

# Introdução

O conjunto de dados de Eficiência Energética (UCI Energy Efficiency) contém 768 amostras com 8 atributos de entrada e dois alvos contínuos: **Y1** (carga de aquecimento) e **Y2** (carga de resfriamento) ([dataset](#)).

O objetivo deste documento é comparar diversos modelos de regressão – incluindo uma regressão linear de referência, quatro modelos *single-output* (treinados separadamente para Y1 e Y2) e o **ExtraTreesRegressor** em modo multi-saída – avaliando métricas clássicas (RMSE, MAE e  $R^2$ ) no conjunto de treino e de teste. Em seguida destacamos o modelo de melhor desempenho, explicamos sua eficiência técnica, e discutimos recomendações para implantação em produção e validação (evitando *overfitting*).

## Modelos avaliados e métricas

Foram testados os principais modelos de regressão no sklearn com *single-output* (um por variável alvo), os melhores colocados Y1 e Y2 foram:

### Y1

- **ExtraTreesRegressor**
- GradientBoostingRegressor
- RandomForestRegressor
- BaggingRegressor

### Y2

- **HistGradientBoostingRegressor**
- GradientBoostingRegressor
- KernelRidge
- BaggingRegressor

Adicionalmente, foi usado o **ExtraTreesRegressor** para previsões multi-saída de Y1 e Y2, em dois cenários: *i*) via **MultiOutputRegressor**(ExtraTreesRegressor) (uma instância por alvo, como meta-estimador) [scikit-learn.org](#) e *ii*) com o próprio ExtraTrees ajustando os dois alvos simultaneamente (suporte multi-saída nativo) [scikit-learn.org](#).

Para cada modelo, utilizamos divisão treino/teste (80%/20%) de forma estratificada, calculando as métricas RMSE (erro quadrático médio), MAE (erro absoluto médio) e  $R^2$  (coeficiente de determinação) em ambos os conjuntos. O RMSE penaliza desvios maiores, enquanto o MAE mede o erro médio absoluto – ambos em mesmas unidades da variável alvo. O  $R^2$  varia entre 0 e 1 e indica a proporção da variância explicada pelo modelo.

## Desempenho comparativo (Y1 e Y2)

As tabelas abaixo resumem as métricas de treino e teste para cada modelo e cada alvo. (Valores obtidos experimentalmente.)

### [Resultados](#)

Na tabela acima observamos que o **ExtraTreesRegressor multi-saída** (nativo) obteve os menores **RMSE** e **MAE** nos conjuntos de teste e os maiores valores de **R<sup>2</sup>** para ambos os alvos Y1 e Y2. Por exemplo, para Y2 o ExtraTrees multi alcançou  $R^2 \approx 0,97$  em teste, indicando que cerca de 97% da variação dos dados de resfriamento foi explicada pelo modelo, e RMSE/teste abaixo de 1.8. Entre os modelos *single-output*, o Gradient Boosting teve desempenho comparável ( $R^2 \approx 0,96$ ), mas ainda ligeiramente inferior. A regressão linear apresentou o pior desempenho geral ( $R^2$  em torno de 0,85–0,87), como esperado por sua limitação a relações lineares.

## Modelo de melhor desempenho

O modelo de **ExtraTreesRegressor multi-output** destaca-se como o mais eficiente, considerando simultaneamente Y1 e Y2 e as métricas de erro.

Assim, o ExtraTreesRegressor multi-output atinge os menores erros e maior  $R^2$  em teste para ambos os alvos, refletindo sua capacidade de aprender padrões complexos e correlacionados. Em suma, ele apresenta o melhor **equilíbrio acurácia/robustez** entre os modelos avaliados.

## Desenvolvimento do sistema

Com base nos resultados, recomenda-se empregar o **ExtraTreesRegressor** (versão multi-output) em produção. Para criar o sistema de previsão proposto pela AT3, as seguintes diretrizes são sugeridas:

- **Uso do modelo em modo multi-output:** Implantar uma única instância de ExtraTreesRegressor treinada com os dois alvos, de modo que ela receba um vetor de atributos de entrada e retorne simultaneamente Y1 e Y2. Como visto, o modelo nativo já aceita **y** com forma **(n\_amostras, 2)** [scikit-learn.org](https://scikit-learn.org). Se fosse necessário usar modelos *single-output*, poderia-se envolver o estimador em **MultiOutputRegressor** do sklearn [scikit-learn.org](https://scikit-learn.org), mas não há necessidade aqui.
- **Separação Treinamento vs Inferência:** O treinamento do modelo deve ser realizado off-line (em ambiente separado) usando dados históricos e validação cruzada para ajustar hiperparâmetros. Após o modelo ser finalizado, ele deve ser serializado e implantado em produção **separadamente** do processo de treinamento. Dessa forma, o modelo só faz inferências em dados novos e não re-treina em

produção, seguindo as práticas padrão de MLops [techtarget.com](https://techtarget.com), [aws](https://aws.com)

Dessa forma, o modelo é usado em produção somente para **inferência**, em ambiente controlado e autenticado, enquanto o processo de treinamento permanece isolado. Isto garante que dados de produção novos são tratados corretamente pelo modelo já ajustado, conforme diferenciado no ciclo de inferência.

## Validação, overfitting e generalização

Para garantir que o modelo **generaliza** bem e não está *overfitting* (ajuste excessivo ao conjunto de treino), foram adotadas boas práticas de validação:

- **Conjunto de teste independente:** Nunca usamos o mesmo conjunto de teste durante o treinamento ou ajuste de hiperparâmetros. O conjunto de teste, totalmente separado dos dados de treinamento.
- **Validação cruzada (k-fold):** No processo de desenvolvimento (seleção de modelo e hiperparâmetros) foi empregada validação cruzada repetida (*k-fold*). Essa técnica divide os dados de treino em várias partições, treinando e avaliando em ciclos diferentes. [k-fold](#)
- **Análise das métricas de treino vs teste:** Após o treinamento final, observamos os valores de RMSE/MAE e  $R^2$  em treino e teste. Se houvesse grande discrepância (por exemplo,  $R^2 \approx 1$  no treino e muito menor no teste), isso indicaria *overfitting*. Nos nossos resultados, os erros de treino e teste foram comparáveis, sugerindo boa generalização. A validação cruzada já havia sinalizado consistência: pontuações médias estáveis sem grandes variâncias entre folds.

Em geral, um modelo que generaliza corretamente deve mostrar desempenho similar em treino e teste, o que foi observado aqui. Se necessário, técnicas adicionais como regulação (limitando `max_depth` ou aumentando `min_samples_leaf`) ou mais dados de treino podem ser usadas para mitigar o overfitting.

## Conclusão

Em suma, a análise comparativa mostrou que o **ExtraTreesRegressor multi-saída** alcançou as melhores métricas de erro (RMSE, MAE) e  $R^2$  para ambos Y1 e Y2, superando tanto a regressão linear quanto outros modelos de ensemble testados. Essa superioridade é atribuída à capacidade dos ensembles de árvores em modelar relações complexas e à maior aleatoriedade do ExtraTrees, que reduz variância. [scikit-learn.org](https://scikit-learn.org).

### Fontes:

[scikit-learn.org](https://scikit-learn.org)[scikit-learn.org](https://scikit-learn.org),

[UCI Machine Learning Repository](#)

<https://doi.org/10.24432/C51307>

[Random Forest Regression in Python | GeeksforGeeks](#)

<https://www.geeksforgeeks.org/random-forest-regression-in-python/>

[MultiOutputRegressor — scikit-learn 1.6.1 documentation](#)

<https://scikit-learn.org/stable/modules/generated/sklearn.multioutput.MultiOutputRegressor.html>

[ExtraTreesRegressor — scikit-learn 1.6.1 documentation](#)

<https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.ExtraTreesRegressor.html>

[GMD - Root-mean-square error \(RMSE\) or mean absolute error \(MAE\): when to use them or not](#)

<https://gmd.copernicus.org/articles/15/5481/2022/>

[Regression Metrics | GeeksforGeeks](#)

<https://www.geeksforgeeks.org/regression-metrics/>

[1.11. Ensembles: Gradient boosting, random forests, bagging, voting, stacking — scikit-learn 1.6.1 documentation](#)

<https://scikit-learn.org/stable/modules/ensemble.html>

[Multioutput Regression in Machine Learning | GeeksforGeeks](#)

<https://www.geeksforgeeks.org/multioutput-regression-in-machine-learning/>

[AI inference vs. training: Key differences and tradeoffs | TechTarget](#)

<https://www.techtarget.com/searchenterpriseai/tip/AI-inference-vs-training-Key-differences-and-tradeoffs>

[Hosting Machine Learning Models as an API Service | by Anshit Vishwakarma | Medium](#)

<https://medium.com/@anshitvishwa111/hosting-machine-learning-models-as-an-api-service-1cf5cb5a1e2f>

[Best Practices for Deploying Machine Learning Models in Production | by GN | Medium](#)

<https://medium.com/@nemagan/best-practices-for-deploying-machine-learning-models-in-production-10b690503e6d>

[Training, validation, and test data sets - Wikipedia](#)

[https://en.wikipedia.org/wiki/Training,\\_validation,\\_and\\_test\\_data\\_sets](https://en.wikipedia.org/wiki/Training,_validation,_and_test_data_sets)

[How K-Fold Prevents overfitting in a model? | GeeksforGeeks](#)

<https://www.geeksforgeeks.org/how-k-fold-prevents-overfitting-in-a-model/>

[How K-Fold Prevents overfitting in a model? | GeeksforGeeks](#)

<https://www.geeksforgeeks.org/how-k-fold-prevents-overfitting-in-a-model/>