Predição de Estabilidade para Smart Grid

Victor Jorge Carvalho Chaves 159740, João Vinicius Farah Colombini 159501

***Abstract*—Com o crescimento do consumo de energia elétrica pelo mundo, é essencial garantir sua disponibilidade e monitoramento, e para isso, é proposto um trabalho que vá prever a estabilidade de uma rede elétrica.**

***Index Terms*—Rede Elétricas Inteligentes, Aprendizado de Máquina**

I. Introdução

As redes elétricas modernas enfrentam desafios crescentes de estabilidade e confiabilidade devido à integração de fontes de energia renováveis e à complexidade das demandas de carga. Neste contexto, este projeto propõe a aplicação de algoritmos de aprendizado de máquina, especificamente técnicas de regressão em aprendizado supervisionado, para estimar com precisão a estabilidade desses sistemas elétricos. Ao utilizar métodos avançados de regressão, busca-se desenvolver modelos capazes de prever valores contínuos que representam indicadores críticos de estabilidade.

Além do uso de regressão é muito eficaz procurar uma série de parâmetros para descrever o funcionamento desses sistemas elétricos e para isso, após uma análise com regressão, também é implementado um Algoritmo Genético para tentar solucionar esse problema.

II. Contexto

Na área de desenvolvimento de soluções inteligentes para aprendizado sobre o comportamento de redes elétricas inteligentes (Smart Grid), foi desenvolvido uma base de dados simulada[8] de uma rede elétrica controlada de forma descentralizada.

A partir desse trabalho, foi realizado diversos trabalhos cientificos utilizando destes dados, todas com foco em resolver tarefas de classificação da estabilidade da rede.

Com isso, este estudo tem como objetivo achar uma solução que resolva a tarefa de regressão da estabilidade da rede.

As váriaveis que há nessa base são as seguintes:

- `tau[x]`: Tempo de reação do participante (valor real no intervalo [0.5, 10] segundos).
  - `tau1` - valor para o produtor de eletricidade;
- `p[x]`: Potência nominal consumida (negativa) ou produzida (positiva) (valor real).
  - Para consumidores, no intervalo [-0.5, -2] segundos^{-2};
  - `p1` = abs(`p2` + `p3` + `p4`);
- `g[x]`: Coeficiente (γ) proporcional à elasticidade de preço (valor real no intervalo [0.05, 1] segundos^{-1}).
  - `g1` - valor para o produtor de eletricidade;
- `stab`: A parte real máxima da raiz da equação característica (se positiva - o sistema é linearmente instável) (valor real).
- `stabf`: A classificação de estabilidade do sistema (categórica: estável/instável).

III. Metodologia

A. Principais Técnologias

As principais bibliotecas utilizadas foram:

- Pandas: Manipulação e análise de dados
- Numpy: Manipulação e análise de dados
- MatplotLib: Visualização
- seaborn: Visualização
- sklearn: Biblioteca de aprendizado de máquina
- lightgbm: Modelo de aprendizado de máquina utilizando o método
- xgboost
- PyGAD: Biblioteca para implementação de algoritmos genéticos.
- Torch: Biblioteca para implementação de redes neurais.

B. Principais Métricas e Avaliação

- **MSE (Mean Squared Error):** Mede o erro quadrático médio entre as previsões e os valores reais. Penaliza mais fortemente grandes erros, pois os erros são elevados ao quadrado antes da média ser calculada.
- **MAE (Mean Absolute Error):** Mede o erro absoluto médio entre as previsões e os valores reais. É mais robusto a outliers, pois considera apenas a magnitude dos erros, sem elevá-los ao quadrado.
- **RMSE (Root Mean Squared Error):** Mede a raiz quadrada do erro quadrático médio entre as previsões e os valores reais. Ele fornece uma estimativa da magnitude média dos erros cometidos pelo modelo, com a mesma unidade dos dados originais. Assim como o MSE, o RMSE penaliza erros maiores de forma mais intensa, mas por estar na mesma escala dos dados, facilita a interpretação do erro médio do modelo.
- **MAPE (Mean Absolute Percentage Error):** Mede o erro absoluto percentual médio, representando o erro em termos percentuais. É útil para interpretar erros em diferentes escalas, mas pode ser problemático se os valores reais forem próximos de zero.
- R^2 (Coeficiente de Determinação): Mede a proporção da variância nos valores reais que é explicada pelas previsões do modelo. Varia de 0 a 1, onde valores mais próximos de 1 indicam um modelo que explica bem a variabilidade dos dados.

C. Extração dos dados

Os dados foram extraídos do Repositório de Aprendizado de Máquina da UC Irvine.

D. Análise na Correlação entre as váriaveis

Para análise da correlação em relação ao banco de dados, o método utilizado foi o de correlação de pearson que varia de -1 a 1 Sendo que 1 é correlação positiva forte e -1 é correlação negativa forte, 0 corresponde a zero correlação.

Para apresentar de maneira gráfica foi utilizado um mapa de calor que deixa evidente os maiores e menores valores.

E. Competição entre os Modelos de regressão

Para esse projeto foram utilizados 18 métodos para regressão, dos mais básicos aos mais avançados, como os que usam boosting de gradiente ou AdaBoosting.

Entre eles:

- Bayesian Ridge Regression
- Automatic Relevance Determination (ARD) Regression
- Lasso Regression
- Ridge Regression
- Linear Regression
- Support Vector Regression (SVR)
- Nu Support Vector Regression (NuSVR)
- Light Gradient Boosting Machine (LGBM)
- k-Nearest Neighbors Regression (KNN)
- Elastic Net Regression
- AdaBoost Regression
- Stochastic Gradient Descent Regression (SGD)
- Extra Trees Regression
- eXtreme Gradient Boosting (XGBoost)
- Multi-layer Perceptron Regression (MLP)
- Random Forest Regression
- Histogram-based Gradient Boosting Regression
- Gradient Boosting Regression
- Decision Tree Regression

Para avalia-los os métodos utilizados em K-folds foram:

- Erro Absoluto Médio (MAE)
- Erro Quadrático Médio (MSE)
- Erro Percentual Absoluto Médio (MAPE)
- Coeficiente de Determinação (R^2)

F. Filtragem dos modelos

E com os resultados obtidos, foram separados 3 modelos principais:

- Nu Support Vector Regression (NuSVR)
- Light Gradient Boosting Machine (LGBM)
- Histogram-based Gradient Boosting Regression

Nu Support Vector Regression é um tipo de regressão por SVM, que utiliza de um parâmetro Nu que é utilizado como dois limites, o limite superior representa o número de vetores de suporte e como um limite inferior para os pontos dentro da margem.

Light Gradient Boosting Machine, utiliza de combinações referentes a vários modelos individuais nesse caso com árvores de decisão para regressão e posteriormente combina vários desses modelos mais ”fracos” e forma um modelo com resultados mais significativos (ou ”Fortes”).

Histogram-based Gradient Boosting Regression Utiliza a mesma ideia de conceito de boosting de gradiente, mas utilizando histogramas. Isso ocorre da seguinte maneira, ao invés de gerar splits diretamente nos nós das árvores de decisão eles formam bins que são intervalos (assim como em um histograma) e portanto isso simplifica o processo de gerar splits melhores.

Com esses três modelos, foi aplicado a técnica de pesquisa em grade (Grid Search) para encontrar os melhores parâmetros em cada um desses modelos, e assim verificar os melhores resultados.

G. Criação de um modelo de com Algoritmo genético

A ideia principal desse tópico é gerar uma função que dados os parâmetros $x_1, x_2, ..., x_n$ onde $n = 12$, tenham constantes $\lambda_1, \lambda_2, ..., \lambda_n$, que multiplicando esses parâmetros cheguem a um resultado dizendo se aquele sistema elétrico é estável ou instavel.

Para isso foram utilizados os parâmetros de:

- Número de gerações;
- Número de soluções a serem selecionadas para cruzamento;
- Soluções por população;
- Porcentagem de Mutação;
- Número de genes.

Esses parâmetros impactam diretamente no Fitness dessas equações.

Além disso na biblioteca PyGAD tem dois jeitos de realizar o treinamento:

1) Utiliza-se uma equação e um resultado, porém esse modelo foi utilizado com multiplas entradas e apenas um fitness para todas elas;
2) E o método para multivariáveis, mas fica extremamente pesado carregar 8000 equações e 8000 fitness por geração.

E para finalizar foram utilizados métodos para verificação da precisão que esse algoritmo gera.

1) K-fold: Para garantir uma diferença nos resultados e um resultado final melhor e sem overfitting
2) Resumir resultados finais em positivo ou negativo, como por exemplo: $y\_real = 0,0023$ e $y\_pred = 0.04$, os dois seriam iguais pois a ideia é verificar se o número é positivo ou negativo. (Isso foi utilizado pois fica impossivel chegar a um valor muito semelhante para um número muito grande de equações).
3) Fazer cálculos com métricas de erro calcular o erro absoluto e relativo.

IV. Resultados

A. Regressão

Um dos primeiros resultados foi o mapa de calor com as correlações entre as variáveis como apresentado na Figura 1.

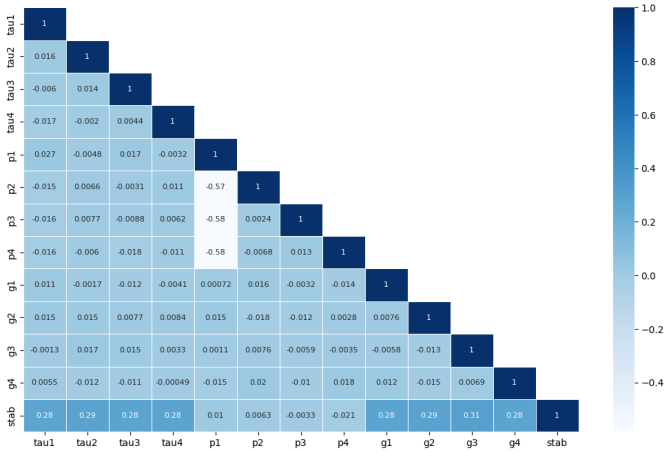

Fig. 1. Mapa de calor - Correlações

Nele ficaram visiveis algumas variaveis com correlação fraca, mas pelo menos apresentam algum tipo de correlação. Fora isso é notável que os g's tem correlação com stab, portanto foi criada uma nova coluna também com as médias de g e ela apresentou a maior correlação positiva com stab que foi de aproximadamente 0.54.

Com base nesses dados os p's não pareciam tão relevantes, pois a busca era por uma correlação positiva forte, o que gerou uma série de dúvidas e algumas decisões como remover ou não os p's para fazer alguma regressão.

Após o tratamento dos dados, como eram simulados não havia a necessidade de preencher neutros nem algo do tipo. Inicia-se a etapa de avaliação de modelos para regressão baseados nesses dados.

Em relação a aqueles 18 modloes foram obtidos os seguintes resultados referentes a R^2 e MAE, MSE, RMSE. (Representadas pelas Figuras 2, 3, 4, 5).

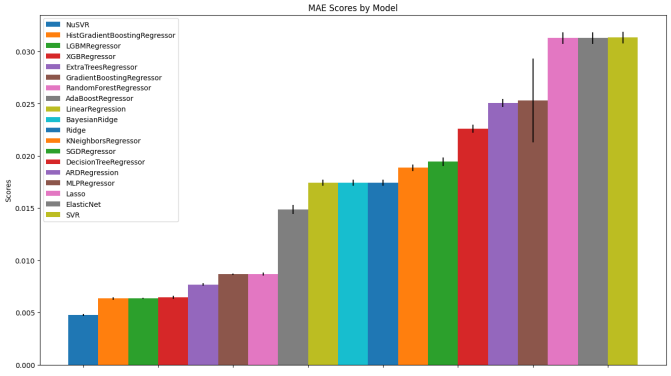

Fig. 2. Resultado MAE dos 18 modelos

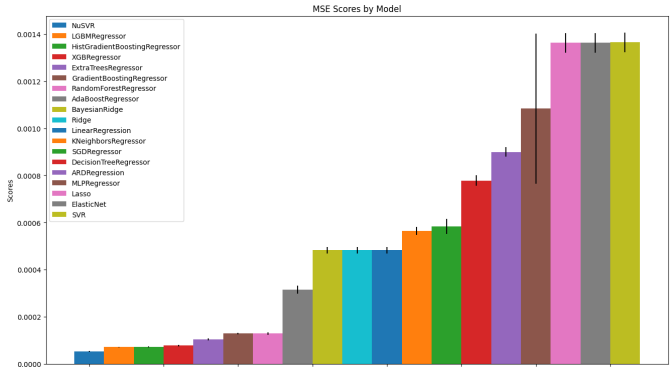

Fig. 3. Resultado MSE dos 18 modelos

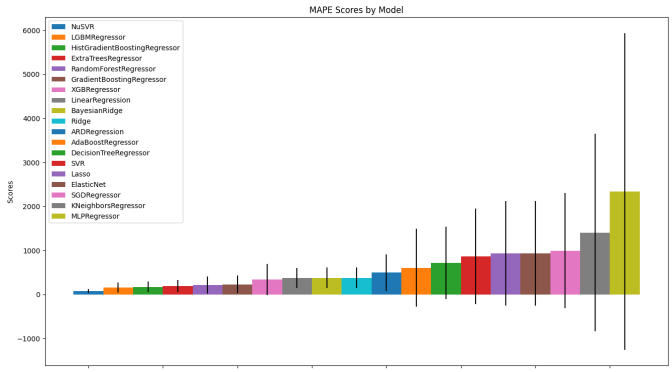

Fig. 4. Resultado MAPE dos 18 modelos

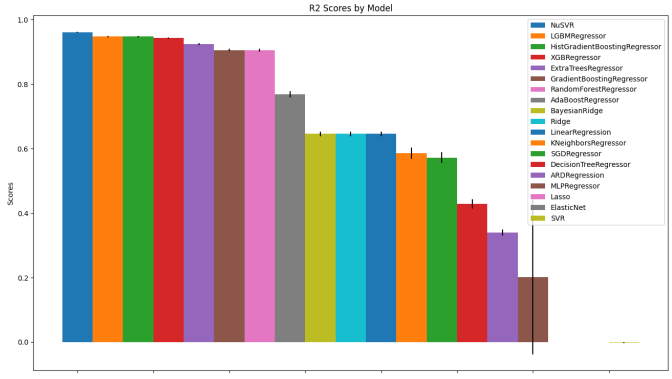

Fig. 5. Resultado R2 dos 18 Modelo

E após isso é simples a visualização de quais escolher, e portanto fazer uma avaliação mais aprofundada.

Como já dito e visualizado os mais performáticos foram: NuSVR, LighGBM e HistGradientBoosting.

E com esses modelos, aplicando uma pesquisa em grade (grid search), para realizer um hiperparâmetro sobre os modelos para alcançar melhores resultados, foi obtido os seguintes valores em MSE, MAPE, MAE e R².

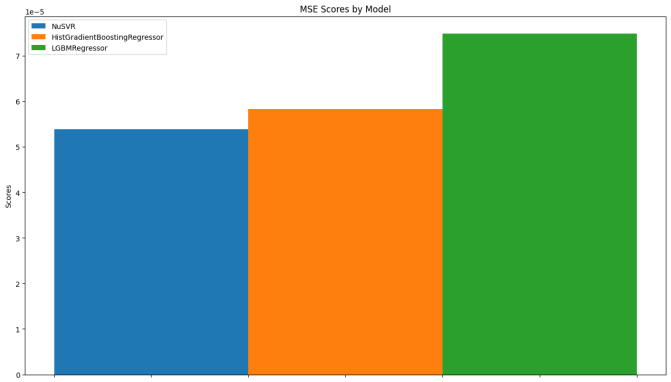

Fig. 6. MSE dos Modelos escolhidos

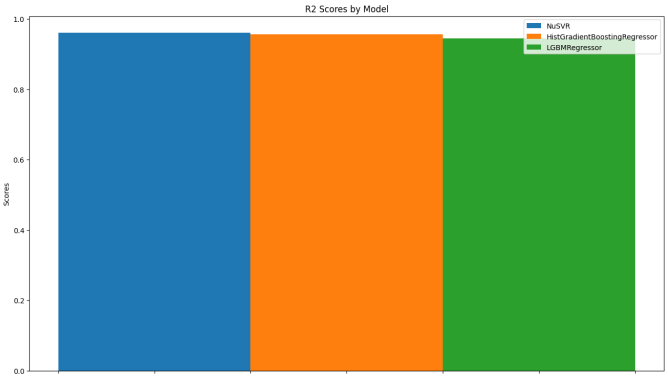

Fig. 9. R² dos modelos escolhidos

E por final também foram criadas visualizações das funções para utilizar da navalha de Occam (ou Ockham) para decidir qual o melhor modelo. (Representadas nas Figuras 10, 11, 12).

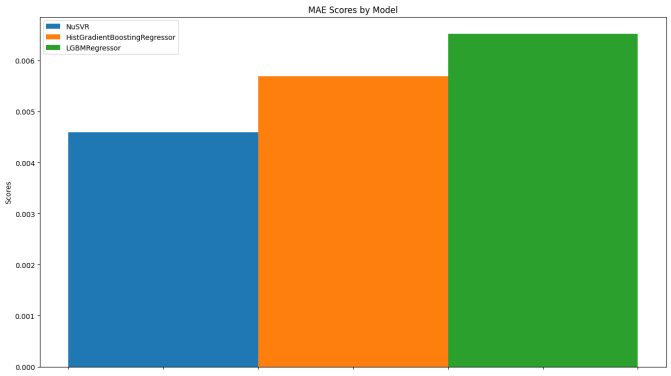

Fig. 7. MAPE dos modelos escolhidos

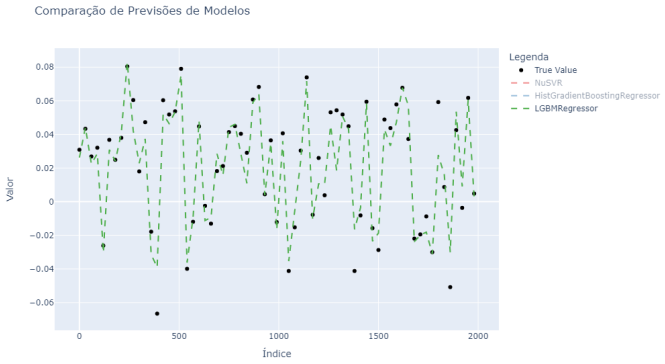

Fig. 10. Função da LightGBMRegressor

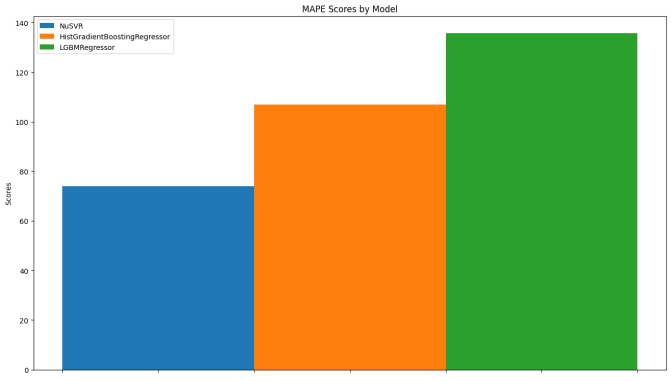

Fig. 8. MAE dos modelos escolhidos

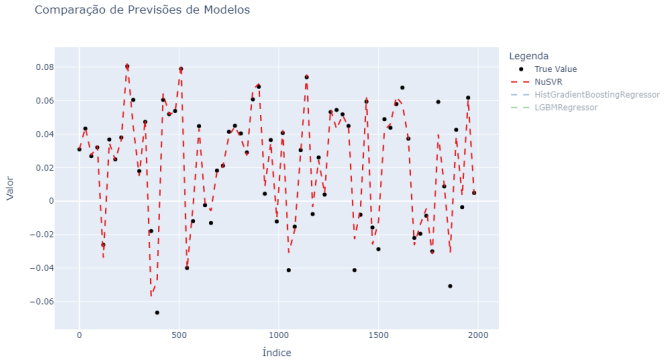

Fig. 11. Função da NuSVR

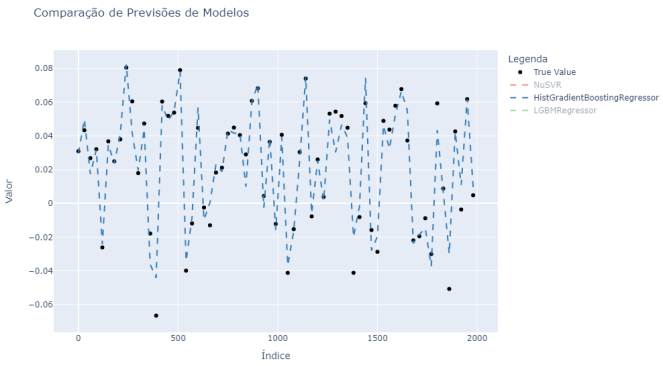

Fig. 12. Função da HistGradientBoostingRegressor

O modelo mais promissor de acordo com esses resultados parece ser o LGBM, por conta que, a função do LGBM aparentar ser e para chegar a uma conclusão é necessária uma discussão mais aprofundada sobre esses dados.

B. Algoritmo Genético

Para o algoritmo genético foram utilizados dois métodos e portanto resultou em alguns resultados diferentes, o primeiro método foi o mais pesado, pois utiliza de várias equações e retorna vários fitness, enquanto o segundo método utiliza várias equações para um fitness só.

Os resultados do primeiro foram:

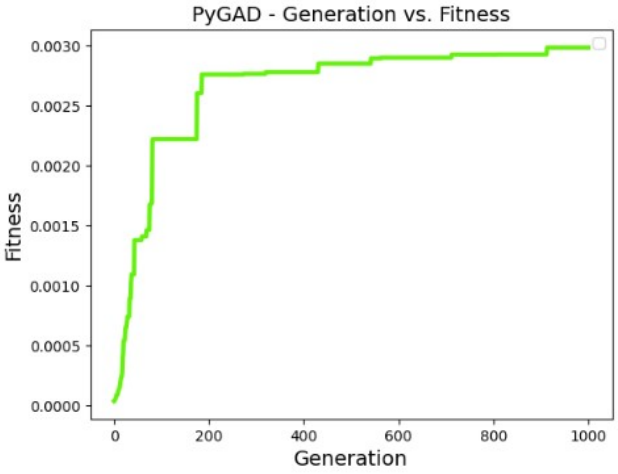

Fig. 13. Gráfico de fitness para AG multivariaveis

Nesse gráfico é possivel verificar que o comportamento do fitness em relação as gerações foi o esperado. Porém tem alguns pontos impotantes sobre esses dados, não foi feito nenhum cross entropy, portanto pode existir overfitting.

As medidas de erro foram:

- **Parâmetros da melhor solução:** [0.00544444, 0.00762837, -0.00338671, -0.00723045, 1.06432844, 1.04125036, 1.07686626, 1.06375533, -0.02777068, -0.01248444, 0.0059718, 0.03094851]
- **Fitness da melhor solução:** 0.0029829439204406482
- **Index da melhor solução:** 0
- **MAE:** 0.04279471565650077
- **MSE:** 0.0028593838969391532
- **RMSE:** 0.05347320728120909
- **MAPE:** 636.6959892684666

Além de ficar evidente os valores das medidas de erro, é possivel também visualizar os valores dos parâmetros utilizados.

E para o AG padrão foram utilizadas 8000 equações por geração com 2000 soluções por geração e 400 separadas para elitização, com 3000 gerações por k fold em k = 5.

O gráfico do fitness em relação as gerações foi:

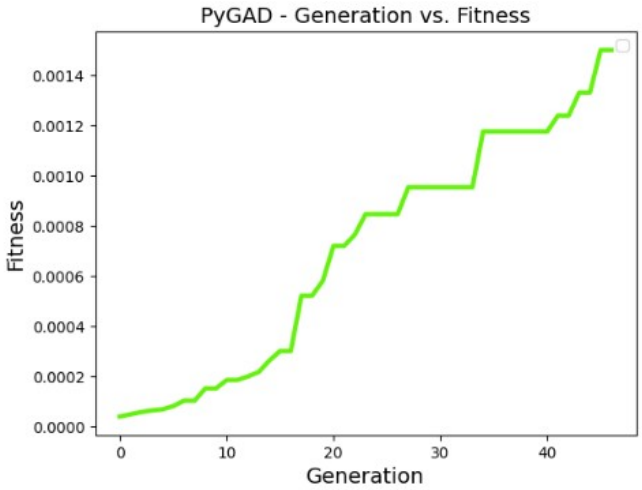

Fig. 14. Gráfico de fitness para AG multivariaveis

Além disso as medidas de erro foram:

- **Parâmetros da melhor solução:** [7.93205155e-03, 4.17784195e-04, 6.45921026-03, 1.92005246e-02, -1.96338026e+00, -1.92890651e+00, -1.99986665e+00, -1.93018442e+00, 7.56142420e-02, 6.09533678e-02, -1.87506567e-01, -1.03931572e-01]
- **Fitness da melhor solução:** 0.001500759904008849
- **Index da melhor solução:** 0
- **MAE:** 0.08038562847020014
- **MSE:** 0.010006329160295811
- **RMSE:** 0.10003164079577927
- **MAPE:** 1064.996180354192

Isso apresenta alguns resultados valiosos, porém com um mape muito alto, e nesse caso quanto menor melhor.

V. Discussões

Ao avaliar o desempenho dos modelos Nu Support Vector Regression (NuSVR), LightGBMRegressor (LGBM) e Histogram-Based Gradient Boosting Regression (HistGradientBoostingRegressor), observa-se que o NuSVR está superando os outros dois em termos de performance. No entanto, este modelo só atinge tais resultados quando utiliza cerca de 90% dos dados como Vetores de Suporte, o que sugere um possível sobreajuste (overfitting). Para mitigar este problema, é recomendável reduzir a proporção de dados utilizados como Vetores de Suporte, visando melhorar a capacidade de generalização do modelo.

Por outro lado, tanto o HistGradientBoostingRegressor quanto o NuSVR demonstram tendência ao sobreajuste em suas atuais configurações.

Em contraste, o LightGBMRegressor se destaca por apresentar uma maior resistência ao overfitting, mantendo um desempenho mais estável e generalizável. Este comportamento sugere que o LightGBMRegressor pode ser uma escolha mais robusta em cenários onde a generalização é crítica, enquanto os outros modelos requerem ajustes mais finos para evitar sobreajuste. Outro ponto importante que reforça a ideia do LightGBMRegressor ser mais geral é ter as medidas de erro mais altas, o que implica que estão ligeiramente mais distantes dos pontos de objetivo, o que significa uma menor aproximação e maior generalização e isso fica ainda mais claro com a visualização da função na imagem 10.

Ficam claros alguns pontos também como a diferença entre os métodos mais simples como regressão linear e árvores de decisão para regressão em relação aos modelos escolhidos, isso é muito proveitoso em relação á otimização e busca de resultados realmente eficientes.

A. Algoritmo Genético

Com os resultados apresentados em relação ao algoritmo genético, algumas informações são importantes, primeiro é que o treino utilizando pygad foi do tipo $\lambda X_1 + \lambda X_2 + ... + \lambda X_n = Y$, portanto não tiveram mudanças nas operações realizadas na equação e sim nos valores de lambda.

Os valores de lambda foram apresentados e ao aplica-los em qualquer linha da base de dados, deveria retornar um stab semelhante. (Para os dois métodos de AG utilizados)

Além disso, é possivel fazer uma comparação dos modelos de AG e suas métricas de erro.

O resultado foi que o MSE MAE e RMSE foi melhor no algoritmo genético, porém o MAPE ficou muito grande para os dois casos.

Olhando pelo grafico e pelos resultados de erro o AG multivariado performou melhor, porém demorou muito mais para rodar, ou seja, seria muito custoso para rodar para as 8000 equações e fazer cross validation assim como o outro método.

O problema do multivariável é que apenas 10 funções das 10000 foram utilizadas como base para criação da equação.

Enquanto isso o AG feito por ultimo usou 80% das equações totais.

Também seria interessante fazer algumas aplicações a mais como colocar maior número de solução por população, mas isso exige demais de um computador, que mesmo rodando em cuda com uma 3050 estava devagar.

Todo o momento que aumentou o numero de solução por população, foi notavel um maior desempenho em relação a fitness.

O que fica claro entre esses modelos é que o que rodou para mais equações teve um resultado geral pior, porém nele também poderiam ter sido feitas alterações e aumentar a quantidade de solução por população, quantidade de gerações e etc... E consequentemente retornar resultados mais eficientes.

VI. Conclusão

Conclui-se que, embora o modelo NuSVR apresente desempenho superior em determinadas condições, sua eficácia está fortemente atrelada a uma configuração que pode levar ao sobreajuste, comprometendo sua capacidade de generalização. Da mesma forma, o HistGradientBoostingRegressor também exibe sinais de overfitting, necessitando de ajustes adicionais para melhorar sua robustez. Em contrapartida, o LightGBMRegressor demonstrou maior resistência ao overfitting, destacando-se como uma opção mais confiável para modelos de regressão que exigem estabilidade e precisão em ambientes de dados variados. Portanto, a escolha do modelo ideal deve considerar não apenas o desempenho imediato, mas também a capacidade de generalização, com o LightGBMRegressor emergindo como uma alternativa sólida em contextos onde a generalização é uma prioridade.

Em relação aos algoritmos genéticos fica possivel concluir que foram encontrados ótimos MAE, MSE e RMSE, porém um MAPE ruim. E além disso o gráfico de fitness da AG multivariavel ficou mais como o desejado de acordo com a literatura, e apresentou um MAPE melhor, portanto o indicado seria aprofundar um pouco no conteúdo desse tipo de AG e melhorar o código para resultados mais otimizados e precisos.

References

[1] Y. Deng, K. K. Cao, W. Hu *et al.*, "Harmonized and Open Energy Dataset for Modeling a Highly Renewable Brazilian Power System," *Scientific Data*, vol. 10, no. 103, 2023, doi: 10.1038/s41597-023-01992-9.

[2] T. Brown, J. Hörsch, and D. Schlachtberger, "PyPSA: Python for Power System Analysis," *Journal of Open Research Software*, vol. 6, no. 1, 2018, arXiv:1707.09913, doi: 10.5334/jors.188.

[3] SAP Insights, "The Smart Grid: How AI is Powering Today's Energy Technologies," Available: *SAP Insights*, Accessed: Jul. 11, 2024.

[4] Md. Satu and Md. Imran Khan, "Machine Learning Approaches To Predict The Stability of Smart Grid," 2024, doi: 10.21203/rs.3.rs-3866218/v1.

[5] Y. Deng, "PyPSA-Brazil: A Free and Open Model of the Brazilian Electrical System," in *Energy Proceedings*, 2021.

[6] X. Zheng, N. Xu, L. Trinh *et al.*, "A multi-scale time-series dataset with benchmark for machine learning in decarbonized energy grids," *Scientific Data*, vol. 9, no. 359, 2022, doi: 10.1038/s41597-022-01455-7.

[7] Y. Shi, G. Ke, D. Soukhavong, J. Lamb, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Liu, N. Titov, and D. Cortes, "lightgbm: Light Gradient Boosting Machine," R package version 4.5.0.99, 2024. Available: https://github.com/Microsoft/LightGBM.

[8] V. Arzamasov, K. Böhm, and P. Jochem, "Towards Concise Models of Grid Stability," in *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, Aalborg, Denmark, 2018, pp. 1-6, doi: 10.1109/SmartGridComm.2018.8587498.

[9] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785-794, doi: 10.1145/2939672.2939785.