Demand Forecasting for Supply Chain Systems

Shreyash Padeer(2024AIB2800) | Sanskar Gupta(2024AIB2293) | Parth Prajapati(2024AIB2288)

1 Introduction

Effective inventory management is crucial for retail businesses to optimize operations, minimize costs, and meet customer demand. This project leverages Machine Learning and Deep Learning to forecast product demand in a retail environment, using a rich dataset containing transactional, product, and contextual information. The objective is to build and evaluate a predictive model that can accurately forecast demand, enabling better inventory planning and decision-making.

1.1 What is Time Series Data?

Time series data is a sequence of observations collected over time, typically at regular intervals such as daily, monthly, or annually. The order of data points is crucial because observations may depend on previous values.

Components of Time Series: Time series data can generally be decomposed into the following components:

- **Trend:** The long-term direction in the data, which may be increasing, decreasing, or stable over time.
- **Seasonality:** Regular, periodic fluctuations occurring at fixed intervals (e.g., weekly, monthly, annually).
- **Cyclic Behavior:** Long-term patterns or fluctuations that are not necessarily periodic and often related to economic or business cycles.
- **Noise/Residual (Irregular Component):** Random variation or residuals in the data that can't be explained by trend, seasonality, or cycles.

2 Dataset Description

In this project, two different time series datasets were explored to analyze and forecast retail sales using classical decomposition techniques.

2.1 Retail Store Inventory Forecasting (Synthetic Data)

This dataset is synthetically generated and represents inventory and sales data for a fictional retail store. While the data includes multiple features such as `Item_ID`, `Store_ID`, `Date`, and `Sales`, it was found that:

- The data is highly **fluctuating** and noisy.
- There is no clear presence of **trend**, **seasonality**, or **cyclic patterns**.
- Due to its synthetic nature, it may not accurately represent real-world business scenarios.

Exploratory Data Analysis (EDA) confirmed that the time series is irregular and does not exhibit meaningful decomposable structure.

2.2 Walmart Sales Forecasting (Real-world Data)

This dataset contains real-world historical weekly sales data for Walmart stores, covering multiple departments and locations. Key features include:

- `Store`, `Dept`, `Date`, `Weekly_Sales`, `IsHoliday` etc.
- Data spans multiple years with consistent and structured time intervals

Upon analysis, this dataset showed clear and useful time series characteristics:

- Strong **seasonality** around specific times of the year
- Possible **cyclical behavior** reflecting economic or business cycles

Hence, the Walmart dataset was selected for further modeling and decomposition analysis, as it provides a richer and more realistic basis for time series forecasting.

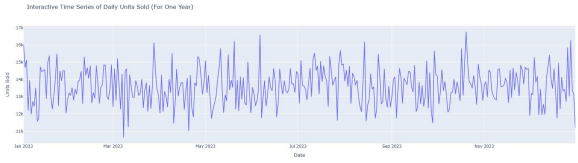
3 Exploratory Data Analysis (EDA)

To understand the structure and behavior of the time series data, we performed detailed Exploratory Data Analysis (EDA) on both datasets. The goal was to detect the presence of key components like trend, seasonality, and cyclicity.

3.1 Retail Store Inventory Forecasting Dataset (Synthetic)

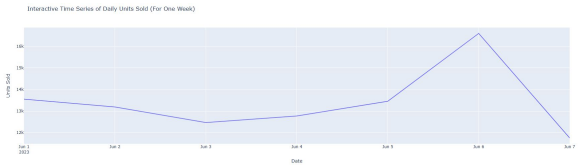
Visualizations were generated across different time scales to investigate the nature of sales:

Sales over Entire Time



Sales over One Year

Sales over One Month



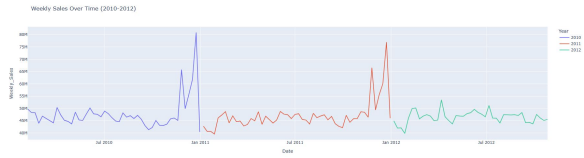
Sales over One Week

Observations:

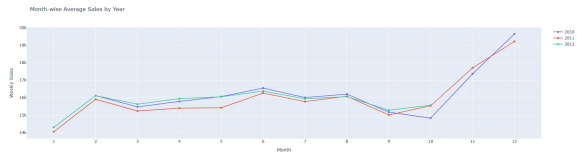
- Sales are highly **irregular and fluctuating**.
- No evident **trend**, **seasonality**, or **cyclical behavior**.
- Dataset lacks consistent temporal structure; not ideal for decomposition.

3.2 Walmart Sales Forecasting Dataset (Real-world)

This real-world dataset was analyzed to understand its temporal structure:



Weekly Sales (Year-wise)



Month-wise Average Sale by Year

Observations:

- Clear **trend** in sales over multiple years.
- Possible **cyclical variations** influenced by economic/business factors.
- Dataset is suitable for decomposition and forecasting models.

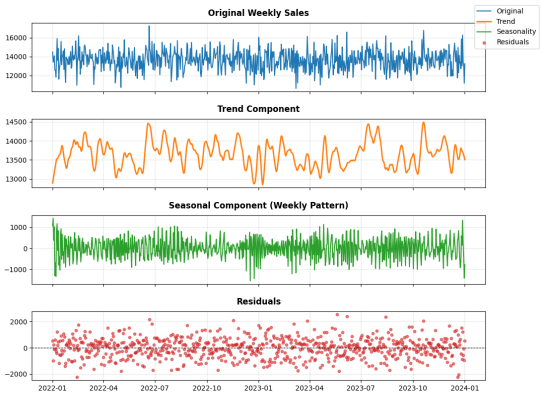
4 Methodology

Based on our EDA insights, we developed a time series analysis pipeline involving preprocessing, decomposition, forecasting, and evaluation.

4.1 Data Preprocessing

- **Data Loading and Exploration:** Loaded the dataset and performed basic inspection using `info()`, `describe()`, and checked for missing values.
- **Feature Engineering:** Create additional features to improve predictive power, such as:
  - Date/time components (hour, day, month)
  - Lagged variables (previous values at specific time lags)
  - Rolling window statistics (moving averages, rolling standard deviations)
- **Data Normalization:** All features were scaled to the range [0, 1] using the `MinMaxScaler` from the `scikit-learn` library to facilitate model training and convergence.

4.2 Time Series Decomposition



- Applied additive decomposition
- No significant trend or seasonality observed
- Residual component dominated the series
- Minimal seasonal variation detected

Conclusion: Time series decomposition was ineffective on this synthetic data as it lacks meaningful temporal patterns.

Figure 1: Decomposition of Synthetic Dataset

- Applied additive decomposition
- Clear seasonality patterns visible
- Consistent upward trend in high-performing stores
- Residuals show minor, normally-distributed fluctuations

Conclusion: The decomposition revealed strong temporal patterns enabling accurate forecasting models using the extracted trend and seasonal components.

Figure 2: Decomposition of Walmart Dataset

4.3 Different Models Implementation and Benchmarking

- **Deep Learning Models:** Implemented and compared several architectures, including RNN, LSTM, BiLSTM, GRU, hybrid model (CNN + LSTM), Autoformer, and Transformer, to capture various temporal dependencies and patterns in the time series data.
- **Traditional Models:** Used ARIMA and SARIMA as statistical baselines to benchmark against non-sequential machine learning methods.
- **Input/Output Design:** Structured the data using sliding windows of previous time steps.

4.4 Model Training

- **Training Configuration:** We used the Adam optimizer and Mean Squared Error (MSE) loss, with appropriate batch size and epoch count. Since this is time series data, shuffling is avoided to preserve temporal order.
- **Overfitting Control:** Regularization techniques such as Dropout and Batch Normalization are applied to improve generalization and model robustness.

4.5 Evaluation

- **Performance Metrics:** Evaluated model performance using metrics such as MAE (Mean Absolute Error), MSE (Mean Squared Error) and RMSE (Root Mean Squared Error), or R^2 score.
- **Visualization:** Plotted predicted vs. actual values to visually assess forecast accuracy.

5 Empirical Results and Model Comparison

Model Type	Walmart Dataset Without Rolling Stats	Walmart Dataset With Rolling Stats
ARIMA	RMSE: 1970118.67 R^2: -0.6667	RMSE: 2128358.50 R^2: -0.9452
SARIMA	RMSE: 205140.79 R^2: 0.7140	RMSE: 137354.36 R^2: 0.8417
RNN (Sequence Length=7)	RMSE: 219150.73 R^2: 0.6828	RMSE: 106867.48 R^2: 0.9245
LSTM (Sequence Length=15)	RMSE: 234465.21 R^2: 0.6373	RMSE: 100475.05 R^2: 0.9334
BiLSTM (Sequence Length=15)	RMSE: 292182.40 R^2: 0.4368	RMSE: 93990.11 R^2: 0.9417
GRU (Sequence Length=30)	RMSE: 227974.33 R^2: 0.6576	RMSE: 86876.69 R^2: 0.9503
CNN-LSTM (Sequence Length=15)	RMSE: 179947.36 R^2: 0.8927	RMSE: 98248.95 R^2: 0.9363
Autoformer (Sequence Length=52)	RMSE: 188687.36 R^2: 0.8477	RMSE: 124785.58 R^2: 0.8814
Transformer (Sequence Length=52)	RMSE: 215732.45 R^2: 0.6939	**RMSE: 82639.39** **R^2: 0.9551**

Table 1: Model Performance Comparison for Test Data

Significance of Rolling Statistical Features

- **Contextual Information:** Rolling statistics (also called moving statistics) were used to identify trends, smooth out noise, and detect patterns by calculating statistics over a sliding window of observations.
- **Enhanced Feature Engineering:** Rolling metrics (e.g., rolling mean, rolling min/max, rolling std) provide temporal context to the model. These features help deep learning models better capture local trends and seasonality within a window of time, improving predictive accuracy.
- **Better Performance in Neural Models:** When applied before feeding data into RNNs, LSTMs, or Transformers, rolling statistics can lead to significantly better convergence and generalization.

Key Concepts of Rolling Statistics

- **Rolling Window:** A fixed-size subset of consecutive observations that moves through the data Window can be defined by Time duration.
- **Common Rolling Statistics:**
  - **Basic Statistics:** Rolling Mean (Moving Average), Rolling Median, Rolling Standard Deviation, Rolling Min/Max
  - **Advanced Statistics:** Rolling Correlation, Rolling Quantiles, Rolling Apply

6 Analysis and Conclusion

6.1 Analysis

From the results shown in above table, it is evident that the inclusion of rolling statistical features significantly enhances the performance of most models across both error (RMSE) and goodness-of-fit (R^2) metrics.

6.1.1 Baseline Models (ARIMA & SARIMA):

- **ARIMA** performs poorly in both setups, showing negative R^2 values and very high RMSEs, indicating it fails to capture the underlying patterns and presence of seasonality in the Walmart dataset.
- **SARIMA**, which accounts for seasonality, performs significantly better than ARIMA. Its performance improves further with rolling stats (RMSE reduced from 205,140.79 to 137,354.36 and R^2 increased from 0.7140 to 0.8417), highlighting the value of temporal smoothing features in time series modeling.

6.1.2 Recurrent Models:

- All RNN-based models show considerable improvements when rolling statistics are used. For instance, the **GRU** model achieves the lowest RMSE (86,876.69) and highest R^2 (0.9503), outperforming all other models with rolling stats.
- **LSTM** and **BiLSTM** models also benefit from rolling stats, with BiLSTM reaching a high R^2 of 0.9417 despite relatively worse performance in the "Without Rolling Stats" scenario.
- **CNN-LSTM** performs consistently well in both setups, suggesting that combining spatial and temporal features is advantageous.

6.1.3 Transformer-Based Models:

- The **Transformer** model achieves the best R^2 score of 0.9551 with rolling stats and a notably low RMSE of 82,639.39, outperforming all others in predictive accuracy.
- The **Autoformer**, though not the top performer, also shows solid improvements with rolling features, validating its utility in long-sequence forecasting.

6.2 Conclusion

The analysis demonstrates that:

1. **Rolling statistics** provide substantial value across all model architectures with major improvements.
2. **Best Overall Performance:** Transformer outperforms all other models on the Walmart dataset with rolling statistics, achieving the lowest RMSE (82639.39) and highest R^2 score (0.9551). This indicates its superior ability to capture complex temporal patterns and long-term dependencies when augmented with smoothed statistical features.
3. **Deep Learning vs. Traditional Models:** Deep learning models consistently outperformed the traditional statistical models (ARIMA, SARIMA). Even the simpler RNN outperformed SARIMA significantly when rolling stats were included. ARIMA performed very poorly on both dataset variations, indicated by extremely high RMSE and negative R^2 values (suggesting its predictions were worse than simply predicting the mean).

Why Some Models Perform Better:

1. **Transformer-based models** work really well with time series because they use attention mechanisms to focus on important parts of the data. When we use rolling statistics to make the input data more stable, these models can learn patterns even better — which is why they perform the best.
2. **Rolling statistics** (like moving averages) help remove random noise and make the data more stable over time. This gives deep learning models cleaner signals to learn from, which makes their predictions more accurate.
3. Models like **LSTM**, **GRU**, and **BiLSTM** are made to handle sequences of data (like time series). They do a much better job when given longer, smoother sequences — which is exactly what rolling statistics help provide.
4. Traditional models like **ARIMA** and **SARIMA** work well for simple and linear patterns. But real-world data (like Walmart sales) is often messy and influenced by many factors, so these models struggle with complex patterns.
5. **Autoformer** performs well even without rolling stats because it's designed to handle seasonal trends and long-term patterns in the data. This makes it more resistant to noise and sudden changes.

7 References

This research aims to enhance demand forecasting accuracy through advanced deep learning techniques, contributing to more efficient supply chain management in retail environments.

1. *Real-World Walmart Dataset*. Kaggle. Available at: `https://www.kaggle.com/datasets/aslanahmedov/walmart-sales-forecast`
2. *Retail Store Inventory Forecasting Dataset(Synthetic Dataset)*. Kaggle. Available at: `https://www.kaggle.com/datasets/anirudhchauhan/retail-store-inventory-forecasting-dataset`
3. Wainaina, Pierre. *The Complete Guide to Time Series Forecasting Models.* Medium, 2023. Available at: Medium Article
4. Huntresselle. *Deep Learning Techniques for Time Series Forecasting: A Comprehensive Guide.* Medium, 2023. Available at: Medium Article
5. Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). *The Performance of LSTM and BiLSTM in Forecasting Time Series*. 2019 IEEE International Conference on Big Data (Big Data), 3285-3292. `https://doi.org/10.1109/BigData47090.2019.9005997`.
6. Wu, Haixu, et al. *Autoformer: Decomposition Transformers with Auto-Correlation for Long-Term Series Forecasting*. arXiv preprint arXiv:2106.13008, 2021. Available at: `https://arxiv.org/pdf/2106.13008`