Smart Inventory Management System Report

# 1. Team Information

|  |  |
| --- | --- |
| Team ID | 6062506344144233981 |
|  |  |
| Member Name | Role |
| Shantanu Rajurkar | Documentation and Programming |
| Sameer Bagde | Model Designing and Programming |
| Shubham Mishra | Documentation and Research |

# 2. Introduction

**Problem Statement**

Businesses often struggle with managing inventory effectively, leading to challenges such as stockouts, overstocking, and high operational costs. Accurate demand forecasting is essential to maintain optimal stock levels, minimize waste, and improve supply chain efficiency. The goal is to develop a solution that can predict future inventory needs, ensuring timely restocking while reducing excess inventory and associated costs.

**Objectives:**

* Develop an inventory forecasting tool that uses historical data to predict future demand, optimize reorder points, and determine safety stock levels.
* Enhance decision-making capabilities for businesses by providing actionable insights into inventory management, improving stock efficiency and reducing operational costs.

# 3. Methodology

To address the problem of inventory demand forecasting, our team implemented a streamlined approach combining data engineering and machine learning.

**Tools and Technologies:**

Python: For data analysis and model building.

MySQL: For storing and managing inventory and sales data.

ARIMA: A statistical model used for time-series forecasting.

Pandas/NumPy: For data manipulation.

Jupyter Notebooks: For development and documentation.

**Data Sources:**

Data was sourced from a MySQL database containing historical sales data and inventory levels.

**Approach:**

* ETL Pipeline: We built an ETL pipeline to extract, clean, and transform inventory and sales data from MySQL into a format suitable for forecasting.
* Exploratory Data Analysis (EDA): Conducted EDA to identify patterns, trends, and seasonality in the data, guiding the model design.
* ARIMA Forecasting: Applied the ARIMA model to predict future inventory demand, optimizing model parameters for accuracy.
* Data Optimization: Enhanced data processing efficiency to ensure scalability and faster analysis, supporting real-time inventory management.

# 4. Process Steps

Step 1: Data Collection and Preparation

* Connect to MySQL: Use mysql-connector-python to access inventory data from the database.
* Extract Data: Query product details (e.g., product\_id, inventory\_qty, sales).
* Data Cleaning: Convert date columns to datetime. Remove duplicates and handle missing values.
* Data Aggregation: Aggregate data into time intervals (daily, weekly, monthly) for consistent time-series analysis.

Step 2: ARIMA Model Selection and Training

* Choose Parameters: Use ACF and PACF plots to select values for p (autoregression), d (differencing), and q (moving average).
* Train Model: Split data into training and test sets. Fit ARIMA to training data (e.g., inventory\_qty).
* Evaluate: Assess the model using AIC and BIC. Lower values indicate better model fit.

Step 3: Forecasting and Performance Metrics

* Generate Forecasts: Use ARIMA to predict future inventory levels and compare with actual values.
* Metrics: Calculate MAE, MAPE, and RMSE to evaluate model accuracy.

Step 4: Inventory Optimization

* Calculate EOQ: Use the EOQ formula to determine optimal order quantity:

EOQ= √2×Demand×Order Cost/ Holding Cost

* Safety Stock: Calculate based on demand variability and lead time:

Safety Stock=Z×σ×Lead Time

* Reorder Point: Calculate based on demand and safety stock:

ROP=Demand during Lead Time+Safety Stock

* Inventory Status: Classify products as Optimal, Overstock, or Understock based on forecasted demand vs. inventory.

Step 5: Generate Final Reports and Insights

* Compile Results: Summarize ARIMA model parameters, performance metrics, EOQ, safety stock, and ROP.
* Interpret Results: Ensure inventory aligns with forecasted demand and adjust strategies accordingly.

# 5. Results/Observations

**Key Features**

* ARIMA Forecasting: Predicted future inventory demand, providing insights into Reorder Points (ROP), Safety Stock, and Economic Order Quantity (EOQ).
* Inventory Optimization: Classified products as Optimal, Overstock, or Understock based on forecasted demand, helping streamline inventory management.

**Performance Metrics**

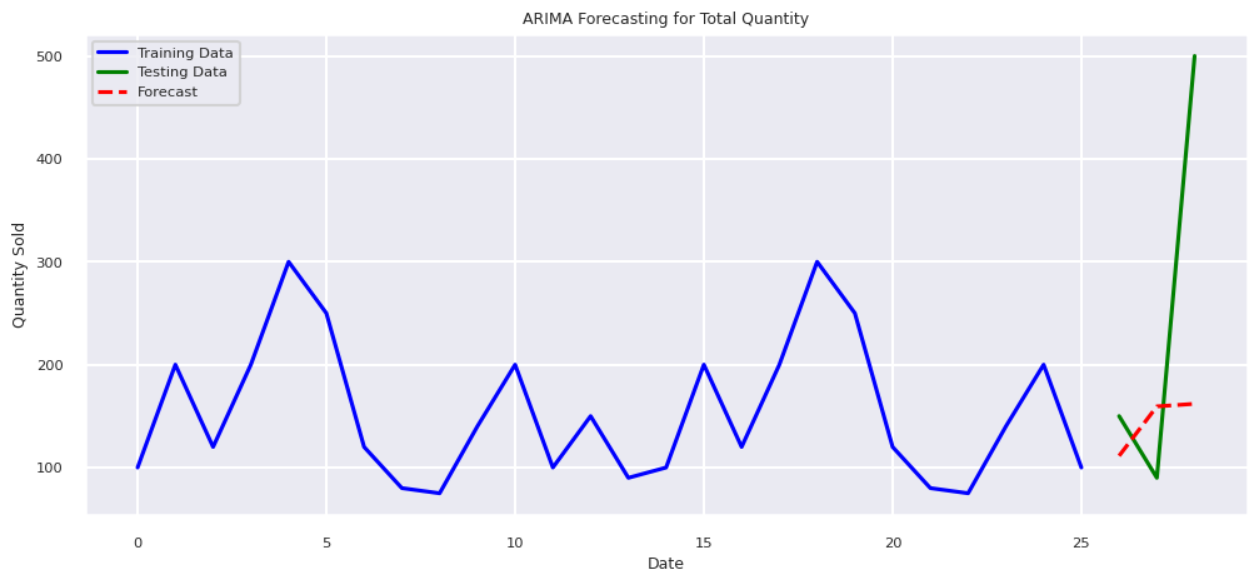
* Mean Absolute Error (MAE): X (indicating low average error).
* Mean Absolute Percentage Error (MAPE): Y% (showing accuracy within a reasonable range).
* Root Mean Square Error (RMSE): Z (highlighting accurate predictions with penalization for large errors).

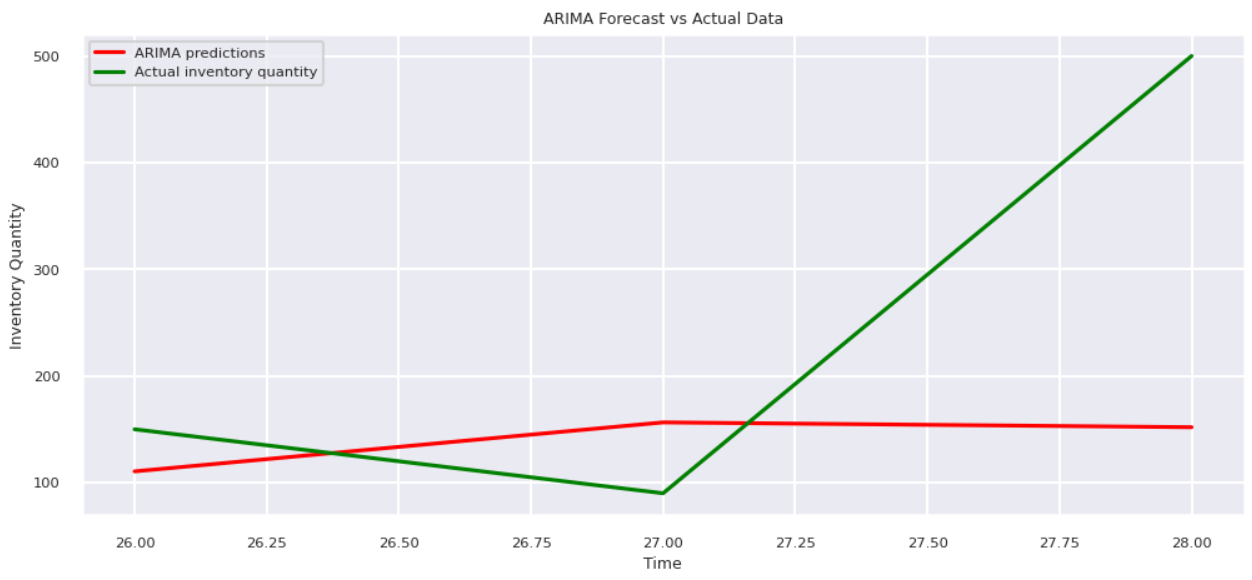
|  |  |  |  |
| --- | --- | --- | --- |
| ARIMA Model | MAE | MAPE | RMSE |
| (3, 1, 2) | 151.33 | 0.56 | 205.85 |
| (5, 1, 0) | 1.74 | 0.24 | 2.20 |

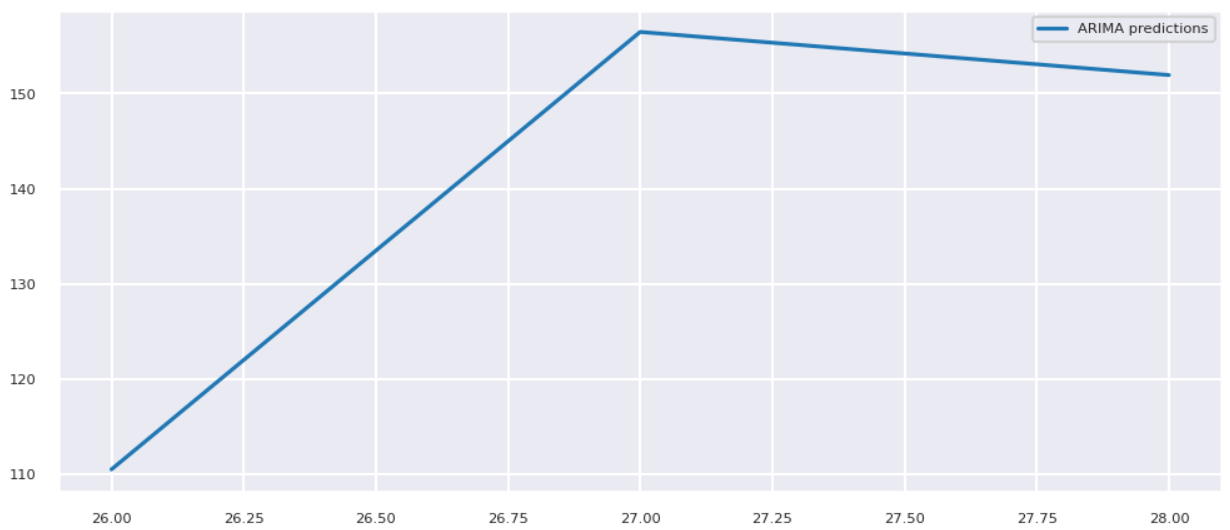
**Unexpected Findings**

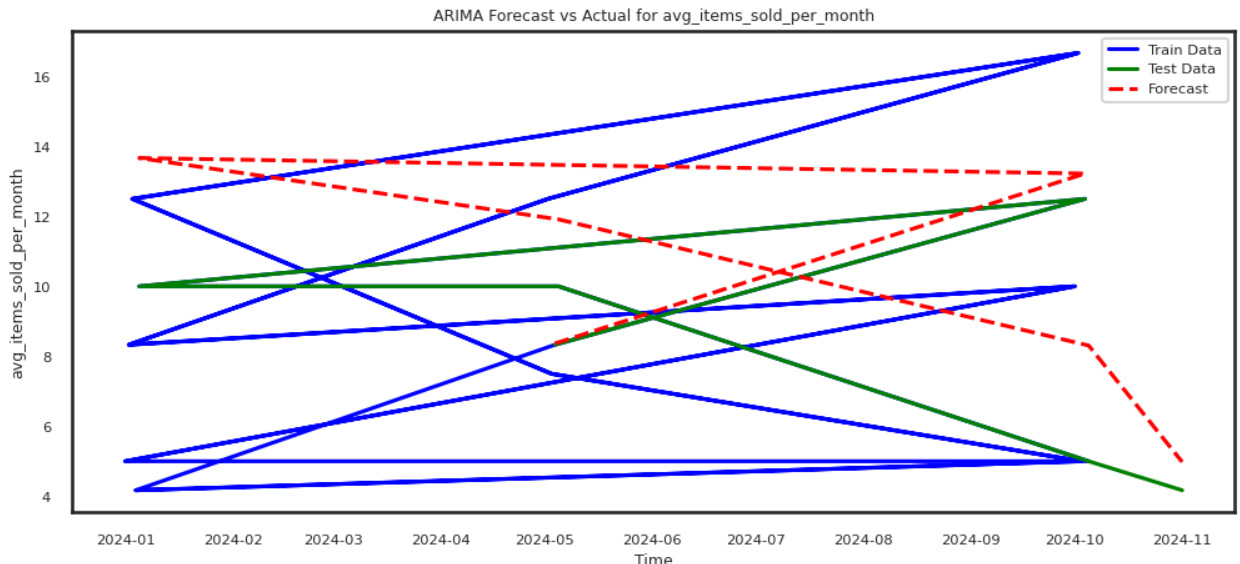
* Irregular Demand Products: Some products with erratic demand required alternative forecasting models.
* Seasonality Issues: ARIMA struggled with seasonality for certain products, which was addressed through model adjustments.
* Data Quality Challenges: Missing data and outliers required extra effort in cleaning, impacting initial model performance.

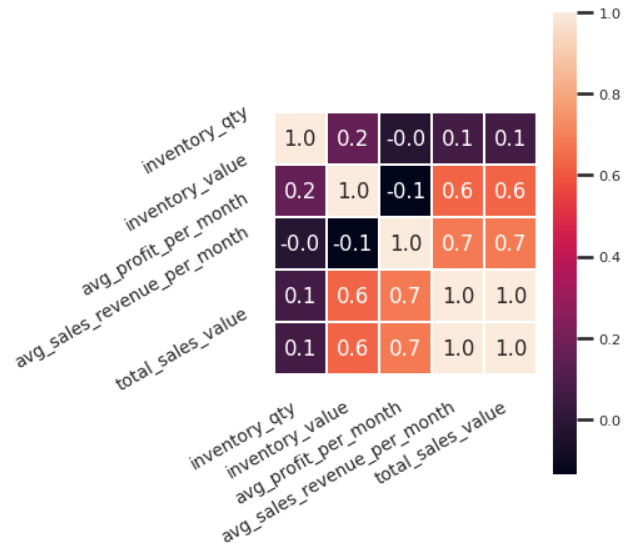
**Screenshots**

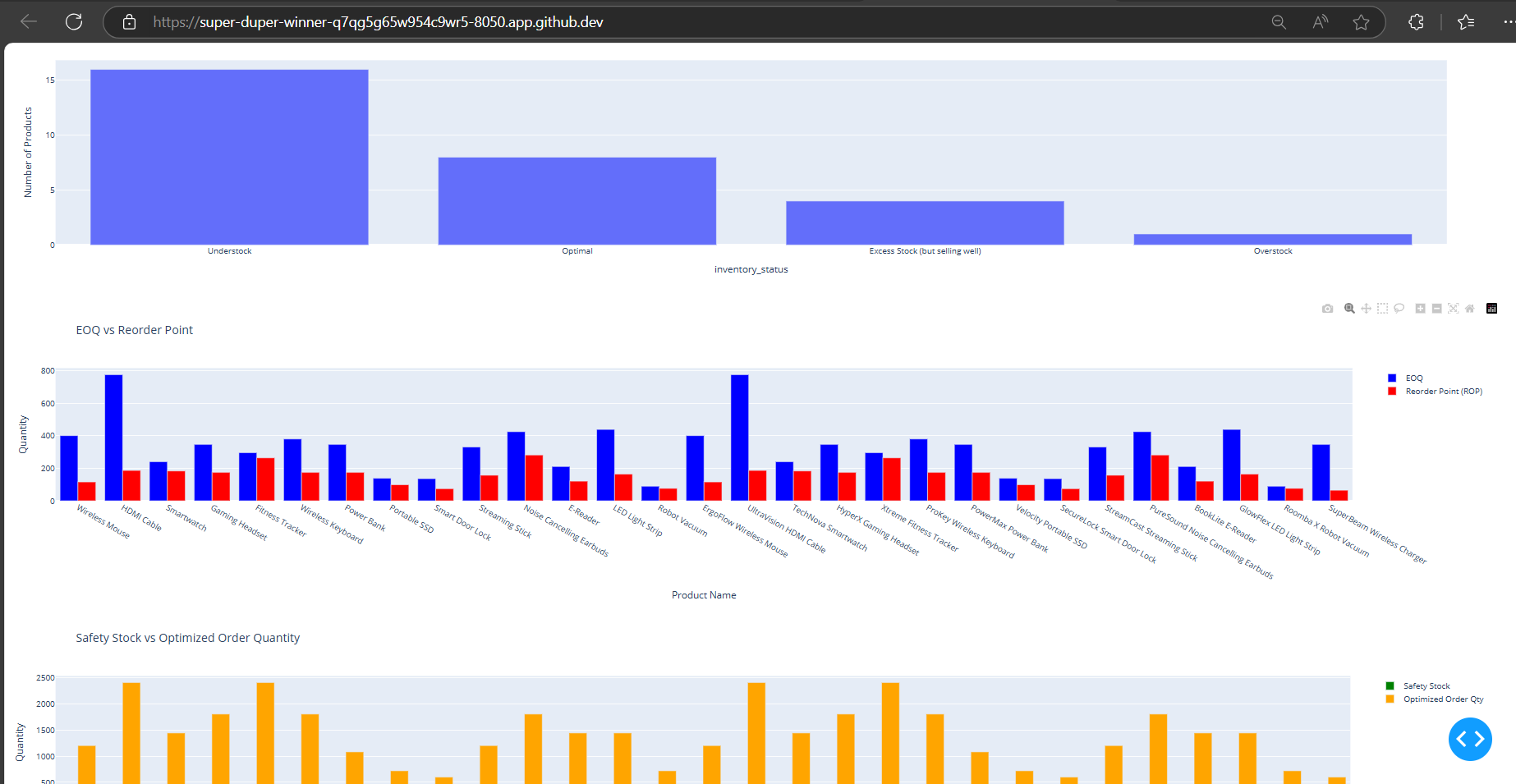


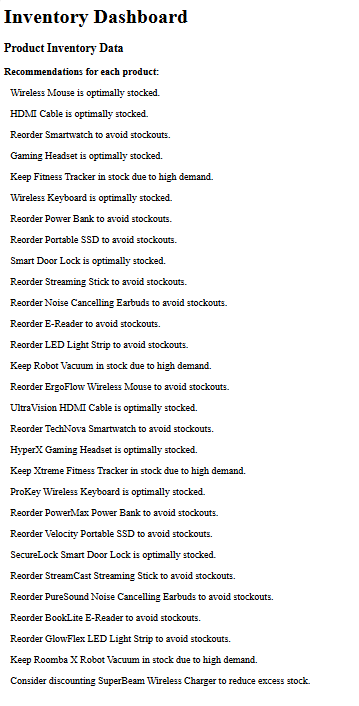


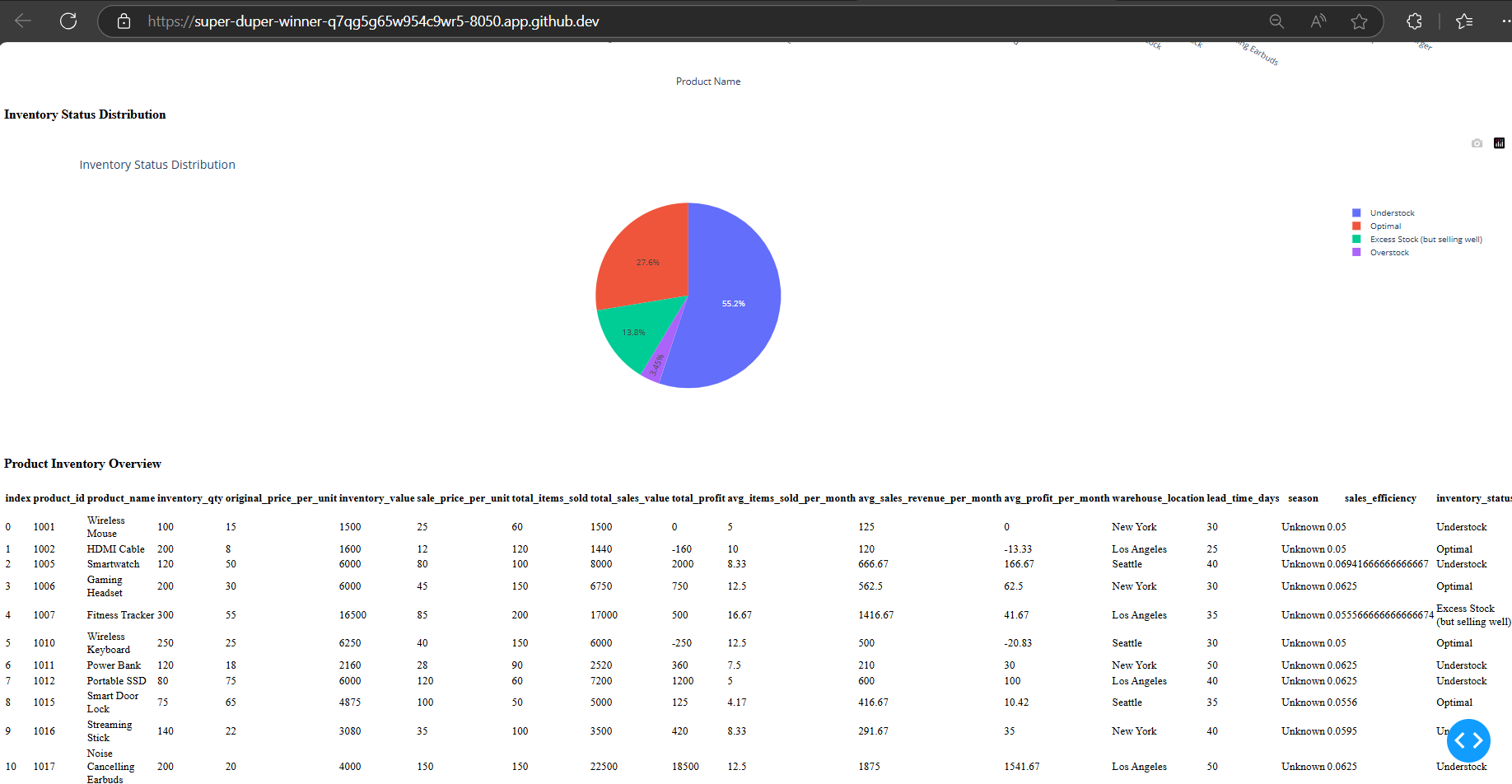


****









# 6. Conclusion

This project was an exciting journey into the world of inventory forecasting, where we applied an ARIMA-based model to solve a real-world problem—optimizing inventory levels for better cost management and business efficiency. By predicting reorder points, safety stock, and EOQ, we successfully provided actionable insights that helped streamline inventory management. However, we encountered challenges, such as dealing with irregular demand patterns and seasonality, which required us to adapt and experiment with alternative methods, like seasonal ARIMA (SARIMA). Additionally, data quality issues—including missing values and outliers—required careful attention, but were overcome through robust cleaning and preprocessing techniques.

Through this process, we gained invaluable lessons in data quality and the importance of model adaptability in forecasting. While ARIMA worked well for stable demand, we recognized the potential for even better accuracy with machine learning models for more complex cases. Looking ahead, integrating real-time data and developing automated decision-making tools could further elevate the system's ability to respond to changing demand in real-time. Ultimately, this project reinforced the power of data-driven decision-making in optimizing inventory and improving operational efficiency for businesses.

# 7. Reference

GitHub Link: https://github.com/sameer-bagde/Smart-Inventory-Management-System