Retail Sales Forecasting for Walmart’s Hobbies Category

I. Introduction........3

Problem Statement and Objective........3

Methodology........4

Data Overview........4

II. Exploratory Data Analysis........4

Time-Series Trends & Moving Averages........4

ACF & PACF Plots........5

Decomposition Plot........5

Variance analysis........5

Power Spectrum Analysis........6

Correlation Heatmap on Rolling & Lagged Features (Store CA_1 as an example)........6

Impact of Events on Sales........7

Sales by Weekday (Store TX_1 as an example)........8

Sales by Store Trend........8

Sales by Store Correlation Analysis........9

III. Hypotheses and Assumptions........9

Data Assumption........9

Model Assumption........10

IV. Data Processing and Feature Engineering........11

Data Processing........11

Feature Engineering........12

V. Model 1: ETS + ARIMA........13

Model Overview and Jurisdiction........13

Trade-Offs........13

Model Results........13

Model Diagnostics........14

VI. Model 2: SARIMAX........15

Model Overview and Jurisdiction........15

Trade-Offs........15

Model Results........16

Model Diagnostics........16

VII. Model 3: Prophet........17

Model Overview and Jurisdiction........17

Modeling Logic Explanation........ 18
Model Results........ 18
Model Diagnostics........ 19
VIII. Model 4: LightGBM........ 20
Model Overview and Jurisdiction........ 20
Modeling Logic Explanation........ 20
Model Results........ 21
IX. Model 5: LSTM........ 22
Model Overview and Jurisdiction........ 22
Modeling Logic Explanation........ 23
Model Results........ 24
X. Model Comparison and Discussion........ 25
XI. Future Improvement........ 26
Future Improvement on Models........ 26
Future Actions on the Business Side........ 26

I. Introduction

Problem Statement and Objective

Investors and lenders evaluating a retail company like Walmart rely on financial data to assess its stability and profitability. One of the most critical metrics is sales projection, as it directly impacts Walmart's ability to generate revenue, repay loans, and deliver returns to shareholders.

Accurate sales forecasting predicts future cash flows, aiding investment and lending decisions. For banks, understanding projected revenue streams helps assess Walmart's ability to meet debt obligations, ensuring timely loan repayments and managing credit risk. For investors, a reliable monthly sales forecast provides insights into Walmart's revenue trends, enabling better valuation analysis and helping them anticipate stock performance.

Traditional forecasting methods often focus on total sales, failing to capture seasonal demand shifts, category-specific trends, and structural patterns across different product lines. For example, a sudden increase in electronics sales may not indicate overall growth if grocery sales decline. A category-wise forecasting model addresses this limitation by capturing unique time series trends within each product category, offering a more granular and actionable sales projection.

To validate this approach, we first apply it to the Hobbies_1 category. Once refined, it can be extended across all categories, creating a scalable and adaptive forecasting solution. This allows banks to accurately evaluate Walmart's cash flow stability and adjust credit terms accordingly, while investors can use category-level sales trends to identify growth areas and assess market positioning. We aim to forecast unit sales for items in Walmart's Hobbies_1 department using a hierarchical dataset that includes daily sales, pricing, promotions, and other key features. Our primary objective is to generate accurate 28-day point forecasts for the period April 25, 2016, to May 22, 2016. By incorporating these factors, we enhance forecast precision, enabling more data-driven investment and lending decisions.

To measure the model's effectiveness, we establish the following benchmark:

- Achieve a Mean Absolute Error (MAE) of 200 units at the category level.
- Alternatively, ensure Root Mean Squared Error (RMSE) remains below 200 units to minimize large forecasting deviations.

By enhancing sales prediction accuracy, this model strengthens financial forecasting, allowing investors and lenders to make more informed decisions about Walmart's risk profile, investment potential, and creditworthiness.

Methodology

Our approach integrates traditional time series models such as ARIMA and SARIMAX with advanced machine learning techniques, Long Short-Term Memory (LSTM) networks, Prophet, and LightGBM. This hybrid modeling approach allows us to effectively capture both linear trends and intricate seasonal patterns within the dataset, resulting in highly accurate and actionable forecasts. The resulting predictions will be critical for strategic inventory management and resource allocation. By anticipating demand fluctuations accurately, Walmart can improve its operational efficiency and reduce waste.

Data Overview

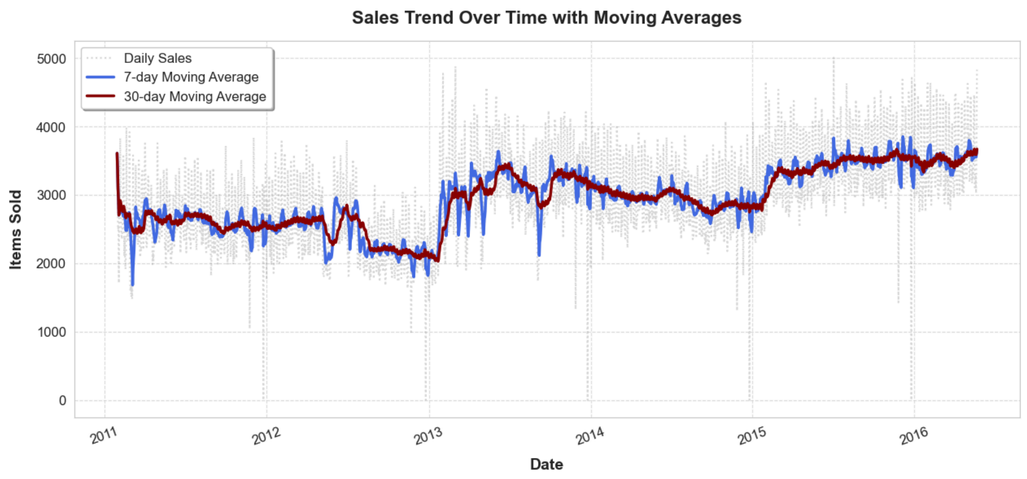
This dataset contains historical sales records from 45 Walmart stores across various regions. It includes detailed information on sales performance at the department and item levels from 2011 to 2016. Here is a general view of features category:

- Time-Series Data: Daily sales records for multiple products over several years
- Store & Product Details: Information on different Walmart stores and product categories
- Price Data: Historical pricing information for various products
- Calendar Data: Includes events, holidays, and other factors influencing sales trends

II. Exploratory Data Analysis

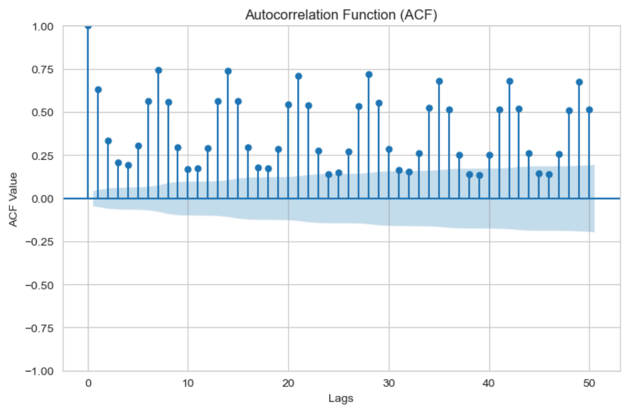
Time-Series Trends & Moving Averages

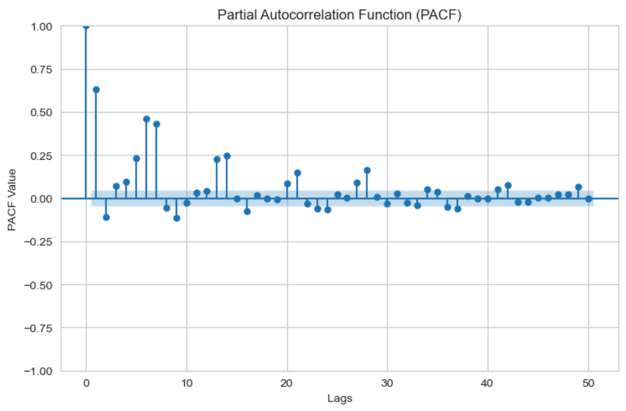
From 2011 to 2016, Walmart's sales showed steady growth with periodic fluctuations. The 7-day moving average highlights weekly seasonality, while the 30-day captures broader monthly trends. These patterns confirm recurring demand cycles and an overall upward trajectory.



ACF & PACF Plots

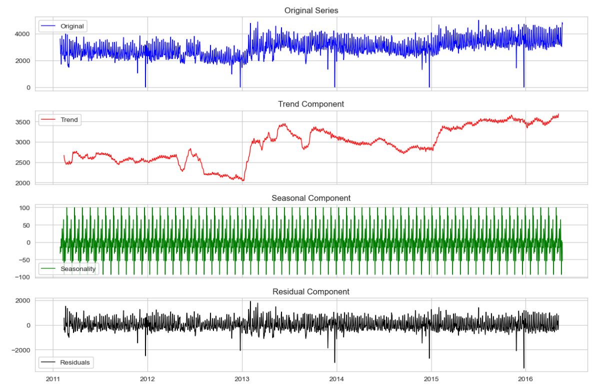
The ACF plot shows strong autocorrelation at multiple lags, indicating a persistent pattern and potential seasonality in the data. The PACF plot has significant spikes at the first few lags before tapering off, suggesting that an AR model of low order may be appropriate. The combination of these patterns implies that the time series likely follows an ARMA or seasonal ARIMA structure.

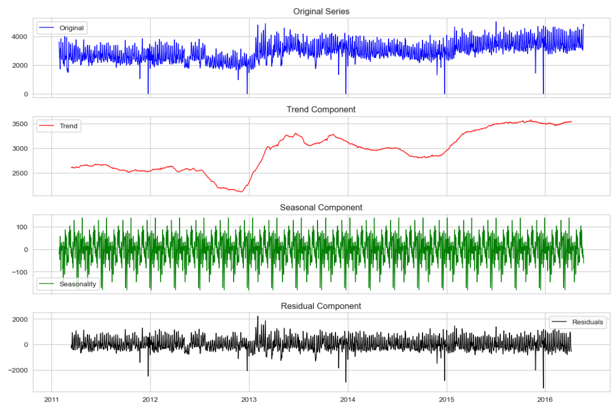




Decomposition Plot

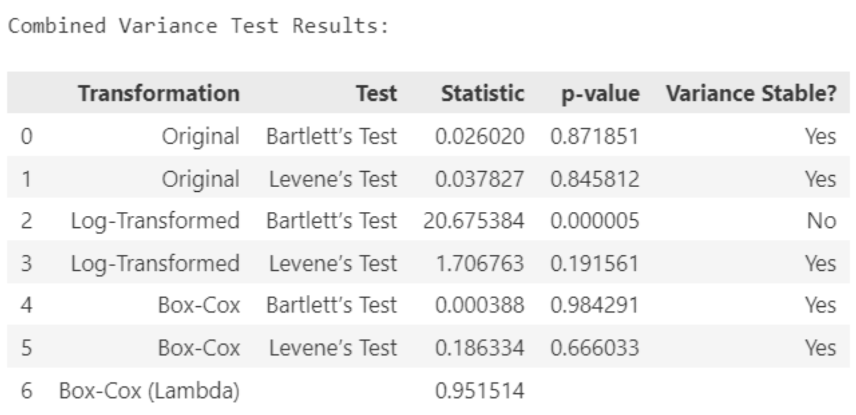
The sales data exhibits strong weekly and monthly seasonality, with repeating cycles confirming a regular pattern in sales volumes. Alongside these cyclical trends, there is an overall growth trajectory, although some periods show slower expansion. Additionally, residual spikes suggest the presence of external events or promotions that are not fully captured by basic trend and seasonality models.





Variance analysis

The variance stability tests indicate that the original data has stable variance, as confirmed by both Bartlett’s and Levene’s tests. However, the log transformation introduces instability, as Bartlett’s test reports a significant p-value, while Levene’s test suggests otherwise. In contrast, the Box-Cox transformation effectively stabilizes variance, with both tests yielding high p-values, confirming its suitability for variance normalization. The Box-Cox lambda value of approximately 0.95 suggests minimal transformation was needed.

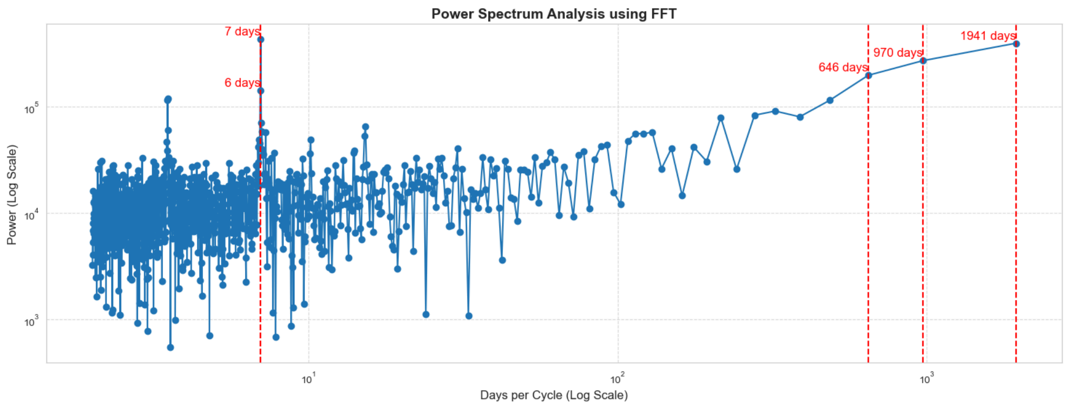


```
Combined Variance Test Results:
```

	Transformation	Test	Statistic	p-value	Variance Stable?
0	Original	Bartlett's Test	0.026020	0.871851	Yes
1	Original	Levene's Test	0.037827	0.845812	Yes
2	Log-Transformed	Bartlett's Test	20.675384	0.000005	No
3	Log-Transformed	Levene's Test	1.706763	0.191561	Yes
4	Box-Cox	Bartlett's Test	0.000388	0.984291	Yes
5	Box-Cox	Levene's Test	0.186334	0.666033	Yes
6	Box-Cox (Lambda)		0.951514		

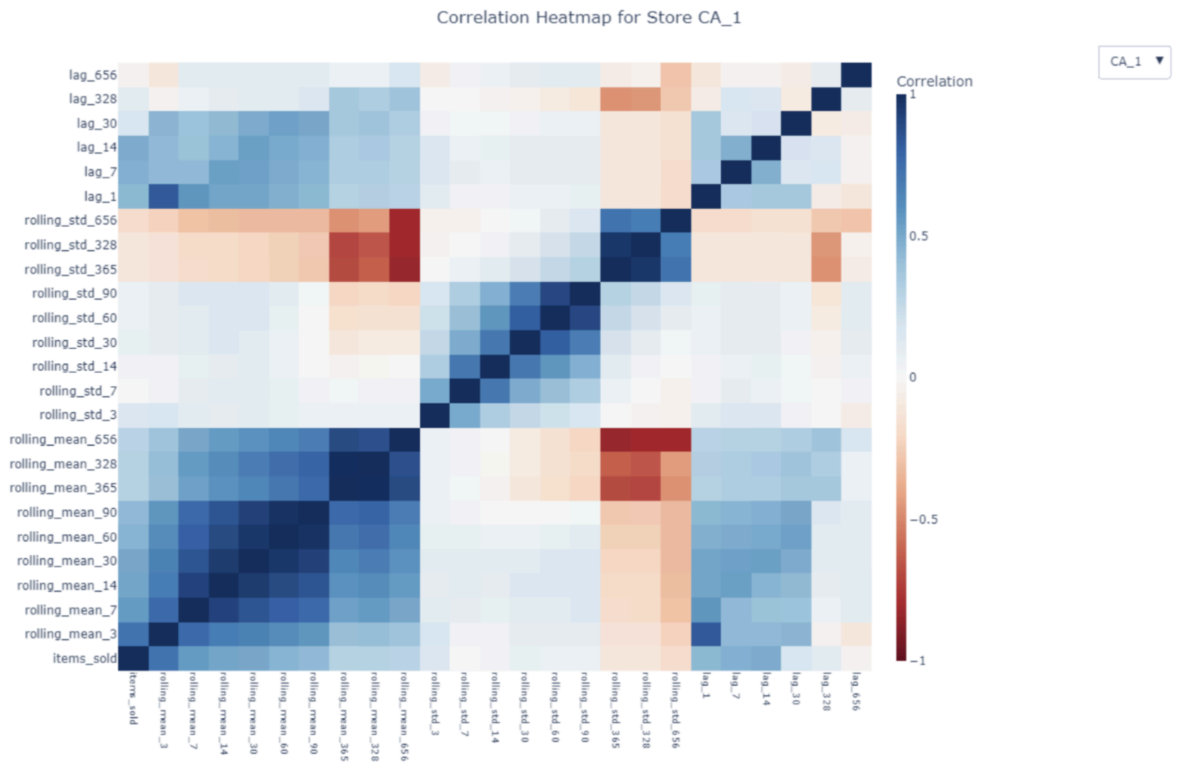
Power Spectrum Analysis

Power spectrum analysis reveals multiple strong periodic components, with each peak representing a dominant cycle in the data. These findings suggest the presence of recurring patterns that can inform model selection and feature engineering. To leverage these insights, creating lagged features can help capture dependencies over time, while using Prophet can effectively model both yearly and weekly seasonality, ensuring better trend and seasonality representation.



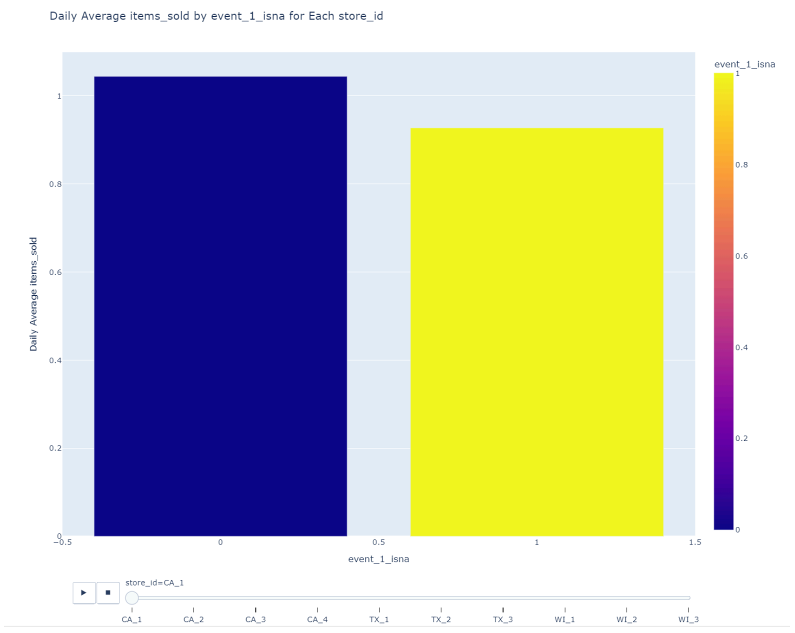
Correlation Heatmap on Rolling & Lagged Features (Store CA_1 as an example)

Lag and rolling feature correlations indicate that specific lags, such as lag_7, and rolling means, like rolling_mean_7, show strong correlations, reflecting weekly patterns. Additionally, some rolling standard deviation features, such as rolling_std_14 and rolling_std_30, exhibit overlapping information. While these features effectively capture seasonality, excessive redundancy may lead to overfitting.



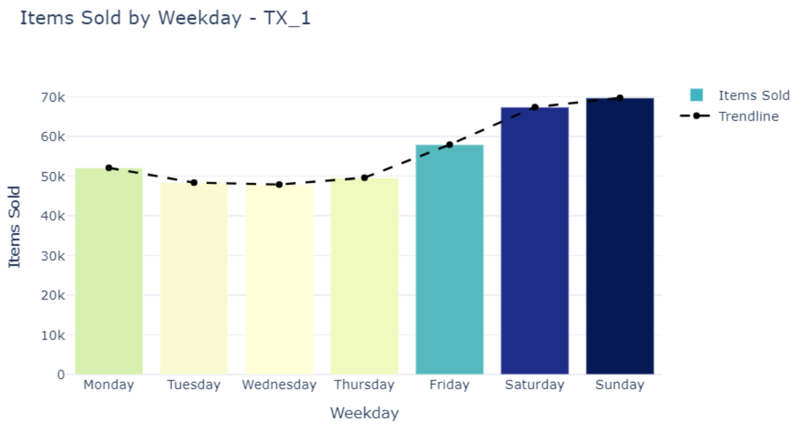
Impact of Events on Sales

Sales patterns differ between event and non-event days, with special events or promotions causing significant shifts in sales. The impact varies across stores, suggesting the influence of regional or demographic factors. To improve forecasting accuracy, incorporating holiday or event-based features is essential, as these days often lead to sales spikes or dips that deviate from normal trends.



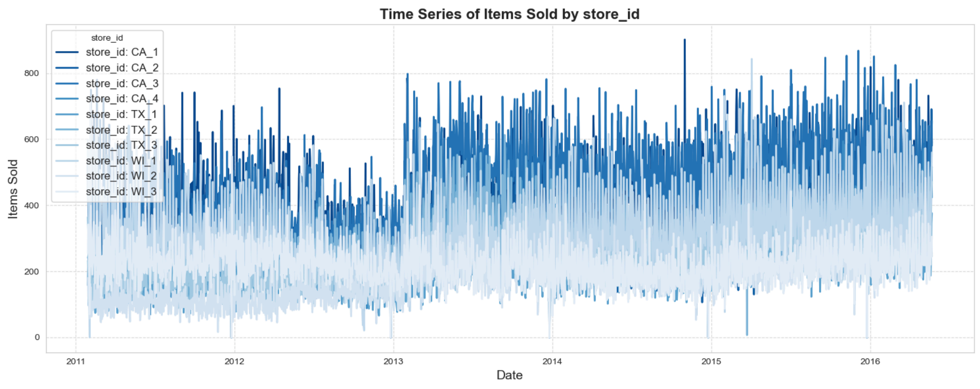
Sales by Weekday (Store TX_1 as an example)

Weekly sales patterns reveal higher sales on weekends, particularly on Saturdays, compared to certain weekdays like Mondays. Trendlines indicate a gradual increase in sales over the week. To account for these intra-week fluctuations, incorporating weekday features in forecasting models is crucial for accuracy.



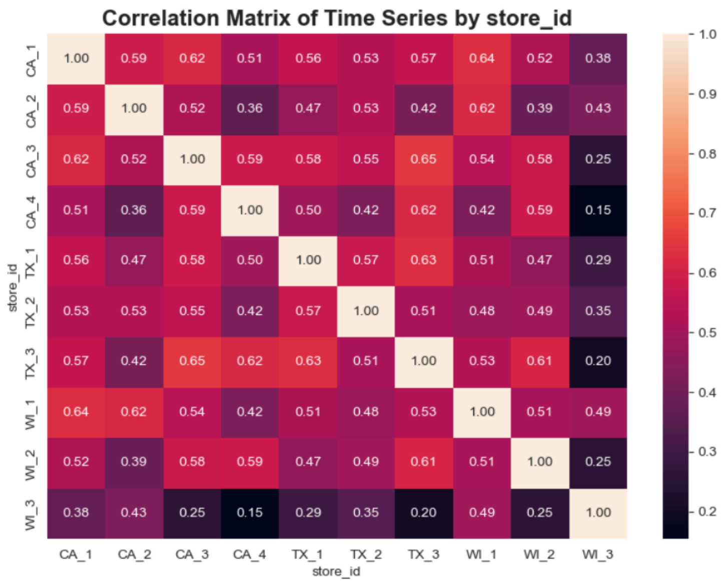
Sales by Store Trend

The time series plot confirms that grouped products across stores follow similar trends and seasonal patterns, supporting the assumption of homogeneity. Despite variations in sales volume, the consistent peaks and troughs indicate shared demand cycles. This validates aggregation within product categories, ensuring essential patterns are captured while reducing noise for effective forecasting.



Sales by Store Correlation Analysis

The correlation matrix demonstrates strong relationships among several stores, indicating that aggregated sales effectively capture common trends while reducing random fluctuations. High correlation ensures that combining sales data preserves essential demand patterns without losing critical signals. While some regional variations exist, the overall consistency supports the validity of aggregation for improving forecasting accuracy.



III. Hypotheses and Assumptions

Data Assumption

Due to the large volume of data, sales are modeled by aggregating products within the same category and store. This aggregation relies on several assumptions: first, that grouped products share similar underlying trends and seasonal behaviors (assumption of homogeneous patterns); second, that there is high correlation among individual products within each category. High correlation ensures that aggregating these products effectively smooths random fluctuations without losing critical signals. Prior exploratory data analysis (EDA) has confirmed that these assumptions are valid, demonstrating consistent patterns and high correlations within categories, making aggregation appropriate for accurate and robust modeling.

Model Assumption

ETS: By using exponential smoothing with additional trend and seasonality, we assume that the underlying sales process can be decomposed into level, trend, and recurring seasonal components. Using damped_trend=True implies the belief that any sales increase or decrease will not continue indefinitely but rather taper off gradually.

Hybrid ETS + ARIMA: We hypothesize that combining the strengths of ETS (capturing the main trend and seasonality) with an ARIMA model on the residuals may provide better forecasting performance. The ARIMA component is thought to capture any remaining autocorrelation not explained by the ETS seasonality or trend.

SRIMAX: For the SARIMA modeling part**,** we assume that sales exhibit a weekly seasonality, meaning similar sales patterns occur on the same day each week. To capture this effect, we use S=7 in the model.

In terms of external factors, we assume that exogenous variables such as weekends, past sales (Sales_lag7), rolling averages (Sales_rolling_mean_7), and holidays significantly influence sales. Additionally, we assume that the relationship between these variables and sales is linear and remains stable over time, with no sudden structural breaks in their impact.

Prophet: Prophet assumes that the time series data can be decomposed into components like trend, seasonality, holidays or special events, and noise. It employs a piecewise linear or logistic trend model, assumes smooth changes in trends occurring at infrequent changepoints, and typically expects residuals to be independent and normally distributed. Seasonality in Prophet can be either additive or multiplicative, depending on how seasonal effects vary over time. This aligns with our time series dataset well since it also presents strong seasonality and incorporates underlying factors like holiday.

One key limitation of Prophet is that, despite its intuitive structure and ease of interpretation, its simplicity can restrict its flexibility and predictive performance, particularly when dealing with complex datasets or highly nonlinear interactions that are not well-captured by its predefined trend and seasonality components.

LightGBM: GBM is non-parametric and does not rely on assumptions such as linearity, normality, or homoscedasticity. It presumes data observations to be independent and identically distributed, a condition often violated in raw time series data. Thus, careful feature engineering such as creating lagged variables or rolling aggregates is essential when applying LGBM to forecasting problems. One limitation of LightGBM is its reduced interpretability and the need for extensive feature engineering when modeling temporal dependencies.

LSTM: The LSTM-based forecasting model operates under the assumption that sequential dependencies in sales history provide meaningful patterns for predicting future demand. The use of Bidirectional LSTM layers assumes that both past and future relationships within a given sequence contribute to forecast accuracy. Additionally, the model relies on recursive forecasting, iteratively feeding predictions back as input for subsequent forecasts. This assumes that small errors will not accumulate significantly over time, maintaining forecast stability. The model is optimized using Mean Squared Error (MSE) loss, under the assumption that minimizing squared errors leads to the best predictive performance. However, this approach may result in bias toward moderate values, potentially underestimating extreme peaks and drops in sales. The LSTM's ability to generalize is contingent on the assumption that historical training data adequately represents future demand trends, meaning that seasonal patterns, pricing effects, and holiday-driven fluctuations will persist similarly in the future.

IV. Data Processing and Feature Engineering

Data Processing

Anomaly Detection: During the data exploration process, we first noticed that the prices of some products were extremely high. Given that this information may be indicative of the model (for example, extreme prices may occur during promotions or holidays), we ultimately chose to retain these outliers without correcting them. In addition, we also found that the sales data on some dates were abnormal: for example, the sales on the 25th were 0, while the sales on the 26th increased sharply. After comparing the holiday information, it is speculated that this is closely related to reasons such as the Christmas holiday (the store was closed on the 25th, resulting in 0 sales, and the sales surged after opening the next day).

Cleaning & Imputation: We used different strategies for handling missing values. For some missing values, we judge their importance in the overall modeling based on business logic. If the missing ratio is too high and has no obvious value for subsequent analysis, delete it directly; if the missing part can be supplemented by time series or adjacent values, the forward-fill and back-fill methods are used to fill in the values before and after the time series in each "store-product" grouping to ensure the continuity of sales price and sales volume data.

We also filled in missing values for lag and rolling features. New lag features (such as sales_lag7, sales_lag28) and rolling mean features (such as sales_roll7, sales_roll28) may generate NaN at the boundary; to prevent excessive interference to model training, we uniformly fill these NaN values with 0.

Transformations: First, we transform the original wide table into a long table (melt) to facilitate merging with calendar (calendar) and price (sell_prices) data. Then sort the data based on date and extract time features such as day_of_week and month. Given that we observed sales anomalies related to holidays, we retained this feature difference in the conversion process and took variables such as "is_holiday" into account in subsequent models. After that, we aggregated the data from the individual product level to the department level, retaining only one row for each date, while retaining the key exogenous variables mentioned above (such as holiday information, price trends, etc.). Finally, we divided the data into training and test sets according to the set time points to evaluate the predictive performance of the model.

Feature Engineering

Our EDA insights have guided our feature engineering:

- Lagged & Rolling Features: Use lags (e.g., lag_7) and rolling averages/stds to capture recurring weekly patterns and volatility. Though PCA might be needed to reduce redundancy.
- Cyclical Transformations: Create sine and cosine features to represent 6–7 day cycles.
- Event-Based Features: Add binary flags or dummy variables for holidays and promotions to capture event-driven effects.
- Distribution Transformations: Apply log transformation to stabilize variance in right-skewed sales data.
- Weekday Patterns: Include day-of-week as a categorical variable to capture differences between weekdays and weekends.

Feature Category	**New Feature**	**Description**	**Model Application**
Lagged Sales Features	7-day Lagged Sales	Captures short-term sales trends.	LSTM, ETS+ARIMA, LGBM, SARIMAX
Lagged Sales Features	14-day Lagged Sales	Captures medium-term sales trends.	LSTM, LGBM
Lagged Sales Features	28-day Lagged Sales	Captures long-term sales trends.	ETS+ARIMA, LGBM
Rolling Sales Features	7-day Rolling Mean	Smooths short-term sales fluctuations, reducing noise.	LSTM, ETS+ARIMA, LGBM, SARIMAX
Rolling Sales Features	14-day Rolling Mean	Smooths medium-term sales fluctuations.	LSTM, LGBM
Rolling Sales Features	28-day Rolling Mean	Smooths long-term sales fluctuations.	ETS+ARIMA, LGBM
Price Features	Price Change	Monitors daily fluctuations in selling price.	LSTM, ETS+ARIMA
Price Features	Price Momentum	Standardizes current selling price relative to historical maximum.	LGBM
Price Features	Normalized Price	Detects price momentum—whether prices are rising or falling relative to history.	LGBM, LSTM
Price Features	Maximum Historical Price	Maximum selling price in the last one year and one day.	LGBM

Temporal & Event-Based Features	Is Weekend	Identifies weekends (Saturday & Sunday).	LSTM, ETS+ARIMA, LGBM, SARIMAX
Temporal & Event-Based Features	Is Holiday	Identifies holidays.	ETS+ARIMA, SARIMAX
Sales Variability Features	Sales Variability (Std Dev)	Measures sales variability over the past 30, 60, 90, and 7 days.	LGBM

V. Model 1: ETS + ARIMA

Model Overview and Jurisdiction

By first fitting an ETS model to capture level, trend, and seasonality, we account for the primary structure of the time series. We then apply ARIMA to the residuals to capture any remaining autocorrelation or patterns not explained by the ETS framework. This “two-stage” approach often produces more robust forecasts if the data exhibit both systematic seasonal patterns (captured by ETS) and more subtle autocorrelation (captured by ARIMA).

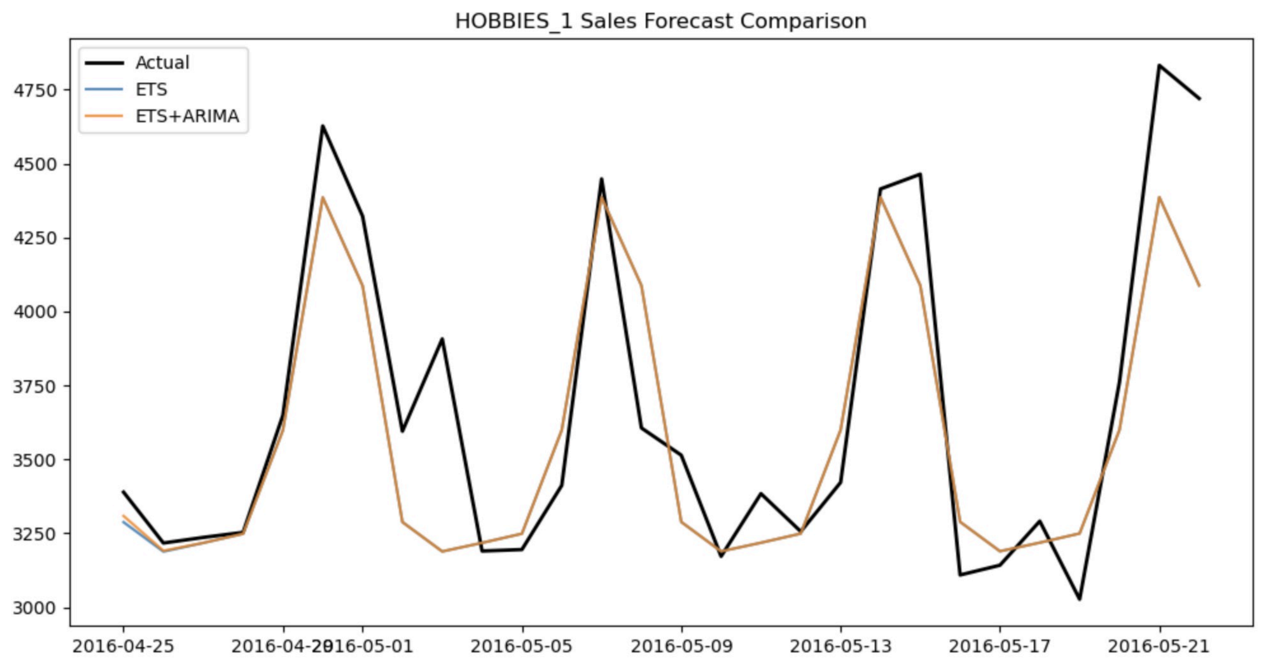
Trade-Offs

This model requires slightly higher modeling and computational complexity because maintaining two models (ETS and ARIMA) and combining forecasts can be more complex than a single approach. There is a risk of overfitting if both models are too complex and the time series are short or noisy.

Model Results

In general, the ETS+ARIMA model follows the main peaks and troughs of the sales data very closely, reflecting trend and seasonality. However, it does show some smoothing compared to the actual data, which can result in underestimation of the largest fluctuations.This flattening is to be expected to some extent, as real-world sales will contain noise that the model cannot track. However, the inclusion of exponential smoothing (ETS) and the ARIMA autoregressive components helps to model short-term dynamics more effectively than applying ETS alone. Inclusion of other exogenous variables can improve such forecasts to the extent of modeling external events that cause sudden changes in sales patterns.

RMSE	MAE	MAPE
265.53	187.57	4.8%



Model Diagnostics

The Ljung-Box test results suggest that the residuals from the ETS + ARIMA model still contain significant autocorrelation, meaning the model hasn't fully captured all the patterns in the data. At short-term lags (1-10), the p-values are extremely low, indicating that immediate dependencies remain. This suggests that the ARIMA component may need refined short-term lag selection or additional differencing to better remove trends.

One of the key concerns is Lag 7, where the test statistic is high, confirming that weekly seasonality is not fully captured. Since the model was built with a seasonal order of (0,0,0,7), this may not be sufficient to handle the recurring weekly fluctuations in sales. Adjusting the seasonal parameters or incorporating external variables, like promotions or holidays, using SARIMAX could help improve accuracy.

At longer lags (28, 30, 60, and 90), the presence of strong autocorrelation suggests that multi-week and quarterly trends are not being fully accounted for. This could mean that the model isn't capturing long-term seasonality, which is crucial for forecasting shifts in consumer demand. Since traditional ARIMA models struggle with multiple seasonal patterns, switching to TBATS or Prophet, which are designed for complex seasonality, might yield better results.

Lag	Ljung-Box Statistic	P-Value	Significant
1	32.03231	1.52E-08	Yes

2	32.74027	7.77E-08	Yes
3	32.88197	3.41E-07	Yes
4	34.58511	5.65E-07	Yes
5	43.25588	3.28E-08	Yes
6	59.12357	6.78E-11	Yes
7	60.78048	1.05E-10	Yes
8	72.29636	1.71E-12	Yes
9	76.22826	9.04E-13	Yes
10	83.23853	1.16E-13	Yes
14	87.58202	1.08E-12	Yes
20	94.47216	1.21E-11	Yes
28	116.6653	8.63E-13	Yes
29	124.4908	8.47E-14	Yes
30	132.5833	7.43E-15	Yes
60	177.988	1.27E-13	Yes
90	217.62	1.43E-12	Yes

VI. Model 2: SARIMAX

Model Overview and Jurisdiction

The SARIMAX model was chosen as a primary forecasting approach because it can capture both historical sales patterns and external influences. Unlike traditional ARIMA, SARIMAX can include exogenous variables such as weekends, past sales (Sales_lag7), and rolling averages (Sales_rolling_mean_7). Given the weekly seasonality (S=7) observed in the sales data, SARIMAX's ability to model seasonal patterns ensures more accurate predictions by accounting for recurring fluctuations.

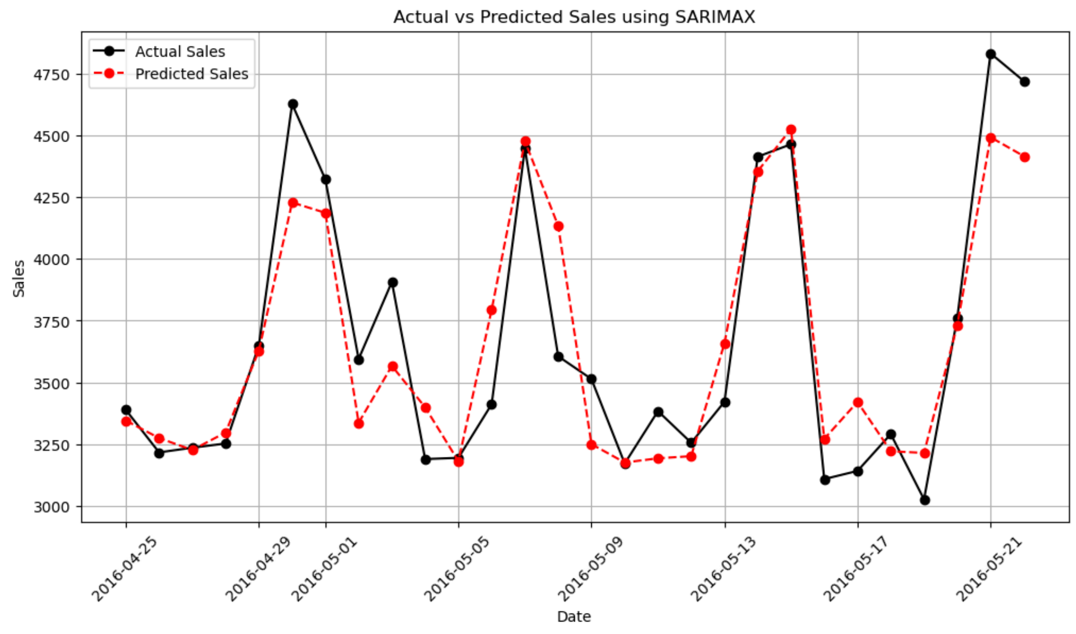
Trade-Offs

SARIMAX assumes linear relationships between exogenous variables and sales, which may not always hold—certain factors such as promotions or holidays could have non-linear effects on demand. For example, a major holiday promotion might lead to a sudden spike in sales that is different from typical seasonal patterns.

Model Results

The ARIMAX model performs well in capturing general trends and seasonality but has some limitations in predicting short-term fluctuations. Some points show a visible gap between actual and predicted sales, suggesting areas where the model struggles to capture unexpected variations. Below are the evaluation metrics:

RMSE	MAE	MAPE
246.07	187.22	4.91%



Model Diagnostics

The Ljung-Box test results indicate that while short-term residual autocorrelation has been reduced, there are still significant dependencies at multiple lags; at Lag 1 and 4, there is no significant autocorrelation, which is a good sign for short-term forecasting. However, Lags 2, 3, and 5 still show significant autocorrelation, suggesting that some short-term dependencies remain, indicating the need for potential refinements in the ARIMA parameters, such as adjusting the AR (p) and MA (q) terms.

The most concerning issue appears in seasonal and long-term lags, particularly at Lag 7 (weekly seasonality) and beyond (lags 10, 14, 20, 28, 30, 60, and 90). The extremely low p-values at these points suggest strong autocorrelation in the residuals, meaning the SARIMAX model is not fully capturing the weekly, multi-week, or long-term cycles in the sales data. This could indicate that the seasonal order (P,D,Q,s) needs to be revised, or that additional seasonal patterns beyond weekly cycles exist. Therefore, further adjustments to seasonal parameters and external variable inclusion are needed, or an alternative approach like TBATS or a hybrid model should be considered to improve forecast accuracy.

Lag	Ljung-Box Statistic	P-Value	Significant
1	3.287477	0.06981	No
2	7.811161	0.020129	Yes
3	8.706573	0.033458	Yes
4	9.45383	0.050705	No
5	11.37624	0.04441	Yes
6	45.70806	3.38E-08	Yes
7	45.77844	9.65E-08	Yes
10	50.7793	1.92E-07	Yes
14	85.43283	2.75E-12	Yes
20	106.8783	7.26E-14	Yes
28	136.8898	2.69E-16	Yes
30	143.9266	7.91E-17	Yes
60	196.942	1.75E-16	Yes
90	244.9809	2.79E-16	Yes

VII. Model 3: Prophet

Model Overview and Jurisdiction

Prophet is an open-source forecasting tool developed by Facebook (now Meta) designed specifically for time series prediction, especially for datasets with strong seasonality, missing data, and outliers. It employs an additive regression model, decomposing data into three key components:

- Trend – Long-term, non-periodic changes.
- Seasonality – Regular daily, weekly, or yearly patterns.
- Holidays & Events – External factors like holidays and promotions.

Prophet explicitly incorporates holidays, making it well-suited for our dataset's holiday event columns. Power Spectrum Analysis indicates strong seasonalities, which Prophet effectively captures through its default yearly and weekly seasonality models while also allowing custom seasonalities. Our data exhibits multiple seasonal patterns, and Prophet's ability to automatically detect and model multiple seasonalities ensures that it can effectively handle these complexities. Additionally, it is robust to missing values and outliers, ensuring forecast stability.

Modeling Logic Explanation

The Prophet requires parameters such as *n_changepoints* and *holidays_prior_scale* and so on. To select the best parameters, we implemented grid search for hyperparameter optimization. We used a time-series cross-validation approach to ensure models are trained only on past data, preventing data leakage; Additionally, the model optimization uses Tweedie deviance as the loss function because: 1) It effectively handles non-negative and skewed data, aligning with our findings from EDA. 2) It ensures that forecasts remain positive, which is essential for sales and revenue predictions where negative values are not meaningful. 3) Since Tweedie deviance emphasizes differences in magnitude, it prevents the model from over-prioritizing low-value sales and ensures that large sales values are forecasted more accurately.

The final optimized parameters favor a model with many potential changepoints (50) but a conservative approach to applying them (low changepoint prior scale of 0.01). The model also incorporates multiplicative seasonality and a relatively high seasonality prior scale (10.0).

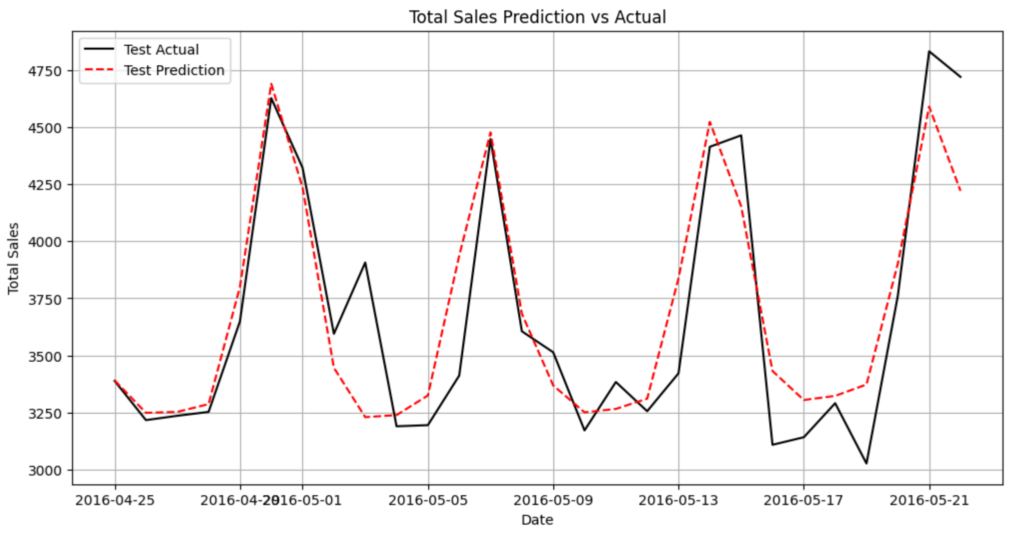
```python
best_params = {
    "n_changepoints": 50,
    "holidays_prior_scale": 1.0,
    "changepoint_range": 0.8,
    "seasonality_mode": "multiplicative",
    "seasonality_prior_scale": 10.0,
    "changepoint_prior_scale": 0.01
}
```

During the model training process, data is grouped by store to create store-level forecasts. To enhance efficiency, parallel processing is used to enable the simultaneous training of multiple store models.

Model Results

The model catches the general trend well but tend to be more flat than the actual sales data; this is expected since actual sales tend to have more noise; It would be more helpful to incorporate exogenous variables to capture more nuances; here are the evaluation metrics:

RMSE	MAE	MAPE
248.14	178.69	4.86%



Model Diagnostics

The Ljung-Box test results indicate that the residuals from the Prophet model exhibit significant autocorrelation at all tested lags (p-value = 0 for most lags). This suggests that the Prophet model has not fully captured the underlying time series structure, leaving patterns in the residuals that should ideally be modeled. In the future,we might need to include external regressors or employ a hybrid approach (combined with machine learning models to adjust for residual errors) to improve accuracy.

Lag	Ljung-Box Statistic	P-Value	Significant
1	600.559278	1.27E-132	Yes
2	1029.380577	2.97E-224	Yes
3	1373.531929	1.63E-297	Yes
4	1692.533843	0	Yes
5	1948.53586	0	Yes
6	2166.679037	0	Yes
7	2405.658351	0	Yes
14	3884.771283	0	Yes
21	5366.083519	0	Yes
28	6641.569319	0	Yes
30	7106.61013	0	Yes
60	11746.36534	0	Yes
90	14307.48405	0	Yes

VIII. Model 4: LightGBM

Model Overview and Jurisdiction

LightGBM uses tree-based learning to iteratively improve weak learners and reduce errors. Its histogram-based learning method organizes continuous features into discrete bins, speeding up the computational process.

The strength is particularly suitable for our large dataset; For LightGBM, we are predicting by item type, which involves over 5,000 distinct items (resulting in 5,000 columns, with rows representing dates). Additionally, our sales data includes interactions between past sales, promotions, pricing, and holidays. LightGBM's tree-based approach effectively captures these nonlinear dependencies, outperforming traditional models. From our EDA, we identified correlations between event-based promotions and price features, making LightGBM well-suited for leveraging categorical encodings, lagged sales, rolling statistics, and holiday effects to enhance predictions.

Modeling Logic Explanation

For this model, the feature engineering strategy involves transforming raw sales data into predictive features through the following: First, I restructured the data from wide to long format and merged sales with calendar and pricing information to create a unified dataset with temporal context. To capture seasonal patterns, I developed calendar features including event indicators, weekend flags, and day-specific markers which help identify weekly shopping patterns and holiday sales spikes. For regional effects, I incorporated state-specific SNAP benefit indicators which reflect periods of increased purchasing power in specific regions.

The core of my approach involved time-series feature creation, generating lagged features at intervals from 1-365 days to capture various sales cycles (weekly, monthly, annual). I calculated rolling statistics (averages, standard deviations) across different windows to measure both trends and volatility. For pricing dynamics, I created normalized price features that express current prices relative to historical maximums and recent averages, capturing how price positioning affects purchasing behavior. Finally, I encoded categorical variables (store IDs, item IDs, etc.) into numerical representations suitable for modeling.

Next, to optimize LightGBM's hyperparameters, I chose Bayesian Optimization due to its probabilistic approach, which efficiently guides the search process and significantly improves tuning speed. This is particularly important because, in this model, predictions are not made at the store level; instead, each item is trained individually, requiring an efficient optimization strategy.

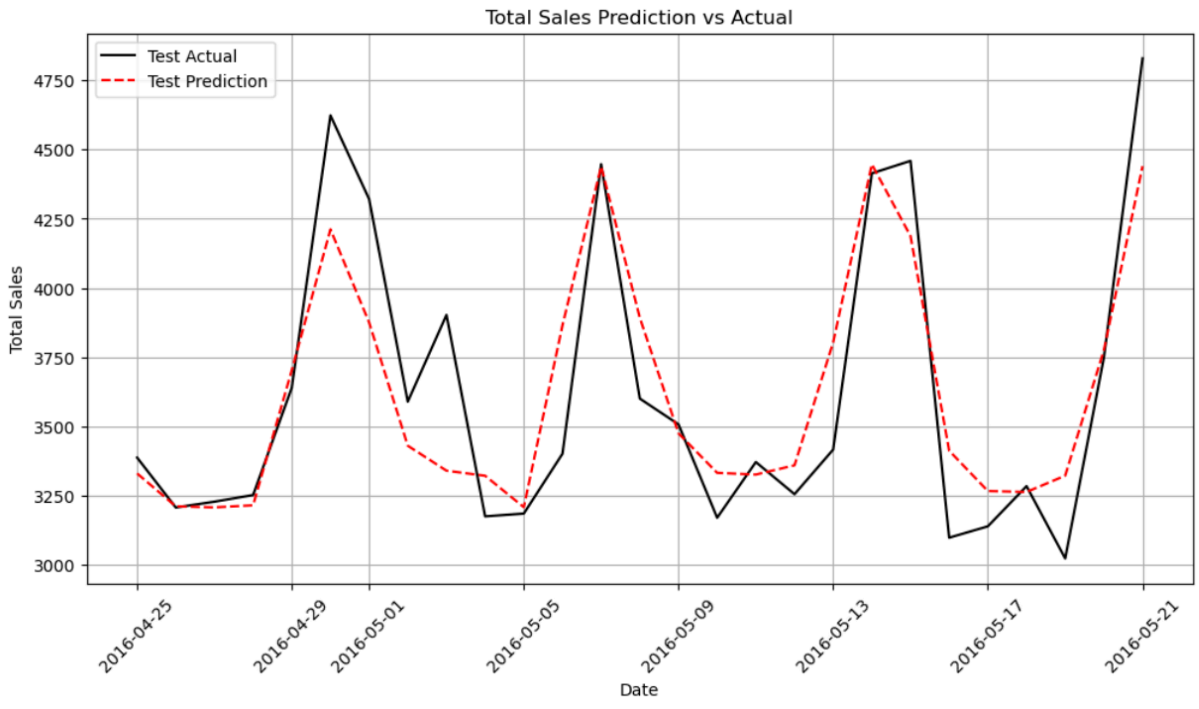
With the best parameters, I proceeded to the model-building stage, after which I used a recursive forecasting approach; The prediction process follows a rolling forecasting approach, where each day's forecast is generated sequentially while dynamically updating rolling statistics for future predictions. At each time step, the model first computes relevant features using historical sales and any previously predicted values. Since some features depend on past sales (e.g., rolling averages, lag statistics), they must be recalculated iteratively. After generating a prediction for a given day using the trained LightGBM model, the forecasted sales values are immediately updated in the dataset. This update ensures that subsequent days' predictions can incorporate the most recent estimated values, maintaining consistency in feature calculations. By continuously updating rolling statistics, such as moving averages and past sales-based features, the model effectively adapts to new trends, ensuring accurate and realistic forecasts over the prediction horizon.

```python
for i in range(num_dates):
    date = start_date + timedelta(days=i)
    # compute the features on-the-fly because some features depend on predictions
    X_features = rolling_updates(hobbies_sales_by_date, date=date)
    X_pred = X_features[X_features['date'] == date][feature_cols]
    # generate predictions
    y_pred = best_m.predict(X_pred) * scaling_factor
    # update predictions to the sales_by_date dataframe
    hobbies_sales_by_date.loc[hobbies_sales_by_date['date'] == date, 'items_sold'] = y_pred
```

Model Results

The model successfully captures the overall trend but fails to accurately predict the sharp increase in sales on Saturdays, despite including a binary feature for Saturdays during training. To address this, further improvements could focus on creating interaction features that combine Saturday-related variables with other influencing factors, such as promotions, holiday effects, or past sales patterns. This approach may enhance the model's ability to recognize and adjust for the recurring sales surges on Saturdays.

RMSE	MAE	MAPE
248.05	181.68	4.9%



IX. Model 5: LSTM

Model Overview and Jurisdiction

Our model leverages LSTM (Long Short-Term Memory) networks to forecast unit sales for Walmart's Hobbies_1 department, using daily sales, pricing, promotions, and external factors to generate accurate 28-day forecasts. LSTM is particularly effective in capturing long-term dependencies, seasonal trends, and nonlinear patterns, making it well-suited for forecasting retail sales fluctuations. By learning from historical data, the model helps inventory planners, supply chain operators, and operations managers optimize stock levels, prevent stockouts, and reduce overstocking costs. Unlike traditional models, LSTMs adapt dynamically to changing demand patterns influenced by price shifts, holidays, and promotions, ensuring more robust and data-driven inventory decisions.

The LSTM forecasting model supports retail analytics, demand planning, and supply chain optimization, providing Walmart with scalable and automated sales predictions. It is evaluated using Mean Absolute Percentage Error (MAPE), Mean Squared Error (MSE), and Directional Accuracy to ensure both low forecast error and accurate trend prediction. By integrating deep learning into inventory forecasting, Walmart can enhance operational efficiency, reduce waste, and align supply with demand more effectively, ultimately strengthening its inventory management and decision-making processes.

Modeling Logic Explanation

My LSTM-based forecasting strategy transforms sequential sales data into meaningful predictions by leveraging deep learning techniques optimized for time-series analysis. Before training, the data is normalized using MinMax scaling, ensuring numerical stability and preventing large magnitude differences between features from skewing the learning process. Key exogenous features, such as sell price, holiday/weekend indicators, and lagged sales statistics, provide additional context, allowing the model to incorporate business drivers affecting consumer behavior.The model is trained using a rolling window approach, where each input sequence consists of 28 days of historical data, predicting the next day's sales. This method enables the LSTM to capture long-term dependencies, short-term fluctuations, and seasonal demand patterns.

```python
def create_sequences(data, features, target, sequence_length=28):
    X, y = [], []
    for i in range(len(data) - sequence_length):
        X.append(data[features].iloc[i:i + sequence_length].values)  # Shape: (28, num_features)
        y.append(data[target].iloc[i + sequence_length])  # Next day's sales
    return np.array(X), np.array(y)

# Generate sequences
sequence_length = 28
X, y = create_sequences(dept_df, features, target, sequence_length)
```

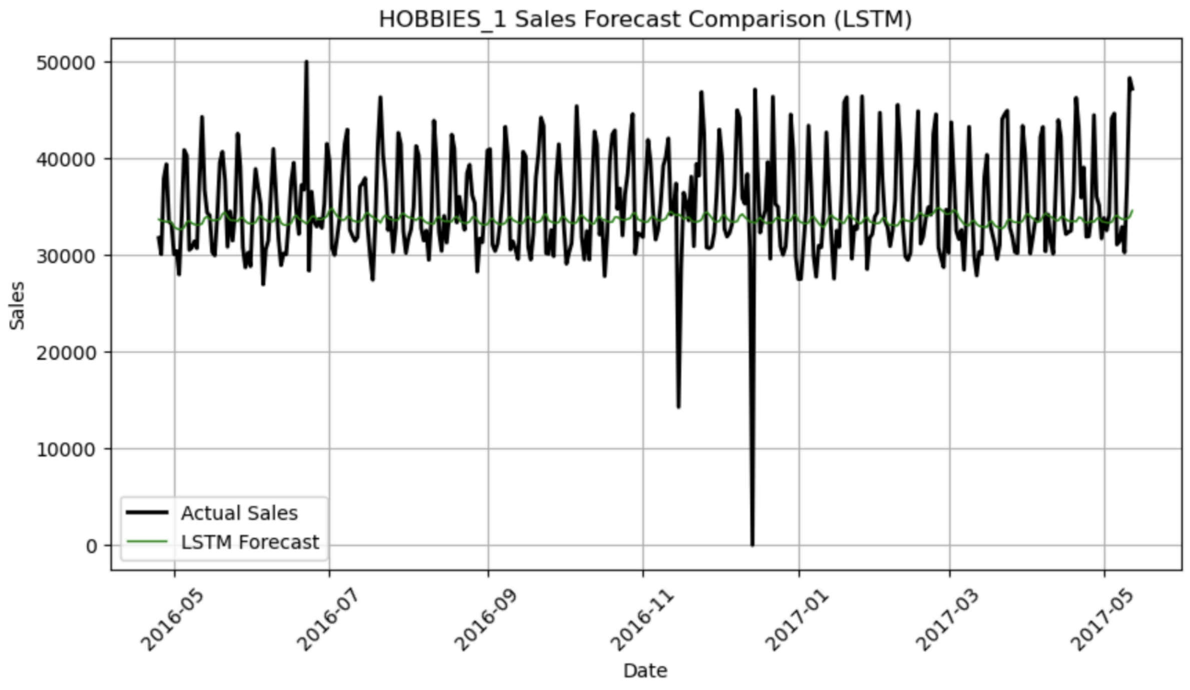
The LSTM model architecture consists of Bidirectional LSTM layers, which enhance predictive accuracy by capturing both forward and backward relationships within the data. Dropout layers are applied to prevent overfitting, ensuring the model generalizes well to unseen data. The model is trained using the Adam optimizer with Mean Squared Error (MSE) loss, balancing computational efficiency with learning stability. Once trained, the model generates forecasts using a recursive forecasting approach, where predictions for each day are iteratively fed back into the model to inform subsequent forecasts. During this rolling forecast process, lagged features and moving averages are dynamically updated, maintaining feature consistency across the prediction horizon. This approach ensures that Walmart's inventory and supply chain teams receive adaptive, real-time forecasts that effectively anticipate demand shifts and optimize inventory levels.

```python
model = Sequential([
    Input(shape=(sequence_length, len(features))),  # Ensure input shape is correct
    Bidirectional(LSTM(64, return_sequences=True)),  # First LSTM layer
    Dropout(0.2),
    Bidirectional(LSTM(32, return_sequences=False)),  # Second LSTM layer
    Dropout(0.2),
    Dense(25, activation="relu"),
    Dense(1)  # Predicts unit sales
])
```

Model Results

The LSTM-based forecasting model was evaluated on test data using Mean Squared Error (MSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). The model achieved an MSE of 31,373,455.26, indicating significant variance in its predictions. The MAE of 4,177.02 suggests that, on average, the model's predictions deviated by approximately 4,177 units from the actual sales values.

The first visualization comparing actual vs. predicted sales highlights that while the LSTM model captures some general trends, it struggles with sharp fluctuations and extreme variations in sales patterns. The second plot, illustrating the 28-day forecast, reveals a consistent downward trend in predicted sales, suggesting potential bias in the model's rolling predictions. This indicates that the LSTM may be over-relying on short-term patterns and failing to fully adjust to longer-term seasonal fluctuations. Moving forward, improving the model's feature selection, hyperparameter tuning, and training data balancing may enhance forecast accuracy and trend adaptation.



X. Model Comparison and Discussion

Since we are forecasting overall sales for a single category (Hobbies_1), we have chosen RMSE as the primary evaluation metric because it penalizes large errors more heavily, making it crucial for minimizing significant forecasting deviations that could impact financial decisions. MAE serves as a secondary metric as it is less sensitive to outliers and provides a more stable measure of average forecasting accuracy. Lastly, MAPE is not suitable for this use case since it becomes unreliable when sales values are close to zero, potentially distorting the evaluation.

Among the models, SARIMAX performs best in RMSE, while Prophet has the lowest MAE. We did not include LSTM in the comparison as it struggles with short-term fluctuations and extreme demand variations.

	RMSE	MAE	MAPE
ETS+ARIMA	265.53	187.57	4.80%
SARIMAX	246.07	187.22	4.91%
Prophet	248.14	178.69	4.86%
LightGBM	248.05	181.68	4.9%

The Ljung-Box test shows SARIMAX handles short-term trends best, with fewer autocorrelated lags than ETS+ARIMA and Prophet, though it still struggles with long-term patterns. ETS+ARIMA has moderate autocorrelation, making it a decent but not optimal choice. While the test isn't required for Prophet, it helps identify issues—Prophet performs the worst, likely due to oversmoothing. LightGBM doesn't need the test since it's tree-based and doesn't assume white noise residuals. Overall, SARIMAX is the best model for short-term forecasting, but adding exogenous variables like promotions or holidays could improve long-term accuracy. A hybrid approach may be needed for better overall performance.

	Short-Term Handling (Lags 1-10)	Long-Term Handling (Lags 14-90)	Overall Performance
ETS+ARIMA	Moderate autocorrelation issues	Poor at long-term cycles	Better than Prophet but weaker than SARIMAX

SARIMAX	Best (least autocorrelation)	Still struggles with long-term trends	Best for short-term forecasting
Prophet	Worst (severe short-term autocorrelation)	Worst at long-term patterns	Not recommended for this dataset
LightGBM	NA	NA	NA

XI. Future Improvement

Future Improvement on Models

To enhance the predictive accuracy and stability of the forecasting models, several refinements can be made; For LightGBM, since it fails to catch the surging sales on Saturdays, we can introduce more complex interactions between pricing, promotions, and seasonality, allowing the model to better understand demand drivers. Prophet struggles to capture sales fluctuations, likely due to insufficient changepoints. To improve responsiveness, we can increase the number of changepoints and include weekends in the "holiday" function, allowing the model to better account for periodic demand shifts.

Because ARIMA and SARIMA are designed to capture only one seasonality effect, we will incorporate alternatives like TBATS to improve since our time series presents multiple seasonality effects; TBATS's ability to use fourier terms help to model complex seasonal patterns.

The LSTM model exhibited forecast drift over the 28-day prediction horizon, indicating a need for improved model regularization and hyperparameter tuning. Optimizing key parameters such as learning rate, batch size, dropout rate, and sequence length using Bayesian Optimization or Grid Search could improve generalization and reduce overfitting to short-term patterns. Additionally, integrating attention mechanisms or transitioning to Transformer-based architectures may allow the model to focus on the most influential time steps, improving its ability to capture demand fluctuations.

Future Actions on the Business Side

Once the model reaches high accuracy for the Hobbies_1 category, we'll expand it to other product categories, building a more comprehensive sales forecasting system. This will give investors deeper insights into category-level growth trends and help lenders assess revenue stability with more precision.

By combining category-level forecasts, we can create a more accurate total sales projection than traditional top-down methods. This provides investors with a clearer view of Walmart's revenue trajectory, while lenders can use these projections to better estimate future cash flow and loan repayment capacity. With more reliable sales forecasts, investors can make better decisions about Walmart's profitability, stock performance, and long-term growth potential.

Beyond Walmart, the forecasting model can be adapted for other retailers, allowing investors and lenders to apply similar data-driven forecasting techniques when evaluating businesses in the retail sector. This approach enhances risk assessment, investment strategy, and loan management, making financial decision-making more informed and reliable.