram registrados de forma incorreta, a que devem ser identificados e analisados cuidadosamente (Goovaerts, 1997).

Os semivariogramas, por sua vez, constituem a principal ferramenta geoestatística para quantificar a continuidade espacial de uma variável regionalizada. Se tratam de uma ferramenta matemática que indica como se segue a dispersão espacial dos pontos amostrados em um dado corpo. De acordo com Yamamoto (2013), a função variograma mede a variância entre dois pontos separados por uma distância ***h*** e, por isso, quanto mais próximos os pontos, menor a variação, a qual tende a aumentar à medida em que aumenta a distância entre eles. Geralmente, a partir de certa distância, a variância tende a se estabilizar em torno da variância máxima (denominada patamar), a partir de uma certa distância, conhecida como alcance, além da qual os pares de amostras passam a representar baixa ou nenhuma correlação espacial. Na figura 3.1, estão representados os principais parâmetros presentes em um semivariograma.

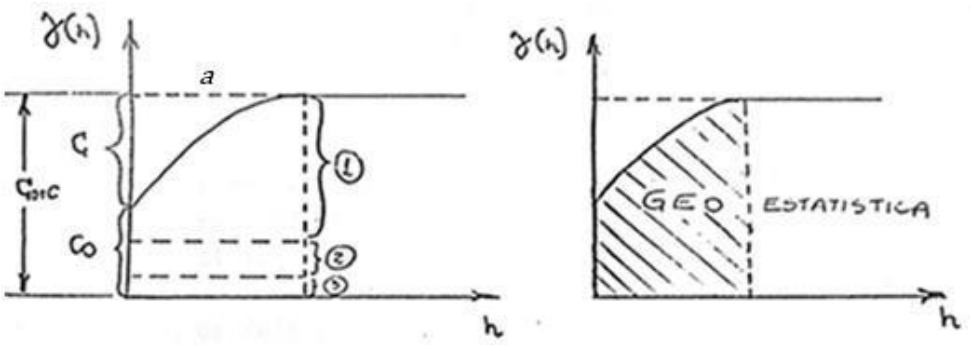

Figura 3.1 - Fenômenos de transição.
Fonte: Adaptado de Guerra (1985).

- ***a*** - Alcance: marca a partir de qual distância ***h*** um determinado ponto não exerce mais influência no ponto localizado a uma distância ***h* + *x***, definindo, assim, o início de uma zona aleatoriedade (no gráfico da Figura 3.1, é a abscissa do ponto a partir do qual a curva se torna horizontal);
- **C** - Variância espacial: é definida como as diferenças espaciais entre dois pontos devido a distâncias cada vez maiores;
- $\mathbf{C_o}$ - Efeito pepita: representa as variações locais ou a pequenas distâncias;
- $\mathbf{C + C_o}$ - Patamar: marca o valor da variância máxima, ou seja, o ponto onde ela se estabiliza.

3.2.3 Conceito de estacionaridade

Segundo Wackernagel (2003), a estacionaridade pode ser entendida como a condição em que determinadas propriedades estatísticas de uma função aleatória, como média e variância, permanecem a mesma quando há o deslocamento de um conjunto de *n* pontos de uma região para outra no espaço, fenômeno denominado invariância por translação. Journel e Huijbregts (1978) sugere que uma função aleatória pode ser dita estacionária quando dois componentes vetoriais (variáveis aleatórias) $\{Z(x_1), ..., Z(x_k)\}$, e $\{Z(x_{1+h}), ..., Z(x_{k+h})\}$ são idênticas em distribuição, ou seja, possuem a mesma distribuição de um conjunto de *k* variáveis.

Esse conceito pode ser dividido em duas possibilidades principais, a estacionaridade de segunda ordem e a hipótese de estacionaridade intrínseca.

De acordo com Journel e Huijbregts (1978, pág.32) uma função aleatória é dita de segunda ordem quando sua média existe e é constante no domínio, isto é (Equação 1):

$$E\{Z(x)\} = m, \quad (1)$$

e quando a covariância entre dois pontos depende apenas do vetor de separação h, e não da posição absoluta desses pontos no espaço. Assim, dois pares de pontos apresentam o mesmo comportamento estatístico desde que estejam separados pela mesma distância e direção. Nessa condição, a covariância pode ser expressa como mostra a Equação 2:

$$C(h) = E\{Z(x+h)\,.\,Z(x)\} - m^2 \quad (2)$$

Por sua vez, a hipótese intrínseca, também apresentada por Journel e Huijbregts (1978, pág. 33), mantém a condição de média constante no domínio, isto é, $E\{Z(x)\} = m$, mas o foco se desloca da variável em si para os seus incrementos espaciais, assumindo-se que, para todo vetor h, a diferença *[Z(x+h) – Z(x)]* possui variância finita e dependente apenas de h e não da posição x. Essa condição é descrita pela Equação 3:

$$Var\{Z(x+h) - Z(x)\} = E\{[Z(x+h) - Z(x)]^2\} = 2\gamma(h) \quad (3)$$

em que $\gamma(h)$ corresponde à função variograma. Dessa forma, a hipótese intrínseca permite modelar a continuidade espacial por meio do semivariograma, mesmo em situações em que a formulação de segunda ordem não seja plenamente satisfeita.

3.3 Uso de variáveis secundárias na geoestatística

As variáveis secundárias constituem uma alternativa para melhorar a estimativa de uma variável primária, a partir de sua correlação com a mesma. A seguir, esse conceito é explicado, assim como sua utilização na geoestatística.

3.3.1 Conceito de variável secundária

Do que diz respeito a estimativa de depósitos, um grande desafio está relacionado à disponibilidade de amostras da variável primária no espaço amostral. Como forma de mitigar essa dificuldade, utiliza-se, então, as variáveis secundárias, de modo a explorar uma correlação existente entre elas, que explique como a variável de interesse em questão está distribuída em um espaço amostral.

Segundo Isaaks & Srivastava (1989), na maioria das vezes, um conjunto de dados conta não apenas com a variável primária, mas também com uma ou mais variáveis secundárias, que são aquelas em que, geralmente, estão correlacionadas espacialmente com a variável primária e contém informações úteis sobre esta, levando em consideração que, em algumas áreas, a única informação disponível sobre a variável primária será essa correlação. Ainda segundo Isaaks & Srivastava (1989, p.401), "[...] o padrão de amostragem da variável amostrada com maior frequência é mais regular do que a variável subamostrada". Ou seja, esses dados secundários irão nos mostrar mais sobre a área estimada do que seria exposto apenas por meio da variável primária de interesse.

3.3.2 Variáveis secundárias exaustivas

As medidas diretas das informações primárias de interesse podem ser complementadas com as informações secundárias, dada uma relação existente entre elas. Estabelecida essa relação, é preciso que a informação (ou variável) secundária esteja "exaustivamente amostrada". Dizer que uma amostra está exaustivamente amostrada significa que ela está disponível em todos os locais em que os dados primários de interesse estão sendo estimados (Goovaerts, 1997). Se a informação secundária é densamente disponível, mas não exaustiva, uma aproximação possível para completar os dados pode ser feita por meio da interpolação ou, ainda melhor, por simulação, como pontua Goovaerts (1997).

Dada essa complementação dos dados primários a partir de dados secundários, pode-se inferir dois tipos de informação a cerca desta correlação:

- Um atributo categórico (como, por exemplo, o tipo de rocha);
- Um atributo de variação contínua *y* (como o teor de um determinado mineral).

3.3.3 Correlação entre variável primária e secundária

Como citado no tópico 3.3.1 e 3.3.2, é possível estimar uma determinada variável primária por meio de uma ou mais variáveis secundárias, desde que elas sejam amostradas exaustivamente, dessa forma, é possível calcular se há algum tipo de correlação entre estas variáveis. Segundo Isaaks & Srivastava (1989, p.30) existem três tipos de correlação entre essas duas variáveis: positivamente correlatadas, negativamente correlatadas ou não correlatadas.

Duas variáveis são ditas positivamente correlatadas se os maiores valores da variável *x* podem ser associados aos maiores valores de uma variável *y*, como é o caso de correlacionar permeabilidade com porosidade, que são tipicamente fatores correlacionados positivamente (Isaaks & Srivastava, 1989). Dessa forma, se plotarmos um *scatterplot* dessas duas grandezas, teremos os seus maiores valores associados entre si, enquanto que, para variáveis correlacionadas negativamente, os maiores valores de uma dada variável x estarão associados aos menores valores de uma variável *y* (como, por exemplo, a correlação entre elementos principais) e, neste caso, plotado um *scatterplot* com os seus respectivos dados, têm-se associados os maiores valores de *x* aos menores valores de *y*. Por último, têm-se a possibilidade de não haver quaisquer correlações entre as variáveis *x* e *y* e, portanto, o aumento ou diminuição de uma das variáveis, não causará nenhum tipo aparente de efeito sobre a outra. (Isaaks & Srivastava, 1989)

3.3.3.1 Coeficiente de correlação

O coeficiente de correlação (ρ) é uma medida estatística que tem por objetivo calcular a relação entre duas variáveis, por meio da seguinte equação:

$$\rho = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - m_x)(y_i - m_y)}{\sigma_x \sigma_y} \quad (4)$$

- *n:* número de dados;
- $x_1, \ldots, x_n$*:* os valores da variável primária;
- m_x*:* média da variável primária;
- σ_x*:* desvio padrão da variável primária;
- $y_1, \ldots, y_n$*:* os valores da variável secundária;
- m_y*:* média da variável secundária;
- σ_y*:* desvio padrão da variável secundária.

Para o caso de uma relação não linear entre as variáveis, pode-se utilizar uma medida para complementar o coeficiente de correlação linear com outra medida de grau de relação entre estas variáveis. Isaaks & Srivastava (1989, p.30) utilizam o coeficiente de correlação de postos, por meio da equação a seguir:

$$\rho_{rank} = \frac{\frac{1}{n}\sum_{i=1}^{n}(\mathrm{R}x_i - m_{Rx})(\mathrm{R}y_i - m_{Ry})}{\sigma_{Rx}\sigma_{Ry}} \quad (5)$$

- *n:* número de dados;
- $Rx_1, \ldots, Rx_n$*:* é o posto de x_i em relação a todos os demais valores de *x*, calculado geralmente, ordenando-se os valores de *x* em ordem crescente e observando a posição que cada valor ocupa;
- m_{Rx}*:* média de todas as posições da variável primária;
- σ_{Rx}*:* desvio padrão de todas as posições da variável primária;
- $Ry_1, \ldots, Ry_n$*:* é o posto de x_i em relação a todos os demais valores de *x*, calculado geralmente, ordenando-se os valores de *x* em ordem crescente e observando a posição que cada valor ocupa;
- m_{Ry}*:* média de todas as posições da variável primária;
- σ_{Ry}*:* desvio padrão de todas as posições da variável primária.

3.3.4 Importância da correlação entre variáveis

A correlação entre variáveis permite identificar a existência de padrões de associação e compreender em que medida esses padrões podem ser utilizados na análise e interpretação de um determinado banco de dados. Segundo Isaaks & Srivastava (1989 p. 32-33), a interpretação adequada do coeficiente de correlação entre as variáveis analisadas fornece um indicativo do quão bem-sucedida pode ser a estimativa de uma variável a partir de outra, utilizando como base as relações estatísticas descritas anteriormente.

Pode-se inferir que, quanto mais distante de zero estiver o valor do coeficiente de correlação, mais correlatados estarão estes dados (levando em consideração o fato que esse valor pode variar de -1 a +1, explicitando os tipos de correlação existentes: positiva, negativa ou inexistente). Essa análise nos permite entender como (e se) é possível relacionar as variáveis secundárias com a variável primária de interesse, por meio dessa correlação entre elas.

3.4 Krigagem simples com médias locais (SKLM)

A seguir será abordado o método de Krigagem simples com médias locais por meio de sua formulação conceitual, suas vantagens e limitações em relação à krigagem tradicional, sua modelagem e as abordagens lineares tradicionais.

3.4.1 Formulação conceitual do método SKLM

O método teve grande desenvolvimento por volta de 1951, na África do Sul, quando depósitos que eram considerados ricos, na prática, não eram realmente ricos e o mesmo acontecia para os depósitos mais pobres. Daniel G. Krige, então, sugeriu que esse problema poderia ser solucionado por meio do uso da teoria de regressão, iniciando uma busca para encontrar um fator de correção que pudesse atenuar os valores altos (David, 1977). A partir dos estudos de Krige, Matheron impulsionou, mais tarde, o surgimento e desenvolvimento da geoestatística.

A krigagem pode ser considerada uma ferramenta de estimativa de valores de variáveis que não foram amostradas por meio de variáveis que já foram amostradas, explorando a dependência espacial existente entre eles. Segundo Journel e Huijbregts e Huijbregts (1978), é definida como uma técnica de estimativa que fornece a melhor estimativa linear sobre características não conhecidas (conhecido como BLUE – *Best Linear Unbiased Estimation*). Essa ferramenta auxilia no desafio de encontrar um valor médio de variáveis regionalizadas em uma estimativa local dado um certo limite de domínio, seja por questões financeiras, licenças, controle geológico ou comportamento estatístico dos dados. De acordo com Yamamoto (2013),

a krigagem pode ser utilizada como algoritmo estimado tanto para a previsão de valores pontuais de variáveis regionalizadas dado um certo domínio, quanto para calcular o valor médio de uma variável regionalizada para volumes maiores do que aquele em que a variável foi estimada. Dessa forma, o método de krigagem permite manipular dados ainda não conhecidos por meio daqueles que estão ao alcance naquele determinado cenário de estimativa. Como citado nos capítulos anteriores, os atributos secundários, quando bem correlacionados com os dados primários, seja de forma negativa ou positiva, podem nos fornecer esse tipo de informação e, neste trabalho, trabalharemos com o método de Krigagem Simples com Médias Locais (SKLM), que permite esse tipo de avaliação.

O método de Krigagem Simples com Médias Locais, (ou *Simple kriging with local mean*) é descrito por Goovaerts (1997, p. 190-192) por meio da equação de krigagem simples (Equação 6) como pode ser descrita a equação de uma SKLM (Equação 7):

$$Z^*_{SK}(u) - m = \sum_{\alpha=1}^{n(u)} \lambda^{SK}_{\alpha}(u)[Z(u_{\alpha}) - m] \quad (6)$$

Apesar da média m não depender da localização u, ela representa uma informação global comum a todos os locais ainda não amostrados. Desse modo, para cada informação secundária disponível em uma localização u, a média estacionária conhecida m, pode ser substituída por uma média variável conhecida $m^*_{SK}(u)$, convertendo-a de krigagem simples para krigagem simples de médias locais (SKLM).

$$Z^*_{SKlm}(u) - m^*_{SKlm}(u) = \sum_{\alpha=1}^{n(u)} \lambda^{SK}_{\alpha}(u)[Z(u_{\alpha}) - m^*_{SK}(u_{\alpha})] \quad (7)$$

3.4.2 Modelagem da média local no método SKLM

Conforme discutido no tópico 3.3.2, uma determinada variável secundária pode ser classificada como categórica ou contínua. Neste trabalho, o atributo secundário y é do tipo contínuo. Nessas condições, a média local da variável primária pode ser expressa como uma função do valor assumido por esse atributo secundário no ponto u, como aponta Goovaerts (1997. Pág. 190). Isso significa que a média da variável primária deixa de ser tratada como constante em todo domínio e passa a variar localmente de acordo com o comportamento espacial da variável auxiliar. Assim, a média local pode ser representada pela Equação 8:

$$m^*_{SK} = f(y(u)) \quad (8)$$

em que m^*_{SK} corresponde à média local estimada da variável primária no ponto u, e $f(y(u))$ representa a função que relaciona a variável secundária ao valor esperado da variável primária.

3.4.3 Vantagens e limitações em relação à krigagem tradicional

O método SKLM, quando comparado à krigagem tradicional, possui uma vantagem principal que está relacionada a possibilidade de incorporar médias locais variáveis (Goovaerts, 1997) relacionadas à informação secundária, que em cenários onde a variável primária é escassa e possui uma forte correlação linear com uma variável auxiliar (Wackernagel, 2003), pode contribuir para uma melhor interpretação e estimativa dos dados. Nesse cenário, quando a média da variável primária não pode ser tratada como constante, a estimativa é trabalhada a partir dos resíduos, que representa a parte sem a tendência global (Wackernagel, 2003. Pág. 86) e que, por possuir uma média que tende a ser centrada em zero, produz resultados satisfatórios. Enquanto isso, a krigagem tradicional se restringe a uma média conhecida e constante no domínio estudado, e nesse caso, o uso da variável secundária pode não ser propriamente aproveitada pelo método.

Apesar disso, o método SKLM necessita não apenas que a variável auxiliar possua correlação com a variável primária, mas também que seja exaustivamente amostrada, como citado no tópico 3.4.1. Dessa forma, se a relação entre as variáveis for fraca ou mal modelada, ou se a variável secundária não estiver amostrada de forma exaustiva (ou de forma densa, com complementação de interpolação), as estimativas de SKLM podem se tornar inconsistentes e pouco condizentes com os valores reais do domínio.

3.4.4 Validação cruzada (*Cross Validation*)

A validação cruzada (ou *cross validation*) pode ser definida com uma forma de verificar se o método de estimativa apresentou um resultado condizente com o domínio estudado. Para isso, o valor de uma amostra $Z(x_\alpha)$ é temporiamente removida do ponto x_α, e o valor $Z^*(x_{[\alpha]})$ é estimado utilizando o restante das amostras. A Figura 3.2 mostra como é feito esse processo, também conhecido como LOOCV – *Leave-one-out.*

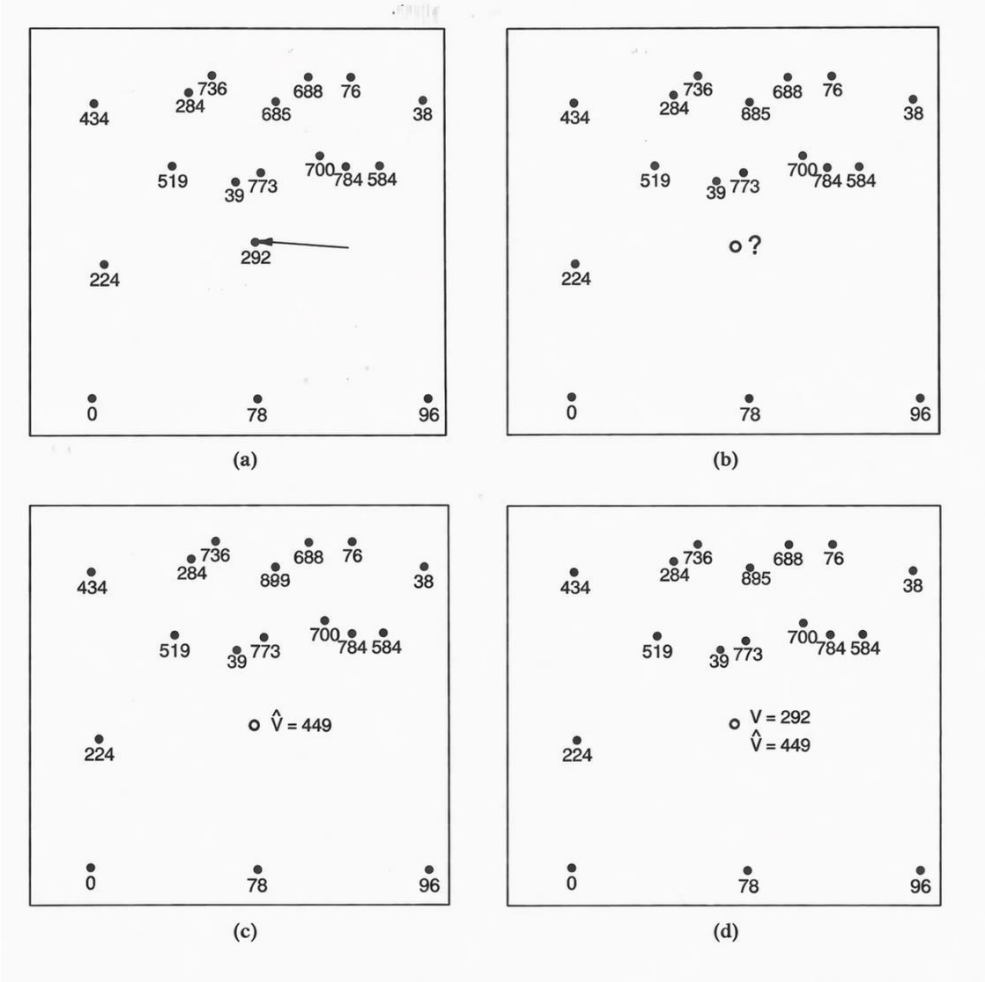

Figura 3.2 – Um exemplo de procedimento de validação cruzada. A amostra na localização destacada pela seta em (a) é removida, restando as 17 amostras mostradas em (b). Utilizando apenas essas 17 amostras, o valor no ponto marcado como '◦' é estimado; em (c), a estimativa é calculada usando ponderação pelo inverso do quadrado da distância. Essa estimativa pode ser então comparada com o valor real que foi removido anteriormente, obtendo-se assim um par de valores estimado e verdadeiro.

Fonte: Isaaks & Srivastava (1989)

Esse processo é repetido para todas as amostras disponíveis, com o intuito de comparar o valor real com o valor estimado para a amostra. A partir desses valores, é possível calcular o erro relacionado a esta estimativa, por meio da Equação 9.

$$Z(x_\alpha) - Z^*(x_{[\alpha]}) \quad (9)$$

Esse valor nos permite analisar o quão boa ou ruim foi a estimativa do método aplicado, representando o quanto este método se ajustou aos valores das amostras vizinhas. Outra forma de analisar os valores da validação cruzada é a partir da média do erro (equação 10): quanto mais próxima de zero for a média do erro da validação cruzada, menos viés ela apresenta,

enquanto uma média mais distante de zero pode representar uma super ou subestimativa da amostra (Wackernagel, 2003. Pág. 87).

$$\frac{1}{n}\sum_{\alpha=1}^{n}(Z(x_{\alpha}) - Z^{*}(x_{[\alpha]})) \cong 0 \quad (10)$$

As métricas utilizadas para avaliar a performance dos modelos de predição utilizados neste trabalho foram RMSE, MAE e R^2.

O RMSE (*Root Mean Square Error*) é a raiz quadrada do erro quadrático médio. Ele mede o desvio padrão dos resíduos por meio da Equação 11:

$$RMSE = \sqrt{\cancel{MSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2} \quad (11)$$

O MAE (*Mean Absolute Error*), ou erro absoluto médio, representa a média das diferenças absolutas entre os valores observados e os valores estimados no conjunto de dados. Ele mede, portanto, a média dos resíduos em termos absolutos; é calculado por meio da Equação 12:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \quad (12)$$

O coeficiente de determinação (R^2) representa a proporção da variância da variável dependente que é explicada pelo modelo de regressão linear. Trata-se de uma métrica adimensional, ou seja, independe da escala dos dados. Seu valor é sempre menor ou igual a 1 e é calculado a partir da Equação a seguir:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \quad (13)$$

3.4.5 Abordagens lineares tradicionais

Entre os métodos clássicos de estimativa geoestatística, as abordagens lineares tradicionais de krigagem apresentam diferença entre si principalmente pelo fato de assumirem hipóteses diferentes quanto a média da variável regionalizada de estudo. A seguir, são apresentados os métodos de regressão linear, krigagem simples e krigagem ordinária.

3.4.5.1 Regressão linear

Como mostrado em tópicos anteriores, quando duas variáveis apresentam uma relação consistente entre si, o conhecimento de uma delas pode contribuir para a estimativa de outras. Uma forma de estimar essa relação é por meio da regressão linear, como sugere Isaaks & Srivsatava (1989). Nessa abordagem, admite-se que a relação entre as variáveis possa ser descrita por uma função linear, expressa pela Equação 11:

$$y = ax + b \quad (14)$$

em que *a* representa o coeficiente angular da reta e *b* a constante ou intercepto do modelo. Essa formulação permite estimar o comportamento médio de uma variável em função da outra, podendo ser utilizada como uma primeira aproximação da relação entre atributos primários e secundários. Uma alternativa a essa representação linear, segundo Isaaks & Srivsatava (1989), consiste em calcular a média dos valores de y para diferentes intervalos de x. Nesse caso, em vez de impor uma única reta para todo o conjunto de dados, busca-se descrever a relação entre as variáveis a partir do comportamento médio observado em classes sucessivas da variável explicativa.

3.4.5.2 Krigagem simples

A krigagem simples (também conhecida como krigagem com médias conhecidas) é utilizada quando a média da variável regionalizada é considerada conhecida e constante em todo domínio de estudo. A estimativa é construída a partir dos resíduos em relação a essa média, de modo que os valores amostrados são corrigidos em função de seu desvio em relação ao valor médio *m*. De acordo com Wackernagel (2003), a estimativa da krigagem simples deve se basear em uma covariância conhecida entre duas variáveis aleatórias nos pontos estudados. Dessa forma, a krigagem simples pode ser entendida como uma combinação linear ponderada dos resíduos em torno da média conhecida, na qual os valores observados $Z(x_\alpha)$ no ponto x_α são utilizadas para estimar os valores de $Z(x_0)$ no ponto x_0. A estimativa pode ser expressa por (Equação 12):

$$Z^*(x_0) = m + \sum_{\alpha=1}^{n} w_\alpha (Z(x_\alpha) - m)) \quad (15)$$

em que m representa a média conhecida da variável e w_α correspondem aos pesos associados aos resíduos $Z(x_\alpha) - m$.

3.4.5.3 Krigagem ordinária

A krigagem ordinária (ou *Ordinary Kriging*) é uma estimativa de combinações lineares ponderadas dos dados disponíveis, procurando construir um estimador não tendencioso, isto é, com erro médio igual a zero, ao mesmo tempo em que minimiza a variância do erro de estimativa. De acordo com Isaaks & Srivastava (1989, pág. 278 e 279) esse método pode ser mais condizente com as situações reais, uma vez que, na prática, raramente se conhece com precisão a média da variável ou sua covariância em todo domínio.

Diferentemente da krigagem simples, a krigagem ordinária não exige o conhecimento prévio da média, e sua estimativa é obtida diretamente a partir de uma combinação linear dos valores observados, sujeita à restrição de que a soma dos pesos seja igual a 1. Essa condição garante não tendenciosidade do estimador e pode ser representada por:

$$Z^*_{OK}(u) = m + \sum_{\alpha=1}^{n(u)} \lambda_\alpha^{OK}(u)\, Z(u_\alpha) \quad (16)$$

com a restrição:

$$\sum_{\alpha=1}^{n(u)} \lambda_\alpha^{OK}(u) = 1 \quad (17)$$

em que $\lambda_\alpha^{OK}(u)$ são os pesos atribuídos aos dados vizinhos utilizados para estimar o valor da variável no ponto u.

3.5 Técnicas de inteligência artificial

Nos dias atuais, as técnicas de inteligência artificial têm ganhado destaque em diferentes áreas do conhecimento devido à sua capacidade em reconhecer padrões, processar e analisar grandes volumes de dados e auxiliar na tomada de decisões. Dessa forma, o estudo e compreensão dessas técnicas se tornam importantes para que seja possível utilizar essas ferramentas para melhoria na análise e interpretação de dados. A seguir, são apresentadas duas ferramentas de inteligência artificial utilizados neste trabalho.

3.5.1 Random Forest

3.5.1.1 Conceito de árvore de decisão

Para uma posterior descrição do que é o *Random Forest*, é preciso conceituar as árvores de decisão, um método de aprendizagem de máquina amplamente utilizado. As árvores de decisão podem ser compreendidas, de forma geral, como uma divisão sucessiva de um problema em vários subproblemas menores, com intuito de chegar a uma solução final por meio da análise de cada uma das soluções dos problemas menores. Mitchell (1997) explica a 'estrutura' de uma árvore de decisão da seguinte forma: uma determinada amostra é classificada encaminhando-se pela árvore desde a raiz até um nó da folha, de forma que cada um dos nós possui um teste sobre algum atributo da amostra em questão, e cada ramo que sai desse nó corresponde a um dos possíveis valores dessa amostra. A Figura 3.5.1 ilustra como seria uma árvore de decisão.

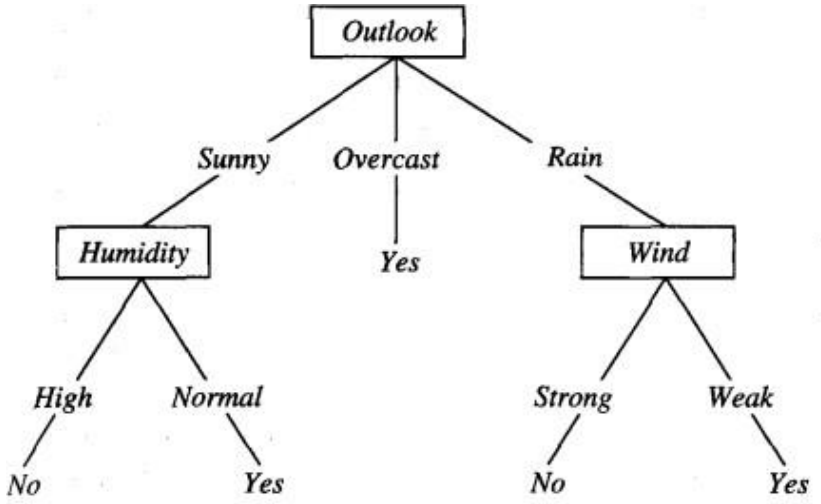

Figura 3.3 - Modelo de uma árvore de decisão para o conceito *PlayTennis*. Um exemplo é classificado ao ser conduzido pela árvore até o nó folha apropriado, retornando, então, a classificação associada a essa folha (neste caso, Sim (*Yes*) ou Não (*No*)). Essa árvore classifica manhãs de sábado de acordo com serem ou não adequadas para jogar tênis.

Fonte: Mitchell (1997).

Na estrutura apresentada, o nó raiz é representado por *Outlook*, que corresponde ao primeiro teste realizado pela árvore no processo de classificação. Os demais nós, *Humidity* e *Wind*, são nós internos, representando os testes das variáveis, e nesse sentido, Outlook, *Humidity* e *Wind* correspondem aos atributos. Os ramos são representados por *Sunny*, *Overcast, Rain, High, Normal*, *Strong* e *Weak*, que são os possíveis valores que os atributos podem assumir. Por fim, os termos *Yes* e *No* são os nós folha, ou seja, os resultados da classificação.

3.5.1.2 Aplicação do Random Forest

Segundo Breiman (2001), o *Random Forest* (ou Florestas Aleatórias) é um método de aprendizado de máquina *ensemble*, baseado na combinação de múltiplas árvores de decisão com o objetivo de aumentar a precisão da classificação, por meio da junção de diversos classificadores que irão definir a resposta final a partir da votação da classe mais frequente. Essa abordagem permite construir um modelo mais robusto, produzindo melhorias significativas no desempenho preditivo.

A ideia central do método é que, para cada *k*-ésima árvore, seja gerado um vetor aleatório $\theta_{k,}$ que é independente dos vetores anteriores ($\theta_{1,}$ ..., $\theta_{k-1,}$), porém segue a mesma distribuição. Em seguida, a árvore é construída com base no conjunto de treinamento e nesse vetor aleatório ($\theta_{k,}$) resultando em um classificador $h\ (x,\ \theta_k)$, em que x representa o vetor de entrada (Breiman, 2001). Depois que um grande número de árvores de decisão é gerado, os classificadores $h\ (x,\ \theta_k)$ votam entre si e a classe mais popular é adotada como o resultado final.

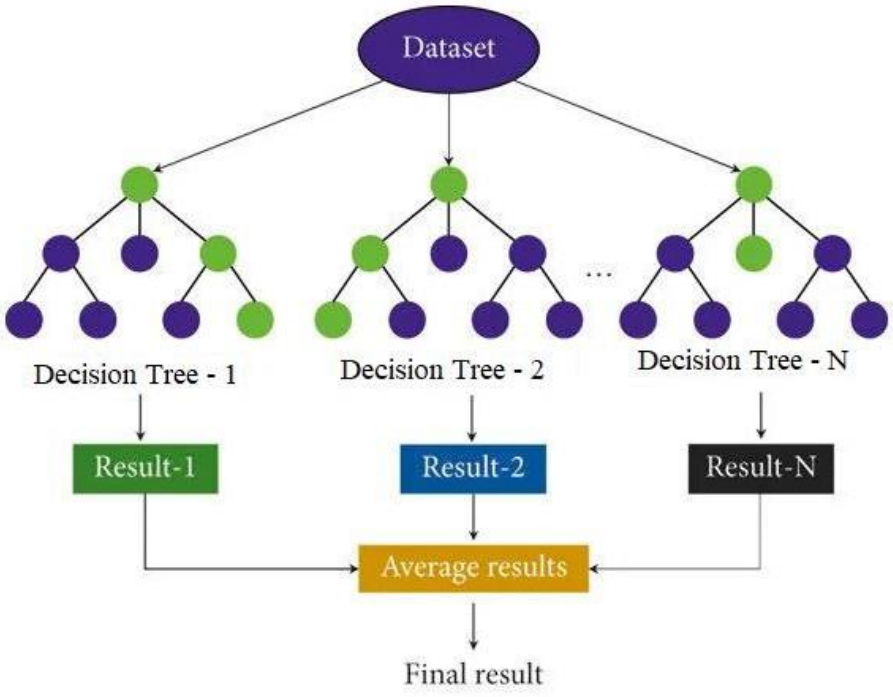

Figura 3.4 - Representação do método *Random Forest.*
Fonte: Adaptado de FU & QI (2022).

A estrutura apresentada representa, de forma esquemática, o funcionamento do método *Random Forest*. Inicialmente, parte de um conjunto de dados (nesse caso, denominado *Dataset*), a partir do qual são gerados múltiplos modelos base. Cada um desses modelos produz

um resultado próprio, indicado na Figura 3.3 como *Result-1, Result-2* e *Result-N*. Em seguida, esses resultados individuais são combinados em uma etapa de agregação, representada por *Average results*, da qual se obtém o resultado final. Assim, a lógica no método não está centrada em uma única árvore de decisão, mas sim na integração de várias predições, o que permite uma maior estabilidade e melhor capacidade de generalização do modelo.

3.5.2 Redes neurais artificiais aplicadas

As redes neurais artificiais (RNAs) tiveram seu primeiro modelo criado por Warren McCulloch, um psiquiatra e neuroanatomista, e Walter Pitts, um matemático recém graduado, no ano de 1943. Inicialmente, o trabalho de McCulloch e Pitts tinha foco principal em descrever e apresentar um modelo de neurônio artificial e suas capacidades computacionais. A implementação de técnicas de aprendizagem de redes biológicas e Artificiais veio mais tarde, em 1949, com Donald Hebb, com uma teoria que explicava o aprendizado em nodos biológicos baseada no reforço das ligações sinápticas entre nodos excitados (Braga, 2000). A partir daí vários estudiosos continuaram a acrescentar novas teorias e ramificações do que hoje conhecemos com uma rede neural artificial.

As RNAs podem ser definidas como sistemas paralelo distribuídos, compostos por unidades de processamento simples (conhecidos como nodos) que calculam determinadas funções matemáticas. Essas unidades são distribuídas na forma de camadas e interligadas entre si por um grande número de conexões associadas a pesos (ou parâmetros); os pesos carregam o conhecimento representado no modelo, funcionando como ponderador da entrada recebida por cada neurônio em uma rede. Segundo BRAGA (2000), as RNAs tentam reproduzir as funções das redes biológicas por meio da implementação de seu comportamento básico e sua dinâmica. Apesar de diferirem em muitos aspectos, possuem algumas similaridades, como por exemplo possuírem sistemas que são baseados em unidades de computação paralela e distribuída, que se comunicam por meio de conexões sinápticas, além de detetores de características, redundância e modularização das conexões.

3.5.2.1 Neurônio biológico x neurônio artificial

Os neurônios biológicos são divididos em três seções:

- dendritos: têm a função de receber as informações provenientes de outros neurônios e conduzi-las até o corpo celular;

- corpo da célula: onde a informação recebida e processada e novos impulsos são gerados;
- axônio: local onde os novos impulsos passam para chegar até o dendrito do próximo neurônio.

O ponto que conecta o axônio de um neurônio até o dendrito de outro é chamado de sinapse, onde os nodos se unem para formar as redes neurais. Na Figura 3.4 são representados, de forma simplificada, os componentes de um neurônio biológico.

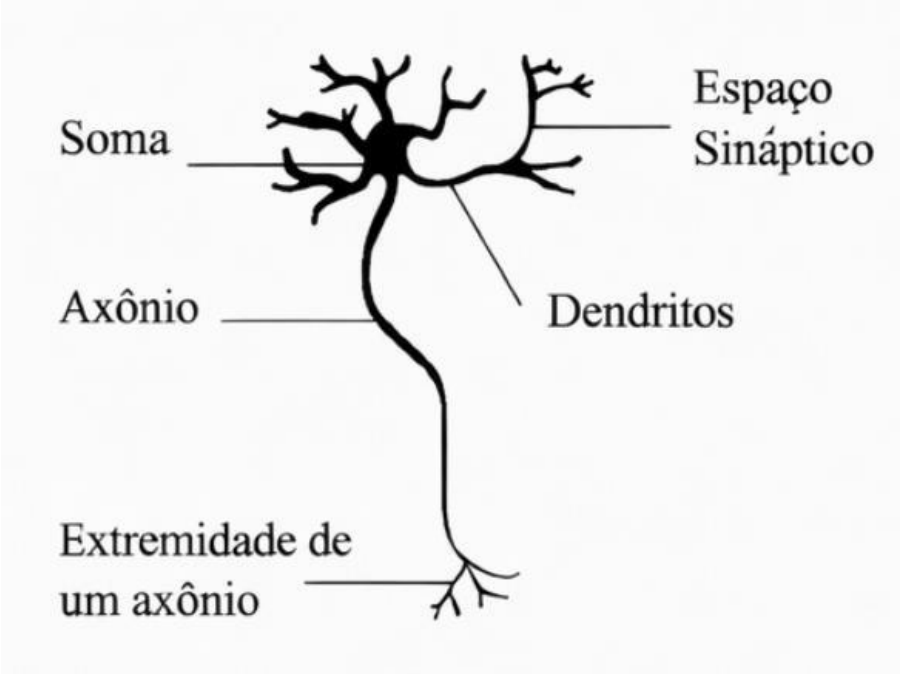

Figura 3.5 – Representação dos componentes de um neurônio biológico.
Fonte: BRAGA (2000)

O modelo de um neurônio artificial proposto por McCulloch e Pitts ficou conhecido como modelo MCP, uma simplificação do que se sabia sobre um neurônio biológico. Na Figura 3.5 encontra-se uma descrição desse modelo.

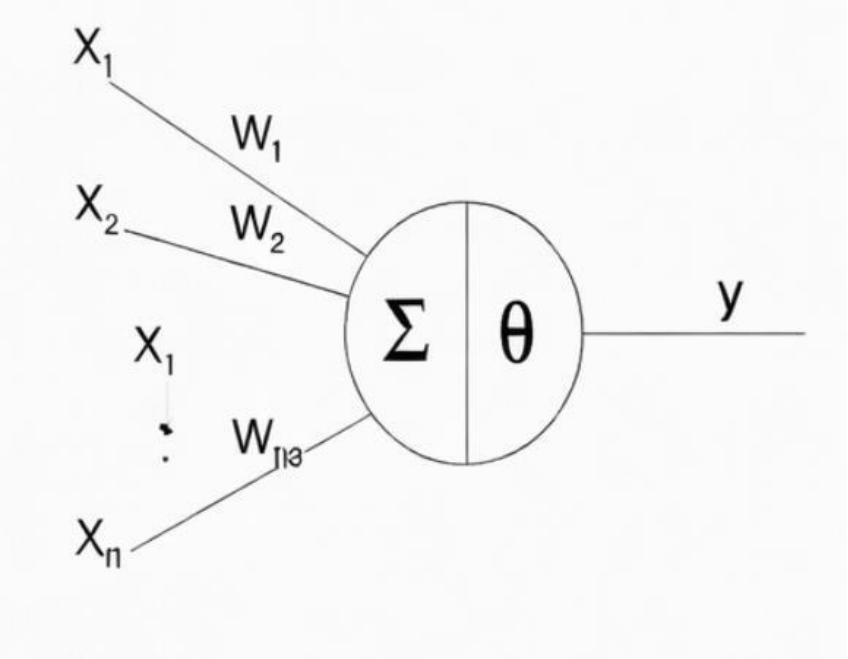

Figura 3.6 – Modelo de um neurônio artificial, conhecido como modelo MCP.
Fonte: BRAGA (2000)

Em sua representação matemática, existem n terminais de entrada $x_1, x_2, ..., x_n$, que representam os dendritos, e apenas um terminal de saída, que representa o axônio. Os terminais de entrada do neurônio têm pesos acoplados $w_1, w_2, ..., w_n$, para simular o comportamento das sinapses, podendo assumir valores positivos ou negativos, que determinam em que grau o neurônio deve considerar sinais de disparo que ocorrem naquela conexão. O efeito de uma sinapse particular *i* no neurônio pós-sináptico é dado por x_1w_1. Posteriormente, um mecanismo simples faz a soma de x_1w_1 recebidos pelo neurônio (é feito uma soma ponderada), que decide se o neurônio deve ou não disparar. Quem ativa ou não a saída dessa soma é a função de ativação, que toma essa decisão baseada no valor da soma ponderada de suas entradas (BRAGA, 2000). A função de ativação é dada pela função limiar, representada na Equação 18

$$\sum_{i=1}^{n} x_i w_i \geq \theta \quad (18)$$

3.5.2.2 Aprendizado

Segundo Braga (2000), para que seja possível utilizar as RNAs para solução de tarefas, elas devem passar por uma fase de aprendizado, a qual a rede extrai as informações de relevância dos dados de entrada e cria uma representação própria do problema a partir dos padrões encontrados. Existem diversos métodos de treinamento das redes, e eles podem ser agrupados em aprendizado supervisionado e aprendizado não supervisionado.

No aprendizado supervisionado, tanto a entrada quanto a saída são fornecidas por um supervisor (professor) externo, com o objetivo de ajustar os parâmetros da rede e encontrar uma ligação entre os pares de entrada e saída fornecidos. Já o aprendizado não-supervisionado não conta com um supervisor para acompanhar o processo de aprendizado e somente os padrões de entrada estão disponíveis para a rede. Assim como pontua Braga (2000), este tipo de aprendizado só é possível quando existe redundância dos dados, para que seja possível encontrar padrões ou características por meio dos dados de entrada.

3.6 Aplicações de inteligência artificial na geoestatística

O uso de técnicas de inteligência artificial na geoestatística vem sendo utilizado de forma cada vez mais crescente, em estudos de estimativas, classificação, definição de domínios, dentre outros, permitindo identificar padrões nos dados para o auxílio em etapas de modelagem espacial. A Tabela 3.1 a seguir, apresenta alguns trabalhos que exemplificam esse uso, destacando os autores, as técnicas aplicadas e os objetivos de cada estudo.

Tabela 3.1 – Aplicações de inteligência artificial na geoestatística
Fonte: Autora.

Autor	Técnica utilizada	Objetivo do estudo
OLEQUES, 2024	K-Means	Definir domínios geoestatísticos estacionários.
SIQUEIRA, 2024	SVM	Prever parâmetros geotécnicos próximos.
CINTRA, 2003	Rede neural artificial (MLP)	Estimar teores de cobre-ouro.
SILVA, 2019	Random Forest	Estimar teores em depósito.
CARVALHO, 2022	K-Means, HC e GHC	Definir domínios de estimativa.
CARVALHO, 2022	Fuzzy k-Prototype (FCP)	Delinear domínios quase-estacionários.
AYACHE et al., 2023	Self-Organizing Maps (SOM)	Aplicação do método SOM para desagrupamento amostral.

4. METODOLOGIA

A metodologia utilizada no presente trabalho consistiu em seis etapas, descritas a seguir: i) área de estudo e dados primários; ii) ambiente computacional; iii) análise exploratória; iv) modelagem da média local; v) análise geoestatística; vi) validação cruzada.

4.1 Área de estudo e dados primários

Os dados utilizados neste trabalho são baseados em um conjunto de dados que foi derivado em um modelo digital de elevação do oeste dos Estados Unidos, a área de Walker Lake, em Nevada. Esse conjunto de dados é utilizado no livro *Applied Geoestatistics*, onde Isaaks & Srivastava (1989) o utilizam para a aplicação da Geoestatística em seus estudos e caso. Os valores utilizados não são os valores originais de elevação, mas sim variáveis que estão relacionados a eles. As variáveis utilizadas neste trabalho são contínuas, descritas por Isaaks & Srivastava (1989) como V e U.

No caso deste trabalho, a variável V, ou variável primária, representa a espessura mineralizada de um corpo, enquanto a variável U representa o teor de Au, como sugere Isaaks & Srivastava (1989): "*[...] as variáveis contínuas, V e U, podem representar espessuras de um horizonte geológico ou a concentração de algum poluente; podem representar medições de resistência do solo ou permeabilidade; podem representar medições de precipitação ou diâmetro de árvores [...]*".

A Figura 4.1 mostra como se dá a disposição espacial dos dados, com 470 pontos distribuídos em uma malha 250x300 metros.

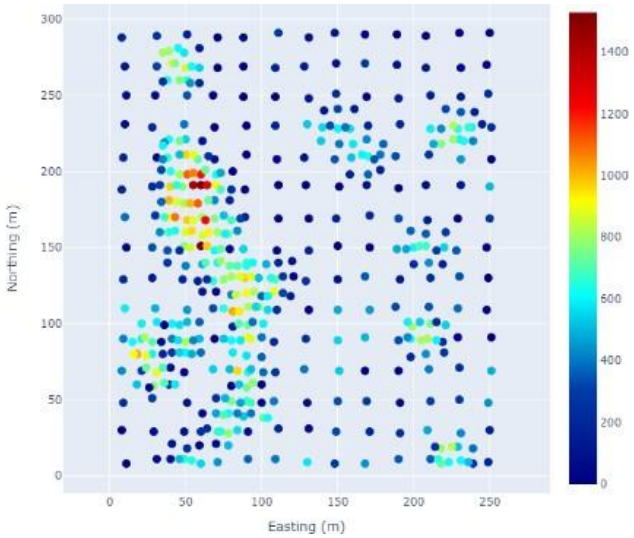

Figura 4.1 – Distribuição da variável V no Walker Lake.
Fonte: AYACHE ET AL (2023).

4.2 Ambiente computacional

O ambiente computacional adotado neste trabalho para execução dos experimentos e análise das relações entre os dados estudados foi o Google *Colaboratory* (conhecido também como Google *Colab*), um projeto de serviço de nuvem criado pela Google para promover o ensino e a pesquisa em aprendizado de máquina. Os *notebooks* do *Colab* são baseados no sistema Jupyter Notebook, uma ferramenta de código aberto baseada em navegador que integra linguagem de programação interpretada, bibliotecas e ferramentas de visualização, a qual cada documento é composto por múltiplas células e, cada uma delas, contém códigos em linguagem de *scripts* ou *mark-down* e as saídas são integradas ao documento, incluindo textos, tabelas, gráficos e imagens (Carneiro *et al.*, 2018).

Outro ponto de destaque desse ambiente, também segundo Carneiro et al. (2018), é que o Google *Colab* pode ser compartilhado e outras pessoas podem contribuir com os códigos criado no mesmo notebook, além de ser capaz de fornecer um ambiente de execução acelerado por GPU.

As análises foram desenvolvidas em Python, com o uso de bibliotecas voltadas ao tratamento de dados, modelagem e visualização gráfica. As bibliotecas *NumPy* e *pandas* foram utilizadas na leitura, organização e manipulação do banco de dados, permitindo estruturar as variáveis amostrais e realizar operações numéricas e tabulares ao longo do fluxo metodológico. A biblioteca *SciPy* foi empregada em rotinas auxiliares de cálculo estatístico e espacial, dando suporte a etapas específicas do processamento de dados. A modelagem da média local foi realizada com a biblioteca *scikit-learn*, por meio dos algoritmos de regressão linear, *Random*

Forest e redes neurais Artificiais, além de rotinas ligadas à validação dos modelos. Já a biblioteca *matplotlib* foi utilizada para a construção dos gráficos empregados na análise exploratória, nos histogramas e nas demais representações visuais dos resultados. Todo o código foi executado no ambiente do Google *Colab*, que permitiu a implementação do *script* em *notebook*, como dito anteriormente neste tópico.

4.3 Análise exploratória

4.3.1 Preparação prévia dos dados

Após a escolha dos dados e o ambiente computacional que integrariam o trabalho, o próximo passo consistiu em realizar uma análise exploratória dos dados estudados, com o intuito de entender e caracterizar a interação existente entre as variáveis escolhidas como primárias e secundárias. Como ponto de partida, os dados foram importados no ambiente *Colab*, passando pequenos ajustes, como é o caso do nome das variáveis, por exemplo, para uma compreensão mais clara dos dados apresentados, de forma que fossem posteriormente manipulados e submetidos as análises estatísticas.

Para a análise exploratória espacial da variável secundária (espessura mineralizada), foram construídas seções verticais ao longo do eixo X e Y, com o intuito de avaliar a variabilidade direcional da espessura no contexto estudado, logo, essa análise foi feita da seguinte forma: nas seções do tipo X, fixou-se um valor específico de X, permitindo uma tolerância de ±0,50 metros para a seleção dos pontos, formando uma faixa vertical de análise e, dentro dessa faixa, avaliou-se a variação da espessura ao longo do eixo Y. De maneira análoga, nas seções do tipo Y, um valor específico de Y também foi fixado e aplicou-se a mesma faixa de tolerância para seleção dos pontos, sendo a análise da espessura feita ao longo do eixo X. Nesta fase, foi possível observar o comportamento da variável de modo a entender como ela se apresenta em diferentes direções e se existe ou não padrões de continuidade espacial na sua distribuição.

Quanto a variável primária (teor de Au), foram calculadas medidas de dispersão com o objetivo de quantificar e compreender a variabilidade dos dados em torno de suas médias, visto que os estudos em torno de teores podem apresentar uma distribuição assimétrica devido aos processos geológicos complexos de formação dos corpos de minério e, por isso, exige uma caracterização prévia para apoiar as etapas posteriores de análise. O cálculo dessas medidas (média, mediana, desvio padrão e moda) foram feitas em classes de espessuras, como mostra a Tabela 4.1:

Class_Thickness	mean	median	std	count
(0.854, 2.737]	0.191019	0.13305	0.210748	54
(2.737, 3.659]	0.347785	0.25830	0.329970	53
(3.659, 4.318]	0.479585	0.32715	0.474109	54
(4.318, 5.295]	0.674781	0.47380	0.580265	53
(5.295, 11.171]	1.402106	1.31355	1.175080	54

Figura 4.2 – Estatísticas descritivas do teor de Au por classes de espessura.
Fonte: Autora.

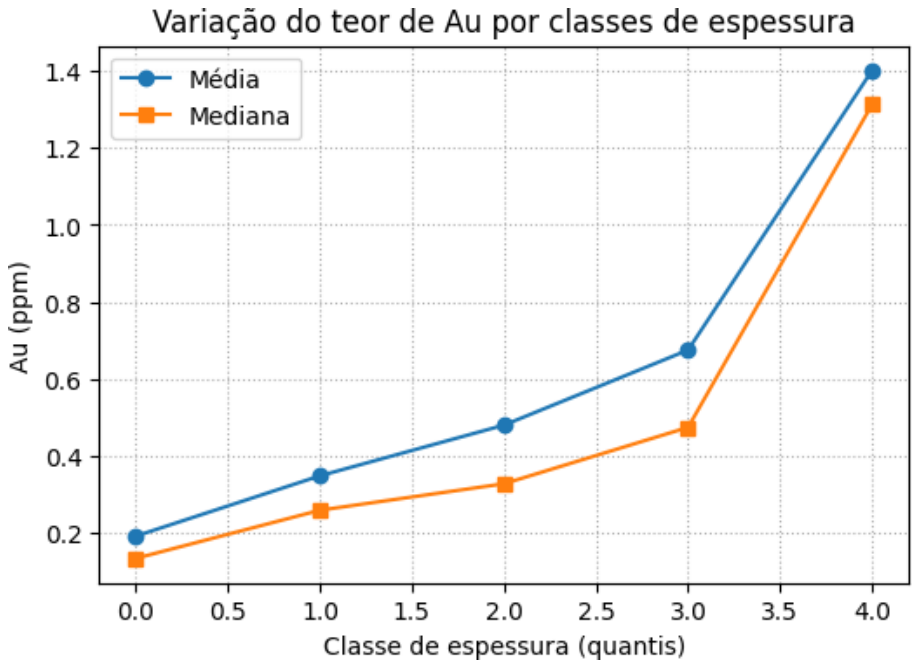

Figura 4.3 – Gráfico da média e mediana dos teores de Au por classes de espessura.
Fonte: Autora.

4.3.2 Método de correlação de Pearson e Spearman

Além das métricas citadas anteriormente, foram utilizados dois métodos de correlação para analisar a relação entre as variáveis: método de correlação linear de Pearson e método de correlação monótona de Spearman.

O método de Pearson mede a relação linear entre duas variáveis contínuas, ou seja, o quanto uma mudança em uma variável pode estar associada a uma mudança proporcional em outra variável. Seus valores podem variar de -1 a +1, inferindo que:

- Quando há uma relação de interação consistente entre as variáveis, de forma crescente (quando há o aumento de ambas as variáveis) ou decrecente (diminuição de ambas as variáveis), o seu valor é igual a +1 ou -1, respectivamente;
- Quando há um aumento de uma variável e a outra variável também aumenta, porém não de forma consistente, o seu valor é positivo, mas menor que +1;
- Quando a relação é aleatória ou inexistente, seu valor é próximo ou igual a 0;
- Se à medida que uma variável aumenta a outra variável diminui, porém de forma inconsistente, o seu valor é negativo, mas maior que -1.

O método de correlação de Spearmam analisa a relação monotônica entre duas variáveis contínuas ou ordinárias, ou seja, o quanto duas variáveis podem variar juntas, mas não a uma taxa constante. Os seus valores também variam de -1 a +1, da seguinte forma:

- Quando há uma relação de interação consistente entre as variáveis, de forma crescente (quando há o aumento de ambas as variáveis) ou decrecente (diminuição de ambas as variáveis), o seu valor é igual a +1 ou -1, respectivamente;
- Quando há um aumento de uma variável e a outra variável também aumenta, porém não de forma consistente, o seu valor também é igual +1;
- Quando a relação é aleatória ou inexistente, seu valor é próximo ou igual a 0;
- Se à medida que uma variável aumenta a outra variável diminui, porém de forma não linear, também é igual a -1.

4.4 Hiperparâmetros das técnicas de inteligência artificial

No processo de modelagem por técnicas de inteligência artificial, foram definidos hiperparâmetros específicos para cada modelo. Para o Random Forest, foram utilizados: número de árvores igual a 400, número mínimo de amostras por folha igual a 5 e semente aleatória para reprodutibilidade igual a 42. Para a rede neural artificial, foram definidas duas camadas com 20 neurônios cada, parâmetro de regularização igual a 10^{-3} e número máximo de iterações igual a 5000. Esses parâmetros foram ajustados conforme a etapa de calibração, buscando um equilíbrio entre capacidade de ajuste e generalização dos modelos.

4.5 Modelagem da média local e análise geoestatística

A seguir, são representadas, por meio de um fluxograma (Figura 4.4), as etapas que compõem a metodologia do trabalho:

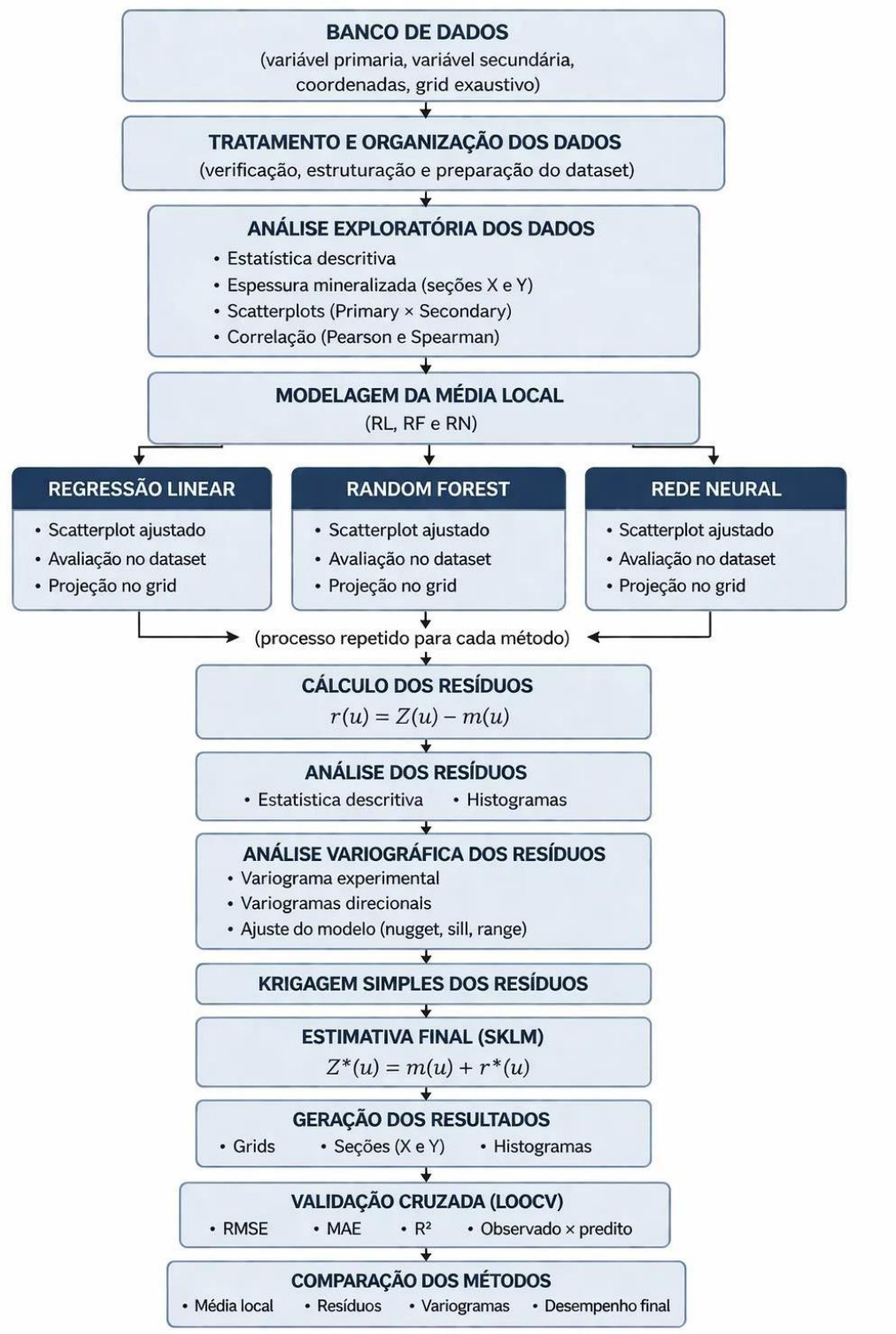

Figura 4.4 – Fluxograma metodológico.
Fonte: Autora.

i) Organização do banco de dados, contendo a variável primária, variável secundária, as coordenadas espaciais e o *grid* exaustivo. Depois disso, os dados foram tratados e analisados de forma estatística;

ii) Análise do comportamento da espessura mineralizada por meio de seções espaciais, para compreender como elas se comportavam no espaço amostral;

iii) Avaliação da relação entre a variável primária e a variável secundária por meio dos coeficientes de correlação (Spearman e Pearson), para verificar o grau de correlação entre as variáveis;

iv) Modelagem da média local para as três abordagens: regressão linear, Random Forest e Redes Neurais Artificiais, em que, para cada um deles, foi avaliado o ajuste no *scatterplot,* a distribuição no *dataset* e a projeção no *grid*;

v) Cálculo dos resíduos por meio da diferença entre os valores observados e estimados pela média local, seguido pela análise dos mesmos por estatística descritiva e histogramas;

vi) Aplicação da krigagem nos resíduos com base nos modelos variográficos ajustados;

vii) Estimativa final obtida pela soma entre a média local e os resíduos krigados, com o intuito de estimar a variável primária levando em consideração a estrutura espacial. Foram gerados grids, histogramas e seções dessa estimativa para melhor visualização dos métodos;

viii) Aplicação da validação cruzada LOOCV, com métricas RMSE, MAE e R^2, para quantificar o desempenho dos métodos;

ix) Comparação dos métodos considerando média local, resíduos, variabilidade espacial e desempenho preditivo, com o auxílio de gráficos, tabelas e histogramas.

5. RESULTADOS E DISCUSSÕES

A seguir são descritos todos os resultados e parâmetros obtidos no presente trabalho. Por meio de análises estatísticas e geoestatísticas foi possível validar a utilização dos modelos não lineares (*Random Forest* e redes neurais artificiais) para estimativa de depósitos por meio da aplicação da SKLM.

5.1 Preparação dos dados e estatística descritiva

Os dados utilizados neste trabalho estão disponibilizados em uma tabela no Apêndice A, contendo, cada um deles, uma coordenada X e uma coordenada Y, seu teor de Au (variável primária) e a espessura da zona mineralizada (variável secundária).

5.1.1 Análise das seções de espessura mineralizada

Depois da preparação dos dados, foi feita a análise de como os pontos estavam distribuídos no espaço, definindo-se 5 seções para o eixo X e 5 seções para o eixo Y, buscando compreender como a espessura do corpo mineralizado se distribui no espaço amostral. Na Figura 5.1 são mostradas as seções:

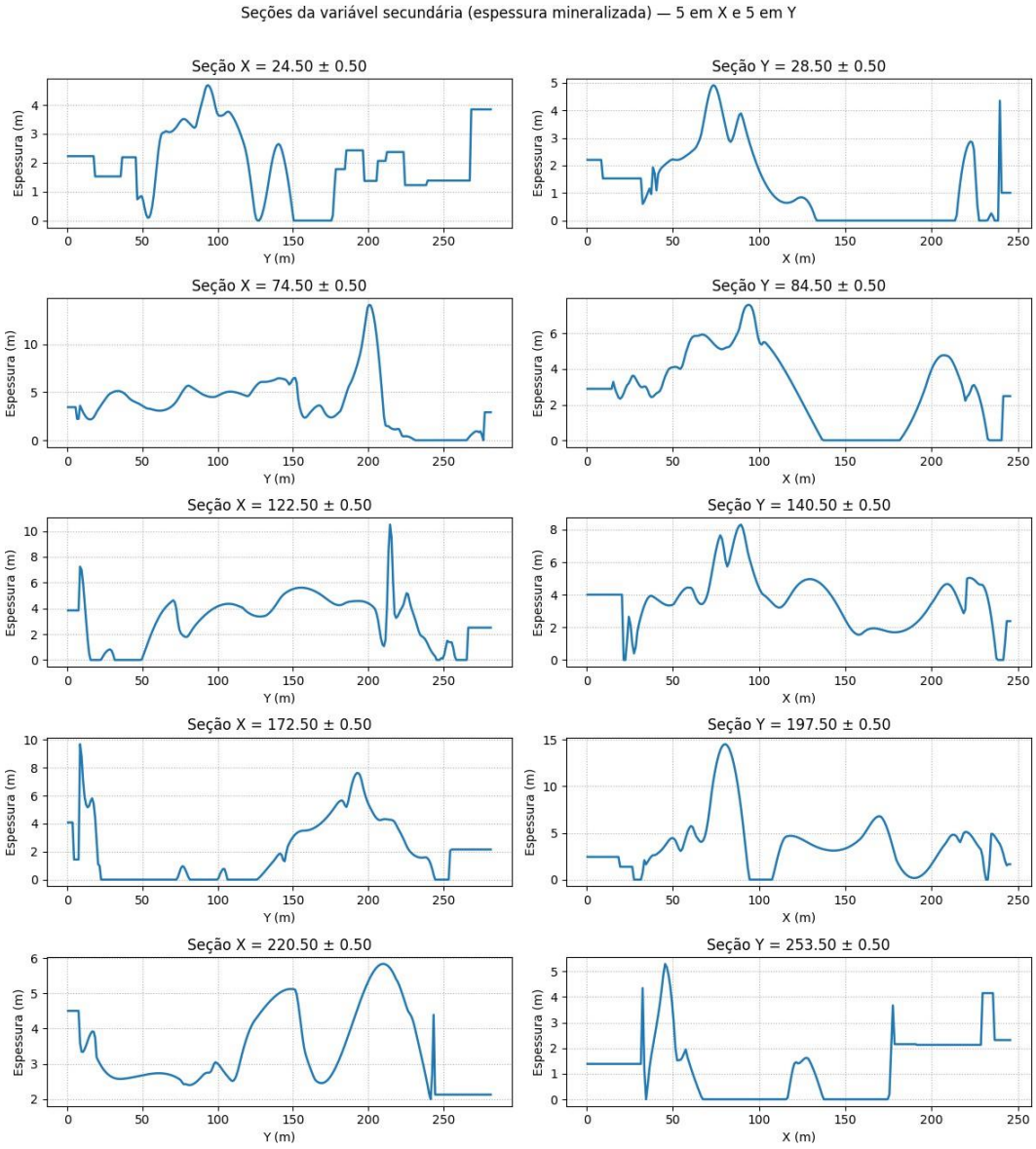

Figura 5.1 – Seções da variável secundária (espessura mineralizada).
Fonte: Autora.

A análise das seções da espessura mineralizada mostra que o corpo apresenta distribuição heterogênea, com trechos em que a espessura se mantém moderada e contínua e outros em que ocorrem aumentos expressivos ou reduções acentuadas. Esse comportamento é compatível com corpos mineralizados que não possuem geometria uniforme, variando em função da própria organização espacial da mineralização. Nas seções X = 74,5 ± 0,50 e Y = 197,50 ± 0,50, por exemplo, é possível notar um aumento expressivo da espessura, com espessuras na casa de 10 a 15 metros, o que pode sugerir a presença de zonas de maior desenvolvimento do corpo mineralizado, valores que não aparecem de forma isolada, mas sim inseridos em perfis que já apresentam elevação progressiva da espessura, indicando que essas seções apresentam áreas mais bem desenvolvidas da mineralização.

Em contrapartida, nota-se trechos em algumas seções em que a espessura se reduz significativamente, ficando muito próximos de zero antes de voltar a aumentar. Esse comportamento pode ser observado nas seções X = 28,50 ± 0,50 e X = 172,50 ± 0,50, nas quais há segmentos com espessuras muito baixas intercalados por zonas de retomada do corpo, padrão que sugere estreitamentos locais da mineralização e possível descontinuidade geométrica ao longo do perfil. Na seção Y = 253,50 ± 0,50, essa redução se torna ainda mais marcante, já que parte significativa do perfil apresenta espessura reduzida, o que possivelmente indica proximidade com limites laterais da mineralização.

Há, ainda, as seções que apresentam um comportamento mais contínuo, com variações graduais da espessura ao longo do perfil, como é o caso da seção Y = 140,50 ± 0,50, em que as espessuras permanecem predominantemente entre dois e oito metros, oscilando de forma um pouco mais sutil ao longo da distância analisada, se comparada às demais seções. De forma semelhante, a seção X = 122,50 ± 0,50, apesar de exibir um pico de localizado, possui espessuras que se mantém em uma faixa de comportamento regular no trecho compreendido entre, aproximadamente, 50 e 200 metros. Essas seções são importantes porque mostram que, entre os extremos de espessuras muito altas e muito baixas, existe uma parcela significativa do corpo com comportamento mais homogêneo, contribuindo para a coerência geométrica do modelo.

5.1.2 Coeficientes de correlação

O coeficiente de Pearson apresentou valor de 0,678, com $p = 1,69 \times 10^{-37}$, enquanto o coeficiente de Spearman foi 0,511, com $p = 3,34 \times 10^{-19}$ (como mostra a Tabela 5.1). Esses resultados mostram que, à medida que a variável secundária aumenta, a variável primária tende também a apresentar valores mais elevados. O valor de Pearson sugere uma associação linear moderada a forte, indicando que a variável secundária possui potencial para explicar parte da variação média da variável primária. Já o coeficiente de Spearman, embora também positivo e significativo, apresenta valor inferior, o que sugere que essa relação não se mantém com a mesma intensidade em toda ordenação dos dados.

Tabela 5.1 – Resultado dos coeficientes de correlação.
Fonte: Autora.

	r/ρ	p
Coeficiente de Pearson	0,678	1,69e-37
Coeficiente de Spearman	0,511	3,34e-19

5.2 Modelagem da média local

5.2.1 Modelo de regressão linear

Na regressão linear, a relação entre as variáveis é descrita por uma única reta crescente, o que implica uma resposta gradual da média local em função da variável secundária. No *scatterplot*, a reta acompanha a tendência geral de aumento do teor de Au com a espessura mineralizada, mas não representa adequadamente a curvatura da nuvem de pontos, sobretudo nos valores acima de 4ppm. Nessa faixa, parte dos dados observados passa a se distribuir acima da reta, como mostra a Figura 5.2, sugerindo que a relação entre as variáveis se intensifica em níveis mais altos da variável secundária, comportamento que não captado adequadamente pela formulação linear.

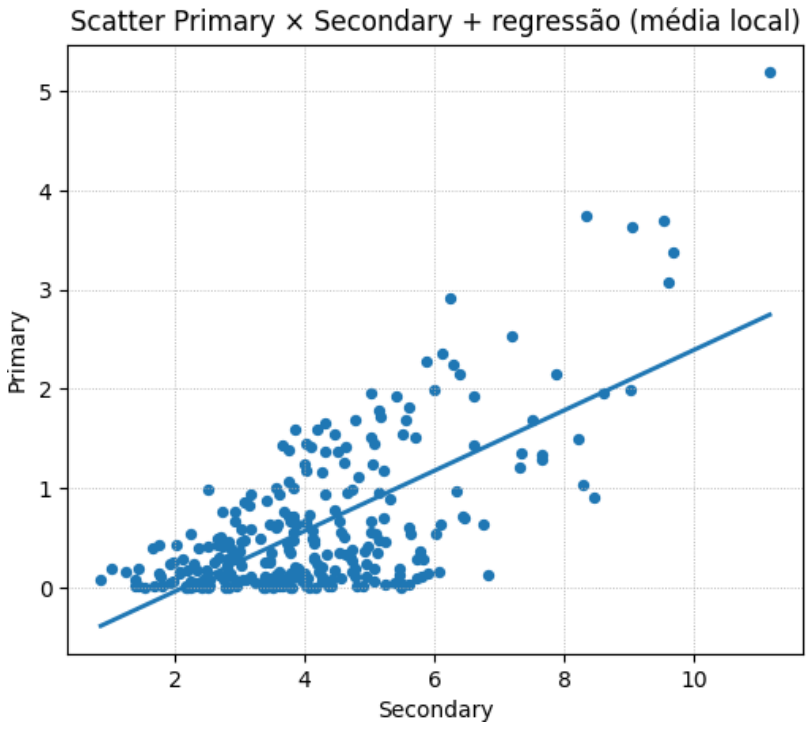

Figura 5.2 – Scatterplot da média local.
Fonte: Autora

A equação que demonstra a relação entre a variável primária (teor de Au) e secundária (espessura do corpo mineralizado) foi P = -0,6458 + 0,3037xS, com um coeficiente de correlação r = 0,678.

Nos pontos do *dataset*, essa limitação se reflete em uma distribuição espacial suavizada da média local, com os maiores valores estimados concentrando-se na porção central do conjunto amostral (na faixa compreendida, aproximadamente, entre X ≈ 75-110 e Y ≈ 180-210)

onde aparecem os pontos com coloração que indica valores mais elevados. Ainda assim, mesmo nessa região, a transição entre valores baixos, médios e altos ocorre de forma bastante gradual, comportamento que indica que, embora a regressão linear identifique a região de maiores valores, ela espalha essa elevação de forma mais suave pelos pontos ao redor, fazendo com que a diferença entre as zonas mais altas e as áreas vizinhas fique menos evidenciada.

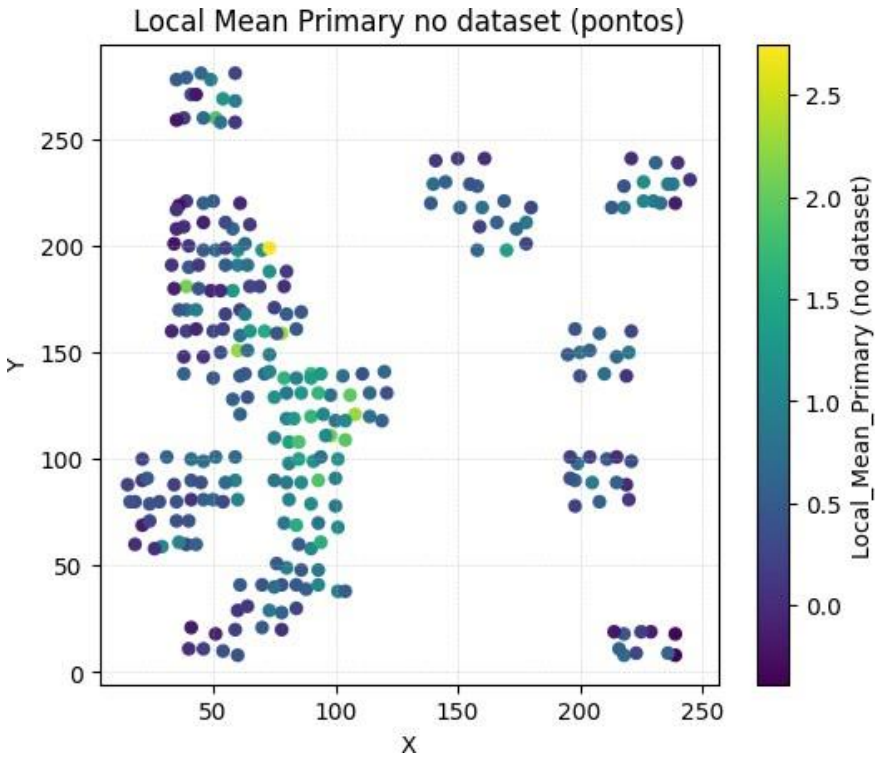

Figura 5.3 – Dataset da média local.
Fonte: Autora.

No grid exaustivo, é possível perceber essa tendência. A superfície de média local apresenta uma anomalia principal alongada na mesma região do *dataset* (na faixa compreendida, aproximadamente, entre X ≈ 75-110 e Y≈180-210), acompanhada de valores intermediários em seu entorno. Além disso, há também a presença de feições suaves distribuídas por outras partes do domínio, o que indica que a solução linear transfere para o espaço uma tendência contínua, sem compartimentação acentuada, comportamento que vai de encontro a construção da média local a partir de uma função única da variável secundária, que tende a representar melhor o padrão médio do que os contrastes locais.

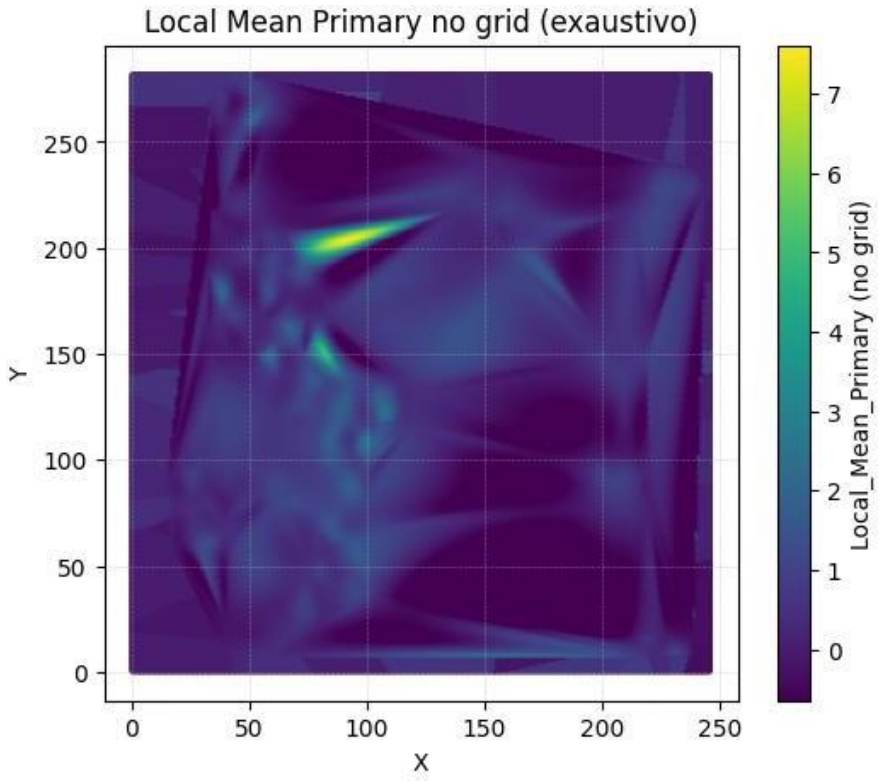

Figura 5.4 – Grid da média local.
Fonte: Autora.

5.2.2 Modelagem da média local por *Random Forest*

Na abordagem com *Random Forest*, a média local passa a refletir melhor as mudanças locais da relação entre as variáveis primárias e secundárias. No *scatterplot*, a curva ajustada deixa de ser uma reta e passa a assumir um formato escalonado, acompanhando de maneira mais próxima a nuvem de pontos, como na faixa da espessura compreendida entre 5 e 9, aproximadamente, faixa em que a dispersão de dados se torna mais ampla. A resposta do *Random Forest* mostra que o aumento da variável primária não ocorre a uma taxa constante, mas por patamares de comportamento, permitindo ao modelo absorver melhor a não linearidade observada na relação entre as variáveis.

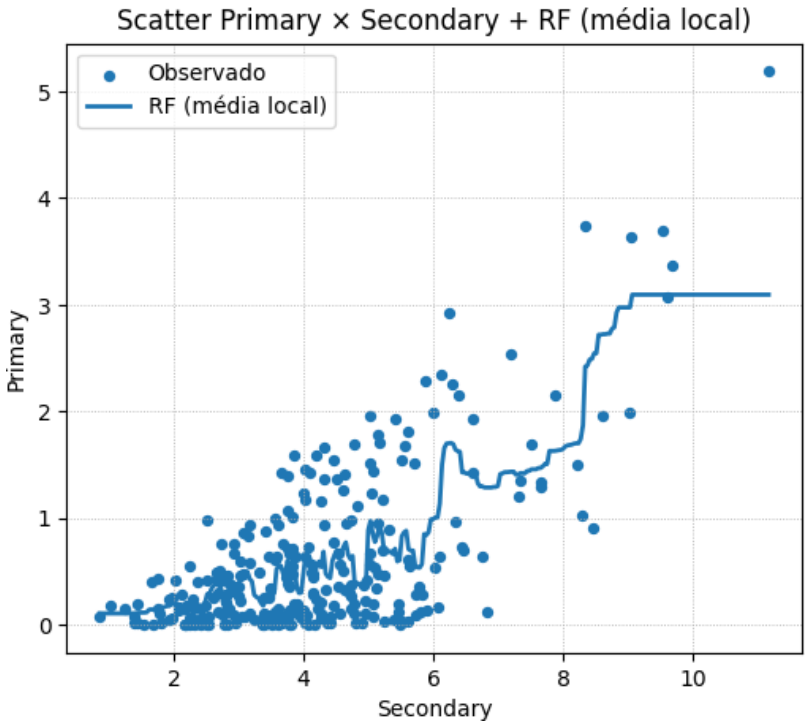

Figura 5.5 – Scatterplot da média local com Random Forest.
Fonte: Autora.

No *dataset* do modelo, essa maior adaptabilidade produz uma distribuição espacial mais contrastada da média local, com os valores mais altos concentrando-se em dois conjuntos principais: nas regiões X ≈ 75-110 e Y≈110-140 e X ≈ 60-80 e Y≈160-190. Enquanto isso, os valores intermediários se encontram na faixa X ≈ 75-110 e Y≈180-210, e os valores baixos na faixa X ≈ 10-50 e Y≈50-100. Isso mostra que o método reconhece a zona de maior valor e também diferencia melhor os núcleos do conjunto amostral, estabelecendo separações mais nítidas entre zonas de mais baixas a mais altas

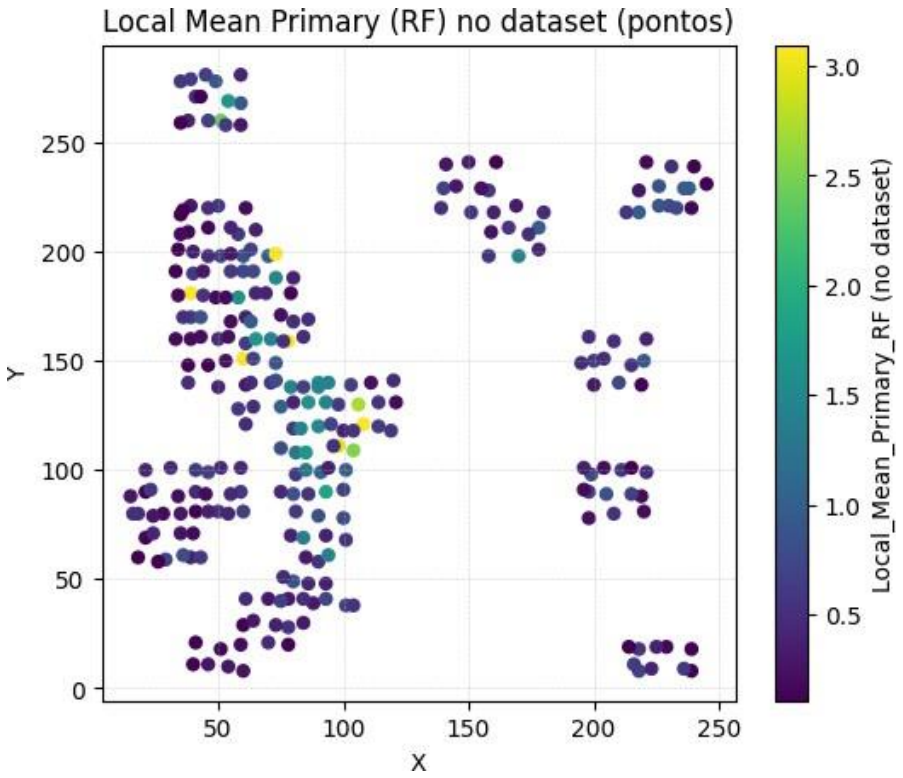

Figura 5.6 – Dataset da média local com Random Forest.
Fonte: Autora.

No grid, a superfície estimada apresenta uma anomalia central bem definida, na mesma região observada na regressão linear. Além dela, há a presença de algumas manchas secundárias mais localizadas na parte central e nas bordas do domínio. Em comparação com a superfície linear, o *grid* do *Random Forest* é menos difuso e mais segmentado, mostrando que as áreas de maior valor são menores, mais concentradas e mais separadas entre si. Esse comportamento indica que o RF transfere para a média local uma parcela maior da heterogeneidade espacial, como já visto nos dados amostrais.

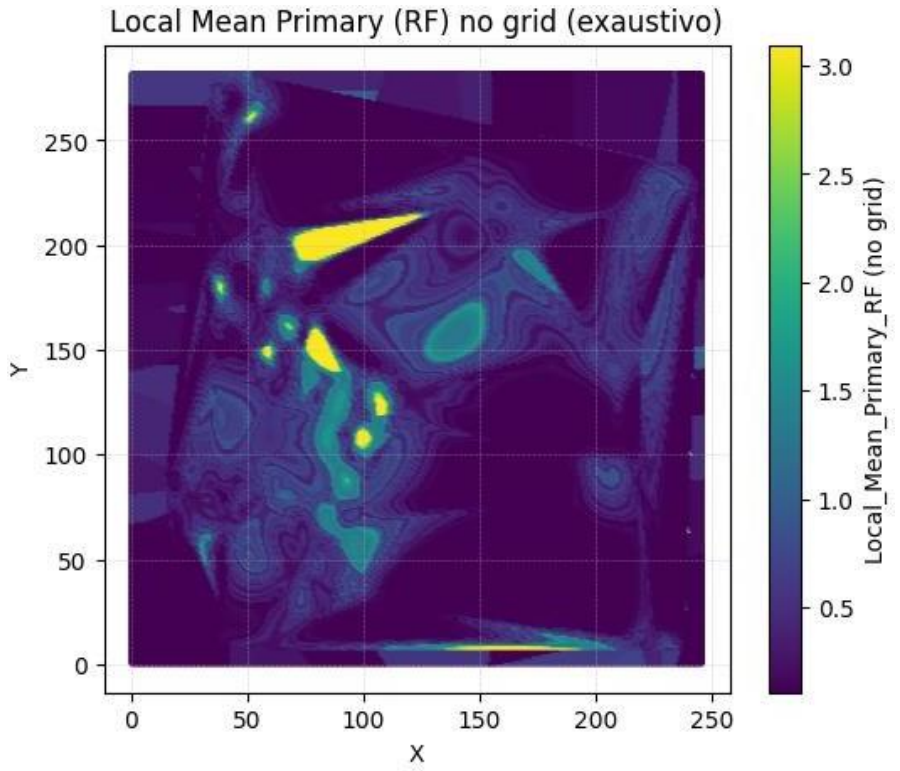

Figura 5.7 – Grid da média local com Random Forest.
Fonte: Autora.

5.2.3 Modelagem da média local por Redes Neurais

Na modelagem com Redes Neurais Artificiais, a média local assume também uma forma não linear, mas com maior continuidade espacial do que o visto na utilização do RF. No *scatterplot* observa-se uma curva suave e crescente, que acompanha a nuvem de pontos de maneira mais progressiva. Na faixa entre 1 e 5 da espessura, a resposta do modelo permanece baixa, e, a partir desse ponto, a curva cresce de forma mais acelerada, que é compatível com a concentração de valores mais altos de Au (variável primária) nas faixas superiores de espessura (variável secundária). As redes neurais conseguem não apenas incorporar a curvatura da relação entre as variáveis, mas também gerar patamares menos abruptos.

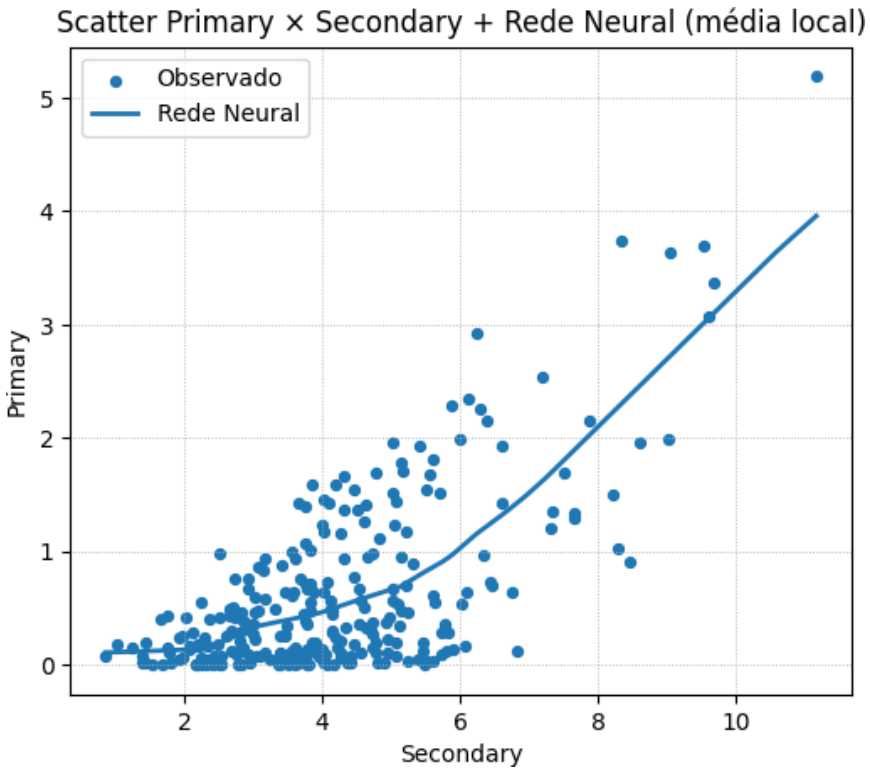

Figura 5.8 – Scatterplot da média local com Redes Neurais Artificiais.
Fonte: Autora.

No *dataset*, a consequência é uma distribuição espacial mais seletiva da média local, com o núcleo de altos valores em uma faixa ainda menor, X ≈ 75-95 e Y≈195-210, de forma mais concentrada. Fora dessa região, predominam valores baixos e moderados, com alguns pontos de valores intermediários na faixa compreendida entre X ≈ 75-100 e Y≈100-140. Isso indica que as RNAs respondem de maneira mais restritiva aos trechos em que a variável secundária realmente sugere maior valor esperado da variável primária, evitando espalhamento excessivo da média local por áreas amostrais próximas.

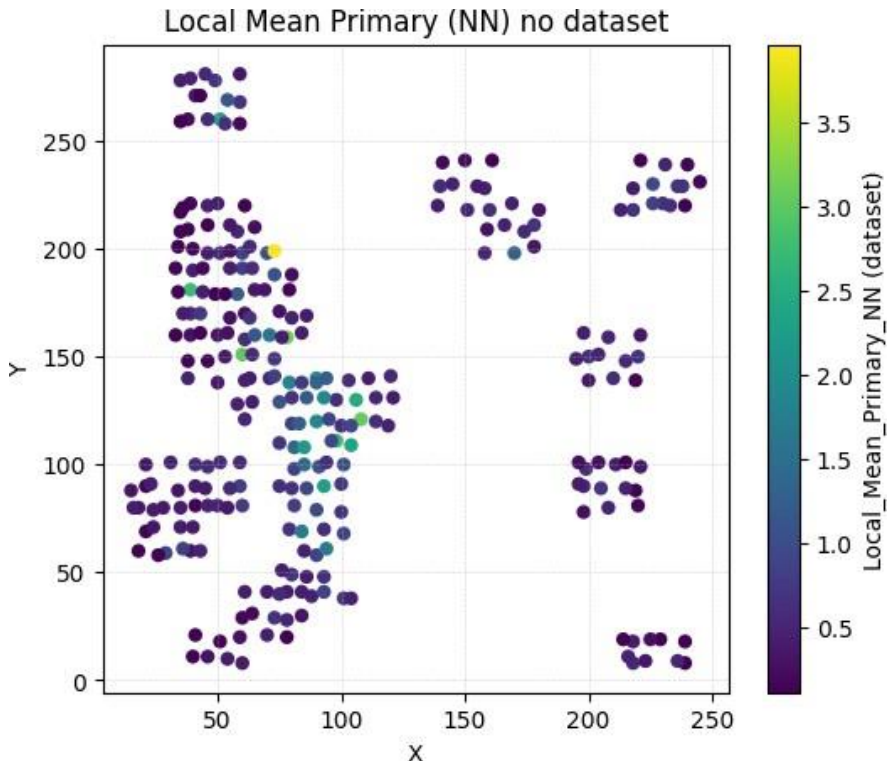

Figura 5.9 – Dataset da média local com Redes Neurais Artificiais.
Fonte: Autora.

A partir da análise do grid, percebe-se que a superfície das RNAs reserva a anomalia principal com forte destaque visual, mas a área de influência dessa anomalia é menor e mais concentrada do que aquelas vistas nos demais métodos. O restante do domínio, assim como observado no *dataset*, permanece com menores valores da média local, com algumas feições secundárias espalhadas em faixas de X ≈ 75-90 e Y≈140-160. A média calculada pelas RNAs tende a preservar a continuidade espacial de forma a evitar soluções excessivamente espalhadas.

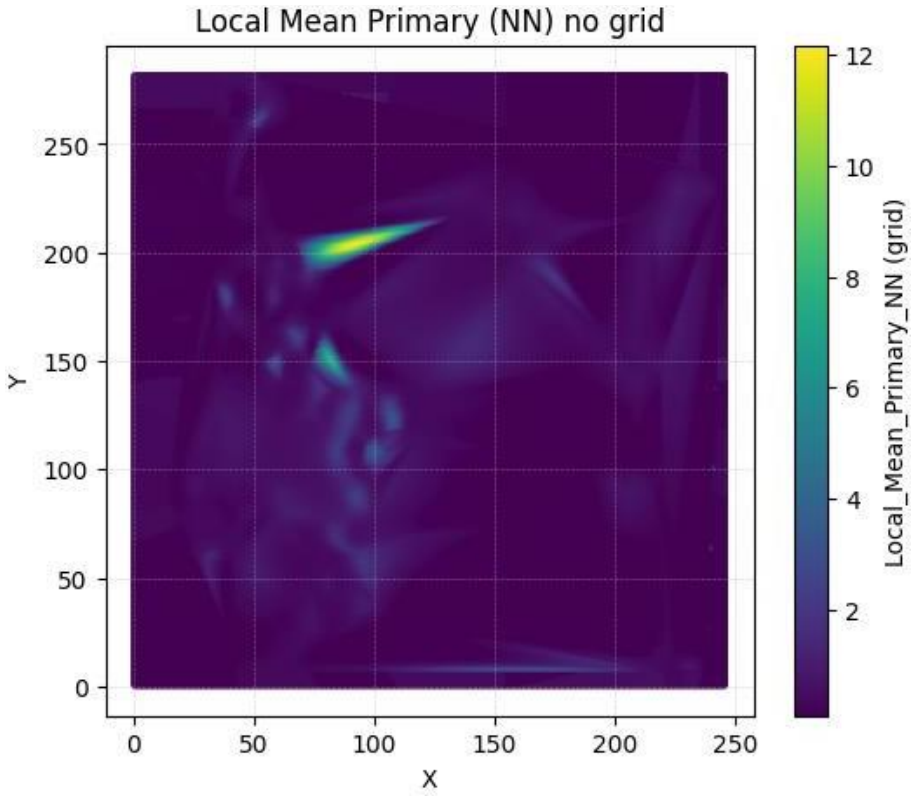

Figura 5.10 – Grid da média local com Redes Neurais Artificiais.
Fonte: Autora.

5.3 Krigagem dos resíduos

5.3.1 Cálculo dos resíduos

Os resíduos foram calculados como a diferença entre os valores observados da variável primária e os valores estimados pela média local em cada método. Esperava-se que os resíduos apresentassem média próxima de zero, sem uma distribuição pronunciada, representando a parcela que não foi explicada após a modelagem da média local.

No método de regressão linear, o histograma apresenta uma distribuição ampla, com concentração principal próxima de zero, mas com assimetria positiva perceptível. Isso indica que, embora a média residual seja praticamente nula, os resíduos não se distribuem de forma equilibrada em torno do centro, comportamento evidenciado por uma dispersão mais longa à direita, alcançando valores mais altos, o que significa a permanência de casos em que os valores observados da variável primária ficaram acima daqueles previstos pela média local linear. O histograma confirma que a solução linear deixou uma parcela maior da variabilidade concentrada nos resíduos, principalmente associada aos teores mais elevados.

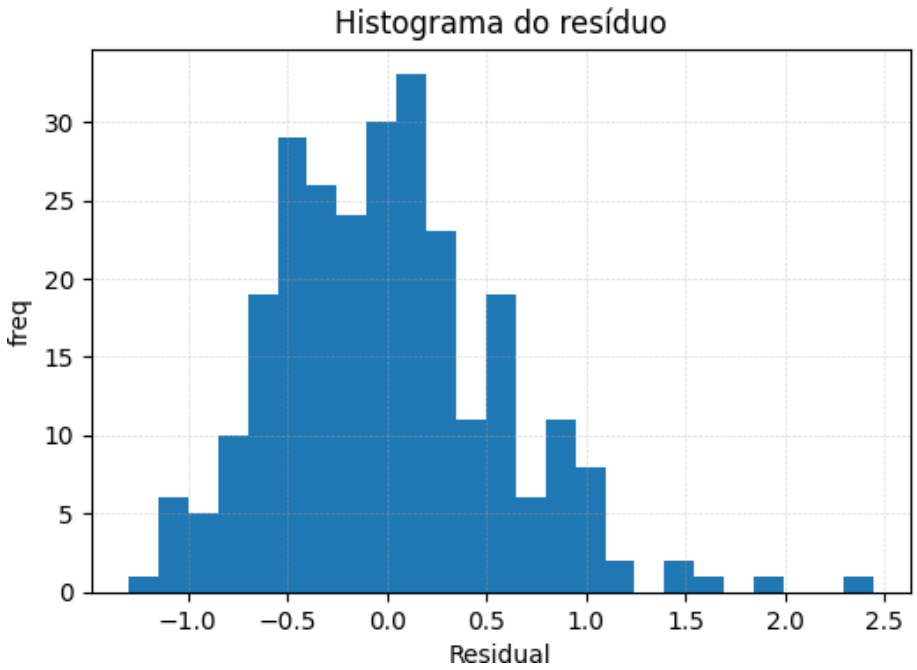

Figura 5.11 – Histograma do resíduo com regressão linear.
Fonte: Autora.

Com relação ao *Random Forest*, o histograma mostra uma distribuição mais concentrada em torno de zero, com menor espalhamento e maior acúmulo de frequências nas classes centrais. Ainda há assimetria positiva, mas de forma menos marcada que a regressão linear, mostrando que a média local estimada pelo RF conseguiu absorver melhor a estrutura sistemática dos dados, reduzindo a variabilidade remanescente que precisou ser tratada como resíduo. Como foi observado tanto no *dataset* quanto o *grid*, o RF modelou de forma bastante aderente a não linearidade entre a variável primária e secundária, resultando em menores resíduos para grande parcela dos dados.

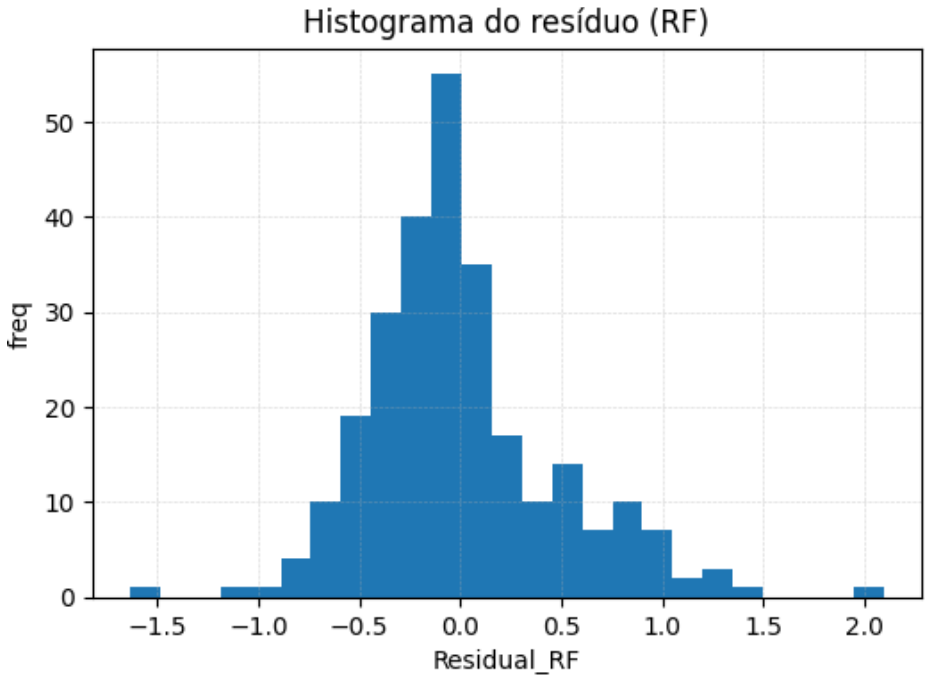

Figura 5.12 - Histograma do resíduo com Random Forest.
Fonte: Autora.

No caso das redes neurais artificiais, o histograma também apresenta resíduos centralizados em zero, mas com distribuição mais alongada, principalmente à direita. A forma observada mostra assimetria positiva, indicando que as RNAs conseguiram ajustar a tendência principal da variável em boa parte do domínio, mas ainda assim houveram casos em que os valores observados superaram de forma mais acentuada os valores estimados pela média local.

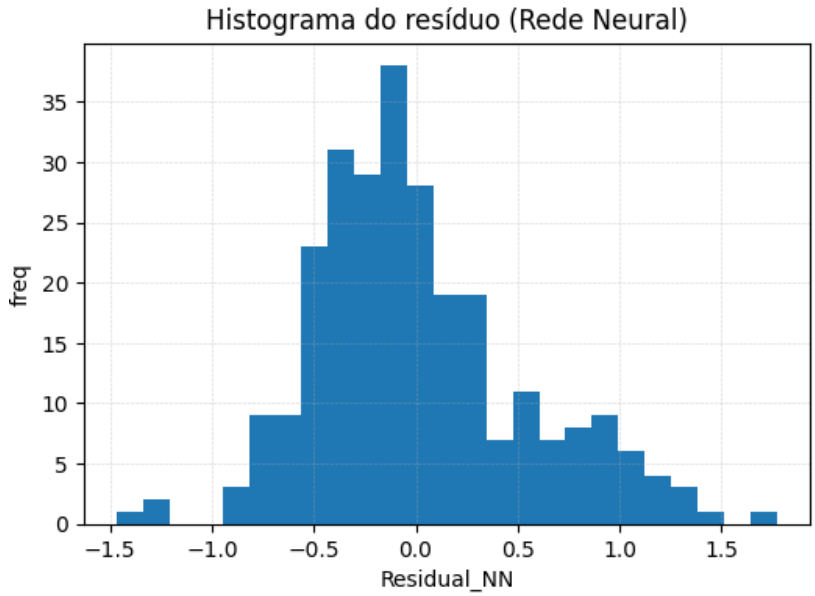

Figura 5.13 - Histograma do resíduo com Redes Neurais Artificiais.
Fonte: Autora.

5.3.2 Variograma do resíduo

A análise dos variogramas dos resíduos mostra que, mesmo após a modelagem da média local, os erros remanescentes ainda apresentam continuidade espacial, ou seja, o resíduo não se comportou de forma totalmente aleatória no espaço, indicando que a modelagem da média local retirou parte da tendência dos dados, mas não eliminou completamente a estrutura espacial do erro. Foram plotados variogramas experimentais com diferentes direções, afim de encontrar o que melhor se adaptadava ao modelo esférico de variograma como mostram a Figura 5.14, 5.16 e 5.18, mas os três métodos estudados. Para o ajuste ao modelo esférico, foi escolhido o variograma omnidirecional.

Na regressão linear, depois de plotados os variogramas em diferentes direções, o variograma omnidirecional (Figura 5.15) apresentou crescimento progressivo da semivariância até atingir um patamar compatível com o modelo ajustado. Como parâmetros, foram estabelecidos os valores indicados na Tabela 5.2:

Tabela 5.2 – Parâmetros adotados para a construção do semivariograma a Regresão Linear.
Fonte: Autora.

Parâmetros	Nugget	Sill	Range
Valores adotados	0.10	0.22	95.0

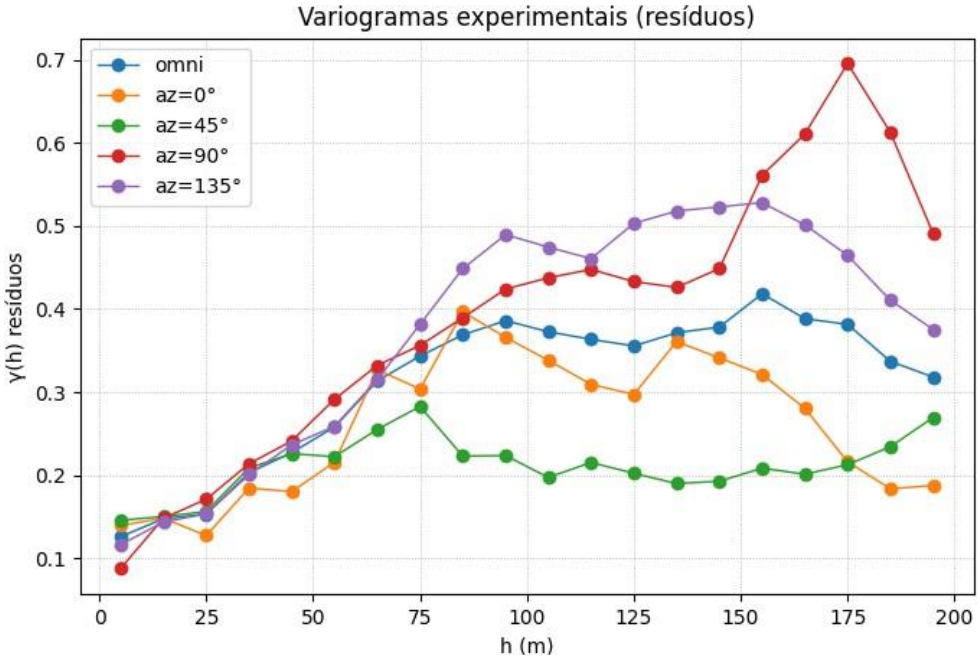

Figura 5.14 – Variogramas experimentais dos resíduos.
Fonte: Autora.

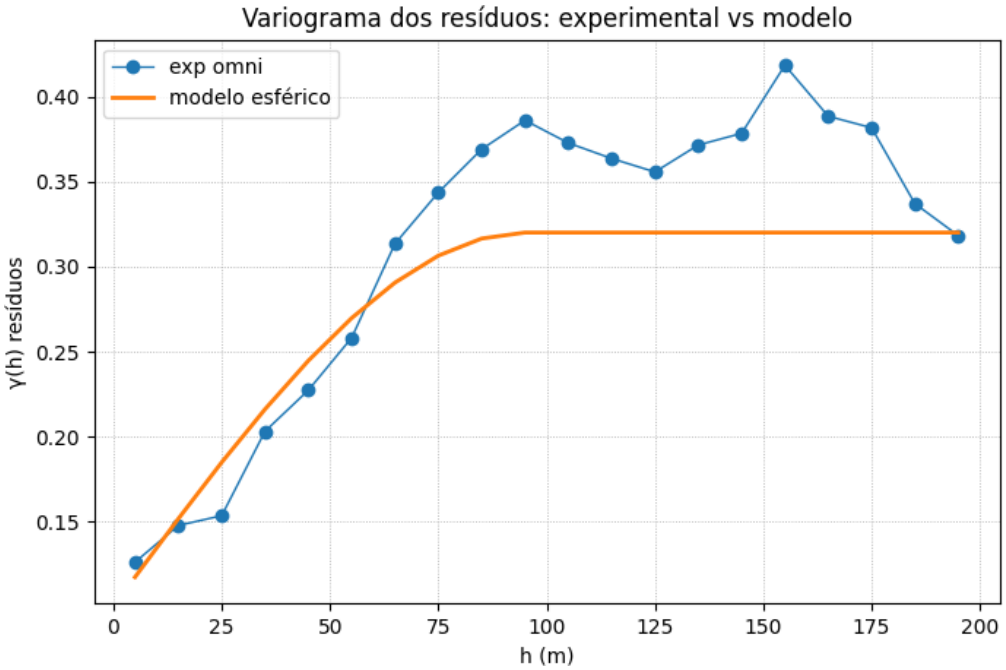

Figura 5.15 – Variograma omnidirectional dos resíduos ajustado ao modelo.
Fonte: Autora.

Observa-se que os resíduos ainda mantêm uma estrutura espacial relativamente forte e um crescimento progressivo até aproximadamente 95 metros (como pré-definido). Essa configuração aponta que a RL conseguiu remover grande parte da tendência dos dados, mas ainda restou uma parcela importante do erro com organização espacial bem definida. Nos

variogramas direcionais, observa-se menor semivariância na direção de 45º, indicando maior continuidade residual nesse azimute, enquanto a direção de 90º apresenta maiores valores, apresentando uma estrutura anisotrópica.

No caso do Random Forest, também foram definidos valores para os parâmetros (Tabela 5.3) a estrutura espacial dos resíduos mostrou-se mais curta e menos intensa, com um variograma omnidirecional (Figura 5.17) que cresce até 75 metros, valor correspondente ao alcance ajustado. Esse valor indica até que ponto a continuidade espacial do erro foi mantida, tornando-se mais fraca a partir daí, o que sugere uma modelagem da média local que absorveu melhor a parte sistemática dos dados e, por isso, houve uma redução da variabilidade espacial remanescente. Nos variogramas direcionais (Figura 5.16), a anisotropia se mostra presente, com maior continuidade aproximadamente em 45º e maiores valores de semivariância nas direções 90º e 135º.

Tabela 5.3 - Parâmetros adotados para a construção do semivariograma para o Random Forest.
Fonte: Autora

Parâmetros	Nugget	Sill	Range
Valores adotados	0.10	0.12	75.0

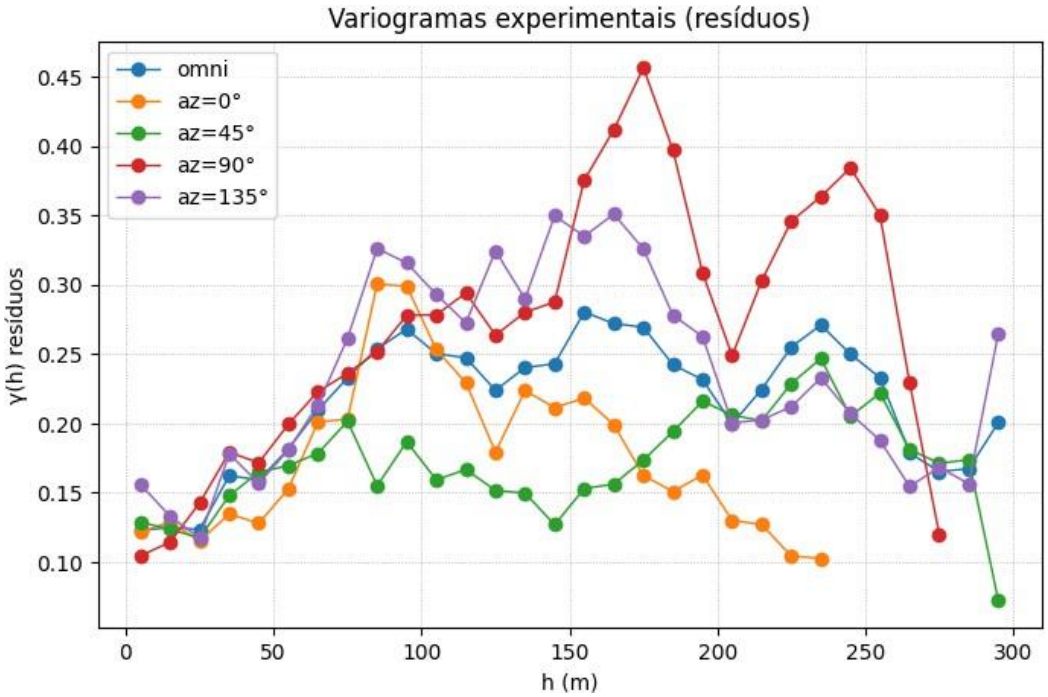

Figura 5.16 – Variogramas experimentais dos resíduos com Random Forest.
Fonte: Autora.

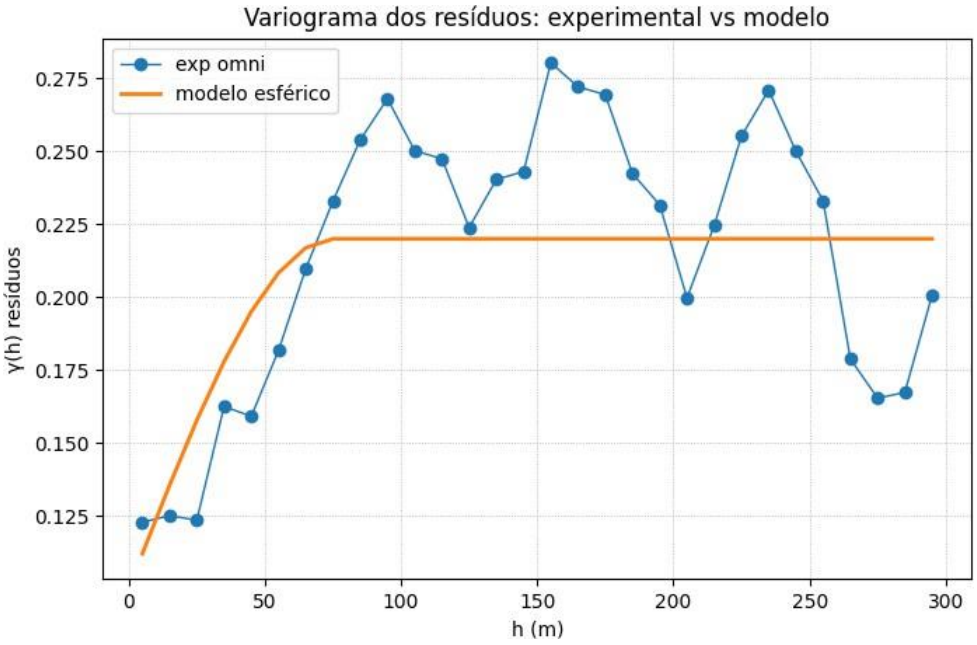

Figura 5.17 – Variograma omnidirectional dos resíduos ajustado ao modelo com Random Forest.
Fonte: Autora.

Já com as RNAs, o variograma omnidirecional (Figura 5.19) apresentou um comportamento intermediário, com os componentes estruturais mostrados na Tabela 5.14:

Tabela 5.4 - Parâmetros adotados para a construção do semivariograma para as RNAs.
Fonte: Autora

Parâmetros	Nugget	Sill	Range
Valores adotados	0.10	0.17	100.0

A continuidade espacial do resíduo permaneceu até 100 metros, uma distância um pouco maior do que aquelas observadas nos métodos anteriores. Com isso, percebe-se que a rede neural reduziu a variabilidade residual, mas ainda deixou uma estrutura espacial relevante ao longo de distâncias moderadas, com o variograma omnidirecional apresentando crescimento consistente até a faixa de 100 metros. Nos variogramas direcionais, a anisotropia também se manifesta, com direções de 90º e 135º com valores de semiariância mais altos, enquanto 0º e 45º apresentam uma continuidade mais forte.

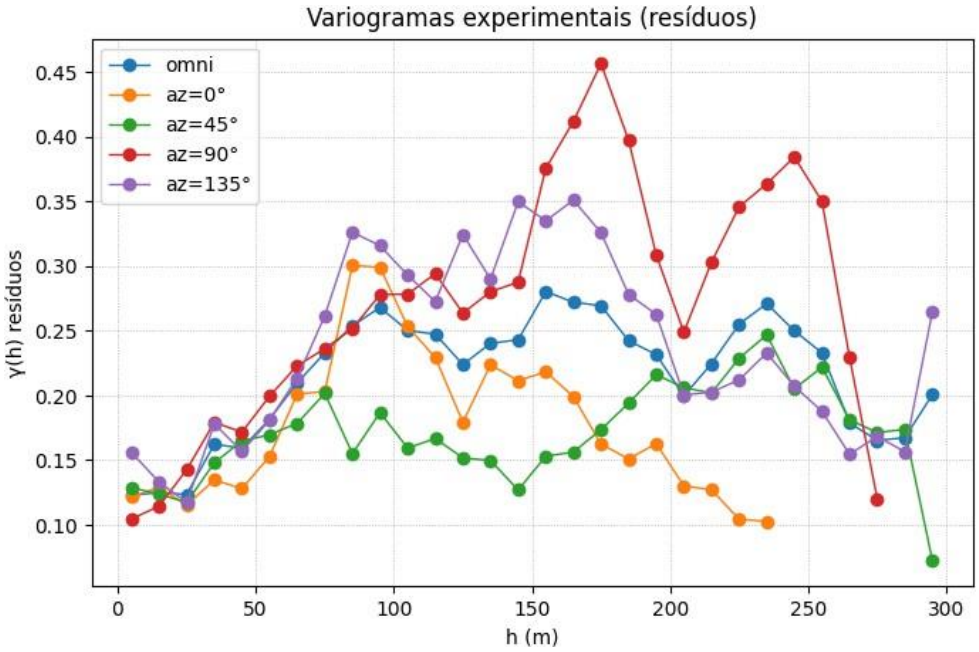

Figura 5.18 - Variogramas experimentais dos resíduos com Redes Neurais Artificiais.
Fonte: Autora.

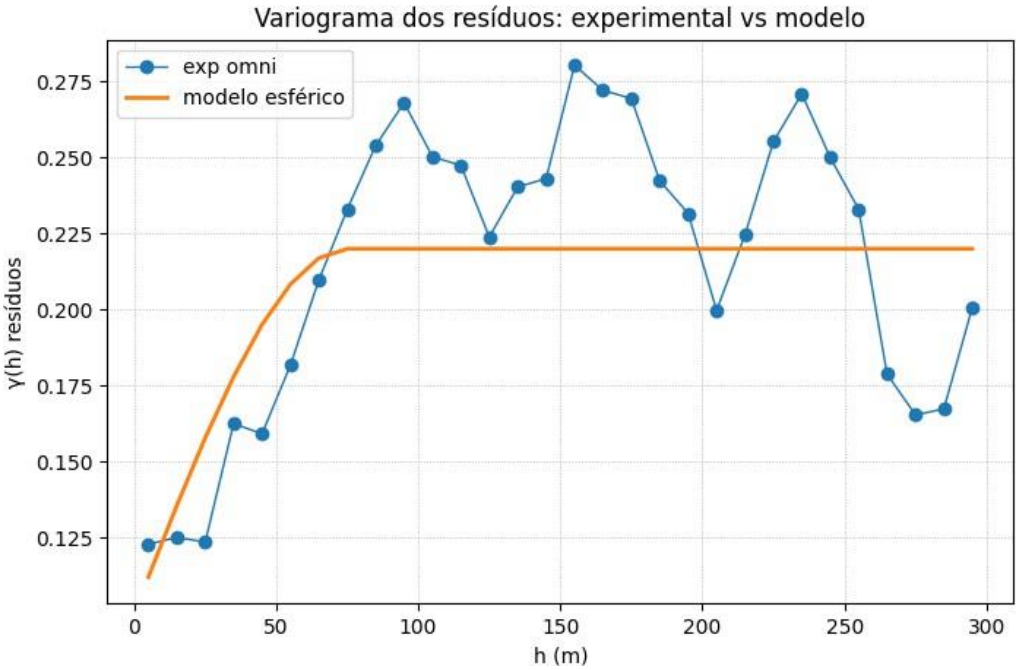

Figura 5.19 - Variograma omnidirectional dos resíduos ajustado ao modelo com Redes Neurais Artificiais.

Fonte: Autora.

5.3.3 Análise das estimativas krigadas

Nesta etapa, as estimativas finais foram obtidas pela soma entre a média local e o resíduo estimado por krigagem simples.

5.3.3.1 SKLM com média local linear

No método de regressão linear, o grid estimado apresenta um padrão espacial relativamente contínuo, predominando valores baixos a médios em grande parte do domínio, com um destaque para uma anomalia principal na porção compreendia entre X ≈ 80-110 e Y ≈ 195-210, onde existe uma concentração dos maiores valores. Fora dessa zona, o mapa mostra faixas alongadas de variação e transições suaves, condizente com a própria natureza da regressão linear como modelador da média local: é capaz de reproduzir uma tendência global, mas apresenta menor sensibilidade em captar variações locais mais complexas, permitindo que a krigagem dos resíduos ainda precise corrigir uma parcela espacial relevante do erro. Além disso, os resultados encontrados no *grid* (Figura 5.20) vai de encontro com o variograma de resíduos, calculado e ajustado anteriormente, que indicou uma continuidade até cerca 95 metros: isso significa que, após a retirada da média local linear, o resíduo ainda manteve dependência espacial relativamente expressiva, ampliando a atuação da krigagem na reconstrução do espaço final.

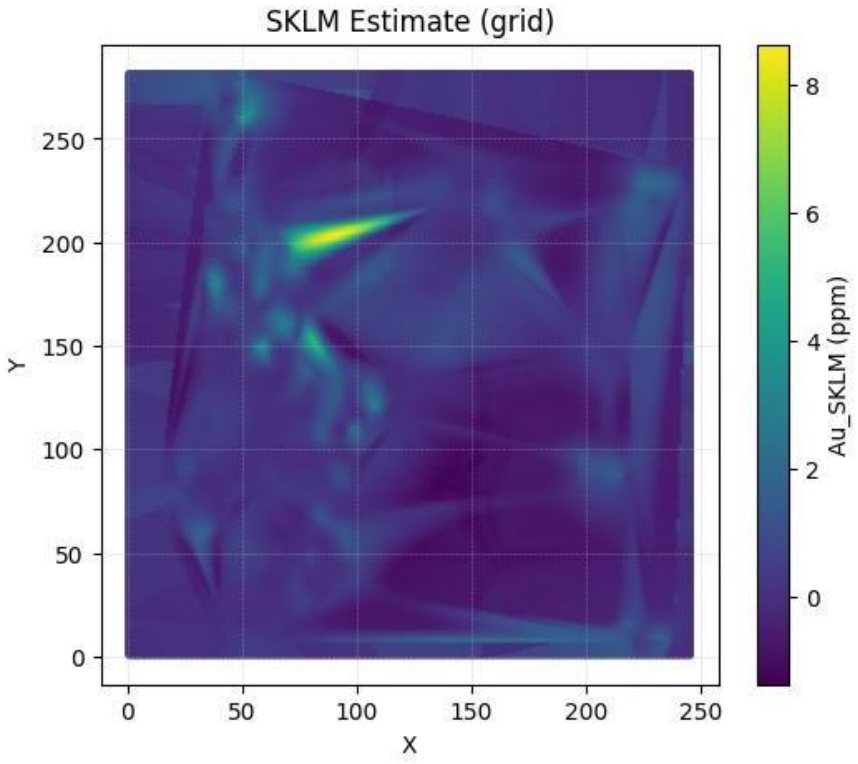

Figura 5.20 – Grid da estimative do método SKLM com regressão linear.
Fonte: Autora.

O histograma das estimativas lineares (Figura 5.21) reforça essa interpretação, apontando forte concentração de frequências em torno dos valores próximos de zero e baixos valores positivos, acompanhada por uma assimetria positiva e cauda longa, indicando que a maior parte das células do grid permanece associada a valores mais baixos, enquanto um conjunto reduzido de áreas concentra os valores mais elevados (> 6 ppm). Esse comportamento é esperado em variáveis com distribuição assimétrica, especialmente em um contexto de corpos mineralizados, nos quais anomalias tendem a ocupar uma fração menor da área total. A superfície linear mostra-se coerente com o comportamento observado ao longo das análises anteriores, apresentando preservação da anomalia principal, porém com maior espalhamento espacial.

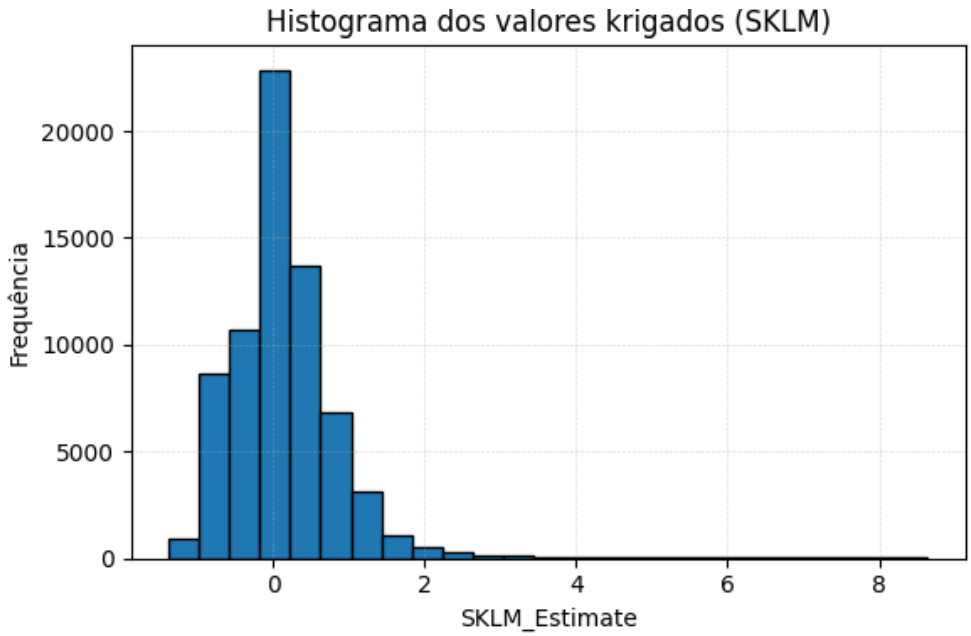

Figura 5.21 – Histograma dos valores krigados com regressão linear.
Fonte: Autora.

5.3.3.2 SKLM com Random Forest

Na abordagem com RF, a anomalia principal também permanece localizada entre as faixas X = 80-120 e Y = 190-210 (Figura 5.22), aproximadamente. Porém, sua distribuição no espaço é mais concentrada, com menor espalhamento para as áreas vizinhas e contornos melhor definidos nas zonas que representam alto valor. Se compararmos com a superfície linear, o mapa do RF apresenta manchas anômalas mais curta, com bordas mais definidas e menor propagação das zonas de maior teor para áreas próximas, sugerindo que a informação principal já foi melhor absorvida pela média local antes da etapa de krigagem. Essa interpretação é consistente com as etapas de análise anteriores, sendo o RF um modelo que produziu resíduos menos dispersos e com menor variabilidade estruturada, e seu variograma apresentou continuidade espacial até cerca de 75 metros, indicando que, após a modelagem da média local pelo RF, o resíduo conservou menor intensidade espacial e menor alcance de continuidade, implicando em uma krigagem de resíduos atuante em escala mais curta e localizada, refletindo em uma superfície final menos espalhada.

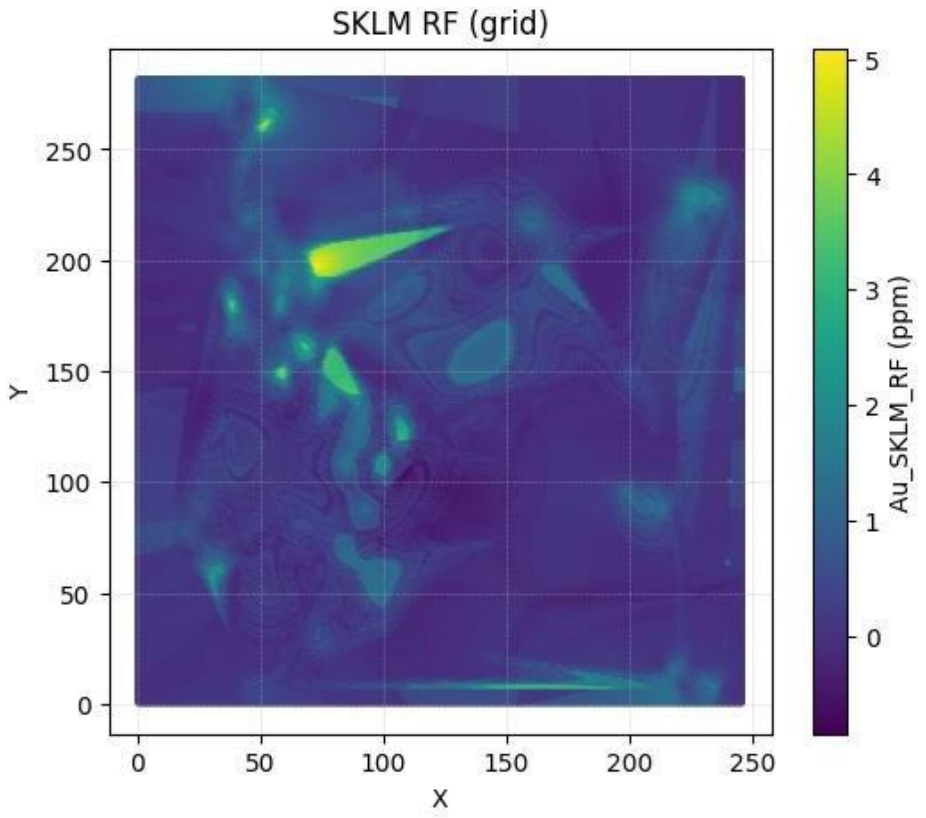

Figura 5.22 - Grid da estimative do método SKLM com Random Forest.
Fonte: Autora.

5.3.3.3 SKLM com Redes Neurais

No método de Redes Neurais, a superfície estimada também preserva a anomalia principal nas faixas correspondidas entres X = 80-110 e Y = 195-210, como mostra o grid da Figura 5.23, uma região que se concentram os maiores valores estimados, com predominância de valores baixos a médios no restante do domínio, distribuídos de maneira mais uniforme ao longo da área, ainda que com a presença de algumas feições secundárias discretas, especialmente na região central e em setores pontuais das bordas do domínio. A interpretação do grid também é coerente tanto com a modelagem da média local, que gerou uma resposta não linear e contínua para a relação entre a variável primária e secundária, refletindo na estimativa final que apresentou transições entre as zonas de fundo e as áreas de maior teor de forma gradual, sem perder a individualização da anomalia central, quanto com as análises dos resíduos e do variograma, que apresentavam resíduos com uma continuidade espacial até cerca de 100 metros, indicando que após a retirada da média local ainda permaneceu uma parcela do erro com uma organização espacial relevante.

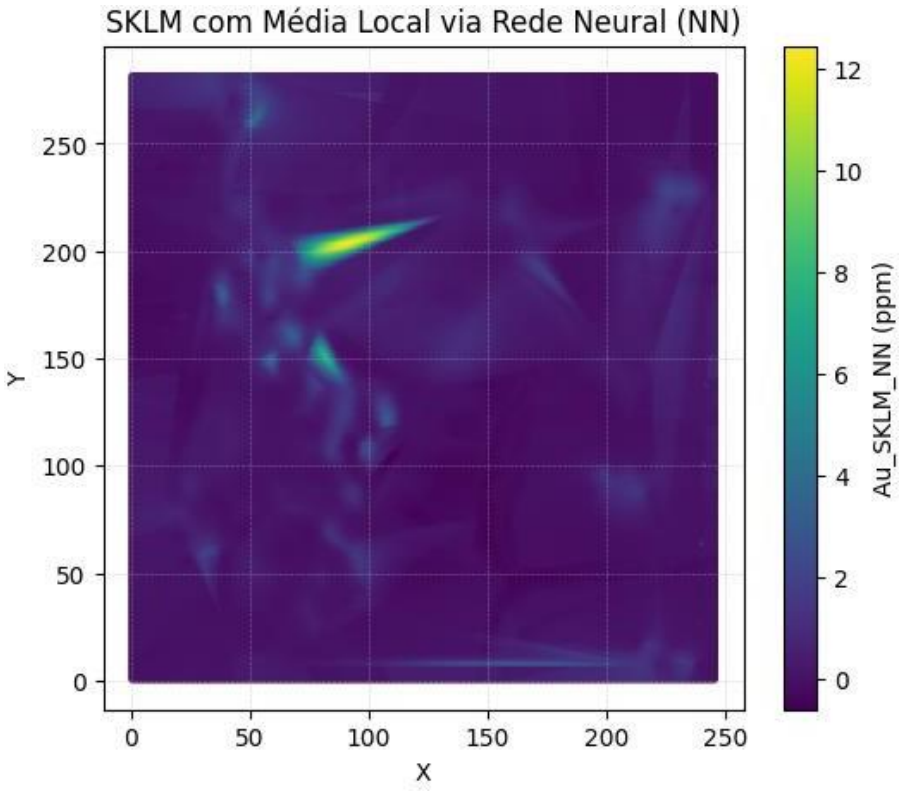

Figura 5.23 - Grid da estimative do método SKLM com Redes Neurais Artificiais.
Fonte: Autora.

5.4 Validação cruzada e análise comparativa

A avaliação preditiva dos modelos foi realizada por meio de validação cruzada *leave-one-out* (LOOCV), utilizando como critérios de comparação o coeficiente de determinação (R^2), o erro quadrático médio (RMSE) e o erro absoluto médio (MAE). De modo geral, os três modelos apresentaram desempenho satisfatório, com boa correspondência entre valores observados e preditos, sobretudo na faixa em que concentra a maior parte das amostras. A seguir, são apresentados os resultados e análises dessa validação.

5.4.1 Resultado da validação cruzada

Os resultados da validação cruzada, mostrados na Tabela 5.2, mostram as diferenças entres os métodos estudados:

Tabela 5.5 – Métricas de validação cruzada.
Fonte: Autora.

	Modelo	RMSE	MAE	R^2
0	SKLM Linear	0.394751	0.282185	0.736917
1	SKLM RF	0.359247	0.257233	0.782112
2	SKLM NN	0.364198	0.266744	0.776066

Considerando que os menores valores de RMSE e MAE indicam menor erro de predição, e que maior R^2 indica melhor capacidade de explicação da variabilidade observada, os dados mostram que os três modelos são consistentes, mas apresentam nível diferente de desempenho. Isso também pode ser observado pelo gráfico Observado x Predito (Figura 5.24), representado abaixo.

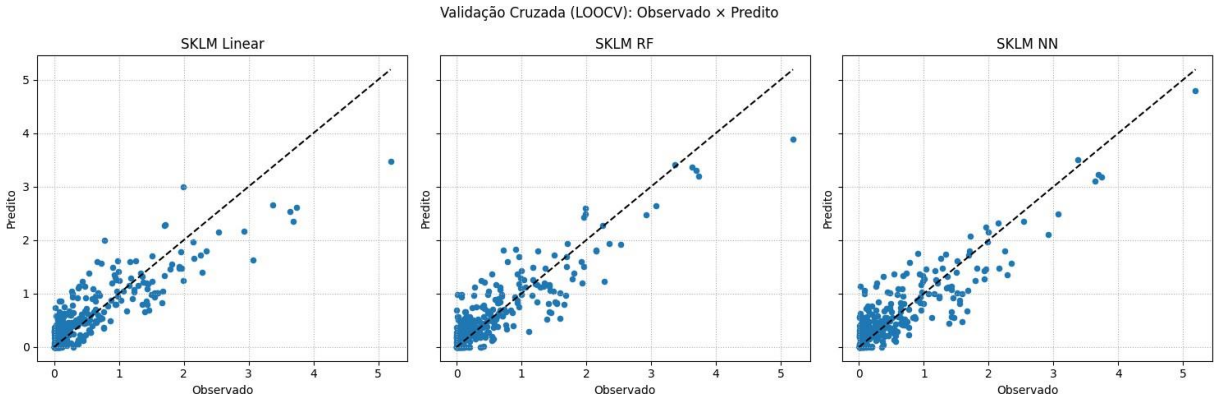

Figura 5.24 – Comparação da validação cruzada para os métodos Linear, Random Forest e Redes Neurais Artificiais.
Fonte: Autora.

A regressão linear foi a solução com maior erro médio e menor capacidade explicativa, resultado que reflete diretamente no gráfico (Figura 5.24), em que a nuvem de pontos acompanha a reta de forma satisfatória na faixa de baixos e médio valores, mas se afasta com maior intensidade à medida que os valores observados aumentam, principalmente com subestimação dos teores mais altos. Esse comportamento tem conexão com os resultados obtidos tanto na média local quanto para os resíduos, apontando um método que representa de forma satisfatória a tendência global, mas possui uma menor precisão nos extremos.

O Random Forest apresentou o melhor desempenho numérico do conjunto, com um maior valor de R^2, seguido de um menor RMSE e MAE, o que representa uma melhor aderência aos dados observados. No gráfico Observado x Predito, isso aparece na forma de uma nuvem

mais compacta em torno da reta, em especial na faixa que concentra a maior parte das amostras, o que é um resultado esperado, visto que o RF modelou melhor a média local e produziu resíduos menos dispersos, resultando em uma menor estrutura espacial residual, atingindo um menor alcance. Ainda assim, houve dispersão dos valores observados, mas de forma menos acentuada.

Enquanto isso, o método de Redes Neurais apresentou um desempenho parecido com o RF, com as métricas de validação assumindo valores que confirmam uma solução consistente. No gráfico (Figura 5.24), a nuvem de pontos também se concentra adequadamente ao redor da reta, com um comportamento competente para valores baixos e médio, porém, a medida em que os valores aumentam, a dispersão volta a crescer, mas ainda assim apresentando uma boa resposta em parte dos valores altos.

5.4.2 Análise e comparação das seções

Outro ponto utilizado para avaliar o desempenho dos métodos, foi a análise dos pontos estimados em função da curva de valores observados por seção, de forma a auxiliar na interpretação dos dados não apenas pelos valores estimados, mas também como cada método distribui espacialmente essa estimativa no domínio estudado. A Figura 5.25 mostra estas seções.

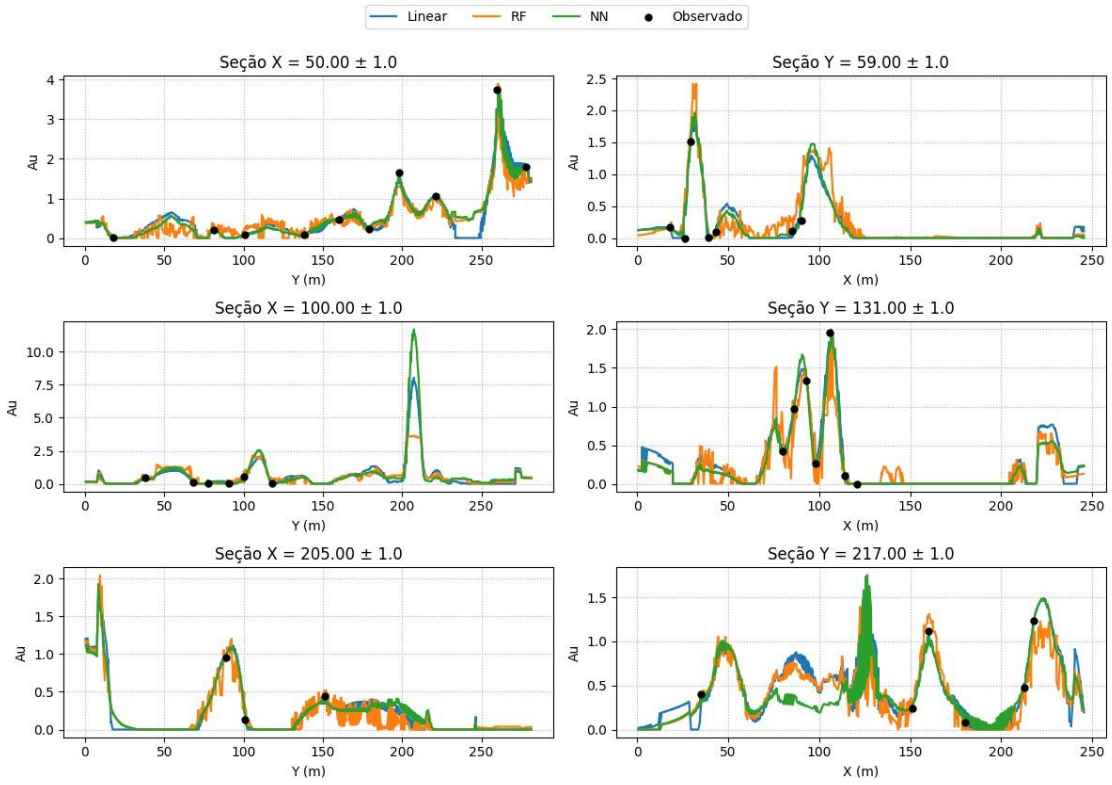

Figura 5.25 – Comparação da distribuição dos pontos estimados em cada método.
Fonte: Autora.

Como pode-se observar nas seções comparativas acima, há trechos em que os três métodos apresentam comportamentos muito similares, sobretudo nas seções em que predominam teores baixos a moderados, como é o caso das seções X = 50 ± 10 e em parte de Y = 205 ± 10, onde as três curvas seguem o padrão dos pontos observados com boa coerência. Nesses casos, a diferença entre os métodos não está na identificação da tendência principal, mas a forma como cada um dos métodos responde aos detalhes locais. Enquanto a RL tende a apresentar um traçado mais regular, o RF responde com maior oscilação às pequenas mudanças; já as redes neurais mantêm continuidade sem perder completamente a capacidade de acompanhar os contrastes do perfil.

Essa distinção de mostra de forma mais clara nas seções em que há maior contraste de valores, como X = 100 ± 10, por exemplo, em que os três métodos reconhecem a presença de um pico muito acentuado (em torno de Y ~ 205 metros), mas a amplitude atribuída a esse máximo varia: a RN produz a resposta mais intensa, seguida pela RL que também apresenta um pico elevado, enquanto o RF mantém um valor mais conservador. Em Y = 131 ± 10, nota-se uma diferença não mais na amplitude, mas na forma do traçado da curva. Os pontos observados indicam uma sequência de máximos concentrados entre X ~ 70 metros e X ~ 100 metros, e os três modelos são capazes de reconhecer esse perfil. A RL suaviza mais o perfil, reduzindo

parcialmente o detalhamento dos picos; o RF acompanha melhor as mudanças de intensidade, gerando uma curva mais irregular e sensível às variações de curto alcance e a RN preserva a feição principal com maior continuidade, reproduzindo os máximos de forma clara, mas sem introduzir o mesmo grau de oscilação observado no RF.

A seção Y = 59 ± 1,0 mostra como os modelos respondem a um perfil com forte elevação localizada no início da seção e redução abrupta logo em seguida. Nesse caso, o RF e a RN reproduzem com maior intensidade o pico principal, enquanto a RL apresenta transição menos acentuada, o que muda à medida que o perfil se altera, com o RF revelando maior irregularidade ao longo do espaço, ao passo que a RN tende a manter um traçado mais contínuo.

Considerando o conjunto de dados obtidos e analisados até aqui nesse trabalho, o SKLM com média local via Random Forest foi o método que apresentou o melhor desempenho global de estimativa. Essa conclusão se sustenta, primeiramente nos indicadores da validação cruzada, uma vez que o RF reuniu o menor RMSE (0,359), o menor MAE (0,257) e maior coeficiente de determinação ($R^2 = 0,782$), evidenciando maior aderência entre valores observados e preditos e menor erro médio de estimativa, além de produzir gráfico 'Observado x Predito' (Figura 5.25) mostrou uma nuvem de pontos mais concentrada em torno da reta, em especial na faixa onde se encontram a maior parte dos valores, lhe conferindo melhor ajuste ao comportamento dos dados. O método RF também representou de forma mais eficiente a relação não linear entre variável primária e secundária, gerando resíduos menos dispersos e com menor estrutura espacial remanescente, com alcance residual mais curto e menor componente estrutural, o que indica que a maior parcela de variabilidade já havia sido absorvida antes da krigagem.

6. LIMITAÇÕES E SUGESTÃO DE TRABALHOS FUTUROS

Para o presente trabalho, pontua-se como limitação o volume do banco de dados utilizado. Em contextos geoestatísticos, a densidade e a distribuição das amostras são fatores de extrema importância para obtenção de uma estimativa de qualidade, uma vez que o conhecimento da distribuição das variáveis influencia diretamente na modelagem da média local e na estrutura dos resíduos. Dessa forma, a utilização de um banco de dados mais amplo poderia resultar em uma modelagem mais robusta, aumentando a confiabilidade dos resultados e permitindo a visualização dos métodos em uma situação real.

Como perspectiva de trabalho futuros, sugere-se a aplicação de abordagens clássicas da geoestatística, como a krigagem ordinária e a cokrigagem, visando estabelecer uma base comparativa com o método adotado neste estudo. A krigagem ordinária permitiria avaliar o

comportamento das estimativas sem a imposição de uma média previamente modelada, enquanto a cokrigagem possibilita incorporar explicitamente a correlação espacial entre variáveis, explorando de forma mais direta a informação contida na variável secundária. A inclusão dessas abordagens é importante para ampliar a análise e aprofundar a compreensão sobre o impacto das diferentes estratégias de estimativa na modelagem espacial.

7. CONCLUSÕES

O presente trabalho teve como objetivo avaliar o uso de interpoladores não lineares na modelagem da média local no método geoestatístico SKLM (*Simple Kriging with Local Means*). Os resultados obtidos mostraram que a proposta foi atingida, uma vez que foi possível analisar, de forma comparativa, o comportamento da regressão linear, do *Random Forest* e das Redes Neurais Artificiais ao longo das etapas do estudo.

A análise exploratória inicial indicou que a variável secundária apresentava relação positiva e estatisticamente significativa com a variável primária, justificando sua utilização como informação auxiliar na modelagem da média local. No entanto, os resultados dos coeficientes de correlação, especificamente o Spearman, também mostraram que essa relação não era perfeitamente uniforme em todo domínio da amostra, fato que sustenta o uso de abordagens que se adaptam melhor as variações do que a regressão linear.

Na etapa de modelagem da média local, a regressão linear representou adequadamente a tendência global entre as variáveis, mas com menor capacidade de reproduzir variações mais intensas e localizadas. O Random Forest apresentou maior aderência ao comportamento empírico dos dados, evidenciando melhor capacidade de captar relações não lineares, enquanto as redes neurais Artificiais, apesar de modelar de forma satisfatória essa não linearidade, gerou uma resposta mais contínua e suave. Esse comportamento foi confirmado na análise dos resíduos, em que o modelo linear apresentou maior dispersão residual, enquanto o *Random Forest* produziu resíduos mais concentrados e com menor estrutura espacial remanescentes, ficando a rede neural em posição intermediária.

Na validação cruzada, o SKLM com média local modelada a partir do Random Forest, apresentou o melhor desempenho global, com menor RMSE, menor MAE e maior R^2 entre os métodos avaliados (RMSE = 0,3592, MAE = 0,2572 e R^2 = 0,7821), seguido pela rede neural que apresentou valores próximos e a regressão linear que, embora estável e coerente como modelo de referência, foi a que apresentou menor desempenho relativo, principalmente na reprodução de valores mais elevados.

Dessa forma, conclui-se que o uso de modeladores não lineares na média local obteve ganho em relação à formulação linear. Entre os métodos testados, o *Random Forest* foi o que apresentou a melhor resposta global, reunindo melhor desempenho preditivo, menor estrutura residual e maior capacidade de representar a variabilidade espacial relevante dos dados. Assim, os resultados deste trabalho indicam que a substituição da regressão linear por abordagens não lineares constitui uma alternativa viável e tecnicamente vantajosa para a modelagem da média local em SKLM, especialmente em situações em que a relação entre variável primária e secundária não é estritamente linear.

8. REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Maria Teresa Fernandes Matos. Contribuições iniciais à modelagem geometalúrgica de uma mina de ferro por meio de análises estatísticas multivariadas. 2025. Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Minas) – Universidade Federal de Ouro Preto, Ouro Preto, 2025.

ASSUMPÇÃO, Henrique César Pereira; HADLICH, Gisele Mara. Estatística descritiva e estacionaridade em variáveis geoquímicas ambientais. Engenharia Sanitária e Ambiental, v. 22, n. 4, p. 671-677, 2017.

AYACHE, N. K. et al. Mapas auto-organizáveis aplicados ao desagrupamento em amostragem preferencial. HOLOS, 2023.

BRAGA, Antônio de Pádua; CARVALHO, André Ponce de Leon F. de; LUDERMIR, Teresa Bernarda. Redes neurais artificiais: teoria e aplicações. Rio de Janeiro: LTC, 2000. CAMARGO, Eduardo Celso Gerbi. Geoestatística: fundamentos e aplicações. In: Geoprocessamento em Projetos Ambientais.

Breiman, L. Random Forests. *Machine Learning* **45**, 5–32 (2001). https://doi.org/10.1023/A:1010933404324

CARVALHO, Ivan Silva. Aplicação de algoritmos de agrupamento na definição de domínios para modelagem e estimativa de teores na mina de ferro de Capitão do Mato Nova Lima-MG. 2022. Trabalho de Conclusão de Curso (Graduação) – Instituto de Geociências, Universidade de São Paulo, São Paulo, 2022. Disponível em: https://bdta.abcd.usp.br/directbitstream/65c15251-9466-40c0-b8f6-4163535d2f20/Monografia_22_25_corrigida.pdf.

CARVALHO, Ivan Silva. Domínios de estimativa em modelagem geoestatística: aplicação de algoritmos de agrupamento na estimativa de recursos minerais. 2025. Dissertação (Mestrado em Ciências do Sistema Terra e Sociedade) - Instituto de Geociências, Universidade de São Paulo, São Paulo, 2025. doi:10.11606/D.44.2025.tde-28072025-075544.

DAVID, Michel. Geostatistical ore reserve estimation. Amsterdam: Elsevier Scientific Publishing Company, 1977.

GOOVAERTS, Pierre. Geostatistics for natural resources evaluation. New York: Oxford University Press, 1997.

GUERRA, P. A. G. Geoestatistica operacional. Brasília-DF: DNPM 1985.

HARRIS, Charles R. et al. Array programming with NumPy. Nature, p. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2.

HUNTER, John D. Matplotlib: a 2D graphics environment. Computing in Science & Engineering, p. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.

ILAMBWETSI, Patrícia de Sousa. Proposta de um interpolador geoestatístico híbrido com aprendizado de máquina. 2020. Tese (Doutorado em Estatística Aplicada e Biometria) – Universidade Federal de Viçosa, Viçosa, 2020.

INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS. Geoestatística: fundamentos e aplicações. Disponível em: http://www.dpi.inpe.br/gilberto/tutoriais/gis_ambiente/5geoest.pdf. Acesso em: 10 abr. 2026.

ISAAKS, Edward H.; SRIVASTAVA, R. Mohan. Applied geostatistics. New York: Oxford University Press, 1989

JOURNEL, A. G.; HUIJBREGTS, Ch. J. Mining geostatistics. London: Academic Press, 1978.

MANCIO, Alini Vieira. Modelagem de superfície de terreno por krigagem simples com médias locais utilizando dados de vegetação como variável secundária. 2021. Dissertação (Mestrado em Engenharia) – Universidade Federal do Rio Grande do Sul, Porto Alegre, 2021.

MATHERON, G. The theory of regionalized variables and its applications. Paris: École Nationale Supérieure des Mines de Paris, 1971.

MCKINNEY, Wes. Data structures for statistical computing in Python. In: VAN DER WALT, Stéfan; MILLMAN, Jarrod (ed.). Proceedings of the 9th Python in Science Conference. 2010. p. 56–61.

MITCHELL, Tom M. Machine learning. New York: McGraw-Hill, 1997.

OLIVEIRA, Sriratna Sousa de. Uso de machine learning na mineração: revisão de literatura e aplicação do algoritmo Random Forest para otimização da recuperação mássica durante o beneficiamento de ferro. 2022. Trabalho de Conclusão de Curso (Bacharelado em Geologia) – Universidade Federal de Uberlândia, Monte Carmelo, 2022.

PEDREGOSA, Fabian et al. Scikit-learn: machine learning in Python. Journal of Machine Learning Research. p. 2825–2830, 2011.

SANTOS, Allan Erlikhman Medeiros. Classificação de maciços rochosos por meio de técnicas da estatística multivariada e inteligência artificial. 2021. Tese (Doutorado em Engenharia Mineral) – Universidade Federal de Ouro Preto, Ouro Preto, 2021.

SILVA, Ricardo Alves da. Determinação de recursos minerais com utilização de otimização e algoritmos de aprendizado de máquina. 2019. 138 f. Tese (Doutorado em Ciência da Computação) – Universidade Federal de Pernambuco, Centro de Informática, Recife, 2019.

SIQUEIRA, Lucas Veras de. Aplicação de técnicas de inteligência artificial para avaliação de parâmetros de solo com foco em projeto geotécnico de poço. 2024. Trabalho de Conclusão de Curso (Graduação em Engenharia de Petróleo) – Universidade Federal de Alagoas, Maceió, 2024.

VIRTANEN, Pauli et al. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods, p. 261–272, 2020. DOI: 10.1038/s41592-019-0686-2.

WACKERNAGEL, Hans. Multivariate geostatistics: an introduction with applications. 3. ed. Berlin: Springer, 2003.

YAMAMOTO, Jorge Kazuo; LANDIM, Paulo M. Barbosa. Geoestatística: conceitos e aplicações. São Paulo: Oficina de Textos, 2013.

APÊNDICE A

Tabela A.1 – Coordenadas espaciais, teor de ouro (variável primária) e espessura da zona mineralizada (variável secundária). Fonte: Autora.

Coordenada X	Coordenada Y	Teor de Ouro (Au)	Espessura da zona mineralizada
40	71	0,0011	3,364461098
21	69	0,0078	1,677812555
28	80	0,1053	3,389746978
29	59	1,5127	5,699088055
41	81	0,0098	2,321785237
18	80	0,8604	3,079531506
39	60	0,0127	2,940866016
18	60	0,1771	2,20136306
41	90	0,0234	3,659791353
21	90	0,1731	2,325569722
31	101	0,2965	4,144547279
41	100	0,3006	4,695026582
21	100	0,2297	3,014293977
60	8	0,2583	3,450660914
40	11	0,0022	2,229023319
51	18	0,0142	1,797539946
59	20	0,0001	2,769280287
41	21	0,0031	1,527955767
59	90	0,1269	5,450463784
51	101	0,0738	4,405131929
50	81	0,2107	4,271777285
59	101	0,0304	4,394999654
60	81	0,0503	5,380137664
60	151	3,6918	9,543950577
38	148	0,05	2,274446961
50	160	0,4742	3,059761615
50	138	0,083	3,676793454
61	158	0,9838	4,73953282
39	160	0,2178	2,803144022
61	139	0,0715	3,945422511
38	140	0,0262	4,008237585
61	170	0,3011	2,688207089
39	170	1,5485	4,451975422
49	179	0,2349	1,928783919
58	179	2,1547	6,394102735
39	181	3,6374	9,051968236
60	191	1,9309	5,420265875
40	190	1,3911	3,759157792
51	198	1,6608	4,318557879
60	198	1,8137	5,597996851
40	200	0,2491	2,592930839
58	208	1,1601	4,253499349
38	209	0,5478	2,232282605
50	221	1,0666	3,749321777
61	220	0,0595	1,975947615
39	221	0,4209	2,503500162
59	268	1,7142	5,175416412
41	271	0,8287	3,142378258
49	278	1,7888	5,147227667
51	260	3,7389	8,333466603
59	281	0,675	2,922165739
39	279	1,1821	4,010298589
59	258	0,9833	2,495570925
38	260	0,7666	2,71506322
78	28	0,5654	4,116729127
60	29	0,0127	2,446346144
70	41	0,0849	3,537416174
70	21	0,0003	3,802770638
78	41	0,1246	3,54581669
61	41	0,0008	3,782168778
78	20	0,07	2,231465493
80	131	0,4211	4,965873997
58	128	0,0043	4,064256195
71	140	0,1751	4,455034574
79	138	1,6945	7,521045447
80	119	0,0351	5,49901833
61	121	0,0018	4,16783705

Tabela A.2 – Continuação: Coordenadas espaciais, teor de ouro (variável primária) e espessura da zona mineralizada (variável secundária). Fonte: Autora.

Coordenada X	Coordenada Y	Teor de Ouro (Au)	Espessura da zona mineralizada
71	160	2,535	7,194449613
78	159	3,3715	9,676376205
80	168	0,6346	3,874151274
69	181	0,7626	2,925580694
79	181	0,2417	2,281799173
80	188	0,4311	2,654931203
70	198	1,9921	5,987649379
80	49	0,4738	5,145508816
90	58	0,2802	5,717088712
88	39	0,0768	3,96089933
101	38	0,464	5,251337176
101	68	0,0978	5,716932087
79	70	0,1039	4,954905108
90	79	0,1383	5,900562304
100	78	0,0709	5,067787751
81	81	0,1972	5,452705507
100	91	0,0389	5,252173915
80	89	0,1923	5,063110944
91	99	0,1596	6,06204904
101	100	0,5393	6,010641463
81	98	0,0013	5,485742258
98	111	1,987	9,021567732
81	108	0,643	6,754151196
90	120	1,2903	7,668288907
100	118	0,0205	4,889781876
98	130	0,2639	4,715686999
90	140	2,1475	7,883473528
90	138	1,9271	6,611986565
121	131	0,0002	3,386370149
111	140	0,0488	3,401174369
108	121	3,0709	9,60836514
120	141	0,0146	4,351975246
119	118	0,0169	4,017199932
158	228	0,5516	3,836726423
140	229	0,5139	4,608221319
150	241	0,1292	2,678575124
151	218	0,2392	4,196851114
161	241	0,0086	2,154432049
141	240	0,0046	2,505514614
160	218	1,1187	4,816809954
139	220	0,4457	3,733282237
178	211	0,5618	5,030194365
159	209	0,1488	2,854193261
169	221	0,1972	3,900202083
170	198	1,4298	6,603731273
180	218	0,0775	3,087358123
178	201	0,0813	3,039905037
158	198	0,1545	4,207993228
219	88	0,1369	2,075774462
198	90	0,7358	4,061749412
211	100	0,7106	3,747352213
208	80	0,4958	4,115964821
221	99	0,5868	3,161335321
199	98	1,419	4,624172036
220	81	0,1773	2,482652125
198	78	0,086	2,751431624
220	150	0,675	5,026687409
200	150	0,3811	4,737944144
208	159	0,2804	4,140360478
210	140	0,3303	4,619072035
221	160	0,3656	2,990234779
198	161	0,1545	3,602166718
219	139	0,0245	2,380296207
200	139	0,0422	3,230186377
239	8	0,0802	0,854565567
218	8	1,3734	4,502333883
229	19	0,1477	1,741464927

Tabela A.3 - Continuação: Coordenadas espaciais, teor de ouro (variável primária) e espessura da zona mineralizada (variável secundária). Fonte: Autora.

Coordenada X	Coordenada Y	Teor de Ouro (Au)	Espessura da zona mineralizada
239	18	0,1867	1,00830133
218	18	1,4297	3,657249147
238	229	1,5109	5,027368835
218	228	0,2652	3,455497395
231	239	0,478	4,150490362
230	221	0,5389	5,085625181
240	239	0,0519	2,316372901
221	241	0,2909	2,127944494
239	220	0,3983	1,648764437
218	218	1,2367	5,04675688
35	71	0,4868	3,313577667
24	71	0,4639	3,016798999
34	88	0,0001	2,816687099
23	91	0,7088	3,817362668
54	10	0,3494	3,480017494
46	11	0,392	2,85740461
55	89	0,0212	4,40338672
45	89	0,0628	3,352900368
53	150	0,873	3,398116836
46	148	0,0075	2,411286188
55	168	0,2881	2,801482744
43	170	1,1749	5,216294787
55	191	1,0047	3,823908817
44	191	0,0768	2,52330091
55	211	0,9388	3,17659729
46	211	0,4297	1,761923053
54	269	2,9224	6,230545938
43	271	0,1981	1,441618058
73	29	0,2317	4,944275274
64	31	0,0108	2,528084771
75	129	0,6393	6,083415228
64	129	0,0086	3,611082085
73	149	0,6961	5,211636128
64	151	0,6641	4,523858339
75	171	0,1448	2,82594826
63	168	1,4461	5,06200861
73	188	2,3515	6,124809235
64	191	1,2576	4,610041378
93	48	0,3732	4,919228735
86	48	0,0318	3,673656088
93	70	0,1961	4,779353343
84	69	1,21	7,32231855
93	90	1,0313	8,301762795
86	89	0,117	5,806594968
96	111	0,3605	5,766393821
85	108	1,4955	8,2127612
93	131	1,3368	7,662905772
86	131	0,9653	6,330844393
114	131	0,1044	4,262400049
106	130	1,9577	8,616764498
155	229	0,0882	3,444746108
145	230	0,0635	3,963997656
174	208	0,335	4,338763514
166	211	0,6476	4,008417206
215	89	1,457	4,031247165
205	89	0,9558	5,139204487
215	148	0,0677	4,272612937
204	151	0,4445	4,149868838
236	9	1,5938	4,187642023
223	9	0,2776	2,992753292
236	229	1,962	5,018730249
226	230	2,2798	5,868831251
35	80	0,002	3,503983877
24	79	0,6357	3,456175698
36	61	1,5478	5,51792692
26	58	0,0003	2,191144317
16	80	0,6348	3,583009482

Tabela A.4 - Continuação: Coordenadas espaciais, teor de ouro (variável primária) e espessura da zona mineralizada (variável secundária). Fonte: Autora.

Coordenada X	Coordenada Y	Teor de Ouro (Au)	Espessura da zona mineralizada
43	60	0,0974	4,042537384
15	88	0,3193	2,892544905
46	99	0,1002	4,566633743
54	80	0,105	3,165048463
46	81	0,1386	3,988906672
54	161	0,4667	2,817155997
43	161	0,2611	1,967635017
65	160	2,2525	6,301411223
33	160	0,0486	1,924107094
36	170	0,6055	3,562803528
53	179	0,4254	2,029264261
44	180	0,9378	3,612069642
65	181	0,5893	3,010908672
34	180	0,1102	1,778837193
33	191	0,0791	2,430238414
55	199	0,3764	2,866356697
46	198	0,6746	3,784277222
63	201	1,3633	4,321917319
34	201	0,0796	1,372617233
65	210	0,4883	2,660884116
35	208	0,1612	2,064523161
46	220	1,2361	3,994366772
36	219	0,1523	1,226800607
35	217	0,397	2,371926812
53	258	1,6964	4,765964502
46	260	0,7798	4,461474115
45	281	0,9348	4,318103833
35	278	1,5883	3,84699399
35	259	0,0189	1,383617553
84	30	0,0853	2,912664682
84	41	0,034	3,743146771
75	40	0,3562	4,529467408
73	141	0,6073	5,599458063
63	140	0,3578	3,779483411
84	138	0,8934	5,32407253
76	159	0,9973	3,558605759
84	161	0,3872	3,741813804
86	169	0,7617	3,681496259
73	199	5,1901	11,17101405
76	51	0,1018	3,803147922
94	61	1,354	7,335961025
85	60	0,1175	3,915752926
104	38	0,2002	3,840367257
93	41	0,0284	5,604862124
75	90	0,113	4,770202007
94	101	0,0129	4,792986641
85	100	0,126	6,816798463
104	109	0,9086	8,464012618
75	110	0,349	5,115494782
95	121	0,5484	5,637607154
83	119	0,7014	6,451482417
94	140	0,7232	6,426561085
103	139	0,5653	4,567089127
114	120	0,1542	4,720810742
104	118	0,2892	5,829291707
196	91	0,2541	3,375827175
215	101	0,004	2,171247716
204	101	0,1273	2,76723284
196	101	0,4194	2,728065023
195	149	0,1419	3,774420698
216	11	1,4248	4,084444329
225	19	0,5122	2,698131078
214	19	0,0156	1,435121695
245	231	0,0261	2,323714837
233	220	0,9597	4,658500181
226	221	1,6815	5,565527964
213	218	0,4762	3,832664213

APÊNDICE B – Código implementado no estudo

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.spatial import cKDTree
from sklearn.neural_network import MLPRegressor
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline
from sklearn.metrics import r2_score
from google.colab import files
uploaded = files.upload()
POINTS_CSV = next(k for k in uploaded.keys() if "point" in k.lower() or "data" in k.lower())
GRID_CSV  = next(k for k in uploaded.keys() if "grid" in k.lower())
print("Points:", POINTS_CSV)
print("Grid  :", GRID_CSV)
df = pd.read_csv(POINTS_CSV, sep="\t")
print("Points columns:", df.columns.tolist())
df.head()
df.columns = df.columns.str.strip()
grid_df = pd.read_csv(GRID_CSV)
print(grid_df.columns.tolist())
grid_df.head()
grid_vals = grid_df.iloc[:, 0].astype(float).to_numpy()
print("Nº de valores no grid:", len(grid_vals))
print("Primeiros valores:", grid_vals[:10])
Nx = 246
Ny = 282
dx = 1.0
dy = 1.0
x0 = 0.5
y0 = 0.5
assert len(grid_vals) == Nx*Ny, "ERRO: Nx*Ny não bate com o número de valores do grid!"
xs = x0 + np.arange(Nx)*dx
```

```
ys = y0 + np.arange(Ny)*dy
Xg, Yg = np.meshgrid(xs, ys)
grid = pd.DataFrame({
    "X": Xg.ravel(order="C"),
    "Y": Yg.ravel(order="C"),
    "Secondary": grid_vals
})
grid.head()
grid_out = "grid_secondary_pointset.csv"
grid.to_csv(grid_out, index=False)
print("Salvo:", grid_out)
COL_X = "X"
COL_Y = "Y"
COL_P = "Primary"
COL_S = "Secondary"
df_sklm = df[[COL_X, COL_Y, COL_P, COL_S]].copy()
df_sklm.columns = ["X","Y","Primary","Secondary"]
df_sklm = df_sklm.replace([-99, -999, -999.0], np.nan).dropna()
df_sklm.head()
import numpy as np
import matplotlib.pyplot as plt
COL_THICK = COL_S
if COL_THICK not in grid.columns:
    raise ValueError(f"Coluna '{COL_THICK}' não existe no grid. Colunas:
{grid.columns.tolist()}")
def section_profile_grid(grid, axis="X", value=100.0, tol=1.0, col=COL_THICK):
    axis = axis.upper()
    if axis == "X":
        grd = grid[np.abs(grid["X"] - value) <= tol].copy()
        coord = "Y"
    else:
        grd = grid[np.abs(grid["Y"] - value) <= tol].copy()
        coord = "X"
    grd = grd.sort_values(coord)
```

```
    return grd, coord
def pick_sections_with_data_grid(grid, axis="X", n=5, base_tol=1.0, min_cells=30,
max_tol=10.0):
    """
    Escolhe seções distribuídas no domínio e garante que cada seção tenha
    pelo menos 'min_cells' células do grid (aumentando tol se precisar).
    """
    axis = axis.upper()
    vals = np.sort(grid[axis].unique())
    targets = np.quantile(vals, np.linspace(0.1, 0.9, n))
    chosen, chosen_tol = [], []
    for t in targets:
        v = vals[np.argmin(np.abs(vals - t))]
        tol = base_tol
        while tol <= max_tol:
            ncells = (np.abs(grid[axis] - v) <= tol).sum()
            if ncells >= min_cells:
                break
            tol += base_tol
        chosen.append(float(v))
        chosen_tol.append(float(tol))
    out, out_tol = [], []
    for v, tol in zip(chosen, chosen_tol):
        if v not in out:
            out.append(v); out_tol.append(tol)
    if len(out) < n:
        extra = np.linspace(vals.min(), vals.max(), n)
        for t in extra:
            v = vals[np.argmin(np.abs(vals - t))]
            if v not in out:
                out.append(float(v)); out_tol.append(float(base_tol))
            if len(out) == n:
                break
    return out[:n], out_tol[:n]
```

```
N_SECTIONS = 5
BASE_TOL = 0.5
MIN_CELLS = 80
MAX_TOL = 5.0
sec_x, tol_x = pick_sections_with_data_grid(grid, axis="X", n=N_SECTIONS,
base_tol=BASE_TOL, min_cells=MIN_CELLS, max_tol=MAX_TOL)
sec_y, tol_y = pick_sections_with_data_grid(grid, axis="Y", n=N_SECTIONS,
base_tol=BASE_TOL, min_cells=MIN_CELLS, max_tol=MAX_TOL)
print("Seções X (valor, tol):", list(zip(sec_x, tol_x)))
print("Seções Y (valor, tol):", list(zip(sec_y, tol_y)))
fig, axes = plt.subplots(nrows=N_SECTIONS, ncols=2, figsize=(12, 2.6*N_SECTIONS),
sharex=False)
for i, (vx, tx) in enumerate(zip(sec_x, tol_x)):
    grd_sec, coord = section_profile_grid(grid, axis="X", value=vx, tol=tx, col=COL_THICK)
    ax = axes[i, 0]
    ax.plot(grd_sec[coord], grd_sec[COL_THICK], "-", lw=1.8)
    ax.set_title(f"Seção X = {vx:.2f} ± {tx:.2f}")
    ax.set_xlabel(f"{coord} (m)")
    ax.set_ylabel("Espessura (m)")
    ax.grid(True, linestyle=":")
for i, (vy, ty) in enumerate(zip(sec_y, tol_y)):
    grd_sec, coord = section_profile_grid(grid, axis="Y", value=vy, tol=ty, col=COL_THICK)
    ax = axes[i, 1]
    ax.plot(grd_sec[coord], grd_sec[COL_THICK], "-", lw=1.8)
    ax.set_title(f"Seção Y = {vy:.2f} ± {ty:.2f}")
    ax.set_xlabel(f"{coord} (m)")
    ax.set_ylabel("Espessura (m)")
    ax.grid(True, linestyle=":")
plt.suptitle("Seções da variável secundária (espessura mineralizada) — 5 em X e 5 em Y",
y=1.01)
plt.tight_layout()
plt.show()
fig.savefig("secoes_espessura_secondary_5x5.png", dpi=200, bbox_inches="tight")
print("Salvo: secoes_espessura_secondary_5x5.png")
```

```
from scipy.stats import pearsonr, spearmanr
import numpy as np
S = df[COL_S].to_numpy(float)
P = df[COL_P].to_numpy(float)
r_pearson, p_pearson = pearsonr(S, P)
r_spearman, p_spearman = spearmanr(S, P)
print("Correlação Pearson (linear): r = %.3f | p = %.2e" % (r_pearson, p_pearson))
print("Correlação Spearman (monótona): ρ = %.3f | p = %.2e" % (r_spearman, p_spearman))
import pandas as pd
df_aux = df[[COL_S, COL_P]].copy()
df_aux["Class_Thickness"] = pd.qcut(df_aux[COL_S], q=5)
stats = df_aux.groupby("Class_Thickness")[COL_P].agg(["mean", "median", "std", "count"])
stats
plt.figure(figsize=(6,4))
plt.plot(stats["mean"].values, marker="o", label="Média")
plt.plot(stats["median"].values, marker="s", label="Mediana")
plt.xlabel("Classe de espessura (quantis)")
plt.ylabel("Au (ppm)")
plt.title("Variação do teor de Au por classes de espessura")
plt.grid(True, linestyle=":")
plt.legend()
plt.show()
S = df[COL_S].to_numpy(float)
P = df[COL_P].to_numpy(float)
b1 = np.cov(S, P, ddof=1)[0,1] / (np.var(S, ddof=1) + 1e-12)
b0 = P.mean() - b1*S.mean()
print("Regressão (média local): P = b0 + b1*S")
print("b0 =", b0)
print("b1 =", b1)
plt.figure(figsize=(6,5))
plt.scatter(S, P, s=18)
sline = np.linspace(S.min(), S.max(), 200)
plt.plot(sline, b0 + b1*sline, linewidth=2)
plt.xlabel(COL_S); plt.ylabel(COL_P)
```

```
plt.title("Scatter Primary × Secondary + regressão (média local)")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.show()
df["Local_Mean_Primary"] = b0 + b1*df[COL_S]
grid["Local_Mean_Primary"] = b0 + b1*grid[COL_S]
df[["Local_Mean_Primary"]].head(), grid[["Local_Mean_Primary"]].head()
plt.figure(figsize=(6,5))
plt.scatter(df[COL_X], df[COL_Y], c=df["Local_Mean_Primary"], s=25)
plt.colorbar(label="Local_Mean_Primary (no dataset)")
plt.title("Local Mean Primary no dataset (pontos)")
plt.xlabel("X"); plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
plt.figure(figsize=(6,5))
plt.scatter(grid[COL_X], grid[COL_Y], c=grid["Local_Mean_Primary"], s=5)
plt.colorbar(label="Local_Mean_Primary (no grid)")
plt.title("Local Mean Primary no grid (exaustivo)")
plt.xlabel("X"); plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
df["Residual"] = df[COL_P] - df["Local_Mean_Primary"]
print(df["Residual"].describe())
plt.figure(figsize=(6,4))
plt.hist(df["Residual"], bins=25)
plt.title("Histograma do resíduo")
plt.xlabel("Residual"); plt.ylabel("freq")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
print("\nVariância:", df["Residual"].var())
coords = df[[COL_X, COL_Y]].to_numpy(float)
r = df["Residual"].to_numpy(float)
n = len(r)
def experimental_variogram(coords, values, lag=10.0, hmax=200.0, azimuth=None,
tol=22.5):
```

```
    dx = coords[:,0][:,None] - coords[:,0][None,:]
    dy = coords[:,1][:,None] - coords[:,1][None,:]
    h = np.sqrt(dx*dx + dy*dy)
    az = (np.degrees(np.arctan2(dy, dx)) + 360) % 180
    gamma = 0.5*(values[:,None] - values[None,:])**2
    iu = np.triu_indices(len(values), k=1)
    h_u, az_u, g_u = h[iu], az[iu], gamma[iu]
    if azimuth is not None:
        diff = np.abs(az_u - azimuth)
        diff = np.minimum(diff, 180-diff)
        mdir = diff <= tol
        h_u, g_u = h_u[mdir], g_u[mdir]
    edges = np.arange(0, hmax+lag, lag)
    centers = (edges[:-1] + edges[1:]) / 2
    gam = np.full(len(centers), np.nan)
    pairs = np.zeros(len(centers), int)
    for k in range(len(centers)):
        m = (h_u >= edges[k]) & (h_u < edges[k+1])
        pairs[k] = int(m.sum())
        if pairs[k] > 0:
            gam[k] = float(np.mean(g_u[m]))
    return centers, gam, pairs
LAG = 10.0
HMAX = 200.0
hc, g_omni, p_omni = experimental_variogram(coords, r, lag=LAG, hmax=HMAX)
plt.figure(figsize=(8,5))
plt.plot(hc, g_omni, marker="o", linewidth=1, label="omni")
for azm in [0,45,90,135]:
    hc2, g_dir, p_dir = experimental_variogram(coords, r, lag=LAG, hmax=HMAX,
azimuth=azm, tol=22.5)
    plt.plot(hc2, g_dir, marker="o", linewidth=1, label=f"az={azm}°")
plt.xlabel("h (m)"); plt.ylabel("γ(h) resíduos")
plt.title("Variogramas experimentais (resíduos)")
plt.grid(True, linestyle=":", linewidth=0.6)
```

```
plt.legend()
plt.show()
NUGGET = 0.10
SILL_PARTIAL = 0.22
RANGE = 95.0
def sph_gamma(h, nug, psill, a):
    h = np.asarray(h, float)
    r = h/a
    out = np.empty_like(r)
    inside = r < 1
    out[inside] = nug + psill*(1.5*r[inside] - 0.5*r[inside]**3)
    out[~inside] = nug + psill
    return out
def cov_from_gamma(h, nug, psill, a):
    return (nug + psill) - sph_gamma(h, nug, psill, a)
plt.figure(figsize=(8,5))
plt.plot(hc, g_omni, marker="o", linewidth=1, label="exp omni")
plt.plot(hc, sph_gamma(hc, NUGGET, SILL_PARTIAL, RANGE), linewidth=2,
label="modelo esférico")
plt.xlabel("h (m)"); plt.ylabel("γ(h) resíduos")
plt.title("Variograma dos resíduos: experimental vs modelo")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.legend()
plt.show()
tree = cKDTree(coords)
K = 12
def skl_residual_predict(query_xy):
    dists, idxs = tree.query(query_xy, k=min(K, len(r)))
    if np.ndim(idxs)==1:
        idxs = idxs[:,None]
    r_pred = np.zeros(len(query_xy), float)
    for i in range(len(query_xy)):
        inds = np.atleast_1d(idxs[i])
        pts = coords[inds]
```

```
        dd = pts[:,None,:] - pts[None,:,:]
        hij = np.sqrt((dd[:,:,0]**2) + (dd[:,:,1]**2))
        C = cov_from_gamma(hij, NUGGET, SILL_PARTIAL, RANGE)
        C.flat[::C.shape[0]+1] += 1e-10*(NUGGET+SILL_PARTIAL)
        ht = np.sqrt(((pts - query_xy[i])**2).sum(axis=1))
        c = cov_from_gamma(ht, NUGGET, SILL_PARTIAL, RANGE)
        w = np.linalg.solve(C, c)
        r_pred[i] = float(np.dot(w, r[inds]))
    return r_pred
grid_xy = grid[[COL_X, COL_Y]].to_numpy(float)
grid["Residual_SK"] = skl_residual_predict(grid_xy)
grid["SKLM_Estimate"] = grid["Local_Mean_Primary"] + grid["Residual_SK"]
grid[["SKLM_Estimate"]].head()
out_csv = "SKLM_grid_results.csv"
grid.to_csv(out_csv, index=False)
files.download(out_csv)
plt.figure(figsize=(6,5))
plt.scatter(grid[COL_X], grid[COL_Y], c=grid["SKLM_Estimate"], s=5)
plt.colorbar(label="Au_SKLM (ppm)")
plt.title("SKLM Estimate (grid)")
plt.xlabel("X"); plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
import matplotlib.pyplot as plt
plt.figure(figsize=(6,4))
plt.hist(grid["SKLM_Estimate"].dropna(), bins=25, edgecolor="black")
plt.title("Histograma dos valores krigados (SKLM)")
plt.xlabel("SKLM_Estimate")
plt.ylabel("Frequência")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.tight_layout()
plt.show()
def loocv_sklm():
    preds = np.zeros(len(r))
```

```
    for i in range(len(r)):
        dists, idxs = tree.query(coords[i], k=min(K+1, len(r)))
        idxs = np.atleast_1d(idxs)
        idxs = idxs[idxs != i][:K]
        pts = coords[idxs]
        dd = pts[:,None,:] - pts[None,:,:]
        hij = np.sqrt((dd[:,:,0]**2) + (dd[:,:,1]**2))
        C = cov_from_gamma(hij, NUGGET, SILL_PARTIAL, RANGE)
        C.flat[::C.shape[0]+1] += 1e-10*(NUGGET+SILL_PARTIAL)
        ht = np.sqrt(((pts - coords[i])**2).sum(axis=1))
        c = cov_from_gamma(ht, NUGGET, SILL_PARTIAL, RANGE)
        w = np.linalg.solve(C, c)
        rhat = float(np.dot(w, r[idxs]))
        preds[i] = df["Local_Mean_Primary"].iloc[i]  + rhat
    return preds
pred = loocv_sklm()
pred = np.maximum(pred, 0.0)
obs = df[COL_P].to_numpy(float)
rmse = np.sqrt(np.mean((pred-obs)**2))
mae = np.mean(np.abs(pred-obs))
r2 = 1 - np.sum((obs-pred)**2)/np.sum((obs-obs.mean())**2)
print("LOOCV: R²=%.3f | RMSE=%.3f | MAE=%.3f" % (r2, rmse, mae))
plt.figure(figsize=(6,6))
plt.scatter(obs, pred, s=18)
mn, mx = min(obs.min(), pred.min()), max(obs.max(), pred.max())
plt.plot([mn,mx],[mn,mx], linewidth=1)
plt.xlabel("Observado"); plt.ylabel("Predito (LOOCV)")
plt.title("SKLM LOOCV")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.show()
def section_profile(df_obs, grid, axis="X", value=100.0, tol=1.0):
    """
    axis: 'X' ou 'Y'
    value: coordenada da seção
```

```
    tol: meia largura da seção
    """
    if axis == "X":
        obs = df_obs[np.abs(df_obs["X"] - value) <= tol].copy()
        grd = grid[np.abs(grid["X"] - value) <= tol].copy()
        coord = "Y"
    else:
        obs = df_obs[np.abs(df_obs["Y"] - value) <= tol].copy()
        grd = grid[np.abs(grid["Y"] - value) <= tol].copy()
        coord = "X"
    obs = obs.sort_values(coord)
    grd = grd.sort_values(coord)
    return obs, grd, coord
import numpy as np
import matplotlib.pyplot as plt
PRED_COL = "SKLM_Estimate"
def section_profile(df_obs, grid, axis="X", value=100.0, tol=1.0):
    axis = axis.upper()
    if axis == "X":
        obs = df_obs[np.abs(df_obs["X"] - value) <= tol].copy()
        grd = grid[np.abs(grid["X"] - value) <= tol].copy()
        coord = "Y"
    else:
        obs = df_obs[np.abs(df_obs["Y"] - value) <= tol].copy()
        grd = grid[np.abs(grid["Y"] - value) <= tol].copy()
        coord = "X"
    obs = obs.sort_values(coord)
    grd = grd.sort_values(coord)
    return obs, grd, coord
def pick_sections_with_data(df_obs, axis="X", n=5, base_tol=1.0, min_obs=3,
max_tol=10.0):
    """
    Escolhe n seções distribuídas no domínio, mas 'encaixa' cada seção
    no valor de X (ou Y) mais próximo que realmente tem observações.
```

```
Se ainda assim não tiver min_obs, aumenta tol até max_tol.
"""
axis = axis.upper()
vals_obs = np.sort(df_obs[axis].unique())
targets = np.quantile(vals_obs, np.linspace(0.1, 0.9, n))
chosen = []
chosen_tol = []
for t in targets:
    v = vals_obs[np.argmin(np.abs(vals_obs  - t))]
    tol = base_tol
    while tol <= max_tol:
        if axis == "X":
            nobs = (np.abs(df_obs["X"] - v) <= tol).sum()
        else:
            nobs = (np.abs(df_obs["Y"] - v) <= tol).sum()
        if nobs >= min_obs:
            break
        tol += base_tol
    chosen.append(float(v))
    chosen_tol.append(float(tol))
out = []
out_tol = []
for v, tol in zip(chosen, chosen_tol):
    if v not in out:
        out.append(v)
        out_tol.append(tol)
if len(out) < n:
    extra = np.linspace(vals_obs.min(), vals_obs.max(), n)
    for t in extra:
        v = vals_obs[np.argmin(np.abs(vals_obs  - t))]
        if v not in out:
            out.append(float(v))
            out_tol.append(float(base_tol))
        if len(out) == n:
```

```
            break
    return out[:n], out_tol[:n]
N_SECTIONS = 5
BASE_TOL = 1.0
MIN_OBS = 3
MAX_TOL = 10.0
sec_x, tol_x = pick_sections_with_data(df, axis="X", n=N_SECTIONS,
base_tol=BASE_TOL, min_obs=MIN_OBS, max_tol=MAX_TOL)
sec_y, tol_y = pick_sections_with_data(df, axis="Y", n=N_SECTIONS,
base_tol=BASE_TOL, min_obs=MIN_OBS, max_tol=MAX_TOL)
print("Seções em X (valor, tol):", list(zip(sec_x, tol_x)))
print("Seções em Y (valor, tol):", list(zip(sec_y, tol_y)))
fig, axes = plt.subplots(nrows=N_SECTIONS, ncols=2, figsize=(12, 2.6*N_SECTIONS),
sharex=False)
for i, (vx, tx) in enumerate(zip(sec_x, tol_x)):
    obs_sec, grd_sec, coord = section_profile(df, grid, axis="X", value=vx, tol=tx)
    ax = axes[i, 0]
    ax.plot(grd_sec[coord], grd_sec[PRED_COL], "-", label=f"Predito ({PRED_COL})")
    ax.scatter(obs_sec[coord], obs_sec["Primary"], c="k", s=25, label="Observado")
    ax.set_title(f"Seção X = {vx:.2f} ± {tx:.1f}")
    ax.set_xlabel(coord + " (m)")
    ax.set_ylabel("Au")
    ax.grid(True, linestyle=":")
    if i == 0: ax.legend()
for i, (vy, ty) in enumerate(zip(sec_y, tol_y)):
    obs_sec, grd_sec, coord = section_profile(df, grid, axis="Y", value=vy, tol=ty)
    ax = axes[i, 1]
    ax.plot(grd_sec[coord], grd_sec[PRED_COL], "-", label=f"Predito ({PRED_COL})")
    ax.scatter(obs_sec[coord], obs_sec["Primary"], c="k", s=25, label="Observado")
    ax.set_title(f"Seção Y = {vy:.2f} ± {ty:.1f}")
    ax.set_xlabel(coord + " (m)")
    ax.set_ylabel("Au")
    ax.grid(True, linestyle=":")
    if i == 0: ax.legend()
```

```
plt.tight_layout()
plt.show()
fig.savefig("secoes_sklm_5x5.png", dpi=200, bbox_inches="tight")
print("Salvo: secoes_sklm_5x5.png")
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
import numpy as np
import matplotlib.pyplot as plt
S = df[[COL_S]].to_numpy(float)
P = df[COL_P].to_numpy(float)
rf = RandomForestRegressor(
    n_estimators=400,
    min_samples_leaf=5,
    random_state=42,
    n_jobs=-1
)
rf.fit(S, P)
P_hat = rf.predict(S)
print("RF (média local): P = f(S)")
print("R² (ajuste no dataset) =", round(r2_score(P, P_hat), 3))
plt.figure(figsize=(6,5))
plt.scatter(S[:,0], P, s=18, label="Observado")
sline = np.linspace(S[:,0].min(), S[:,0].max(), 300).reshape(-1,1)
plt.plot(sline[:,0], rf.predict(sline), linewidth=2, label="RF (média local)")
plt.xlabel(COL_S); plt.ylabel(COL_P)
plt.title("Scatter Primary × Secondary + RF (média local)")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.legend()
plt.show()
df["Local_Mean_Primary_RF"] = rf.predict(df[[COL_S]].to_numpy(float))
grid["Local_Mean_Primary_RF"] = rf.predict(grid[[COL_S]].to_numpy(float))
plt.figure(figsize=(6,5))
plt.scatter(df[COL_X], df[COL_Y], c=df["Local_Mean_Primary_RF"], s=25)
plt.colorbar(label="Local_Mean_Primary_RF (no dataset)")
```

```
plt.title("Local Mean Primary (RF) no dataset (pontos)")
plt.xlabel("X"); plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
plt.figure(figsize=(6,5))
plt.scatter(grid[COL_X], grid[COL_Y], c=grid["Local_Mean_Primary_RF"], s=5)
plt.colorbar(label="Local_Mean_Primary_RF  (no grid)")
plt.title("Local Mean Primary (RF) no grid (exaustivo)")
plt.xlabel("X"); plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
df["Residual_RF"] = df[COL_P] - df["Local_Mean_Primary_RF"]
print(df["Residual_RF"].describe())
plt.figure(figsize=(6,4))
plt.hist(df["Residual_RF"], bins=25)
plt.title("Histograma do resíduo (RF)")
plt.xlabel("Residual_RF"); plt.ylabel("freq")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
res = df["Residual_RF"]
print(res.describe())
print("\nVariância:", res.var())
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import cross_val_predict
from sklearn.metrics import r2_score
X_sec = df[["Secondary"]].to_numpy()
y = df["Primary"].to_numpy()
rf = RandomForestRegressor(
    n_estimators=300,
    max_depth=None,
    min_samples_leaf=5,
    random_state=42
)
rf.fit(X_sec, y)
```

```
y_pred_rf = rf.predict(X_sec)
print("R² RF (ajuste):", r2_score(y, y_pred_rf))
df["Local_Mean_Primary_RF"] = rf.predict(df[["Secondary"]])
grid["Local_Mean_Primary_RF"]  = rf.predict(grid[["Secondary"]])
df[["Primary", "Local_Mean_Primary_RF"]].head()
df["Residual_RF"] = df["Primary"] - df["Local_Mean_Primary_RF"]
print("Média do resíduo RF:", df["Residual_RF"].mean())
print("Std do resíduo RF:", df["Residual_RF"].std())
print("Variancia do resíduo RF:", df["Residual_RF"].var())
coords = df[[COL_X, COL_Y]].to_numpy(float)
r = df["Residual_RF"].to_numpy(float)
n = len(r)
def experimental_variogram(coords, values, lag=10.0, hmax=200.0, azimuth=None,
tol=22.5):
    dx = coords[:,0][:,None] - coords[:,0][None,:]
    dy = coords[:,1][:,None] - coords[:,1][None,:]
    h = np.sqrt(dx*dx + dy*dy)
    az = (np.degrees(np.arctan2(dy, dx)) + 360) % 180
    gamma = 0.5*(values[:,None] - values[None,:])**2
    iu = np.triu_indices(len(values), k=1)
    h_u, az_u, g_u = h[iu], az[iu], gamma[iu]
    if azimuth is not None:
        diff = np.abs(az_u - azimuth)
        diff = np.minimum(diff, 180-diff)
        mdir = diff <= tol
        h_u, g_u = h_u[mdir], g_u[mdir]
    edges = np.arange(0, hmax+lag, lag)
    centers = (edges[:-1] + edges[1:]) / 2
    gam = np.full(len(centers), np.nan)
    pairs = np.zeros(len(centers), int)
    for k in range(len(centers)):
        m = (h_u >= edges[k]) & (h_u < edges[k+1])
        pairs[k] = int(m.sum())
        if pairs[k] > 0:
```

```
            gam[k] = float(np.mean(g_u[m]))
    return centers, gam, pairs
LAG = 10.0
HMAX = 300.0
hc, g_omni, p_omni = experimental_variogram(coords, r, lag=LAG, hmax=HMAX)
plt.figure(figsize=(8,5))
plt.plot(hc, g_omni, marker="o", linewidth=1, label="omni")
for azm in [0,45,90,135]:
    hc2, g_dir, p_dir = experimental_variogram(coords, r, lag=LAG, hmax=HMAX,
azimuth=azm, tol=22.5)
    plt.plot(hc2, g_dir, marker="o", linewidth=1, label=f"az={azm}°")
plt.xlabel("h (m)"); plt.ylabel("γ(h) resíduos")
plt.title("Variogramas experimentais (resíduos)")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.legend()
plt.show()
NUGGET = 0.10
SILL_PARTIAL = 0.12
RANGE = 75.0
def sph_gamma(h, nug, psill, a):
    h = np.asarray(h, float)
    r = h/a
    out = np.empty_like(r)
    inside = r < 1
    out[inside] = nug + psill*(1.5*r[inside] - 0.5*r[inside]**3)
    out[~inside] = nug + psill
    return out
def cov_from_gamma(h, nug, psill, a):
    return (nug + psill) - sph_gamma(h, nug, psill, a)
plt.figure(figsize=(8,5))
plt.plot(hc, g_omni, marker="o", linewidth=1, label="exp omni")
plt.plot(hc, sph_gamma(hc, NUGGET, SILL_PARTIAL, RANGE), linewidth=2,
label="modelo esférico")
plt.xlabel("h (m)"); plt.ylabel("γ(h) resíduos")
```

```
plt.title("Variograma dos resíduos: experimental vs modelo")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.legend()
plt.show()
tree = cKDTree(coords)
K = 12
def skl_residual_predict(query_xy):
    dists, idxs = tree.query(query_xy, k=min(K, len(r)))
    if np.ndim(idxs)==1:
        idxs = idxs[:,None]
    r_pred = np.zeros(len(query_xy), float)
    for i in range(len(query_xy)):
        inds = np.atleast_1d(idxs[i])
        pts = coords[inds]
        dd = pts[:,None,:] - pts[None,:,:]
        hij = np.sqrt((dd[:,:,0]**2) + (dd[:,:,1]**2))
        C = cov_from_gamma(hij, NUGGET, SILL_PARTIAL, RANGE)
        C.flat[::C.shape[0]+1] += 1e-10*(NUGGET+SILL_PARTIAL)
        ht = np.sqrt(((pts - query_xy[i])**2).sum(axis=1))
        c = cov_from_gamma(ht, NUGGET, SILL_PARTIAL, RANGE)
        w = np.linalg.solve(C, c)
        r_pred[i] = float(np.dot(w, r[inds]))
    return r_pred
grid_xy = grid[[COL_X, COL_Y]].to_numpy(float)
grid["Residual_RF"] = skl_residual_predict(grid_xy)
grid["SKLM_RF"] = grid["Local_Mean_Primary_RF"] + grid["Residual_RF"]
grid[["SKLM_RF"]].head()
out_csv = "SKLM_RF_grid_results.csv"
grid.to_csv(out_csv, index=False)
files.download(out_csv)
plt.figure(figsize=(6,5))
plt.scatter(grid[COL_X], grid[COL_Y], c=grid["SKLM_RF"], s=5)
plt.colorbar(label="Au_SKLM_RF (ppm)")
plt.title("SKLM RF (grid)")
```

```
plt.xlabel("X"); plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
def loocv_sklm_rf():
    preds = np.zeros(len(r))
    for i in range(len(r)):
        dists, idxs = tree.query(coords[i], k=min(K+1, len(r)))
        idxs = np.atleast_1d(idxs)
        idxs = idxs[idxs != i][:K]
        pts = coords[idxs]
        dd = pts[:,None,:] - pts[None,:,:]
        hij = np.sqrt((dd[:,:,0]**2) + (dd[:,:,1]**2))
        C = cov_from_gamma(hij, NUGGET, SILL_PARTIAL, RANGE)
        C.flat[::C.shape[0]+1] += 1e-10*(NUGGET+SILL_PARTIAL)
        ht = np.sqrt(((pts - coords[i])**2).sum(axis=1))
        c = cov_from_gamma(ht, NUGGET, SILL_PARTIAL, RANGE)
        w = np.linalg.solve(C, c)
        rhat = float(np.dot(w, r[idxs]))
        preds[i] = df["Local_Mean_Primary_RF"].iloc[i] + rhat
    return preds
pred_rf = loocv_sklm_rf()
pred_rf = np.maximum(pred_rf, 0.0)
obs = df[COL_P].to_numpy(float)
rmse_rf = np.sqrt(np.mean((pred_rf-obs)**2))
mae_rf  = np.mean(np.abs(pred_rf-obs))
r2_rf   = 1 - np.sum((obs-pred_rf)**2)/np.sum((obs-obs.mean())**2)
print("LOOCV SKLM RF: R²=%.3f | RMSE=%.3f | MAE=%.3f" % (r2_rf, rmse_rf, mae_rf))
plt.figure(figsize=(6,6))
plt.scatter(obs, pred_rf, s=18)
mn, mx = min(obs.min(), pred_rf.min()), max(obs.max(), pred_rf.max())
plt.plot([mn,mx],[mn,mx], linewidth=1)
plt.xlabel("Observado")
plt.ylabel("Predito (LOOCV RF)")
plt.title("SKLM RF – LOOCV")
```

```
plt.grid(True, linestyle=":", linewidth=0.6)
plt.show()
import numpy as np
import matplotlib.pyplot as plt
PRED_COL = "SKLM_RF"
if PRED_COL not in grid.columns:
    raise ValueError(f"Coluna {PRED_COL} não existe no grid. Veja grid.columns:
{grid.columns.tolist()}")
def section_profile(df_obs, grid, axis="X", value=100.0, tol=1.0):
    axis = axis.upper()
    if axis == "X":
        obs = df_obs[np.abs(df_obs["X"] - value) <= tol].copy()
        grd = grid[np.abs(grid["X"] - value) <= tol].copy()
        coord = "Y"
    else:
        obs = df_obs[np.abs(df_obs["Y"] - value) <= tol].copy()
        grd = grid[np.abs(grid["Y"] - value) <= tol].copy()
        coord = "X"
    obs = obs.sort_values(coord)
    grd = grd.sort_values(coord)
    return obs, grd, coord
def pick_sections_with_data(df_obs, axis="X", n=5, base_tol=1.0, min_obs=3,
max_tol=10.0):
    axis = axis.upper()
    vals_obs = np.sort(df_obs[axis].unique())
    targets = np.quantile(vals_obs, np.linspace(0.1, 0.9, n))
    chosen = []
    chosen_tol = []
    for t in targets:
        v = vals_obs[np.argmin(np.abs(vals_obs - t))]
        tol = base_tol
        while tol <= max_tol:
            if axis == "X":
                nobs = (np.abs(df_obs["X"] - v) <= tol).sum()
```

```
            else:
                nobs = (np.abs(df_obs["Y"] - v) <= tol).sum()
            if nobs >= min_obs:
                break
            tol += base_tol
        chosen.append(float(v))
        chosen_tol.append(float(tol))
    out = []
    out_tol = []
    for v, tol in zip(chosen, chosen_tol):
        if v not in out:
            out.append(v)
            out_tol.append(tol)
    if len(out) < n:
        extra = np.linspace(vals_obs.min(), vals_obs.max(), n)
        for t in extra:
            v = vals_obs[np.argmin(np.abs(vals_obs - t))]
            if v not in out:
                out.append(float(v))
                out_tol.append(float(base_tol))
            if len(out) == n:
                break
    return out[:n], out_tol[:n]
N_SECTIONS = 5
BASE_TOL = 1.0
MIN_OBS = 3
MAX_TOL = 10.0
sec_x, tol_x = pick_sections_with_data(df, axis="X", n=N_SECTIONS,
base_tol=BASE_TOL, min_obs=MIN_OBS, max_tol=MAX_TOL)
sec_y, tol_y = pick_sections_with_data(df, axis="Y", n=N_SECTIONS,
base_tol=BASE_TOL, min_obs=MIN_OBS, max_tol=MAX_TOL)
print("Seções em X (valor, tol):", list(zip(sec_x, tol_x)))
print("Seções em Y (valor, tol):", list(zip(sec_y, tol_y)))
fig, axes = plt.subplots(nrows=N_SECTIONS, ncols=2, figsize=(12, 2.6*N_SECTIONS),
```

```
sharex=False)
for i, (vx, tx) in enumerate(zip(sec_x, tol_x)):
    obs_sec, grd_sec, coord = section_profile(df, grid, axis="X", value=vx, tol=tx)
    ax = axes[i, 0]
    ax.plot(grd_sec[coord], grd_sec[PRED_COL], "-", label=f"Predito ({PRED_COL})")
    ax.scatter(obs_sec[coord], obs_sec["Primary"], c="k", s=25, label="Observado")
    ax.set_title(f"Seção X = {vx:.2f} ± {tx:.1f}")
    ax.set_xlabel(coord + " (m)")
    ax.set_ylabel("Au")
    ax.grid(True, linestyle=":")
    if i == 0: ax.legend()
for i, (vy, ty) in enumerate(zip(sec_y, tol_y)):
    obs_sec, grd_sec, coord = section_profile(df, grid, axis="Y", value=vy, tol=ty)
    ax = axes[i, 1]
    ax.plot(grd_sec[coord], grd_sec[PRED_COL], "-", label=f"Predito ({PRED_COL})")
    ax.scatter(obs_sec[coord], obs_sec["Primary"], c="k", s=25, label="Observado")
    ax.set_title(f"Seção Y = {vy:.2f} ± {ty:.1f}")
    ax.set_xlabel(coord + " (m)")
    ax.set_ylabel("Au")
    ax.grid(True, linestyle=":")
    if i == 0: ax.legend()
plt.tight_layout()
plt.show()
fig.savefig("secoes_sklm_rf_5x5.png", dpi=200, bbox_inches="tight")
print("Salvo: secoes_sklm_rf_5x5.png")
S = df[[COL_S]].to_numpy(float)
P = df[COL_P].to_numpy(float)
nn = Pipeline([
    ("scaler", StandardScaler()),
    ("mlp", MLPRegressor(
        hidden_layer_sizes=(20, 20),
        activation="relu",
        solver="adam",
        alpha=1e-3,
```

```
        max_iter=5000,
        random_state=42
    ))
])
nn.fit(S, P)
P_hat_nn = nn.predict(S)
print("Rede Neural (média local): P = f(S)")
print("R² (ajuste no dataset) =", round(r2_score(P, P_hat_nn), 3))
plt.figure(figsize=(6,5))
plt.scatter(S[:,0], P, s=18, label="Observado")
sline = np.linspace(S[:,0].min(), S[:,0].max(), 300).reshape(-1,1)
plt.plot(sline[:,0], nn.predict(sline), linewidth=2, label="Rede Neural")
plt.xlabel(COL_S)
plt.ylabel(COL_P)
plt.title("Scatter Primary × Secondary + Rede Neural (média local)")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.legend()
plt.show()
df["Local_Mean_Primary_NN"] = nn.predict(df[[COL_S]].to_numpy(float))
grid["Local_Mean_Primary_NN"] = nn.predict(grid[[COL_S]].to_numpy(float))
plt.figure(figsize=(6,5))
plt.scatter(df[COL_X], df[COL_Y], c=df["Local_Mean_Primary_NN"], s=25)
plt.colorbar(label="Local_Mean_Primary_NN (dataset)")
plt.title("Local Mean Primary (NN) no dataset")
plt.xlabel("X"); plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
plt.figure(figsize=(6,5))
plt.scatter(grid[COL_X], grid[COL_Y], c=grid["Local_Mean_Primary_NN"], s=5)
plt.colorbar(label="Local_Mean_Primary_NN (grid)")
plt.title("Local Mean Primary (NN) no grid")
plt.xlabel("X"); plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
```

```
df["Residual_NN"] = df[COL_P] - df["Local_Mean_Primary_NN"]
print("Média do resíduo NN:", df["Residual_NN"].mean())
print("Std do resíduo NN:", df["Residual_NN"].std())
print("Variancia do resíduo NN:", df["Residual_NN"].var())
plt.figure(figsize=(6,4))
plt.hist(df["Residual_NN"], bins=25)
plt.title("Histograma do resíduo (Rede Neural)")
plt.xlabel("Residual_NN")
plt.ylabel("freq")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
coords = df[[COL_X, COL_Y]].to_numpy(float)
r = df["Residual_NN"].to_numpy(float)
n = len(r)
def experimental_variogram(coords, values, lag=10.0, hmax=200.0, azimuth=None,
tol=22.5):
    dx = coords[:,0][:,None] - coords[:,0][None,:]
    dy = coords[:,1][:,None] - coords[:,1][None,:]
    h = np.sqrt(dx*dx + dy*dy)
    az = (np.degrees(np.arctan2(dy, dx)) + 360) % 180
    gamma = 0.5*(values[:,None] - values[None,:])**2
    iu = np.triu_indices(len(values), k=1)
    h_u, az_u, g_u = h[iu], az[iu], gamma[iu]
    if azimuth is not None:
        diff = np.abs(az_u - azimuth)
        diff = np.minimum(diff, 180-diff)
        mdir = diff <= tol
        h_u, g_u = h_u[mdir], g_u[mdir]
    edges = np.arange(0, hmax+lag, lag)
    centers = (edges[:-1] + edges[1:]) / 2
    gam = np.full(len(centers), np.nan)
    pairs = np.zeros(len(centers), int)
    for k in range(len(centers)):
        m = (h_u >= edges[k]) & (h_u < edges[k+1])
```

```
        pairs[k] = int(m.sum())
        if pairs[k] > 0:
            gam[k] = float(np.mean(g_u[m]))
    return centers, gam, pairs
LAG = 10.0
HMAX = 300.0
hc, g_omni, p_omni = experimental_variogram(coords, r, lag=LAG, hmax=HMAX)
plt.figure(figsize=(8,5))
plt.plot(hc, g_omni, marker="o", linewidth=1, label="omni")
for azm in [0,45,90,135]:
    hc2, g_dir, p_dir = experimental_variogram(coords, r, lag=LAG, hmax=HMAX,
azimuth=azm, tol=22.5)
    plt.plot(hc2, g_dir, marker="o", linewidth=1, label=f"az={azm}°")
plt.xlabel("h (m)"); plt.ylabel("γ(h) resíduos")
plt.title("Variogramas experimentais (resíduos)")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.legend()
plt.show()
NUGGET = 0.10
SILL_PARTIAL = 0.17
RANGE = 100.0
def sph_gamma(h, nug, psill, a):
    h = np.asarray(h, float)
    r = h/a
    out = np.empty_like(r)
    inside = r < 1
    out[inside] = nug + psill*(1.5*r[inside] - 0.5*r[inside]**3)
    out[~inside] = nug + psill
    return out
def cov_from_gamma(h, nug, psill, a):
    return (nug + psill) - sph_gamma(h, nug, psill, a)
plt.figure(figsize=(8,5))
plt.plot(hc, g_omni, marker="o", linewidth=1, label="exp omni")
plt.plot(hc, sph_gamma(hc, NUGGET, SILL_PARTIAL, RANGE), linewidth=2,
```

```
label="modelo esférico")
plt.xlabel("h (m)"); plt.ylabel("γ(h) resíduos")
plt.title("Variograma dos resíduos: experimental vs modelo")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.legend()
plt.show()
from scipy.spatial import cKDTree
import numpy as np
tree = cKDTree(coords)
K = 12
def skl_residual_predict_nn(query_xy):
    dists, idxs = tree.query(query_xy, k=min(K, len(r)))
    if np.ndim(idxs) == 1:
        idxs = idxs[:, None]
    r_pred = np.zeros(len(query_xy), float)
    for i in range(len(query_xy)):
        inds = np.atleast_1d(idxs[i])
        pts = coords[inds]
        dd = pts[:, None, :] - pts[None, :, :]
        hij = np.sqrt(dd[:, :, 0]**2 + dd[:, :, 1]**2)
        C = cov_from_gamma(hij, NUGGET, SILL_PARTIAL, RANGE)
        C.flat[::C.shape[0] + 1] += 1e-10 * (NUGGET + SILL_PARTIAL)
        ht = np.sqrt(((pts - query_xy[i])**2).sum(axis=1))
        c = cov_from_gamma(ht, NUGGET, SILL_PARTIAL, RANGE)
        w = np.linalg.solve(C, c)
        r_pred[i] = float(np.dot(w, r[inds]))
    return r_pred
grid_xy = grid[[COL_X, COL_Y]].to_numpy(float)
grid["Residual_SK_NN"] = skl_residual_predict_nn(grid_xy)
grid["SKLM_NN"] = grid["Local_Mean_Primary_NN"] + grid["Residual_SK_NN"]
grid[["SKLM_NN"]].head()
out_csv = "SKLM_NN_grid_results.csv"
grid.to_csv(out_csv, index=False)
files.download(out_csv)
```

```
plt.figure(figsize=(6,5))
plt.scatter(
    grid[COL_X],
    grid[COL_Y],
    c=grid["SKLM_NN"],
    s=5
)
plt.colorbar(label="Au_SKLM_NN (ppm)")
plt.title("SKLM com Média Local via Rede Neural (NN)")
plt.xlabel("X")
plt.ylabel("Y")
plt.grid(True, linestyle=":", linewidth=0.4)
plt.show()
df["Residual_NN"] = df[COL_P] - df["Local_Mean_Primary_NN"]
r = df["Residual_NN"].to_numpy(float)
coords = df[[COL_X, COL_Y]].to_numpy(float)
tree = cKDTree(coords)
K = 12
def loocv_sklm_nn():
    preds = np.zeros(len(r))
    for i in range(len(r)):
        dists, idxs = tree.query(coords[i], k=min(K+1, len(r)))
        idxs = np.atleast_1d(idxs)
        idxs = idxs[idxs != i][:K]
        pts = coords[idxs]
        dd = pts[:, None, :] - pts[None, :, :]
        hij = np.sqrt(dd[:, :, 0]**2 + dd[:, :, 1]**2)
        C = cov_from_gamma(hij, NUGGET, SILL_PARTIAL, RANGE)
        C.flat[::C.shape[0] + 1] += 1e-10 * (NUGGET + SILL_PARTIAL)
        ht = np.sqrt(((pts - coords[i])**2).sum(axis=1))
        c = cov_from_gamma(ht, NUGGET, SILL_PARTIAL, RANGE)
        w = np.linalg.solve(C, c)
        rhat = float(np.dot(w, r[idxs]))
        preds[i] = df["Local_Mean_Primary_NN"].iloc[i] + rhat
```

```
    return preds
pred_nn = loocv_sklm_nn()
pred_nn = np.maximum(pred_nn, 0.0)
obs = df[COL_P].to_numpy(float)
rmse_nn = np.sqrt(np.mean((pred_nn - obs)**2))
mae_nn  = np.mean(np.abs(pred_nn  - obs))
r2_nn   = 1 - np.sum((obs  - pred_nn)**2) / np.sum((obs - obs.mean())**2)
print("LOOCV SKLM-NN: R²=%.3f | RMSE=%.3f | MAE=%.3f" % (r2_nn, rmse_nn,
mae_nn))
plt.figure(figsize=(6,6))
plt.scatter(obs, pred_nn, s=18)
mn, mx = min(obs.min(), pred_nn.min()), max(obs.max(), pred_nn.max())
plt.plot([mn, mx], [mn, mx], linewidth=1)
plt.xlabel("Observado")
plt.ylabel("Predito (LOOCV – SKLM NN)")
plt.title("Validação Cruzada – SKLM com Rede Neural")
plt.grid(True, linestyle=":", linewidth=0.6)
plt.show()
import numpy as np
import matplotlib.pyplot as plt
PRED_COL = "SKLM_NN"
if PRED_COL not in grid.columns:
    raise ValueError(
        f"Coluna {PRED_COL} não existe no grid. "
        f"Veja grid.columns: {grid.columns.tolist()}"
    )
def section_profile(df_obs, grid, axis="X", value=100.0, tol=1.0):
    axis = axis.upper()
    if axis == "X":
        obs = df_obs[np.abs(df_obs["X"] - value) <= tol].copy()
        grd = grid[np.abs(grid["X"] - value) <= tol].copy()
        coord = "Y"
    else:
        obs = df_obs[np.abs(df_obs["Y"] - value) <= tol].copy()
```

```
        grd = grid[np.abs(grid["Y"] - value) <= tol].copy()
        coord = "X"
    obs = obs.sort_values(coord)
    grd = grd.sort_values(coord)
    return obs, grd, coord
def pick_sections_with_data(df_obs, axis="X", n=5, base_tol=1.0, min_obs=3,
max_tol=10.0):
    axis = axis.upper()
    vals_obs = np.sort(df_obs[axis].unique())
    targets = np.quantile(vals_obs, np.linspace(0.1, 0.9, n))
    chosen = []
    chosen_tol = []
    for t in targets:
        v = vals_obs[np.argmin(np.abs(vals_obs - t))]
        tol = base_tol
        while tol <= max_tol:
            if axis == "X":
                nobs = (np.abs(df_obs["X"] - v) <= tol).sum()
            else:
                nobs = (np.abs(df_obs["Y"] - v) <= tol).sum()
            if nobs >= min_obs:
                break
            tol += base_tol
        chosen.append(float(v))
        chosen_tol.append(float(tol))
    out, out_tol = [], []
    for v, tol in zip(chosen, chosen_tol):
        if v not in out:
            out.append(v)
            out_tol.append(tol)
    if len(out) < n:
        extra = np.linspace(vals_obs.min(), vals_obs.max(), n)
        for t in extra:
            v = vals_obs[np.argmin(np.abs(vals_obs - t))]
```

```
            if v not in out:
                out.append(float(v))
                out_tol.append(float(base_tol))
            if len(out) == n:
                break
    return out[:n], out_tol[:n]
N_SECTIONS = 5
BASE_TOL = 1.0
MIN_OBS = 3
MAX_TOL = 10.0
sec_x, tol_x = pick_sections_with_data(
    df, axis="X", n=N_SECTIONS,
    base_tol=BASE_TOL, min_obs=MIN_OBS, max_tol=MAX_TOL
)
sec_y, tol_y = pick_sections_with_data(
    df, axis="Y", n=N_SECTIONS,
    base_tol=BASE_TOL, min_obs=MIN_OBS, max_tol=MAX_TOL
)
print("Seções em X (valor, tol):", list(zip(sec_x, tol_x)))
print("Seções em Y (valor, tol):", list(zip(sec_y, tol_y)))
fig, axes = plt.subplots(
    nrows=N_SECTIONS, ncols=2,
    figsize=(12, 2.6*N_SECTIONS),
    sharex=False
)
for i, (vx, tx) in enumerate(zip(sec_x, tol_x)):
    obs_sec, grd_sec, coord = section_profile(df, grid, axis="X", value=vx, tol=tx)
    ax = axes[i, 0]
    ax.plot(grd_sec[coord], grd_sec[PRED_COL], "-", label="Predito (SKLM-NN)")
    ax.scatter(obs_sec[coord], obs_sec["Primary"], c="k", s=25, label="Observado")
    ax.set_title(f"Seção X = {vx:.2f} ± {tx:.1f}")
    ax.set_xlabel(coord + " (m)")
    ax.set_ylabel("Au")
    ax.grid(True, linestyle=":")
```

```
    if i == 0:
        ax.legend()
for i, (vy, ty) in enumerate(zip(sec_y, tol_y)):
    obs_sec, grd_sec, coord = section_profile(df, grid, axis="Y", value=vy, tol=ty)
    ax = axes[i, 1]
    ax.plot(grd_sec[coord], grd_sec[PRED_COL], "-", label="Predito (SKLM-NN)")
    ax.scatter(obs_sec[coord], obs_sec["Primary"], c="k", s=25, label="Observado")
    ax.set_title(f"Seção Y = {vy:.2f} ± {ty:.1f}")
    ax.set_xlabel(coord + " (m)")
    ax.set_ylabel("Au")
    ax.grid(True, linestyle=":")
    if i == 0:
        ax.legend()
plt.tight_layout()
plt.show()
fig.savefig("secoes_sklm_nn_5x5.png", dpi=200, bbox_inches="tight")
print("Salvo: secoes_sklm_nn_5x5.png")
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import r2_score
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.neural_network import MLPRegressor
S1 = df[COL_S].to_numpy(float)
S = df[[COL_S]].to_numpy(float)
P = df[COL_P].to_numpy(float)
sline = np.linspace(S1.min(), S1.max(), 400)
sline2d = sline.reshape(-1, 1)
b1 = np.cov(S1, P, ddof=1)[0, 1] / (np.var(S1, ddof=1) + 1e-12)
b0 = P.mean() - b1 * S1.mean()
P_lin_line = b0 + b1 * sline
print("Regressão (média local): P = b0 + b1*S")
print("b0 =", b0)
```

```
print("b1 =", b1)
rf = RandomForestRegressor(
    n_estimators=400,
    min_samples_leaf=5,
    random_state=42,
    n_jobs=-1
)
rf.fit(S, P)
P_hat_rf = rf.predict(S)
P_rf_line = rf.predict(sline2d)
print("\nRF (média local): P = f(S)")
print("R² (ajuste no dataset) =", round(r2_score(P, P_hat_rf), 3))
nn = Pipeline([
    ("scaler", StandardScaler()),
    ("mlp", MLPRegressor(
        hidden_layer_sizes=(20, 20),
        activation="relu",
        solver="adam",
        alpha=1e-3,
        max_iter=5000,
        random_state=42
    ))
])
nn.fit(S, P)
P_hat_nn = nn.predict(S)
P_nn_line = nn.predict(sline2d)
print("\nRede Neural (média local): P = f(S)")
print("R² (ajuste no dataset) =", round(r2_score(P, P_hat_nn), 3))
fig, axes = plt.subplots(1, 3, figsize=(16, 4.5), sharey=True)
axes[0].scatter(S1, P, s=18, alpha=0.7, label="Observado")
axes[0].plot(sline, P_lin_line, linewidth=2.5, label="Linear")
axes[0].set_title("Linear (média local)")
axes[0].set_xlabel(COL_S)
axes[0].set_ylabel(COL_P)
```

```
axes[0].grid(True, linestyle=":", linewidth=0.6)
axes[0].legend()
axes[1].scatter(S1, P, s=18, alpha=0.7, label="Observado")
axes[1].plot(sline, P_rf_line, linewidth=2.5, label="RF")
axes[1].set_title("Random Forest (média local)")
axes[1].set_xlabel(COL_S)
axes[1].grid(True, linestyle=":", linewidth=0.6)
axes[1].legend()
axes[2].scatter(S1, P, s=18, alpha=0.7, label="Observado")
axes[2].plot(sline, P_nn_line, linewidth=2.5, label="Rede Neural")
axes[2].set_title("Rede Neural (média local)")
axes[2].set_xlabel(COL_S)
axes[2].grid(True, linestyle=":", linewidth=0.6)
axes[2].legend()
plt.suptitle("Primary × Secondary — comparação da média local (Linear × RF × NN)")
plt.tight_layout(rect=[0, 0, 1, 0.92])
plt.show()
fig.savefig("scatter_primary_secondary_LIN_RF_NN.png", dpi=200, bbox_inches="tight")
print("\nSalvo: scatter_primary_secondary_LIN_RF_NN.png")
import matplotlib.pyplot as plt
import numpy as np
residuals = {
    "SKLM Linear": df["Residual"].to_numpy(float),
    "SKLM RF":     df["Residual_RF"].to_numpy(float),
    "SKLM NN":     df["Residual_NN"].to_numpy(float),
}
fig, axes = plt.subplots(1, 3, figsize=(14,4), sharey=True)
for ax, (title, r) in zip(axes, residuals.items()):
    ax.hist(r, bins=25, edgecolor="k", alpha=0.8)
    ax.axvline(0.0, color="k", linestyle="--", linewidth=1)
    ax.set_title(title)
    ax.set_xlabel("Resíduo (Au)")
    ax.set_ylabel("Frequência")
    ax.grid(True, linestyle=":", linewidth=0.6)
```

```
plt.suptitle("Distribuição dos resíduos — SKLM Linear × RF × NN")
plt.tight_layout()
plt.show()
import pandas as pd
import numpy as np
def metrics(obs, pred):
    rmse = np.sqrt(np.mean((pred-obs)**2))
    mae  = np.mean(np.abs(pred-obs))
    r2   = 1 - np.sum((obs-pred)**2)/np.sum((obs-obs.mean())**2)
    return rmse, mae, r2
obs = df[COL_P].to_numpy(float)
rmse_lin, mae_lin, r2_lin = metrics(obs, pred)
rmse_rf, mae_rf, r2_rf = metrics(obs, pred_rf)
rmse_nn, mae_nn, r2_nn = metrics(obs, pred_nn)
metrics_df = pd.DataFrame({
    "Modelo": ["SKLM Linear", "SKLM RF", "SKLM NN"],
    "RMSE": [rmse_lin, rmse_rf, rmse_nn],
    "MAE": [mae_lin, mae_rf, mae_nn],
    "R²": [r2_lin, r2_rf, r2_nn]
})
metrics_df
import matplotlib.pyplot as plt
fig, axes = plt.subplots(1, 3, figsize=(15,5), sharex=True, sharey=True)
models = [
    ("SKLM Linear", pred),
    ("SKLM RF",     pred_rf),
    ("SKLM NN",     pred_nn)
]
mn = min(obs.min(), pred.min(), pred_rf.min(), pred_nn.min())
mx = max(obs.max(), pred.max(), pred_rf.max(), pred_nn.max())
for ax, (title, pred) in zip(axes, models):
    ax.scatter(obs, pred, s=20)
    ax.plot([mn, mx], [mn, mx], "k--")
    ax.set_title(title)
```

```
    ax.set_xlabel("Observado")
    ax.set_ylabel("Predito")
    ax.grid(True, linestyle=":")
plt.suptitle("Validação Cruzada (LOOCV): Observado × Predito")
plt.tight_layout()
plt.show()
NONNEG_METHOD = "limit_residual"
FLOOR = 0.0
def enforce_nonneg(grid, sklm_col, lm_col, res_col=None, method="limit_residual",
floor=0.0, keep_raw=True):
    if sklm_col not in grid.columns:
        print(f"[AVISO] {sklm_col} não existe no grid.")
        return
    if lm_col not in grid.columns:
        print(f"[AVISO] {lm_col} não existe no grid (não dá para limitar resíduo).")
        if method == "clip0":
            grid[sklm_col] = np.maximum(grid[sklm_col].to_numpy(float),  floor)
        return
    if keep_raw and f"{sklm_col}_raw" not in grid.columns:
        grid[f"{sklm_col}_raw"] = grid[sklm_col]
    if method == "limit_residual" and res_col is not None and res_col in grid.columns:
        if keep_raw and f"{res_col}_raw" not in grid.columns:
            grid[f"{res_col}_raw"] = grid[res_col]
        lm = grid[lm_col].to_numpy(float)
        r  = grid[res_col].to_numpy(float)
        r_limited = np.maximum(r,  floor - lm)
        grid[res_col]  = r_limited
        grid[sklm_col] = lm + r_limited
    if method == "clip0":
        grid[sklm_col] = np.maximum(grid[sklm_col].to_numpy(float),  floor)
enforce_nonneg(grid, sklm_col="SKLM_Estimate",  lm_col="Local_Mean_Primary",
res_col="Residual_SK",
            method=NONNEG_METHOD, floor=FLOOR, keep_raw=True)
enforce_nonneg(grid, sklm_col="SKLM_RF", lm_col="Local_Mean_Primary_RF",
```

```
res_col="Residual_RF",
            method=NONNEG_METHOD, floor=FLOOR, keep_raw=True)
enforce_nonneg(grid, sklm_col="SKLM_NN", lm_col="Local_Mean_Primary_NN",
res_col="Residual_SK_NN",
            method=NONNEG_METHOD, floor=FLOOR, keep_raw=True)
for col in ["SKLM_Estimate", "SKLM_RF", "SKLM_NN"]:
    if col in grid.columns:
        v = grid[col].to_numpy(float)
        print(f"{col}: % negativos = {100*np.mean(v<FLOOR):.2f}% | min={v.min():.6f}")
fig, axes = plt.subplots(1, 3, figsize=(15,5), sharex=True, sharey=True)
vmin = min(
    grid["SKLM_Estimate"].min(),
    grid["SKLM_RF"].min(),
    grid["SKLM_NN"].min()
)
vmax = max(
    grid["SKLM_Estimate"].max(),
    grid["SKLM_RF"].max(),
    grid["SKLM_NN"].max()
)
models = [
    ("SKLM Linear", "SKLM_Estimate"),
    ("SKLM RF",     "SKLM_RF"),
    ("SKLM NN",     "SKLM_NN")
]
sc = None
for ax, (title, col) in zip(axes, models):
    sc = ax.scatter(
        grid[COL_X], grid[COL_Y],
        c=grid[col],
        s=5,
        vmin=vmin,
        vmax=vmax,
        cmap="viridis"
```

```
    )
    ax.set_title(title)
    ax.set_xlabel("X")
    ax.set_ylabel("Y")
    ax.grid(True, linestyle=":")
cbar = fig.colorbar(
    sc,
    ax=axes,
    orientation="vertical",
    fraction=0.03,
    pad=0.04
)
cbar.set_label("Au (ppm)")
fig, axes = plt.subplots(1, 3, figsize=(15,4), sharey=True)
errors = [
    ("SKLM Linear", pred - obs),
    ("SKLM RF",     pred_rf  - obs),
    ("SKLM NN",     pred_nn  - obs)
]
for ax, (title, err) in zip(axes, errors):
    ax.hist(err, bins=25)
    ax.axvline(0, color="k", linestyle="--")
    ax.set_title(title)
    ax.set_xlabel("Erro")
    ax.set_ylabel("Frequência")
    ax.grid(True, linestyle=":")
plt.suptitle("Distribuição dos erros (LOOCV)")
plt.tight_layout()
plt.show()
SECTION_AXIS = "X"
SECTION_VALUE = 120.0
SECTION_TOL = 1.0
obs_sec, grd_sec, coord = section_profile(df, grid, axis=SECTION_AXIS,
value=SECTION_VALUE, tol=SECTION_TOL)
```

```
plt.figure(figsize=(10,4))
plt.plot(grd_sec[coord], grd_sec["SKLM_Estimate"], label="SKLM Linear")
plt.plot(grd_sec[coord], grd_sec["SKLM_RF"],  label="SKLM RF")
plt.plot(grd_sec[coord], grd_sec["SKLM_NN"],  label="SKLM NN")
plt.scatter(obs_sec[coord], obs_sec[COL_P], c="k", s=30, label="Observado")
plt.xlabel(coord)
plt.ylabel("Au")
plt.title(f"Seção {SECTION_AXIS} = {SECTION_VALUE} ± {SECTION_TOL}")
plt.grid(True, linestyle=":")
plt.legend()
plt.show()
import numpy as np
import matplotlib.pyplot as plt
GRID_COLS = {
    "Linear": "SKLM_Estimate",
    "RF":     "SKLM_RF",
    "NN":     "SKLM_NN",
}
for k, col in GRID_COLS.items():
    if col not in grid.columns:
        raise ValueError(f"Falta '{col}' no grid para o modelo '{k}'. Colunas:
{grid.columns.tolist()}")
def section_profile(df_obs, grid, axis="X", value=100.0, tol=1.0):
    axis = axis.upper()
    if axis == "X":
        obs = df_obs[np.abs(df_obs["X"] - value) <= tol].copy()
        grd = grid[np.abs(grid["X"] - value) <= tol].copy()
        coord = "Y"
    else:
        obs = df_obs[np.abs(df_obs["Y"] - value) <= tol].copy()
        grd = grid[np.abs(grid["Y"] - value) <= tol].copy()
        coord = "X"
    return obs.sort_values(coord), grd.sort_values(coord), coord
def pick_sections_with_data(df_obs, axis="X", n=3, base_tol=1.0, min_obs=3,
```

```
max_tol=10.0):
    axis = axis.upper()
    vals_obs = np.sort(df_obs[axis].unique())
    targets = np.quantile(vals_obs, np.linspace(0.2, 0.8, n))
    out, out_tol = [], []
    for t in targets:
        v = vals_obs[np.argmin(np.abs(vals_obs  - t))]
        tol = base_tol
        while tol <= max_tol:
            nobs = (np.abs(df_obs[axis] - v) <= tol).sum()
            if nobs >= min_obs:
                break
            tol += base_tol
        if float(v) not in out:
            out.append(float(v)); out_tol.append(float(tol))
    return out[:n], out_tol[:n]
N_SECTIONS = 3
BASE_TOL, MIN_OBS, MAX_TOL = 1.0, 3, 10.0
sec_x, tol_x = pick_sections_with_data(df, axis="X", n=N_SECTIONS,
base_tol=BASE_TOL, min_obs=MIN_OBS, max_tol=MAX_TOL)
sec_y, tol_y = pick_sections_with_data(df, axis="Y", n=N_SECTIONS,
base_tol=BASE_TOL, min_obs=MIN_OBS, max_tol=MAX_TOL)
fig, axes = plt.subplots(nrows=N_SECTIONS, ncols=2, figsize=(12, 2.8*N_SECTIONS),
sharex=False)
for i, (vx, tx) in enumerate(zip(sec_x,  tol_x)):
    obs_sec, grd_sec, coord = section_profile(df, grid, axis="X", value=vx, tol=tx)
    ax = axes[i, 0]
    for name, col in GRID_COLS.items():
        ax.plot(grd_sec[coord], grd_sec[col], "-", label=name if i == 0 else None, lw=1.6)
    ax.scatter(obs_sec[coord], obs_sec[COL_P], c="k", s=25, label="Observado" if i == 0 else
None, zorder=3)
    ax.set_title(f"Seção X = {vx:.2f} ± {tx:.1f}")
    ax.set_xlabel(f"{coord} (m)"); ax.set_ylabel("Au")
    ax.grid(True, linestyle=":")
```

```
for i, (vy, ty) in enumerate(zip(sec_y, tol_y)):
    obs_sec, grd_sec, coord = section_profile(df, grid, axis="Y", value=vy, tol=ty)
    ax = axes[i, 1]
    for name, col in GRID_COLS.items():
        ax.plot(grd_sec[coord], grd_sec[col], "-", label=name if i == 0 else None, lw=1.6)
    ax.scatter(obs_sec[coord], obs_sec[COL_P], c="k", s=25, label="Observado" if i == 0 else
None, zorder=3)
    ax.set_title(f"Seção Y = {vy:.2f} ± {ty:.1f}")
    ax.set_xlabel(f"{coord} (m)"); ax.set_ylabel("Au")
    ax.grid(True, linestyle=":")
handles, labels = axes[0,0].get_legend_handles_labels()
fig.legend(handles, labels, loc="upper center", ncol=4, frameon=True)
plt.tight_layout(rect=[0, 0, 1, 0.95])
plt.show()
fig.savefig("secoes_comparativas_LIN_RF_NN_3secoes.png", dpi=200,
bbox_inches="tight")
print("Salvo: secoes_comparativas_LIN_RF_NN_3secoes.png")
import numpy as np
import pandas as pd
from google.colab import files
COL_LIN_LM = "Local_Mean_Primary"
COL_LIN_SK = "Residual_SK"
COL_LIN_SKLM = "SKLM_Estimate"
COL_RF_LM = "Local_Mean_Primary_RF"
COL_RF_SK = "Residual_SK_RF"
COL_RF_SKLM = "SKLM_RF"
COL_NN_LM = "Local_Mean_Primary_NN"
COL_NN_SK = "Residual_SK_NN"
COL_NN_SKLM = "SKLM_NN"
def ensure_residual_points(df, lm_col, out_col):
    if out_col not in df.columns:
        df[out_col] = df[COL_P].to_numpy(float) - df[lm_col].to_numpy(float)
def save_csv(df_out, name):
    df_out.to_csv(name, index=False)
```

```
    print("Salvo:", name)
    files.download(name)
def poly_equation_from_model(predict_fn, s_min, s_max, degree=5, n=600):
    """
    Aproxima P=f(S) por polinômio (grau=degree) no intervalo [s_min, s_max].
    Retorna coeficientes (maior grau -> constante) e string da equação.
    """
    sline = np.linspace(s_min, s_max, n).reshape(-1,1)
    y = predict_fn(sline).ravel()
    coef = np.polyfit(sline.ravel(), y, deg=degree)
    terms = []
    d = degree
    for i, a in enumerate(coef):
        p = d - i
        if p == 0:
            terms.append(f"{a:.6g}")
        elif p == 1:
            terms.append(f"{a:.6g}*S")
        else:
            terms.append(f"{a:.6g}*S^{p}")
    eq = "P ≈ " + " + ".join(terms)
    return coef, eq
S_pts = df[COL_S].to_numpy(float)
P_pts = df[COL_P].to_numpy(float)
b1 = np.cov(S_pts, P_pts, ddof=1)[0,1] / (np.var(S_pts, ddof=1) + 1e-12)
b0 = P_pts.mean() - b1*S_pts.mean()
eq_lin = f"P = {b0:.6g} + {b1:.6g}*S"
ensure_residual_points(df, COL_LIN_LM, "Residual_LIN")
ensure_residual_points(df, COL_RF_LM, "Residual_RF")
ensure_residual_points(df, COL_NN_LM, "Residual_NN")
save_csv(
    grid[[COL_X, COL_Y, COL_S, COL_LIN_LM]].copy(),
    "grid_localmean_linear.csv"
)
```

```
save_csv(
    df[[COL_X, COL_Y, COL_P, COL_S, "Residual_LIN"]].copy(),
    "points_residual_linear.csv"
)
if COL_LIN_SK in grid.columns:
    save_csv(
        grid[[COL_X, COL_Y, COL_LIN_SK]].copy(),
        "grid_residualSK_linear.csv"
    )
if COL_LIN_SKLM in grid.columns:
    save_csv(
        grid[[COL_X, COL_Y, COL_LIN_SKLM]].copy(),
        "grid_sklm_linear.csv"
    )
save_csv(
    grid[[COL_X,COL_Y,COL_S, COL_RF_LM]].copy(),
    "grid_localmean_rf.csv"
)
save_csv(
    df[[COL_X, COL_Y, COL_P, COL_S, "Residual_RF"]].copy(),
    "points_residual_rf.csv"
)
if COL_RF_SK in grid.columns:
    save_csv(
        grid[[COL_X, COL_Y, COL_RF_SK]].copy(),
        "grid_residualSK_rf.csv"
    )
if COL_RF_SKLM in grid.columns:
    save_csv(
        grid[[COL_X, COL_Y, COL_RF_SKLM]].copy(),
        "grid_sklm_rf.csv"
    )
save_csv(
    grid[[COL_X, COL_Y, COL_S, COL_NN_LM]].copy(),
```

```
    "grid_localmean_nn.csv"
)
save_csv(
    df[[COL_X, COL_Y, COL_P, COL_S, "Residual_NN"]].copy(),
    "points_residual_nn.csv"
)
if COL_NN_SK in grid.columns:
    save_csv(
        grid[[COL_X, COL_Y, COL_NN_SK]].copy(),
        "grid_residualSK_nn.csv"
    )
if COL_NN_SKLM in grid.columns:
    save_csv(
        grid[[COL_X, COL_Y, COL_NN_SKLM]].copy(),
        "grid_sklm_nn.csv"
    )
eq_rows = []
eq_rows.append(["Linear", "exata", eq_lin])
if "rf" in globals():
    coef_rf, eq_rf = poly_equation_from_model(
        predict_fn=lambda s: rf.predict(s),
        s_min=float(S_pts.min()),
        s_max=float(S_pts.max()),
        degree=5
    )
    eq_rows.append(["RF", "aprox_polynomial_deg5", eq_rf])
    pd.DataFrame({"coef_rf_deg5": coef_rf}).to_csv("eq_rf_poly_deg5_coeffs.csv",
index=False)
    files.download("eq_rf_poly_deg5_coeffs.csv")
else:
    eq_rows.append(["RF", "N/A", "Modelo 'rf' não encontrado no notebook."])
if "nn" in globals():
    coef_nn, eq_nn = poly_equation_from_model(
        predict_fn=lambda s: nn.predict(s),
```

```
        s_min=float(S_pts.min()),
        s_max=float(S_pts.max()),
        degree=5
    )
    eq_rows.append(["NN", "aprox_polynomial_deg5", eq_nn])
    pd.DataFrame({"coef_nn_deg5": coef_nn}).to_csv("eq_nn_poly_deg5_coeffs.csv",
index=False)
    files.download("eq_nn_poly_deg5_coeffs.csv")
else:
    eq_rows.append(["NN", "N/A", "Modelo 'nn' não encontrado no notebook."])
eq_df = pd.DataFrame(eq_rows, columns=["Tecnica", "Tipo_equacao", "Equacao"])
eq_df.to_csv("equacoes_primary_secondary_por_tecnica.csv", index=False)
print(eq_df)
files.download("equacoes_primary_secondary_por_tecnica.csv")
```