

Predição de Tempo de Entrega em Redes de Delivery: Uma Abordagem com XGBoost

Ewerthon José Kutz¹

¹ Departamento de Computação – Universidade Estadual de Londrina (UEL)
Caixa Postal 10.011 – CEP 86057-970 – Londrina– PR – Brasil

`ewerthon.jose.kutz@uel.br`

Abstract. *This work presents the development of a machine learning model for predicting delivery times in a food delivery network using historical data. The XGBoost algorithm was chosen and optimized through hyperparameter tuning, achieving a Mean Absolute Error (MAE) of 9.5 minutes on the test set, representing a 25.8% improvement over the baseline model (12.8 minutes). The solution was developed following the CRISP-DM methodology and deployed as a web application on Google Cloud Platform using Streamlit.*

Resumo. *Este trabalho apresenta o desenvolvimento de um modelo de aprendizado de máquina para predição de tempos de entrega em uma rede de delivery de alimentos utilizando dados históricos. O algoritmo XGBoost foi escolhido e otimizado através de ajuste de hiperparâmetros, alcançando um Erro Absoluto Médio (MAE) de 9,5 minutos no conjunto de teste, representando uma melhoria de 25,8% em relação ao modelo baseline (12,8 minutos). A solução foi desenvolvida seguindo a metodologia CRISP-DM e implantada como aplicação web na Google Cloud Platform utilizando Streamlit.*

1. Introdução

Este trabalho aborda o problema de predição de tempos de entrega em uma rede de delivery de alimentos. Atualmente, o negócio utiliza a média histórica dos tempos de entrega como estimativa, abordagem que não considera a variabilidade de fatores como características da loja, horários e distância até o cliente. O dataset utilizado foi extraído da plataforma iFood e contém 19.285 pedidos de 90 dias com features relacionadas a: loja, data e hora, serviço logístico, distância e valores monetários. A solução proposta é desenvolver um modelo de regressão supervisionada usando XGBoost como alternativa à abordagem atual.

O objetivo é construir um modelo preditivo que considere múltiplas variáveis do contexto do pedido, resultando em estimativas mais precisas e confiáveis. O dataset contém 19.285 pedidos realizados entre 09/08/2025 e 07/11/2025, totalizando 15 features após pré-processamento e uma variável alvo (tempo de entrega em minutos).

O projeto seguiu a metodologia CRISP-DM (*Cross-Industry Standard Process for Data Mining*), abrangendo seis fases: (1) **Business Understanding** para definição do problema e regras de negócio; (2) **Data Understanding** com análise exploratória e identificação de padrões; (3) **Data Preparation** incluindo limpeza, transformações e feature engineering; (4) **Modeling** com desenvolvimento de modelos baseline e otimizados;

(5) **Evaluation** através de métricas, validação cruzada e análise de curvas de aprendizado; e (6) **Deployment** com implantação da solução como aplicação web na Google Cloud Platform.

2. Solução

2.1. Business Understanding e Data Understanding

O problema consiste em prever o tempo de entrega de pedidos em uma rede de restaurantes de delivery. Foram identificadas regras de negócio importantes: (1) colunas de tempos parciais da entrega não estão disponíveis no momento do pedido; (2) pedidos cancelados devem ser desconsiderados; (3) a coluna de tempo prometido não deve ser utilizada como feature; (4) fatores relevantes ausentes incluem condições meteorológicas, demanda da cozinha, disponibilidade de entregadores e tipo de alimento.

Por meio da Análise Exploratória de Dados no dataset com 19.285 pedidos com 15 features e 1 target, indentificou-se distribuição aproximadamente normal no target (média=37min e desvio padrão=18min). A análise de correlações revelou baixo poder preditivo geral: maior correlação foi *distancia_percorrida_ate_o_cliente_km*=0,4, seguida por *nome_da_loja*=0,3. O PPS (*Predictive Power Score*) indicou apenas duas features com poder preditivo relevante: *taxa_de_entrega_paga_pelo_cliente* (4,1%) e *servico_logistico* (0,3%). Features temporais e características da loja foram identificadas como potencialmente relevantes através de visualização com árvore de decisão.

2.2. Data Preparation, Modeling e Evaluation

Pré-processamento: Foi construído um pipeline com as transformações apresentadas na Tabela 1.

Tabela 1. Pipeline de pré-processamento

Transformação	Variáveis
Ordinal Encoding	turno, prioridade_do_pedido
Count Frequency Encoding	marca_da_loja, nome_da_loja
One-Hot Encoding	servico_logistico
DropFeatures	data_e_hora_do_pedido

XGBoostRegressor foi escolhido por sua eficácia em capturar relações não-lineares através de *gradient boosting*, capacidade de lidar com features mistas (categóricas, numéricas, temporais), robustez a outliers e eficiência computacional. Um Dummy Regressor (estratégia de média) foi implementado como baseline.

Foram treinados três modelos: (1) Dummy Regressor (baseline), (2) XGBoost inicial (*n_estimators*=100), e (3) XGBoost otimizado. A otimização utilizou Optuna com 30 trials e validação cruzada de 3 folds, buscando minimizar MAE. A Tabela 2 apresenta os hiperparâmetros otimizados.

Tabela 2. Hiperparâmetros otimizados do XGBoost

Hiperparâmetro	Faixa	Valor / Objetivo
n_estimators	[100, 2000]	785 / Complexidade
learning_rate	[0.001, 0.1]	0.012 / Evitar overfitting
max_depth	[1, 10]	8 / Balancear Evitar overfitting
subsample	[0.05, 1.0]	0.82 / Evitar overfitting
colsample_bytree	[0.05, 1.0]	0.76 / Evitar overfitting
min_child_weight	[1, 20]	18 / Regularização
gamma	[0, 5.0]	4.42 / Regularização
alpha (L1)	[0, 10.0]	2.38 / Regularização
lambda (L2)	[1, 10.0]	9.82 / Regularização

Os modelos foram avaliados com MAE (métrica primária) e RMSE (secundária) através de 3-fold cross-validation nos conjuntos de treino/validação, e teste final em hold-out de 20%. Curvas de aprendizado e gráficos de resíduos foram utilizados para detectar overfitting/underfitting. As Figuras 1 e 2 mostram os resultados para o modelo otimizado.

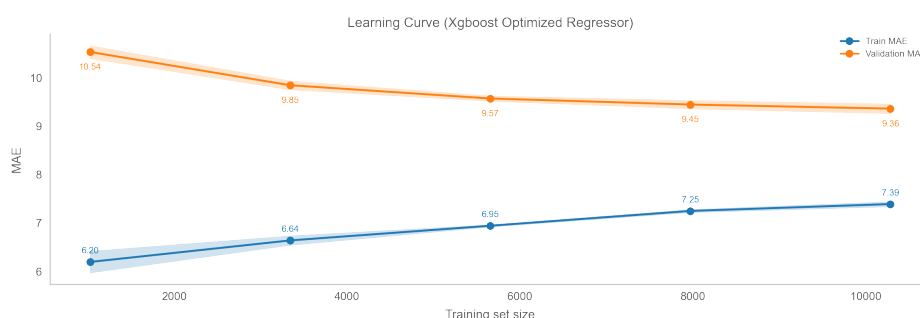


Figura 1. Curva de aprendizado do XGBoost otimizado

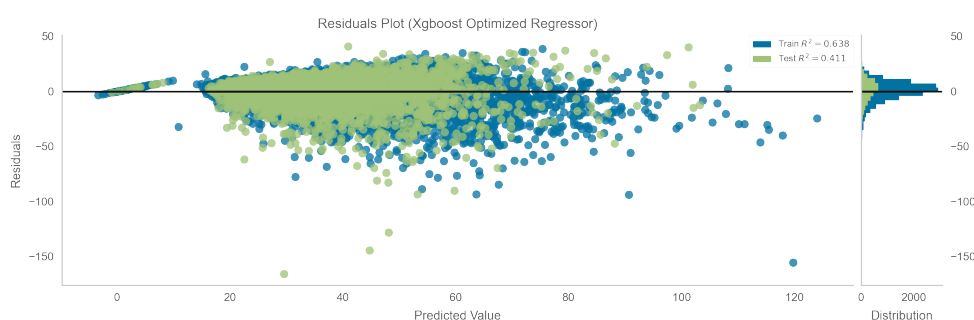


Figura 2. Gráfico de resíduos do XGBoost otimizado

2.3. Deployment

A solução foi implantada como aplicação web interativa utilizando Streamlit, permitindo visualização de métricas e realização de previsões em tempo real. A arquitetura de deployment inclui: (1) containerização com Docker; (2) build automatizado via Google

Cloud Build; (3) armazenamento de imagens no Artifact Registry; e (4) deployment no Google Cloud Run (região southamerica-east1). A aplicação está acessível publicamente e permite aos usuários interagir com todas as etapas do CRISP-DM através de interface intuitiva.

3. Resultados e Discussão

3.1. Desempenho dos Modelos

Os modelos foram avaliados com MAE (métrica primária) e RMSE (secundária) através de 3-fold CV:

Tabela 3. Desempenho comparativo dos modelos

Modelo	MAE Treino	MAE Val	MAE Teste
Dummy	12,9 ± 0,1	12,9 ± 0,1	12,8
XGBoost Inicial	5,7 ± 0,1	9,8 ± 0,1	9,8
XGBoost Otimizado	7,4 ± 0,1	9,4 ± 0,1	9,5

O XGBoost otimizado alcançou MAE de 9,5 minutos (redução de 25,8% sobre baseline), representando 25,7% do tempo médio de entrega (37 min). A otimização reduziu overfitting: discrepância treino-validação caiu de 4,1 para 2,0 minutos. Curvas de aprendizado indicam que mais dados podem melhorar o desempenho.

3.2. Limitações e Melhorias

As principais limitações identificadas foram: (1) Baixo poder preditivo das features: maior correlação foi *distancia_percorrida*=0,4; PPS máximo de 4,1% (*taxa_de_entrega*); (2) Features ausentes identificadas com negócio: meteorologia, demanda da cozinha, disponibilidade de entregadores, tipo de alimento; (3) Dataset limitado a 90 dias com curvas de aprendizado indicam ganho potencial com mais amostras.

Por isso, as melhorias propostas são: (1) Integração com APIs meteorológicas e dados operacionais do negócio; (2) Modelagem segmentada por restaurante (curva ABC por volume); (3) Ampliação temporal do dataset.

4. Conclusões

Este trabalho demonstrou a viabilidade de substituir média histórica por machine learning para predição de tempos de entrega. O XGBoost otimizado alcançou MAE de 9,5 minutos (redução de 25,8% sobre baseline), com erro de 25,7% do tempo médio.

A metodologia CRISP-DM permitiu desenvolvimento estruturado até implantação web na Google Cloud Platform. A otimização com Optuna produziu configuração que balanceia capacidade e generalização.

As limitações principais – ausência de features meteorológicas e operacionais – indicam que melhorias dependem mais de enriquecimento de dados que de refinamento algorítmico. Recomenda-se a incorporação de dados meteorológicos/operacionais e desenvolvimento de modelos segmentados por restaurante.