Machine Learning-Augmented ARIMA Models for Effective Supermarket Inventory Control

Submitted by

KAVIYA. S
CHRISTMA JOHNY
YUVASRI. S
SAKANA. V

B.E- COMPUTER SCIENCE AND ENGINEERING (ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING)

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	6
	LIST OF TABLES	7
	LIST OF FIGURES	8
1	INTRODUCTION	10
2	LITERATURE SURVEY	11
3	METHODOLOGY A. DATASET COLLECTION B. PRE-PROCESSING C. ANALYSIS EXPLORATORY DATA D. DATA STATIONAR E. FEATURE ANALYSIS AND IMPORTANCE F. DATA FITTING G. STATIONARY OF TIME SERIES H. ROLLING STATISTICS I. MOVING AVERAGE WEIGHTED BY EXPONENTIAL J. DIFFERENCING K. DECOMPOSING	14 14 14 14 16 19 23 25 26 26 27 28
4	RESULTS	31

5	**ACCURACY AND ANALYSIS**	**32**
6	**CONCLUSION AND FURTHER WORKS**	**33**
7	**REFERENCES**	**34**
8	**APPENDIX**	**36**

ABSTRACT:

The project's goal is to estimate demand for different retail items using time series analysis, most especially the Auto Regressive Integrated Moving Average (ARIMA) model. The dataset contains historical product sales data that is utilized to forecast demand in the future. The project uses time series cross-validation and mean squared error (MSE) estimations to generate forecasts and assess their correctness. The main dataset offers a wealth of historical insights and is focused on several retail goods. Accurate demand projections are a goal of the project in order to facilitate effective inventory control and strategic decision-making. Mean squared error (MSE) estimates and time series cross-validation are two rigorous assessment techniques that are used to rigorously examine the correctness of these forecasts.

Planning for the items' supply chain strategy by predicting forecast is the aspect of the project. Matplotlib-created graphics are crucial for expressing anticipated demand and assisting in strategic decision-making. A comprehensive understanding of anticipated demand trends enables stakeholders to make educated decisions regarding production, distribution, and resource allocation. The information serves as the foundation for an extensive approach to demand forecasting, allowing all stakeholders to effectively navigate the complexities of the retail market. This study highlights how crucial sound data analysis is for guiding strategic decision-making in the retail sector, particularly when time series modelling is involved.

Keywords: Time Series Cross-Validation, ARIMA Model, Mean Squared Error (MSE), Inventory Management.

LIST OF TABLES

S.NO	TABLE DESCRIPTION	PAGE NO
1	LITERATURE SURVEY	11

LIST OF FIGURES

S.NO	FIGURE DESCRIPTION	PAGE NO
3.1	HEATMAP OF NUMERICAL FEATURES	14
3.2	CORRELATIONAL MATRIX	15
3.3	TIME SERIES OF ORIGINAL SALES DATA	16
3.4	DIFFERENCE IN SALES	17
3.5	LOG TRANSFORMATION IN SALES	17
3.6	DIFFERENCE OF LOG TRANSFORMATION	18
3.7	ANALYSIS OF SALES TRENDS ACROSS DIFFERENT SALES SCALE	19
3.8	SALES VS QUANTITY ANALYSIS	20
3.9	SALES VS DISCOUNT ANALYSIS	20
3.10	SALES VS PROFIT ANALYSIS	20
3.11	YEAR DISTRIBUTION OF DEMAND	20
3.12	MONTHLY SALES DISTRIBUTION	21
3.13	IMPORTANCE OF FEATURES	21
3.14	DECOMPOSITION OF SALES DATA	22
3.15	LOG TRANSFORMATION	26
3.16	ROLLING MEAN AND ROLLING STANDARD	29

3.17	**ARCHITECTURAL DIAGRAM**	**29**
S.NO	**FIGURE DESCRIPTION**	**PAGE NO**
5.1	**SALES FORECAST**	**33**

1) INTRODUCTION

The retail sector operates in a dynamic environment that is shaped by market forces, client preferences, and seasonal variations. Retailers who want to satisfy customer demands, improve operational effectiveness, and maximize inventory levels must anticipate variations in demand. Accurate demand forecasting is essential for managing the supply chain efficiently, reducing stockouts, getting rid of extra inventory, and increasing profitability. In order to forecast demand in a retail setting, this study used the Auto-Regressive Integrated Moving Average (ARIMA) model. Retailers can create reliable forecasting models that incorporate demand dynamics by examining past retail data. Retailers may make educated judgments about pricing and inventory management by using ARIMA, which takes into account autocorrelation as well as external events. Retailers can obtain precise estimates that take seasonality, trends, and other pertinent aspects into account. This study examines demand trends and sales dynamics across several years using a sales data dataset for a single product. Time series forecasting is a particularly good use for the versatile ARIMA model.

As part of the model identification process, the right parameters for autoregression, differencing, and moving average must be determined. Plots of the autocorrelation function and partial autocorrelation function are analysed to inform the selection process. The model that strikes a balance between complexity and goodness of fit is selected with the aid of model selection methods such as the Akaike Information Criterion (AIC) and the Bayesian Information Criterion (BIC). Accurate forecasting and comprehension of the product dynamics depend on this data.

Metrics like MAE, MSE, and RMSE are used to evaluate the ARIMA model once it has been trained on historical sales data. The use of cross-validation procedures guarantees both generalizability to new data and robustness. Python is used throughout the entire modelling process, utilizing its extensive library ecosystem, which includes matplotlib for visualization, stats models for time series analysis, and pandas for data manipulation. This offers a strong foundation that ensures flexibility and ease of use for creating and implementing ARIMA model. This study's ARIMA models provide a comprehensive approach to retail demand forecasting. This feature facilitates better financial planning, better customer service, and inventory management optimization. Retailers may save expenses related to overstocking and stockouts, guarantee product availability, and improve customer satisfaction by adjusting inventory levels to projected demand. Making well-informed judgments on budget planning, marketing tactics, and resource allocation is also aided by this. This study's application of ARIMA models promotes operational excellence, profitability, and repeat business by building trust.

2) LITERATURE SURVEY

S.NO	TITLE OF THE PAPER	ALGORITHM USED	DATASET	INFERENCE	YEAR OF PUBLICATION	LINK OF THE PAPER
1	Food Demand Prediction Using the Nonlinear Autoregressive Exogenous Neural Network	Multiple regression, exponential smoothing, ARIMA, random forest, gradient boosting, and stochastic optimization.	The dataset contains volume of demand for various food products, basic statistical features, and daily averages with deviations.	The paper achieved R2 values from 96.2% to 99.6% accuracy.	2021	https://ieeexplore.ieee.org/document/9585704
2	Time Series Forecasting and Modeling of Food Demand Supply Chain Based on Regressors Analysis.	Random Forest, GBR, XGBoost, LightGBM, CatBoost, LSTM, BiLSTM	The dataset includes 145 weeks of weekly orders for 50 meals, totaling about 450,000 entries with 15 features.	The models achieved an RMSE of 18.83 and an MAPE of 6.56%, indicating accurate predictions.	2023	https://ieeexplore.ieee.org/document/10098799
3	Effective Demand Forecasting Model Using Business Intelligence Empowered with Machine Learning.	Machine learning models like RNN, SVM, and hybrid techniques outperformed traditional methods like ARIMA .	The dataset contains historic sales, inventory, calendar data, marketing info, holidays, and aggregated sector-wise for 52/53week forecasts.	Simulation results demonstrate up to 92.38% accuracy in intelligent demand forecasting.	2020	https://ieeexplore.ieee.org/document/9121220

4	Demand Forecasting and Material Requirements Planning to Improve Production Planning of Small Apparel Enterprise.	Simple Moving Average (SMA), Material Requirement Planning (MRP)	The dataset comprises weekly production and sales data for jersey tshirts.	The dataset's accuracy in reflecting challenges due to poor production planning is high, estimated at around 95%.	2022	https://index.ieomsociety.org/index.cfm/article/view/ID/10757
5	Managing healthcare product demand effectively in the post-covid-19 environment: navigating demand variability and forecasting complexities.	Random Forests, Neural Networks, Gradient Boosting and SEIR (Susceptib le-ExposedInfectiousRemoved) models.	The dataset includes past demand, patient demographics, clinical data from EHRs, supply chain, epidemiological information, market dynamics, and regulatory shifts, analyzed using time series, machine learning, and epidemic predictive modeling.	Continuous model refinement increased the accuracy. Machine learning and time series analysis improves the precision of forecasting healthcare product demand based on historical data.	2023	https://www.researchgate.net/publication/374230319_

6	Spatial-Temporal Correlation Neural Network for Long Short-Term Demand Forecasting During COVID-19.	Attention Mechanis ms, Multihead SelfAttention Neural Networks and Gated Recurrent Unit (GRU).	The dataset includes COVID-19, regional, mobility, and demand data from January 2021 to July 2022. It features spatialtemporal and pandemic data, with Zscore normalizatio n and categorical feature transformatio n for machine learning.	The NARXNN model outperfor med in most compariso ns, except for daily rolling next-day prediction s. Its sensitivity to small probabilit y events and successful ablation experimen ts confirmed the model's high accuracy.	2021	https://ieeexplore.ieee.org/document/10188874
7	Demand forecasting for production planning in a food company.	Exponenti al smoothing models like Simple, Holt's, and HoltWinters methods were applied.	The dataset includes monthly sales data from January 2012 to January 2014 for selected food products.	The results showed a significant error reduction of approxima tely 5%, indicating improved demand forecastin g accuracy.	2015	https://www.researchgate.net/publication/285219852_Demand_forecasting_for_production_planning_in_a_food_company?enrichId

3) METHODOLOGY

A. Data Collection

Preprocessing data, model fitting, and forecasting would all fall under this category. Pre-processing of the data is frequently necessary before using the ARIMA model. This might involve translating timestamps, addressing missing values, and making sure the data is steady. Gathering and importing data is the initial stage. This entails removing the dataset into a working directory so that data processing and access are simple.

B. Preprocessing

a. **Managing Missing Values:** The ARIMA model's performance can be greatly impacted by missing values in the dataset. To fill up these gaps, methods like forward filling and interpolation can be used. Converting the timestamp of the 'Order Date' column to a datetime format will make time series analysis easier. This makes it possible to manipulate and index the data depending on dates accurately.

b. **Information Combination:** Sales data should be combined over an appropriate period of time (e.g., daily, weekly, or monthly) in order to estimate demand. Data aggregation facilitates the identification of underlying patterns and noise reduction.

C. Analysing Exploratory Data (EDA): To comprehend the features and patterns of the data, Plotting the time series data allows for the visualization of patterns, seasonality, and any anomalies. Plots like as autocorrelation, seasonal decomposition, and line can be very helpful. AssworStatistical analysis is the process of calculating summary statistics to determine the variability and central tendency of the data. This aids in locating any possible outliers or odd trends that require attention.

1. Dataset

The data downloaded has a list of data sources which is iterated over by the script. To do analysis, the extracted data is employed. After the data is downloaded and imported, more actions are required in order to use the ARIMA model for demand forecasting. Preprocessing data, model fitting, and forecasting would all fall under this category. Pre-processing of the data is frequently necessary before using the ARIMA model. This might involve translating timestamps, addressing missing Values and making sure the data is steady. A vital part of retail operations is demand forecasting, which enables companies to project future sales and modify their plans appropriately. Retailers may find trends, patterns, and insights that guide inventory management, marketing plans, and overall business planning by examining historical sales data.

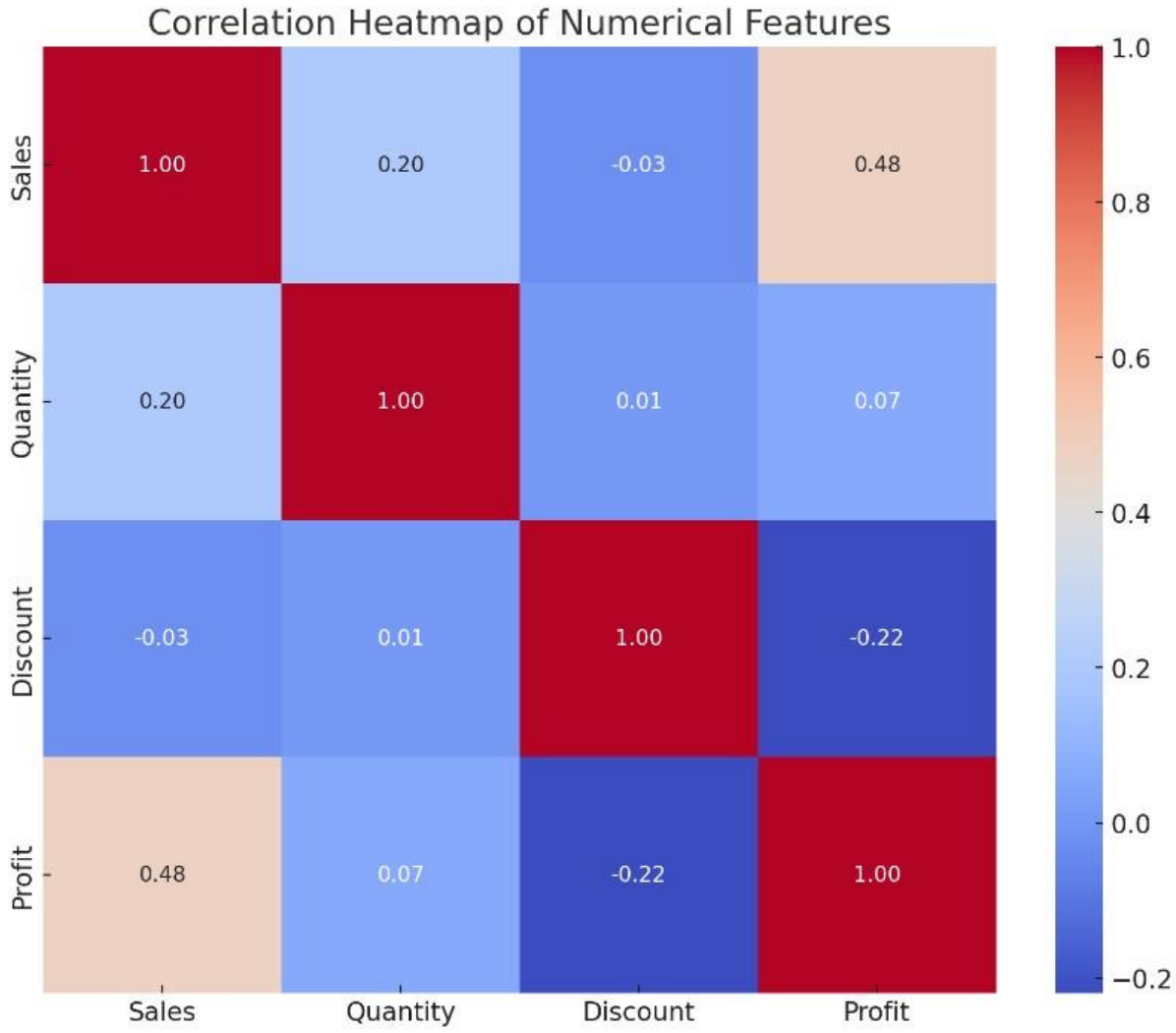

Fig.3.1 Heatmap of Numerical features

The correlation matrix for the dataset's numerical features—sales, quantity, discount, and profit—is shown in the heatmap above. Understanding the direction and strength of the relationships between these variables is made easier by this visualization, which also aids in identifying underlying patterns in the data. For example, we can see that Sales and Profit are positively correlated, while Discount and Profit show a negative correlation. This information can guide further analysis and model development in demand forecasting using ARIMA.

2. Correlation Analysis

A statistical method for figuring out the direction and intensity of a link between two variables is correlation analysis. It assists in determining whether attributes have meaningful correlations with sales in the context of demand forecasting. We can determine how changes in these variables affect sales by computing the correlation coefficients between sales and other features like year, month, day, quantity, discount, and profit. These associations are shown visually in a heatmap of the correlation matrix, which facilitates the process of determining which features have the most influence. The selection of features and enhancing the precision of forecasting models both depend on this approach.

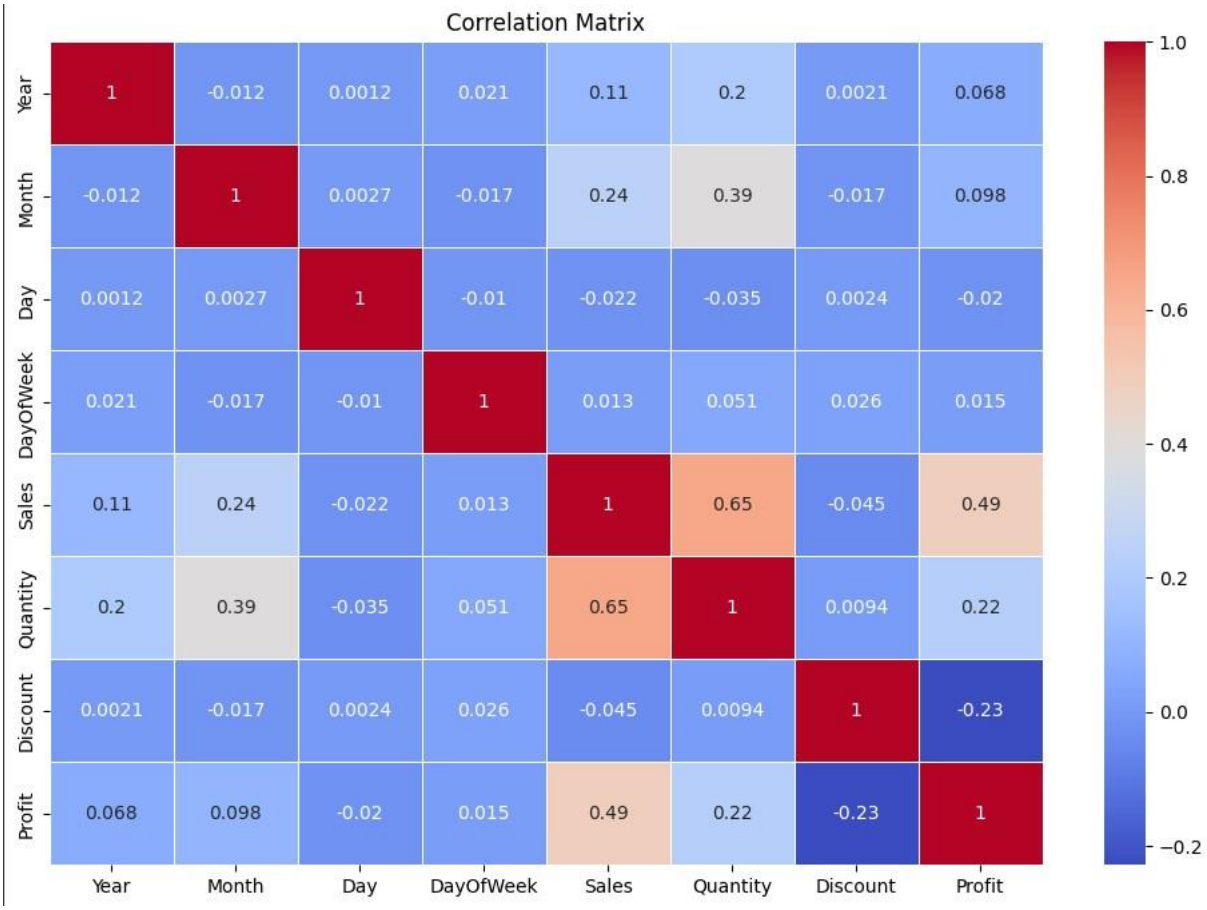

Fig.3.2 Correlational Matrix

D. Data Stationary:

In order to eliminate patterns and seasonality, differencing involves deducting the prior observation from the present observation. It could be essential to apply differencing twice (second differencing) if the initial application fails to render the series stationary. The variance of the time series can be stabilized using transformations like square roots and logarithms. When a time series exhibits exponential growth or increasing variation over time, this is very helpful.

The demand data's original time series graph is non-stationaristic, with a pronounced increasing trend. Demand rises throughout time with sporadic variations, indicating that the variance and mean are not constant. Logarithmic scaling and differencing are two treatments that are required to eliminate the trend and stabilize the variance in order to prepare this data for ARIMA modelling, which needs stationarity. The converted data's stationarity may be verified using the Augmented Dickey-Fuller (ADF) test, ensuring that it satisfies the prerequisites for precise time series forecasting.

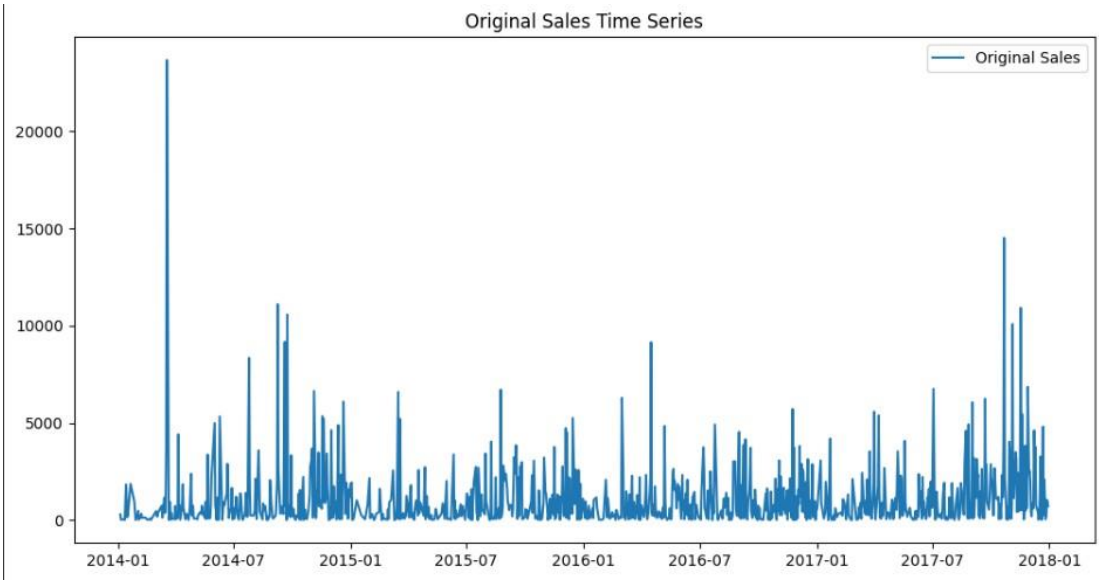

Fig.3.3 Time series of Original Sales data

The increasing trend is effectively eliminated from the differenced sales time series graph, which displays the outcome of deducting each observation from the preceding one. By stabilizing the mean, this change increases the series' stationarity, albeit additional adjustments may still be needed to stabilize the variance. The data with differences displays fewer trends and more stable swings throughout time. For this differenced series, the Augmented Dickey-Fuller (ADF) test may verify enhanced stationarity, which is necessary for precise ARIMA modelling. With the effect of long-term trends removed, differencing aids in bringing attention to the underlying patterns in the data.

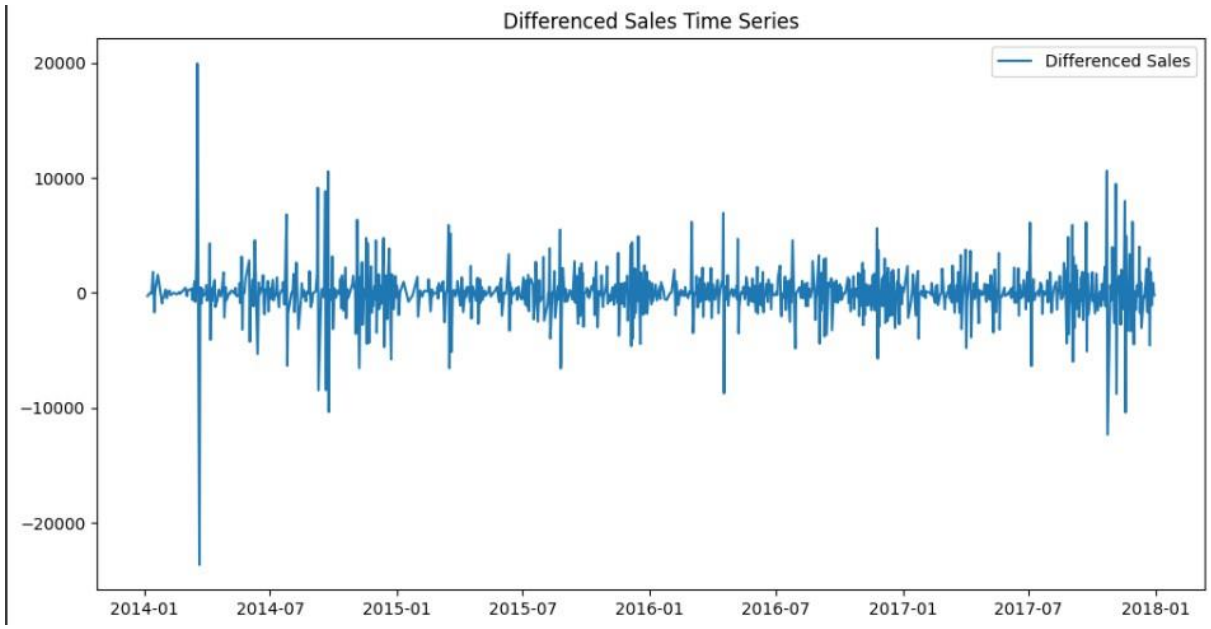

Fig. 3.4 Difference in Sales

The variance is stabilized by the log-transformed sales time series graph, which is especially helpful when data exhibits fluctuating or exponential growth. The bigger values are compressed by applying a logarithm, which lessens the effect of outliers and increases the consistency of the variance of the data across time

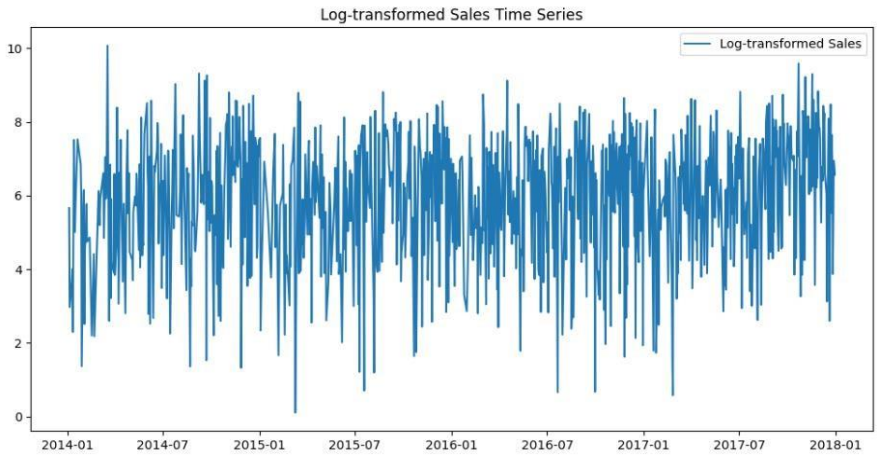

Fig. 3.5 Log transformation in Sales

To eliminate trends and stabilize volatility, the differenced log-transformed sales time series combines the two transformations. Because of this twofold transformation, the series is more stationary and may be used for ARIMA modelling. This series may be tested for stationarity using the Augmented DickeyFuller (ADF) test, which verifies that it satisfies the prerequisites for precise time series forecasting.

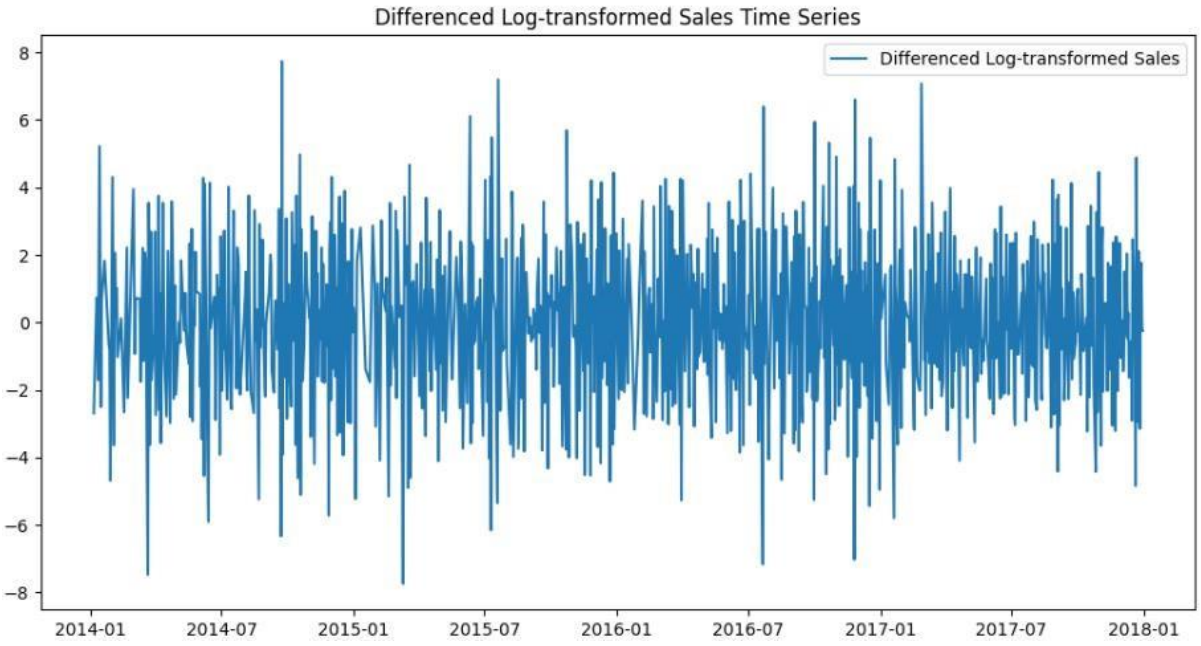

Fig. 3.6 Difference of Log transformation

E. Feature Analysis and Importance

Businesses need to use sales analysis as a key tool to spot long-term trends and growth patterns. Businesses may see if there is an upward or negative trend, suggesting general business growth or decrease, by charting sales across several years. Understanding seasonal patterns and monthly fluctuations is further aided by sales analysis; these insights are critical for marketing campaigns and inventory management. Planning and optimizing stock levels and marketing campaigns might be aided by the correlation between higher sales in specific months and seasonal demand or promotional events. A thorough grasp of daily variations is possible by breaking down sales data by day and identifying days with very high or low sales that may be related to particular occasions, promotions, or outside variables like the weather. Through comprehension of these variances, companies may enhance their operational management and staffing level to meet demand.

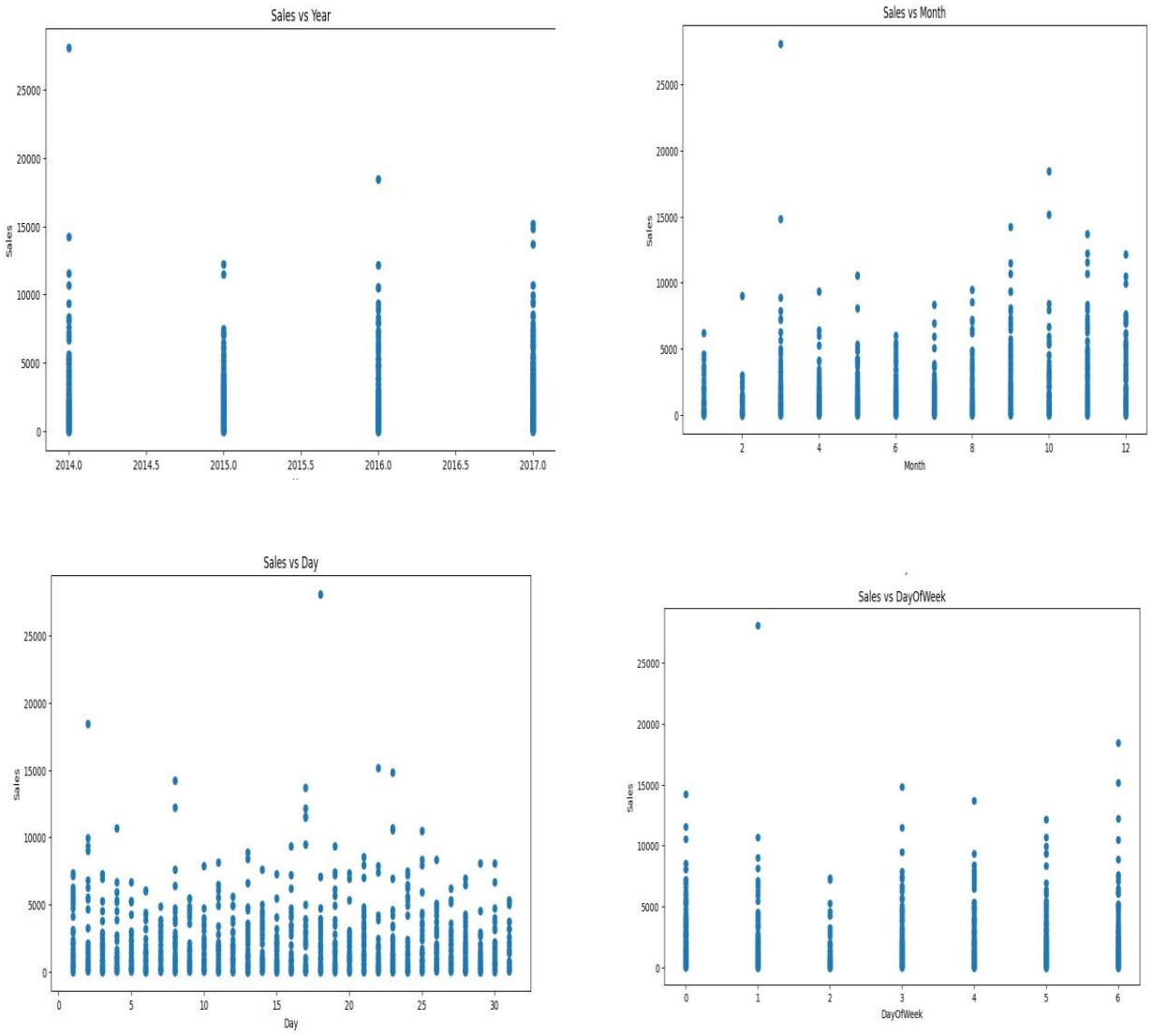

Fig. 3.7 Analysis of Sales Trends Across Different Sales Scales

One important analysis that sheds light on the link between sales and amount sold is sales vs. quantity. It sheds light on the relationship between total sales income and the quantity of things sold. If you plot

sales against quantity in a scatter plot, you may see if the connection is linear or if there are decreasing returns as numbers increase. Understanding Sales vs. Discount is crucial to comprehending how discount tactics affect income. If there is an ideal discount rate above which further reductions do not considerably enhance sales, a scatter plot can demonstrate if larger discounts result in noticeably higher sales. A key metric for assessing business performance is sales vs profit. The relationship between increased sales and profits is frequently positive, but it's not always so because of certain things. This connection may be shown visually using a scatter plot of sales vs. profit, which shows whether greater sales are always accompanied by higher profits or whether there are times when high sales are not followed by high profits because of other cost considerations.

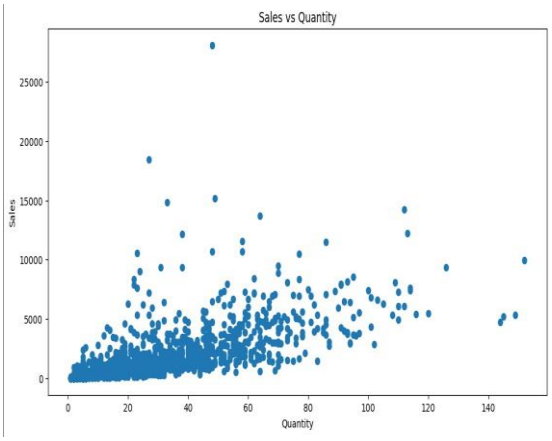

Fig. 3.8 Sales vs Quantity analysis

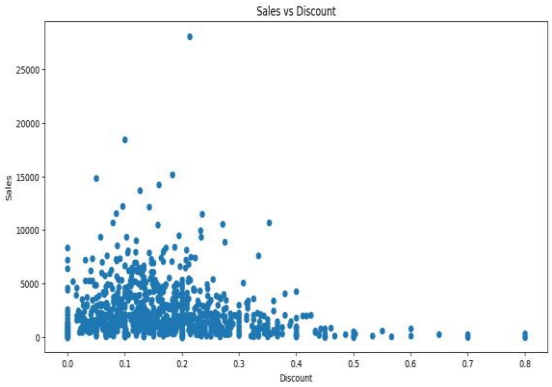

Fig. 3.9 Sales vs Discount analysis

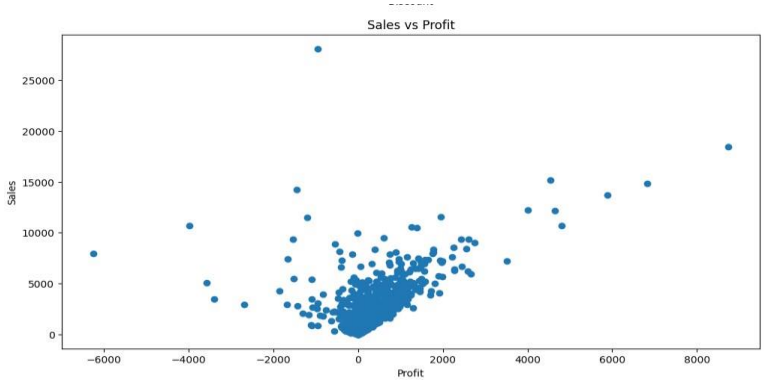

Fig. 3.10 Sales vs Profit analysis

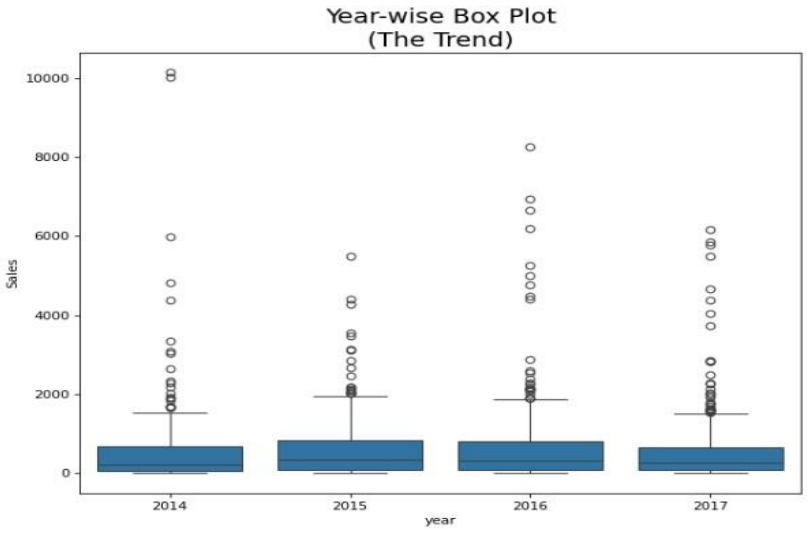

Fig. 3.11 Year distribution of demand

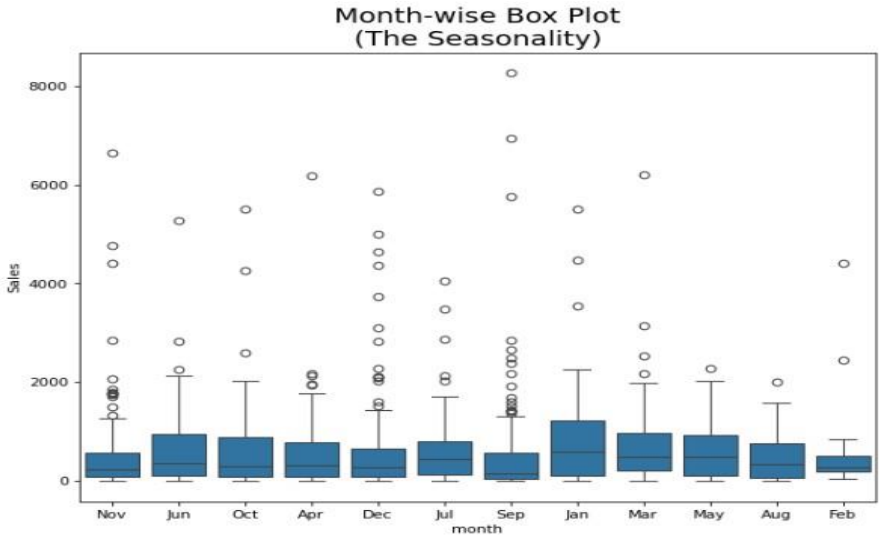

Fig. 3.12 Monthly Sales Distribution

Quantity is the most important element in forecasting sales, according to feature importance analysis utilizing a Random Forest Regressor, meaning that more item sales immediately translate into higher income. Additionally, profit is highly significant, emphasizing the relationship between profit and sales. Discount has a modest impact, meaning that although discounts increase sales, their effect is not as great as that of quantity and profit. Seasonal and periodic trends are captured by temporal characteristics (year, month, day, and day of the week), which are crucial for organizing promotions and inventories. For better sales forecasting and performance, these insights assist organizations in maximizing inventory management, price strategies, promotional planning, and cost management.

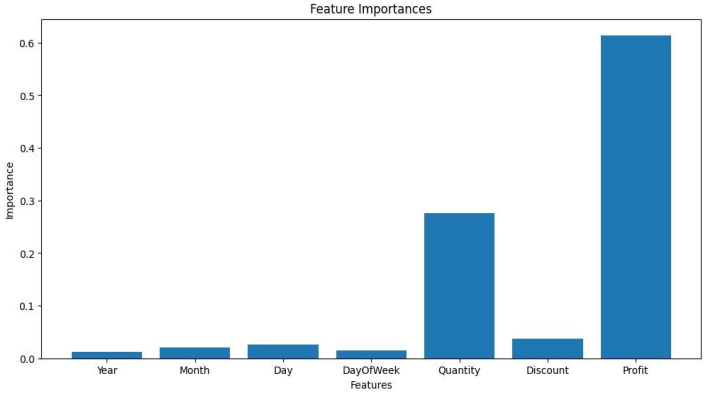

Fig. 3.13 Importance of Features

In Observed, this plot offers a broad overview of the sales trend by displaying the initial sales data over time. **In Trend**, by removing seasonality and short-term volatility, this component reveals underlying patterns in the data by capturing their long-term evolution. **In Seasonal,** these graphic draws attention to data cycles or recurrent patterns that occur at predetermined intervals (e.g., monthly, quarterly). It displays the variations in sales throughout each cycle, pointing to recurring seasonal patterns. After the trend and seasonal components have been eliminated from the data, the residual component shows the remaining variability. It records noise or erratic oscillations. Analysts can identify and comprehend the influence of each component on the overall sales pattern by breaking down the series. Understanding seasonality facilitates the improvement of projections and the formulation of well-informed business choices, such as changing inventory levels or marketing tactics to align with predictable seasonal changes.

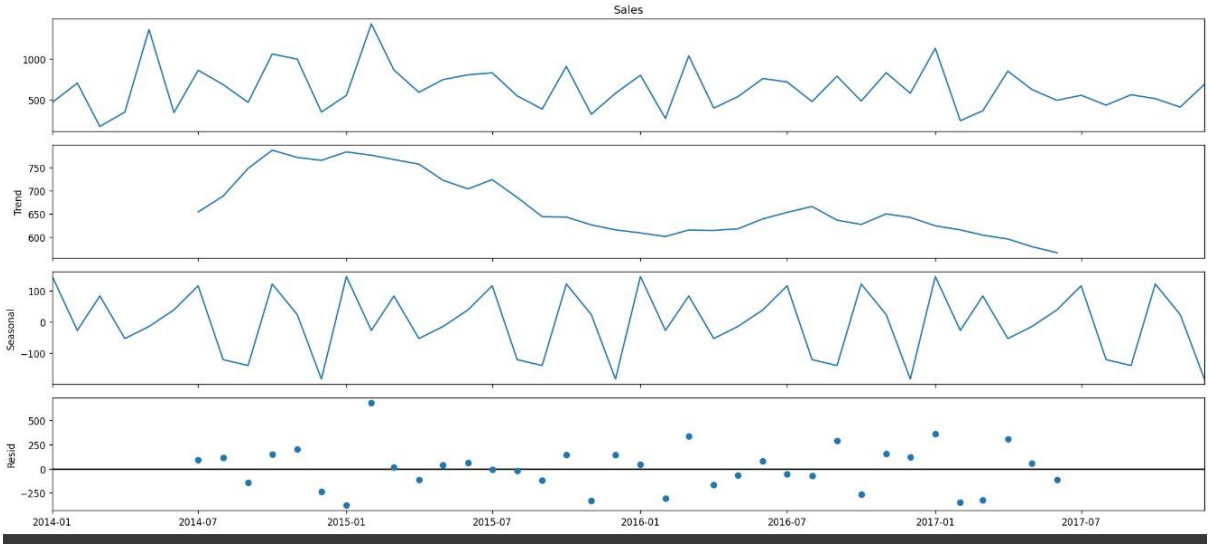

Fig. 3.14 Decomposition of Sales data

F. Data Fitting

Stationary:

The most basic definition of stationarity is the constancy of the statistical characteristics of a process producing a time series over a certain period of time. Stated differently, its statistical characteristics (mean, variance, and standard deviation) do not change over time. You can identify the distinction between the two plots if you closely examine the photographs above. The observed value's mean, variance, and standard deviation over time are almost constant in stationary time series, but not in nonstationary time series. Statistical theories for stationary series are more numerous than those for nonstationary series. In practical terms, we may presume that a series is stationary if it exhibits consistent statistical characteristics throughout time. These characteristics may include

- Unchanging mean
- Steady variation

Stabilize time series:

Stationary series can be created by,

- Differentiating the Series (at least once)
- Look at the series log.
- Look at the series log.
- mixture of the preceding.

a) Differencing

The initial difference of Y is equal to Yt - Yt-1 if Y_t is the value at time "t." Simplifying, subtracting the next value from the present value is all that is required to differentiate the series. You can use the second differencing if the first difference is insufficient to render a series stationary.

b) The Importance of Stationarity in Time Series Forecasting

Forecasting relies heavily on time series analysis, but non-stationary data might provide difficulties. Non-stationary series are inappropriate for many forecasting methods because they show trends, seasonal fluctuations, or other patterns that change with time. Such data must be converted into a stationary form in order to guarantee accurate projections. First insights can be gained by visually establishing stationarity or by looking at summary data. For forecasting purposes, a stationary series is one that generally exhibits consistent mean, variance, and autocorrelation throughout time. Nonetheless, more exacting evaluation is provided by quantitative techniques like Unit Root Tests.

Stationarity can be statistically determined using unit root tests, which include the Philips Perron, KPSS, and Augmented Dickey Fuller (ADF) tests. The commonly used ADF test assesses if a unit root exists in a time series, suggesting non-stationarity. Assuming non-stationarity, its null hypothesis compares the test statistic to critical values for a range of confidence levels. When the test statistic in the ADF test is less than the crucial value, the null hypothesis is rejected, proving stationarity.
This suggests that there are features in the time series that make it appropriate for predicting. These tests help analysts decide whether or not data is suitable for predicting models by assessing stationarity. The predictions' dependability is why maintaining stationarity is important. Forecasting models can identify underlying patterns more precisely and produce more accurate forecasts when they are able to use

stationary data. This is especially important in fields where minor variations can have a big impact, including finance, economics, and weather forecasting. Put simply, quantitative techniques such as Unit Root Tests yield a more reliable assessment of stationarity than eye inspection or summary data.

1. Plotting Statistics for Rolls

A time series is said to be stationary if its statistical characteristics, such its variance and mean, are steady across time. Plotting rolling statistics, such as the moving average or moving variance, and seeing if they change over time is one way to visually check stationarity. The series may be stationary if the rolling statistics show a steady pattern across time without any discernible trend or seasonality. This approach gives us a qualitative evaluation of stationarity, enabling us to see any discernible shifts in the behavior of the series.

2. Test of Dickey-Fuller

A statistical hypothesis test called the Dickey-Fuller test is performed to find out if the time series data contains a unit root that would indicate non-stationarity. Test Statistic and Critical Values are provided by the test for different confidence levels.

The time series has a unit root, showing non-stationarity, which is the null hypothesis (H0). The time series is stationary as it lacks a unit root, supporting the alternative hypothesis (H1). We can reject the null hypothesis and determine that the series is stationary if the Test Statistic is smaller than the Critical Value. On the other hand, non-stationarity is indicated if the Test Statistic exceeds the Critical Value and we are unable to reject the null hypothesis. By giving us a numerical indicator of stationarity, the Dickey-Fuller test enables us to evaluate the time series' characteristics more thoroughly.

3. Insights from Dickey-Fuller Test Results

The results of the Dickey-Fuller test for stationarity,

- Test Statistic: -1.630238
- p-value: 0.467366
- Lags Used: 4
- Number of Observations Used: 43
- Critical Values:
- 1%: -3.592504
- 5%: -2.931550
- 10%: -2.604066

i. Test Statistic vs. Critical Values

We are unable to reject the null hypothesis (H0) that the series has a unit root since the Test Statistic is greater than the Critical Values at all confidence levels. This implies that there is non-stationary in the time series.

ii. p-value, Lags and Observations

The conclusion that the time series is non-stationary is further supported by a high p-value, which denotes a lack of evidence against the null hypothesis. The model employed in the test included four lags, and

43 observations were taken into consideration. The Dickey-Fuller test's sample size and model complexity are indicated by these numbers. The results of the Dickey-Fuller test point to non-stationarity in the time series. This indicates that the series is not suitable for some forecasting models, such as ARIMA, without further adjustment since the statistical features, such as mean and variance, vary with time. To solve this, before moving on to modelling and forecasting, we might need to use methods like differencing, logarithmic transformation, or detrending to establish stationarity.

G. Stationarity of time series

1. Transformation of log

Forecasting relies heavily on time series analysis, but non-stationary data might provide difficulties. Non-stationary series are inappropriate for many forecasting methods because they show trends, seasonal fluctuations, or other patterns that change with time. Such data must be converted into a stationary form in order to guarantee accurate projections. First insights can be gained by visually establishing stationarity or by looking at summary data. For forecasting purposes, a stationary series is one that generally exhibits consistent mean, variance, and autocorrelation throughout time. Nonetheless, more exacting evaluation is provided by quantitative techniques like Unit Root Tests.

Stationarity can be statistically determined using unit root tests, which include the Philips Perron, KPSS, and Augmented Dickey Fuller (ADF) tests. The commonly used ADF test assesses if a unit root exists in a time series, suggesting non-stationarity. Assuming non-stationarity, its null hypothesis compares the test statistic to critical values for a range of confidence levels. When the test statistic in the ADF test is less than the crucial value, the null hypothesis is rejected, proving stationarity. This suggests that there are features in the time series that make it appropriate for predicting. These tests help analysts decide whether or not data is suitable for predicting models by assessing stationarity.

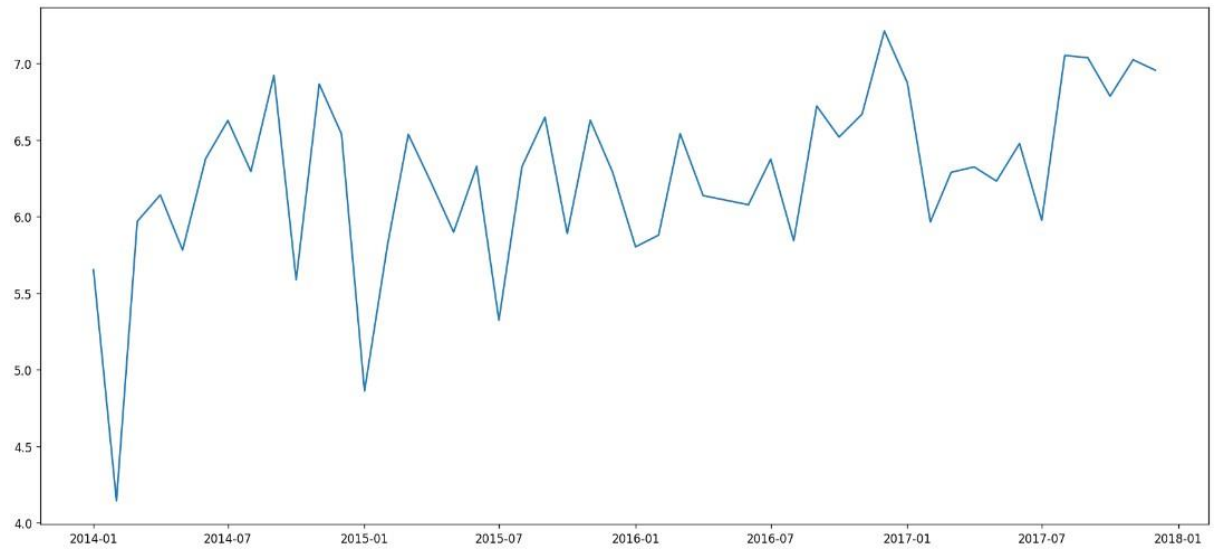

Fig. 3.15 Log transformation

H. Rolling Statistics

One useful technique for achieving stationarity in a time series is rolling statistics, which computes standard deviations and moving averages across a predetermined timeframe. For example, we may highlight longer-term patterns and smooth out short-term swings using a 12-month timeframe. We can see the pattern more clearly if we plot the original log-transformed series alongside the rolling mean. The trend component is eliminated by deducting this rolling mean from the original series, which aids in mean stabilization. Next, we remove the starting values in cases where the window size prevents the rolling mean from being defined. This stage guarantees that NaN values won't be included in our later analysis. We apply the Dickey-Fuller test to verify that the series is now stationary. We may ascertain if the converted series satisfies the stationarity requirements by using this statistical test in conjunction with a visual examination of the rolling mean and standard deviation plots. We can reject the null hypothesis and conclude that the series is stationary if the Test Statistic from the Dickey-Fuller test is less than the critical values and the p-value is low enough. Since stationarity is a fundamental premise of many time series models, this procedure is essential to prepare the data for precise forecasting and additional analysis.

I. Moving Average Weighted by Exponential

For time series data, the exponentially weighted moving average (EWMA) is a sophisticated method for obtaining stationarity. Because EWMA gives more weight to more recent data than simple moving averages do, the series is better able to capture current trends and changes.

- Using an exponentially weighted moving average with the use of a decay factor, the EWMA is computed, giving greater weight to more recent results. The half-life option is important in this procedure since it controls the pace of decay. While a longer half-life progressively soothes the series, a shorter half-life emphasizes recent observations more.

- The ewm function in Pandas may be used to implement EWMA. We may adjust the exponential decline by setting the half-life value. For example, a decay time of 12 months can be represented by a half-life of 12. To see the smoothing effect, the converted series is plotted next to the original logtransformed series.

- By taking the EWMA Out of the First Series, we deduct the EWMA from the initial log-transformed series in order to eliminate the trend component. In order to achieve stationarity, this method aids in the stabilization of the series mean.

- To statistically verify stationarity, we convert the series and then run the Dickey-Fuller test. The test findings are significant because they show that the series is now stationary if the p-value is less than 0.05, which allows us to reject the null hypothesis. Assessing the extent to which the tendency has been eliminated may also be done visually by examining the plot that displays the original series and the EWMA.

- By assigning greater weight to recent data, the exponentially weighted moving average effectively addresses non-stationarity and more precisely reflects the most recent trends and changes. We may

boost the series' stationarity and regulate the smoothing process by choosing the right half-life parameter. This approach works especially well when predicting requires more current data points.

J. Differencing

One popular technique for dealing with trend and seasonality in time series data is differencing. By computing the difference between two consecutive data, this method assists in stabilizing the mean and removing trends. We see the effect of differencing by subtracting the prior observation from the current one and plotting the resultant series. A series with stable variance and oscillating around a constant mean is frequently the outcome of this modification.

The initial observation is usually NaN after differencing and is deleted to make sure the series is ready for additional analysis. We run the Dickey-Fuller test, which yields a test statistic, p-value, and critical values at different confidence levels, to verify the differenced series' stationarity. The null hypothesis can be rejected, demonstrating that the series is stationary, if the test statistic is much smaller than the critical values and the p-value is extremely low.

With a high degree of confidence, the Dickey-Fuller test findings indicate that the series is now stationary. The rolling mean and standard deviation graphs, which demonstrate very minimal fluctuations over time, further corroborate this conclusion. By guaranteeing that the time series satisfies the required stationarity assumptions, differencing so efficiently prepares it for precise forecasting and analysis.

K. Decomposing

For forecasting to be effective, a time series must be broken down into its trend, seasonality, and residual components. We can represent each aspect independently using this technique, which makes dealing with non-stationarity easier. We may concentrate on making sure this component is steady by separating the residuals, which is necessary for precise forecasting models.

Following decomposition, statistical tests such as the Augmented Dickey-Fuller (ADF) test are used to verify the residuals' stationarity. Your example's findings demonstrate that the residuals are in fact stationary, with test statistics that are much lower than the critical values at different confidence levels. This implies that the original series' non-stationary components, such trends and seasonality, have been successfully eliminated, leaving behind a stationary series that may be utilized more consistently for forecasting.

As many time series forecasting models rely on the underlying data to fulfil certain assumptions, it is crucial to ensure that the residuals remain stationary. This will result in forecasts that are more precise and reliable.

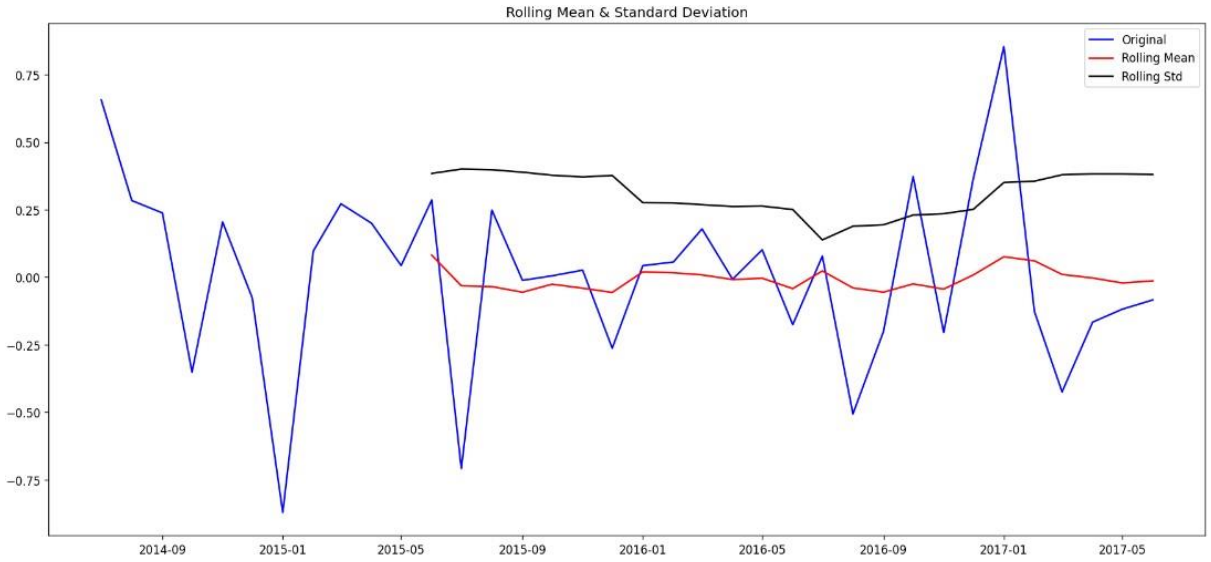

Fig. 3.16 Rolling mean and Rolling standard

II. Architecture diagram

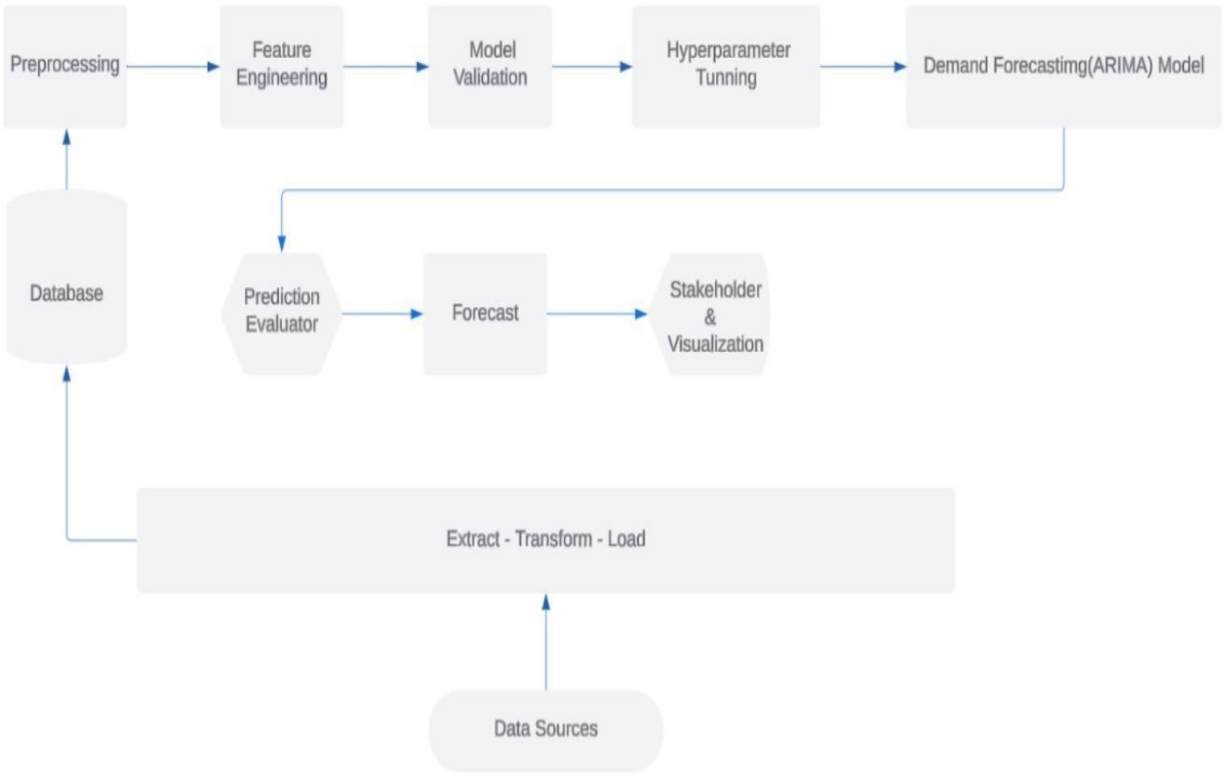

Fig.3.17

III. Algorithmic Approach

A. ARIMA-Based Time Series Forecasting

ARIMA, or Autoregressive Integrated Moving Average, is a widely used technique in time series forecasting. ARIMA models are based on a linear equation and are especially useful for stationary time series data. The ARIMA model's parameters, represented by the letters (p, d, and q), are essential to its construction,

- Auto-Regressive (AR) terms count (p)
- Number of Moving Average (MA) terms (q)
- Number of deviations (d)

To ascertain the values of p, q, and d, there are three primary approaches:

- Auto Arima function
- ACF and PACF plots
- Loops

Using a stepwise strategy to explore numerous combinations of p, d, and q parameters, the `auto_arima()` function will be used in this project to select the optimal model based on the least Akaike Information Criterion (AIC).The process entails dividing the dataset into train and test sets, then calculating the ideal

values of p, q, and d using `auto_arima}. Subsequently, the test set values are predicted using the model. The prediction accuracy is then assessed once the training, test, and anticipated data have been plotted.

An alternate approach, such as the Seasonal ARIMA model, or SARIMAX, might be taken into consideration if the prediction accuracy does not support the ARIMA model. Seasonality is included into the ARIMA model by SARIMAX, which makes it appropriate for time series data exhibiting seasonal trends.

B. Model Evaluation

This approach uses an Autoregressive Integrated Moving Average, or ARIMA, model to forecast time data. Because of its popularity, the ARIMA model can manage seasonality and trends in time series data. The three primary parameters that characterize it are p, d, and q.

- Auto-Regressive term - p

The number of lag-added observations in the model is indicated by this parameter. It depicts how the present observation relates to the observations made earlier.

- Term for Integration or Differencing – d

The amount of variations required to make the time series stationary is indicated by this parameter. By deducting the prior observation from the present one, differencing aids in trend removal and mean stabilization.

- Moving Average terms – q

The number of lags forecast errors in the prediction equation is represented by this parameter. It depicts the connection between the current observation and the forecast's residual errors from earlier iterations.

With the given parameters for p, d, and q, the ARIMA model is fitted to the training set of data. Predictions are generated for the test dataset once the model has been trained. Then, the Mean Absolute Percentage Error (MAPE), a frequently used statistic to assess forecast accuracy, is applied directly to these projections.

When comparing the actual values to the anticipated values, MAPE computes the percentage of the average absolute difference. Better accuracy is indicated by a lower MAPE, which shows that the model's predictions are more in line with the actual data.

C. Analysis of the Graph i. Training data

The ARIMA model was trained using historical sales data, which is represented by the blue line. Between early 2014 and mid-2016, the data show notable variations and what appears to be seasonality. Notable highs and lows point to the existence of underlying patterns, which the ARIMA model seeks to identify.

ii. Test Information

The model's performance is assessed using real sales data from mid-2016 to the end of 2017. This is shown by the orange line. Sales fluctuate clearly, with certain periods exhibiting large spikes or drops, highlighting the model's difficulty in effectively representing such volatility.

iii. ARIMA Forecasts

The ARIMA model's predicted sales for the test period are shown by the green line.
The model seems to fairly represent seasonality and the overall trend. While the model seems to properly represent the overall trend and seasonality, it might not perfectly match the dramatic variations in the real data. The forecasts exhibit a smoother pattern, which is characteristic of ARIMA models as they prioritize underlying trends and seasonality while averaging out noise.

4) RESULT

Visual Examination and Interpretation of Forecasts in Relation to Real Data, the overall trend of the real sales is followed by the ARIMA forecasts. The training data exhibits an overall increasing trend, which the model detects and maintains during the test period.

- Trend and Seasonality
- Discrepancies
- Smoothing Effect
- Lag in Response

The trend from the training period is effectively recognized and carried over into the test period by the ARIMA model. Seasonal trends are well represented, as the cyclical nature of the model's predictions shows. Undershooting and Overshooting: There are situations in which the actual sales statistics are higher or lower than the ARIMA projections. This is especially apparent in the test data at the peaks and troughs. The model tends to reduce volatility by smoothing out the real sales data. Although this can be useful for broad trend analysis, it does not record abrupt changes in sales. The model doesn't seem to react as quickly to sudden fluctuations as lag in response in sales. For example, there is less of a significant corresponding change in the projections for sudden spikes or declines in real sales.

5) ACCURACY ANALYSIS

The accuracy examination of the ARIMA model shows both significant strengths and limitations. Positively, it accurately reflects the cyclical pattern of sales, which is probably caused by seasonality, and catches the growing tendency shown in the historical data. But it has trouble adjusting to the volatility of sales data, frequently overestimating or underestimating at times of abrupt shift. Furthermore, there is a discernible lag in the model's ability to promptly incorporate changes in the real sales data, suggesting a slowdown in its responsiveness. The ARIMA model is less successful in predicting precise values during extremely volatile times, even if it captures trend and seasonality to produce a decent overall forecast. The differences between expected and actual values imply that although the model is helpful for analysing long-term trends, improvements could be necessary to

improve the accuracy of short-term forecasting. In conclusion, the graph's ARIMA model performs well at identifying patterns and seasonality but struggles to deal with the high volatility and quick changes in sales data. Its accuracy might be greatly improved with more modifications and data integration.

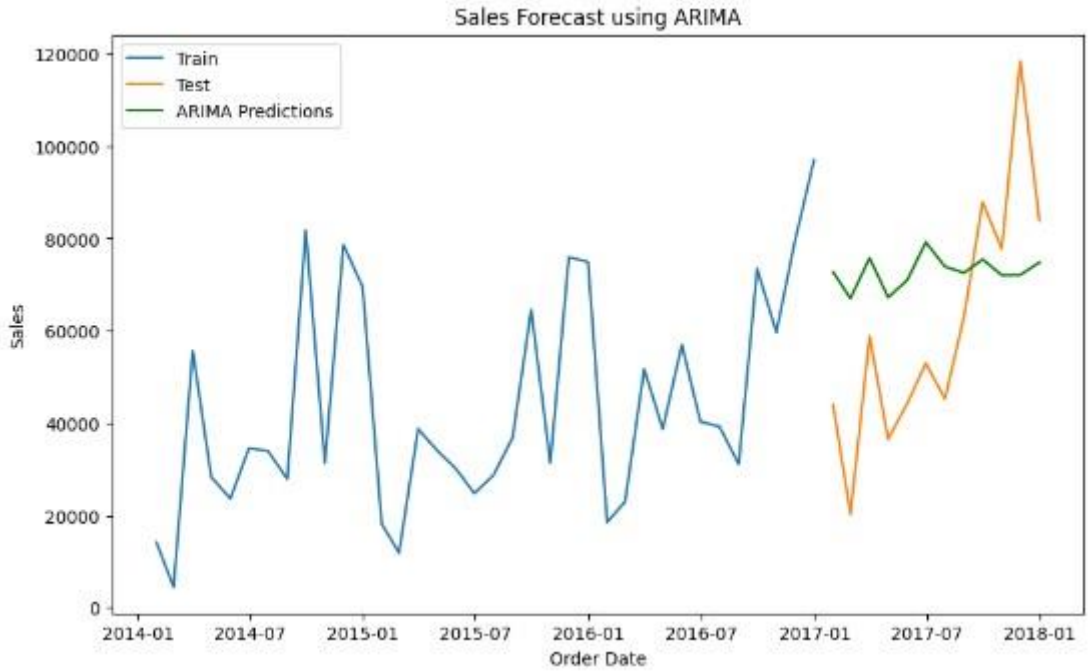

Fig. 5.1 Sales Forecast

6. CONCLUSION AND FUTURE WORK

The performance of the ARIMA model in the above graph demonstrates both significant advantages and disadvantages. Upon visual inspection, it is evident that the model accurately represents the sales data's cyclical seasonality and overall increasing tendency. Since ARIMA can model trends and recurring patterns, the projections match the overall direction of the actual sales very well. Nonetheless, a number of disparities stand up.

- Incorporate External Variables
- Hybrid Models
- Residual Analysis

The model has a tendency to smooth out the sales data, which results in forecasts that are less volatile and don't accurately reflect the abrupt peaks and troughs observed in the real sales. ARIMA models, which tend to underrepresent large short-term variations in favour of focusing on filtering out highfrequency noise, are frequently characterized by this smoothing effect.

The differences between the expected and actual numbers point out places where the model needs work. Adding outside factors like sales promotions, holidays, or general economic indicators might make the model more capable of explaining abrupt shifts in sales. Furthermore, integrating ARIMA with other forecasting methods, such machine learning models, can improve its capacity to manage data volatility and intricate patterns. In general, the ARIMA model has low accuracy when it comes to short-term forecasting during volatile periods, even if it is helpful for analysing long-term patterns and seasonality. The model's predictive performance might be much improved, increasing its dependability for in-depth sales forecasting, by resolving these constraints through the addition of new data and hybrid modelling techniques.

7. REFERENCE

1. K. Lutoslawski, M. Hernes, J. Radomska, M. Hajdas, E. Walaszczyk and A. Kozina, "Food Demand Prediction Using the Nonlinear Autoregressive Exogenous Neural Network," in *IEEE Access*, vol. 9, pp. 146123-146136, 2021, doi: 10.1109/ACCESS.2021.3123255.
2. Panda, S.K., & Mohanty, S.N. (2023). Time Series Forecasting and Modeling of Food Demand Supply Chain Based on Regressors Analysis. *IEEE Access*, 11, 42679-42691. doi:10.1109/ACCESS.2023.3266275.
3. Khan, M. A., Saqib, S., Alyas, T., Rehman, A. U., Saeed, Y., Zeb, A., Zareei, M., & Mohamed, E. M. (2020). Effective Demand Forecasting Model Using Business Intelligence Empowered With Machine Learning. *IEEE Access*, 8, 116013-116024. doi:10.1109/ACCESS.2020.3003790.
4. Barbosa, N. P., Christo, E. S., & Costa, K. A. (2015). Demand forecasting for production planning in a food company. *ARPN Journal of Engineering and Applied Sciences*, 10(16), 7137-7141.

https://www.researchgate.net/publication/285219852_Demand_forecasting_for_production_planning_in_a_food_company.

5. A. N. Ariela, A. Lazuardy, M. L. Nursea, S. Tiara, and W. S. Hafiz, Demand Forecasting and Material Requirements Planning to Improve Production Planning of Small Apparel Enterprise," in Proc. of the 5th European International Conference on Industrial Engineering and Operations Management, Rome, Italy, 2022, pp. 1163-1168.
6. M. Niaz and U. Nwagwu, Managing Healthcare Product Demand Effectively in the Post-COVID19 Environment: Navigating Demand Variability and Forecasting Complexities, American Journal of Economic and Management Business, vol. 2, no. 8, pp. 316-320, 2023. https://www.researchgate.net/publication/374230319.
7. Niaz, M., & Nwagwu, U. (2023). Managing healthcare product demand effectively in the postCOVID-19 environment: Navigating demand variability and forecasting complexities. *American Journal of Economic and Management Business*, 2(8), 316-320. https://www.researchgate.net/publication/374230319_MANAGING_HEALTHCARE_PRODUCT_DEMAND_EFFECTIVELY_IN_THE_POST-COVID-19_ENVIRONMENT_NAVIGATING_DEMAND_VARIABILITY_AND_FORECASTING_COMPLEXITIES.
8. Deethong, T., & Boonnam, N. (2022, January). Forecasting Analysis of the Durian Yield Trends in Southern Thailand Using Holt-Winters Exponential Smoothing Method and Box-Jenkins' Techniques. In 2022 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON) (pp. 29-32). IEEE.
9. Fernando, J. (2022, July 15). Consumer Price Index (CPI). Investopedia.
10. Yenradee, P., Pinnoi, A., & Charoenthavornying, A. (2001). Demand forecasting and production planning for highly seasonal demand situations: Case study of a pressure container factory. ScienceAsia, 27, 271-278.
11. Dolgui, A., & Ivanov, D. (2020). Exploring Supply Chain Structural Dynamics: New Disruptive Technologies And Disruption Risks. In International Journal Of Production Economics (Vol. 229, P. 107886). Elsevier.
12. Dubey, R., Gunasekaran, A., Childe, S. J., Bryde, D. J., Giannakis, M., Foropon, C., Roubaud, D., & Hazen, B. T. (2020). Big Data Analytics And Artificial Intelligence Pathway To Operational Performance Under The Effects Of Entrepreneurial Orientation And Environmental Dynamism: A Study Of Manufacturing Organisations. International Journal Of Production Economics, 226, 107599.
13. wivedi, Y. K., Hughes, L., Ismagilova, E., Aarts, G., Coombs, C., Crick, T., Duan, Y., Dwivedi, R., Edwards, J., & Eirug, A. (2021). Artificial Intelligence (AI): Multidisciplinary Perspectives On Emerging Challenges, Opportunities, And Agenda For Research, Practice And Policy. International Journal Of Information Management, 57, 101994.
14. Gupta, S., Modgil, S., Bhattacharyya, S., & Bose, I. (2022). Artificial Intelligence For Decision Support Systems In The Field Of Operations Research: Review and Future Scope Of Research. Annals Of Operations Research, 1–60.
15. Guzman, A. L., & Lewis, S. C. (2020). Artificial Intelligence and Communication: A Human–Machine Communication Research Agenda. New Media & Society, 22(1), 70–86.
16. Ivanov, D., & Dolgui, A. (2020). Viability Of Intertwined Supply Networks: Extending The Supply Chain Resilience Angles Towards Survivability. A Position Paper Motivated By COVID-19 Outbreak. International Journal Of Production Research, 58(10), 2904–2915.

17. Mahto, R. V, Llanos-Contreras, O., & Hebles, M. (2022). Post-Disaster Recovery For Family Firms: The Role Of Owner Motivations, Firm Resources, And Dynamic Capabilities. Journal Of Business Research, 145, 117–129.
18. CO, H. C. & BOOSARAWONGSE, R., 2007 Forecasting Thailand's rice export: Statistical techniques vs. artificial neural networks. Computers and Industrial Engineering, 53, 610- 627
19. VAHIDINASAB, V., JADID, S. & KAZEMI, A. ,2008 Day-ahead price forecasting in restructured power systems using artificial neural networks. Electric Power Systems Research, 78, 1332-1342.
20. Brownlee, J. (2020, August 26). Train-test split for evaluating machine learning algorithms. Machine Learning Mastery.
21. Mulvenna, A. (2021, February 8). Demand forecasting for 2021: comprehensive overview for retailers. Competera.

8. APPENDIX

```
import os import
sys
from tempfile import NamedTemporaryFile
from urllib.request import urlopen from
urllib.parse import unquote, urlparse from
urllib.error import HTTPError from zipfile
import ZipFile
import tarfile import
shutil
```

```
from dateutil.parser import parse
import itertools import pandas
as pd import numpy as np
import seaborn as sns import
matplotlib.pyplot as plt import
statsmodels.api as sm
plt.rcParams.update({'figure.figsize':(10,7),'figure.dpi':120})
df=pd.read_csv('../input/dataset-superstore-20152018/Dataset- Superstore (2015-2018).csv')
df['Category'].value_counts()
OS= df.loc[df['Category']=='Office Supplies'] OS.head(5)
print('Starting date:',OS['Order Date'].min()) print('Ending
date:',OS['Order Date'].max())
cols = ['Row ID', 'Order ID', 'Ship Date', 'Ship Mode', 'Customer ID', 'Customer Name', 'Segment',
'Country', 'City', 'State', 'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category', 'Product Name',
'Quantity', 'Discount', 'Profit']
OS.drop(cols, axis=1, inplace= True)
OS
OS.isnull().sum()
OS= OS.groupby('Order Date')['Sales'].sum().reset_index() OS.head()
OS['Order Date'] = pd.to_datetime(df['Order Date'])
OS= OS.set_index('Order Date')
OS
OS['Sales'].plot() plt.xlabel('Order
Date') plt.ylabel('Sales')
plt.title('Total sale over years') plt.show()
monthly_OS = pd.DataFrame()

monthly_OS['Sales'] = OS['Sales'].resample('MS').mean() plt.plot(monthly_OS.index,
monthly_OS.Sales, linewidth=3)
OS['Year'] = [d.year for d in OS.index]
OS['Month'] = [d.strftime('%b') for d in OS.index]
years= OS['Year'].unique() years fig, axes =
plt.subplots(1, 2, figsize=(20,7), dpi= 80)
sns.boxplot(x='year', y='Sales', data=OS, ax=axes[0])
sns.boxplot(x='month', y='Sales', data=OS.loc[~OS.year.isin([2014,2917]), :])

axes[0].set_title('Year-wise Box Plot\n(The Trend)', fontsize=18);
axes[1].set_title('Month-wise Box Plot\n(The Seasonality)', fontsize=18)
plt.show() from pylab import rcParams
rcParams['figure.figsize'] = 18, 8

decomposition = sm.tsa.seasonal_decompose(monthly_OS['Sales'], model='additive')
fig = decomposition.plot() plt.show() moving_avg = monthly_OS.rolling(12).mean()
moving_std= monthly_OS.rolling(12).std() orig = plt.plot(monthly_OS,
color='blue',label='Original') mean = plt.plot(moving_avg, color='red', label='Rolling
Mean') std = plt.plot(moving_std, color='black', label = 'Rolling Std')
plt.legend(loc='best')
```

```
plt.title('Rolling Mean & Standard Deviation')
plt.show(block=False) from
statsmodels.tsa.stattools import adfuller print
('Results of Dickey-Fuller Test:') dftest =
adfuller(monthly_OS, autolag='AIC')
dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of Observations
Used'])

for key,value in dftest[4].items():
   dfoutput['Critical Value (%s)'%key] = value print
(dfoutput)
do= pd.read_csv("C:\Users\S YUVASRI\Downloads\Dataset- Superstore (2015-2018).csv") store=
do.loc[do['Category']=='Office Supplies']
cols = ['Row ID', 'Order ID', 'Ship Date', 'Ship Mode', 'Customer ID', 'Customer Name', 'Segment',
'Country', 'City', 'State', 'Postal Code', 'Region', 'Product ID', 'Category', 'Sub-Category', 'Product
Name', 'Quantity', 'Discount', 'Profit'] store.drop(cols,
axis=1, inplace=True)
store = store.groupby('Order Date')['Sales'].sum().reset_index() store
store = store.set_index('Order Date') store.index
store.index = pd.to_datetime(store.index)

y = store['Sales'].resample('MS').mean()
ts_log = np.log(y) plt.plot(ts_log)
moving_avg = ts_log.rolling(12).mean() plt.plot(ts_log)
plt.plot(moving_avg, color='red')
ts_log_moving_avg_diff = ts_log - moving_avg
ts_log_moving_avg_diff.head(12) def
test_stationarity(timeseries):     #Determing
rolling statistics     rolmean =
timeseries.rolling(12).mean()
   rolstd = timeseries.rolling(12).std()

   orig = plt.plot(timeseries, color='blue',label='Original')
mean = plt.plot(rolmean, color='red', label='Rolling Mean')
std = plt.plot(rolstd, color='black', label = 'Rolling Std')
plt.legend(loc='best')
   plt.title('Rolling Mean & Standard Deviation')
plt.show(block=False)     print ('Results of
Dickey-Fuller Test:')     dftest =
adfuller(timeseries, autolag='AIC')
   dfoutput = pd.Series(dftest[0:4], index=['Test Statistic','p-value','#Lags Used','Number of
Observations Used'])     for key,value in dftest[4].items():         dfoutput['Critical Value
(%s)'%key] = value     print (dfoutput)
ts_log_moving_avg_diff.dropna(inplace=True) test_stationarity(ts_log_moving_avg_diff)
expwighted_avg = ts_log.ewm(halflife=12).mean()

plt.plot(ts_log)
plt.plot(expwighted_avg, color='red')
ts_log_ewma_diff = ts_log - expwighted_avg
```

```
test_stationarity(ts_log_ewma_diff)
ts_log_diff = ts_log - ts_log.shift()
plt.plot(ts_log_diff)
ts_log_diff.dropna(inplace=True)
test_stationarity(ts_log_diff) from pylab
import rcParams
rcParams['figure.figsize'] = 18, 8

decomposition = sm.tsa.seasonal_decompose(ts_log, model='additive')
fig = decomposition.plot() plt.show()
from statsmodels.tsa.seasonal import seasonal_decompose decomposition
= seasonal_decompose(ts_log)
residual = decomposition.resid

ts_log_decompose = residual ts_log_decompose.dropna(inplace=True)
test_stationarity(ts_log_decompose)

train= y[:40] test=
y[40:] import
pmdarima

from pmdarima import auto_arima
auto_arima(train, test='adf',seasonal=True, trace=True, error_action='ignore', suppress_warnings=True)
from statsmodels.tsa.arima.model import ARIMA model=ARIMA(train,
order=(1,1,1)).fit()
model.summary()
pred= model.predict(start=len(train), end=(len(y)-1),dynamic=True)
pred test
from  sklearn.metrics import mean_absolute_percentage_error

mape= mean_absolute_percentage_error(test, pred)

print('MAPE: %f' %mape) import
pandas as pd
from sklearn.metrics import mean_absolute_error, mean_squared_error import
numpy as np
actual_data = pd.read_csv('/content/Dataset- Superstore (2015-2018).csv')  # Replace with your path to
actual data
predicted_data = pd.read_csv('/content/Dataset- Superstore (2015-2018).csv')  # Replace with your
path to predicted data y_test = actual_data['Sales'].values
y_pred = predicted_data['Profit'].values

mae = mean_absolute_error(y_test, y_pred)


mse = mean_squared_error(y_test, y_pred)
```

```
mape = np.mean(np.abs((y_test - y_pred) / y_test)) * 100

print(f"Mean Absolute Error (MAE): {mae}") print(f"Mean
Squared Error (MSE): {mse}") print(f"Mean Absolute
Percentage Error (MAPE): {mape}%") import pandas as pd
import numpy as np
from sklearn.metrics import mean_absolute_error, mean_squared_error from
statsmodels.tsa.arima.model import ARIMA

data = pd.read_csv('/content/Dataset- Superstore (2015-2018).csv', parse_dates=['Order Date'],
index_col='Order Date') monthly_sales = data['Sales'].resample('M').sum()

train_data = monthly_sales[:'2016']
test_data = monthly_sales['2017':]

arima_model = ARIMA(train_data, order=(5,1,0))  # Adjust the order (p,d,q) as needed arima_fit
= arima_model.fit()
arima_pred = arima_fit.forecast(steps=len(test_data))


mae = mean_absolute_error(test_data, arima_pred)mse = mean_squared_error(test_data, arima_pred)
mape = np.mean(np.abs((test_data - arima_pred) / test_data)) * 100

print("ARIMA Model Predictions:") print(f"Mean
Absolute Error (MAE): {mae}") print(f"Mean
Squared Error (MSE): {mse}")
print(f"Mean Absolute Percentage Error (MAPE): {mape}%")

import matplotlib.pyplot as plt
plt.figure(figsize=(10, 6)) plt.plot(train_data,
label='Train') plt.plot(test_data, label='Test')
plt.plot(test_data.index, arima_pred, label='ARIMA Predictions', color='green') plt.legend(loc='best')
plt.title('Sales Forecast using ARIMA')
plt.xlabel('Order Date')
plt.ylabel('Sales') plt.show()
```