UNIVERSIDADE FEDERAL DE LAVRAS
CAMPUS SEDE

Alexandre Guimarães Vartuli

Modelagem Preditiva da Demanda de Energia Elétrica: Uma Abordagem com Machine Learning e o Framework CRISP-ML(Q)

LAVRAS
2023

ALEXANDRE GUIMARÃES VARTULI

Modelagem Preditiva da Demanda de Energia Elétrica : Uma Abordagem com Machine Learning e o Framework CRISP-ML(Q)

Trabalho de Conclusão de Curso submetido à Universidade Federal de Lavras, como requisito necessário para obtenção do grau de Bacharel em Engenharia de Controle e Automação

Lavras, dezembro de 2023

Agradecimentos

Em primeiro lugar, agradeço à minha avó, Maria Luiza, que não poupou esforços para me apoiar em todos os sentidos durante meu percurso acadêmico e na vida. Sem ela, nenhum trabalho dessa magnitude seria possível. Meus eternos agradecimentos a Dada, um presente de Deus em minha vida.

À Tia Nice que tanto me motivou com suas palavras cativantes. Obrigado, Titia.

Às minhas queridas irmãs, Giulia e Priscilla, quero expressar minha profunda gratidão pelo carinho e apoio constantes. Sem a presença delas, a trajetória teria sido ainda mais árdua

Agradeço ao Dodô, cujas experiências de vida contribuíram significativamente para o meu crescimento pessoal

Ao meu amigo Gabriel Sena, que sempre esteve presente para me ouvir e apoiar. Sua amizade, sem dúvida, foi valiosa ao longo da minha jornada acadêmica.

Agradeço imensamente à minha namorada, Rafaelle, pelo apoio incondicional. Seu zelo e carinho foram fontes de renovo constantes.

Ao meu amigo Luiz Fernando, sou grato pelas extensas conversas que tivemos durante a graduação, as quais contribuíram para a elaboração dos meus interesses profissionais.

Aos meus colegas de classe da turma de 2017/2, agradeço por tornarem toda a jornada acadêmica mais agradável. Obrigado pelo apoio ao longo deste longo percurso

Gostaria de expressar minha sincera gratidão ao Professor Paulo Guimarães, que prontamente aceitou meu pedido para ser meu orientador. Seu profissionalismo e vasto conhecimento foram de grande auxílio para a realização deste trabalho

Quero reconhecer todas as pessoas que, de diferentes formas, colaboraram para o desenvolvimento deste trabalho. Mesmo que não tenham sido mencionadas individualmente, sua contribuição foi valorosa e fundamental. Obrigado a todos por fazerem parte deste processo.

Resumo

Este Trabalho de Conclusão de Curso (TCC) investiga a previsão da demanda de energia elétrica, um aspecto crucial para a gestão e otimização de redes elétricas. Diante da crescente demanda e da necessidade de previsibilidade no setor, este estudo foca na aplicação de modelos avançados de Machine Learning para predizer a demanda de energia oferecendo contribuições para a tomada de decisão no setor de energia. Especificamente, utilizou=se o Extreme Gradient Boosting (XGBoost) e o Prophet, para modelar e prever essas demandas. A metodologia adotada inclui uma análise exploratória detalhada, seguida de uma preparação dos dados, e finalmente a implementação dos modelos mencionados dentro do framework CRISP-ML(Q), um padrão emergente em análises preditivas.

Os resultados demonstram que , em uma iteração no framework, o modelo XGBoost obteve um melhor desempenho.

Palavras-chave: Machine Learning, CRISP-ML(Q), Previsão de Demanda de Energia, Análise de Séries Temporais, Modelagem Preditiva, PROPHET, XGBoost

Abstract

This Final Year Project investigates the forecasting of electric power demand, a crucial aspect for the management and optimization of electrical networks. Given the growing demand and the need for predictability in the sector, this study focuses on the application of advanced Machine Learning models to predict power demand, offering contributions to decision-making in the energy sector. Specifically, Extreme Gradient Boosting (XGBoost) and Prophet were used to model and predict these demands. The adopted methodology includes a detailed exploratory analysis, followed by data preparation, and finally the implementation of the mentioned models within the CRISP-ML(Q) framework, an emerging standard in predictive analyses.

The results demonstrate that, in one iteration within the framework, the XGBoost model performed bette

Keywords: Machine Learning, CRISP-ML(Q), Energy Demand Forecasting, Time Series Analysis, Predictive Modeling, PROPHET, XGBoost.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ML - Machine Learning

XGBoost - eXtreme Gradient Boosting

CRISP-ML(Q) - Cross Industry Standard Process for Machine Learning with Quality assurance

ARIMA - Auto Regressive Integrated Moving Average

Sumário

1 Introdução

O crescimento econômico de cada nação está altamente relacionado à sua infraestrutura, rede e disponibilidade de eletricidade. A eletricidade, tornando-se um componente essencial da vida moderna, impulsionou um aumento notável na demanda global, especialmente para fins residenciais e comerciais.

Paralelamente, os preços da eletricidade têm apresentado flutuações significativas ao longo dos anos. Há também uma preocupação crescente com a inadequação na geração de eletricidade para satisfazer a demanda global. Esses desafios têm impulsionado estudos que visam estimar a demanda futura de energia elétrica. Este planejamento é crucial para que geradores, distribuidores e fornecedores possam se preparar adequadamente e promover a conservação de energia entre os consumidores.

Um dos principais desafios enfrentados pela indústria de energia é a previsão da demanda elétrica, um problema persistente desde a invenção da energia elétrica, conforme indicado por Nti et al. [Nti et al. 2020]. Uma eficiente gestão de rede, essencial para a redução do custo de produção e aumento da capacidade de geração, exige um planejamento minucioso da demanda. Uma previsão precisa da demanda de energia é, portanto, fundamental para maximizar a eficiência do processo de planejamento nas indústrias de geração de energia.

Para melhorar a precisão dessas previsões de consumo e demanda, diversas técnicas computacionais e estatísticas vêm sendo empregadas. Este cenário é amplificado com o crescente interesse e aplicação de técnicas de Machine Learning (ML) na otimização dos modelos de previsão.

O Machine Learning é uma área de pesquisa da Inteligência Artificial que visa ao desenvolvimento de programas de computador com a capacidade de aprender a executar uma dada tarefa com sua própria experiência [Faceli et al. 2011]. Com o uso de técnicas de ML é possível desenhar programas capazes de aprender por si só, utilizando-se um conjunto de dados, rotulados ou não, como insumo para o algoritmo reconhecer algum padrão alvo. "O aprendizado de máquina provê aos computadores a habilidade de aprender sem que algo seja programado explicitamente" [Samuel 2000].

Dentro do ML, existem tarefas descritivas e tarefas preditivas [Cerri e Carvalho 2019]. Em tarefas descritivas, busca-se o desenvolvimento de algoritmos que descreverão os dados. As tarefas preditivas podem ser divididas em tarefas de classificação e tarefas de regressão.

Em tarefas de classificação, busca-se atribuir categorias predefinidas a dados, como por exemplo, quando se deseja saber se, dada uma imagem de um animal, o animal é um gato ou não. Nas tarefas de regressão, objetiva-se prever o valor de uma variável numérica (atributo de saída), dadas outras variáveis (atributos de entrada). Um exemplo significativo de uma tarefa de regressão, que será amplamente abordado neste trabalho, é a modelagem preditiva da demanda de energia elétrica.

Resolver problemas da realidade com Machine Learning não é uma tarefa trivial. Idealmente trabalha-se com um grande volume de dados, sendo necessário percorrer diversas etapas que vão desde o tratamento de dados até treinar um modelo. Como forma de mitigar essa dificuldade, faz-se necessário estabelecer processos e diretrizes a serem seguidas para garantir a qualidade do modelo [Studer et al. 2021]. Neste contexto, o framework CRISP-ML(Q) surge como uma proposta valiosa.

O CRISP-ML(Q), que expande o modelo CRISP-DM tradicional, é especificamente projetado para aplicações de Machine Learning, incorporando uma abordagem de Garantia de Qualidade (Quality Assurance - QA) em cada fase do processo de desenvolvimento. Este modelo enfatiza a importância de uma monitorização e manutenção contínuas para lidar com a degradação potencial do modelo em ambientes em mudança, garantindo assim a eficácia e confiabilidade do modelo ao longo do tempo. Ao seguir este framework, é possível endereçar de forma mais eficiente os desafios intrínsecos ao desenvolvimento de soluções de Machine Learning, desde a compreensão dos dados e do negócio até a implantação e manutenção do modelo em produção. [Studer et al. 2021]

1.1 Motivação

Realizar um projeto de Machine Learning , seguindo um framework estruturado , é algo pouco difundido no Brasil, sobretudo quando aplicado à energia. Aplicar, difundir, provar que é algo palpável o uso de tecnologias computacionais e estatísticas dentro dessa temática é fundamental para a adoção de soluções potenciais.

A demanda de energia elétrica é uma área de crescente interesse e desafio. A habilidade de prever a demanda de energia auxilia na gestão e otimização das redes elétricas, trazendo previsibilidade às concessionárias de energia para estruturem suas operações. Esta aplicação específica do ML em previsão da demanda de energia ilustra a relevância e o potencial das tarefas de regressão em fornecer insights valiosos para questões críticas da vida real. [Fu et al. 2022]

1.2 Objetivo

1.2.1 Objetivo Geral

Exemplificar o uso e aplicar modelos de ML à prática da previsão da demanda de energia elétrica, utilizando o framework CRISP-ML(Q).

1.2.2 Objetivos Específicos

Destacam-se os seguintes objetivos específicos:

- Prever a demanda de Energia Elétrica com o auxílio das Séries Temporais e algoritmos de Machine Learning;
- Fazer uso de algumas fases importantes do *Framework* CRISP-ML(Q);
- Fazer um comparativo dos modelos PROPHET e XGBoost para o caso específico desse trabalho;

1.3 Organização

O presente trabalho está organizado da seguinte forma:

Capítulo 2: os conceitos relacionados as Séries Temporais, CRISP-ML(Q), aos modelos XGBoost e Prophet

Capítulo 3: Trata-se a respeito da Metodologia utilizada.

Capítulo 4: Discussões e Resultados.

Capítulo 6: Conclusão.

2 Revisão Bibliográfica

2.1 Inteligência Artificial e Machine Learning

Defini-se Inteligência Artificial (IA) como qualquer dispositivo que percebe seu ambiente, aprenda e tome decisões que maximizam sua chance de sucesso em algum objetivo, após se especializarem, podendo ser através de dados, modelo mais comum, ou não [Abbas, Nasser e Ahmad 2015].

Sendo uma porção da IA, o aprendizado de máquina, é um conjunto de métodos matemáticos e estatísticos associados a programação de computadores, a fim de solucionar problemas através de classificação, regressão e segmentação. Esse aprendizado é parametrizável, além disso, pode ser feito através de dados, treinamento supervisionado, ou de experiências passadas, chamado de treinamento por reforço [Mohri, Rostamizadeh e Talwalkar 2018].

Segundo [Dutt, Chandramouli e Das 2023], existem diferentes tipos de aprendizado quando se trata de ML, são eles:

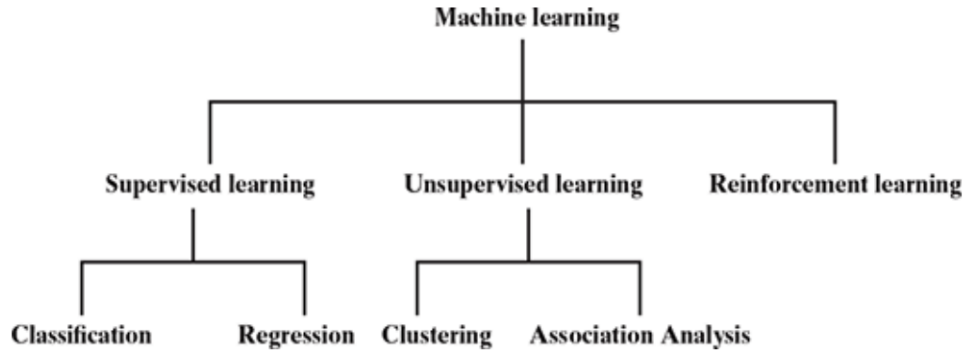

Figura 1 – Tipos de Machine Learning, disponível em [Dutt, Chandramouli e Das 2023]

- Aprendizado Supervisionado - Também chamado de aprendizado preditivo. Uma máquina prevê a classe de objetos desconhecidos com base em informações prévias relacionadas à classe de objetos semelhantes. Sendo dividido em problemas de Classificação e Regressão
- Aprendizado Não Supervisionado - Também chamado de aprendizado descritivo. Uma máquina encontra padrões em objetos desconhecidos agrupando objetos semelhantes.
- Aprendizado por Reforço - Uma máquina aprende a agir por conta própria para alcançar os objetivos dados.

Para o propósito desse trabalho, focaremos no Aprendizado Supervisonado - Supervised Learning mais especificamente em problemas de Regressão.

2.1.1 Aprendizado Supervisionado, Regressão e Séries Temporais

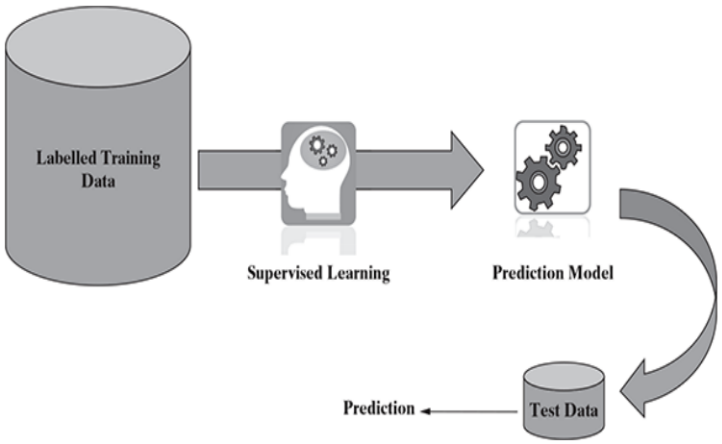

Figura 2 – Aprendizado Supervisionado

A motivação principal do aprendizado supervisionado é aprender com informações passadas. Para conseguir "aprender"é necessário que ela tenha uma informação básica sobre os dados, os chamados rótulos e é fornecido à ela por meio do Conjunto de Treinamento ou Training Data. Dentro do universo de Aprendizado Supervisionado existem dois grandes grupos de problemas:

- Classificação - Quando estamos tentando prever uma variável categórica ou nominal;
- Regressão - Quando estamos tentando prever uma variável de valor real, como por exemplo, a Demanda de Energia Elétrica.

Em um cenário de previsão da demanda por energia elétrica, um modelo de regressão pode ser treinado com dados históricos que incluem temperaturas, horários do dia, e o consumo de energia. O modelo aprende a relação entre essas variáveis e o consumo de energia, permitindo a previsão da demanda futura.

Dentro do domínio da regressão, existem regressões aplicadas a séries temporais que representa um campo de estudo particularmente intrigante. Séries temporais são definidas como conjuntos de dados coletados de maneira sequencial em intervalos de tempo regulares. Exemplos clássicos desses dados incluem registros diários de temperaturas ou as cotações semanais do mercado de ações. Nestes contextos, a habilidade de prever valores futuros baseando-se em registros históricos é crucial, especialmente em áreas como análise econômica, previsões meteorológicas e planejamento urbano.

A técnica de regressão em séries temporais é empregada para modelar e antecipar comportamentos de variáveis ao longo do tempo. Regressão é uma técnica estatística para estimar a relação entre uma variável dependente e uma ou mais variáveis independentes.

A forma matemática mais geral de uma equação de regressão é a seguinte:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_k x_k + \varepsilon \tag{2.1}$$

Onde:

- y é a variável dependente.
- $x_1, x_2, \ldots, x_k$ são as variáveis independentes.
- $\beta_0, \beta_1, \ldots, \beta_k$ são os coeficientes de regressão.
- ε é o termo de erro.

Existem muitos tipos diferentes de regressão, cada um com suas próprias características e aplicações. Alguns dos tipos mais comuns de regressão incluem:

- **Regressão Linear:** Este é o tipo de regressão mais simples. Ela assume que a relação entre a variável dependente e as variáveis independentes é linear.
- **Regressão Não Linear:** Utilizada quando a relação entre a variável dependente e as variáveis independentes não é linear. Este tipo de regressão pode modelar relações mais complexas.
- **Regressão Múltipla:** Empregada quando há mais de uma variável independente. É uma extensão da regressão linear que permite a análise de múltiplas variáveis preditoras.
- **Regressão Logística:** Usada quando a variável dependente é categórica, como por exemplo, para prever a probabilidade de ocorrência de um evento.

Um exemplo aplicado de regressão é na análise da demanda de energia elétrica. Aqui, esses são usados para estimar demandas futuras, levando em consideração fatores como padrões sazonais, tendências históricas e variáveis ambientais.

Várias metodologias são empregadas especificamente para a regressão em séries temporais. A regressão linear temporal, por exemplo, integra variáveis temporais (como tempo do dia ou dia da semana) como preditores no modelo. Os Modelos ARIMA (AutoRegressive Integrated Moving Average), são particularmente eficazes para lidar com séries temporais

não estacionárias, combinando técnicas autoregressivas com médias móveis. Além disso, a regressão com delays temporais (lagged regression) utiliza valores antecedentes na série como variáveis independentes para prognosticar observações futuras.

Ao abordar séries temporais, enfrentamos desafios específicos que diferem significativamente dos encontrados em outros tipos de análise de dados. Em muitos modelos de regressão, há uma suposição fundamental de que os dados são estacionários. A estacionariedade refere-se à propriedade segundo a qual as características estatísticas dos dados, como média e variância, permanecem constantes ao longo do tempo. No entanto, esta condição raramente se aplica a séries temporais, onde é comum observar mudanças ao longo do tempo devido a fatores como tendências e sazonalidade.

Para lidar com a não estacionariedade das séries temporais, uma técnica comumente utilizada é a diferenciação. Diferenciação significa subtrair o valor atual da série pelo seu valor anterior. Este processo ajuda a remover tendências ou padrões sazonais dos dados, tornando a série mais próxima de ser estacionária. Por exemplo, se uma série temporal mostra um aumento consistente ao longo do tempo, a diferenciação reduzirá esta tendência, facilitando a análise e modelagem subsequentes.

Outro aspecto crucial no trabalho com séries temporais é a autocorrelação. Autocorrelação ocorre quando uma observação em uma série temporal é correlacionada com observações anteriores. Em outras palavras, o valor atual em uma série pode ser influenciado por seus valores passados. A presença de autocorrelação pode distorcer os resultados de um modelo de regressão tradicional e deve ser cuidadosamente considerada. Métodos como a função de autocorrelação (ACF) e a função de autocorrelação parcial (PACF) são ferramentas diagnósticas usadas para identificar e quantificar a autocorrelação em séries temporais.

Além disso, a identificação e modelagem de sazonalidade e tendências são essenciais. Sazonalidade refere-se a padrões que se repetem em intervalos regulares, como variações diárias, semanais ou anuais. Por outro lado, tendências são mudanças de longo prazo na média da série. A decomposição da série temporal, que separa os dados em componentes de tendência, sazonalidade e ruído, é uma técnica útil para entender esses elementos. Compreender e modelar adequadamente a sazonalidade e tendências permite a criação de modelos de regressão mais precisos e confiáveis para séries temporais.

2.2 Prophet

O modelo Prophet, desenvolvido pela equipe de Ciência de Dados do Facebook, é uma biblioteca de código aberto projetada para previsão de séries temporais, especialmente útil

para previsões de negócios. Seus recursos incluem a adaptação a mudanças de tendência, incorporação de efeitos de feriados e eventos e robustez a dados ausentes e outliers. Ele utiliza um modelo de regressão aditiva que combina componentes de tendência, sazonalidade e feriados, além de permitir a inclusão de outras variáveis para aprimorar a previsão. O Prophet é aplicável em diversos campos, como previsão de vendas no varejo, análise do mercado de ações, previsão meteorológica e alocação de recursos em organizações

Segundo [Taylor e Letham 2018] , o modelo Prophet utiliza uma função de crescimento logístico para modelar a tendência, permitindo a flexibilidade em prever pontos de mudança na tendência dos dados. Ele aborda a sazonalidade usando séries de Fourier, que capturam padrões sazonais complexos. Para feriados e eventos especiais, o modelo inclui um componente aditivo específico que pode ser ajustado para cada evento.

2.3 Extreme Gradient Boosting - XGBoost

Extreme Gradient Boosting (XGBoost) representa uma inovação significativa no campo do aprendizado de máquina, como evidenciado por [DHALIWAL, NAHID e ABBAS 2018]. Esse oferece uma implementação eficiente do algoritmo de aumento de gradiente (Gradient Boosting), uma técnica avançada em aprendizado de máquina para tarefas de modelagem preditiva tanto em classificação quanto em regressão.

O Gradient Boosting é um método de conjunto que integra dois modelos principais: árvore de decisão e boosting. O modelo de boosting é particularmente notável por sua habilidade de atribuir pesos diferenciados às variáveis de entrada, baseando-se na força de sua correlação com a saída do algoritmo. Isso significa que parâmetros previsores mais influentes recebem um "impulso"maior, tornando-se mais preponderantes nas decisões do modelo. As árvores são adicionadas sequencialmente ao conjunto, cada uma ajustada para corrigir erros cometidos pelos modelos anteriores, um processo que define o conceito de boosting.

Os algoritmos de árvore de decisão, conforme [DHALIWAL, NAHID e ABBAS 2018] detalham, analisam os atributos do conjunto de dados, onde esses atributos, também conhecidos como recursos ou colunas, funcionam como nós condicionais ou internos. Dependendo da condição no nó raiz, a árvore se ramifica em galhos, representando diferentes percursos de decisão.

Além disso, o aumento do gradiente constrói árvores de forma inteligente, permitindo obter pontuações de recursos que indicam a importância de cada um no modelo de treinamento. Assim, quanto mais um recurso é utilizado em decisões críticas dentro das árvores, maior

será sua pontuação de importância.

2.4 Python

Diante da vasta gama de linguagens de programação disponíveis no mundo da tecnologia, Python se destaca como uma escolha primordial, especialmente no campo da Ciência de Dados, conforme apontado por [VANDERPLAS 2016]. Essa linguagem de alto nível e código aberto é reconhecida por sua facilidade de aprendizado e comandos enxutos, características que a tornam acessível tanto para iniciantes quanto para programadores experientes.

A riqueza das bibliotecas disponíveis em Python é um de seus maiores atrativos. Essas bibliotecas abrangem uma ampla gama de funcionalidades, desde manipulação de dados até algoritmos complexos de aprendizado de máquina e inteligência artificial. Estruturas de dados como listas, arrays, tuplas, matrizes, dicionários e dataframes são fundamentais nessa linguagem, pois oferecem maneiras eficientes de lidar com grandes volumes de dados, uma necessidade crítica na Ciência de Dados.

Além disso, a natureza de código aberto do Python tem atraído o apoio de gigantes da tecnologia, como Google e Microsoft. Essas empresas não apenas utilizam Python em diversos de seus produtos e serviços, mas também contribuem ativamente para o seu desenvolvimento, criando e disponibilizando uma variedade de frameworks. Essa colaboração enriquece ainda mais o ecossistema do Python, proporcionando ferramentas mais robustas e avançadas para os desenvolvedores.

A comunidade de desenvolvedores de Python também desempenha um papel crucial no crescimento e na evolução contínua da linguagem. Essa comunidade ativa e engajada contribui constantemente com novas soluções e melhorias, garantindo que Python permaneça na vanguarda da inovação tecnológica e científica. Seja em análise de dados, automação, desenvolvimento web ou inteligência artificial, Python prova ser uma linguagem versátil e indispensável no arsenal de qualquer profissional da tecnologia.

2.5 Visual Studio Code

O Visual Studio Code (VS Code) é um ambiente de desenvolvimento integrado (IDE) leve e eficiente, desenvolvido pela Microsoft. É amplamente reconhecido por sua interface de usuário compacta, mas intuitiva, que facilita a codificação e depuração. Sua característica mais notável é a extensão e personalização, permitindo que os usuários adicionem linguagens,

depuradores e ferramentas conforme necessário. O VS Code suporta múltiplas linguagens de programação, tornando-o ideal para uma variedade de projetos de software.

2.6 CRISP-ML(Q)

A importância de frameworks em projetos de Machine Learning, como discutido no artigo [Studer et al. 2021] , é essencial para a organização e sucesso desses projetos. O artigo enfatiza a necessidade de um modelo de processo específico para aplicações de ML, evidenciando o papel do CRISP-DM e modelos semelhantes. Estes frameworks são fundamentais para assegurar a adaptação dos modelos a ambientes dinâmicos, além de enfatizar a manutenção contínua e a garantia de qualidade em todas as fases do desenvolvimento. Essa abordagem estruturada é vital para que os projetos de ML atendam às expectativas e mantenham sua relevância e funcionalidade ao longo do tempo

No artigo [Studer et al. 2021], as fases do framework para projetos de Machine Learning (ML) são detalhadas da seguinte forma:

- **Entendimento de Negócios e Dados**: Esta fase inicial é crucial para definir os objetivos do negócio e traduzi-los em objetivos de ML, coletar e verificar a qualidade dos dados e avaliar a viabilidade do projeto.
- **Preparação dos Dados**: Envolve a seleção, limpeza, construção e padronização dos dados, preparando-os para o modelo.
- **Modelagem**: Foca na escolha de técnicas de modelagem apropriadas, levando em conta os objetivos de ML e as condições limitantes do projeto.
- **Avaliação**: Avalia o desempenho do modelo, sua robustez, explicabilidade e outros critérios de sucesso definidos anteriormente.
- **Implantação**: Envolve o uso prático do modelo no campo de aplicação designado, abordando questões como hardware de inferência, avaliação do modelo sob condições de produção, aceitação e usabilidade do usuário.
- **Monitoramento e Manutenção**: Essencial para avaliar a eficácia do modelo ao longo do tempo e para realizar ajustes conforme necessário, garantindo que o modelo continue a ser relevante e eficaz.

3 Metodologia

Com o intuito de evidenciar o impacto das tecnologias computacionais e estatísticas na gestão e otimização de redes elétricas, realizou-se a aplicação e comparação de dois modelos de Machine Learning – Prophet e XGBoost – na previsão da demanda de energia elétrica. Para tanto, seguiram-se as principais etapas do framework CRISP-ML(Q)

3.1 Ambiente de Programação/Linguagem escolhida e bibliotecas

Para o desenvolvimento do trabalho, utilizou-se o Visual Studio Code - Vscode , pois traz ótimas ferramentas de visualizações, fornece a instalação de pacotes do Python e um ambiente gráfico agradável.

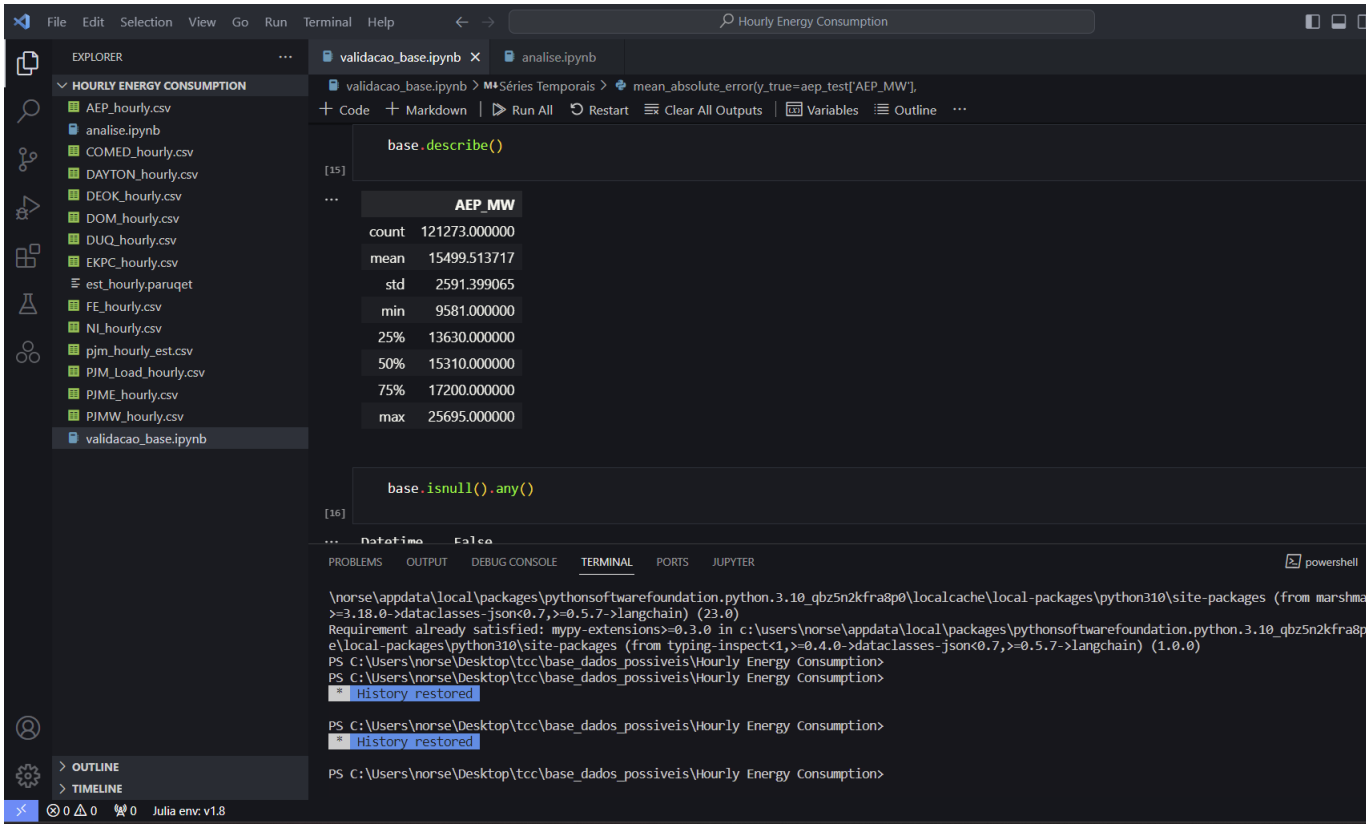

Figura 3 – Ambiente do Visual Studio Code, Fonte: Autor

3.2 Entendimento de Negócios e Dados

3.2.1 Dataset

Os dados que serviram de insumo para esse estudo vieram do Kaggle - uma comunidade global de Ciência de Dados , entitulado de : "Hourly Energy Consumption". O Dataset traz mais de 10 anos de informações verificadas e públicas acerca do consumo horário de energia elétrica em Megawatts de diversas concessionarias de energia elétrica que atuam em diversos estados dos Estados Unidos. Para o estudo, concentramos na American Electric Power - AEP , cuja área de jurisdição se destaca em laranja na figura abaixo. A escolha dela se pautou na alta quantidade de dados , não nulos ou vazios, oque facilita a implementação e eficiência dos modelos.

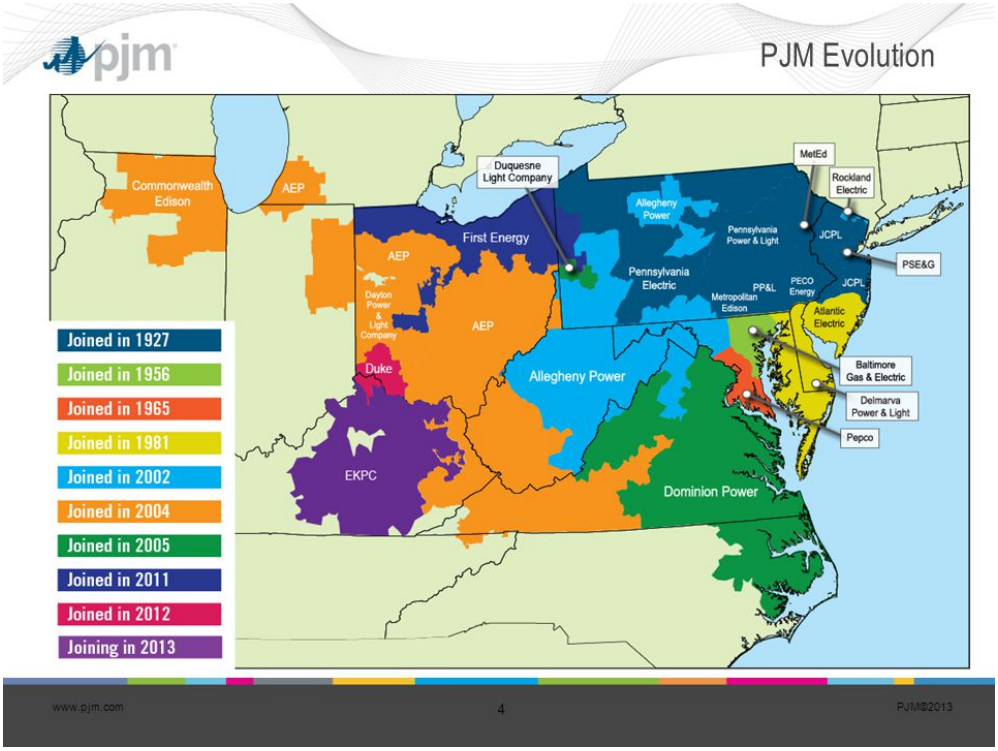

Figura 4 – Áreas de Atuação da AEP, Fonte: PJM Website

O conjunto possui 121.273 linhas com duas colunas: Datetime que representa a informação do dia e hora do consumo e a segunda coluna AEP_MW que mostra o consumo respectivo à determinada hora e data.

```
(121273, 2)
Datetime    datetime64[ns]
AEP_MW             float64
dtype: object
```

	Datetime	AEP_MW
0	2004-12-31 01:00:00	13478.0
1	2004-12-31 02:00:00	12865.0
2	2004-12-31 03:00:00	12577.0
3	2004-12-31 04:00:00	12517.0
4	2004-12-31 05:00:00	12670.0

Figura 5 – Tamanho e Cabeçalho da base AEP

3.2.2 Análise Exploratória

Nessa etapa realizaram-se as devidas análises para entender o comportamento dos dados.

1. Intervalo de Datas

As datas variam de 10/2004 a 08/2018

```
base['Datetime'].describe(datetime_is_numeric=True)
[15] ✓ 0.0s
...
count                           121273
mean     2011-09-02 03:17:01.553025024
min                2004-10-01 01:00:00
25%                2008-03-17 15:00:00
50%                2011-09-02 04:00:00
75%                2015-02-16 17:00:00
max                2018-08-03 00:00:00
Name: Datetime, dtype: object
```

Figura 6 – Intervalo de Data Considerada

2. Estudo da Série Temporal

Ao visualizar os dados de consumo de energia ao longo do tempo, identifica-se um padrão sazonal marcante. Para aprofundar a compreensão dessa sazonalidade, é proveitoso examinar as variações no consumo ao longo dos meses e em diferentes horas do dia. Para tanto, optou-se pela criação de gráficos de caixa, ou box plots, para cada mês e hora, oferecendo uma síntese visual eficaz das tendências e variações observadas na série temporal. Esses gráficos proporcionam uma visão clara e resumida do comportamento do consumo em diferentes períodos, realçando aspectos chave da sazonalidade.

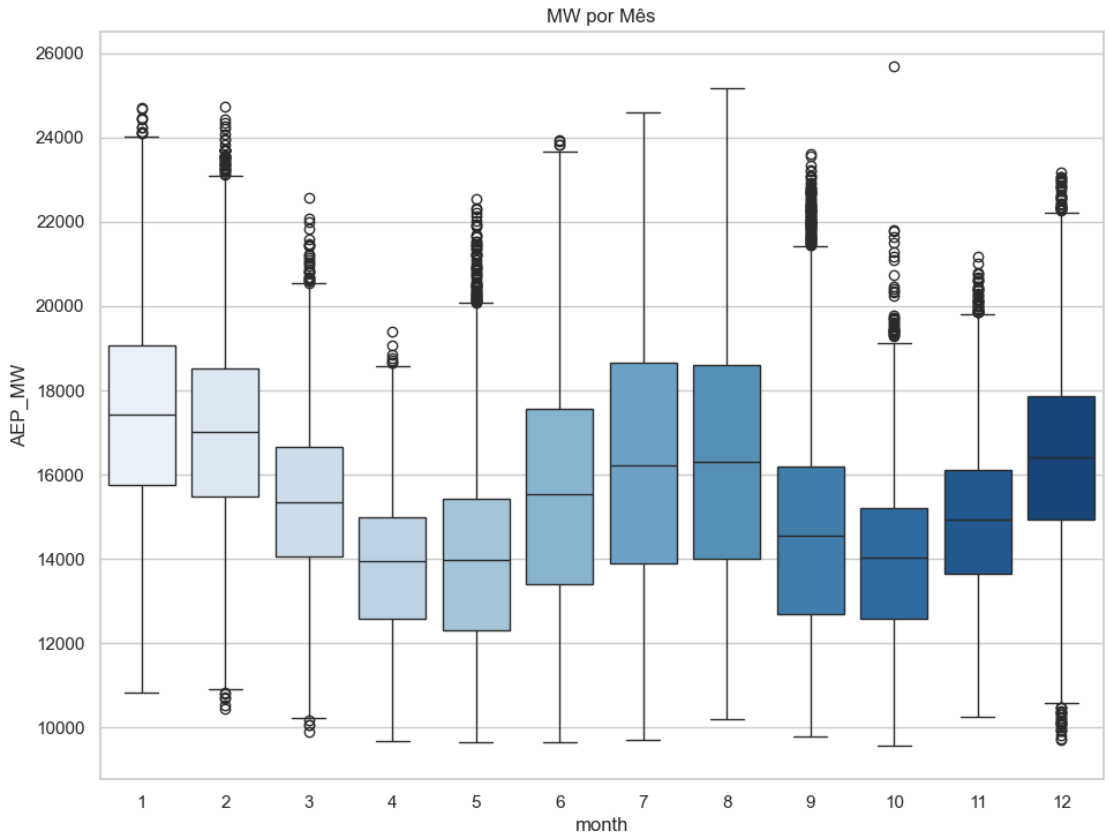

Figura 7 – Box Plot dos dados de consumo do Mês - Fonte: Autor

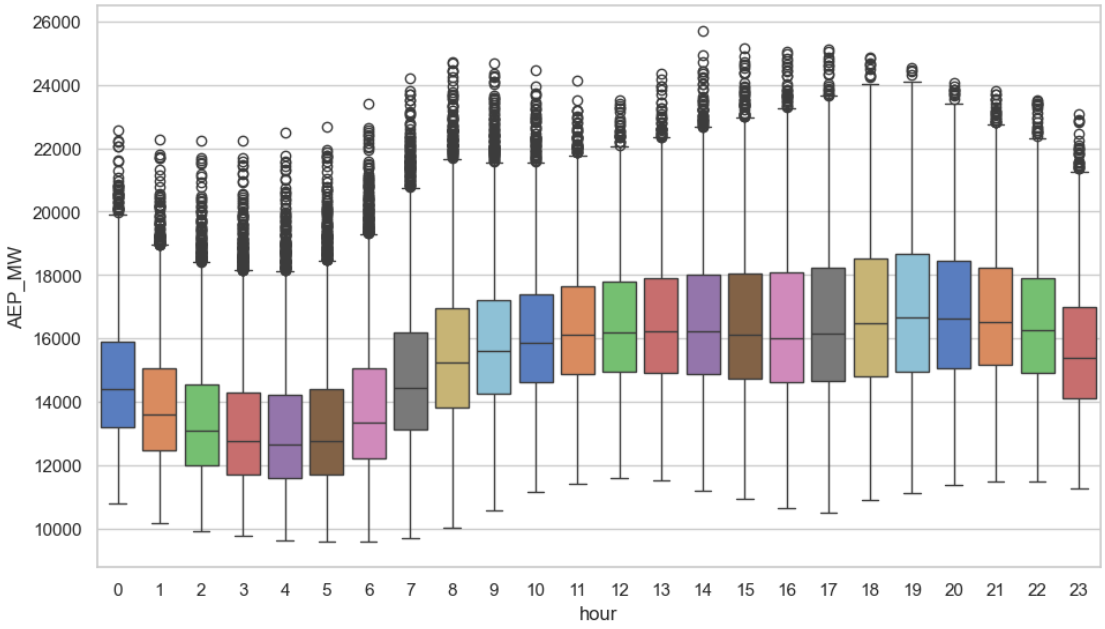

Figura 8 – Box Plot dos dados de consumo das Horas - Fonte: Autor

Com a figura 8 e 9 , observa-se que os maiores consumos de energia se dão no meio e no final do ano . Os horários de ponta observados são 17h às 21h e fora de ponta das 22h até 5 da manhã.

3.2.3 Entendimento de Negócio

A premissa fundamental adotada e sustentada por [Nti et al. 2020] é de que a habilidade de prever a demanda de energia auxilia na gestão e otimização das redes elétricas, trazendo previsibilidade às concessionárias de energia para estruturem suas operações.

Previsão de demanda é o principal problema de negócio para a referida problemática.

3.3 Preparação dos Dados

3.3.1 Verificar Nulos e Tipo de Dados

Dados nulos ou inexistentes podem causar problemas de desbalanceamento de classes. Então verifica-se a existência e o tipo de dados para garantir que AEP_MW seja um número real e a Data seja Data. Pela figura abaixo percebe-se a inexistência de dados nulos ou vazios em toda a base.

Além disso, a característica AEP_MW é do tipo Float e Datetime é do tipo Data, validando a etapa e indicando que pode-se continuar para o próximo passo do framework.

Figura 9 – Nulos e Tipos de Dados - Fonte: Autor

3.3.2 Preparar os dados para o modelo - XGBoost

A fase de preparação de dados no processo de modelagem envolve a criação de novos atributos a partir dos dados existentes, bem como a transformação desses dados em formatos mais adequados para a modelagem. No contexto do modelo XGBoost, foram desenvolvidas características adicionais. A construção de dados tem como objetivo refinar o conjunto de dados para maximizar a eficácia e a precisão do modelo de Machine Learning aplicado. Nesse processo, foram construídas características como dia da semana, hora, trimestre, mês, ano e semana do ano, todas fundamentais para o aprimoramento do modelo.

3.3.3 Preparar os dados para o modelo - Prophet

Segundo [Brownlee 2020] , os dados devem estar em um DataFrame do Pandas com uma estrutura específica: a primeira coluna deve ser nomeada 'ds' e conter as datas, enquanto a segunda coluna, nomeada 'y', deve conter as observações. Além disso, é importante que a coluna de datas esteja no formato de data e hora. O Prophet é projetado para ser fácil de usar e configurar automaticamente um bom conjunto de hiperparâmetros, tornando-o adequado para dados com tendências e estruturas sazonais Salientado essa ponto deve-se mudar o nome das colunas para ds e y:

```
aep_train.reset_index() \
    .rename(columns={'Datetime':'ds',
                     'AEP_MW':'y'}).head()
```

[ ]

...

	ds	y
0	2004-12-31 01:00:00	13478.0
1	2004-12-31 02:00:00	12865.0
2	2004-12-31 03:00:00	12577.0
3	2004-12-31 04:00:00	12517.0
4	2004-12-31 05:00:00	12670.0

Figura 10 – Dados formatados para o Prophet - Fonte: Autor

3.3.4 Teste de Kwiatkowski-Phillips-Schmidt-Shin (KPSS)

Segundo [Hyndman e Athanasopoulos 2007], existem alguns modelos , como por exemplo, o ARIMA e SARIMA que requerem que os dados estejam na condição de estacionaridade (variância e média não variam com o tempo). Conforme visto, os dados deste trabalho são não estacionários, pois apresentam sazonalidade. Para implementar modelos que requerem essa informação deve-se fazer uma transformação, como , por exemplo, a diferenciação. Esta técnica envolve a subtração de uma observação pelo seu valor anterior. No caso de séries temporais com tendências ou padrões sazonais, a diferenciação pode ajudar a remover essas características, tornando a série mais estacionária.

Aplicando a diferenciação para gerar um novo conjunto de dados e , a posteriori, realizar o teste de Kwiatkowski-Phillips-Schmidt-Shin desse novo conjunto verificará a estacionariedade da nova série temporal testando a hipótese nula de que os dados são estacionários em relação a uma tendência. Um valor-p acima de 0,05 indica que os dados são estacionários em relação à tendência.

A análise da estacionariedade não é essencial para este estudo, já que os modelos de Machine Learning selecionados para análise não requerem essa condição. Contudo, vale ressaltar que a verificação da estacionariedade é um passo crucial para a aplicação de diversos outros modelos na área de Machine Learning e Séries Temporais.

3.3.5 Separação dos dados para treinamento e teste

Para treinar modelos de Machine Learning, é fundamental dividir os dados disponíveis em conjuntos de treinamento e teste. O conjunto de treinamento é usado para ensinar e ajustar o modelo, permitindo que ele aprenda padrões e relações nos dados. Já o conjunto de teste, que não é utilizado durante o treinamento, serve para avaliar a performance do modelo em dados não vistos anteriormente. Esta separação é crucial para garantir que o

modelo generalize bem e não apenas memorize os dados de treinamento, um fenômeno conhecido como sobreajuste.

Para a divisão dos dados de treinamento e teste, foi estabelecido que o dia 1º de Janeiro de 2015 funcionará como um ponto de corte. Datas anteriores a esta serão utilizadas para o conjunto de treinamento, enquanto que datas posteriores serão destinadas ao conjunto de teste. Essa abordagem garante que o modelo seja treinado com dados históricos e testado com informações mais recentes, facilitando a avaliação de sua capacidade de generalização e eficácia em dados novos.

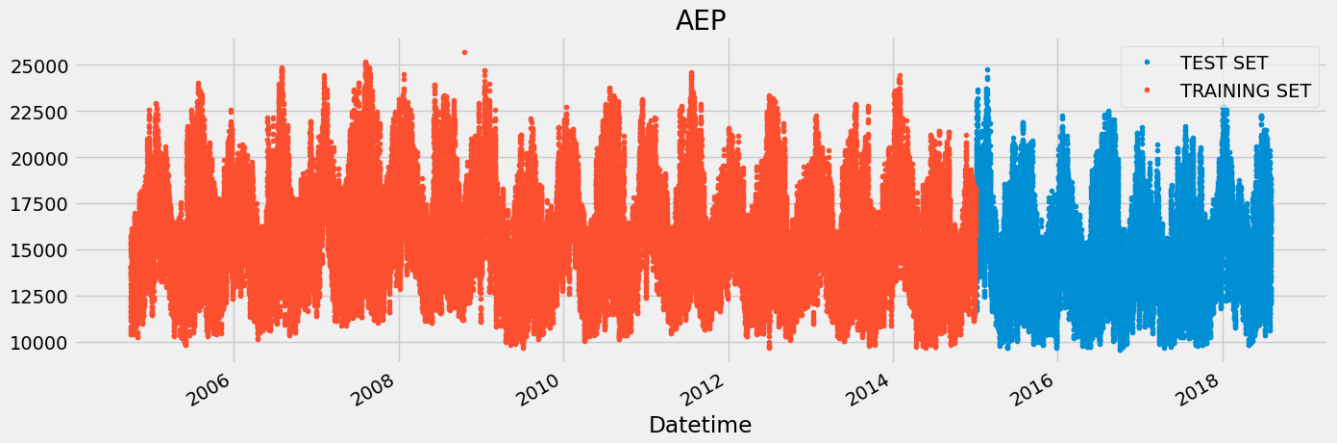

Figura 11 – Divisão em Training e Test Set - Fonte: Autor

3.4 Modelagem

Nesta etapa, o foco foi criar e aplicar os modelos Prophet e XGBoost para a previsão de séries temporais. O modelo Prophet, conhecido por sua eficácia em capturar tendências e padrões sazonais, foi empregado para analisar e prever padrões subjacentes nos dados. Paralelamente, bem como o XGBoost. O resultado desta etapa consiste nas previsões geradas por ambos os modelos, oferecendo insights para o problema de negócio.

3.5 Avaliação

A avaliação dos modelos de Machine Learning neste trabalho envolveu o cálculo dos erros associados a cada algoritmo, comparando valores reais com os previstos. Utilizaram-se o erro quadrático médio e o erro absoluto para medir a precisão dos modelos Prophet e XGBoost. Conforme o princípio de realimentação do CRISP-ML(Q), havia a possibilidade de iteração adicional no caso de resultados insatisfatórios. No entanto, neste estudo, não foram realizadas iterações adicionais. Uma potencial iteração futura poderia envolver a ajuste dos parâmetros dos modelos, incluindo feriados nacionais, para avaliar os impactos nas previsões.

4 Resultados e Discussões

4.0.1 Comparação dos Modelos

Feito o treinamento plota-se , de forma individual, gráficos que visam demonstrar uma comparação entre valores reais e valores previstos para cada tipo de modelo. O modelo que mais se aproximar do valor real, terá erros associados menores.

4.0.1.1 Horizonte de Previsão

Na figure 13 , percebe-se que o valor que o PHOPHET previu segue o padrão dos dados reais, em preto, mantendo a sazonalidade e o comportamento quando comparado ao real. Além disso, é importante salientar que quanto maior o horizonte de previsão maior é o grau de incerteza da previsão (destacado no azul claro). O mesmo fenômeno é observado no XGBoost]

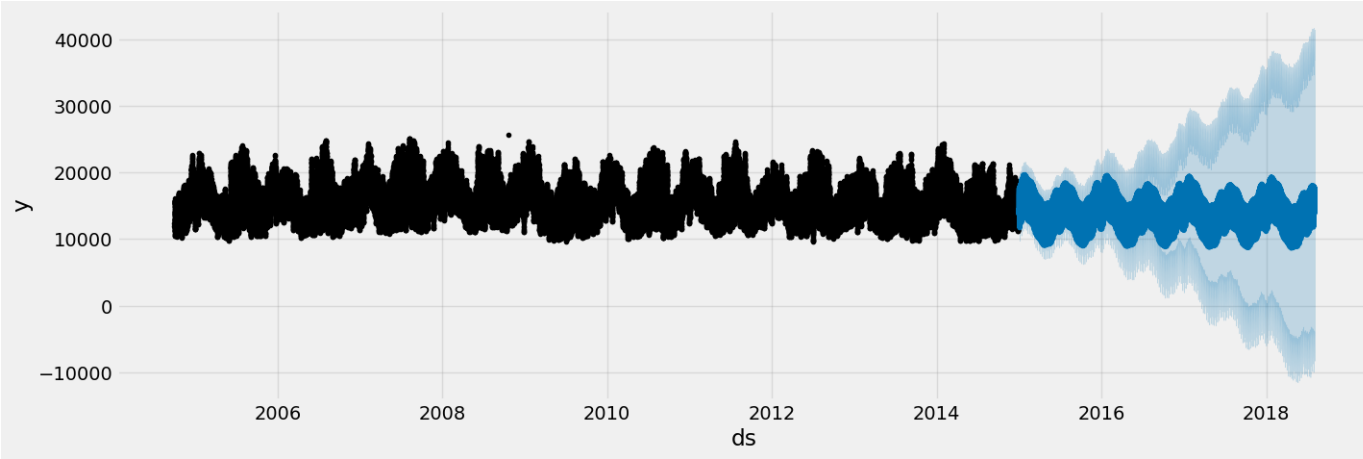

Figura 12 – Demonstração do Horizonte de Previsão - Fonte: Autor

4.0.1.2 Comparação Gráfica a Nível Mensal

Comparando os modelos a nível mensal , observa-se que a sazonalidade é mantida entre os dias, em ambos os modelos. Algumas tendências não são capturadas , contudo são detalhes pontuais que , possivelmente, com a melhoria dos parâmetros esse efeito diminua. De forma visual, observe-se que a previsão do XGBoost contorna melhor os dados reais quando comparado ao PROPHET

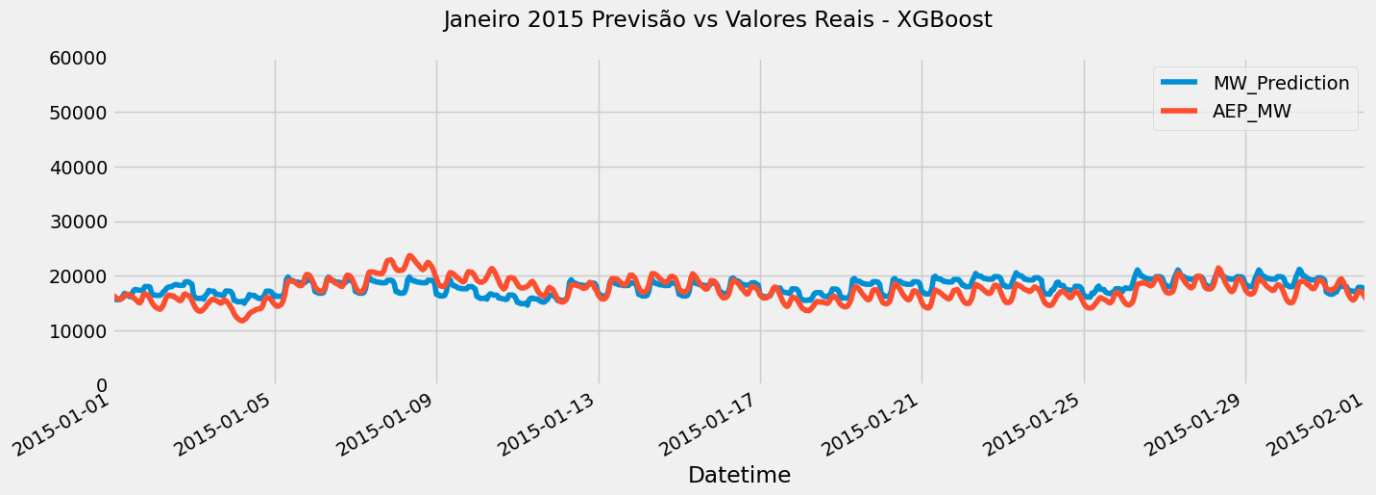

Figura 13 – XGBoost Previsão Janeiro 2015: Dados reais: Laranja, Dados Preditos: Azul - Fonte: Autor

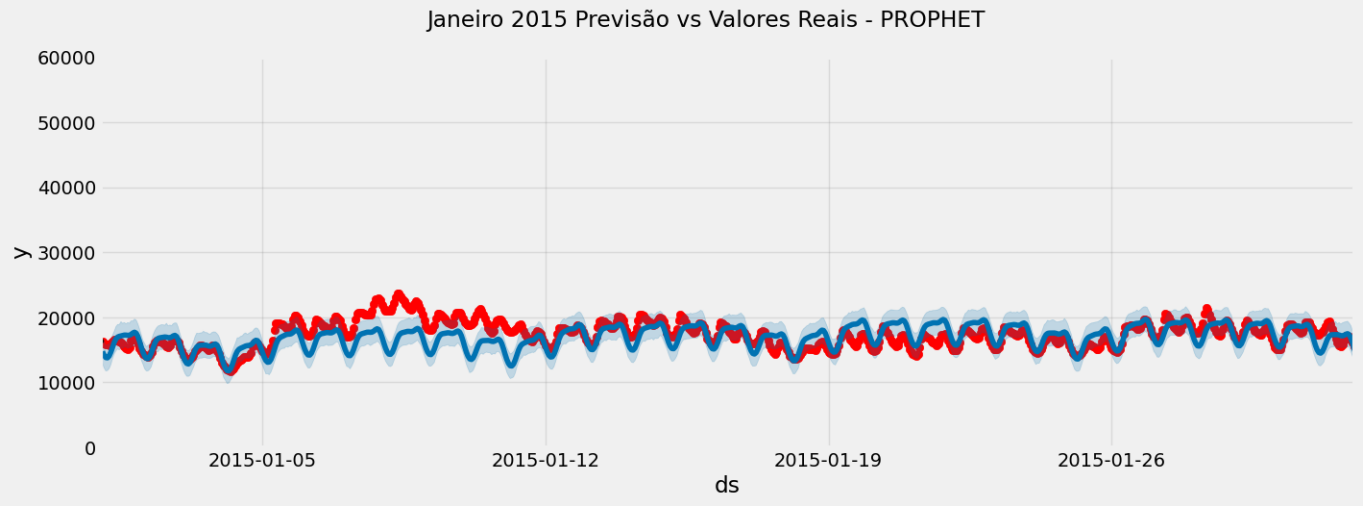

Figura 14 – XGBoost Previsão Janeiro 2015: Dados reais: Laranja, Dados Preditos: Azul - Fonte: Autor

4.0.1.3 Comparação Gráfica a Nível Semanal

Observa-se convergência com os dados reais nos dois modelos, demonstrando a captura das tendências de queda e alta no consumo de energia.

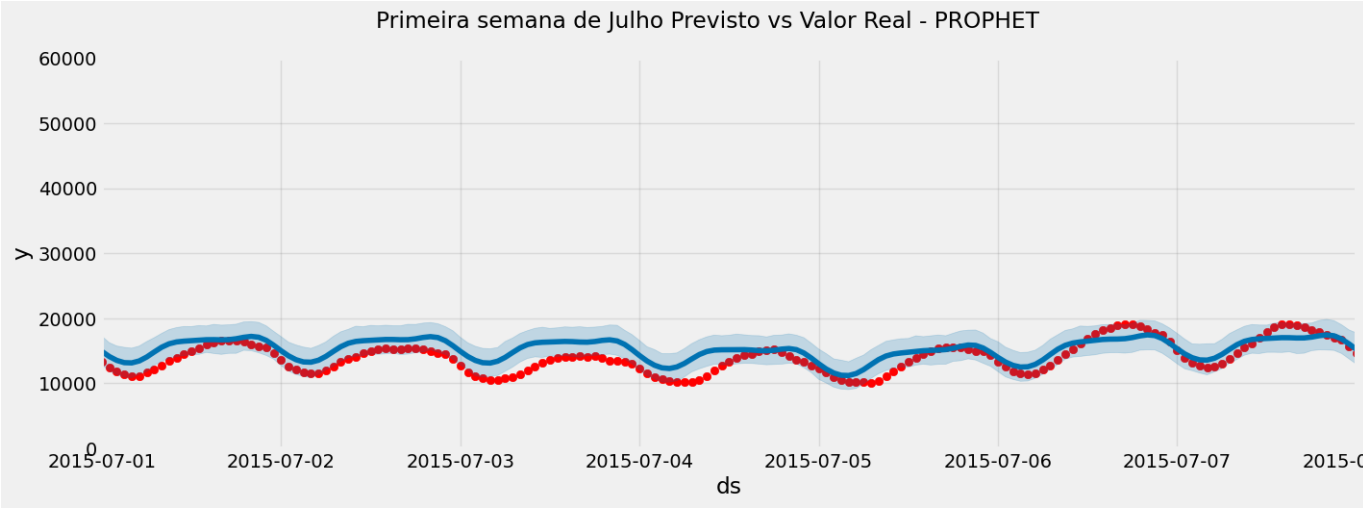

Figura 15 – Prophet Previsão Primeira Semana de Jul 2015: Dados reais: Laranja, Dados Preditos: Azul - Fonte: Autor

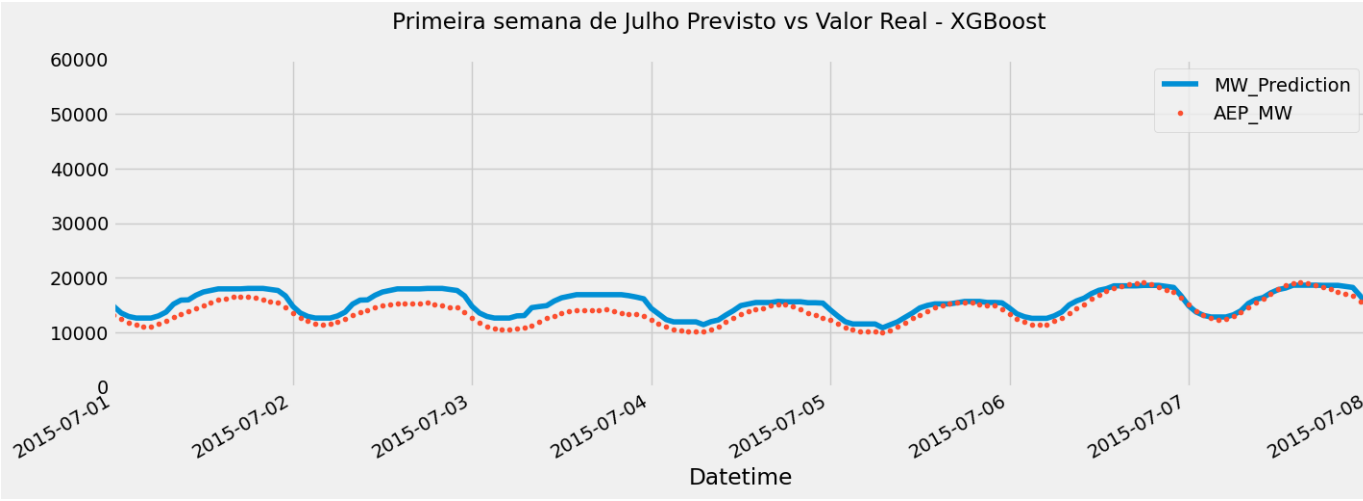

Figura 16 – XGBoost Previsão Primeira Semana de Jul 2015: Dados reais: Laranja, Dados Preditos: Azul - Fonte: Autor

4.0.2 Comparação dos Erros

Graficamente percebe-se que o XGBoost contorna melhor os dados. No entanto, para ter uma certeza quantitava é necessário calcular os erros associados de cada modelo: Erro Quadrático Médio e Erro Absoluto:

Modelo	Erro Quadrático Médio	Erro Absoluto
Prophet	[6772514]	[2080]MW
XGBoost	[2785781]	[1315]MW

Tabela 1 – Comparação dos Erros dos Modelos

Percebe-se que o modelo de melhor desempenho, com apenas uma iteração no *framework*, para essa análise é o XGBoost demonstrando menores erros associados e suas previsões se aproximando melhor dos dados reais

4.0.3 Próximos passos

Para análises futuras, planeja-se ajustar os parâmetros, incluindo considerações sobre dias de feriado, entre outros. Isso contribuirá para aprimorar a precisão dos modelos.

5 Conclusão

Neste trabalho, conduziu-se uma análise abrangente da previsão de consumo de energia elétrica, empregando os modelos Prophet e XGBoost. O objetivo foi avaliar a eficácia desses modelos na previsão de padrões de demanda.

Ao longo deste estudo, realizamos comparações das previsões com os dados reais de consumo de energia. Os resultados obtidos são satisfatórios par uma primeira análise , pois ambos os modelos, Prophet e XGBoost, demonstraram a capacidade de capturar a sazonalidade e o comportamento dos dados reais.

O modelo XGBoost se destacou visualmente, apresentando um ajuste mais próximo aos dados reais em várias análises. Na avaliação quantitativa, o XGBoost revelou menores erros associados, indicando uma maior precisão em suas previsões. Observou-se que, em análises mensais e semanais, ambas as abordagens mantiveram a sazonalidade dos dados e conseguiram capturar as tendências de queda e aumento no consumo de energia.

Os objetivos gerais e específicos do trabalho foram atendidos , pois foram implementados a maioria dos componentes presentes no framework CRISP-ML(Q).

Como próximos passos, sugere-se considerar a otimização dos parâmetros dos modelos, levando em conta fatores como feriados e eventos especiais, para melhorar ainda mais a precisão das previsões.

Referências

ABBAS, N.; NASSER, Y.; AHMAD, K. E. Recent advances on artificial intelligence and learning techniques in cognitive radio networks. *EURASIP Journal on Wireless Communications and Networking*, Springer, v. 2015, n. 1, p. 1–20, 2015. Citado na página 19.

BROWNLEE, J. *Time Series Forecasting With Prophet in Python*. 2020. <https://machinelearningmastery.com/time-series-forecasting-with-prophet-in-python/>. Acessado em: [13/12/2023]. Citado na página 32.

CERRI, R.; CARVALHO, A. C. P. d. L. F. d. Aprendizado de máquina: Breve introdução e aplicações. 2019. Disponível em: <https://seer.sct.embrapa.br/index.php/cct/article/view/26381/14242>. Citado na página 15.

DHALIWAL, S. S.; NAHID, A.-A.; ABBAS, R. Effective intrusion detection system using xgboost. *Information*, v. 9, n. 7, p. 149, 2018. Citado na página 23.

DUTT, S.; CHANDRAMOULI, S.; DAS, A. K. *Machine Learning*. [S.l.]: Pearson, 2023. Citado 2 vezes nas páginas 7 e 19.

FACELI, K. et al. *Inteligência Artificial: uma abordagem de aprendizado de máquina*. Rio de Janeiro: LTC, 2011. Citado na página 15.

FU, T. et al. Predicting peak day and peak hour of electricity demand with ensemble machine learning. *Frontiers in Energy Research*, v. 10, p. 944804, 2022. Citado na página 16.

HYNDMAN, R. J.; ATHANASOPOULOS, G. Arima and sarima models: A gentle introduction. *Journal of Statistical Software*, v. 27, n. 3, p. 1–22, 2007. Citado na página 33.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. *Foundations of Machine Learning*. [S.l.]: MIT Press, 2018. Citado na página 19.

NTI, I. K. et al. Electricity load forecasting: a systematic review. *Journal of Electrical Systems and Information Technology*, v. 7, p. 13, 2020. Citado 2 vezes nas páginas 15 e 31.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, v. 44, p. 206–226, 2000. <https://ieeexplore.ieee.org/abstract/document/5389202>. Citado na página 15.

STUDER, S. et al. Towards crisp-ml(q): A machine learning process model with quality assurance methodology. *Machine Learning and Knowledge Extraction*, v. 3, p. 392–413, 2021. Citado 2 vezes nas páginas 16 e 25.

TAYLOR, S. J.; LETHAM, B. Forecasting at scale. *The American Statistician*, Taylor & Francis, v. 72, n. 1, p. 37–45, 2018. Citado na página 23.

VANDERPLAS, J. *Python data science handbook: Essential tools for working with data*. [S.l.]: "O'Reilly Media, Inc.", 2016. Citado na página 24.