Journal of Economics, Finance and Accounting Studies
ISSN: 2709-0809
DOI: 10.32996/jefas
Journal Homepage: www.al-kindipublisher.com/index.php/jefas

JEFAS
AL-KINDI CENTER FOR RESEARCH AND DEVELOPMENT

| RESEARCH ARTICLE

Novel AI-Powered Dynamic Inventory Management Algorithm in the USA: Machine Learning Dimension

Md Zahidul Islam[1], Nisha Gurung[2], Md Sumon Gazi[3] and Md Rokibul Hasan[4] ✉
[1234]MBA Business Analytics, Gannon University, Erie PA, USA
Corresponding Author: Md Rokibul Hasan, **E-mail**: prorokibulhasanbi@gmail.com

| ABSTRACT
Dynamic inventory management revolves around the practice of progressively modifying inventory degrees to adapt to fluctuations in client demand, production, and supply chain dynamics. At the center, inventory management focuses on upholding enhanced levels of stock to balance consumer service via availability with the costs related to holding excess inventory. This research paper aimed to explore the dynamic inventory management activities employed by organizations in the USA, shedding light on the machine learning strategies that can be deployed and their implications. The performance of the algorithms was empirically evaluated in a Python program experiment utilizing real-world data. To facilitate the data for input into the Neural Network, feature engineering, and selection were imposed to affirm its suitability. This study proposes the Sequence-to-Sequence (Seq2Quant) algorithm, a neural network-powered technique for demand prediction in inventory management. The current experiment compared and contrasted the performance of the Neural Networks against the following baselines, most notably, Naïve Seasonal Forecast, Moving Average Forecast, ARIMA, Naïve Seasonal Forecast with Averaging over four periods, SARIMAX. From the experiment, it was evident that the Seq2Seq had the lowest MAE (17.44) and the lowest SMAPE (66.91), suggesting that it was the best-performing algorithm overall. Besides, SARIMAX and ARIMAX also performed well, with MAE values of 18.33 and 18.09, respectively.

| KEYWORDS

Dynamic Inventory; Neural Network (NN); Recurrent Neural Network (RNN); Seq2Quant; Naïve Seasonal Forecast; Moving Average Forecast; ARIMA; SARIMAX.

| ARTICLE INFORMATION

ACCEPTED: 02 April 2024 **PUBLISHED:** 27 April 2024 **DOI:** 10.32996/jefas.2024.6.2.12

1. Introduction

Chen & Chao (2020), states that inventory management is an imperative function for companies across sectors in the United States, allowing them to balance demand and supply efficiently. Dynamic inventory management revolves around the practice of progressively modifying inventory degrees to adapt to fluctuations in client demand, production, and supply chain dynamics. Efficient inventory management is a strategic advantage for companies, and dynamic methods afford several competitive benefits. By assessing demand trends and modifying inventory levels respectively, organizations can diminish stockouts, minimize carrying costs, enhance production schedules, and boost customer satisfaction. As per, Cuartas & Aguilar (2023), dynamic inventory management also enables businesses to respond to market changes rapidly, improving overall agility and competitiveness. This research paper intends to explore the dynamic inventory management activities employed by organizations in the USA, shedding light on the strategies deployed and their implications.

Chaudhary et al. (2023), indicate that at the center, inventory management focuses on upholding enhanced levels of stock to balance consumer service via availability with the costs related to holding excess inventory. Conventional inventory management frameworks were grounded on presumptions of supply lead times and solid demand patterns. schedules and replenishment

quantities were fixed. Nevertheless, in the contemporary complex business atmosphere, demand changes continuously and is impacted by a different factor such as new product launches, seasonal trends, competitor activities, and the macroeconomic state. Supply chains have also become internationally circulated with variable lead times and a greater risk of disruptions.

Demizu et al. (2023), argue that to resolve the uncertain and dynamic business landscape, dynamic inventory management has arisen as an advanced method. Dynamic inventory management depends on the progressive assessment of supply chain events, real-time demand signals, sales predictions, and other associated data sources to adapt inventory replenishment frameworks dynamically on a progressive basis. The prime concept is that supply chain variables and demand trends are in constant flux, so inventory frameworks need to be adaptive and fluid based on the most up-to-date real-time data. Contrary to the mainstream fixed policies, dynamic inventory management targets to right-size stock levels in real-time by taking a more data-oriented and responsive approach.

2. Literature Review

While the present prevalence revolving around Artificial Intelligence is relatively new, the notion of utilizing machine learning for predictive purposes has been in existence for many decades. In a comprehensive examination of current literature, De Melo (2019), highlighted that the first usage of Neural Networks in prediction as far back as 1974. Nonetheless, research in this domain was quite limited until the development of the backpropagation models in the 1980s, which enabled the training of deep networks. This era of early investigations is attributed to enthusiasm for the new technique, which was deemed a capable technique for modeling non-linear time series. Based on an extensive review of the existing literature, De Melo (2019), concluded that Neural Networks exemplified usefulness and suitability in delivering satisfactory performance and predicting tasks.

Lolli et al. (2022) conducted a comprehensive empirical study on predicting inventories using Neural Networks. Particularly, they performed a comparative examination between an automated Box-Jenkins forecasting and a Multilayer Perceptron (MLP) network system, evaluating a dataset comprising 75-time series. Their research exposed that the simple neural network algorithm achieved similar level outcomes compared to the Box-Jenkins predictive system. On the other hand, Meisheri et al. (2023) found out through their research on 384 demographic and economic time series that Neural Networks produced relatively accurate predictions compared to traditional methods. The predictions produced by the networks were comparatively superior to those of a linear regression algorithm and a weighted average of six simpler techniques.

Even though Machine Learning techniques, exemplified good performance, Namir & Labriji (2022), presented a hybrid approach that consolidated Machine Learning with a statistical technique, which yielded excellent results. Chaudhary et al. (2020) observed that this specific algorithm employed a Neural Network to learn and predict seasonality trends. Impressively, the Neural Network did not only learn these trends for every individual series independently but rather from a consolidation of all the series. These results led Makridakis and coworkers to hypothesize that utilizing information from multiple series to forecast individual series yields favorable outcomes.

In their research, Demizu et al. (2023) proposed a global Long Short-Term Memory (LSTM) framework. As an extension to their method, they adopted different clustering models to pinpoint identical time series and suggested training the frameworks exclusively on these clusters. Their investigation ascertained that this clustering method led to enhanced predictions. The recommended algorithms demonstrated competitive performance when assessed on the datasets from two previous prediction competitions.

3. Methodology

The performance of the algorithms was empirically evaluated in a Python program experiment utilizing real-world data. The algorithm will forecast the demand for several hundred inventories in the product segment of an American wholesale organization over a three-month timeline. Subsequently, a simulation was performed to compute the effect on the inventory advancement of the product segment under situations that closely resemble reality (proAIrokibul, 2024). The system will be compared and compared to a statistical baseline framework to ascertain if the possible enhancement in performance is substantial enough to justify the increased computational complexity.

3.1 Dataset

For the current research experiment, the organization supplied extensive data associated with their product segment. The data was broadly classified into three main categories, most notably, logistical process information, historical data, and product-specific data coupled with logistical properties. The historical data comprised the period from April 1, 2018, to April 31, 2020, which indicated the maximum timeline available from the company system during that time. This data was gathered from the organization's five primary warehouses situated in America, which attend to both clients and intermediate warehouses in the supply chain (proAIrokibul, 2024). The available data comprised three different datasets. The initial dataset comprises data on daily

sales and the quantity of discarded products. The second dataset concentrates on client orders for items and the corresponding degree of fulfillment. Ultimately, the third historical dataset offers details regarding past promotional activities carried out for the product.

Pre-processing

In [5]:
```
# Preprocessing
# Handling Missing Values
imputer = SimpleImputer(strategy='mean')
df[['rating']] = imputer.fit_transform(df[['rating']])
```

In [6]:
```
# Handling missing values in the 'description' column
df['description'].fillna('No description available', inplace=True)
```

In [7]:
```
# Handling missing values in the 'product' column
df['product'].fillna('Unknown', inplace=True)
```

In [8]:
```
# Normalization/Scaling
scaler = StandardScaler()
df[['sale_price', 'market_price']] = scaler.fit_transform(df[['sale_price', 'market_price']])
```

In [9]:
```
# Exploratory Data Analysis (EDA)
# Summary statistics
print("Summary Statistics:")
print(df.describe())
```

The data preprocessing entailed two main phases. Initially, the data required to be formatted in a manner that the software could read and understand. The initial data was unstructured and came in Excel and CSV files. The data was normalized by transforming it into preserved panda data frames in Python with constant feature labeling. The second segment of preprocessing was pilot data filtering. An initial assessment of the sales data via visualization identified outlier values that potentially skewed the analysis (proAIrokibul, 2024). This demonstrated the demand for some initial filtering of outliers. Subsequently, a rolling window filter was employed utilizing z-scores to point out outlier records in the sales data.

3.2 Feature Engineering and Selection

To facilitate the data for input into the Neural Network, feature engineering, and selection were imposed to affirm its suitability. In particular, to compute equidistant annotations within the time series, the data was resampled to comprise a single annotation for each business day, ranging from Monday to Friday. In the rare scenario where demand happened on Saturdays, possibly because of shifted delivery schedules due to holidays, this demand was viewed as if it had materialized on the previous Friday (proAIrokibul, 2024). This technique affirmed that the time series affirmed a constant rhythm, with observations partitioned at consistent intervals, a pivotal prerequisite for efficient analysis by the Neural Network.

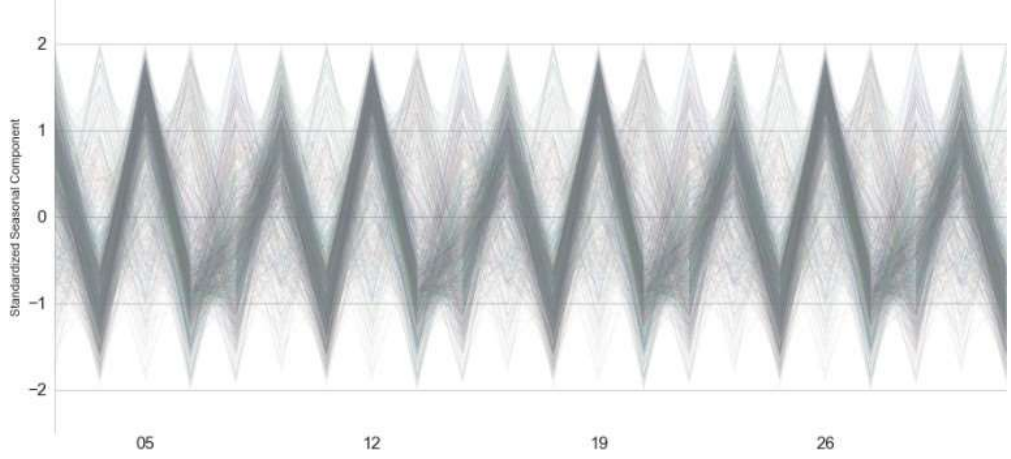


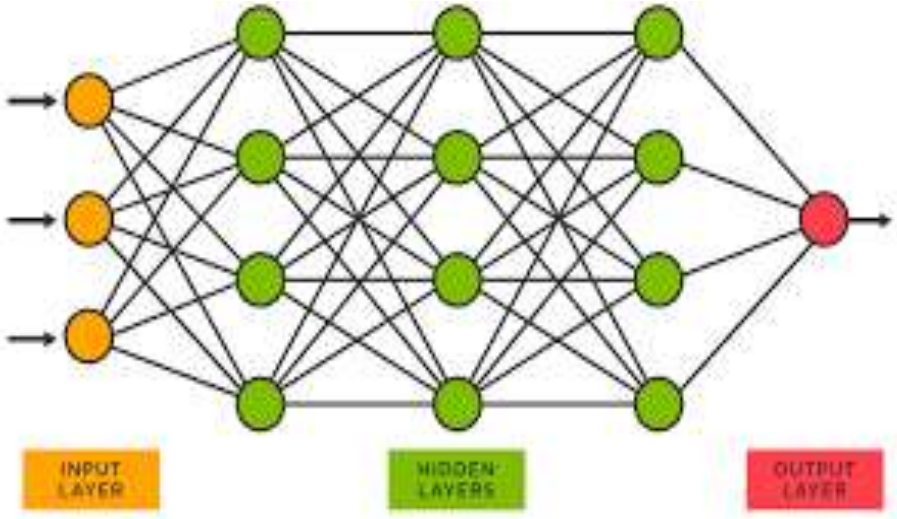
Fig 1: Showcases normalized seasonality attribute for every individual time series

For the training algorithm, the system utilized two types of features - exogenous features and engineered features. Apart from the actual product sales values, different categorical variables were also inserted into the system as supplementary predictors or covariates (proAIrokibul, 2024). The categorical attributes that were passed entailed the product element that the product belonged to, the place where the item was sold, and a distinct index comprised of the product ID and location name. This index facilitated the independent time series for every item-location combination to be observed.

Feature	Depiction	Utilized in
Categorical Data	Index, location, and product segment	Decoder & Encoder
Time Series Observation	Definite Sales per SKU per geographical location	Encoder
Date Features	Day, Month, Year, or Weekday	Decoder & Encoder
Weather Data	Average temperature, precipitation, and Sunshine hours	Decoder & Encoder
Related Item Sales	Weekly and Daily average of product segment and location	Encoder
Holiday	Sixteen binary attributes	Decoder & Encoder

Models and Metrics

Neural Networks



The neural network algorithm is likened to the functionality of the human brain, which thrives at pinpointing patterns. A basic form of the neural network is the single-layer NN, also termed a "perceptron." A Neural Network encompasses an input layer, an activation function, and an output layer. In the displayed diagram, x1...xn denotes the input attribute characteristic. Each attribute characteristic is multiplied by the equivalent weight w1...wn, which ascertains the intensity of the nodes (proAIrokibul, 2024). The bias value b is utilized to modify the activation formula, facilitating it to be shifted to lower or higher levels.

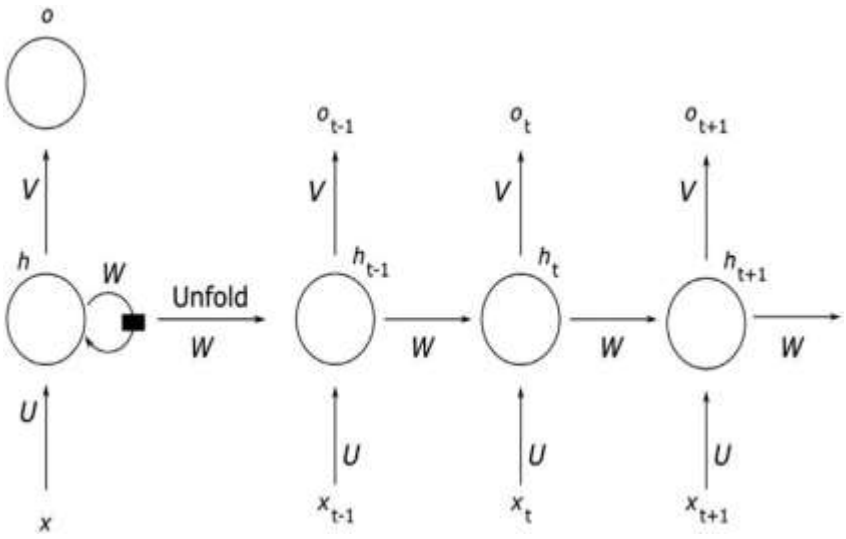
$$y = x_0w_0 + x_1w_1 + x_2w_2 + \cdots + x_nw_n + b$$

Where,

- Y is the targeted variable.
- $X.=X_{1.....}X_n$ denotes the inserted feature characteristic.
- $W_i=W_{1.....}$ is the weight

3.3 Recurrent Neural Network

Recurrent Neural Networks (RNNs) are a form of neural network that integrates a feedback loop system, enabling them to uphold information from previous inputs. These networks pass a vector via a distinct hidden layer, deemed as the hidden state (h_t), at every input phase. This hidden condition facilitates RNNs to offer a setting for interpreting inputs, hindering them from being modeled in seclusion. As a consequence, RNNs can process progressive information and manage sequences of varying lengths. A typical Recurrent Neural Network's (RNN) basic structure is demonstrated on the right-hand side of Figure 3. To showcase how data flows across sequential inputs, the network is showcased unrolled overtime on the left-hand side of the figure.



In this algorithm, time t, the input is represented as xt, while the output is ot. The hidden state ht is impacted by xt and the previous hidden state ht-1, regulated by the weight matrices W and U. The output is retrieved from the present hidden state and the weight matrix V. This association can be mathematically depicted as:

$$h_t = f_9(Ux_t + Wh_{t\&1}),$$
$$o_t = f_\alpha(Vh_t),$$

where $f\alpha$ & f_9, represent the non-linear formula as the ReLu-function or the tanh-function.

3.4 Experimentation

The performance of the algorithms was empirically assessed in a Python program experiment utilizing real-world data. The algorithm was used to forecast the demand for several hundred inventories in the product segment of an American wholesale organization over a three-month timeline. Subsequently, a simulation will be performed to compute the effect on the inventory advancement of the product segment under situations that closely resemble reality.

Importing Libraries

In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from wordcloud import WordCloud
from sklearn.preprocessing import StandardScaler
from sklearn.impute import SimpleImputer
```

In [2]:
```
df = pd.read_csv("BigBasket Products.csv")
df
```

Output:

Out[2]:

	index	product	category	sub_category	brand	sale_price	market_price	type	rating	description
0	1	Garlic Oil - Vegetarian Capsule 500 mg	Beauty & Hygiene	Hair Care	Sri Sri Ayurveda	220.00	220.0	Hair Oil & Serum	4.1	This Product contains Garlic Oil that is known...
1	2	Water Bottle - Orange	Kitchen, Garden & Pets	Storage & Accessories	Mastercook	180.00	180.0	Water & Fridge Bottles	2.3	Each product is microwave safe (without lid), ...
2	3	Brass Angle Deep - Plain, No.2	Cleaning & Household	Pooja Needs	Trm	119.00	250.0	Lamp & Lamp Oil	3.4	A perfect gift for all occasions, be it your m...
3	4	Cereal Flip Lid Container/Storage Jar - Assort...	Cleaning & Household	Bins & Bathroom Ware	Nakoda	149.00	176.0	Laundry, Storage Baskets	3.7	Multipurpose container with an attractive desi...
4	5	Creme Soft Soap - For Hands & Body	Beauty & Hygiene	Bath & Hand Wash	Nivea	162.00	162.0	Bathing Bars & Soaps	4.4	Nivea Creme Soft Soap gives your skin the best...
...	...	...	...	...	...	...	...	...	...	...
27550	27551	Wottagirl! Perfume Spray - Heaven, Classic	Beauty & Hygiene	Fragrances & Deos	Layerr	199.20	249.0	Perfume	3.9	Layerr brings you Wottagirl Classic fragrant b...
27551	27552	Rosemary	Gourmet & World Food	Cooking & Baking Needs	Puramate	67.50	75.0	Herbs, Seasonings & Rubs	4.0	Puramate rosemary is enough to transform a dis...
27552	27553	Peri-Peri Sweet Potato Chips	Gourmet & World Food	Snacks, Dry Fruits, Nuts	FabBox	200.00	200.0	Nachos & Chips	3.8	We have taken the richness of Sweet Potatoes (...
27553	27554	Green Tea - Pure Original	Beverages	Tea	Tetley	396.00	495.0	Tea Bags	4.2	Tetley Green Tea with its refreshing pure, ori...
27554	27555	United Dreams Go Far Deodorant	Beauty & Hygiene	Men's Grooming	United Colors Of Benetton	214.53	390.0	Men's Deodorants	4.5	The new mens fragrance from the United Dreams ...

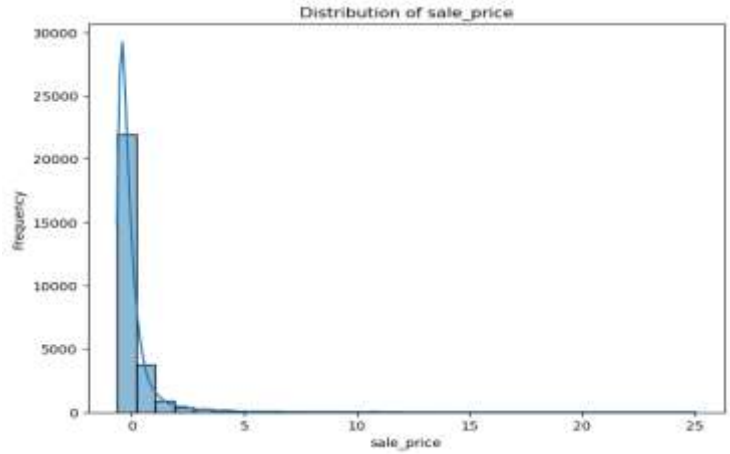
27555 rows × 10 columns

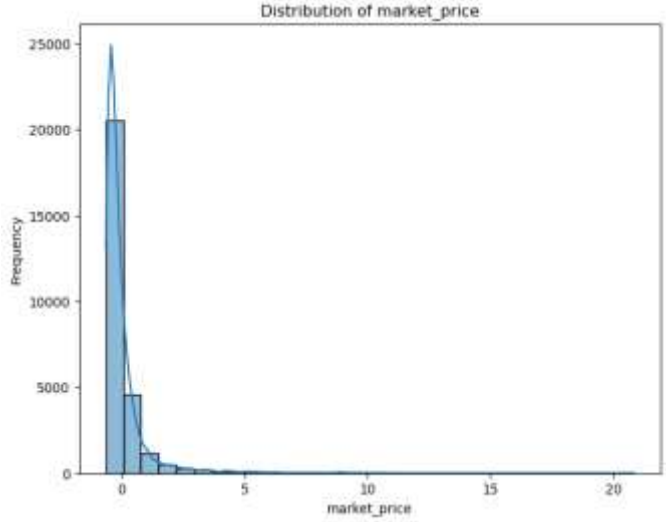
Subsequently, a code snippet was imposed to visualize the distribution of sales, the outcome can be displayed as follows:

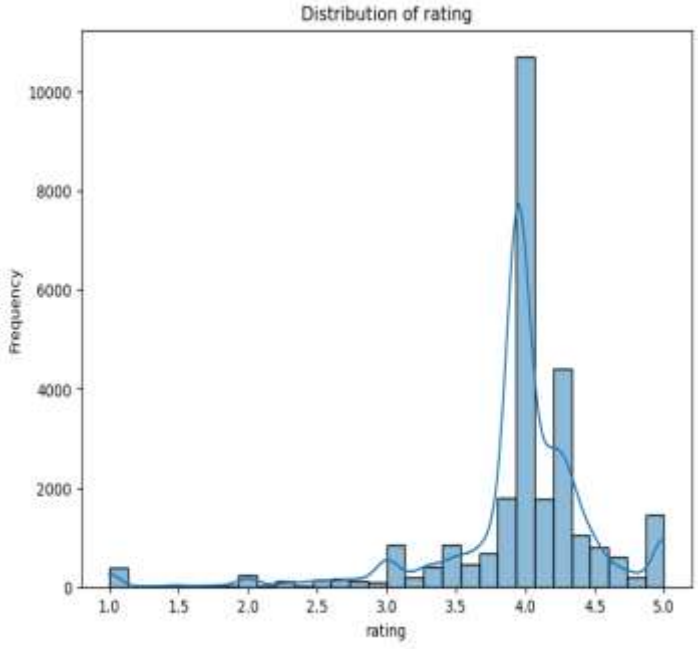
Output:

In [12]:

```
# Distribution of numerical features
numerical_features = ['sale_price', 'market_price', 'rating']
for feature in numerical_features:
    plt.figure(figsize=(8, 6))
    sns.histplot(df[feature], bins=30, kde=True)
    plt.title(f"Distribution of {feature}")
    plt.xlabel(feature)
    plt.ylabel("Frequency")
    plt.show()
```





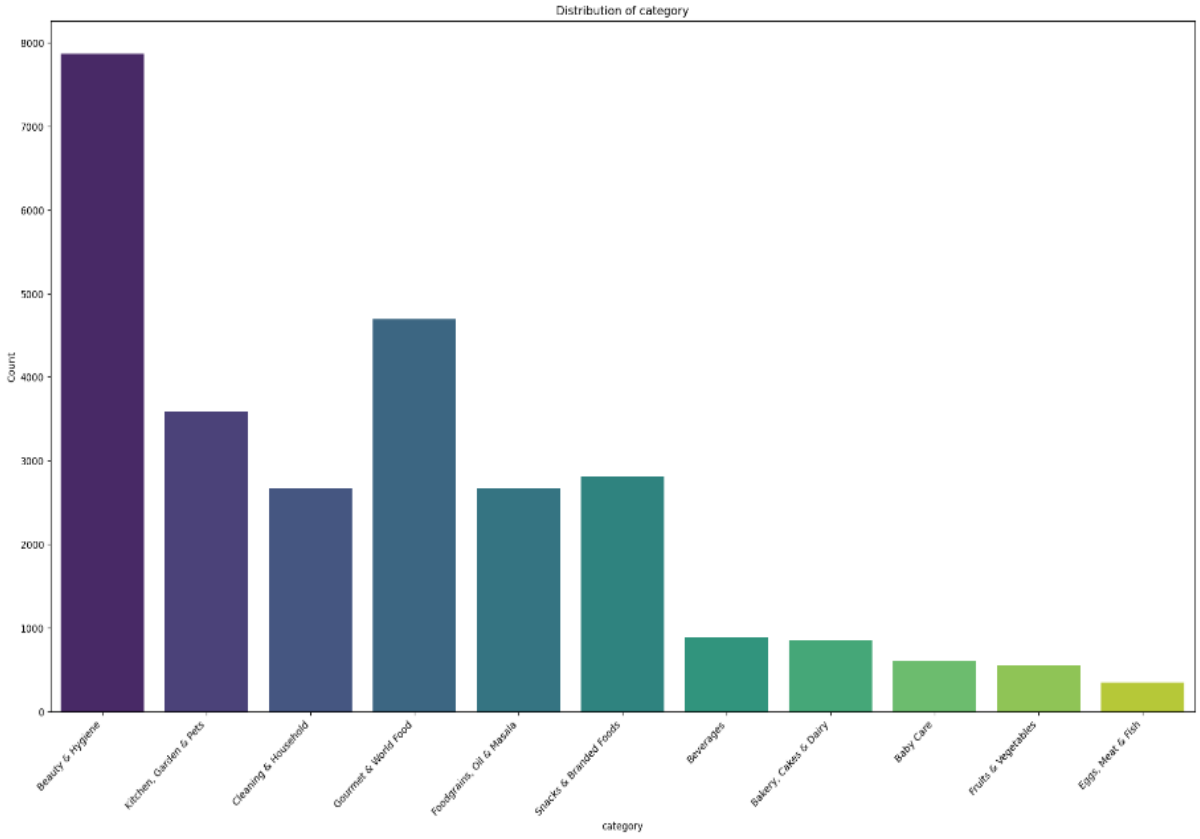


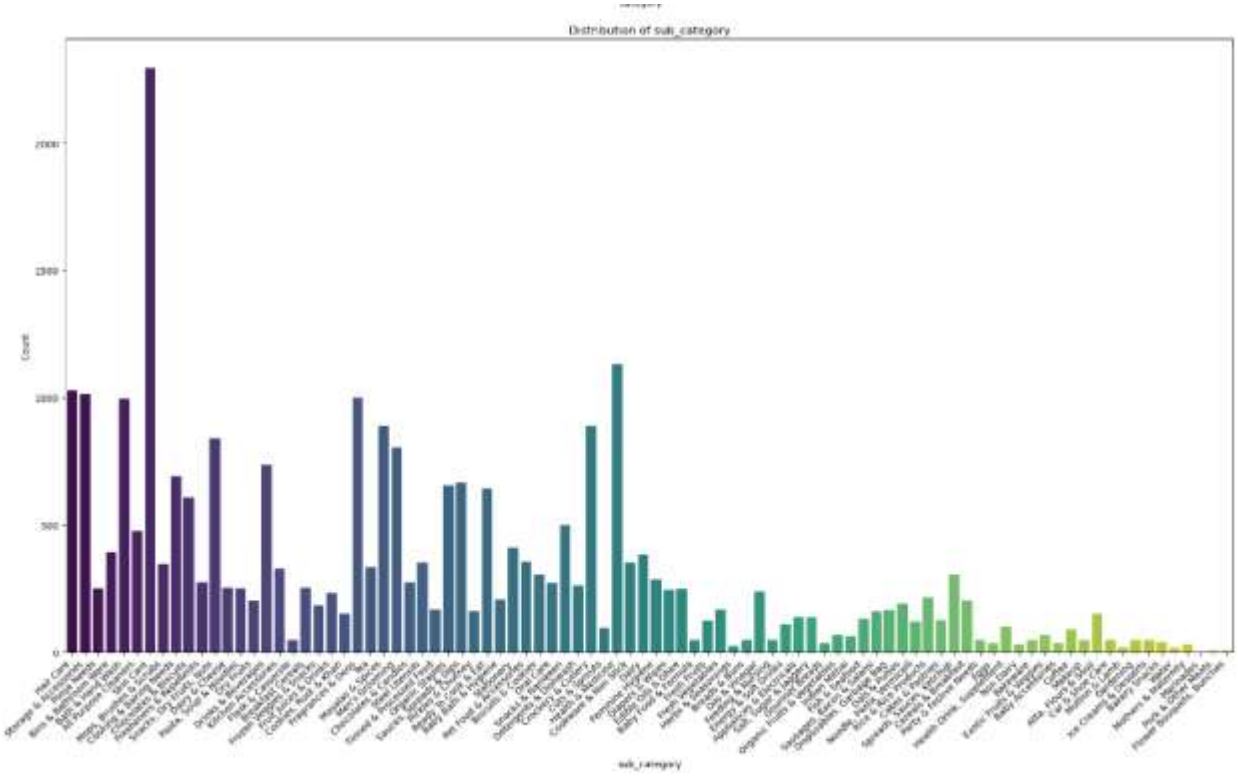
Equally important to visualize the distribution of categorical features, a code snippet was applied respectively as follows:

Output:

In [17]:

```
# Distribution of categorical features
categorical_features = ['category', 'sub_category']
for feature in categorical_features:
    plt.figure(figsize=(22, 12))
    sns.countplot(data=df, x=feature, palette='viridis')
    plt.title(f"Distribution of {feature}")
    plt.xlabel(feature)
    plt.ylabel("Count")
    plt.xticks(rotation=45, ha='right')
    plt.show()
```



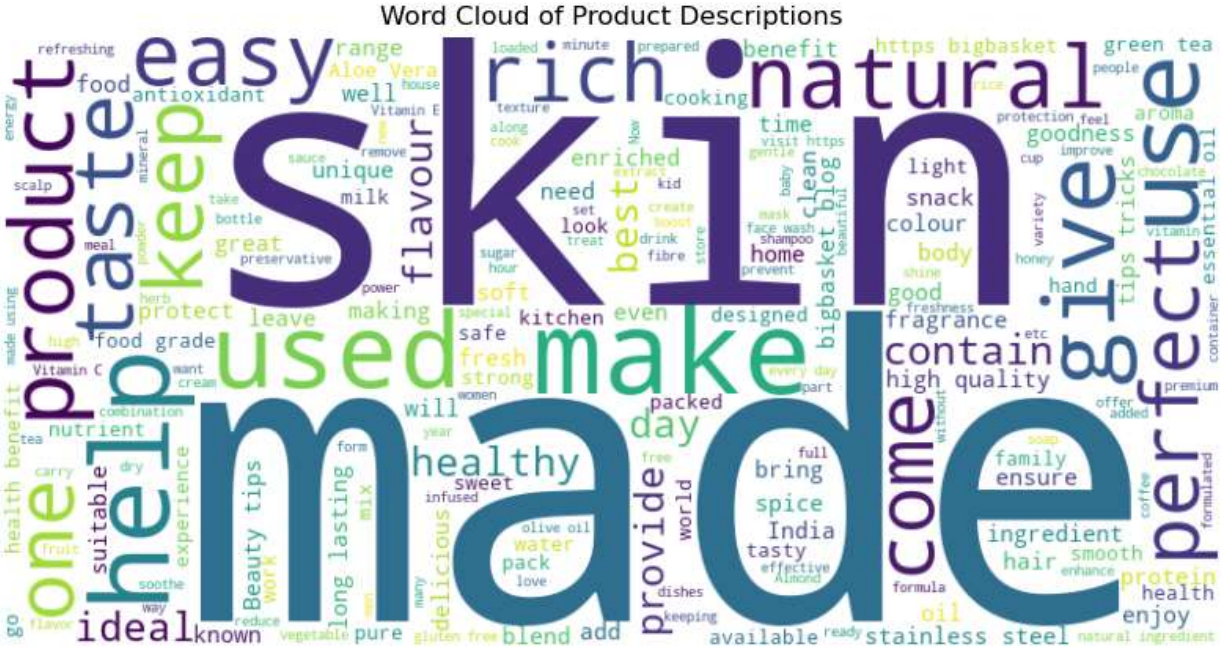


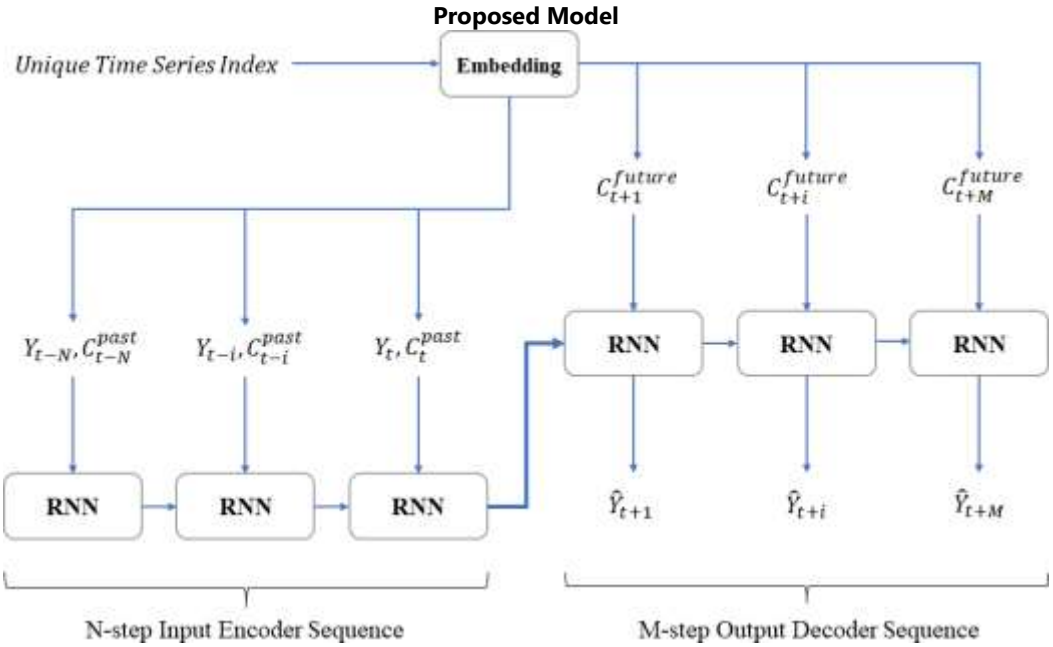
Furthermore, the analyst attempted to impose a code snippet to visualize the world cloud of product description as follows:

```
In [20]: # Analysis of 'description' column (word cloud)
text = ' '.join(df['description'].dropna())
wordcloud = WordCloud(width=800, height=400, background_color='white').generate(text)
```

```
In [21]: plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation='bilinear')
plt.title('Word Cloud of Product Descriptions')
plt.axis('off')
plt.show()
```

Output:





This study proposes the Sequence-to-Sequence (Seq2Quant) algorithm, a neural network-powered technique for demand prediction in inventory management. The Seq2Seq algorithm is a relatively novel demand predictive solution crafted to address the specific pressing issue in inventory management. The recommended model is a global modeling technique that converts multivariate input sequences of historical data and covariates into a sequence of approximated demand quantiles. Particularly, the algorithm is a sequence-to-sequence mapping that forecasts the quantiles of cumulative demand.

3.5 Models Performance Evaluations

The current experiment compared and contrasted the performance of the Neural Networks against the followingbaselines:

- Naïve Seasonal Forecast
- Moving Average Forecast
- ARIMA
- Naïve Seasonal Forecast with Averaging over four periods
- SARIMAX

4. Results

Model	MAE		SMAPE	
	Average Value	Average Rank	Average Value	Average Rank
MA	21.74 (6)	4.57 (6)	80.38 (6)	4.45 (6)
Naïve	19.97 (5)	4.54 (5)	70.44(3)	4.34 (5)
Naïve A.	18.42 (4)	3.11 (4)	69.19 (2)	3.24 (3)
ARIMAX	18.09 (2)	2.95 (2)	76.87 (4)	3.17 (2)
SARIMAX	18.33 (3)	2.73 (1)	79.68 (5)	3.60 (4)
Seq2Seq	17.44 (1)	3.08 (3)	66.91 (1)	2.21 (1)

From the table above, it was evident that the Seq2Seq had the lowest MAE (17.44) and the lowest SMAPE (66.91), suggesting that it was the best-performing algorithm overall. Besides, SARIMAX and ARIMAX also perform well, with MAE values of 18.33 and 18.09, respectively.

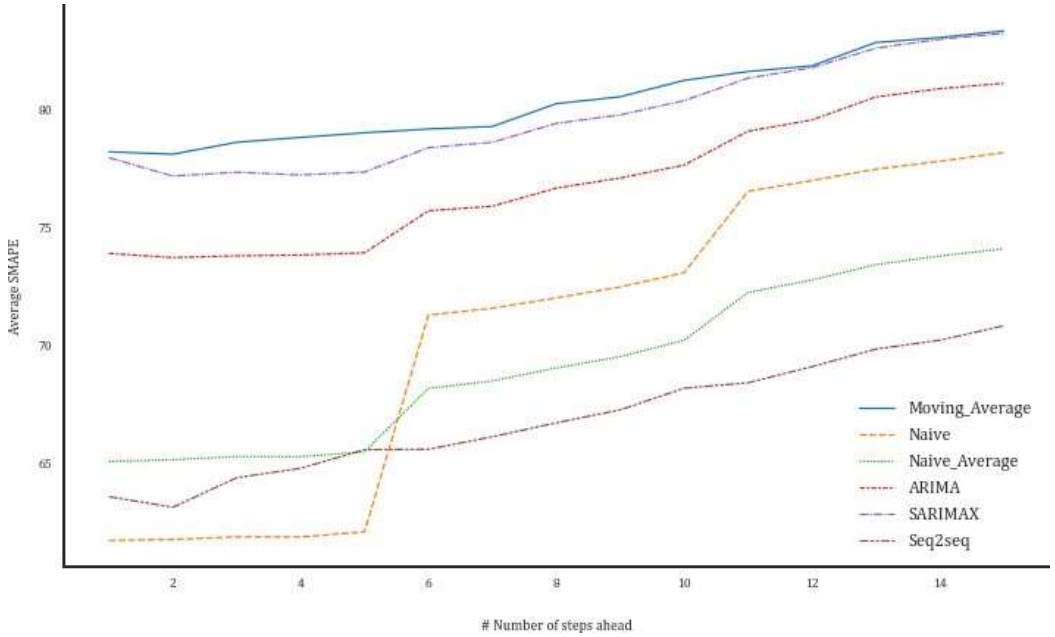


Figure 3 above exhibits the average SMAPE of all algorithms for forecasting n steps. As expected, prediction accuracy reduces with the length of the predicting horizon. The Seq2seq algorithm outperformed the others for the majority of predicting horizons. Nonetheless, for short-term predictions of less than a week ahead, the Seasonal Naïve technique has a slightly better average SMAPE.

4.1 Business Impact

4.1.1 Benefits to the Organization

1. **Enhanced Inventory Optimization:** By deploying the proposed Seq2quant algorithms to assess market trends, historical sales data, and different factors, businesses in the USA can attain precise demand predictions. As a result, this algorithm facilitates companies to optimize inventory management, reduce stockouts, and elevate overall customer satisfaction.
2. **Improved Supply Chain Visibility**: The seq2quant algorithm affords businesses in America real-time visibility on supply chain data, facilitating companies to optimize inventory levels and reorder points in multiple locations. By progressively evaluating supply chain data, the se2quant system can dynamically modify inventory levels and reorder points depending on the fluctuating demand and supply conditions, facilitating real-time optimization.
3. **Reduced Costs and Waste:** By streamlining inventory levels and enhancing demand prediction, SMEs in America can minimize the costs related to holding excess inventory. This entails obsolescence costs, storage costs, and the cost of disposing and managing waste.

4.1.2 How to Implement the Seq2Seq Algorithm

1. **Step 1: Define the problem** The problem statement for the dynamic inventory management framework should be determined using the seq2seq algorithm revolving around predicting the inventory levels for the company demand patterns.
2. **Step 2: Prepare the data-** Businesses need to prepare the data for training the seq2seq algorithm. Businesses can utilize historical data to train the algorithm.
3. **Step 3: Define the algorithm architecture-** The seq2seq algorithm comprises two main elements: decoder and encoder. The encoder obtains the input sequence and converts it into a fixed-length vector. Conversely, the decoder obtains the fixed-length vector and produces the output sequence.
4. **Step 4: Train the algorithm-** Businesses can train the algorithm utilizing a backpropagation through time (BPTT) system. Subsequently, business analysts can utilize the mean squared error (MSE) loss formula to compute the loss between the forecasted inventory levels and the real-time inventory levels.
5. **Step 5: Evaluate the model-**Business can assess the algorithm utilizing root mean squared error (RMSE) or mean absolute error (MAE) metrics. Besides, companies can also adopt visualization methods to compare the predicted inventory levels with the actual inventory levels.
6. **Step 6: Deploy the algorithm**: Businesses can go ahead to implement the algorithm in the dynamic inventory management framework to forecast the inventory levels for new demand patterns. SME can deploy the algorithm, to generate alerts when the inventory levels are low or high.

Funding: This research received no external funding.
Conflicts of Interest: The authors declare no conflict of interest.
Publisher's Note: All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers.

5. Conclusion

This research paper aimed to explore the dynamic inventory management activities employed by organizations in the USA, shedding light on the machine learning strategies that can be deployed and their implications. The performance of the algorithms was empirically evaluated in a Python program experiment utilizing real-world data. To facilitate the data for input into the Neural Network, feature engineering, and selection were imposed to affirm its suitability. This study proposes the Sequence-to-Sequence (Seq2Quant) algorithm, a neural network-powered technique for demand prediction in inventory management. The current experiment compared and contrasted the performance of the Neural Networks against the followingbaselines, most notably, Naïve Seasonal Forecast, Moving Average Forecast, ARIMA, Naïve Seasonal Forecast with Averaging over four periods, SARIMAX. From the experiment, it was evident that the Seq2Seq had the lowest MAE (17.44) and the lowest SMAPE (66.91), suggesting that it was the best-performing algorithm overall. Besides, SARIMAX and ARIMAX also performed well, with MAE values of 18.33 and 18.09, respectively.

References

[1] Chen, B., & Chao, X. (2020), Dynamic inventory control with stockout substitution and demand learning. *Management Science*, *66*(11), 5108-5127.

[2] Cuartas, C., & Aguilar, J. (2023). Hybrid algorithm based on reinforcement learning for smart inventory management. *Journal of intelligent manufacturing*, *34*(1), 123-149.

[3] Chaudhary, V., Bharadwaja, K., Meena, R. S., Bikash, P., Acharjee, D. N. C. C., & Gopinathan, R. (2023). Exploring the Use of Machine Learning in Inventory Management for Increased Profitability.

[4] Demizu, T., Fukazawa, Y., & Morita, H. (2023). Inventory management of new products in retailers using model-based deep reinforcement learning. *Expert Systems with Applications*, *229*, 120256.

[5] De Melo, Á. S. (2019). A machine learning approach to the optimization of inventory management policies.

[6] Lolli, F., Balugani, E., Ishizaka, A., Gamberini, R., Rimini, B., & Regattieri, A. (2022). Machine learning for multi-criteria inventory classification applied to intermittent demand. *Production Planning & Control*, *30*(1), 76-89.

[7] Meisheri, H., Baniwal, V., Sultana, N. N., Khadilkar, H., & Ravindran, B. (2023). Using reinforcement learning for a large variable-dimensional inventory management problem. In Adaptive learning agents workshop at AAMAS (pp. 1-9).

[8] Namir, K., & Labriji, H. (2022). Decision support tool for dynamic inventory management using machine learning, time series and combinatorial optimization. *Procedia Computer Science*, *198*, 423-428.

[9] ProAIrokibul. (2024). *Dynamic-Inventory-Management/Model/Inventory Management.ipynb at main · proAIrokibul/Dynamic-Inventory-Management*. GitHub. https://github.com/proAIrokibul/Dynamic-Inventory-Management/blob/main/Model/Inventory%20Management.ipynb