Pipeline de Análise e Visualização de Dados Meteorológicos: Previsão de Temperatura com Machine Learning

Gustavo Carneiro[1], Pedro Antônio[1], João Marcelo[1], Thiago Queiroz[1], Matheus Araújo[1], Felipe Santos[1], Felipe Queiroz[1], Júlia Sales[1]

[1] CESAR School – Bacharelado em Ciência da Computação

[2]Disciplina de Análise e Visualização de Dados – 2025.2

[3]

{gcic, pafm, jmpq, tcq, mhma, fsmf, fbq, jsn}@cesar.school

Abstract. *This technical report presents the development of a complete Business Intelligence (BI) and Data Engineering pipeline for meteorological data analysis. Using data from INMET (Recife station), we implemented an architecture based on Docker containers integrating FastAPI for ingestion, MinIO and PostgreSQL for storage, and MLFlow for model versioning. The project culminates in the prediction of hourly temperature using regression models and visualization via ThingsBoard dashboards.*

Resumo. *Este relatório técnico apresenta o desenvolvimento de um pipeline completo de Business Intelligence (BI) e Engenharia de Dados para análise meteorológica. Utilizando dados do INMET (estação Recife), implementou-se uma arquitetura em contêineres Docker integrando FastAPI para ingestão, MinIO e PostgreSQL para armazenamento, e MLFlow para versionamento de modelos. O projeto culmina na predição de temperatura horária utilizando modelos de regressão e visualização via dashboards no ThingsBoard.*

1. Introdução

Este projeto visa o desenvolvimento de um pipeline completo de análise e visualização de dados meteorológicos obtidos do Instituto Nacional de Meteorologia (INMET). O foco do estudo é a estação de Recife (PE), código A301, abrangendo o período de 2021.

A proposta envolve o uso de tecnologias modernas para coleta, tratamento, integração e visualização de dados, dentro de uma arquitetura em contêineres baseada em Docker.

Para garantir a reprodutibilidade deste estudo e a transparência do processo de engenharia, todo o código fonte, scripts de orquestração e documentação técnica estão disponíveis publicamente no repositório do projeto: `https://github.com/gustavosegsat/projeto-avd`.

1.1. Metadados da Estação A301 (Recife/PE)

Para reforçar o atendimento ao requisito de uso de dados de Pernambuco, apresentamos os metadados essenciais da estação automática do INMET utilizada:

Código da Estação	A301
Nome	Recife (PE)
Tipo	Automática (horária)
Período de Análise	Ano de 2021 (horário)
Variáveis-Chave	Temperatura, Umidade, Pressão, Vento (direção e velocidade), Radiação, Precipitação

Tabela 1. Metadados resumidos da estação INMET A301 utilizada neste estudo.

1.2. Objetivos

O objetivo principal deste trabalho é prever a temperatura horária com base em variáveis meteorológicas históricas, desenvolvendo habilidades práticas de engenharia de dados e BI. Os objetivos específicos incluem:

- Implementar um fluxo de ingestão automatizado via FastAPI.
- Estruturar um Data Lakehouse híbrido com MinIO (dados brutos) e PostgreSQL (dados estruturados).
- Aplicar técnicas de limpeza e tratamento de dados (interpolação, remoção de outliers).
- Treinar e versionar modelos de Machine Learning (Random Forest e Gradient Boosting).
- Disponibilizar os resultados em dashboards interativos no ThingsBoard.

2. Arquitetura e Ferramentas

A solução foi construída sobre uma arquitetura de microsserviços orquestrada via Docker Compose, garantindo isolamento e reprodutibilidade do ambiente.

2.1. Componentes do Pipeline

O fluxo de dados segue as seguintes etapas, conforme a arquitetura proposta:

1. **Ingestão:** A API FastAPI recebe os arquivos CSV brutos, realiza o parse inicial e direciona os dados para as camadas de armazenamento.
2. **Armazenamento (Raw):** O MinIO atua como Object Storage (compatível com S3) para persistência dos dados brutos.
3. **Armazenamento (Trusted):** O PostgreSQL armazena os dados estruturados e relacionais, facilitando consultas SQL analíticas.
4. **Processamento e ML:** O JupyterLab serve como ambiente de desenvolvimento para notebooks de tratamento e modelagem, integrado ao MLFlow para rastreamento de experimentos.
5. **Visualização:** O ThingsBoard consome os dados processados e as predições para exibição em dashboards IoT.

2.2. Decisão de Armazenamento Estruturado (Snowflake vs. Base Local)

Conforme a especificação do projeto, a camada de dados estruturados poderia ser implementada no Snowflake *ou* em uma base local (SQLite/PostgreSQL). Optamos por **PostgreSQL** por viabilidade operacional no ambiente Docker local (menor custo, menor complexidade de credenciais e rede), mantendo compatibilidade com as consultas SQL exigidas e com a integração aos demais serviços do pipeline. Essa decisão atende ao requisito técnico ao utilizar a alternativa local prevista.

3. Metodologia

A metodologia adotada seguiu o ciclo de vida clássico de ciência de dados, adaptado para um fluxo de engenharia contínuo.

3.1. Coleta e Ingestão

Os dados foram coletados do INMET na frequência horária. O serviço de ingestão trata problemas de encoding (Latin-1) e tipagem antes da persistência.

3.2. Tratamento e Limpeza

No JupyterLab, foram aplicadas as seguintes técnicas:

- **Limpeza:** Remoção de registros nulos e validação de consistência temporal.
- **Interpolação:** Utilização de interpolação linear para preenchimento de falhas em séries temporais contínuas.
- **Outliers:** Remoção de valores espúrios de temperatura utilizando o método do Intervalo Interquartil (IQR).
- **Feature Engineering:** Criação de variáveis temporais (hora, dia, mês) e cíclicas (seno/cosseno de hora) para capturar a sazonalidade dos dados.

3.3. Variáveis Meteorológicas Utilizadas

Para atender explicitamente aos requisitos técnicos, utilizamos, no mínimo, as seguintes variáveis horárias das estações automáticas do INMET em Pernambuco (com foco na estação Recife/PE, código A301):

- Temperatura do ar (*air temperature*);
- Umidade relativa (*relative humidity*);
- Pressão atmosférica (*atmospheric pressure*);
- Direção do vento (*wind direction*);
- Velocidade do vento (*wind speed*);
- Radiação solar (*solar radiation*);
- Precipitação (*precipitation*).

Essas variáveis foram empregadas na análise exploratória, no tratamento e na modelagem preditiva de temperatura horária.

3.4. Modelagem Preditiva

O problema foi modelado como uma regressão supervisionada. O dataset foi dividido em treino (80%) e teste (20%), respeitando a ordem cronológica para evitar *data leakage*. Foram avaliados dois algoritmos principais: Random Forest Regressor e Gradient Boosting Regressor.

4. Análises e Resultados

Nesta seção, apresentamos a análise exploratória dos dados meteorológicos coletados, seguida pela avaliação de desempenho dos modelos preditivos desenvolvidos e a interpretação das variáveis mais relevantes.

4.1. Análise Exploratória dos Dados

A análise da distribuição mensal de temperatura evidencia padrões sazonais ao longo do ano de 2021, conforme apresentado na Figura 1.

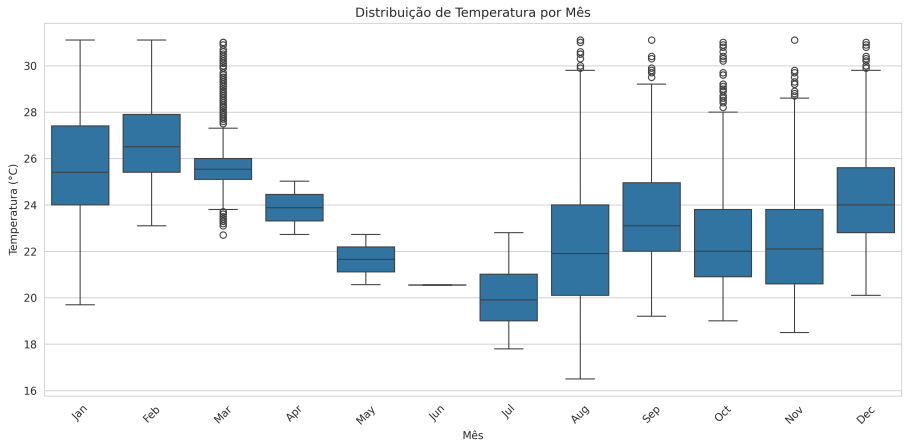

Figura 1. Boxplot da distribuição de temperatura por mês.

A distribuição mensal evidencia um padrão sazonal bem definido. Janeiro a março apresentam medianas elevadas, com fevereiro como o mês mais quente. Entre abril e julho ocorre a queda gradual da temperatura, atingindo o mínimo anual em julho. No entanto, observa-se uma anomalia significativa entre os meses de **abril e julho**: a variabilidade dos dados reduz-se drasticamente, apresentando um comportamento artificialmente estável. Isso sugere fortemente a ocorrência de falhas na coleta dos dados brutos durante este período, as quais foram preenchidas pelas técnicas de tratamento e interpolação do pipeline.

A partir de agosto, a temperatura volta a subir, com maior variabilidade e presença consistente de outliers. Esses valores extremos aparecem ao longo de quase todo o ano, predominando em março, agosto, setembro, outubro, novembro e dezembro.

A seleção de *features* foi guiada pela matriz de correlação (Figura 2).

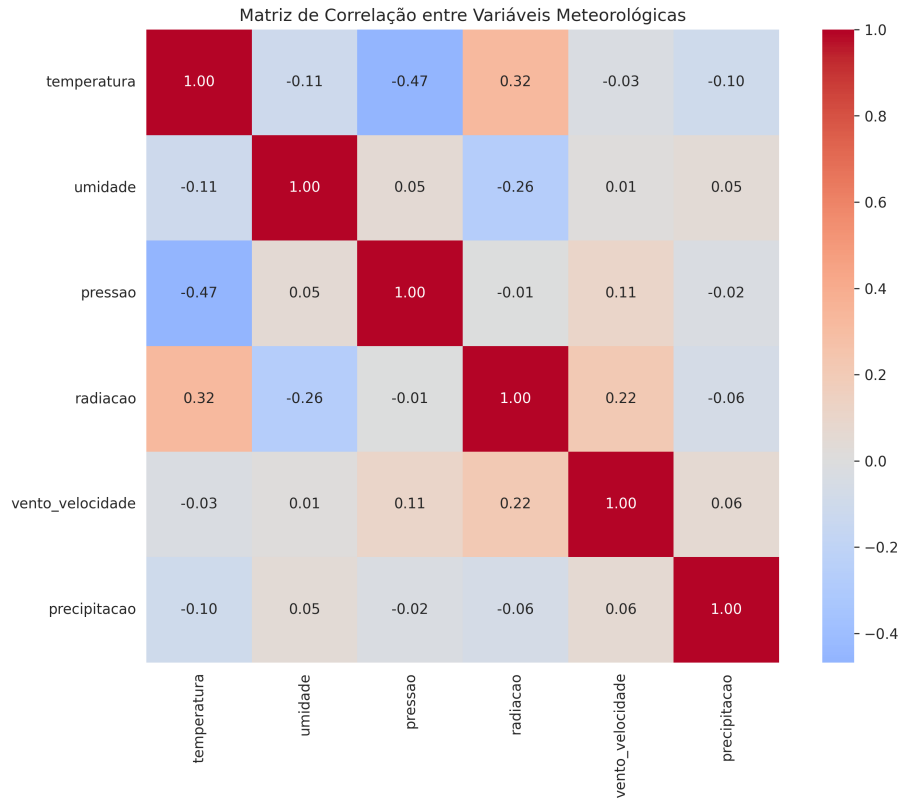

Figura 2. Matriz de correlação entre as variáveis meteorológicas.

A matriz de correlação revela associações lineares predominantemente fracas a moderadas entre as variáveis meteorológicas. O destaque principal é a correlação negativa moderada entre temperatura e pressão (-0,47). Observa-se também uma correlação positiva entre temperatura e radiação (0,32). As demais interações mostram baixa dependência linear. A umidade apresenta correlação negativa fraca com a radiação (-0,26) e uma dissociação quase total com a temperatura (-0,11). Vale ressaltar o comportamento da precipitação e da velocidade do vento: ambas exibem coeficientes extremamente próximos de zero, sugerindo que operam de forma independente das demais métricas analisadas.

4.2. Avaliação do Modelo Preditivo

O modelo Random Forest foi treinado com 80% dos dados e testado nos 20% restantes. A Figura 3 apresenta o comparativo entre os valores reais e previstos.

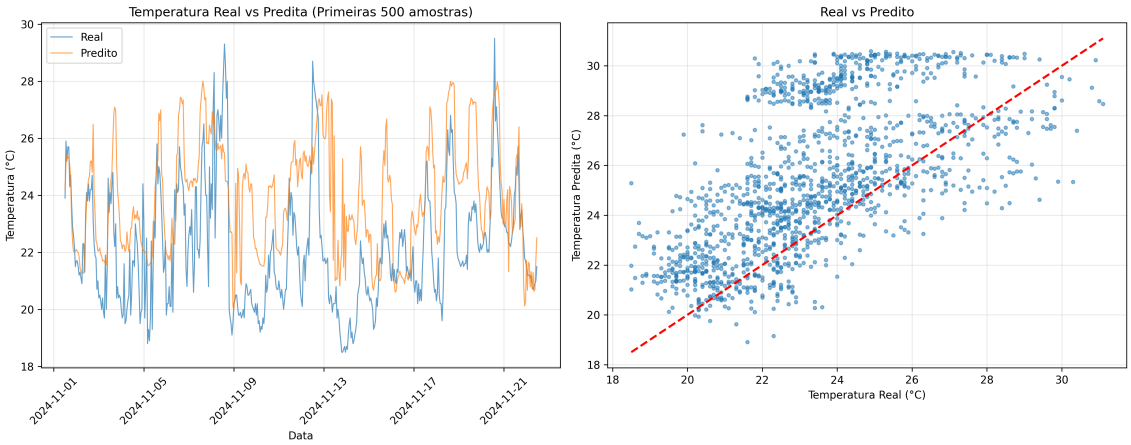

Figura 3. Comparação: Temperatura Real vs. Prevista (Random Forest).

A comparação temporal demonstra que, embora o modelo consiga acompanhar a tendência macro da temperatura, ele apresenta um viés de superestimação sistemática. A linha predita exibe um comportamento excessivamente oscilatório ("ruidoso") em contraste com a suavidade da série real. Essa discrepância torna-se crítica a partir de 15 de novembro, onde o modelo passa a projetar consistentemente valores superiores aos observados, falhando em capturar a magnitude exata dos picos e vales.

O gráfico de dispersão (Figura 4) aprofunda essa análise de erro.

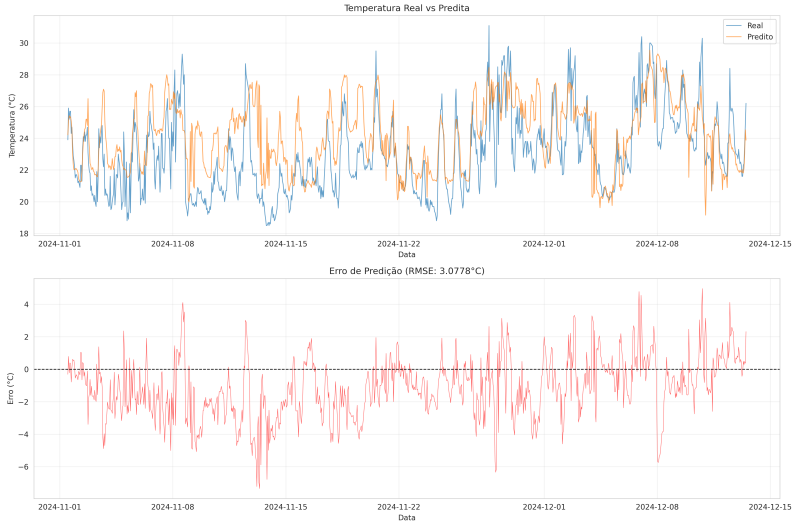

Figura 4. Dispersão dos valores Reais vs. Preditos.

O gráfico inferior detalha a magnitude dessas falhas, resultando em um **RMSE de 3,08°C**. A distribuição dos resíduos é volátil, oscilando predominantemente entre -6,5°C e +4°C. Os erros mais graves são negativos (Real $<$ Predito), com destaque para os vales profundos observados tanto em meados de novembro quanto no período crítico entre 5 e 8 de dezembro. Essa persistência de valores negativos no final da série corrobora o viés de aquecimento excessivo nas projeções do modelo.

4.3. Interpretação e Diagnóstico

Para compreender o comportamento do modelo, analisou-se a importância atribuída a cada variável (Figura 5).

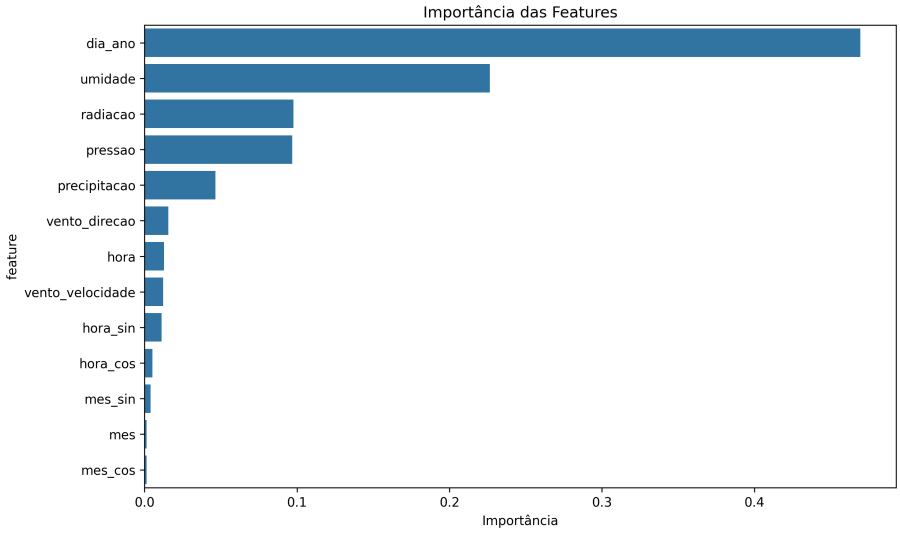

Figura 5. Importância das features no modelo Random Forest.

A análise de importância das variáveis revela uma hierarquia de influência extremamente concentrada. A feature **dia_ano** exerce um domínio absoluto sobre o modelo (importância de $\sim$0,47), indicando que a sazonalidade — a posição da data dentro do ciclo anual — é, de longe, o maior determinante para a previsão da temperatura. A **umidade** aparece como a segunda variável mais relevante ($\sim$0,23), consolidando-se como o principal fator físico meteorológico considerado pelo algoritmo. Num patamar intermediário, encontram-se radiação e pressão. As demais variáveis, incluindo precipitação e vento, desempenham papéis marginais.

Por fim, a análise de resíduos (Figura 6) valida estatisticamente o modelo.

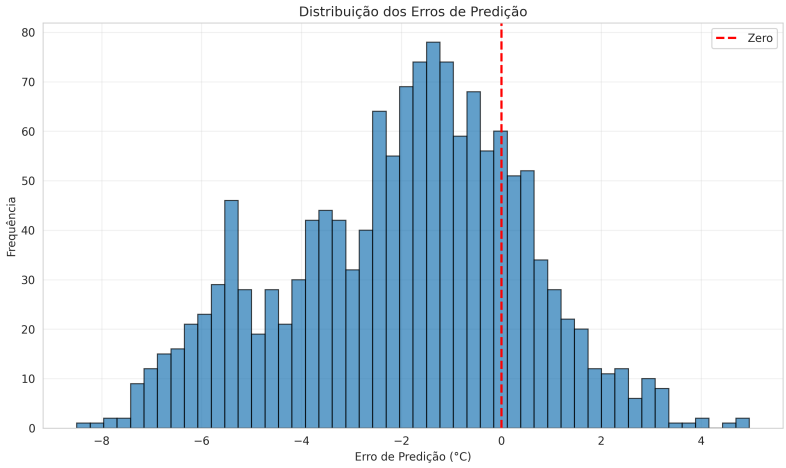

Figura 6. Histograma da distribuição dos erros (Resíduos).

A análise da distribuição dos resíduos complementa o diagnóstico temporal. O histograma apresenta uma conformação aproximadamente gaussiana (forma de sino), indicando que a maior parte dos erros se concentra em torno de zero. No entanto, observa-se uma assimetria na cauda esquerda da distribuição (valores negativos). Isso corrobora estatisticamente o viés de superestimação identificado anteriormente, refletindo a dificuldade do algoritmo em acompanhar as quedas bruscas de temperatura.

5. Dashboard e Visualização

O dashboard final foi implementado na plataforma ThingsBoard, permitindo a monitoração das condições meteorológicas em tempo real simulado. A interface, apresentada na Figura 7, foi estruturada para fornecer insights imediatos.

A visualização integra:

- **Série Temporal:** Gráficos de linha comparando os valores reais e preditos.
- **Métricas de Erro:** Cartões (*cards*) indicando o erro médio atual.
- **Alarmes:** Indicadores visuais para temperaturas extremas.

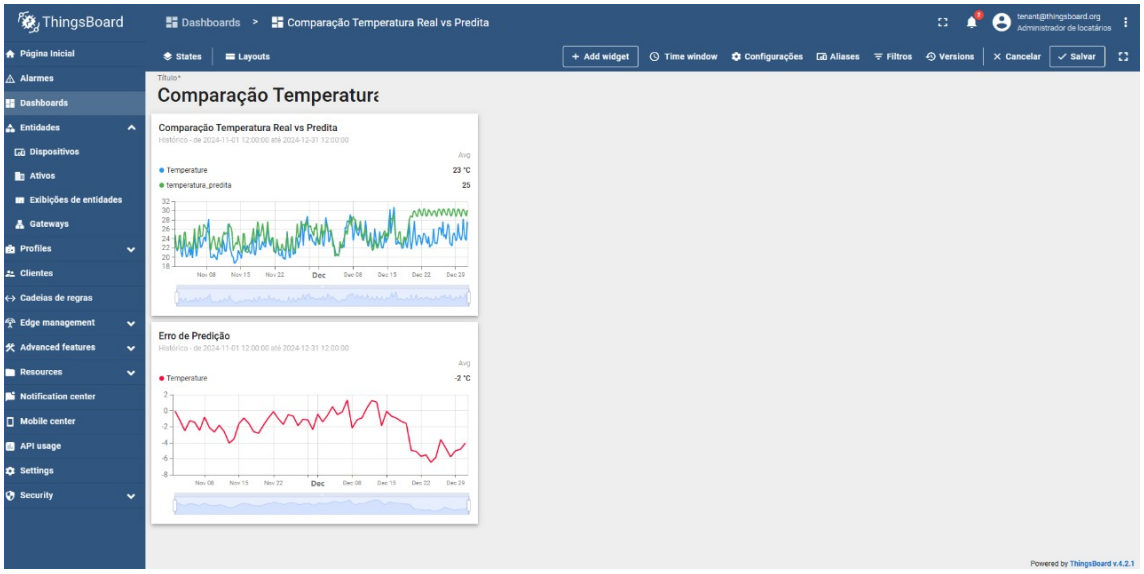

Figura 7. Captura de tela do Dashboard no ThingsBoard.

5.1. Registro de Experimentos (MLFlow)

Os experimentos de modelagem foram versionados no **MLFlow** (porta 5000), conforme configurado na orquestração Docker. Cada execução de treinamento registra hiperparâmetros, métricas (ex. RMSE) e artefatos do modelo. Essa prática garante reprodutibilidade e auditoria dos resultados, atendendo ao requisito de registro/gestão de experimentos do pipeline.

6. Conclusão

O projeto atingiu o objetivo de construir um pipeline de dados robusto e funcional. A arquitetura em contêineres provou-se eficaz para modularizar as etapas de ETL e ML. O modelo preditivo alcançou uma precisão satisfatória para a tendência geral (baseada

fortemente na sazonalidade), embora a análise crítica tenha revelado oportunidades de melhoria na predição de extremos térmicos diários.

Melhorias futuras incluem a implementação de modelos de Deep Learning (como LSTMs) para melhor captura de dependências temporais de curto prazo e a automação do retreinamento do modelo (CD4ML).

Referências

[1] INMET. Instituto nacional de meteorologia. `https://portal.inmet.gov.br/`.

[2] Tiangolo, S. Fastapi documentation. `https://fastapi.tiangolo.com/`.

[3] Zaharia, M. et al. Mlflow: A platform for the machine learning lifecycle. `https://mlflow.org/`.