رواد مصر الرقمية

وزارة الاتصالات وتكنولوجيا المعلومات

Digital Egypt Pioneers Initiative (DEPI) Final Project

MLOps Report: Milestone (4)

Sales Forecasting and Optimization

Group Members:

Basmala Ehab Mohamed Yousry

Mostafa Mahmoud Mohamed Elshahat

Mohab Mohamed Ibrahim Mohamed

Ziad Ahmed Gharieb

SHR2_AIS4_S2

Submission Date:
9 May 2025

YOUth co.

Table of Contents

Introduction

In this milestone, we implemented MLOps practices for tracking, deploying, and monitoring the sales forecasting model. The goal was to ensure the model's deployment is efficient and scalable, provide real-time or batch predictions, and set up continuous monitoring and feedback mechanisms to improve the model over time.

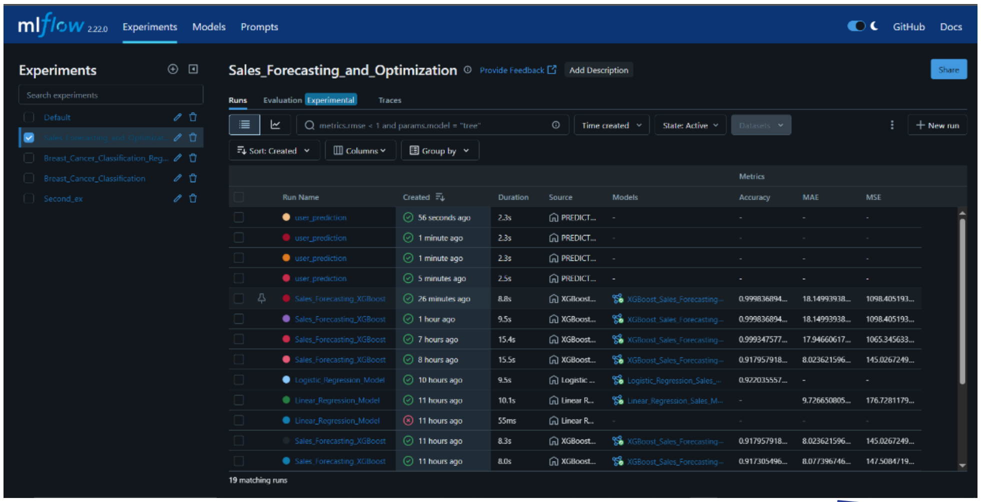
MLOps Implementation

a. MLflow for Experiment Tracking and Model Management

- **Experiment Tracking:** We used MLflow as the primary tool for experiment tracking, model management, and logging metrics and parameters. MLflow allows us to maintain detailed records of each experiment, including the parameters, metrics, and versions of models used, providing a clear history of the model's performance

- **Model Versioning:** Through MLflow, we set up version control for our forecasting models, enabling us to track changes and update the model with new data. This ensures that any improvements in the model or its predictions can be easily tracked and managed.

- **Logging Metrics and Parameters:** All metrics (e.g., accuracy) and model parameters were logged in MLflow during training. This provides a comprehensive overview of the model's training process and results, enabling better comparison between experiments.



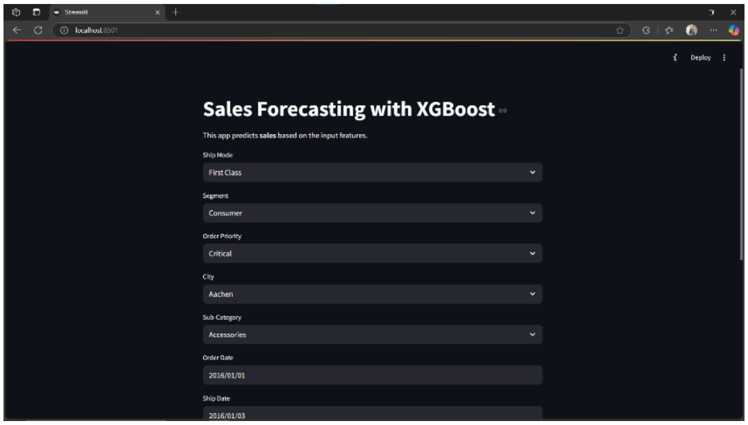
b. MLflow Setup for Model Deployment

- We set up **MLflow** to track all model deployments and to serve the forecasting model directly for real-time predictions via Streamlit.

Deployment

a. Streamlit Web App for Real-Time Predictions

- The model was deployed using **Streamlit**, which provides an interactive interface for users to input features and get real-time predictions. The app allows users to select inputs like order date, ship date, and other features related to the sales transaction.



- The model serves predictions instantly after the user inputs their data, providing an intuitive and fast user experience.

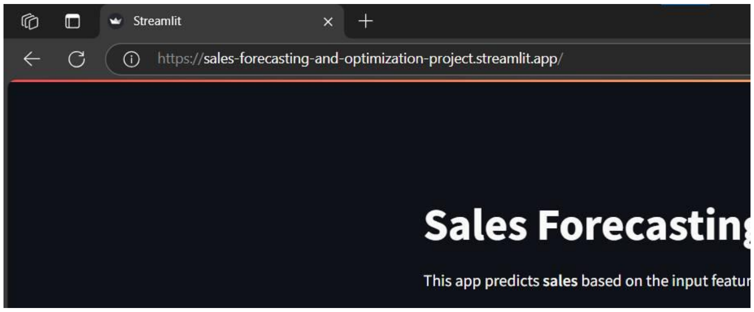
b. Real-Time vs. Batch Predictions

- The deployed model is capable of handling real-time predictions where users input data and get instant sales forecasts.
- We also have the capability to handle batch predictions, where a series of data points can be processed to generate forecasts for multiple entries simultaneously.

c. Deployment via GitHub and Streamlit

- The app is stored and managed using **GitHub**, which ensures version control and collaboration. GitHub also acts as a centralized repository for the app's code and assets.

- For deployment, **Streamlit** is used to host the web app, making it publicly accessible. Streamlit's platform provides a fast, easy way to deploy and share Python-based apps with minimal configuration.



https://sales-forecasting-and-optimization-project.streamlit.app/

Model Monitoring

a. Performance Tracking Over Time

- **Model Drift Monitoring:** We integrated **MLflow** and set up alerts to monitor the model's performance over time. This allows us to detect any degradation in prediction accuracy (i.e., model drift). Alerts are triggered if prediction accuracy falls below a set threshold.

- **Feedback Loop for Continuous Improvement:** We implemented a feedback loop where actual sales values are collected after predictions and compared with the forecasted values. This data is then used to log errors (e.g., MAE, RMSE) to MLflow for monitoring and future improvements.

b. Logging Actual vs Predicted Values

Users are prompted to input actual sales values after the forecast is made. This data is logged back into MLflow, allowing us to track model performance and adjust the model if needed.

Performance Reporting

- To ensure the model performs as expected, we log key performance metrics such as absolute error, mean absolute error (MAE), and root mean squared error (RMSE) to MLflow.

- An alert system has been set up to notify stakeholders (e.g., via email) if the model's accuracy drops below a defined threshold, allowing for timely intervention and retraining of the model.