E-Commerce Inventory Management and Sales Forecasting System

Harshil Nandwani, David Ryan, Rishub Talreja

Github: https://github.com/harshil017/DS5110

nandwani.h@northeastern.edu, ryan.d5@northeastern.edu, talreja.ri@northeastern.edu

Abstract

The E-Commerce Inventory Management and Sales Forecasting System is aimed toward enhancing operational efficiency and decision-making for online retail platforms. Employees and admins can manage sales data, monitor stock levels, and optimize inventory strategies. It would also contain three predictive models tailored o distinct business needs: (1) a SARIMAX model for time series forecasting with a Mean Absolute Error (MAE) of ~18,000, enabling accurate daily sales predictions; (2) a K-Means clustering model for customer segmentation, identifying buyer groups for personalized marketing; and (3) a Random Forest regressor for gross sales prediction. The system also features a scalable application for real-time sales analysis, providing insights to optimize inventory, and ability to add additional customers and orders to current database.

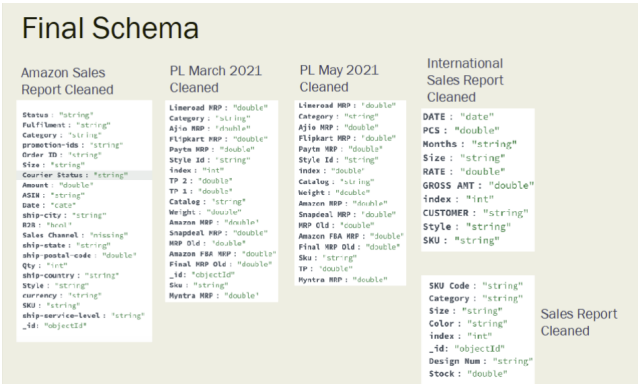
Introduction

The objective of our system is to manage an e-commerce store's sales data, order history, and customer transactions, based on the dataset we found. The system will allow the admin to monitor product performance, edit any incorrect information, and review sales history. Managers will be able to view product history and data-driven insights will be provided to optimize inventory management and sales strategies. Additionally, the system will include tools for forecasting product demand, ensuring better stock availability, and analyzing customer purchasing patterns.

Database Design

The E-Commerce Inventory Management and Sales Forecasting System is designed to empower online retail platforms by providing robust tools for managing sales data, monitoring inventory, and generating actionable insights. Admins will have the ability to edit and update records, monitor real-time stock levels, track product performance, and utilize forecasting models such as SARIMAX for daily sales predictions and Random Forest for gross sales forecasting. Additionally, clustering models such as K-Means will enable customer segmentation, helping admins design personalized marketing strategies. Managers will benefit from viewing historical sales data, accessing real-time inventory updates, and leveraging data-driven insights to optimize operational strategies. The system will track sales transactions, inventory levels, customer purchase patterns, and product performance metrics. In addition, both users can view the prices their competitors are posting for the same products to help make informed decisions.

The system's adoption of MongoDB provides significant advantages. MongoDB's flexible, schema-less structure is ideal for handling diverse data formats from multiple sources, as seen in this project. Its scalability ensures the system can seamlessly manage growing datasets as the business expands. Real-time querying capabilities allow for rapid retrieval and updating of data, which is critical for live inventory monitoring and sales forecasting. Furthermore, MongoDB's support for rich queries simplifies complex operations like filtering by category, aggregating sales metrics, or tracking customer purchasing patterns. This versatility, combined with its ability to handle unstructured and semi-structured data, makes MongoDB an excellent choice for developing a dynamic and efficient system that adapts to evolving business needs.



1. **The Amazon Sales Report Cleaned Table** consists of detailed documentation for each transaction on Amazon, including fields like the order status, fulfillment type, product category, courier status, and shipping details such as city, state, and postal code. It also tracks critical sales metrics such as order amount, quantity,

and the sales channel. Additionally, it holds SKU and ASIN identifiers for product tracking, enabling admins to analyze performance effectively.

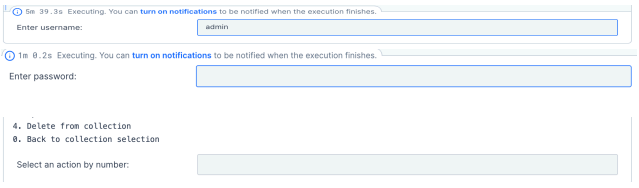
2. **The PL March 2021 Cleaned Table** contains comprehensive pricing data for various e-commerce platforms, including Amazon, Flipkart, Myntra, Snapdeal, and Ajio. It provides details such as the MRP, category, catalog, weight, and style ID for each product. This table serves as a reference for price comparisons and performance metrics.
3. **The PL May 2021 Cleaned Table** mirrors the structure of the March 2021 table, with updated data for May. It includes similar fields like MRP across platforms, weight, catalog details, and style IDs, ensuring consistency in analyzing price trends over time.
4. **The International Sales Report Cleaned Table** tracks global sales data, including metrics such as the number of pieces sold (PCS), rates, gross amounts, and customer details. It also includes information about the month of sale, product size, and SKU, providing insights into international trends and customer preferences.
5. **The Sales Report Cleaned Table** contains SKU-level details, including the design number, category, size, color, and available stock. It helps track product availability and provides a foundation for inventory management and replenishment planning.

Application Description

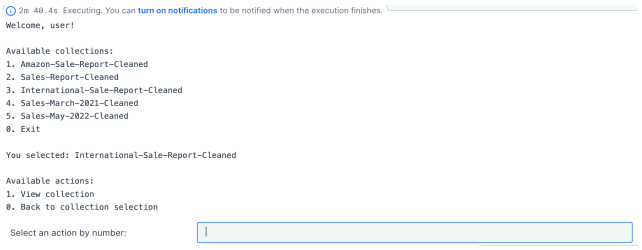
The entire data for our ecommerce system is stored in a MongoDB database, which is accessed and managed using PyMongo. This application provides a command-line interface for both admin and regular users, allowing them to interact with the data. Admins have complete control over the collections, while regular users have limited access.

Users: Users are authenticated with predefined credentials. Based on the role (admin or user), appropriate options are displayed. User authentication ensures only authorized access.

1.Admin: Admins can perform all operations such as viewing, inserting, updating, and deleting records within any collection.



2. Other Users: Regular users can only view collection data for analysis and exploration.

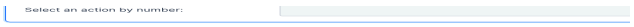


Functionalities:

1. View Collections:

Fetches all collection names dynamically from the MongoDB database and displays them as options. Users can select any available collection from the database.

Displays the data's head, tail, and summary statistics for numerical and categorical fields.



2. Insert Records:

Admins can insert new records into a selected collection. The application automatically provides field prompts based on existing documents in the collection.

3. Update Records:

Admins can update specific fields of a document by providing its unique identifier (_id).

Ensures only valid fields and documents are updated. Automatically adapts to database changes without requiring code modifications.

4. Delete Records:

Admins can delete any document by specifying its unique identifier (_id). Ensures data integrity with confirmation.

```
3. Update collection
4. Delete from collection
0. Back to collection selection

Select an action by number:
```

Data Collection and Data Cleaning

The data for our project was sourced from a publicly available uncleaned e-commerce sales dataset from Kaggle, stored on our MongoDB database. The dataset comprises approximately 200,000 records collections, including sales reports from domestic and international transactions. These collections were accessed using the PyMongo library (MongoDB's Python API package) and transformed into pandas DataFrames for analysis and processing.

Handling Missing Values

There contained missing data points in several critical columns, including 'Amount', 'currency', 'ship-city', and 'promotion-ids' from the *Amazon-Sales-Report* collection. We've addressed these gaps by doing the following:

- 'Amount': Missing values in this column, a key metric for sales analysis were imputed using the mean of the corresponding product category. This ensured while avoiding significant distortion of the data.
- 'currency': For transactions originating in India (identified through the ship-country column), the currency was consistently set to "INR." This ensured uniformity and resolved ambiguities for domestic transactions.
- 'ship-city' **and related address fields:** Missing location data were forward-filled where logical, leveraging neighboring values from chronologically or geographically related records. This approach ensured that imputed values remained contextually relevant.
- 'promotion-ids': Missing values were replaced with "None" to indicate the absence of promotional activity for the transaction, preserving clarity without introducing bias.

Removing Irrelevant Data

Several columns were deemed unnecessary for the projects analytical tasks:

- **Irrelevant columns such as 'Unnamed: 22'** were dropped as they provided no meaningful information.
- The 'fulfilled-by' column, which contained sparse and inconsistent data, was excluded after determining it did not contribute significantly to inventory or sales forecasting tasks.
- Duplicate records, which were identified through unique identifiers such as Order ID, were removed to prevent double-counting and ensure the accuracy of aggregate analyses.

Standardization

To prepare the data for analysis and modeling, numerous fields were standaradized:

- **'Date'** in the Date column were converted to a uniform datetime format to enable consistent time-based analyses such as seasonal trends and sales forecasting.
- Numerical data, such as 'Qty' and 'Amount', were validated to remove outliers and ensure all values fell within plausible ranges.

Data Analysis

Using MongoDB queries we extracted subsets of data from collections like 'Amazon-Sale-Report' and 'International-Sale-Report'. After cleaning, we stored the data into a cleaned version of the e-commerce MongoDB database. Key metrics, such as total sales, product performance, and customer trends, were computed directly within the database for efficiency. For example, group-by aggregation was used to get the Total Sales by Category in 'Amazon-Sale-Report':

```
}
```

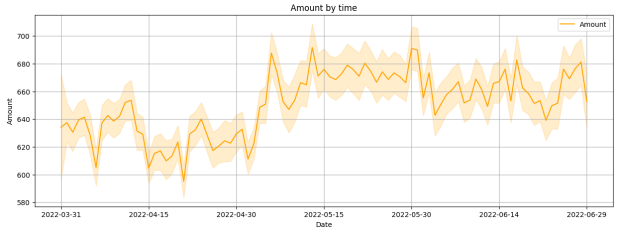
Order metrics per customers from 'International-Sale-Report' were also generated to show how much each customer spent and what the average rate is. Here is a subset of those metrics:

```
}
```

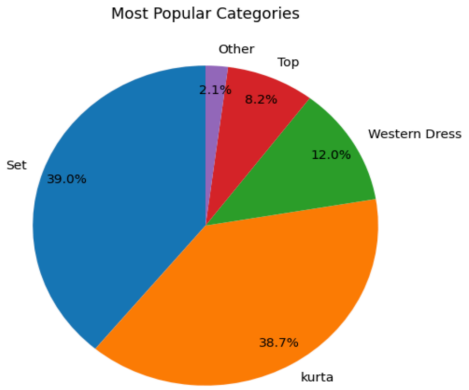
Data is then loaded into pandas DataFrame for further analysis and visualization.

Sales Trends

A clear pattern of seasonal spikes in sales was observed, particularly during promotional periods and festivals. Largely, the Kurta and Set items consistently dominated sales, contributing over 77% of total revenue.

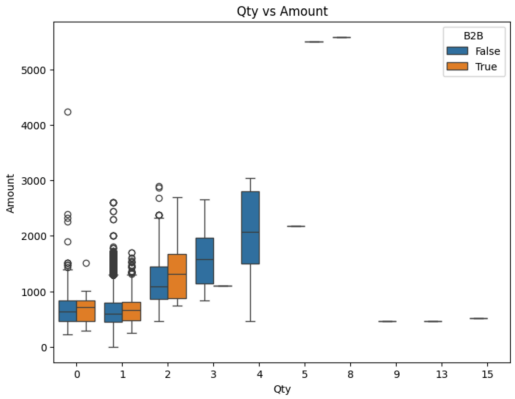


Line plot of the trend in sales amounts by time:



Pie Chart showing the most popular categories in the 'Amazon-Sale-Report' (Note: Smaller categories grouped into "Other" for clarity (threshold set to 1% of total sales).

Customer Insights

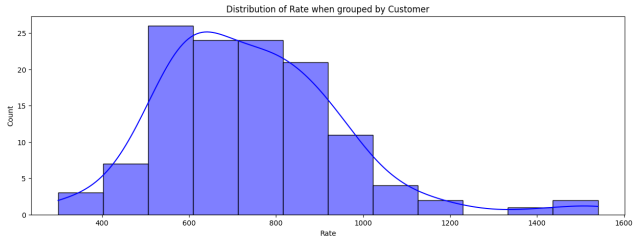


A significant portion of revenue was driven by repeat customers, particularly in the B2B (business-to-business) segment. This indicates potential benefits from targeted loyalty programs for these high-value clients. The following box-plot representation represent the relationship between quantity and sales amount, categorized by whether the transactions are B2B:

The spread of amounts widens as quantity increases, indicating more variability in transaction amounts for larger orders. B2B orders show more variability in amounts for smaller quantities possibly due to diverse product pricing or customer-specific agreements. For certain quantities (e.g. 'Qty' ≥ 5), some data points are separated far from the rest, indicating transactions that are unusually high-value or outliers in the dataset.

Behavioral Trends

Most customers prefer small-to-mid purchases, which shows that most customers engaged in low-risk purchasing behavior. This could indicate sensitivity to pricing or preference for regular consumption. The following distribution plot illustrates the distribution of rates (average revenue per purchase) grouped by customers.



Machine Learning Models

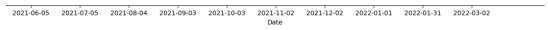
Machine learning enables data-driven solutions for business challenges in e-commerce. This project utilized key techniques to address sales forecasting, customer segmentation, and predictive modeling. A SARIMAX model predicted sales trends using historical data and external factors like quantities sold and B2B transactions, improving demand forecasting. K-Means clustering segmented customers based on spending and order patterns, supporting targeted strategies. A Random Forest regression model predicted sales by leveraging product attributes, with preprocessing steps like scaling and encoding ensuring model effectiveness. By applying these methods, we demonstrated how machine learning improves operational efficiency and provides actionable insights for data-driven decision-making in e-commerce.

Reports

Correlation Insights: Positive correlation between 'RATE and 'GROSS AMT' in 'International-Sales-Report'. Weak correlation between the two represents diverse purchasing patterns.

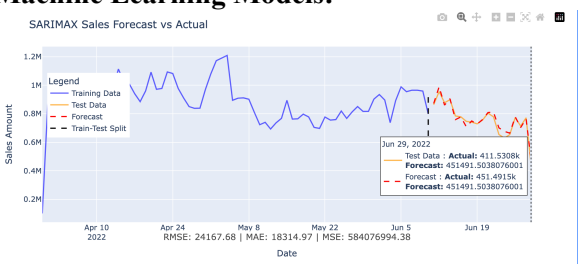
Correlation Matrix

Rate by Date



Gross Amount by Date:

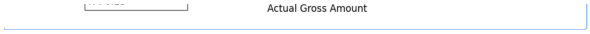
Machine Learning Models:



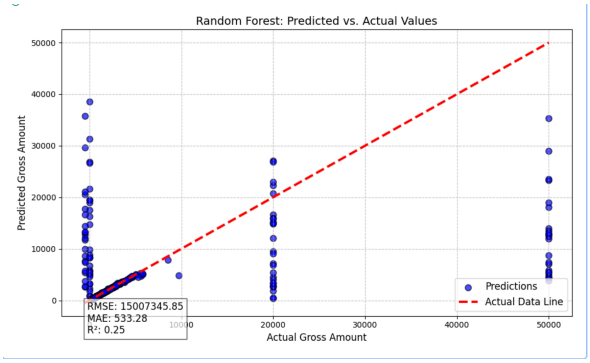
Sarimax Time Series Model for Sales Data Forecasting

Elbow Method to find Optimal Number of Clusters

K-Mean Clustering for Customers based on location and spending (Perfect Split k=2)(Silhouette =1)



Regressor to Predict Gross Sales Amount for Amazon



Conclusions/Future DirectionsRegressor to Predict Gross Sales Amount for Amazon

This project provided insights into the complexities of designing and implementing an E-Commerce Inventory Management and Sales Forecasting System. Learning how to apply data cleaning methods, advanced predictive modeling, and a flexible MongoDB database, the project taught the importance of balancing technical implementation with business-focused objectives. The main learning curve was how to manage a large and diverse dataset, ensuring data integrity for analysis and modeling. Additionally, the development of predictive models, including SARIMAX and Random Forest, taught us the trade-offs between model complexity and interpretability. The use of MongoDB showcased its value in providing schema flexibility and scalability for handling unstructured data in dynamic environments.

For the future, the project will be enhanced by creating more models to include advanced techniques like neural networks, enabling the exploration of more complex patterns within the data. In addition, a real-time analytics GUI dashboard will be developed to make the system more accessible and user-friendly for stakeholders. Integrating external APIs for live competitor pricing and inventory tracking will be helpful to constantly view updated insights. Furthermore, applying Natural Language Processing (NLP) to analyze customer feedback would allow for targeted marketing strategies.

Future students undertaking similar projects are advised to begin with a clear and structured plan to define objectives and develop an early understanding of the dataset. Emphasis should be placed on thorough data cleaning, as a well-prepared dataset is fundamental to achieving accurate and meaningful results. Students should also be prepared to iterate on their models and make sure to maintain detailed documentation throughout the process. Going to office hours for regular feedback is a must to find ways for potential improvements.

In conclusion, this project exemplified the integration of technical and analytical skills with business objectives, resulting in a system capable of addressing practical challenges in e-commerce. The experience underscored the importance of aligning data-driven solutions with operational goals, providing a foundation for further exploration and application in the field.