**Milestone 5 - Term Project Paper**

**Optimizing E-Commerce Store Profits through Product Recommendation and Inventory Management**

Nathanael I. Ochoa

Bellevue University

DSC630: Predictive Analytics

Professor A. Hua

August 10, 2024

# Introduction

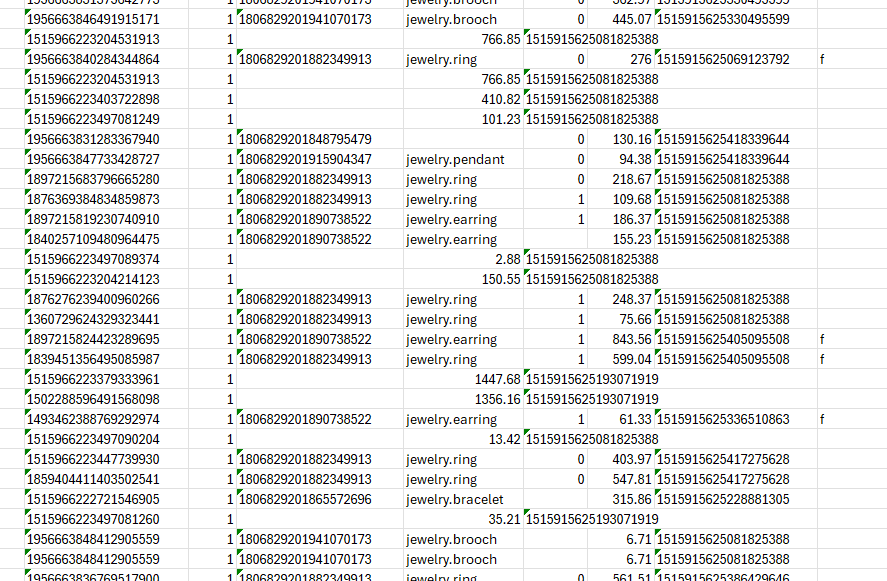
The objective of this project is to enhance the profitability of an e-commerce jewelry store by refining the shopping experience through personalized product recommendations and optimizing inventory management. Referred to as "630 Jewelers" for the purposes of this project, the endeavor aims to enhance customer satisfaction and optimize operations by developing a product recommender and an inventory management model. These tools will leverage insights from purchase history and demand patterns to address the unique requirements of 630 Jewelers.

Upon encountering this dataset, I became intrigued by the prospect of analyzing real-world data within a simulated scenario. Considering that businesses often enlist contractors for short-term projects, I saw this as an opportunity to develop a project mirroring potential future endeavors in my career. My aim is to conceive an idea closely aligned with practical business scenarios.

## **Data Selection**

I’ve selected a dataset from [Kaggle.com](http://kaggle.com) titled “[eCommerce purchase history from jewelry store](https://www.kaggle.com/datasets/mkechinov/ecommerce-purchase-history-from-jewelry-store).” This dataset includes various columns such as Order datetime, Order ID, Purchased product ID, Quantity of SKU in the order, Category ID, Category alias, Brand ID, Price in USD, User ID, Product gender, Main color, Main metal, and Main gem (note that this is before any data cleanup). Collected from an online jewelry store, the data spans from December 2018 to December 2021, encompassing 95,911 records, which should suffice for model building. I would have loved to have more data on the consumers making these purchases, but that may have violated privacy laws. However, I am content with the data I have to work with. This dataset was chosen for its rich information on customer purchases, pivotal for developing a recommendation system and predicting inventory demand.

## **Data Preparation**

In my initial inspection of the data in Excel, I noticed that some rows had cells shifted two spaces to the left, placing them in the wrong columns. I encountered difficulties when trying to correct this directly in Excel, so I made the necessary adjustments using Python instead.

After fixing the column alignment issues, I used the ‘fillna()’ function to address any null values. Specifically, I filled null values in the ‘main\_gem’ column with ‘unknown-gem’, aligning with the existing syntax in the ‘main\_color’ column (‘unknown-color’). I also replaced null values in the ‘brand\_id’ column with ‘-1’ to indicate ‘unbranded’ products. Additionally, I filled null values in the ‘product\_gender’ column with ‘u’ to denote ‘unisex’. These decisions were made to the best of my ability and aimed at preserving the integrity of the data.

Any remaining rows with ‘NaN’ values were removed from the dataset. The original dataset contained 95,911 records; after these modifications, the dataset was reduced to 78,391 records.

Next, I updated the data types of the columns as needed. Some integer columns had been interpreted as floating-point values due to the presence of ‘NaN’. I also converted the ‘order\_datetime’ column to datetime format and set it as the index of the dataframe.

# Models & Methods

For the product recommendation system, I originally planned to use TF-IDF vectorization in conjunction with cosine similarity. However, due to frequent crashes on my PC, I switched to content-based filtering. This method recommends items similar to those a user has purchased based on the item's features.

To implement this, I first performed one-hot encoding to convert categorical features (such as ‘category\_alias’, ‘product\_gender’, ‘main\_color’, and ‘main\_metal’) into a binary format. This encoding creates a feature matrix where each row represents an item and each column corresponds to a specific feature. I then calculated item similarities using cosine similarity, which measures the angle between feature vectors to provide a similarity score ranging from 0 (no similarity) to 1 (identical). Finally, I generated recommendations by constructing an item similarity matrix and aggregating scores for items not yet interacted with, based on the similarity to items the user has previously purchased. I then implemented a for loop to print the ‘product\_id’ and some attributes in a readable format. 

For the inventory management portion, I encountered issues with my dataset while attempting to create an ARIMA model. Consequently, I opted for an Exponential Smoothing model instead. I focused on the product with the highest frequency in the dataset, which is a women’s ring, and applied the Holt-Winters' additive seasonal method to this data. To evaluate the model, I plotted the observed data alongside the fitted and forecasted values and calculated the Mean Absolute Error (MAE) and Mean Squared Error (MSE), which were 5.13 and 40.17, respectively. Although this level of error was relatively reasonable, I aimed to improve the model's performance. I initially used the additive seasonal model but switched to the multiplicative model, which resulted in a significantly better error rate.

# Results

The updated model produced an MAE of 2.85, indicating that, on average, each forecast deviates by 2.85 units. Given the large size of the dataset and the fact that there are 7,644 different items sold, an MAE of 2.85 is quite good. I would be more concerned if my estimates deviated by values greater than 10, but an average deviation of 3 units is considered quite satisfactory.

# Conclusion & Recommendations

I am pleased that, after the data cleanup phase, I was able to retain up to 81% of the original records in the dataset. Had I removed all records with null values without addressing them, I would have been left with only a third of the original dataset. While the models I initially planned did not materialize as expected, I am satisfied with the results I achieved.

Now, for my recommendations to the owner of 630 Jewelers: I suggest implementing the product recommender I developed and enhancing its effectiveness by displaying images of the recommended items, instead of just their attributes. It would also be beneficial to include the store’s stock count in the data and to prioritize inventory that is struggling to sell. If such items are included in the recommendation list, moving them to the top could help increase their sales. However, this approach should be used sparingly and primarily when there is a need to move slow-selling inventory. Additionally, incorporating the recommender at checkout for online shoppers and including it in email marketing campaigns could further boost sales, as many consumers are more likely to purchase from stores they have previously had a positive experience with.

The inventory management model has a relatively low error of about 3 units. I recommend ordering an additional 3 units beyond the forecasted amount. As more sales data becomes available, the model can be refined and updated to improve its accuracy.

# 

# References

Kechinov, M. (2021, December 1). *ECommerce purchase history from Jewelry Store*.

Kaggle. https://www.kaggle.com/datasets/mkechinov/ecommerce-purchase-history

-from-jewelry-store

*Drive more revenue with niche-specific personalization engine*. REES46. (n.d.).

https://rees46.com/

GeeksforGeeks. (2024b, May 23). *Movie Recommender based on plot summary using*

*TF-IDF vectorization and cosine similarity*. https://www.geeksforgeeks.org/movie-

recommender-based-on-plot-summary-using-tf-idf-vectorization-and-cosine-similarity/