Sales Forecasting for DMart: A Time Series Analysis

1. Introduction

Sales forecasting is a crucial aspect of retail business operations, allowing companies to predict future sales and optimize inventory management. This report explores various time series forecasting techniques to predict sales based on historical data. I analyse and compare different models, including SARIMA and LSTM, to identify the best-performing approach.

2. Problem Statement

The goal of this project is to develop a robust forecasting model that can accurately predict future sales for a retail store based on past sales data. Accurate forecasting will help optimize inventory levels, reduce stockouts, and improve profitability.

3. Dataset Description

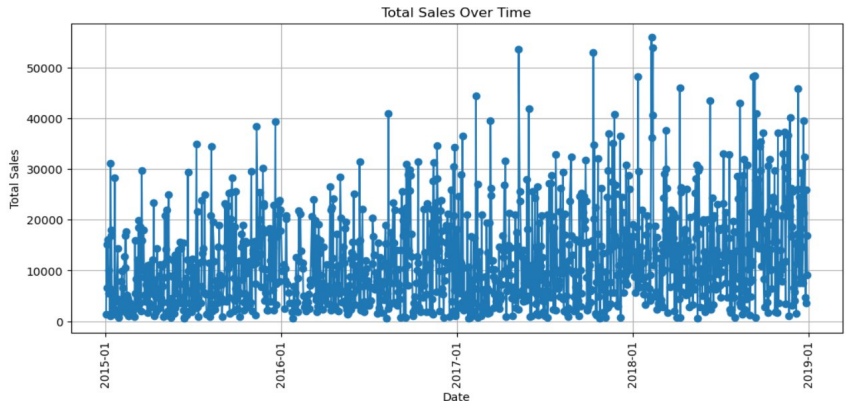
The dataset used for this analysis contains monthly sales data for a retail store. The key variables include:

- Sales: The total revenue generated in a month
- Discount: The total discount offered during the month
- Profit: The total profit made
- Sales Difference: Difference between consecutive sales values (used for stationarity)
- Dataset Source: kaggle.com (https://www.kaggle.com/datasets/nikolaireeds/dmart-grocery-sales)

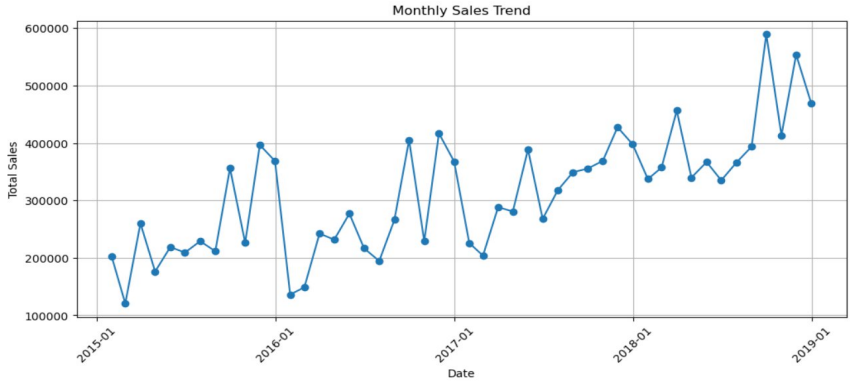
4. Exploratory Data Analysis (EDA)

4.1 Data Visualization

Let’s begin by visualizing the sales trend over time to understand seasonal patterns and long-term trends.



The graph looks messy with little to no recognizable pattern. We smoothen the graph to identify a pattern or trend.



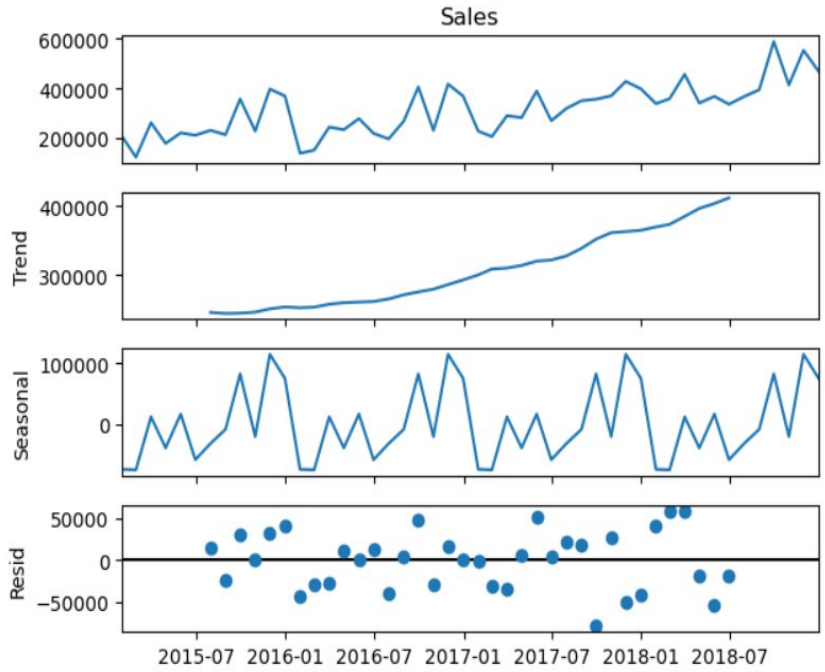
We see an upward trend suggesting growing sales over time.

4.2 Seasonal Decomposition

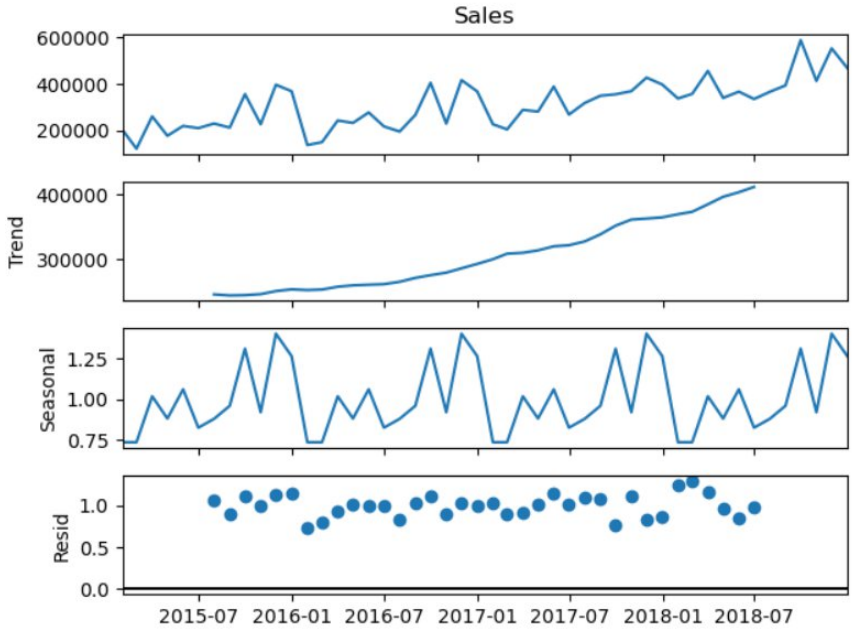
Using seasonal decomposition, I break down the time series into trend, seasonality, and residual components.

I use both ‘additive’ and ‘multiplicative’ models for STL decomposition.

- Additive model:
  In this model, the trend, seasonality, and residual components are assumed to be added together to form the original time series. This applies when the seasonal variations remain constant regardless of the overall trend. This model decomposed the data as follows;



- Multiplicative model:
  Here, the seasonal component is multiplied by the trend-cycle component to obtain the original series. This is suitable for situations where the seasonal fluctuations are proportional to the level of the trend. This model decomposed the data as follows;



Both the additive and multiplicative decomposition look very similar, that means the seasonal pattern is not strongly dependent on the trend.

4.3 Checking for Stationarity

A crucial requirement for time series modelling is stationarity. I apply the Augmented Dickey-Fuller (ADF) test to determine whether the sales data is stationary. This is a widely used test for stationarity that checks for the presence of a unit root in the data. A unit root essentially means the data has a non-stationary trend, and differencing might be necessary.

The ADF test works by fitting a specific model to the data and then evaluating the null hypothesis (H_0) that there is a unit root (non-stationary) against the alternative hypothesis (H_1) that the series is stationary.

ADF Test Hypothesis:

- Null Hypothesis (H_0): The time series has a unit root (i.e., it is non-stationary).
- Alternative Hypothesis (H_1): The time series is stationary.

Decision Criteria:

- If the ADF statistic is less than the critical value at a given significance level, I reject the null hypothesis.
- A low p-value (typically less than 0.05) indicates strong evidence against the null hypothesis, suggesting the time series is stationary.

Following are the results of the ADF test:

- ADF Statistic: 0.947
- p-value: 0.993
- Critical Values: {1%: -3.620, 5%: -2.943, 10%: -2.610}

These results suggest that the data is non stationary. To resolve this, I apply first order differencing. Differencing is a method to remove trends and seasonality from a time series. It involves subtracting the previous observation from the current observation.

Following are the results of the ADF test after differencing:

- ADF Statistic after differencing: -5.722
- p-value: 6.874×10^{-7}
- Critical Values: {1%: -3.626, 5%: -2.945, 10%: -2.611}

These results suggest that the data is now stationary after differencing.

5. Model Selection and Implementation

I tested multiple forecasting techniques to find the best-performing model.

5.1 SARIMA Model

The Seasonal Auto-Regressive Integrated Moving Average (SARIMA) model is an extension of ARIMA that accounts for seasonality. The model is defined by the parameters:

- (p, d, q) for the non-seasonal component
- (P, D, Q, s) for the seasonal component

SARIMA Equation:

$$Y_t = c + \sum_{i=1}^{p} \varnothing_i Y_{t-i} + \sum_{i=1}^{p} \theta_j \epsilon_{t-j} + S_t$$

where:

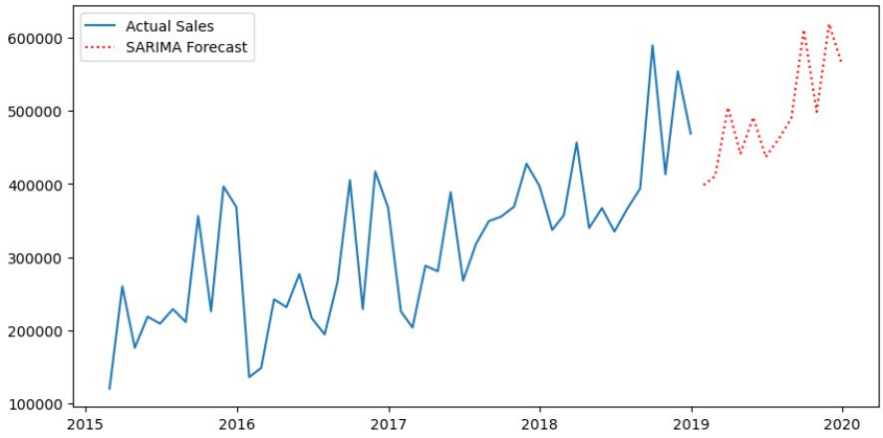
- Y_t is the sales value at time t
- $\varnothing_i$ are autoregressive (AR) terms
- θ_j are moving average (MA) terms
- S_t accounts for seasonality
- c is the constant term
- ϵ_t is the error term

SARIMA Model Evaluation

The model was fine-tuned using grid search, and the final parameters Ire selected based on the loIst AIC score.

5.1.1 SARIMA Forecasting:

Using the fine-tuned model, I forecasted the sales for the year 2020. Below is the forecasted graph.



5.2 LSTM Model

Long Short-Term Memory (LSTM) networks are a type of recurrent neural network (RNN) designed for sequence prediction. They use memory cells to capture long-term dependencies in time series data.

LSTM Equation:

Each LSTM unit consists of three gates:

- Forget Gate: $f_t = \sigma\left(W_f\left[h_{t-1}, x_t\right] + b_f\right)$
- Input Gate: $i_t = \sigma\left(W_i\left[h_{t-1}, x_t\right] + b_i\right)$
- Output Gate: $o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$

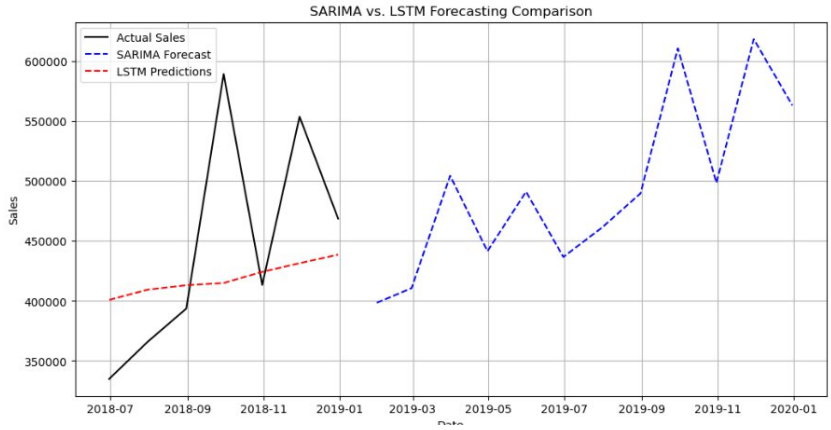
where σ is the sigmoid activation function, and W and b are Iight matrices and biases, respectively.

5.3 Model Comparison

I evaluated both models using three key metrics:

- Mean Absolute Error (MAE): $MAE = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \widehat{y_i}\right|$
- Root Mean Squared Error (RMSE): $RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2}$
- Mean Absolute Percentage Error (MAPE): $MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y_i}}{y_i}\right|$

Model	MAE	RMSE	MAPE
SARIMA	45325.487	58957.790	10.96%
LSTM	66657.812	87023.009	13.89%



Final Decision

The SARIMA model outperformed the LSTM model in terms of loIr MAE, RMSE and MAPE. A possible reason for this could be the limited size of the dataset. Given the poor performance of LSTM, I decided to use SARIMA as our final forecasting model.

6. Conclusion

This project explored time series forecasting methods for demand prediction in a DMart store. I performed extensive EDA, tested SARIMA and LSTM models, and found that SARIMA provided the best results. The LSTM model, despite being more complex, failed to outperform SARIMA, likely due to the limited dataset size and lack of strong temporal dependencies.

7. References

- Box, G. E. P., & Jenkins, G. M. (1970). *Time Series Analysis: Forecasting and Control.* San Francisco: Holden-Day.
- Hochreiter, S., & Schmidhuber, J. (1997). *Long Short-Term Memory.* Neural Computation.
- Chatfield, C. (2000). *Time-Series Forecasting.* Chapman and Hall/CRC.

Author: Prathamesh Ugle