M5 Forecasting-Accuracy

Arnab Roy[1], Ryan Wlynetz[2], Sujan Shrestha[3], and Urvesh Bhagat[4]

[1]arn97, arnabroy@udel.edu, Literature Manager
[2]Legacy560, rlguyfl@udel.edu, Communications Manager
[3]Osujan, sujan@udel.edu, Visualization Manager
[4]urvesh2012, urveshb@udel.edu, Data and Analysis Manager

March 17, 2025

Abstract

Accurate demand forecasting is essential for inventory management and supply chains. This project evaluates time-series forecasting models using Transformer-based architectures, statistical methods (ETS, ARIMA, Prophet), and feature engineering techniques. We compare these approaches in terms of accuracy, efficiency, and interpretability using RMSE and MASE. The final deliverable includes model evaluations and insights into best practices for large-scale forecasting.

1 Introduction

Accurate demand forecasting is crucial in supply chain management, enabling businesses to optimize inventory and anticipate sales fluctuations. The M5 Forecasting - Accuracy competition on Kaggle provides a real-world challenge where participants predict daily sales for thousands of Walmart products using historical data, calendar events, and pricing information.

Traditional time-series models, such as Exponential Smoothing (ETS) and ARIMA, effectively capture trends and seasonality but struggle with large-scale data and complex dependencies [1]. Facebook's Prophet model offers flexibility in handling missing data and seasonal variations [**?**]. However, these approaches may not fully exploit intricate patterns present in high-dimensional datasets.

Deep learning models, particularly Transformer-based architectures, have demonstrated remarkable success in sequence modeling tasks, including time-series forecasting. Originally designed for natural language processing [3], Transformers have been adapted to capture long-range dependencies in time-series data [4]. Additionally, feature engineering techniques such as lag-based features, rolling statistics, and hierarchical aggregation have been shown to enhance forecasting performance [5].

In this project, we aim to compare Transformer-based models, traditional statistical approaches (ETS, ARIMA, Prophet), and advanced feature engineering techniques to assess their effectiveness in demand forecasting. Our goal is to identify best practices for large-scale forecasting and provide insights into model performance, computational efficiency, and interpretability.

Data

The datasets used in this project are provided by the M5 Forecasting - Accuracy competition on Kaggle [6]. These datasets contain historical sales data, calendar-related information, and product pricing details. Below is a brief description of each dataset:

- **calendar.csv**: Contains date-related information such as holidays, special events, and SNAP program eligibility indicators for California, Texas, and Wisconsin. Several columns related to events have significant missing values, but they primarily indicate the absence of events rather than data corruption.
- **sales_train_validation.csv**: Includes daily sales data for 30,490 products across Walmart stores over 1,914 days (ending at day 1913). It is used for model validation.
- **sales_train_evaluation.csv**: Similar to the validation dataset but extends the sales data beyond day 1913, adding 28 extra days. It serves as the evaluation set for final predictions.
- **sell_prices.csv**: Provides historical selling prices of products in different stores, indexed by week.

The following table summarizes key characteristics of these datasets:

Dataset Name	**Rows**	**Columns**	**Valid Rows (No NaNs)**
calendar.csv	1969	14	Nearly all rows contain NaNs in event columns
sales_train_validation.csv	30490	1919	30490 (No NaNs)
sales_train_evaluation.csv	30490	1919	30490 (No NaNs)
sell_prices.csv	6841121	4	6841121 (No NaNs)

Table 1: Overview of the datasets used in this project

Feature selection is crucial for model performance, but its effectiveness depends on the chosen methodology. Instead of pre-selecting ”relevant” columns, we will determine feature importance dynamically during model development. This may involve techniques such as lag features, rolling statistics, and categorical encodings.

To gain insights into the datasets, we performed exploratory data analysis (EDA). The following figures summarize key findings:

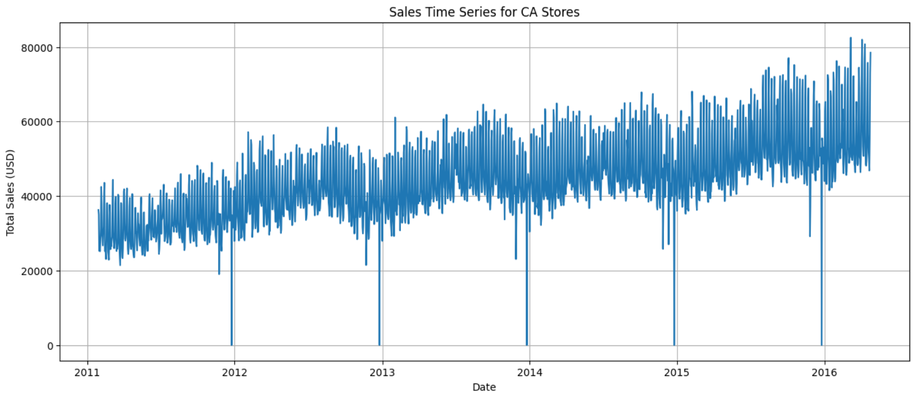

Figure 1: Time series of total sales in California stores over the dataset period. The trend shows seasonal fluctuations, periodic peaks, and occasional drops in sales.

Methodology

The objective of this project is to develop and evaluate machine learning models for time-series forecasting using the M5 competition dataset. Our initial plan is to explore three main approaches:

- **Transformer-Based Models**: Transformers have demonstrated superior performance in sequence modeling tasks, including time-series forecasting. We plan to experiment with architectures such as the Temporal Fusion Transformer (TFT) [4], which can handle multi-horizon forecasting and provide interpretability in feature importance.
- **Traditional Statistical Models**: We will explore classical time-series forecasting methods, including:
  - **Exponential Smoothing (ETS)**: A well-established method for capturing trends and seasonality in time series data [1].
  - **ARIMA (AutoRegressive Integrated Moving Average)**: A widely used statistical model effective for capturing linear dependencies in time-series data.
  - **Prophet**: A forecasting model developed by Facebook, designed to handle missing data, seasonal effects, and trend shifts [2].
- **Feature Engineering for Forecasting**: Since feature engineering plays a crucial role in model performance, we plan to create:
  - Lag-based features to capture past dependencies.
  - Rolling window statistics (mean, median, standard deviation) to summarize historical trends.

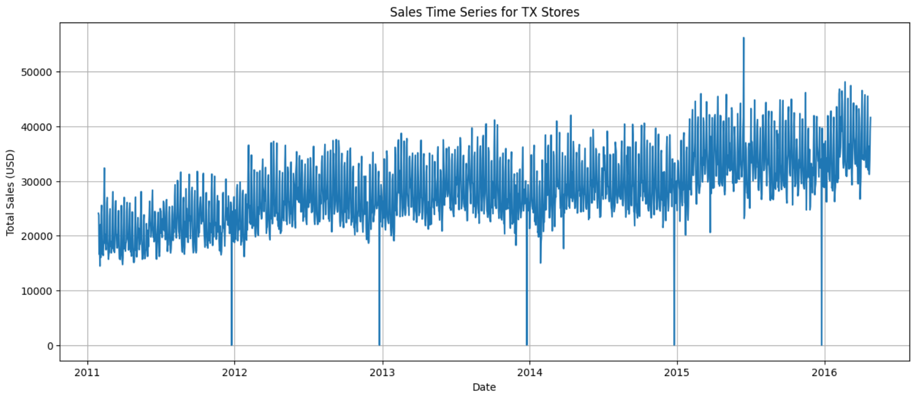

Figure 2: Time series of total sales in Texas stores over time. Similar to California, Texas shows seasonal variations and noticeable peaks, reflecting retail demand cycles.

  - Categorical encodings for event-based and store-specific features.
  - Hierarchical aggregation of sales data to model dependencies at different levels.

Flexibility in Method Selection: The final choice of models and techniques will depend on several factors, including:

- The quality and predictive power of each method on validation data.
- Computational feasibility, given that some deep learning models require extensive resources.
- Model interpretability, which is crucial for understanding feature importance in forecasting.
- Performance metrics such as RMSE (Root Mean Squared Error) and MASE (Mean Absolute Scaled Error).

Thus, while we have outlined an initial methodology, our final approach may change based on experimental results, computational limitations, and insights gained from feature engineering.

Deliverables

This project will produce the following key deliverables:

- **Comparative Study of Forecasting Methods**: A performance comparison of Transformer-based models (such as Temporal Fusion Transformer), traditional statistical models (ETS, ARIMA, Prophet), and feature engineering techniques.

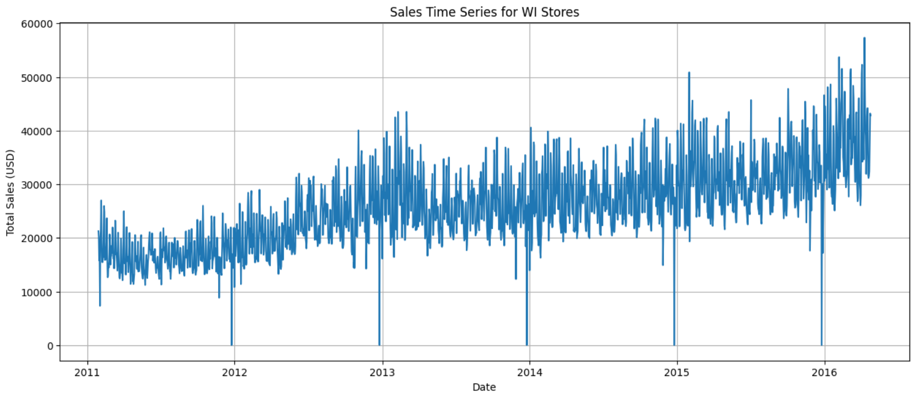

Figure 3: Time series of total sales in Wisconsin stores. The pattern follows similar seasonal variations seen in other states, highlighting consistent retail trends across regions.

- **Trained Models and Evaluation**: Implementations of selected models trained on the M5 dataset, evaluated using RMSE and MASE, with insights into feature importance and interpretability.
- **Visualization and Interpretability**: Graphical representations of predictions, error distributions, and time-series trends for better model understanding.
- **Code Repository**: A publicly available GitHub repository containing:
  - Model implementations and preprocessing scripts.
  - A structured notebook for reproducibility.
- **Final Report**: A structured summary of methodology, results, and best practices in time-series forecasting.

These deliverables ensure transparency, reproducibility, and valuable insights into forecasting performance.

Link to GitHub Repo

The project's code, datasets, and implementation details will be maintained in the following GitHub repository:
GitHub Repository: https://github.com/Legacy560/MTLSA25-Sales-Forecasting-Project

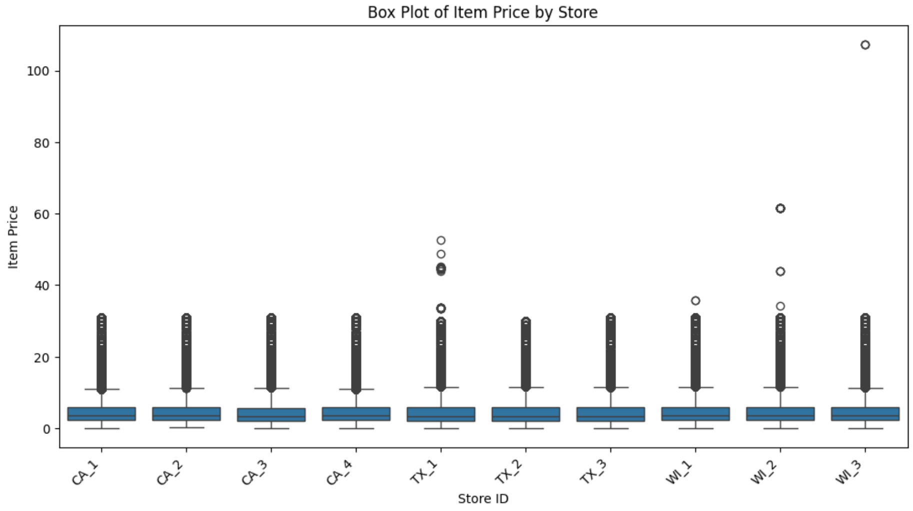

Figure 4: Box plot showing the distribution of item prices across different stores. The median price remains consistent across locations, but outliers are present, indicating varying price ranges for certain items.

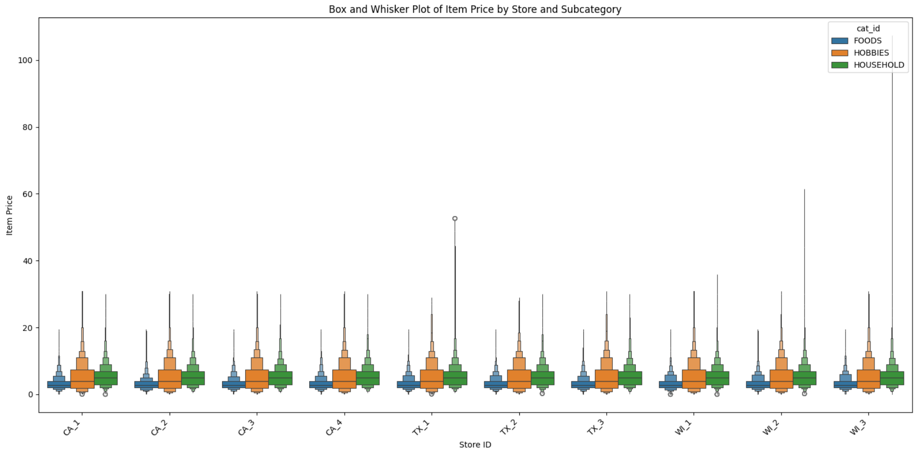

Figure 5: Box and whisker plot of item prices across stores and subcategories. Price distributions vary across product categories, with food items generally having lower median prices than household and hobby items.

Bibliography

References

[1] Hyndman, R. J., & Athanasopoulos, G. (2018). *Forecasting: Principles and Practice* (2nd ed.). OTexts. Available at: https://otexts.com/fpp2/

[2] Taylor, S. J., & Letham, B. (2018). "Forecasting at scale." *The American Statistician*, 72(1), 37-45.

[3] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., & Polosukhin, I. (2017). "Attention is all you need." *Advances in Neural Information Processing Systems*, 30.

[4] Lim, B., Arik, S. O., Loeff, N., & Pfister, T. (2021). "Temporal fusion transformers for interpretable multi-horizon time series forecasting." *International Journal of Forecasting*, 37(4), 1748-1764.

[5] Bojer, C. S., & Meldgaard, J. P. (2021). "Feature engineering for time series forecasting: A review." *International Journal of Forecasting*, 37(1), 13-28.

[6] Kaggle. "M5 Forecasting - Accuracy Data." Available at: `https://www.kaggle.com/competitions/m5-forecasting-accuracy/data`.

[7] OpenAI. (2025). "ChatGPT (March 2024 version)." Retrieved from `https://openai.com/chatgpt`.