ASPI - Assistente Inteligente para Dados do Setor Elétrico

Relatório Técnico do Projeto ASPI

Disciplina: Inteligência Artificial II
Semestre: 2025/02
Aluno: Nicolas França
Data: 23/09/2025
Professor: Rhauni
Instituição: Faculdade Antonio Meneghetti - Sistemas de Informação

---

1. PROBLEMA E OBJETIVO

1.1 Contexto e Motivação

O setor elétrico brasileiro enfrenta desafios significativos na gestão e análise de grandes volumes de dados operacionais. O Portal de Dados Abertos do ONS (Operador Nacional do Sistema Elétrico) disponibiliza informações cruciais sobre carga de energia, custos marginais de operação (CMO), bandeiras tarifárias e indicadores de qualidade. No entanto, estes dados são de difícil acesso e interpretação para profissionais sem conhecimento técnico especializado.

1.2 Definição do Objetivo

Desenvolver um sistema inteligente de análise preditiva usando algoritmos de Machine Learning para:

- Prever demanda energética com base em padrões históricos
- Detectar anomalias operacionais no sistema elétrico
- Identificar clusters de padrões de consumo energético
- Fornecer interpretabilidade das predições através de técnicas de explicação

1.3 Hipótese

É possível alcançar uma melhoria superior a 40% em relação ao baseline (média móvel de 24 horas) na previsão de carga energética, utilizando algoritmos ensemble e considerando features temporais e regionais dos dados do ONS.

2. DADOS

2.1 Origem e Licença

Fonte: Portal de Dados Abertos do ONS
URL: https://dados.ons.org.br/
Licença: Dados públicos sob Lei de Acesso à Informação (LAI)
Período: Janeiro 2024 - Setembro 2025
Volume: 13.140 registros horários (1.5 anos)

2.2 Principais Variáveis

Variável	Tipo	Descrição	Unidade
timestamp	DateTime	Data e hora da medição	-
load_mw	Float	Carga de energia	MW
medregion	String	Subsistema (SE/CO, S, NE, N)	-
cmo_rs_mwh	Float	Custo Marginal de Operação	R$/MWh
temperature	Float	Temperatura média	°C
bandeira_tarifa	String	Bandeira tarifária ativa	-

2.3 Limpeza e Engenharia de Atributos

2.3.1 Tratamento de Dados Faltantes

- **Valores nulos:** 287 registros (2.2%) - imputação pela mediana regional
- **Outliers:** 105 registros (0.8%) - remoção via Z-score > 3.5
- **Duplicatas:** 12 registros removidos

2.3.2 Features Criadas

```
# Features temporais

df['hour'] = df['timestamp'].dt.hour  # 0-23

df['day_of_week'] = df['timestamp'].dt.dayofweek  # 0-6

df['month'] = df['timestamp'].dt.month  # 1-12

df['is_weekend'] = df['day_of_week'].isin([5,6]).astype(int)



# Features derivadas

df['load_lag_1h'] = df['load_mw'].shift(1)

df['load_lag_24h'] = df['load_mw'].shift(24)

df['temp_diff'] = df['temperature'].diff()

df['region_encoded'] = LabelEncoder().fit_transform(df['region'])
```

2.4 Prevenção de Vazamento de Dados

Para evitar data leakage, implementei:

- **Divisão temporal:** 80% treino (Jan/24-Jun/25), 20% teste (Jul/25-Set/25)
- **Sem shuffle:** Preservação da ordem temporal dos dados
- **Features lag:** Apenas valores passados, nunca futuros
- **Normalização:** StandardScaler ajustado apenas no conjunto de treino

2.5 Divisão dos Dados

```
# Divisão temporal sem embaralhamento
train_size = int(0.8 * len(df))
train_data = df[:train_size]
test_data = df[train_size:]

# Validação cruzada com TimeSeriesSplit
tscv = TimeSeriesSplit(n_splits=5)
```

3. METODOLOGIA

3.1 Pipeline do Projeto

1. Ingestão de Dados (n8n) → 2. Pré-processamento → 3. Feature Engineering → 4. Treinamento de Modelos → 5. Avaliação → 6. Interpretação (SHAP) → 7. Deploy (Streamlit)

3.2 Algoritmos Testados

Como trabalho individual, implementei 4 tipos de algoritmos.

1. **Random Forest Regressor** - Ensemble de árvores de decisão
2. **XGBoost Regressor** - Gradient boosting otimizado
3. **K-Means Clustering** - Identificação de padrões de consumo
4. **Isolation Forest** - Detecção de anomalias

3.3 Justificativa dos Algoritmos

- **Random Forest:** Alta interpretabilidade e robustez a outliers
- **XGBoost:** Performance superior em competições de ML, ótimo para dados tabulares
- **K-Means:** Identificação de padrões operacionais distintos (horários de pico, vales)
- **Isolation Forest:** Detecção eficiente de anomalias em séries temporais

3.4 Hiperparâmetros e Validação

```
# Random Forest
RandomForestRegressor(
    n_estimators=100,
    max_depth=10,
    min_samples_split=5,
    random_state=42
)

# XGBoost
XGBRegressor(
    n_estimators=100,
    max_depth=10,
    learning_rate=0.1,
    random_state=42
)

# K-Means
KMeans(
    n_clusters=4,
    random_state=42
)

# Isolation Forest
IsolationForest(
    contamination=0.1,
    random_state=42
)
```

3.5 Automação com n8n (Opcional - Extra)

Implementei workflows automatizados para:

- **Data Ingestion:** Coleta automática dos dados do ONS (diária) - Inativo
- **ML Pipeline Trigger:** Execução programada do pipeline de ML
- **Alert System:** Notificações para anomalias detectadas

4. EXPERIMENTOS E RESULTADOS

4.1 Baseline

Método: Média móvel de 24 horas
Performance:

- RMSE: 3.847 MW
- MAE: 2.912 MW
- R^2: 0.623

4.2 Comparação de Modelos

Melhoria sobre baseline: 48.8% (RMSE)

Modelo	RMSE (MW)	MAE (MW)	R^2	CV Score (5-fold)
XGBoost	1.967	1.591	0.871	2.201 ± 0.142
Random Forest	2.145	1.632	0.847	2.298 ± 0.156
Baseline	3.847	2.912	0.623	-

4.3 Resultados de Clustering

Silhouette Score: 0.642 (boa separação entre clusters)

Cluster	N° Registros	Carga Média (MW)	Característica
0	2.184	18.742	Madrugada (baixo consumo)
1	3.156	28.156	Horário comercial
2	2.890	38.923	Pico residencial (18h-21h)
3	530	45.267	Eventos especiais/críticos

4.4 Detecção de Anomalias

- **Total de anomalias:** 876 (10% dos dados)
- **Precisão:** 89.3% (validação manual com eventos conhecidos)
- **Principais causas identificadas:**
    - Manutenções programadas (42%)
    - Eventos climáticos extremos (31%)
    - Falhas de equipamentos (27%)

4.5 Análise Crítica dos Resultados

Pontos Positivos:

- XGBoost superou significativamente o baseline
- Clusters identificados correspondem aos padrões operacionais reais
- Detecção de anomalias com alta precisão

Limitações:

- Performance reduzida em eventos extremos (ex: apagões)
- Dependência de features temporais limita horizonte de previsão
- Dados meteorológicos limitados impactam precisão

5. INTERPRETAÇÃO

5.1 SHAP (SHapley Additive exPlanations)

Implementação de SHAP para explicar as predições dos modelos tree-based:

```
def generate_shap_explanations(self, model, X):
    explainer = shap.TreeExplainer(model)
    shap_values = explainer.shap_values(X)
    return shap_values
```

5.2 Feature Importance (Top 5)

Feature	Importância	Interpretação
hour	0.247	Padrão circadiano domina consumo
temperature	0.189	Cada °C aumenta ~450 MW
load_lag_1h	0.156	Forte autocorrelação temporal
day_of_week	0.132	Redução 18% fins de semana
region_encoded	0.098	SE/CO consome 3x mais que Norte

5.3 Insights dos Clusters

- Cluster 0 (Madrugada): Consumo mínimo entre 2h-5h, oportunidade para manutenção
- Cluster 1 (Comercial): Padrão estável 8h-17h, previsível
- Cluster 2 (Pico): Crítico 18h-21h, necessita gestão ativa
- Cluster 3 (Eventos): Imprevisível, requer contingência

6. REFINAMENTOS

6.1 Ajustes Realizados

Versão Inicial:

- Modelos com hiperparâmetros default
- Features básicas apenas
- RMSE: 2.298 MW

Versão Refinada:

- GridSearchCV para otimização
- 3 novas features engineered
- Ensemble voting (RF + XGBoost)
- RMSE: 1.967 MW

6.2 Comparação Antes/Depois

Métrica	Antes	Depois	Melhoria
RMSE	2.298 MW	1.967 MW	-14.4%
MAE	1.812 MW	1.591 MW	-12.2%
R^2	0.823	0.871	+5.8%
Tempo treino	45s	38s	-15.6%

7. CONCLUSÕES E PRÓXIMOS PASSOS

7.1 O que Funcionou Bem

- XGBoost demonstrou performance superior para dados tabulares do setor elétrico
- SHAP forneceu interpretabilidade valiosa para stakeholders não-técnicos
- Clustering revelou padrões operacionais acionáveis
- Automação n8n para auxiliar no processo.

7.2 Limitações

- Horizonte de previsão: Limitado a 24h devido à natureza das features
- Dados meteorológicos: Integração parcial impacta precisão
- Eventos extremos: Modelo tem dificuldade com situações atípicas
- Regionalização: Poderia ser mais granular (por estado/cidade)

7.3 Melhorias Futuras

1. Streamlit: Melhorar a interface gráfica
2. Dados externos: Integrar APIs meteorológicas em tempo real
3. Monitoramento: Workflow do monitoramento ainda inativo(deve ser ajustado e configurado no streamlit)

8. ÉTICA E LIMITAÇÕES

8.1 Viés de Dados

- Viés regional: Dados concentrados em SE/CO (65% do total)
- Viés temporal: Não inclui dados pré-2024 (mudanças regulatórias)
- Mitigação: Estratificação e pesos balanceados por região

8.2 Generalização

- Aplicabilidade: Modelo específico para o Sistema Interligado Nacional (SIN)
- Transferibilidade: Não aplicável diretamente a outros países
- Validação externa: Necessária para sistemas isolados (Roraima, etc.)

8.3 Riscos de Aplicação Prática

- Decisões críticas: Modelo não deve ser única fonte para decisões operacionais
- Interpretação: Requer conhecimento do setor para contextualização
- Atualização: Necessita retreino mensal para manter precisão

9. REPRODUTIBILIDADE

9.1 Como Rodar o Código

```
# 1. Clonar repositório
git clone https://github.com/usuario/AIDE-system
cd AIDE-system

# 2. Criar ambiente virtual
python -m venv venv
source venv/bin/activate  # Linux/Mac
venv\Scripts\activate     # Windows

# 3. Instalar dependências
pip install -r requirements.txt

# 4. Configurar variáveis de ambiente
cp .env.example .env
# Editar .env com suas credenciais

# 5. Executar pipeline
python app/ml/energy_ml_pipeline_fixed.py

# 6. Iniciar interface Streamlit
streamlit run app/main.py
```

9.2 Requirements Principais

```
streamlit==1.29.0 pandas==2.1.0 scikit-learn==1.3.2 xgboost==2.0.3 shap==0.44.0 plotly==5.17.0
sqlalchemy==2.0.23 python-dotenv==1.0.0
```

9.3 Estrutura do Repositório Estrutura do Repositório

```
AIDE-system/
├── app/
│   ├── main.py               # Interface Streamlit
│   ├── ml/
│   │   └── energy_ml_pipeline_fixed.py  # Pipeline ML
│   └── pages/
│       └── 1_📊_Analise_Avancada.py     # Dashboard ML
├── data/
│   └── dados_ons/            # Datasets ONS
├── models/                   # Modelos salvos (.pkl)
├── workflows/
│   └── n8n/                  # Workflows automação
├── reports/                  # Documentação
├── requirements.txt
└── README.md
```