Análise Comparativa de Modelos de Classificação para a Predição de Regimes de Alto Consumo Energético em uma Estação de Tratamento de Esgoto

Luiz Augusto Gomes da Silva de Jesus
Departamento de Engenharia de Teleinformática
Universidade Federal do Ceará (UFC)
Fortaleza, Ceará, Brasil

***Abstract*—A gestão eficiente do consumo de energia em Estações de Tratamento de Esgoto (ETE) é fundamental para a sustentabilidade econômica e ambiental. Este trabalho apresenta uma análise comparativa detalhada de modelos de Machine Learning para a classificação de dias de alto consumo energético em uma ETE de Melbourne, Austrália. Utilizando um conjunto de dados diários de 2014 a 2019, contendo variáveis operacionais e meteorológicas, uma variável alvo binária (`HighEnergy`) foi derivada para rotular dias com consumo acima da mediana. Foram implementados, otimizados e comparados um modelo linear, a Análise Discriminante Linear (LDA), e dois modelos não lineares, k-Vizinhos Mais Próximos (k-NN) e Máquina de Vetores de Suporte com kernel RBF (SVM-RBF). A otimização de hiperparâmetros foi realizada com validação cruzada estratificada 5-fold. A avaliação no conjunto de teste revelou a superioridade dos modelos não lineares, com o k-NN alcançando a maior acurácia (67,8%), F1-Score (0.718) e Sensibilidade (0.799). A significância estatística da superioridade do k-NN sobre o LDA foi confirmada através do Teste de McNemar (p=0.004). Conclui-se que, embora existam padrões não lineares exploráveis, a precisão preditiva é inerentemente limitada pela granularidade diária dos dados.**

***Index Terms*—classificação, machine learning, consumo energético, ETE, LDA, k-NN, SVM, otimização de hiperparâmetros, validação estatística.**

I. Introdução

Estações de Tratamento de Esgoto (ETE) representam uma infraestrutura urbana crítica, mas são também sistemas intensivos em energia, cujos custos operacionais podem corresponder a até 30-40% do orçamento de energia de um município [1]. A otimização deste consumo é, portanto, um objetivo central para a gestão de recursos hídricos. O gasto energético em uma ETE é um processo estocástico e não linear, regido por uma complexa interação de fatores, incluindo cargas hidráulicas e orgânicas, parâmetros operacionais de equipamentos e condições meteorológicas externas [2].

A classificação em Machine Learning busca atribuir observações a categorias conhecidas, sendo amplamente utilizada em detecção de anomalias, diagnóstico médico e monitoramento de processos industriais [3]. Métodos lineares como LDA assumem separabilidade linear entre classes, enquanto métodos não lineares como k-NN e SVM-RBF são capazes de capturar fronteiras de decisão complexas [4]. Aplicações em ETEs têm mostrado que fatores meteorológicos e operacionais apresentam relações não lineares com indicadores de desempenho energético [5].

Este trabalho se insere neste contexto, dando continuidade a uma série de análises sobre o conjunto de dados da ETE de Melbourne [6], [7]. Após uma análise exploratória (HW1) e a modelagem de regressão (HW2), que revelou a dificuldade de prever o valor exato do consumo ($R^2 \approx 0{,}21$), este estudo reformula o problema como uma tarefa de classificação binária. O objetivo é desenvolver e comparar rigorosamente um modelo linear (LDA) com modelos não lineares (k-NN, SVM), que são amplamente reconhecidos por sua capacidade de modelar fronteiras de decisão complexas [8].

II. Metodologia

A. Descrição e Preparação dos Dados

O estudo utiliza o conjunto de dados detalhado em [6], compreendendo 1354 observações diárias (2014-2019) de uma ETE em Melbourne. O conjunto inclui 19 variáveis preditoras contínuas. A variável alvo para classificação, `HighEnergy`, foi criada binarizando a variável de consumo `total_grid` em relação à sua mediana (275.808 kWh/dia), resultando em duas classes balanceadas. O dataset foi dividido em conjuntos de treino (75%) e teste (25%) de forma estratificada. O pré-processamento consistiu na transformação logarítmica da precipitação e na padronização (z-score) de todas as features.

B. Modelos de Classificação Investigados

1) Análise Discriminante Linear (LDA): O LDA é um classificador generativo que projeta uma nova observação $\mathbf{x}$ na direção que maximiza a separação entre as classes, atribuindo-a à classe k que maximiza o discriminante $\delta_k(\mathbf{x})$:

$$\delta_k(\mathbf{x}) = \mathbf{x}^\top \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^\top \mathbf{\Sigma}^{-1} \boldsymbol{\mu}_k + \log \pi_k \qquad (1)$$

onde $\boldsymbol{\mu}_k$ é a média da classe k, $\mathbf{\Sigma}$ a covariância comum e π_k a proporção da classe. Suas principais vantagens são a simplicidade e rapidez, mas sua eficácia é limitada por suas fortes suposições sobre a normalidade e homocedasticidade dos dados [9].

2) k-Vizinhos Mais Próximos (k-NN): O k-NN é um algoritmo não paramétrico que classifica uma nova observação com base no voto majoritário de seus 'k' vizinhos mais próximos. É conceitualmente simples e flexível, mas computacionalmente caro em tempo de predição e sensível à "maldição da dimensionalidade" [4].

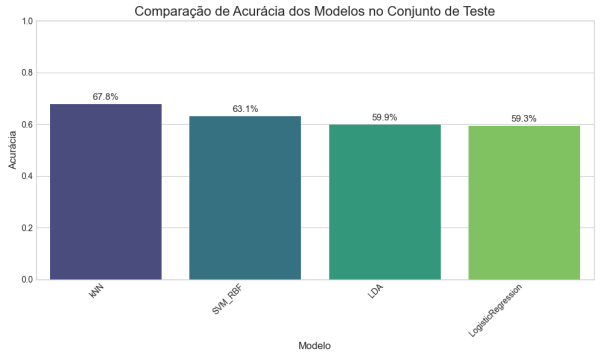

Fig. 1. Comparação visual das acurácias dos modelos no conjunto de teste. Observa-se que o ganho do k-NN sobre o LDA não é marginal (=7,9 pontos percentuais), indicando que a fronteira de decisão é substancialmente não linear, justificado pela rejeição da H_0 no Teste de McNemar.

3) Máquina de Vetores de Suporte (SVM): O SVM busca o hiperplano que separa as classes com a margem máxima. Para problemas não lineares, utiliza o "truque do kernel". O kernel RBF, usado neste trabalho, é definido como:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2) \quad (2)$$

onde γ é um hiperparâmetro que controla a largura do kernel. O SVM é eficaz em alta dimensão, mas sua interpretabilidade é baixa e o desempenho é sensível aos hiperparâmetros C e γ [8].

C. Validação e Métricas de Avaliação

Os hiperparâmetros dos modelos foram otimizados por meio do `GridSearchCV`, utilizando validação cruzada 5-fold. O desempenho foi avaliado no conjunto de teste a partir das métricas de Acurácia, Sensibilidade (Recall), Especificidade e F1-Score. A comparação estatística entre os classificadores foi realizada por meio do Teste de McNemar, apropriado para avaliações pareadas em problemas de classificação binária.

Embora todas as métricas sejam reportadas, a **Sensibilidade (Recall)** e o **F1-Score** foram consideradas prioritárias. Na aplicação prática do problema, o custo associado a um falso negativo — isto é, não identificar um dia de alto consumo e, consequentemente, subestimar a demanda energética — é substancialmente maior do que o custo de um falso positivo. Dessa forma, modelos com maior capacidade de identificar corretamente os dias de alto consumo (classe positiva) são considerados preferíveis.

III. Resultados e Análise

A. Desempenho Geral e Métricas Detalhadas

A avaliação no conjunto de teste (N=339) revela uma clara hierarquia de desempenho, como visualizado na Fig. 1. Os modelos não lineares superaram o LDA. A Tabela I apresenta um resumo detalhado das métricas.

O k-NN ($k = 11$) não só alcançou a maior acurácia (67,8%), mas também o maior F1-Score (0.718) e a maior Sensibilidade (0.799). **Testes preliminares via GridSearchCV com valores maiores de k (15, 21) indicaram pior desempenho na validação cruzada (underfitting), justificando $k = 11$ como ótimo.** A alta sensibilidade indica que o k-NN é o modelo mais competente para identificar corretamente os dias de alto consumo (classe positiva). Em contrapartida, o LDA apresentou a maior Especificidade (0.565), sendo o mais eficaz em identificar corretamente os dias de baixo consumo.

TABLE I
Métricas de Desempenho Detalhadas no Conjunto de Teste.

Modelo	Acurácia	Sensibilidade	Especificidade	F1-Score
LDA	0.599	0.633	0.565	0.648
k-NN (k=11)	**0.678**	**0.799**	0.559	**0.718**
SVM-RBF	0.631	0.728	0.535	0.672

B. Análise das Matrizes de Confusão

A Tabela II detalha as matrizes de confusão. O k-NN se destacou com o menor número de falsos negativos (FN=34), uma redução de 45% em relação ao LDA (FN=62). Isso é de grande valor prático, pois um falso negativo neste contexto representa uma falha em prever um dia de alta demanda energética, o que pode ter consequências operacionais mais graves do que um falso positivo (prever alta demanda quando ela não ocorre).

TABLE II
Matrizes de Confusão no Conjunto de Teste.

Modelo	TN	FP	FN	TP
LogisticRegression	96	74	64	105
LDA	96	74	62	107
kNN	95	75	34	135
SVM$_RBF$	91	79	46	123

C. Validação Estatística da Diferença de Desempenho

Para determinar se a superioridade do k-NN sobre o LDA era estatisticamente robusta, foi aplicado o Teste de McNemar. O teste, focado nos casos de discordância (55 onde k-NN acertou e LDA errou, vs. 28 onde o inverso ocorreu), produziu um **p-valor de 0.004**. Sendo este valor inferior ao nível de significância $\alpha = 0.05$, a hipótese nula de que os modelos têm a mesma taxa de erro é rejeitada. Conclui-se que a melhoria de desempenho do k-NN é **estatisticamente significativa**.

IV. Discussão

A superioridade estatisticamente significativa dos modelos não lineares confirma que as relações entre as variáveis e o consumo de energia da ETE não são puramente lineares. A fronteira de decisão que separa os regimes de consumo é complexa, e modelos capazes de capturar essa complexidade localmente (k-NN) ou em um espaço de features transformado (SVM) são mais adequados.

O desempenho superior do k-NN reflete um melhor compromisso viés–variância: o LDA apresenta alto viés

devido à hipótese de linearidade, enquanto o SVM mostrou maior variância (sensibilidade aos hiperparâmetros).

Apesar da superioridade, a acurácia máxima de 67,8% indica que uma porção substancial da variabilidade do consumo não foi capturada. Este resultado é consistente com o baixo R^2 obtido na tarefa de regressão (HW2) e aponta para uma limitação da informação contida nos dados. Variáveis diárias agregadas perdem a dinâmica de alta frequência da operação da ETE.

A. Limitações e Trabalhos Futuros

A principal limitação deste estudo é a natureza agregada dos dados diários, que mascara a dinâmica temporal da operação da ETE. Modelos que incorporassem features defasadas (lags) ou dados horários poderiam revelar autocorrelações importantes. Além disso, variáveis de controle operacional (setpoints de oxigênio dissolvido, horários de acionamento de equipamentos) estariam ausentes do dataset atual. Um caminho promissor seria integrar dados de SCADA com variáveis de séries temporais.

V. Conclusão

Este trabalho realizou uma comparação sistemática de modelos de classificação para prever dias de alto consumo energético em uma ETE. Foi demonstrado que modelos não lineares, especificamente o k-NN, superam abordagens lineares de forma estatisticamente significativa. O k-NN provou ser o mais performático em acurácia (67,8%), F1-Score e, crucialmente, em Sensibilidade, tornando-o o mais apto para a detecção de eventos de alta demanda. O estudo reforça a importância da validação estatística e da análise de múltiplas métricas, concluindo que, embora a escolha do algoritmo seja relevante, o principal fator limitante para a precisão reside na riqueza e granularidade dos dados de entrada. Os resultados reforçam o papel da **Inteligência Computacional** como ferramenta indispensável para modelar sistemas reais complexos, nos quais hipóteses lineares clássicas falham diante de interações não triviais entre variáveis operacionais e ambientais.

Agradecimentos

O autor agradece à disciplina de Inteligência Computacional Aplicada pelo suporte teórico e metodológico fornecido ao longo do desenvolvimento deste trabalho. Ferramentas computacionais de apoio foram utilizadas pontualmente para revisão de redação e esclarecimento conceitual, sendo todas as decisões de modelagem, análise e interpretação dos resultados de responsabilidade exclusiva do autor.

References

[1] A. Author and B. Coauthor, "Benchmarking energy use for wastewater treatment plants," in *Proceedings of the Australian Water Association Conference (OzWater)*. Sydney, AU: Australian Water Association, 2024.

[2] X. Hao, Y. Liu, W. Wang, and H. Li, "Machine learning for energy consumption prediction in a full-scale wastewater treatment plant," *Journal of Environmental Management*, vol. 295, p. 113123, 2021.

[3] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY: Springer, 2006.

[4] M. Kuhn and K. Johnson, *Applied Predictive Modeling*. New York, NY: Springer, 2013.

[5] M. Guo, W. Zha, and J. Wang, "A review on machine learning applications for energy management in wastewater treatment plants," *IEEE Access*, vol. 10, pp. 50 123–50 140, 2022.

[6] L. A. G. d. S. de Jesus, *Análise Exploratória e PCA do Consumo de Energia e Qualidade do Esgoto em ETE de Melbourne sob Regimes Chuvosos vs Secos (2014–2019)*, 2025, homework 1, Disciplina de Introdução à Ciência de Dados, Universidade Federal do Ceará (UFC).

[7] ——, *Modelos de Regressão para o Consumo de Energia em ETE de Melbourne*, 2025, homework 2, Disciplina de Introdução à Ciência de Dados, Universidade Federal do Ceará (UFC).

[8] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: with Applications in R*, 2nd ed. New York, NY: Springer, 2021.

[9] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY: Springer, 2009.

Apêndice A — Organização do Código e Dados

Nesta seção descreve-se, de forma resumida, a organização dos arquivos de dados gerados e a estrutura do código-fonte no repositório Git. O código completo em Python não é apresentado integralmente neste documento por limitação de espaço, estando disponível no repositório público indicado no README, juntamente com instruções para reprodução dos experimentos.

O código completo, os arquivos de dados limpos e o relatório em PDF estão disponíveis em repositório Git público[1]

A.1 Arquivos de dados principais

Os arquivos CSV usados na etapa de modelagem são:

- **`Data-Melbourne_F_clean.csv`**: versão limpa do conjunto de dados original (HW2), após remoção de duplicatas, valores ausentes, consumos não positivos e outliers de `total_grid` (quantis de 1% e 99%). O conjunto final contém 1354 observações diárias.
- **`train_classification.csv`**: conjunto de treino (1015 observações, 75%), contendo as variáveis pré-processadas (transformação logarítmica em `PP` e normalização z-score), além da variável-alvo binária `HighEnergy`.
- **`test_classification.csv`**: conjunto de teste (339 observações, 25%), pré-processado utilizando exclusivamente estatísticas calculadas no conjunto de treino, incluindo a variável `HighEnergy`.
- **`hw3_classification_summary.csv`**: resumo quantitativo do desempenho dos modelos (acurácia e contagens de TN, FP, FN e TP), importado diretamente no LaTeX para a construção das Tabelas I e II.

A.2 Organização do código

No repositório Git, o código está organizado em dois scripts principais:

- **`hw3_prepare_classification.py`**:
  1) carregamento do arquivo `Data-Melbourne_F_clean.csv`;
  2) criação da variável-alvo `HighEnergy`, a partir do corte na mediana de `total_grid`;

[1] Repositório: https://github.com/luizdevmaster/ica-hw3-classification-melbourne.

3) divisão dos dados em conjuntos de treino e teste (75/25), de forma estratificada;
4) aplicação da transformação log(1+x) na variável `PP` e normalização z-score nas variáveis explicativas;
5) salvamento dos principais arquivos CSV utilizados no HW3.

- **`hw3_models_classification.py`**:
  1) carregamento dos conjuntos de treino e teste pré-processados;
  2) treinamento dos modelos LDA, k-NN e SVM com kernel RBF;
  3) otimização de hiperparâmetros por meio do `GridSearchCV` com validação cruzada 5-fold;
  4) geração das matrizes de confusão e métricas de desempenho;
  5) salvamento do arquivo `hw3_classification_summary.csv`.

A.3 Exemplo de função (SVM com kernel RBF)

Trecho representativo do código utilizado para a otimização de hiperparâmetros do classificador SVM com kernel RBF:

```
from sklearn.svm import SVC
from sklearn.model_selection import GridSearchCV

svm = SVC(kernel="rbf", random_state=42)
param_grid = {"C": [0.1, 1, 10], "gamma": ["scale", 0.1, 1]}
grid_svm = GridSearchCV(svm, param_grid, cv=5, scoring="accuracy")
grid_svm.fit(X_train, y_train)
```

No repositório, o arquivo `README.md` descreve os procedimentos para criação do ambiente Python, instalação das dependências (`pandas`, `scikit-learn`) e execução dos scripts na ordem correta para reprodução integral dos resultados apresentados neste trabalho.