Sales Forecasting Using Time Series Decomposition and Predictive Modeling in MATLAB

[1]. Problem Identification:

In engineering time series analysis plays a critical role. Tasks such as capacity planning, demand forecasting, predictive maintenance, and production optimization utilize time series frequently. In case of predictive maintenance, it allows engineers to monitor sensor data such as vibration and temperature to foresee equipment failures and schedule timely interventions, reducing downtime and costs [1]. On the other hand, in manufacturing and supply chain operations, demand forecasting through time series helps optimize inventory, streamline production, and allocate workforce efficiently. Also, for energy management, utilities rely on time series models to anticipate electricity and gas consumption, improving generation and distribution strategies [2]. In case of quality control, time series-based statistical process control detects anomalies in product quality, guiding adjustments in production parameters [3]. Furthermore, in environmental engineering, forecasting pollution levels, water usage, and climate patterns through time series supports sustainable resource planning and environmental protection [4].

The assigned time series forecasting task involves analyzing monthly observations for 36 months that exhibit several crucial components. The trend reflects a long-term upward or downward movement in the data, such as consistent sales growth over time. On the other hand, seasonality captures repeating patterns that occur every 12 months, like increased sales during holiday seasons. Also, cyclical variations represent medium-term variations that span more than a year, often influenced by broader economic cycles. Finally, random noise comprises of unpredictable and irregular variations caused by external events, such as sudden supply chain disturbances. The objective of this analysis is to decompose the series into these individual components, understand their interactions, and develop a reliable model to forecast sales for the next six months using a decomposition-based approach.

Raw time series can be converted into actionable insights by concise implementation of Mathematical models. Through decomposition modeling, the series is broken down into trend (T_t), seasonality (S_t), cyclical variations (C_t), and residual noise (R_t). This can be represented using either a multiplicative model $Y_t = T_t * S_t * C_t$ or an additive model $Y_t = T_t + S_t + C_t$, depending on the nature of the data. This decomposition helps uncover hidden patterns, such as quantifying the impact of holiday sales or promotional drives. Apart from this, the forecasting mathematical techniques such as Least Squared Regression, SARIMA (Seasonal ARIMA) and Exponential Smoothing (ETS) can be applied to extrapolate trends and seasonal effects, while also capturing autocorrelation structures**.** The regression models play an important role by establishing linear relationships between time-dependent variables and external predictors, offering interpretable overall insights for forecasting. However, advanced machine learning models like XGBoost, RNN (Recurrent Neural Networks), and LSTM (Long Short-Term Memory) networks are capable of modeling complex, nonlinear relationships within the data. The insights generated from these forecasts support strategic decision-making across various domains, guiding inventory management, marketing budget allocation, and workforce planning accordingly.

[2]. MATLAB Implementation

The MATLAB based implementation file is attached as "*Script.m*". Below are some plots generated based on implementation of the script:

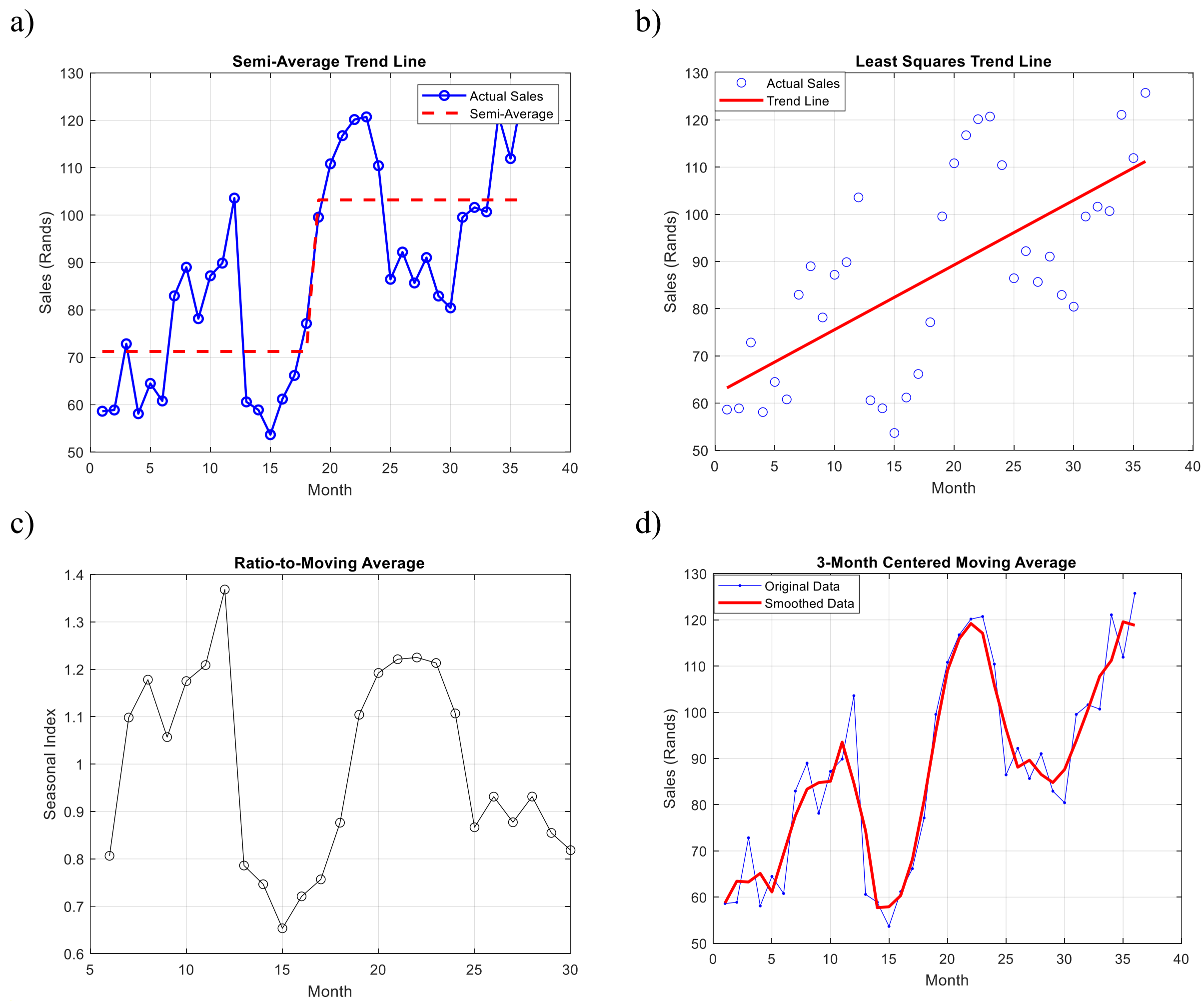

Figure 1 Components of given Time Series

Multiple Mathematical Models were implemented to deeply analyze the data and make predictive models to get future insights. The stepwise breakdown of workflow is:

Step 1**:** In this first step the given data was vectorized to enhance computational and performance efficiency. Let y_t denote the sales at time t, where $t = 1,2, \dots, n$; we defined:

$$y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}, t = \begin{bmatrix} 1 \\ 2 \\ \vdots \\ n \end{bmatrix}$$

Step 2: To calculate Semi-Average Trendline, we split the time series into two halves $y'_1 \rightarrow y_1, \dots, y_{n/2}$ and $y'_2 \rightarrow y_{n/2}, \dots, y_n$. Then semi-average trend is:

$$y_t^{semi} = \begin{cases} y_1' \ , & t \le \frac{n}{2} \\ y_2' \ , & t > \frac{n}{2} \end{cases}$$

This provides a simple piecewise constant trend approximation based on semi-average.

Step 3: To fit a least square trendline we fit linear model of the form $y_t^{linear} = \beta_0 + \beta_1 * t$ where the parameters β_0 and β_1 are obtained by minimizing the objective function as:

$$min_{\beta_0, \beta_1} \sum_{t=1}^{n} (y_t - \beta_0 - \beta_1 * t)^2$$

MATLAB's $polyfit(t, y, 1)$ function yields:

$$\beta_1 = \frac{n \sum ty - \sum t \sum y}{n \sum t^2 - (\sum t)^2}, \qquad \beta_0 = y^{'} - \beta_1 t'$$

Step 4: To perform seasonal decomposition, we first computed moving average for 12-month window as:

$$y_t^{R\ MA} = \left(\frac{1}{12}\right) \sum_{i=t-5}^{t+6} y_i$$

The seasonal index component (ratio to moving average) is isolated from given time series as:

$$S_t = \frac{y_t}{y_t^{RMA}}$$

Step 5: For smoothing of given time series by filtering out fluctuations and noise, the 3-monhts (k=3) centered moving average was calculated as:

$$y_t^{MA} = \left(\frac{1}{k}\right) \sum_{i=t-k/2}^{t+k/2} y_i$$

Step 6: Using the trained linear model y_t^{linear} forecast for future values for $t = n + 1, \dots, n + h$ where $h = 6$ (months) was performed to get insights into future sales.

Step 7: Residual analysis for predicted values was performed by calculating parameter $e_t = y_t - y_t^{linear}$ and then creating a scatter plot of e_t against t.

While applying the time series forecasting models based on above-mentioned steps, few challenges were met. One of the major issues was the loss of data points at the edges when implementing moving average technique, especially with larger windows like the 12-month centered moving average. To deal this, the *'Endpoints'* parameter in MATLAB's '*movmean*' function was set to *'discard*', confirming that only complete averages were included, hence maintaining accuracy in seasonal ratio calculations. The other challenge was about selecting the correct seasonal period for decomposition and modeling. This was dealt by using domain knowledge, as the data was monthly, a seasonality period of 12 was used to show annual cycles.

[3]. Analysis and Evaluation:

The overall quality of the models applied to the sales data validates a solid understanding of trend and seasonality analysis. The semi-average trend line (Figure 1a) divides the data into two clear phases and captures a general change in sales performance. The least squares trend line (Figure 1b) provides a more accurate linear representation of the upward trend over time, effectively modeling the long-term direction of sales growth. However, this linear model does not account for repeating seasonal fluctuations that are obvious in the data. The ratio-to-moving average plot (Figure 1c) highlights strong seasonal effects, with consistent peaks and troughs recurring across fixed intervals, depicting the presence of seasonality in the dataset. The 3-month centered moving average (Figure 1d) further smooths short-term irregularities and discloses the cyclical nature of the data, making it easier to identify underlying patterns. The residual scatter plot (Figure 2) supports this finding, rather than being randomly scattered around zero, the residuals follow a wave-like pattern. This non-random structure in the residuals indicates that the model is missing seasonal or cyclical components. In conclusion, while the trend models successfully capture the overall direction of the data, the residual analysis suggests that a more advanced model incorporating both trend and seasonality would better represent the true structure of the sales series and improve forecast accuracy. To achieve this, a non-linear models, such as seasonal exponential smoothing, Holt-Winters method, or SARIMA, should be considered to capture the complex dynamics present in the data.

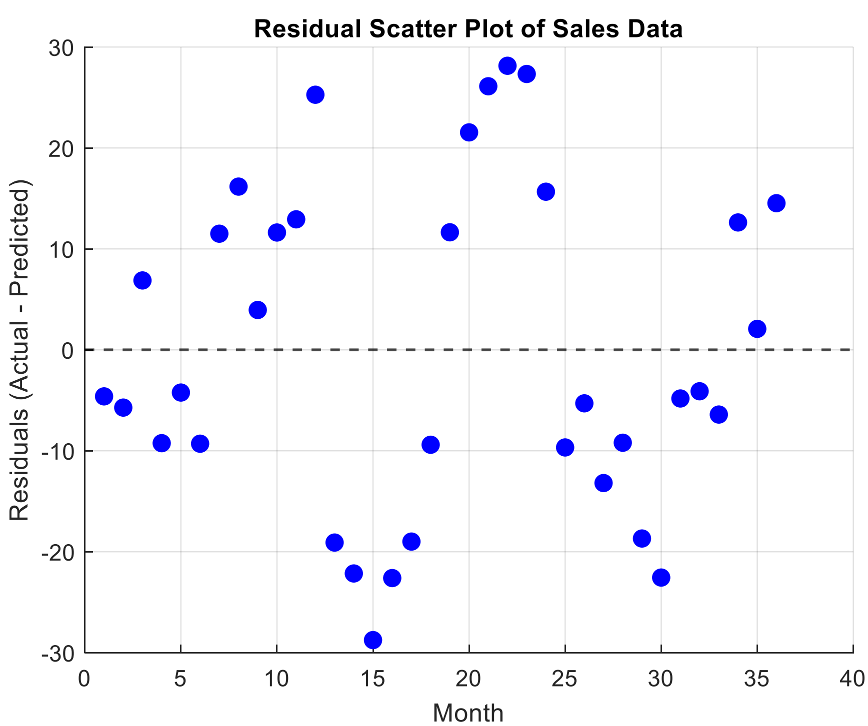

Figure 2 Residual Scatter Plot

[4]. Forecasting and Interpretation:

The forecasted sales for the next 6 months have been generated using a least-squares linear trend model**,** as shown in the Figure 3a. The trend line follows a steady upward slope, and the forecast extends that pattern, predicting a gradual increase in sales over the upcoming half-year period. This model reflects the overall positive growth in sales and is particularly useful for understanding long-term directional trends. However, as the historical sales data clearly exhibits seasonal fluctuations, the trend-only model does not fully capture these repeating patterns. The forecasted points lack the peaks and troughs seen in the actual data, indicating that the model oversimplifies the sales behavior by ignoring cyclical and seasonal components.

To improve forecast accuracy and capture the real dynamics of sales, more advanced models should be employed. Techniques such as Holt-Winters exponential smoothing, SARIMA, or even machine learning models (like XGBoost with time-based features) can incorporate both trend and seasonality. These models can better adapt to changing sales behavior, especially when consistent seasonal effects are present. In conclusion, while the trend model provides a good starting point for forecasting, future analysis should integrate non-linear models with seasonal adjustment to provide more reliable and insightful sales predictions.

a) b)

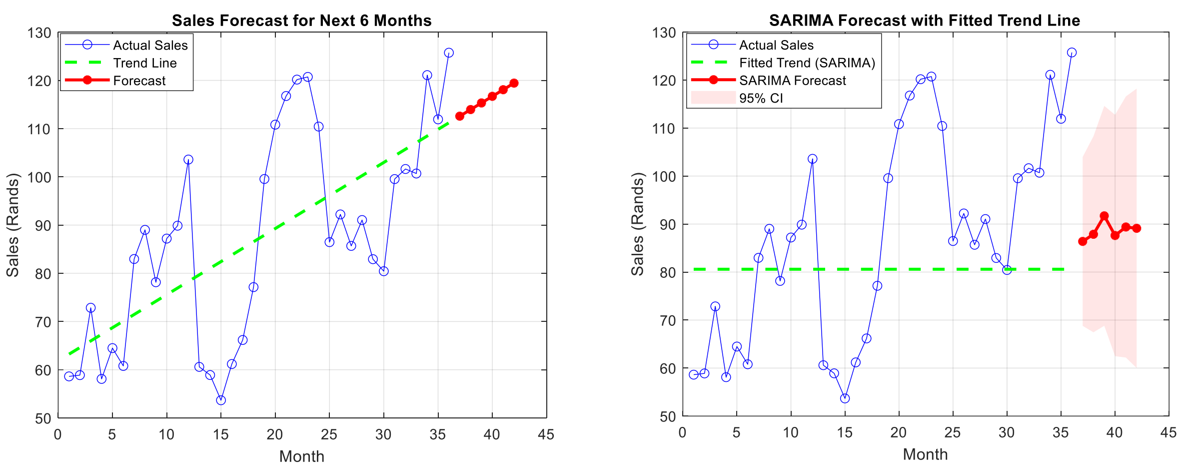

Figure 3: a) Sales Forecast (Next 6-months) based on Regression Model
b) Sales Forecast (Next 6-months) based on SARIMA Model

As an advanced step in predictive modeling, we applied the SARIMA (Seasonal Auto-Regressive Integrated Moving Average) model, which effectively captures both trend and seasonal patterns in time series data. Figure 3b illustrates the six-month forecast generated by the SARIMA model for the given dataset. The forecasted points are accompanied by a shaded pink region, representing the 95% confidence interval, indicating the range within which future observations are expected to fall with high probability. Notably, this confidence band widens as the forecast horizon extends, reflecting increasing uncertainty over time. While this model provides a statistically sound and seasonally adjusted forecast, it demonstrates limitations in modeling non-linear relationships present in the data. To further enhance predictive performance, especially for complex patterns, more sophisticated models such as tree-based algorithms (e.g., XGBoost) or deep learning architectures (e.g., RNN, LSTM, GRU) could be employed, which are known to capture non-linear dependencies and temporal dynamics more efficiently.

References:

[1]. Jardine, A. K. S., Lin, D., & Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical Systems and Signal Processing*, 20(7), 1483–1510. https://doi.org/10.1016/j.ymssp.2005.09.012.

[2]. Taylor, J. W., & McSharry, P. E. (2007). Short-term load forecasting methods: An evaluation based on European data. *IEEE Transactions on Power Systems*, 22(4), 2213–2219. https://doi.org/10.1109/TPWRS.2007.907583.

[3]. Woodall, W. H. (2000). Controversies and contradictions in statistical process control. *Journal of Quality Technology*, 32(4), 341–350. https://doi.org/10.1080/00224065.2000.11980013.

[4]. Delavar, M. R., Gholami, A., Shiran, G. R., Rashidi, Y., Nakhaeizadeh, G. R., Fedra, K., & Hatefi Afshar, S. (2019). A Novel Method for Improving Air Pollution Prediction Based on Machine Learning Approaches: A Case Study Applied to the Capital City of Tehran. *ISPRS International Journal of Geo-Information*, *8*(2), 99. https://doi.org/10.3390/ijgi8020099.

---THE END---