



Digital Egypt Pioneers Initiative (DEPI) Final Project

FINAL REPORT

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



Introduction

This project, titled "Sales Forecasting and Optimization", aims to leverage historical sales data to accurately predict future sales. Accurate forecasting is critical for businesses to make informed decisions regarding inventory management, promotional planning, and overall strategic direction. By identifying patterns and trends in past performance, businesses can proactively adjust to changes in demand and enhance customer satisfaction.

In Milestone 1, we focused on data collection, exploration, and cleaning. This involved evaluating data quality, identifying trends and seasonal effects, examining relationships between variables, and addressing issues such as missing values and outliers. These steps were essential to prepare the data for robust modeling.

A core challenge in this project was to analyze the data in a way that supports decision-making. By predicting future sales and calculating total expected revenue, we aim to provide insights into whether the business trajectory is upward or downward—helping stakeholders set realistic goals and strategies.

To model sales forecasting, we explored a mix of statistical and machine learning techniques, including:

- ARIMA and SARIMA for capturing linear and seasonal trends,
- XGBoost for modeling complex nonlinear patterns,
- Prophet for handling multiple seasonalities and holiday effects,
- Linear Regression as a baseline predictor,
- Logistic Regression for exploratory binary classification of demand levels.

In the final stages, we applied MLOps practices to support model deployment, tracking, and monitoring. This ensures that our forecasting system is scalable and can adapt to real-time or batch predictions, with continuous performance feedback loops to improve the model over time.

Data Overview

- **Dataset:** Daily sales data from 2010 to 2016 for a large store, with over 50,000 records and 24 columns.
- **Important Columns:**
 - *Date* (when the sale happened)
 - *Sales, Quantity, Discount, Profit* (numbers)
 - *Category, Sub-Category, Region, Segment* (labels)
- **Size & Types:** About 51,000 rows. Mix of dates, numbers, and object labels.
- **Variety:** 15+ sub-categories, 23 regions, many products and customers.

Missing Values and Data Quality

- **Missing Data:** Overall, less than 3.35% of values are missing, except in the Postal Code column, where approximately 80% of entries are missing.
- **Fixing Missing Data:**
 - That column wasn't very important, and since most of its values were missing, we dropped it entirely.
- **Duplicates & Format:**
 - There are no duplicate transaction IDs.
 - All dates are formatted as YYYY-MM-DD.
 - Profit values range from negative to positive.

Trend Analysis

- **Overall Trend:** Sales rose from 2010 to mid-2015, then stayed flat in 2016.
- **By Category:**
 - *Technology* grew fastest (~12% per year).
 - *Office Supplies* grew ~5%–7% per year.
- **Top Products:** Phones, Copiers, Chairs, Bookcases, Storage items leading in sales.
- **Charts:**
 - *The monthly sales line plot shows steady growth.*
 - *Bar plot shows the top 10 sub-categories by total sales.*

Seasonality Analysis

- **Weekly Pattern:** Highest sales on Fridays and Sundays; lowest on Mondays.
- **Monthly/Quarterly Pattern:**
 - *Q4 (Oct–Dec) has the biggest peaks (holidays, Black Friday).*
 - *Q2 (Apr–Jun) has moderate boosts (spring sales).*
- **Holiday Effects:** Black Friday boost sales by 30%+. Other holidays vary by region.

Correlation and Influencing Factors

- **Key Correlations:**
 - *Sales and shipping cost:* strong positive (high sales → high profit).
 - *Profit and Sales:* moderate positive (more discount → more sales).
 - *Discount and Profit:* negative (more discount → less profit margin).
- **By Category:**
 - Technology sees a 50% sales lift during promotions.
 - Furniture profit holds up better under discounts.

Outliers and Anomalies

- **Finding Outliers:** Used box plots and IQR on sales, quantity, discount, profit, and shipping cost.
- **Results:** About 14.35% of records are extreme (very large single orders).
- **Handling Outliers:**
 - *Remove records beyond $3 \times IQR$ to avoid skewing models.*
 - *Keep mild outliers but mark them for special handling.*
 - *After removal, the dataset has ~ 20,411 fewer records*

Key Findings and Next Steps

- **Data Quality:** Data is mostly clean and complete.

- **Trends & Seasons:** Clear growth trend up to 2015 and strong seasonal patterns.
- **Opportunities:** Technology and Office Supplies grow fast; Furniture could use more targeted promotions.
- **Modeling Tips:**
 - Include holidays\weekends and Black Friday flags in models.
 - Create features like day/week/month markers.

Data Cleaning

As we learned in milestone1, there are problems in the data. It is time to learn how to process the data in the right way that serves our results and our work.

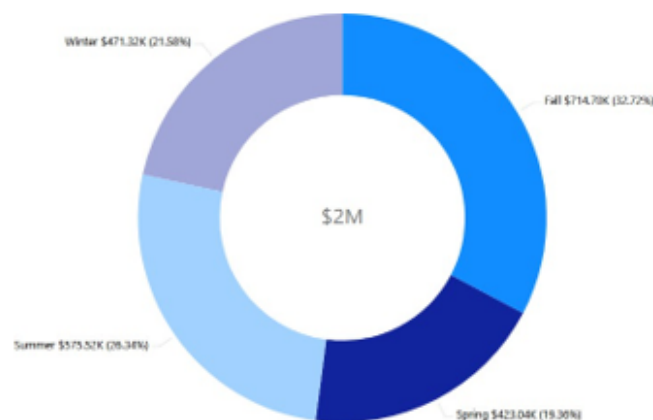
Steps we did follow:

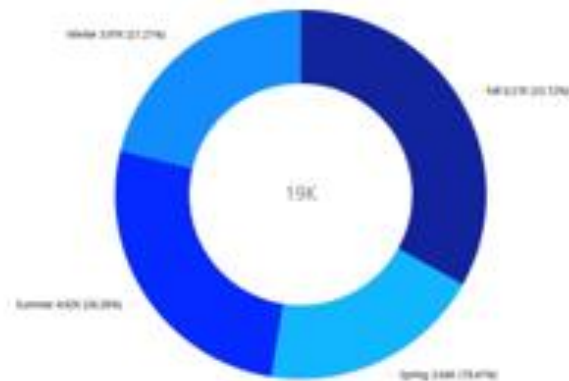
- **Handle Missing Values:**
 - Removed column called Postel code from data because more than 80% of data is null.
- **Duplicated Rows:**
 - We found that all rows do not have duplicated.
- **Removing unimportant features:**
 - We removed these features: Row ID – Postel Code – Customer Name – Product Name because we realized that not important in our Analysis
- **Extracting New Features:**
 - Extracting new columns called [Order Day – Order Month - Order Year – Order Month Name - Order Day Name] from existing column called Order Date
 - Creating column called Order Season from Order Month column by made function to did that
 - Creating column called Is_Black_Friday from Order Date to know if orders ordered on Black Friday or not
 - Creating a column called Ship_Days from column Order_Date and Ship_date to know how many days it will take to order reach to customer

- Handling Outliers:
 - Remove outlier data and became from 51290 row to 30652
More than 20638 rows removed
- Data consistency:
 - We checked that no data is reflected have spelling

Data Analysis

- statistical analysis:
 - We detected that **positive** relationship between Sales & Profit equal **0.48** mean Increased sales lead to increased profit.
 - Sales & Shipping Cost: **Strong positive** relationship **0.77**, meaning higher sales increase shipping costs.
 - Quantity & Sales: **Moderate positive** relationship **0.31**, large quantity increases sales slightly
 - Discount & Profit: **Moderately negative** relationship **-0.32**, large discount reduces profit.
 - Profit & Shipping Cost: **Moderate positive** relationship **0.35**, higher shipping costs are associated with higher profit.
- Seasonality Analysis:
 - We get Total sales & Orders in **Fall season** is a highest percentage equal **\$714.70k (32.72%)** after that second is **Summer \$575.52k (26.34%)** after that coming **Winter** and **Spring**

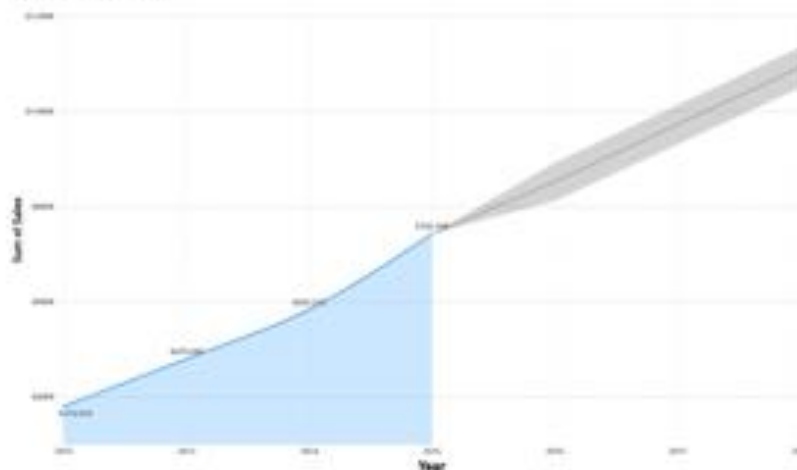




Data Visualization

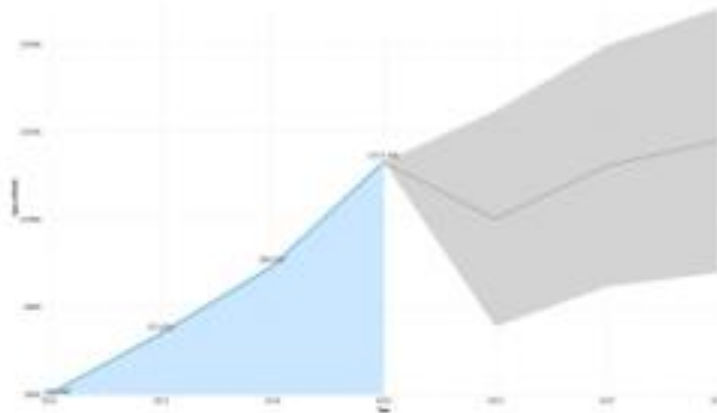
- Analysis of Sales & Profit Overtime:**

- Total sales increase year by year we see that in 2012 total sales is \$379.91k , 2013 is \$479.20k , in 2014 equal \$583.31k and in 2015 equal \$742.16k and so on. We predict that in 2018 total sales will reach \$1,134,689M

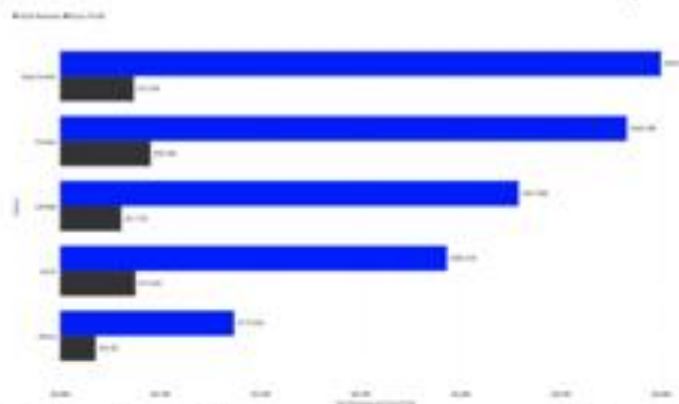




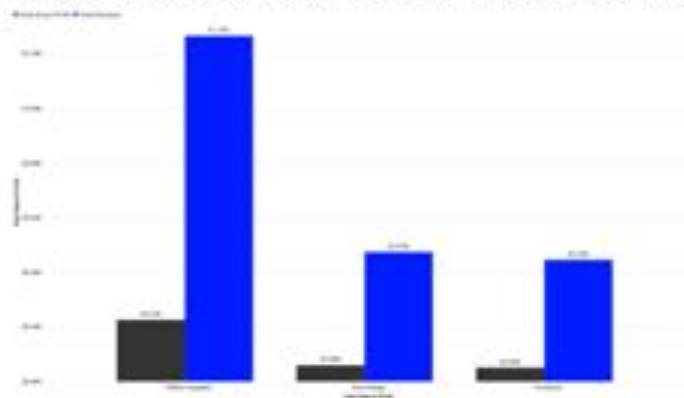
- Total Profit year by year profit increase a little bit profit from 2012 to 2015
Increase at a \$73k almost and we predict that in 2018 profit will increase to be \$148,263k



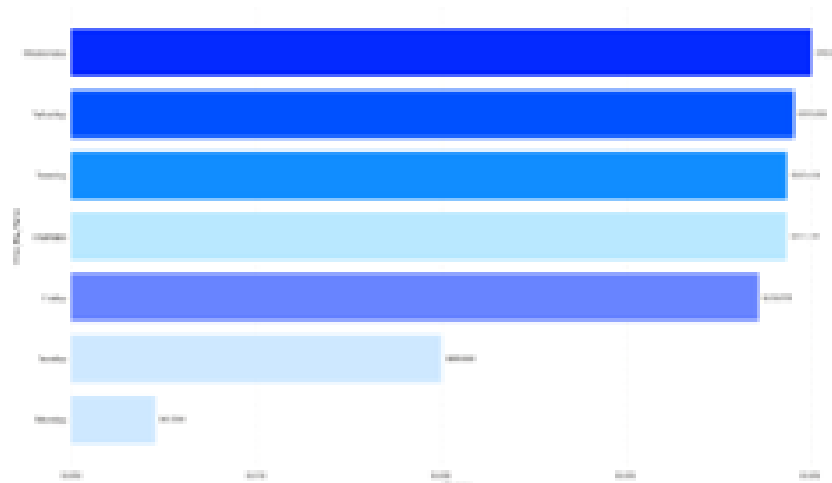
- Here we get that Market Asia is the highest market inside Sales = \$599.99k but is not the highest in profit.
And the lowest market is Africa in both sales and profit



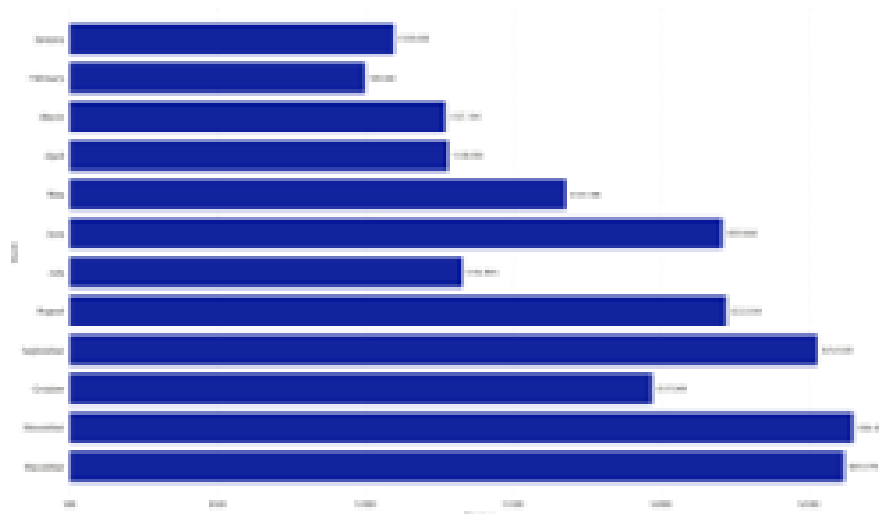
- Sales and profit by categories the category Office Supplies is achieve sales to equal \$1.27M and Profit \$23k is the highest category by the way
And the lowest category inside sales & profit we will get Furniture



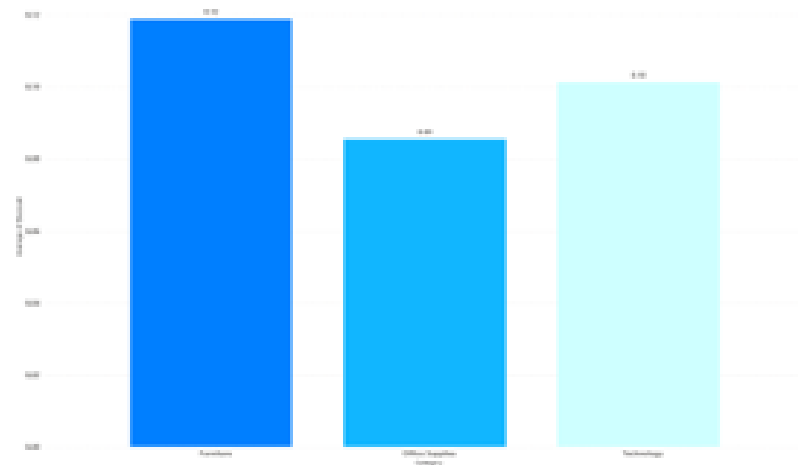
- Now we get the best day that sales is much we get Wednesday & Saturday is the high between \$390k to \$400k & Sunday, Monday is the lowest level between \$45k to \$120k



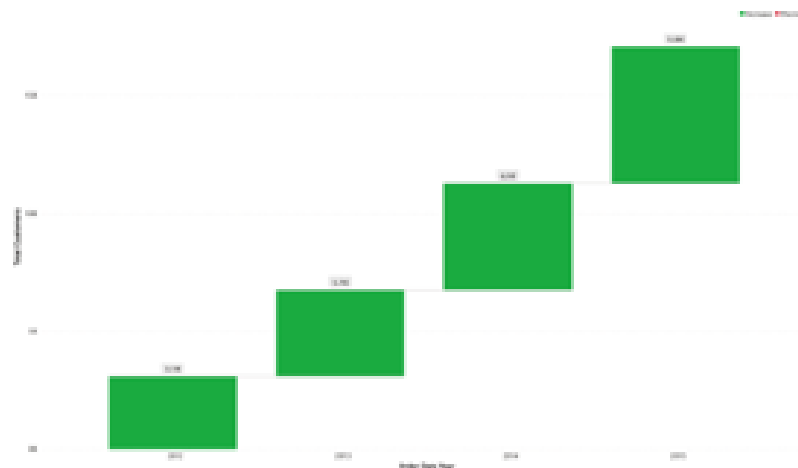
- Last not least we get the best month that sales is high in November & December sales are between \$260k to \$264k
And the bad months are January & February are between \$99k to \$109k



- We know the main factor in making a profit & sales is not good is the high discount. The average discount between all categories is between 9% to 12%



- The last thing we want to discuss is over year the total number of customers is high we will get in 2012 number of customer is 3.1k and in 2015 the total is 5.8k



Key Findings and Insights

- Sales:**
There has been a significant increase in sales volume over the years, especially in 2015, and it is expected that in 2018 sales will exceed 1 million.
- Profit:**
The profit margin increases by a very small percentage due to:

- The extremely high volume of discounts
 - And the very high shipping cost.
- **Market:**

The sales rate in the Asian and European markets is good, even the Latin American market is somewhat acceptable, but the African market is very low in terms of sales because of the poor countries' expectation of the proposal to make a relative reduction to increase profits there.
- **Increase number of customers:**

Customer growth rate is a very good indicator over the years.
- **Sales in Months & Days:**

It is suggested to make promotions on days when there are not many sales to attract customers to buy.

Model Selection

For our sales forecasting task, we explored both statistical and machine learning models to capture various temporal and nonlinear patterns in our data. The models selected were:

- **ARIMA (AutoRegressive Integrated Moving Average):** To capture linear trends and seasonality in time series.
- **SARIMA (Seasonal ARIMA):** Extended ARIMA with seasonal components to handle periodic patterns.
- **XGBoost (Extreme Gradient Boosting):** A powerful ensemble model used to capture nonlinear interactions and complex patterns in sales data.
- **Prophet (by Facebook):** A time series forecasting model suitable for daily data with multiple seasonalities and holiday effects.
- **Linear Regression:** As a baseline machine learning approach to model the relationship between time and sales.

Logistic Regression: Though typically used for classification, it was tested for binary sales outcomes (e.g., high vs. low demand) as an exploratory approach.

Model Training

- Each model was trained using historical sales data, which was split into training and validation sets. The time-series models (ARIMA, SARIMA, Prophet) used date-based indexing, while machine learning models (XGBoost, Linear, and Logistic Regression) were trained using engineered features including lagged sales, day-of-week indicators, and promotional flags.

Model Evaluation and Tuning

Models were evaluated based on standard forecasting metrics including:

- MAE (Mean Absolute Error)**
- RMSE (Root Mean Squared Error)**

Hyperparameter tuning was performed as follows:

- ARIMA/SARIMA:** Parameters (p, d, q) and (P, D, Q, s) were selected using AIC/BIC scores and grid search.
- XGBoost:** Grid search and cross-validation were applied to tune max_depth, learning_rate, and n_estimators.
- Prophet:** Holidays, changepoints, and seasonalities were adjusted for optimal performance.
- Linear/Logistic Regression:** Regularization techniques (L1/L2) were tested for generalization improvement.

Forecasting Model Performance Report

The performance of the models was summarized as follows:

Model	MAE	RMSE	Comments
ARIMA	677.87	88935.096	Stable on linear trends, underperforms on seasonality
SARIMA	5575.2	9776.7	Best at modeling seasonality and trend
XGBoost	137	897.86	Outperformed traditional models with engineered features
Prophet	244.22	26353.7	Easy to tune, performs well on multiple seasonal patterns
Linear Regression	50.36358	18847.08	Simple, acts as baseline model

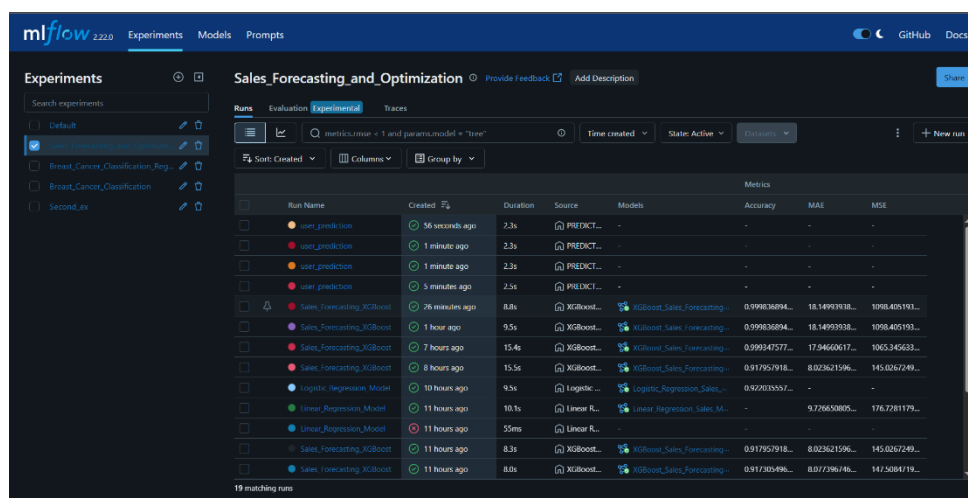
Final Forecasting Model

After comparison, [state final selected model, XGBoost] was selected as the final model due to its superior performance in capturing [linear/seasonal/nonlinear] patterns with the lowest forecast error. The final model was used to generate forecasts for the next period and will be deployed for production monitoring and periodic retraining.

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.



The screenshot shows the MLflow Experiments interface. The left sidebar lists experiments: 'Default', 'Breast_Cancer_Classification_Reg...', 'Breast_Cancer_Classification', and 'Second_ex'. The main panel displays the 'Sales Forecasting and Optimization' experiment. A table lists 19 runs with columns for Run Name, Created, Duration, Source, Models, and Metrics (Accuracy, MAE, MSE). The runs include 'user_prediction', 'Sales_Forecasting_XGBoost', 'Logistic_Regression_Model', and 'Linear_Regression_Model'.

Run Name	Created	Duration	Source	Models	Accuracy	MAE	MSE
user_prediction	56 seconds ago	2.3s	PREDICT...	-	-	-	-
user_prediction	1 minute ago	2.3s	PREDICT...	-	-	-	-
user_prediction	1 minute ago	2.3s	PREDICT...	-	-	-	-
user_prediction	3 minutes ago	2.3s	PREDICT...	-	-	-	-
Sales_Forecasting_XGBoost	26 minutes ago	8.8s	XGBoost...	XGBoost_Sales_Forecasting...	0.99936894...	18.14919318...	1098.405191...
Sales_Forecasting_XGBoost	1 hour ago	9.5s	XGBoost...	XGBoost_Sales_Forecasting...	0.99936894...	18.14919318...	1098.405191...
Sales_Forecasting_XGBoost	7 hours ago	15.4s	XGBoost...	XGBoost_Sales_Forecasting...	0.999347577...	17.9460617...	1065.345633...
Sales_Forecasting_XGBoost	8 hours ago	15.5s	XGBoost...	XGBoost_Sales_Forecasting...	0.917957918...	8.023621596...	145.0267249...
Logistic_Regression_Model	10 hours ago	9.5s	Logistic...	Logistic_Regression_Sales...	0.920035557...	-	-
Linear_Regression_Model	11 hours ago	10.1s	Linear R...	Linear_Regression_Sales...	-	9.726650805...	176.7281179...
Sales_Forecasting_XGBoost	11 hours ago	55ms	Linear R...	-	-	-	-
Sales_Forecasting_XGBoost	11 hours ago	8.3s	XGBoost...	XGBoost_Sales_Forecasting...	0.917957918...	8.023621596...	145.0267249...
Sales_Forecasting_XGBoost	11 hours ago	8.0s	XGBoost...	XGBoost_Sales_Forecasting...	0.917305496...	8.077396748...	147.5084719...

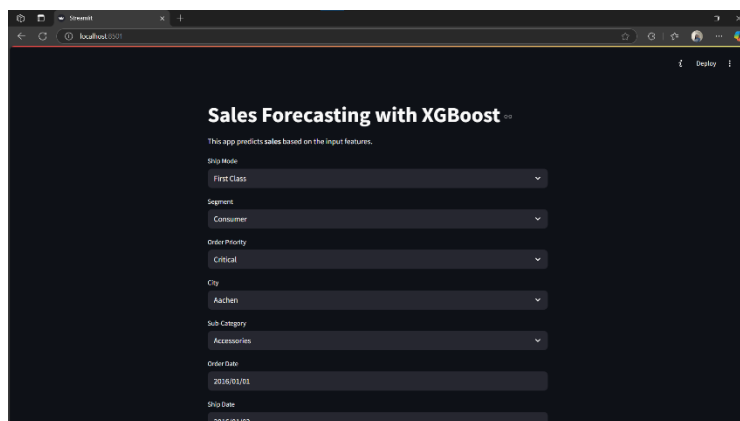
b. MLflow Setup for Model Deployment

- 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.

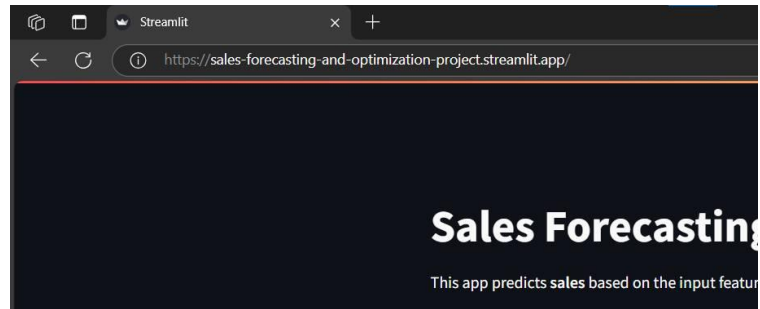
b. Real-Time vs. Batch Predictions

- 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.