Evaluating the Effectiveness and Impact of Machine Learning Models for Forecasting Meal Demand in University Restaurants

Maria Fernanda Santos[1], **Janaina Gomide**[1]

[1]Instituto Politécnico – Universidade Federal do Rio de Janeiro (UFRJ)
Macaé – RJ – Brasil

{mariasantosunv, janainagomide}@gmail.com

Abstract. Research Context: *University restaurants face the challenge of balancing meal supply and demand, where efficient food management is essential to reduce waste and optimize costs, both crucial for service sustainability.* ***Scientific and/or Practical Problem:*** *Forecasting meal demand is complex due to consumption variability, the influence of holidays, and frequent menu changes.* ***Proposed Solution and/or Analysis:*** *This study applies machine learning models to forecast daily meal demand, using historical consumption data, holiday information, and menu attributes. The research also investigates the impact of these variables on forecast quality and accuracy.* ***Related IS Theory:*** *The study is based on Information Systems theory, particularly in the use of data and analytical intelligence to support decision-making. In this context, machine learning models act as information processors, capable of transforming real data into useful predictions for operational management.* ***Research Method:*** *The research followed the CRISP-DM framework: definition of objectives and context (Business Understanding); data collection and exploration (Data Understanding); cleaning and transformation (Data Preparation); training and calibration of models (Decision Tree, XGBoost, LSTM) (Modeling); performance evaluation with R^2 and comparison to the restaurant's forecasts (Evaluation); and preparation for practical application (Deployment).* ***Summary of Results:*** *The XGBoost model achieved the best performance, reaching R^2=0.92, outperforming the restaurant's original forecast (R^2=0.83). This represents a significant improvement in forecast accuracy, contributing to waste reduction and better inventory management. The inclusion of menu information was also shown to enhance prediction quality.* ***Contributions and Impact to IS area:*** *The study demonstrates how machine learning can be applied in institutional foodservice, providing operational, economic, and environmental benefits. It reinforces the role of Information Systems in enabling more accurate and sustainable decision-making.*

1. Introdução

A alimentação desempenha um papel essencial na rotina dos estudantes, fornecendo a nutrição necessária para manter o foco, a energia e o bom desempenho acadêmico. A gestão eficiente da oferta de refeições também contribui para

a redução do desperdício e para a otimização dos custos operacionais dos restaurantes universitários, aspectos cruciais para a sustentabilidade desse serviço. Em [United Nations Deputy Secretary-General 2024], a Organização das Nações Unidas destaca que programas de alimentação, como as refeições escolares, representam uma estratégia eficaz e de baixo custo para impulsionar o progresso em diversos Objetivos de Desenvolvimento Sustentável [ONU 2024], ao promover a segurança alimentar, a educação e a saúde infantil, reforçando a importância de investir em sistemas alimentares eficientes e acessíveis.

Além disso, alguns estudos têm evidenciado que a análise detalhada das refeições é relevante não apenas para a operação cotidiana, mas também para a construção de modelos preditivos de demanda, uma vez que o cardápio e os padrões de consumo exercem impacto direto sobre a quantidade necessária de refeições [Rodrigues et al. 2024]. Essa perspectiva é reforçada pela literatura mais ampla sobre aplicações de Inteligência Artificial na indústria alimentícia, que aponta que o uso de tecnologias avançadas, incluindo aprendizado de máquina, pode otimizar a produção, reduzir desperdícios e aprimorar a distribuição de alimentos, destacando a importância de compreender e monitorar o consumo alimentar de forma estruturada [Yang et al. 2025].

Nesse cenário, o uso de técnicas de aprendizado de máquina tem-se destacado como uma abordagem promissora para explorar grandes volumes de dados, identificar padrões de consumo e subsidiar o planejamento e a tomada de decisões em restaurantes universitários. Diversos estudos demonstram o potencial dessas técnicas na previsão de demanda, aplicando modelos de redes neurais a dados históricos de cardápios, períodos letivos e feriados, como nos restaurantes universitários da Universidade Federal de Viçosa [Pereira et al. 2007] e da Universidade Federal de Uberlândia [Santos et al. 2021].

Adicionalmente, a aplicação prática desses modelos se beneficia de interfaces amigáveis para os gestores, que permitem visualizar os resultados de forma clara e interativa. Nesse sentido, [Pereira et al. 2025] apresenta um sistema integrado que combina modelos preditivos com visualizações acessíveis e importação de dados provenientes de múltiplas fontes, evidenciando como a disponibilização intuitiva dos resultados pode potencializar o impacto das previsões.

Este estudo tem como objetivo aplicar técnicas de aprendizado de máquina para automatizar e aprimorar a previsão da demanda por refeições em um restaurante universitário da Universidade Federal do Rio de Janeiro do Centro Multidisciplinar de Macaé (UFRJ), que atualmente realiza esse processo de forma manual. A proposta busca tornar a previsão mais precisa e eficiente, reduzindo a dependência de estimativas subjetivas e minimizando erros que podem levar ao desperdício de alimentos ou à falta de refeições. Para isso, são utilizados dados históricos de consumo, cardápios e feriados, avaliando o desempenho de diferentes modelos preditivos, incluindo árvores de decisão, XGBoost e redes neurais LSTM, com foco em apoiar uma gestão alimentar mais sustentável e assertiva.

Assim, as principais contribuições deste artigo são: (i) avaliação da efetividade do aprendizado de máquina na melhoria das estimativas de refeições; (ii) análise do impacto do menu do restaurante na demanda prevista; e (iii) desenvolvimento e implantação de uma ferramenta para previsão automática das refeições baseada no modelo proposto.

2. Trabalhos relacionados

Diversos estudos evidenciam o papel central dos dados para apoiar a tomada de decisão em diferentes contextos. Por exemplo, [Bartolomei et al. 2022] investigou o planejamento urbano em dois municípios brasileiros, utilizando entrevistas semi-estruturadas com representantes das secretarias locais para identificar demandas de dados essenciais à tomada de decisão. O estudo demonstrou que a qualidade e a integração de diferentes tipos de dados, sociais, econômicos, históricos e ambientais, são fundamentais para estruturar soluções eficientes, como sistemas de apoio à decisão. Essa pesquisa evidencia que uma coleta de dados rigorosa e contextualizada é determinante para construir modelos preditivos confiáveis, destacando a relevância do gerenciamento de dados de entrada para suportar decisões estratégicas em qualquer setor.

A relevância da coleta e gestão de dados também se estende a outros contextos, como restaurantes e serviços de alimentação, nos quais decisões estratégicas dependem de informações confiáveis sobre consumo e demanda. [Kanwal et al. 2024] discute os fatores que contribuem para o desperdício de alimentos em diferentes etapas, desde o preparo até o consumo, identificando que aproximadamente 12–14% dos alimentos preparados nesses setores são desperdiçados globalmente. O estudo destaca os impactos econômicos, sociais e ambientais desse desperdício e demonstra que a aplicação de modelos de aprendizado de máquina para previsão de demanda pode alcançar acurácia de até 85%, fornecendo suporte para decisões mais precisas e redução de perdas. Esses achados evidenciam a relevância de integrar técnicas preditivas e gestão de consumo, fornecendo uma base conceitual sólida para pesquisas em contextos de restaurantes universitários.

Considerando a relevância da previsão de demanda para reduzir desperdícios, diversos estudos aplicaram técnicas de aprendizado de máquina em restaurantes universitários, avaliando diferentes modelos e tipos de dados.

Trabalhos iniciais utilizaram redes neurais artificiais do tipo Perceptron de Múltiplas Camadas (MLP) para estimar o número de refeições servidas. Em [Pereira et al. 2007], dados coletados entre janeiro e julho de 2006 na Universidade Federal de Viçosa incluíram o prato principal, o dia da semana e o consumo dos três dias anteriores. Apesar de o modelo alcançar um RMSE = 17,01, ele não superou o sistema de previsão da instituição (RMSE = 10,06). De forma semelhante, [Santos et al. 2021] avaliou MLP, KNN e Random Forest utilizando dados da Universidade Federal de Uberlândia (UFU) de 2015 e 2016, incluindo feriados e períodos acadêmicos. Alguns resultados ficaram abaixo da previsão humana, mas, em geral, os modelos superaram a média aritmética dos dados. Ainda assim, algumas estimativas não alcançaram o desempenho das previsões adotadas pela própria instituição, indicando que ainda há espaço para aprimoramento.

Abordagens posteriores incorporaram modelos mais sofisticados e sistemas computacionais acessíveis a gestores. Em [Rocha et al. 2011], dados de 254 dias de operação de um restaurante universitário entre 2008 e 2009 foram utilizados com oito variáveis de entrada, incluindo médias móveis e feriados. A rede neural apresentou erro médio de 9,5% e desempenho superior à média simples na maioria dos dias, embora com limitações em contextos atípicos. Já [Rodrigues et al. 2024] comparou LSTM, Transformer, Random Forest e LightGBM em três refeitórios, sendo dois universitários

e um corporativo, abrangendo histórico de consumo, condições climáticas e calendário acadêmico. Os resultados mostraram que Random Forest teve melhor desempenho nos ambientes universitários, enquanto LSTM se destacou no contexto corporativo.

Além de evidenciar a importância dos dados para a tomada de decisão, alguns estudos têm se dedicado à criação de ferramentas que tornam essa informação acessível e utilizável por gestores. Nesse sentido, [Pereira et al. 2025] apresentou o desenvolvimento do SmartAPSUS, um sistema voltado para a previsão de demanda na Atenção Primária à Saúde (APS) no Brasil. A ferramenta integra modelos de previsão baseados em técnicas de regressão e redes neurais (RNN, LSTM e GRU) com dados heterogêneos provenientes de diferentes fontes, como o IBGE, DATASUS e e-SUS, permitindo estimar com antecedência a quantidade de atendimentos futuros. O sistema foi projetado para oferecer uma experiência intuitiva, por meio de interfaces web com gráficos, mapas e tabelas, facilitando a interpretação dos resultados pelos gestores. A coleta e integração rigorosa de dados históricos, socioeconômicos e epidemiológicos mostraram-se fundamentais para garantir a confiabilidade das previsões, evidenciando como uma ferramenta bem estruturada pode apoiar decisões estratégicas, otimizar recursos e melhorar a qualidade do serviço prestado à população.

O estudo proposto se diferencia ao aplicar uma abordagem comparativa que envolve modelos baseados em árvores de decisão, como Árvore de Decisão e XGBoost, e redes neurais recorrentes, como LSTM, utilizando dados de uma universidade distinta. Além disso, a metodologia contempla diferentes formas de tratamento dos dados de entrada, incluindo variáveis relacionadas ao cardápio, feriados e datas anteriores, permitindo identificar quais combinações contribuem para um melhor desempenho preditivo. Para tornar esses resultados acessíveis e úteis para gestores, desenvolveu-se uma ferramenta amigável que integra o modelo de previsão, oferecendo visualizações claras e interativas. Essa solução demonstra como a análise de dados e o uso de técnicas preditivas podem ser aplicados de forma prática, apoiando a tomada de decisão e a otimização de recursos em restaurantes universitários.

3. Metodologia

Para o desenvolvimento deste estudo, foi adotada a metodologia *Cross Industry Standard Process for Data Mining* (CRISP-DM) que organiza o processo em seis etapas principais: compreensão do negócio, compreensão dos dados, preparação dos dados, modelagem, avaliação e implantação [Chapman 2000]. A figura 1 apresenta essas etapas e a seguir é feita a contextualização de cada etapa para o problema abordado nesse trabalho.

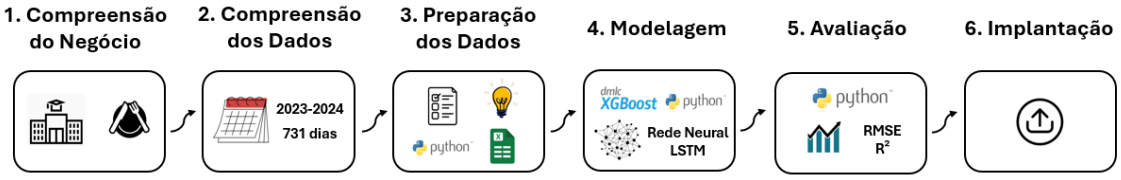


Figura 1. Resumo da metodologia aplicada no contexto do projeto.

Compreensão do Negócio: entendimento do problema do ponto de vista organizacional e definição da questão de pesquisa. Neste projeto, o objetivo é prever a demanda de refeições em um restaurante universitário localizado na cidade de Macaé,

no estado do Rio de Janeiro. Essa previsão apoia a gestão do restaurante na tomada de decisões relacionadas às compras, preparo e distribuição das refeições, contribuindo para a redução do desperdício e otimização de recursos.

Compreensão dos Dados: análise dos dados disponíveis e identificação de suas principais características e possíveis adversidades. No contexto deste trabalho, os dados incluem registros históricos do restaurante, como a quantidade diária de refeições servidas e datas. Avalia-se a completude, a presença de valores ausentes e a necessidade de criação de novas variáveis, orientando as etapas seguintes.

Preparação dos Dados: limpeza, transformação e organização dos dados para uso nos modelos preditivos. Neste estudo, foram realizadas tarefas como o tratamento de valores ausentes, a criação de variáveis derivadas e a normalização de variáveis numéricas, influenciando diretamente o desempenho dos modelos.

Modelagem: treinamento de modelos de aprendizado de máquina utilizando os dados preparados. Para este estudo, aplicaram-se três algoritmos distintos: Árvores de Decisão, XGBoost e *Long Short-Term Memory* (LSTM), cada um oferecendo uma abordagem diferente e perspectivas complementares à análise.

Inicialmente, foi utilizada uma árvore de decisão como modelo base. Esse algoritmo divide os dados em ramos de decisão, criando uma estrutura hierárquica que permite classificar ou prever valores com base em perguntas sobre os atributos. Foi escolhida por sua facilidade de interpretação e visualização. Além de ser rápida de treinar, essa abordagem permite identificar quais variáveis mais influenciam o comportamento do modelo, fornecendo insights valiosos que podem orientar ajustes em algoritmos mais complexos [Géron 2022].

O XGBoost, que é uma versão mais robusta baseada em árvores de decisão, combina múltiplas árvores em sequência usando *boosting* e gradiente descendente para otimizar gradualmente os erros. Cada árvore adicional corrige os erros das anteriores, aumentando a precisão e a robustez do modelo [Chen and Guestrin 2016]. O XGBoost trata as observações como independentes, permitindo compreender o impacto isolado de cada variável em cada dia, sendo especialmente eficiente em tarefas com dados estruturados.

A rede neural recorrente do tipo LSTM foi escolhida para capturar a dimensão temporal dos dados. Diferentemente dos modelos baseados em árvores, a LSTM mantém informações de longo prazo por meio de unidades de memória específicas, sendo capaz de identificar padrões sequenciais e tendências ao longo do tempo [Hochreiter and Schmidhuber 1997]. Essa característica a torna particularmente eficaz para a modelagem de séries temporais.

A comparação entre esses modelos justifica-se pelas abordagens distintas que oferecem: enquanto a árvore de decisão prioriza explicabilidade e simplicidade, o XGBoost busca maior precisão por meio da combinação de múltiplas árvores, e a LSTM explora a dependência temporal dos dados. Essa diversidade permite avaliar qual estratégia oferece melhor desempenho na previsão da demanda de refeições, considerando tanto relações estruturais quanto padrões sequenciais.

Avaliação: medição do desempenho dos modelos com base em métricas

quantitativas, verificando se atingem os objetivos propostos [Géron 2022]. As métricas utilizadas foram o RMSE (*Root Mean Squared Error*) e R^2 (Coeficiente de Determinação), conforme fórmulas a seguir:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

onde n representa o número total de observações, y_i é o valor real (observado) da i-ésima instância, $\hat{y}_i$ é o valor previsto pelo modelo para essa instância, e $\bar{y}$ é a média dos valores observados. O RMSE mede o erro médio entre os valores previstos e os reais, penalizando mais fortemente erros maiores, enquanto o R^2 indica a proporção da variância dos dados explicada pelo modelo, variando de 0 a 1.

Além das métricas, a Avaliação incluiu a comparação das previsões do modelo com os dados reais do restaurante, permitindo verificar se os modelos oferecem melhoria em relação ao método tradicional. Também foram utilizadas técnicas de interpretação, como a Importância da Covariável por Permutação, que evidencia quais variáveis têm maior impacto no desempenho do modelo. Visualizações gráficas, como plots de valores observados versus previstos, foram geradas para facilitar a interpretação e a identificação de padrões capturados pelos modelos.

Implantação: aplicação prática do modelo selecionado por meio de uma ferramenta interativa que suporta a tomada de decisão. A ferramenta, implementada em Python com Streamlit e hospedada no GitHub, permite ao usuário inserir parâmetros relevantes, como feriado, prato do dia e quantidade de refeições anteriores, e obter previsões de forma interativa. Nessa etapa, o foco está em disponibilizar o modelo de forma operacional, facilitando a utilização prática pelos gestores do restaurante.

4. Resultados

Nesta seção, são apresentados os resultados de cada etapa da metodologia proposta. Durante a **Compreensão do negócio e dos dados**, houve contato com a gerência do restaurante universitário, que forneceu planilhas de janeiro de 2023 a dezembro de 2024. Esses arquivos incluíam informações sobre a data, o cardápio, a quantidade real de refeições servidas e a previsão de refeições do restaurante para cada dia. As planilhas de 2024 também continham dados de jantar, iniciados naquele ano, além dos dados de almoço, já disponíveis em 2023.

A primeira tarefa consistiu em unificar todas as planilhas em uma única base de dados, consolidando informações mensais para facilitar a análise. Optou-se por trabalhar exclusivamente com os dados de almoço, devido à maior base histórica disponível. Também foram considerados apenas os dias úteis, uma vez que o restaurante não opera aos sábados e domingos, que, portanto, não entraram no modelo. Os feriados, entretanto, foram mantidos, por representarem eventos relevantes para a variação da demanda.

Com a base de dados unificada pronta, iniciou-se a **Preparação dos dados** com a manipulação dos atributos necessários para a construção do modelo. A linguagem de programação utilizada é Python e todo o código foi desenvolvido no ambiente do Google Colaboratory[1]. Foram criadas variáveis para identificar períodos de férias e feriados, incluindo classificações de pré e pós-feriado. Para isso, elaborou-se uma lista com todas as datas correspondentes a esses períodos e verificou-se, para cada registro da base, se ele se enquadrava em alguma dessas classificações. Caso positivo, a variável recebeu o valor 1; caso contrário, recebeu 0. Essa codificação em 0 e 1, conhecida como variável binária, permite que o modelo interprete de forma objetiva a presença ou ausência de determinado evento em cada dia, facilitando a identificação de padrões de demanda relacionados a feriados e períodos especiais.

Além do tratamento anterior, foram desenvolvidas duas funções específicas para flexibilizar e estruturar a preparação dos dados para os testes. A primeira função lida com as variáveis temporais, como dia da semana e mês, e possui um parâmetro booleano. Quando definido como True, a função transforma as variáveis em sua versão cíclica usando funções trigonométricas (seno e cosseno), representando de forma mais adequada a natureza periódica dos dados. Por exemplo, na codificação inteira, segunda-feira (0) parece distante de sexta-feira (4), embora estejam próximas no ciclo semanal. Quando definido como False, os valores permanecem como inteiros (0 a 4 para dias úteis e 1 a 12 para meses), mantendo a codificação tradicional sem considerar a periodicidade.

A segunda função está relacionada ao uso do cardápio. Inicialmente, foram considerados 95 pratos distintos, muitos com pequenas diferenças, como temperos ou tipos de molho, além de apresentarem inconsistências na grafia e na escrita dos nomes. Para reduzir essa variabilidade, a função agrupa pratos semelhantes em categorias mais amplas, mantendo separados apenas os pratos com maior demanda ou percepção diferenciada pelos consumidores. Alguns exemplos de categorias e pratos incluídos são:

- Peixe e frutos do mar: filé de peixe grelhado com molho de limão, bobó de camarão, estrogonofe de camarão;
- Suíno: lombo suíno ao molho de limão, mignon suíno ao molho de cebola, guisado de lombinho ao limão;
- Misto: picadinho misto ao molho ferrugem, goulash misto, churrasquinho misto;
- Aves cremosas: fricassê de frango, estrogonofe de frango, lasanha de frango;
- Bovino cremoso: estrogonofe de carne, lasanha à bolonhesa, lasanha de carne;
- Aves: frango à pizzaiolo, escalope de frango ao molho shoyo, frango ao molho mostarda;
- Bovino: picadinho de carne ao molho espanhol, carne ao molho de champignon, almôndegas de carne.

Além dessas categorias, foram criadas duas colunas adicionais para situações especiais:

1. Prato sem serviço: dias em que não houve oferta de refeições.
2. Prato não informado: dias em que houve atendimento, mas a informação sobre o prato servido estava ausente.

[1]Google Colab: https://colab.google/

A partir desse pré-processamento, aplica-se a técnica de one-hot encoding, transformando cada categoria de prato em uma coluna binária. Cada coluna indica a presença (1) ou ausência (0) da categoria no dia, permitindo que o modelo aprenda padrões na demanda associados aos tipos de prato.

A função foi implementada de forma flexível: quando o parâmetro é True, ela realiza o agrupamento, a padronização e a codificação do cardápio; quando o parâmetro é False, a coluna de pratos é removida do conjunto de dados, e o cardápio não é considerado na construção do modelo. A Tabela 1 apresenta a distribuição final dos pratos, considerando o agrupamento realizado pela função, incluindo as categorias especiais.

Tabela 1. Distribuição dos pratos principais no conjunto de dados.

ID	Prato Principal	Qtd	ID	Prato Principal	Qtd
1	Aves	113	6	Aves cremosas	32
2	Bovino	108	7	Peixe e frutos do mar	25
3	Sem serviço	100	8	Prato não informado	13
4	Suíno	65	9	Bovino cremoso	12
5	Misto	54			

Em seguida, foi definido o atributo de data como índice do conjunto de dados. Durante esse processo, identificou-se que dois dias, 27 e 30 de setembro de 2024, apresentavam a variável de quantidade de refeições igual a zero sem justificativa evidente, como feriados ou recessos. Para evitar distorções no desempenho dos modelos, optou-se pela remoção dessas duas observações.

Nos casos em que a variável de quantidade de refeições estava ausente (valores NaN), observou-se que essas situações correspondiam a períodos de férias ou feriados, já indicados pelas variáveis auxiliares. Nesses casos, os valores ausentes foram substituídos por zero, refletindo corretamente a realidade do restaurante, em que não houve oferta de refeições.

Além disso, foram criados atributos que representam a quantidade de refeições servidas nos cinco dias úteis imediatamente anteriores a cada observação. Esses valores, conhecidos como lags, permitem ao modelo considerar o histórico recente de consumo e identificar tendências de curto prazo. Para cada dia considerado, o modelo recebe informações sobre a demanda dos cinco dias anteriores, considerando apenas os dias úteis (segunda a sexta-feira), seguindo a ordem cronológica dos registros.

As janelas de cinco dias são construídas de forma sobreposta, de modo que cada novo dia considerado compartilha parte dos dias anteriores com a janela anterior. Essa organização ajuda o modelo a capturar padrões contínuos na demanda, incluindo tendências recentes e oscilações semanais, refletindo a lógica de previsão aplicada no dia a dia pela administração. A Tabela 2 apresenta a configuração final dos atributos disponíveis para a construção dos modelos.

Na etapa de **Modelagem**, definiu-se como variável-alvo a quantidade de refeições servidas, enquanto as demais variáveis foram utilizadas como atributos de entrada. A coluna referente à previsão do restaurante foi excluída desse processo, sendo utilizada

Tabela 2. Atributos utilizados na criação dos modelos.

Atributo	Descrição
Quantidade de refeições diárias	Total de refeições servidas por dia.
Férias	1 se for período de férias, 0 caso contrário.
Feriado	1 se o dia for feriado, 0 caso contrário.
Pré-feriado	1 se o dia for anterior a um feriado, 0 caso contrário.
Pós-feriado	1 se o dia for posterior a um feriado, 0 caso contrário.
Dia da semana	Representado numericamente (0–4) ou ciclicamente (seno e cosseno).
Mês	Representado numericamente (1–12) ou ciclicamente (seno e cosseno).
Cardápio	Codificação one-hot dos pratos (uso opcional).
Valores dos 5 dias anteriores	Quantidade de refeições nos 5 dias anteriores.

apenas para comparação com os resultados obtidos pelos modelos. Foram avaliadas quatro combinações possíveis de pré-processamento, resultantes das decisões de manter ou não as variáveis temporais em formato cíclico e de considerar ou não o cardápio. Dessa forma, os cenários testados foram:

- variáveis temporais cíclicas com cardápio;
- variáveis temporais cíclicas sem cardápio;
- variáveis temporais não cíclicas com cardápio;
- variáveis temporais não cíclicas sem cardápio.

Essa sistematização permitiu avaliar o impacto de cada escolha no desempenho final dos modelos.

Os modelos de aprendizado de máquina, árvore de decisão, XGBoost e LSTM, foram implementados em Python utilizando a biblioteca *sci-kit learn*[2] e o código está disponível no GitHub[3].

O conjunto de dados foi dividido em treino (80%) e teste (20%) para os modelos de Árvore de Decisão e XGBoost. Como os dados são temporais, o conjunto de treino corresponde aos primeiros 80% dos registros, preservando a ordem cronológica. A partir dessa divisão, realizou-se a busca pelos melhores hiperparâmetros de cada modelo utilizando validação cruzada com 5 folds, automatizada pelo método *GridSearchCV* disponível na biblioteca *sci-kit learn*. A validação cruzada consiste em dividir o conjunto de treino em cinco partes (*folds*) e treinar o modelo em quatro delas, utilizando a parte restante como conjunto de validação, repetindo esse processo cinco vezes. Essa técnica permite avaliar de forma mais robusta o desempenho do modelo antes de aplicá-lo ao conjunto de teste.

Após a divisão dos dados em conjuntos de treino e teste, os cinco primeiros dias de cada conjunto foram removidos por motivos distintos. No treino, esses dias não possuíam

[2]Biblioteca sci-kit learn link: https://scikit-learn.org/

[3]Link para GitHub do projeto:https://github.com/MariaFMSantos/ml-meal-demand

dados anteriores suficientes para criar os atributos de cinco dias anteriores (lags); por exemplo, o primeiro dia disponível, 1º de janeiro de 2023, não tinha informações dos cinco últimos dias de dezembro de 2022. No conjunto de teste, a remoção foi necessária para evitar vazamento de dados. Por exemplo, se o teste começa em 1º de junho de 2023, os cinco dias anteriores (27 a 31 de maio) pertencem ao conjunto de treino e, na prática, não estariam disponíveis no momento da previsão.

Dessa forma, os cinco dias foram eliminados de cada conjunto e, para garantir comparabilidade com a previsão realizada pelo restaurante, os mesmos dez dias correspondentes também foram excluídos da previsão deles. Isso assegura que o cálculo das métricas seja feito sobre a mesma base de dados, permitindo uma comparação justa entre os modelos e a previsão original do restaurante.

Para o modelo LSTM, os dados foram divididos em três conjuntos: treino com 64%, validação com 16% e teste com 20%, preservando a ordem dos registros. O treino ajusta os pesos internos da rede, a validação monitora o desempenho para ajustes de hiperparâmetros e o teste avalia a acurácia final. Assim como nos demais modelos, os cinco primeiros dias de cada conjunto foram removidos, totalizando quinze dias eliminados.

A variável-alvo foi normalizada para o intervalo de 0 a 1, procedimento que auxilia na estabilidade do treinamento e melhora a capacidade da rede de lidar com diferentes magnitudes de valores.

A estrutura da rede foi testada em dois formatos distintos: um com uma camada LSTM seguida de uma camada densa, e outro com duas camadas LSTM antes da camada densa final. Os resultados dessas duas abordagens são apresentados separadamente. Assim como no XGBoost, as quatro combinações de atributos, com ou sem variáveis cíclicas e com ou sem cardápio, também foram testadas, permitindo uma análise comparativa completa entre os diferentes cenários. A Tabela 3 apresenta um resumo dos algoritmos utilizados, os hiperparâmetros considerados em cada caso e os valores avaliados durante o processo de ajuste.

Tabela 3. Hiperparâmetros utilizados na busca por melhores modelos.

Algoritmo	Hiperparâmetro	Valores Testados
Árvore de Decisão	max_depth	[3, 4, 5, 6, 7, 8, 10, 12]
	min_samples_split	[2, 5, 10, 15, 20]
	min_samples_leaf	[1, 2, 4, 6, 8]
	max_features	[“None”, “sqrt”, “log2”]
	criterion	[“squared_error”, “absolute_error”]
XGBoost	max_depth	[3, 4, 5, 6, 7, 8]
	n_estimators	[300, 400, 500, 600, 700]
	learning_rate	[0.015, 0.020, 0.025, 0.05]
	base_score	[0.5]
	booster	[“gbtree”]
	objective	[“reg:squarederror”]
LSTM / LSTM (Camada Extra)	learning_rate	[0.005, 0.01, 0.025, 0.05]
	units	[5, 10, 20, 25]
	batch_size	[4, 8, 16]

Após a construção dos modelos, a etapa de **Avaliação** teve como objetivo comparar os resultados obtidos para cada combinação de atributos e configurações, conforme apresentados na Tabela 4. Nela, os valores de teste que superaram ou igualaram a previsão manual foram destacados em negrito para a métrica em questão, os melhores valores de teste de cada métrica foram sublinhados e a melhor combinação de atributos para cada modelo (considerando ou não as variáveis temporais cíclicas e o uso do cardápio) foi indicada em itálico. Essa formatação facilita a visualização das combinações que apresentaram melhor desempenho para cada algoritmo.

Tabela 4. Comparativo dos valores de RMSE e R^2 entre modelos de aprendizado de máquina e previsão do restaurante.

Modelo / Referência	Combinação		RMSE			R^2		
	Cardápio	Cíclica	Treino	Val.	Teste	Treino	Val.	Teste
Árvore de Decisão	*Sim*	*Sim*	–	59.58	**64.62**	–	0.91	**0.83**
	Sim	Não	–	79.58	83.76	–	0.84	0.72
	Não	Sim	–	50.90	**67.90**	–	0.93	0.82
	Não	Não	–	57.55	**70.64**	–	0.91	0.80
XGBoost	*Sim*	*Sim*	–	26.93	**<u>43.69</u>**	–	0.98	**<u>0.92</u>**
	Sim	Não	–	35.05	**47.82**	–	0.97	**0.91**
	Não	Sim	–	7.57	**61.46**	–	0.99	**0.85**
	Não	Não	–	27.45	**58.22**	–	0.98	**0.87**
LSTM	Sim	Sim	71.43	95.07	104.88	0.87	0.70	0.56
	Sim	Não	57.17	90.13	105.87	0.92	0.73	0.55
	Não	*Sim*	69.11	90.28	**78.38**	0.88	0.73	0.75
	Não	Não	77.97	93.52	123.89	0.85	0.70	0.43
LSTM (Camada Extra)	Sim	Sim	50.71	94.91	107.77	0.94	0.70	0.53
	Sim	Não	61.05	93.51	107.59	0.91	0.71	0.54
	Não	*Sim*	57.81	88.07	82.94	0.92	0.74	0.72
	Não	Não	69.66	95.68	103.49	0.88	0.69	0.60
Restaurante	-	-	-	-	79.65	-	-	0.83

Como mostrado na Tabela 4, o desempenho do restaurante apresentou um RMSE = 79,65 e $R^2 = 0,83$. Para calcular essas métricas, utilizou-se como valor previsto a previsão atualmente fornecida pelo restaurante, que envolve certa subjetividade, e como valor real os quantitativos efetivamente observados. Alguns modelos de aprendizado de máquina obtiveram resultados superiores às previsões atualmente realizadas pelo restaurante, demonstrando maior precisão na estimativa do número de refeições.

O XGBoost destacou-se como o modelo mais eficiente. A menor taxa de erro (RMSE = 43,69) e o maior valor de R^2 (0,92) no conjunto de teste foram obtidos quando foram utilizadas tanto as variáveis do cardápio quanto as transformações cíclicas das datas. Esses resultados indicam que a inclusão de informações detalhadas sobre as refeições e a representação periódica das datas foram fundamentais, permitindo que o algoritmo capturasse padrões complexos nos dados e modelasse interações não lineares entre as variáveis.

A Árvore de Decisão, embora mais simples, também apresentou resultados satisfatórios, especialmente quando combinou informações do cardápio com variáveis cíclicas (RMSE = 64,62, R^2 = 0,83). Isso evidencia que, mesmo modelos mais básicos, podem se beneficiar significativamente do pré-processamento adequado das variáveis

temporais e categóricas.

Os modelos baseados em LSTM, com uma ou duas camadas, obtiveram desempenho inferior ao XGBoost, apesar de considerarem a ordem temporal dos dados. O melhor resultado entre os LSTMs ocorreu sem a utilização do cardápio e com variáveis cíclicas (RMSE = 78,38, R^2 = 0,75 para LSTM; RMSE = 82,94, R^2 = 0,72 para LSTM com camada extra). Essa diferença de desempenho pode estar relacionada ao tamanho limitado do conjunto de dados e à necessidade de ajustes mais refinados na arquitetura e nos hiperparâmetros da rede.

Além disso, observou-se que os modelos LSTM que incluíam o cardápio apresentaram desempenho inferior aos que utilizavam apenas variáveis temporais. Isso não significa que o cardápio não influencie a demanda, mas sim que, nas condições testadas, a rede não conseguiu extrair padrões adicionais de forma consistente. Em contraste, o XGBoost conseguiu identificar relações mais complexas entre as variáveis, indicando que o cardápio pode contribuir para melhorar as previsões, ainda que seu potencial não tenha sido plenamente capturado pelos modelos recorrentes.

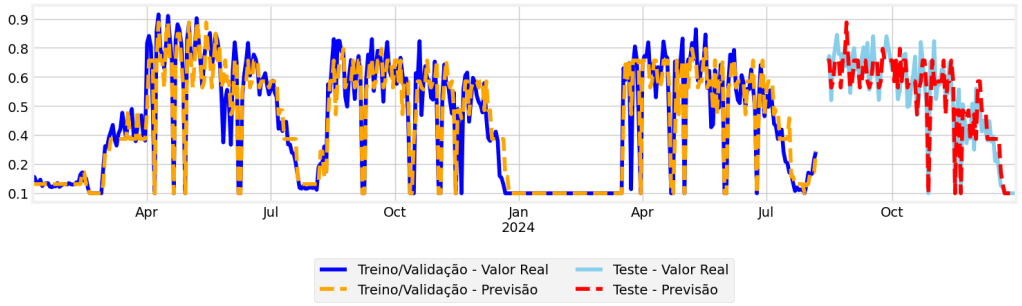
A Figura 2 apresenta, de forma visual, a quantidade de refeições oferecidas (normalizada para garantir o sigilo do restaurante) e as previsões realizadas pelos melhores modelos de cada algoritmo, permitindo uma interpretação intuitiva da proximidade entre valores observados e previstos.

A Tabela 5 apresenta os hiperparâmetros utilizados nos melhores modelos de cada algoritmo. Para a Árvore de Decisão, o melhor desempenho foi obtido com profundidade máxima de 5, divisão mínima de 4 amostras e folhas com no mínimo 10 amostras. Essa configuração reforça o controle de sobreajuste, pois o modelo somente cria regras quando sustentadas por um número maior de observações, mantendo a árvore simples e mais generalizável. A ausência de limitação em $max\_features$ indica que o uso de todas as variáveis disponíveis, incluindo cardápio e variáveis cíclicas, foi benéfico.

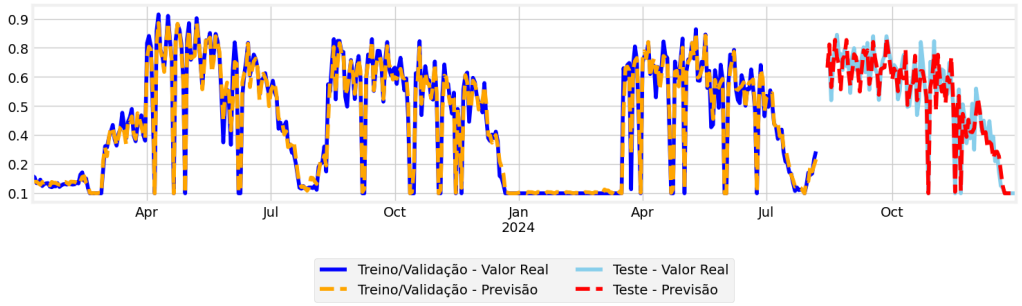
Tabela 5. Hiperparâmetros dos melhores modelos de cada algoritmo.

Algoritmo	**Combinação**		**Hiperparâmetros**
	Cardápio	**Cíclica**	
Árvore de Decisão	Sim	Sim	criterion = “squared_error”, max_depth = 5, max_features = “None”, min_samples_leaf = 4, min_samples_split = 10
XGBoost	Sim	Sim	base_score = 0.5, booster = “gbtree”, learning_rate = 0.015, max_depth = 4, n_estimators = 600, objective = “reg:squarederror”
LSTM	Não	Sim	learning_rate = 0.05, units = 25, batch_size = 16
LSTM (Extra)	Não	Sim	learning_rate = 0.05, units = 25, batch_size = 16

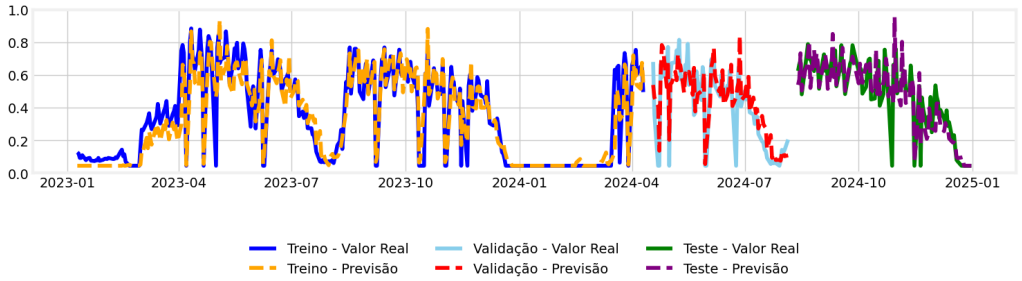
O XGBoost, que apresentou o melhor desempenho geral, foi ajustado com profundidade de árvore relativamente baixa ($max\_depth = 4$), $learning\_rate = 0.015$ e 600 estimadores. Essa configuração favorece um aprendizado progressivo e estável, permitindo que o modelo construa um grande número de árvores simples para captar relações mais sutis nos dados, reduzindo o risco de overfitting associado a árvores muito profundas. O melhor resultado foi obtido na configuração com cardápio e variáveis



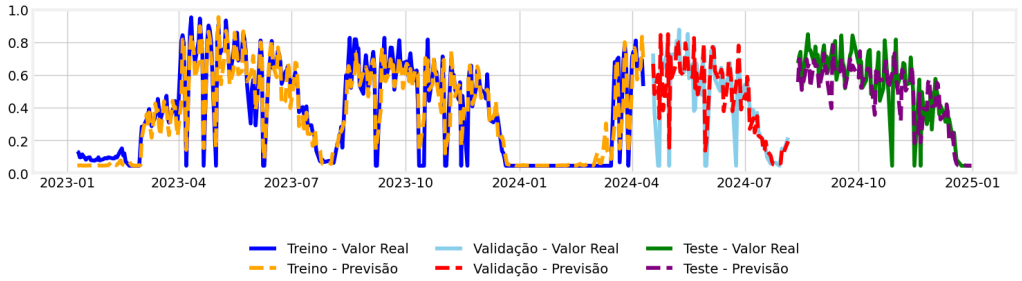
(a) Árvore de Decisão



(b) XGBoost



(c) LSTM



(d) LSTM com camada extra

Figura 2. Desempenho dos melhores modelos dos algoritmos.

cíclicas, mostrando que ambas as informações contribuíram para capturar padrões complexos na demanda.

No caso das redes LSTM, tanto o modelo padrão quanto a versão com camada extra tiveram melhor desempenho com inclusão de variáveis cíclicas e exclusão do cardápio, reforçando a importância da representação temporal para redes recorrentes. Ambos os modelos utilizaram 25 unidades, $learning\_rate = 0.05$ e $batch\_size = 16$, indicando uma estratégia de treinamento consistente entre as duas arquiteturas.

O melhor modelo para a previsão foi o XGBoost, que se destacou em relação aos demais algoritmos testados. Além do bom desempenho, ele oferece explicabilidade, permitindo compreender quais variáveis têm maior influência sobre as previsões. Isso aumenta a confiança dos desenvolvedores e facilita que gestores entendam o processo, tornando-o mais transparente.

A relevância das variáveis foi avaliada pela métrica de Importância da Covariável por Permutação [Breiman 2001], que verifica quanto o erro de predição aumenta quando os valores de uma variável são embaralhados. Quanto maior o impacto, mais relevante é a variável para o modelo.

A Figura 3 apresenta a visualização gráfica da importância das variáveis obtida por meio dessa técnica. É possível identificar claramente quais atributos mais impactam o desempenho do melhor modelo, fornecendo insights não apenas para análise interna dos desenvolvedores, mas também para auxiliar a gestão do restaurante na compreensão dos fatores que influenciam a demanda por refeições.

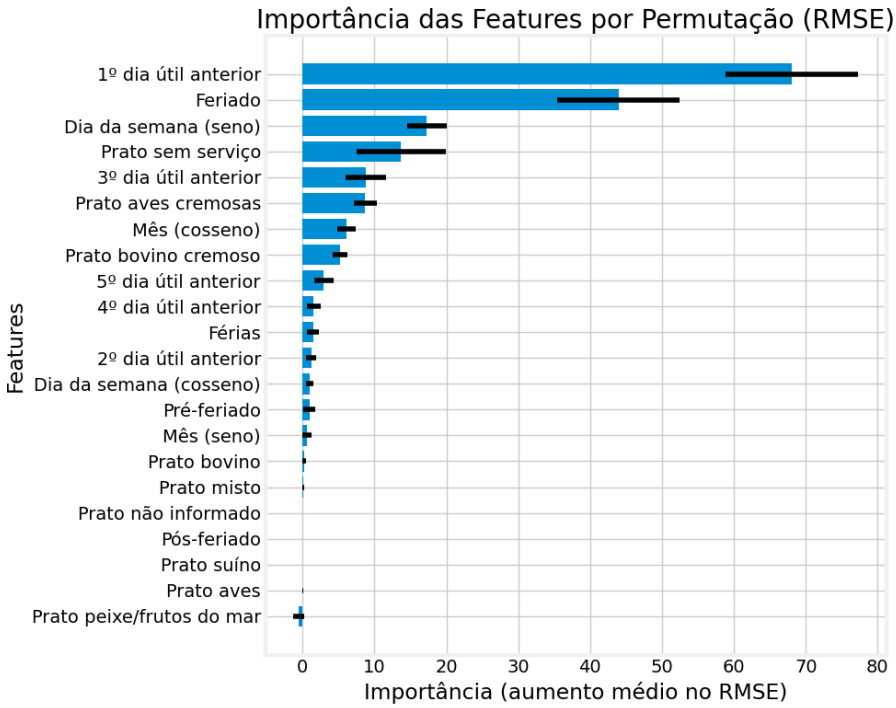


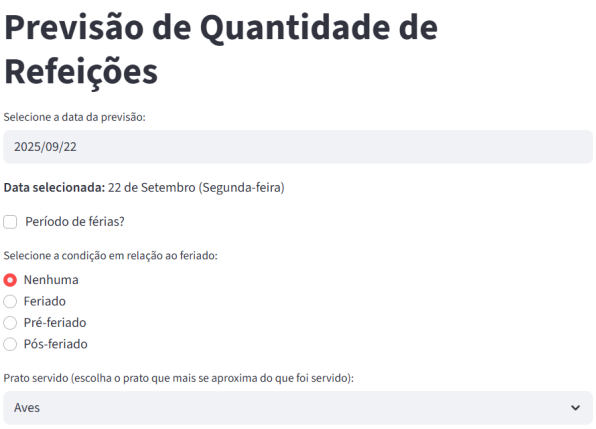
Figura 3. Importância das features por permutação (RMSE).

Por fim, na etapa de **Implantação**, para tornar o modelo acessível de forma prática, foi desenvolvido um aplicativo web utilizando a biblioteca Streamlit, que permite

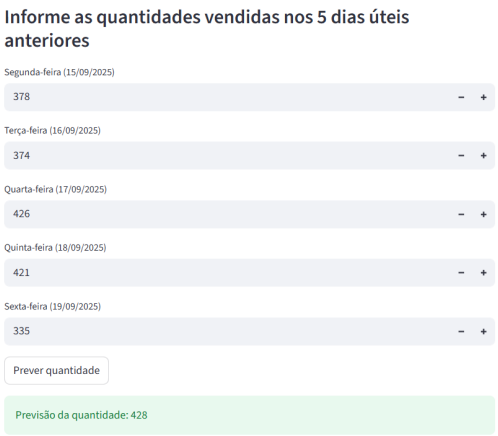
criar interfaces interativas a partir de scripts Python. O aplicativo principal, armazenado no arquivo `app.py`, contém toda a lógica da interface, incluindo campos para inserção de parâmetros relevantes, como feriado, prato do dia e quantidade de refeições anteriores, além do código que carrega o melhor modelo treinado e realiza a previsão com base nos dados fornecidos pelo usuário.

Todos os arquivos necessários para o funcionamento do aplicativo, incluindo o `app.py`, o arquivo do melhor modelo treinado e o `requirements.txt` com as bibliotecas utilizadas, foram armazenados em um repositório público no GitHub. A partir desse repositório, o aplicativo foi publicado no Streamlit Cloud, que provisiona automaticamente um ambiente de execução e disponibiliza o site em uma URL acessível, sem necessidade de instalações locais. Dessa forma, o restaurante pode acessar a ferramenta diretamente via navegador, inserir os parâmetros desejados e obter previsões de maneira rápida e intuitiva.

A Figura 4 apresenta a interface do aplicativo, destacando os campos de entrada e a exibição do resultado das previsões. Com essa abordagem, é possível não apenas melhorar a acurácia das previsões, mas também proporcionar maior transparência e compreensão sobre os fatores que influenciam a demanda de refeições.



(a) Corte 1 da interface (parte superior).



(b) Corte 2 da interface (parte inferior).

Figura 4. Interface da ferramenta de previsão de refeições.

5. Conclusão

Este estudo teve como objetivo desenvolver uma solução baseada em aprendizado de máquina para apoiar a previsão da demanda de refeições em um restaurante universitário, contribuindo para a otimização operacional e a redução do desperdício. Utilizando dados históricos de consumo, feriados e cardápios do restaurante do Centro Multidisciplinar da UFRJ em Macaé, foram avaliados diferentes algoritmos, dentre os quais o XGBoost apresentou o melhor desempenho, alcançando $R^2 = 0,92$ e superando a previsão originalmente utilizada pelo restaurante ($R^2 = 0,83$).

Os resultados demonstram que o modelo proporciona previsões mais precisas, permitindo alinhar a produção à demanda real e favorecendo o uso eficiente dos recursos.

Com um RMSE de 43,69, o XGBoost mostrou-se capaz de apoiar decisões que reduzem desperdícios, otimizam processos e fortalecem a sustentabilidade operacional. O modelo foi incorporado a uma ferramenta web desenvolvida em Streamlit, tornando todo o processo de previsão mais ágil, acessível e menos dependente de procedimentos manuais.

Para assegurar a efetividade da adoção da solução, recomenda-se, como trabalho futuro, a realização de uma avaliação de usabilidade com os gestores do restaurante, utilizando diretamente a ferramenta implementada. Essa análise permitirá verificar a clareza das informações, a facilidade de navegação e eventuais ajustes necessários para sua plena integração ao fluxo de trabalho.

Outro ponto importante é a definição de uma estratégia de atualização contínua do modelo, contemplando ingestão automatizada de novos dados, monitoramento do desempenho e adaptações perante mudanças no cardápio ou no perfil de consumo. A inclusão de variáveis contextuais, como eventos acadêmicos ou condições climáticas, também pode ampliar a robustez do sistema.

Nesse contexto de expansão, a aplicação da metodologia em outros restaurantes é possível, mas requer atenção a diferenças estruturais. A quantidade de cursos oferecidos, o número de alunos matriculados e a qualidade dos dados coletados são fatores críticos que influenciam diretamente o desempenho dos modelos. Assim, embora os resultados sejam promissores, a generalização para outros cenários exige testes complementares, ajustes específicos e validação contínua para assegurar a efetividade e a confiabilidade da solução.

Em síntese, o estudo evidencia que modelos de aprendizado de máquina, aliados a uma interface web prática e intuitiva, constituem uma ferramenta eficiente para previsão de demanda em restaurantes universitários, proporcionando benefícios operacionais, econômicos e ambientais e abrindo caminho para aplicações em contextos institucionais mais amplos.

Agradecimentos

Ao Restaurante Universitário da Universidade Federal do Rio de Janeiro do Centro Multidisciplinar de Macaé pelos dados para condução dessa pesquisa e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro por meio da bolsa de Iniciação Científica.

Referências

Bartolomei, B., Paula, M., and Souza, V. (2022). Identificação de demandas de dados para auxiliar o planejamento urbano municipal a partir de um estudo de caso. In *Anais Estendidos do XVIII Simpósio Brasileiro de Sistemas de Informação*, pages 285–293, Porto Alegre, RS, Brasil. SBC.

Breiman, L. (2001). Random forests. *Machine learning*, 45(1):5–32.

Chapman, P. (2000). *CRISP-DM 1.0: Step-by-step Data Mining Guide*. SPSS.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, page 785–794, New York, NY, USA. Association for Computing Machinery.

Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. "O'Reilly Media, Inc.".

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Kanwal, N., Zhang, M., Zeb, M., Batool, U., khan, I., and Rui, L. (2024). From plate to palate: Sustainable solutions for upcycling food waste in restaurants and catering. *Trends in Food Science Technology*, 152:104687.

ONU (2024). Objetivos de desenvolvimento sustentável (ods). `https://brasil.un.org/pt-br/sdgs`.

Pereira, H. A., Felix, L. B., Cordeiro, L. L., and Antunes, H. M. A. (2007). Predição do número de refeições utilizando redes neurais artificiais. In *Anais do 8 Congresso Brasileiro de Redes Neurais*, pages 1–5, Florianópolis, SC. SBRN.

Pereira, L., Ferreira, L., Nascimento, D., Vanderlei, I., Silva, D., Santos, V., Gomes, M., and Melo, I. (2025). Design, integration, and evaluation of a demand forecasting service in the context of primary healthcare. In *Anais do XXI Simpósio Brasileiro de Sistemas de Informação*, pages 506–514, Porto Alegre, RS, Brasil. SBC.

Rocha, J. C., Matos, F. D., and Frei, F. (2011). Utilização de redes neurais artificiais para a determinação do número de refeições diárias de um restaurante universitário. *Revista de Nutrição*, 24(5):735–742.

Rodrigues, M., Miguéis, V., Freitas, S., and Machado, T. (2024). Machine learning models for short-term demand forecasting in food catering services: A solution to reduce food waste. *Journal of Cleaner Production*, 435:140265.

Santos, Y. D. C. d. F., Saqui, D., and Dos Santos, P. C. (2021). Um método baseado em aprendizado de máquina para previsão da produção de refeições em restaurantes universitários: A method based on machine learning to predict meal production in university restaurants. In *Proceedings of the XVII Brazilian Symposium on Information Systems*, SBSI '21, New York, NY, USA. Association for Computing Machinery.

United Nations Deputy Secretary-General (2024). School Meals Provide 'Powerful, Cost-effective Way' to Drive Progress across Multiple Sustainable Development Goals, Deputy Secretary-General Tells Event. `https://press.un.org/en/2024/dsgsm1920.doc.htm`.

Yang, H., Jiao, W., Zouyi, L., Diao, H., and Xia, S. (2025). Artificial intelligence in the food industry: innovations and applications. *Discover Artificial Intelligence*, 5(1):60.