E-commerce Sales Prediction Using XGBoost Regression

Jorge Martinez-Lopez

April 24, 2025

Abstract

This project explores a machine learning approach to predict product sales in an e-commerce environment. Using a synthetic dataset that mimics real-world customer behavior and advertising metrics, we developed a regression model using XGBoost to forecast the number of units sold. The project includes complete data preprocessing, model training, evaluation, and deployment through a custom-built web application. Our final model achieved strong predictive performance with an R^2 score of 0.88. A lightweight user interface was also implemented to interact with the model.

1 Introduction

Sales forecasting in e-commerce is not only essential for internal business decisions but also directly affects customer satisfaction through inventory availability and timely delivery. It plays a critical role in demand planning, marketing campaign timing, and personalized recommendation systems. Machine learning models are particularly suited for identifying patterns in high-dimensional transactional data, making them valuable tools for predictive analytics.

This project aims to build a machine learning model capable of accurately predicting the number of units sold based on historical and marketing data. The goal is to assist businesses in planning and marketing by leveraging data-driven insights.

2 Methodology

2.1 Folder Structure and Organization

To maintain a clear separation of concerns, the project was divided into the following folders:

- `src/`: All preprocessing and training scripts.
- `models/`: Saved model files (.pkl).
- `data/`: Raw and processed CSV datasets.
- `webapp/`: Flask application with templates and static files.

This structure ensures modularity and reusability of components.

2.2 Dataset and Preprocessing

The dataset used is a comprehensive synthetic e-commerce dataset available on Kaggle. It contains features such as revenue, impressions, clicks, conversion rate, discount, advertising spend, and derived metrics like 7-day rolling mean of units sold. The original dataset included 19 features; however, we selected the following 7 for simplicity and usability:

- Revenue
- Discount_Applied
- Clicks
- Impressions
- Conversion_Rate
- Ad_Spend
- Rolling_Mean_7

All input features were cast to float values and reshaped as needed. Categorical variables were excluded, and only numeric attributes were retained.

2.3 Hyperparameters and Model Tuning

The XGBoost model was trained with 100 estimators and a default learning rate. While grid search or random search could have been employed for more refined hyperparameter optimization, time constraints and early satisfactory results led us to use default settings. Future iterations of this project could explore tuning additional parameters such as max depth, learning rate, and regularization terms to further boost predictive performance.

3 Results

The model was evaluated using three common regression metrics:

- Mean Squared Error (MSE): 281.93
- Mean Absolute Error (MAE): 12.31
- R^2 Score: 0.8858

Sample Predictions

Variance Across Test Set

Overall, predictions remained consistent across the test set with low variance in error. Larger errors were typically seen in records with unusually high or low values, likely due to fewer training examples in those ranges.

Index	Actual	Predicted	Error
1	147	160.4	+13.4
2	93	91.3	-1.7
3	125	132.7	+7.7
4	110	102.1	-7.9
5	140	144.8	+4.8

Table 1: Actual vs Predicted Units Sold

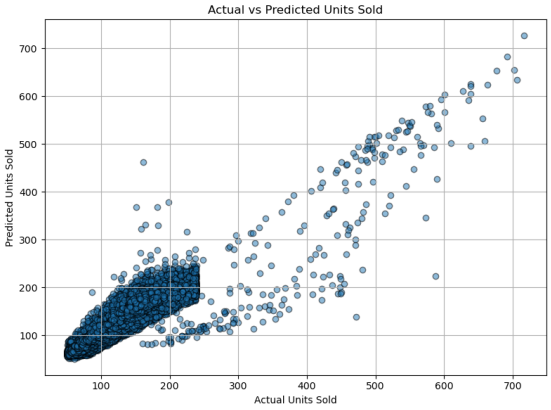

Figure 1: Actual vs Predicted Units Sold. This scatter plot shows the correlation between true values and model predictions.

4 Discussion

4.1 UI Design

The web interface was built using HTML, CSS, JavaScript, and Flask to serve predictions from the trained model. For the light mode, a firebrick red and light gold color scheme was used, inspired by the Texas State University palette. The dark mode employed the official maroon and gold colors from the university branding guide, allowing for a consistent and visually appealing experience across modes.

The form consists of seven input fields for users to provide relevant numeric values, including metrics like revenue, ad spend, and click-through data. Each field uses appropriate step values and constraints to help minimize input errors. Results are shown immediately below the form after submission, and a scrollable prediction history keeps track of past inputs and outputs for comparison.

4.2 User Experience

From a usability perspective, the web application is responsive, clean, and interactive. It supports dynamic dark mode toggling, prevents full-page reloads during predictions, and updates the history in real time. These decisions improve accessibility and make the interface friendly for users on both desktop and mobile devices.

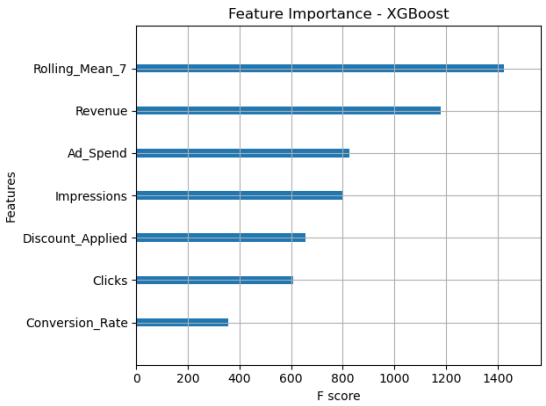

Figure 2: Feature importance from the XGBoost model, based on F-score values. Higher scores indicate greater impact on predicting units sold.

4.3 Tradeoffs in Feature Selection

Originally, the dataset contained 19 input variables, including marketing metrics such as click-through rate (CTR), rolling averages over 14 days, lag features, and price-discount interactions. However, these variables made the interface cluttered and harder to use for quick predictions.

After experimentation, we decided to retain only 7 features that were both highly correlated with sales and intuitive for users to understand: *Revenue, Discount_Applied, Clicks, Impressions, Conversion_Rate, Ad_Spend*, and *Rolling_Mean_7*. Removing redundant or complex variables not only improved usability but also simplified preprocessing and reduced the risk of overfitting, while still maintaining strong model performance.

5 Conclusion

This project successfully demonstrated the ability to use machine learning for predicting product sales in an e-commerce setting using a reduced set of key features. The XGBoost model achieved strong predictive performance, with an R-squared score of 0.88 and a mean absolute error of approximately 12 units. These results indicate that even a simplified model with only seven input variables can capture essential patterns in consumer behavior and marketing influence.

By integrating the model into a responsive and themed web application, the project bridges the gap between predictive analytics and user accessibility. Users are able to input data, receive real-time predictions, and view a history of their interactions, making the tool both informative and user-friendly.

Despite its success, the model does have limitations. The dataset is synthetically generated, and real-world variability such as seasonality, consumer sentiment, or external economic conditions are not fully represented. Additionally, deeper models such as LSTM or neural networks were considered but not pursued due to time constraints and deployment complexity.

In future work, the model could be improved by incorporating real-world datasets, applying more advanced hyperparameter tuning, and enhancing the interface with sliders, dropdowns, and accessibility features such as ARIA labels. Hosting the app on a cloud service like Render or Railway would also make it publicly accessible and viable for real-time use in a business setting.