Forecasting Weekly Refrigerated Orange Juice Sales: A Comparative Study of Econometric and Machine Learning Methods

Group 25X

Authors: Noam Diop, Xander Pauwels, Gabriela Rueda, Xuanbo Zhao

Date: June 20 2025

Word Count: 4995

Abstract

Retail success heavily relies on anticipating consumer demand with precision, a challenge that has lately been amplified by the complexity of sales dynamics. This study tackles the problem of forecasting weekly sales for refrigerated orange juice across 11 SKUs and 10 stores using a dataset spanning 102 weeks. We performed variable selection and rolling window forecast evaluations through the use of advanced econometric methods and machine learning techniques. Our work offers valuable insight into the balance of forecast accuracy and model complexity in a retail setting, and showcase that a simple SARIMAX outperforms more complex ML models.

1 Introduction

In the retail sector, accurate demand forecasting plays a critical role in generating operational efficiency and profits. In particular, supermarkets needmust align inventory levels with consumer demand to avoid stock shortages and excess inventory. If they were to underestimate demand, it would result in lost sales, while overestimating it would lead to increased holding costs. With more data available, including prices, promotions, and historical sales, there are greater opportunities to improve forecast accuracy. Advanced econometric and machine learning techniques can help make better use of this information.

Due to the large number of predictors and modeling choices, forecasting sales of Store Keeping Units (SKUs) can quickly become a complex and overwhelming task. Traditional statistical approaches to variable selection often perform poorly in high-dimensional settings, especially when the number of predictors exceeds the number of observations. This makes regularization and machine learning techniques particularly compelling in this case.

Our study focuses on forecasting the weekly sales of refrigerated orange juice products. We use data that contains 10 stores of the same chain, covering 11 different SKUs over 102 weeks. Our goal is to identify relevant predictors and transformations in order to develop accurate forecasts. We compare several approaches, including GETS, LASSO, Ridge, TFT, SARIMAX, SEQ2SEQ, LSTM, XGBoost, and evaluate their predictive performance.

This paper's contribution lies within its application of several modern methods, building on a growing literature on demand forecasting. Real-world sales often contain entry errors or extreme spikes that can skew model results. We therefore apply an IQR-based trimming procedure with a multiplier of four, clipping each weekly sales observation for each (store, brand) series to lie between the first quartile minus four times the interquartile range and the third quartile plus four times the interquartile range, thus capping only the most extreme outliers while preserving genuine promotional peaks [Wan et al., 2014]. Regularization techniques such as LASSO have been shown effective in high-dimensional forecasting problems [Masini et al., 2020], while LSTM networks capture temporal dependencies, offering improvements [Gołąbek et al., 2020] [Ampountolas, 2024]. Trend and seasonality decomposition combined with LightGBM has been found to improve sales forecasts [Zhou, 2023], and ensemble stacking methods further increase

predictive accuracy [Sun, 2022]. We build on these findings by analyzing model performance variation across SKUs and stores, and identifying key predictors driving forecast accuracy.

The remainder of this paper is structured as follows. In Section 2, we present and analyze the data. Section 3 introduces the models and methodology used. In Section 4, we report and compare forecasting results. Section 5 discusses the results and implications. Finally, Section 6 concludes and suggests directions for future research.

2 Data

This analysis uses the `OrangeJuiceX25` dataset, which contains 11,220 weekly observations of orange juice sales across 10 stores and 11 brands. The data were collected from the Dominick's Finer Foods chain across the greater Chicago area from September 4, 1989, through August 1991. Each observation corresponds to a specific brand in a particular store and week, including both unit sales and explanatory variables. Our primary variable of interest is sales, measured in units sold each week. All records also include the retail prices of the 11 brands that are sold in that store during the same week. The column corresponding to the brand of interest represents its own price. For instance, if a record corresponds to brand 3, then the variable `price3` gives its price, while the other price variables represent competitor prices.

In order to capture promotional activity, we made use of binary flags, processed into 0/1 indicators for modeling. In addition to the raw variables, we engineered a set of 17 regressors to help capture key demand drivers. These include log-transformed own prices and their lags, minimum competitor prices (both raw and log), lagged promotion flags, category-level metrics such as the number and share of brands on deal or feature, interaction terms capturing combined promotional pressure, lagged log sales, market share in logs, and interactions between price and promotion. We also included a holiday flag to account for recurring seasonal events.

All continuous variables were standardized to have mean zero and unit variance - this facilitates model convergence and interpretation. The resulting dataset forms a balanced panel, with each store carrying all 11 brands over 102 weeks. This enables us to capture both temporal dynamics and cross-sectional variation in pricing and promotional strategies.

During preprocessing, we handled zeros in variables that required log transformation, standardized the continuous features, and applied a log transformation to the dependent variable within certain models (e.g., regression). To ensure unbiased predictions on the original sales scale, we applied Duan's smearing estimator when reverting log-transformed predictions.

Table 1 presents summary statistics for the variables used in the analysis.

Additionally, we computed the weekly sales trajectories for all 10 stores. Figures 1 and 2 show examples, with the remaining stores' plots provided in the Appendix. Figure 3 presents the seasonal decomposition for store 1, brand 1. These visuals reveal clear trends in the sales data, along with some evidence of seasonality, as sales tend to be higher in summer compared to winter.

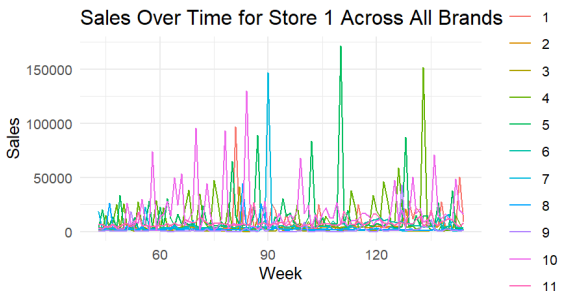

Figure 1: Sales over time for store 1

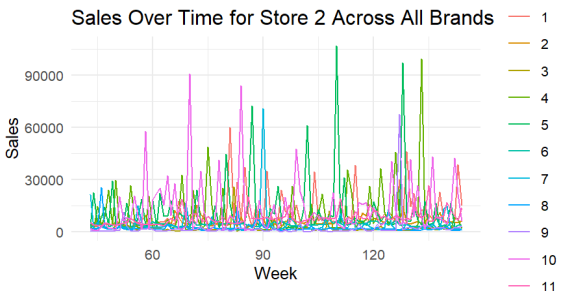

Figure 2: Sales over time for store 2

Table 1: Summary Statistics (Transposed)

Variable	Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
Price1	0.02	0.04	0.05	0.04	0.05	0.06
Price2	0.03	0.04	0.05	0.05	0.05	0.06
Price3	0.02	0.04	0.05	0.04	0.05	0.05
Price4	0.02	0.03	0.04	0.03	0.04	0.05
Price5	0.02	0.03	0.03	0.03	0.04	0.05
Price6	0.03	0.04	0.04	0.04	0.05	0.05
Price7	0.01	0.03	0.03	0.03	0.04	0.05
Price8	0.01	0.03	0.03	0.03	0.04	0.04
Price9	0.01	0.03	0.03	0.03	0.04	0.05
Price10	0.01	0.02	0.02	0.03	0.03	0.04
Price11	0.02	0.02	0.03	0.03	0.03	0.04
Deal	0	0	0	0.44	1	1
Feat	0	0	0	0.18	0	1
Sales	64	2624	5056	10492	9536	406080
Log Sales	4.17	7.87	8.53	8.56	9.16	12.91

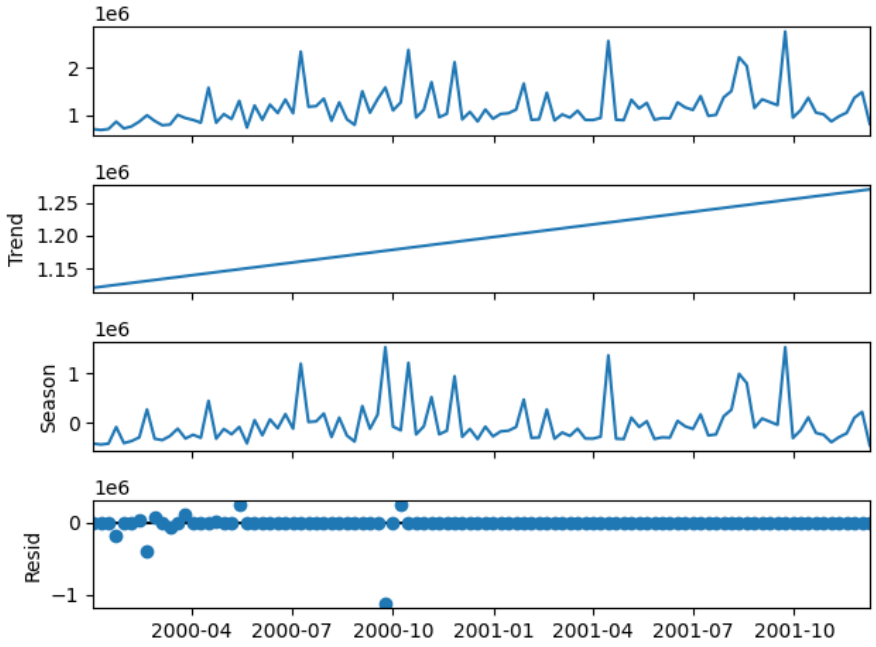

Figure 3: Seasonal Decomposition

We also examined autocorrelation in sales across weeks by computing correlations between current sales and lagged sales up to multiple weeks. As illustrated on Figure 4 only one of these correlations exceeded 0.14, indicating weak temporal dependence. This is a crucial diagnostic for assessing whether explicit time-series structures, such as autoregressive components, are needed. The low correlations suggest that most of the predictive signal is captured by contemporaneous explanatory variables rather than past sales. As such, we include a one-week lag of log sales to account for any residual serial influence, but overall modeling assumptions can rely on cross-sectional independence without a strong loss of information.

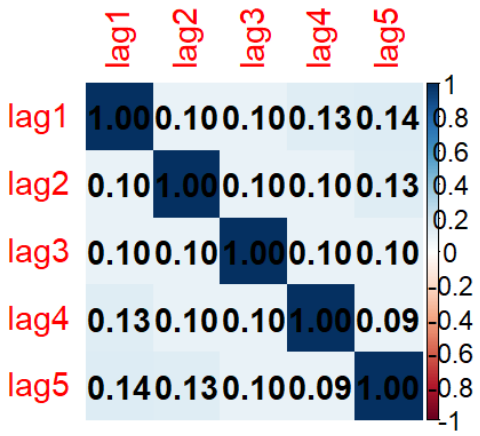

Figure 4: Correaltion between lagged sales

Figure 5 illustrates the impact of promotional activity on weekly sales. The binary variable `feat` equals 1 when a promotion is active and 0 otherwise. In the boxplot, the central line shows the median sales, while the lower and upper edges represent the 25th and 75th percentiles, respectively. Points outside the box indicate outliers—extremely high or low sales values.

The figure clearly demonstrates that sales are higher during promotional periods, with both the median and interquartile range increasing when `feat` equals 1. Furthermore, sales variability also grows under promotion, as reflected by the wider box and the greater number of outliers. This indicates that promotions not only raise typical sales levels but also occasionally trigger significant demand spikes. Note that sales are measured in ounces.

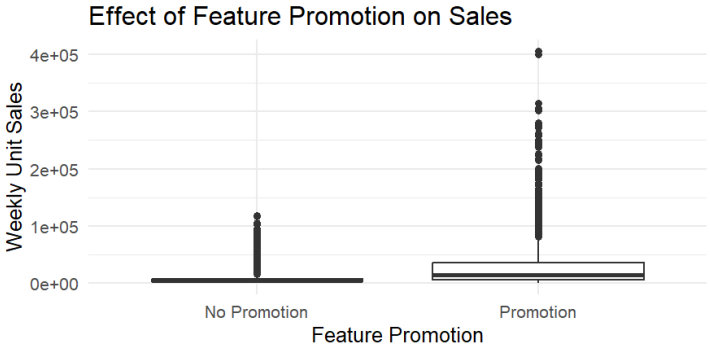

Figure 5: Box plot of weekly unit sales : with vs without promotion

Finally we examine our data to see if we can detect any relevant outliers. If we look at the Box plot of the original data (left panel in Figure 6) we can see that there is a significant number of outliers with some values being higher than 400 and even 1500 while the median is very low (around 50). After applying IQR capping, we can see that there are still a few extreme values but most extreme outliers have been removed.It should also be noted that the spread of the data seems lower, meaning that the variability has been reduced.

The capping significantly reduces the number of outliers while keeping the central tendency and general structure of the data intact.

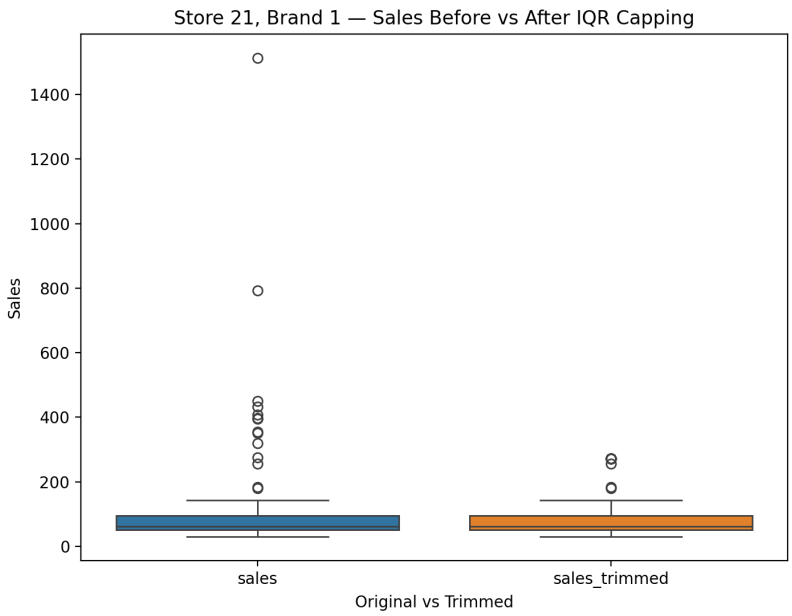

Figure 6: Box plot of sales : Original vs trimmed

The effects of IQR capping are also visible in Figure 7, which shows the distribution of sales before and after applying the outlier treatment method.

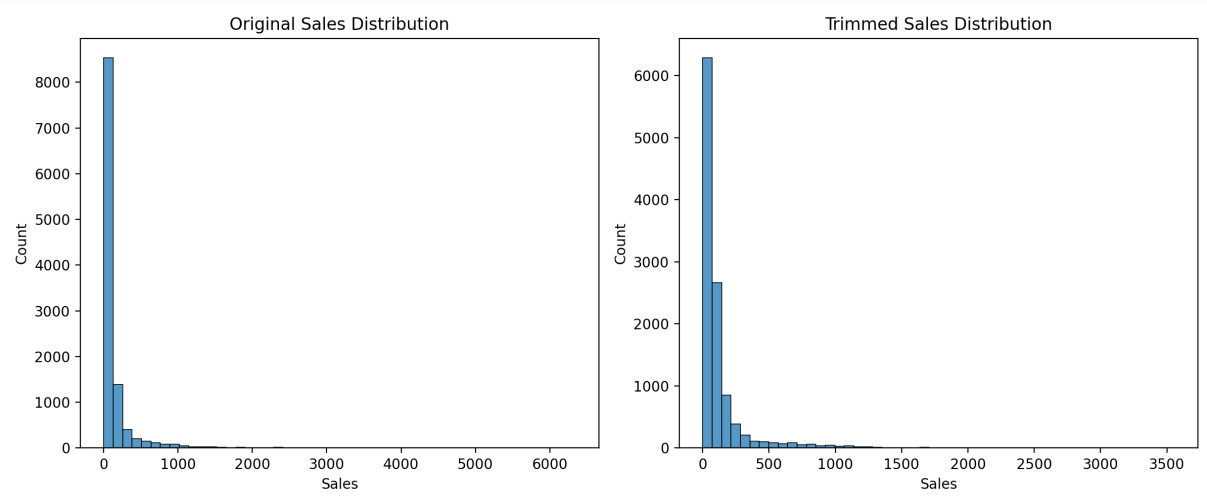

Figure 7: Distribution of sales : Original vs trimmed

We can see that in the original data (graph on the left), the majority of the values are low, but there is a small number of high values that correspond to outliers, as shown in the box plot (6). Therefore, the distribution is highly right-skewed, with long right tails. In the right panel (trimmed sales), the distribution remains right-skewed but appears less extreme. The highest sales values have been brought closer to the main distribution.

Thus, the IQR capping successfully removes extreme outliers while preserving the overall distribution. Moreover, the data is now more suitable for modeling, which we will carry out in the next section.

3 Methodology

We define the weekly unit sales for each SKU i, at week t, in store ℓ as the target variable

$$S_{i,t,\ell} \in \mathbb{Z}_{\geq 0},$$

which we aim to forecast one week ahead. The price per ounce (in cents) for SKU i at week t in store ℓ is given by

$$\text{price}_{i,t,\ell} \in \mathbb{R}_{\geq 0}.$$

We also define two binary indicator variables representing promotional activities:

$$\text{feat}_{i,t,\ell} = \begin{cases} 1 & \text{if SKU } i \text{ is featured in a promotional advertisement at week } t \text{ in store } \ell, \\ 0 & \text{otherwise,} \end{cases}$$

and

$$\text{deal}_{i,t,\ell} = \begin{cases} 1 & \text{if SKU } i \text{ is supported by an in-store coupon at week } t \text{ in store } \ell, \\ 0 & \text{otherwise.} \end{cases}$$

3.1 Set of Regressors

To build an interpretable forecasting model that captures key drivers of demand, we design the following regressors based on microeconomic theory and retail practice:

Price Effects (`OwnLogPrice`, $\texttt{LaggedLogPrice}_{t-1}$, $\texttt{LaggedLogPrice}_{t-2}$): According to demand theory, consumers respond to current price (instantaneous price elasticity) and form expectations based on past prices; hence we include the log of current and two lagged prices. Log-transforming converts elasticities into linear coefficients and facilitates comparisons across SKUs.

Competitive Effects (`MinCompPrice`, $\ln(\texttt{MinCompPrice})$): In the orange-juice category, SKUs are substitutes. We use the minimum competitor price and its logarithm to capture cross-price elasticity—how our SKU's sales respond when competitors raise or lower prices.

Promotional Drivers (`FeatFlag`, `DealFlag`, $\texttt{FeatFlag}_{t-1}$, $\texttt{DealFlag}_{t-1}$, `PromoInteraction`): Feature advertisements and in-store coupons are two common pull promotions. We include current and lagged binary indicators for each, and their interaction captures potential synergy when both promotions run simultaneously.

Category-Level Pressure (`CategoryFeatCount`, `CategoryFeatShare`, `CategoryDealCount`, `CategoryDealShare`, `CrossFeatPressure`): Multiple SKUs promoted concurrently can dilute shelf space and consumer attention. Counts and shares measure overall promotional intensity, while `CrossFeatPressure` tests whether our SKU's uplift is affected when competitors are also featured.

Demand Inertia and Market Position (`LaggedLogSales`, `MarketShareLog`): Lagged sales capture baseline demand and stock-turn dynamics; log market share measures our SKU's standing within the category, reflecting nonlinear competitive effects.

Seasonality and Holidays (`HolidayDummy`): Holidays often alter shopping behavior. A dummy for store–week holidays isolates these shocks to improve fit on regular weeks.

Price–Promotion Interaction ($\texttt{OwnLogPrice} \times \texttt{FeatFlag}$, $\texttt{OwnLogPrice} \times \texttt{DealFlag}$): Discounts may have nonlinear effects when combined with promotions. Interaction terms capture how the marginal effect of price changes differs during feature or coupon weeks.

Table 2 introduces the notation and definitions for all regressors used in our forecasting models.

Table 2: Regressor Names and Formulas

Variable Name	**Formula**
$\text{OwnLogPrice}_{i,\ell,t}$	$\ln\big(\text{price}_{\text{brand}(i),\,\ell,\,t}\big)$
$\text{LaggedLogPrice}_{i,\ell,t-1}$	$\ln\big(\text{price}_{\text{brand}(i),\,\ell,\,t-1}\big)$
$\text{LaggedLogPrice}_{i,\ell,t-2}$	$\ln\big(\text{price}_{\text{brand}(i),\,\ell,\,t-2}\big)$
$\text{MinCompPrice}_{i,\ell,t}$	$\min_{j\neq i}\{\text{price}_{\,j,\,\ell,\,t}\}$
$\ln(\text{MinCompPrice}_{i,\ell,t})$	$\ln\big(\min_{j\neq i}\text{price}_{\,j,\,\ell,\,t}\big)$
$\text{Feat(Deal)Flag}_{i,\ell,t}$	$\text{feat(deal)}_{\,i,\,\ell,\,t}$
$\text{Feat(Deal)Flag}_{i,\ell,t-1}$	$\text{feat(deal)}_{\,i,\,\ell,\,t-1}$
$\text{CategoryFeatCount}_{\ell,t}$	$\sum_{j=1}^{11}\text{feat}_{\,j,\,\ell,\,t}$
$\text{CategoryFeatShare}_{\ell,t}$	$\frac{1}{11}\sum_{j=1}^{11}\text{feat}_{\,j,\,\ell,\,t}$
$\text{CategoryDealCount}_{\ell,t}$	$\sum_{j=1}^{11}\text{deal}_{\,j,\,\ell,\,t}$
$\text{CategoryDealShare}_{\ell,t}$	$\frac{1}{11}\sum_{j=1}^{11}\text{deal}_{\,j,\,\ell,\,t}$
$\text{PromoInteraction}_{i,\ell,t}$	$\text{FeatFlag}_{i,\ell,t}\times\text{DealFlag}_{i,\ell,t}$
$\text{CrossFeatPressure}_{i,\ell,t}$	$\text{FeatFlag}_{i,\ell,t}\times I\big(\sum_{j\neq i}\text{feat}_{\,j,\,\ell,\,t}>0\big)$
$\text{LaggedLogSales}_{i,\ell,t-1}$	$\ln\big(\text{sales}_{\,i,\,\ell,\,t-1}\big)$
$\text{MarketShareLog}_{i,\ell,t}$	$\ln\big(\text{sales}_{\,i,\,\ell,\,t}\big)-\ln\big(\sum_{j=1}^{11}\text{sales}_{\,j,\,\ell,\,t}\big)$
$\text{HolidayDummy}_{\ell,t}$	$\begin{cases}1, & \text{if store } \ell \text{ in week } t \text{ includes a holiday;}\\ 0, & \text{otherwise.}\end{cases}$
$\text{Price}\times\text{Feat(Deal)}_{i,\ell,t}$	$\ln\big(\text{price}_{\,i,\,\ell,\,t}\big)\times\text{Feat(Deal)Flag}_{\,i,\,\ell,\,t}$

3.2 Feature Set Variants

In order to investigate how the choice and number of features affect forecast performance, we constructed three versions of our modeling dataset, differing in which regressors are included:

Dataset A (Core Economic Drivers) This highly parsimonious set contains only 4 fundamental features that are straightforward drivers of demand:

`OwnLogPrice`

`FeatFlag` (current week feature promotion flag)

`DealFlag` (current week deal promotion flag)

`PromoInteraction` (an interaction term for when both a feature and a deal occur together)

This subset intentionally omits lagged sales or competitive variables, focusing purely on the SKU's own price and promotions in the current week. We try this model to concretely see if promotions and prices can accurately describe expected sales, as we saw earlier that promotion has a big impact on the sales. It represents a scenario emphasizing interpretability and key economic factors.

Dataset B (Full Feature Set) This dataset contains the entire set of engineered predictors described above (17 custom regressors capturing own-price, competition, promotions, lagged effects, etc., along with basic price and promotion flags). It represents the richest information scenario, albeit with potential issues of high dimensionality and multicollinearity due to many correlated variables.

Dataset C (LASSO-Selected Subset) We applied LASSO regression on the training data to estimate each variable's selection frequency. Variables selected in the majority of runs were retained, and we computed pairwise Pearson correlations among these candidates, removing one variable from each pair with $|r| \geq 0.8$ to mitigate multicollinearity. The resulting feature subset was then used for model evaluation; the specific predictors selected are reported in Section 4.3.

3.3 Forecasting procedure

Rolling-Window Forecasting Setup

All models are evaluated using a one-step-ahead rolling forecasting framework. For each store–brand series, a sliding window of the past 52 weeks is used to train the model (allowing it to capture the full season cycle). This is then applied to predict the following week's sales. As new data arrive, the window advances by one week, and the model is retrained. As such, consistent training size and fair comparisons are ensured.

With 110 series and 50 rolling steps per series on average, each model produces approximately 5,500 forecasts. All models are fitted independently to each time series, with the only exception being the globally trained Temporal Fusion Transformer (TFT).

Forecasts are evaluated on the original sales scale. However, some training differences among the models should be noted. Linear models (Ridge, LASSO, GETS) are trained on log-transformed sales to stabilize variance. SARIMAX is fitted directly on raw sales while XGBoost is trained on $\log(1 + S)$ and inverted via $\exp(\cdot) - 1$ without correction. Finally, Neural networks (LSTM, Seq2Seq, TFT) are trained to predict sales directly. In all cases, the models' outputs are compared to the true values in their original units.

Regression Models with Variable Selection

We implemented three regression models with distinct variable selection strategies: Ridge, LASSO, and General-to-Specific (GETS) regression. For all three models, the predictions of log-sales were transformed back into sales using Duan's smearing correction.

Ridge Regression. Ridge imposes an L2 penalty; its main characteristic is that it shrinks less relevant coefficients without actually eliminating them. For each 52-week window, features were standardized, and `RidgeCV` was used with time-series cross-validation over a grid of α values.

LASSO Regression. LASSO applies an L1 penalty, meaning it sets the least relevant coefficients to zero. For each window, we trained a LASSO model with α selected via 5-fold time-series cross-validation using `LassoCV`. The selected features were then tracked across windows.

GETS Regression. The GETS (General-to-Specific) approach applies stepwise OLS, starting from the full model and iteratively removing the least significant variables until all remaining coefficients have $|t| \geq 1.96$ (where $|t|$ denotes the t-statistic). This enforces strong significance and helps avoid overfitting.

All three methods used lagged variables to capture persistence and any post-promotion effect. For example, `DealFlag_L1` helps model post-promotion declines in sales, while lagged sales terms capture autoregressive behavior.

SARIMAX Time-Series Model

As a benchmark, we fit a Seasonal ARIMA with eXogenous variables (SARIMAX) model for each series. Given the limited time span of 102 weeks and the large number of predictors, we used a parsimonious ARIMA(1,0,0) specification augmented by a seasonal AR term:

```
seasonal_order=(26, 0, 0, 0)
```

This adds an AR(26) component, capturing a 26-week (half-year) cycle in weekly orange-juice demand, which reflects biannual patterns such as summer versus midwinter promotions or weather-driven consumption shifts. We chose 26 lags rather than full 52 because the strongest seasonal autocorrelation appeared at the half-year horizon, and this simpler seasonal term improved fit without over-parameterizing the model.

When applying the 52-week rolling window, we increased the optimizer's iteration limit and suppressed convergence warnings to improve model stability.

Machine Learning and Neural Network Models

We also evaluated more advanced models capable of capturing nonlinear relationships and interactions between variables.

XGBoost. We trained a separate per-series XGBoost model using a 52-week rolling window. Targets were transformed via $\log(1+\text{sales})$, and predictions were back-transformed with $\exp(\cdot)-1$.

We used a fixed set of key hyperparameters, `max_depth=4`, `learning_rate=0.05`, `subsample=0.7`, `colsample_bytree=0.7`, and `min_child_weight=3`, tuned offline for bias–variance trade-off and stability. Each model was trained for `num_boost_round=300` on 52 data points, with column subsampling and moderate leaf-weight regularization, using GPU acceleration (`tree_method="gpu_hist"`).

LSTM. We also trained a univariate Long Short-Term Memory (LSTM) model on each 52-week window, using the last 3 weeks as input (lookback = 3). All available exogenous variables (e.g., prices, promotions) were included. The LSTM had two hidden layers with 32 units each and was trained for 10 epochs.

Seq2Seq Encoder–Decoder. We implemented a sequence-to-sequence LSTM model using an encoder for the previous 3 weeks and a decoder that received next-week covariates. This setup used the same rolling window and 10-epoch training scheme as the standard LSTM. Although future features like promotion flags may not always be known in practice, we included them here for forecasting purposes. Both LSTM models were implemented in PyTorch, with sequences rebuilt at each iteration and any missing data removed.

Temporal Fusion Transformer (TFT). We trained a single TFT model across all series using a 52-week rolling window approach. This multivariate time-series model leverages attention and gating mechanisms, with store and brand IDs included as categorical inputs. For each window, the first 80% of weeks were used for training, the next 10% for validation (with early stopping on validation RMSE), and the final 10% for testing—reporting test RMSE on that segment. The architecture comprised an LSTM encoder, static feature encoders, and multi-head attention layers.

Some limitations are worth noting. First, although the TFT can incorporate known future covariates (e.g., sine/cosine seasonal terms and planned promotions), these may not always be available in practice. Second, with only 110 series $\times$ 102 weeks of data, the model likely remained under-trained, which could have constrained its forecasting accuracy.

3.4 Outlier Handling (IQR Trimming)

Before feature construction, we applied IQR-based trimming separately for each (store, brand) series in Python. First, we computed the 25th percentile Q_1 and 75th percentile Q_3, then calculated $\text{IQR} = Q_3 - Q_1$. We chose a trimming multiplier of $k = 4$, striking a balance between removing implausible extremes and preserving genuine promotional spikes. Each sales observation y was clipped to lie within the interval:

$$[\, Q_1 - k\,\text{IQR},\ Q_3 + k\,\text{IQR}\,].$$

3.5 Reproducibility

The entire analysis was conducted in Python (version 3.9), within a dedicated Conda environment configured specifically for this project. Key packages included `NumPy` (v1.21) and `pandas` (v1.3) for numerical computation and data manipulation, `scikit-learn` (v1.0) for preprocessing and evaluation metrics, `statsmodels` (v0.13) for SARIMAX modeling, `XGBoost` (v1.5) for gradient boosting, and `PyTorch` (v1.10) for developing and training neural networks.

All scripts, including data preprocessing, feature engineering, model training, and evaluation, were version-controlled to ensure full reproducibility of results.

4 Results

This section presents the evaluation results of our forecasting models. All models were executed on a laptop equipped with an Intel i9-13950HX processor and an RTX 4070 GPU. The total computational time for all experiments was approximately 21 hours.

4.1 Economic Drivers Dataset Performance

We first evaluate model performance using a minimal subset of economic drivers, including `OwnLogPrice`, `FeatFlag`, `DealFlag`, and `PromoInteraction`.

Model	**RMSE**
SARIMAX	108.83
GETS	203.13
LASSO Regression	204.05
Ridge Regression	205.56
LSTM (Deep Learning)	181.93
Seq2Seq (Deep Learning)	182.35
TFT (Deep Learning)	191.84
XGBoost	129.81

Table 3: RMSE values for different models under minimal feature set

As shown in Table 3, the SARIMAX model clearly outperforms all other approaches in this setting. This reflects its effectiveness in capturing seasonality and autocorrelation, which are critical for modeling time-dependent patterns. In contrast, deep learning models such as LSTM and Seq2Seq perform relatively poorly—likely due to the limited feature set, lack of explicit lagged inputs, and insufficient data volume to leverage their complexity. Similarly, linear models like LASSO and Ridge regression underperform, as they are not designed to model temporal dynamics. These results suggest that model performance may benefit significantly from incorporating additional regressors, including competitor pricing, promotional activity, and lagged historical features. From a forecasting perspective, the findings emphasize that simpler, well-specified time series models can provide robust and interpretable predictions in retail environments. This is especially important when seasonal patterns dominate, as it can support more effective inventory management and demand planning decisions.

4.2 Full Dataset Performance

To evaluate the effect of incorporating a more comprehensive set of predictors, we retrain each model using the complete feature set. This includes lagged variables, promotional indicators, and competitor pricing data. The goal is to evaluate whether additional temporal and contextual information can improve forecasting accuracy compared to the minimal feature set.

Model	RMSE
LASSO Regression	117.45
Ridge Regression	120.09
SARIMAX	116.55
GETS	2845.26
XGBoost	129.07
Seq2Seq (Deep Learning)	183.06
LSTM (Deep Learning)	182.64
TFT (Deep Learning)	198.10
Naive Persistence	282.68

Table 4: RMSE values for full data

Interestingly, as shown in Table 4, SARIMAX continues to outperform the other models, although the performance gap between SARIMAX and linear models such as Ridge and LASSO narrows. The inclusion of lagged variables and competitor pricing notably improves the performance of these linear models, enabling them to better capture temporal dependencies and external influences. Deep learning models still underperform relative to SARIMAX and XGBoost, indicating that they may require larger datasets or more extensive hyperparameter tuning to fully leverage the expanded feature space. Notably, the GETS model performs the worst in this setting, likely due to its sensitivity to multicollinearity and redundant features, which hamper its generalization ability.

To further investigate this issue and gain deeper insight into the key drivers of forecast accuracy, we analyze the frequency with which each regressor is selected during the rolling-window estimation procedure.

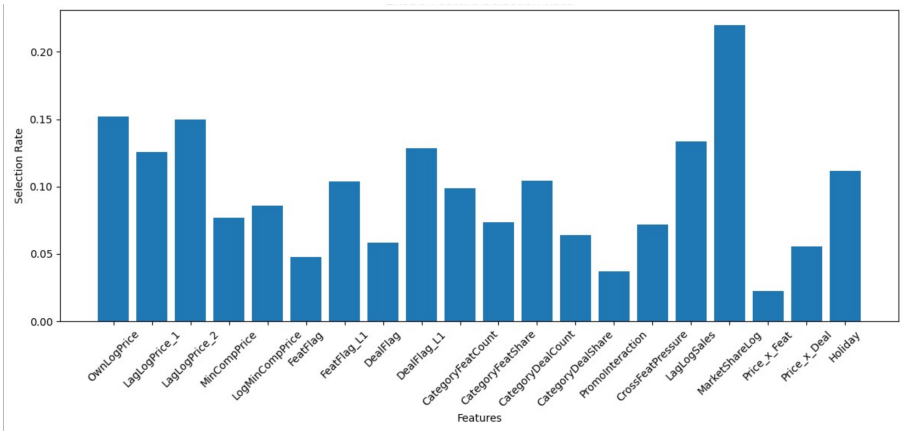

Figure 8: LASSO feature selection rate

Figure 8 illustrates the selection frequency of each feature in the LASSO model. We observe a clear pattern, with each rolling-window iteration typically retaining between five and eight regressors.

Among the most consistently selected features are those related to price sensitivity, brand momentum, and promotional timing. Notably, `MarketShareLog` emerges as the most frequently chosen variable, appearing in over 20% of the rolling windows. This highlights the importance of a brand's relative position within its competitive set. Close behind are `LagLogSales`, `OwnLogPrice`, `LagLogPrice_1`, and `DealFlag_L1`, each selected in more than 15% of cases. Collectively, these variables capture recent demand trends, current pricing strategies, and lingering promotional effects, emphasizing their relevance in modeling short-term sales dynamics. These

findings align well with our case regarding orange juice, where pricing and promotional activity are critical drivers of weekly sales performance.

However, selection frequency alone does not guarantee model robustness. As shown in Figure 9, many of the most frequently chosen regressors exhibit high pairwise correlations, which may introduce multicollinearity and reduce model interpretability. This could partly explain the decline in performance of the GETS model when using the expanded set of regressors.

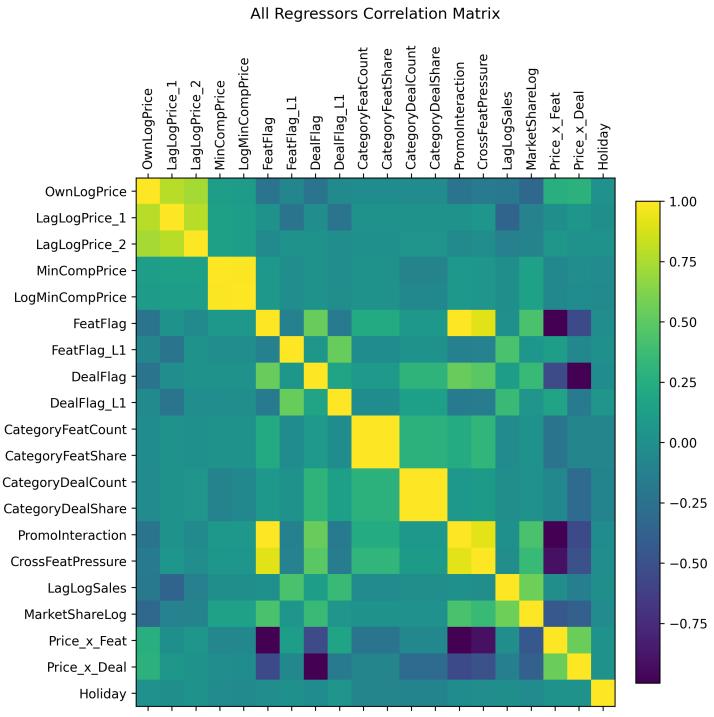

Figure 9: Correlation matrix of modified regressors

4.3 LASSO-Based Performance

To mitigate multicollinearity and reduce the risk of overfitting, we prune the feature space by constructing a reduced pool of regressors. Variables are retained based on their selection frequency across rolling windows and filtered using pairwise correlation thresholds. The following features emerged from the LASSO model as the most consistently informative:

- `OwnLogPrice`
- `LagLogPrice_2`
- `MinCompPrice`
- `CategoryFeatShare`
- `CrossFeatPressure`
- `LagLogSales`
- `DealFlag_L1`

Figure 10 presents the correlation matrix for the reduced set of LASSO-selected features. The pruning strategy has effectively reduced multicollinearity among the regressors, as evidenced by the lower pairwise correlations.

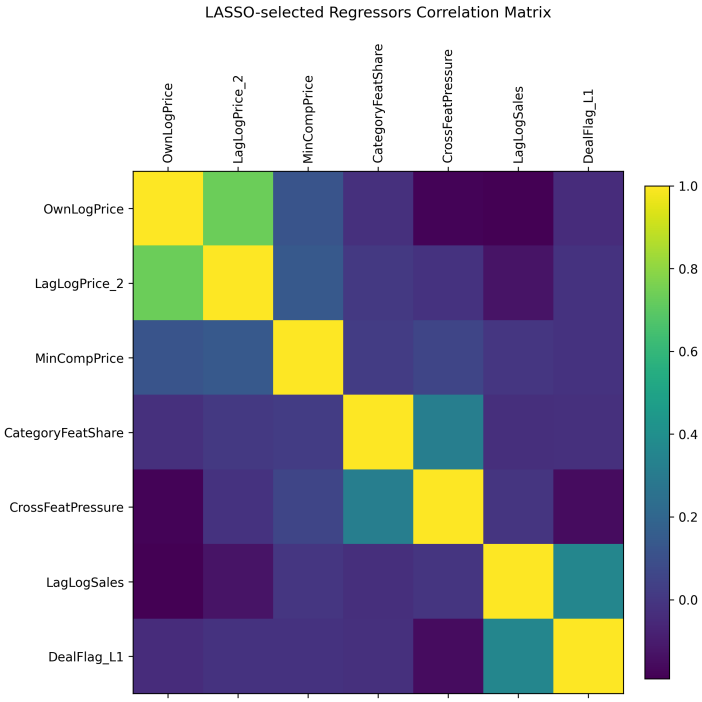

Figure 10: Correlation matrix for LASSO-selected regressors

4.4 LASSO-Selected Subset Performance

We then re-evaluate our models using the reduced set of seven critical predictors selected by LASSO. The top three models remain unchanged: SARIMAX, LASSO, and Ridge, all delivering comparable accuracy. This confirms that the selected subset retains nearly all of the predictive information from the full dataset. The marginal difference in performance compared to the full dataset suggests that noise has been effectively reduced without a substantial loss in forecasting capability. GETS now performs significantly better, securing the fourth position. This improvement highlights the importance of the feature set for GETS, with the reduction in multicollinearity enhancing its accuracy. In contrast, the deep learning models—LSTM, TFT, and Seq2Seq—perform the worst, further indicating that the primary challenge for these models is the limited size of our dataset.

These findings emphasize that, for our retail sales forecasting problem, which has limited historical data and a moderate number of SKUs and stores, parsimonious models such as SARIMAX and regularized linear regressions are able to reach an effective balance between interpretability and predictive performance. Meanwhile, the challenges faced by more complex deep learning models highlight the need for larger datasets when applying them in similar retail settings.

Model	**RMSE**
LASSO Regression	118.68
Ridge Regression	120.55
GETS	120.92
LSTM (Deep Learning)	183.19
Seq2Seq (Deep Learning)	184.41
SARIMAX	113.98
TFT (Deep Learning)	197.09
XGBoost	127.99

Table 5: RMSE values for different models on the LASSO-selected subset

4.5 Cross-Dataset Comparison

A summary of RMSE across models and datasets is provided in Table 6.

Model	Full Data	LASSO Subset	Economic Drivers
SARIMAX	**116.55**	**113.98**	**108.83**
Naïve Persistence	282.68	282.68	282.68
Ridge Regression	120.09	120.55	205.56
LASSO Regression	117.45	118.68	204.05
GETS	2845.26	120.92	203.13
XGBoost	129.07	127.99	129.81
Seq2Seq	183.06	184.41	182.35
LSTM	182.64	183.19	181.93
Temporal Fusion Transformer	198.10	197.09	191.84

Table 6: RMSE of forecasts by model and dataset.

In summary, SARIMAX remains the most reliable model across both sparse and rich feature configurations. Its strength lies in its structured modeling of seasonality, trends, and lagged dependencies, which are critical factors in retail time series data. SARIMAX performs well even in small-data scenarios by embedding domain knowledge into the model structure, allowing it to generalize effectively, even when compared to more complex, data-hungry models.

On the other hand, deep learning models underperform in this context due to data limitations and insufficient tuning. These models, which require large datasets to capture intricate patterns, struggle to achieve superior accuracy given the relatively small size of the dataset and the need for further optimization.

Econometric methods such as GETS are highly dependent on the dataset configuration. GETS performs poorly when excessive features are included but delivers excellent results when only minimal, relevant inputs are used. This highlights the importance of feature selection in econometric models, with performance being significantly impacted by the choice of variables.

We finally conducted a Diebold–Mariano test to assess whether the forecast accuracy of the SARIMAX model significantly exceeds that of the LASSO model. The test did not yield statistically significant evidence in favor of SARIMAX, suggesting that differences in predictive performance are not robust under this formal comparison. As a result, we rely primarily on RMSE as our evaluation criterion. While this approach has limitations, such as not accounting for forecast variance, it remains appropriate for the scope and aims of this study.

4.6 Discussion

Our evaluation of orange juice sales forecasting yielded two key insights: SARIMAX excels with a minimal set of economic drivers, while LASSO (and Ridge) effectively focus on important predictors in high-dimensional settings. Complex ML and deep-learning models offered no accuracy gains and often performed worse.

SARIMAX outperformed other methods on the economic-drivers subset by leveraging autoregressive and seasonal components to capture intrinsic time-series dynamics. These are patterns that linear regressions without lagged inputs could not learn. Its built-in "memory" allowed it to anticipate holiday effects and post-promotion rebounds, illustrating that strong inductive biases trump purely data-driven approaches when the signal-to-noise ratio is low.

With the full feature set, LASSO delivered the best accuracy by shrinking irrelevant coefficients and isolating predictive signals from dozens of potential drivers. This automatic variable selection yielded a parsimonious model whose performance matched that of the full

dataset. This result echoes prior findings on LASSO's efficacy in macroeconomic forecasting [Li and Chen, 2014].

In contrast, XGBoost, Seq2Seq, LSTM, and TFT underperformed—likely due to limited data volume (weekly sales of a single product), the difficulty of tuning numerous hyperparameters, and the need for large datasets to realize deep models' potential. Their failure underscores the value of interpretability and domain-informed structures in retail forecasting.

As stepwise t-tests can be misled by multicollinearity and multiple comparaisons, GETS struggled on the full set because stepwise t-tests.However, when applied to the LASSO-filtered subset, it produced reliable estimates. This suggests a hybrid workflow where LASSO is used for initial variable screening, followed by GETS for detailed econometric diagnostics.

For practitioners, these results imply that simple statistical models and regularized regressions should be prioritized before deploying complex AI methods. When features are scarce, structured time-series models like SARIMAX are robust; when features abound, regularization ensures parsimony without sacrificing accuracy. Baseline forecasts must always be outperformed to justify model complexity.

5 Conclusion

In this paper, we conducted an extensive comparison of forecasting methods for weekly orange juice sales, using a common dataset and three different feature sets. The methods ranged from classical statistical approaches (SARIMAX, GETS) to simple benchmarks (naïve persistence), modern regularized regressions (Ridge, LASSO), machine learning (XGBoost), and advanced deep learning models (Seq2Seq LSTM, LSTM, and Temporal Fusion Transformer). Our analysis highlights the critical role of feature selection. Although the full dataset included numerous potential predictors, a sparse subset identified by LASSO retained nearly all the predictive power. Models trained on this reduced feature set performed better than those using the full set, suggesting that most additional variables contributed little to forecast accuracy. The key drivers were the product's own price, promotional activity, and recent sales trends. Among all regressors, SARIMAX consistently outperformed other models, particularly when only price and promotion data were used (excluding lagged sales). It achieved the lowest RMSE (108.83), far surpassing other approaches due to its strength in capturing seasonality and autocorrelation. When using the full feature set, LASSO delivered the second-best performance (117.45 RMSE), slightly ahead of Ridge (120.09). Its built-in feature selection helped prevent overfitting and preserved interpretability by focusing on the most relevant inputs. Machine learning models did not outperform these simpler methods. XGBoost showed decent results (129.07 RMSE on the full set) but lagged behind the linear models. Deep learning models underperformed significantly, with RMSE values nearly 50% higher than LASSO's. For a single, moderately sized time series, the added complexity of ML/DL models did not yield better results.

Forecasting orange juice sales illustrates the broader lesson that the best method depends on data characteristics. A hybrid approach that integrates domain-informed statistical modeling with data-driven feature selection produces forecasts that are robust, accurate, and interpretable. Future work could extend this to multi-product models, real-time updating, and additional data sources (e.g., weather or search trends) to explore when modern ML can surpass traditional baselines in retail forecasting.

While our findings are clear for this specific case, they may not universally apply. Deep learning models, like transformers and RNNs, could offer advantages with much larger datasets or when training a global model across many similar product lines, especially for complex seasonalities or nonlinearities. However, our study reflects a typical business scenario with moderate data and a need for immediate interpretability.

Future work could explore hybrid models (e.g., SARIMAX with XGBoost for residuals) or probabilistic forecasting to provide prediction intervals, which is crucial for risk assessment.

While our models focused on point forecast accuracy (RMSE), some, like TFT, can produce quantile forecasts for uncertainty.

For forecasting thousands of products, deep learning models could learn across series (multivariate time series forecasting), potentially revealing cross-series patterns (e.g., similar seasonality across all juices). This scalability and automation could favor machine learning, even if classic models perform well on individual series.

Ultimately, the best model varies on the specific context. For forecasting orange juice sales, simpler linear models outperformed the more complex machine learning approaches. This underscores the value of traditional statistical methods and thoughtful feature engineering. It also highlights that for time series forecasting, success lies in carefully selecting relevant features for classical models and embedding domain-specific structures, such as seasonality, to unlock the full potential of machine learning models.

References

[Ampountolas, 2024] Ampountolas, A. (2024). Forecasting orange juice futures: Lstm, convlstm, and traditional models across trading horizons. *Journal of Risk and Financial Management*, 17(11):475.

[Gołąbek et al., 2020] Gołąbek, M., Senge, R., and Neumann, R. (2020). Demand forecasting using long short-term memory neural networks.

[Li and Chen, 2014] Li, J. and Chen, W. (2014). Forecasting macroeconomic time series: Lasso-based approaches and their forecast combinations with dynamic factor models. *International Journal of Forecasting*, 30(4):996–1015.

[Masini et al., 2020] Masini, R. P., Medeiros, M. C., and Mendes, E. F. (2020). Machine learning advances for time series forecasting.

[Sun, 2022] Sun, C. (2022). Forecasting retail sales via the use of stacking model. In *Proceedings of the 2022 International Conference on E-Society, E-Education and E-Technology (ICEDBC 2022)*, volume 225 of *Advances in Economics, Business and Management Research*, pages 405–411. Atlantis Press.

[Wan et al., 2014] Wan, X., Wang, W., Liu, J., and Tong, T. (2014). Estimating the sample mean and standard deviation from the sample size, median, range and/or interquartile range. *BMC medical research methodology*, 14:1–13.

[Zhou, 2023] Zhou, T. (2023). Improved sales forecasting using trend and seasonality decomposition with lightgbm.

Appendix

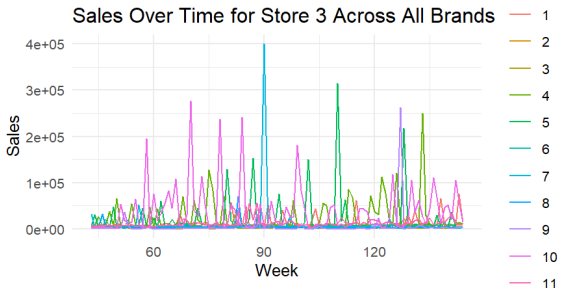

Figure 11:

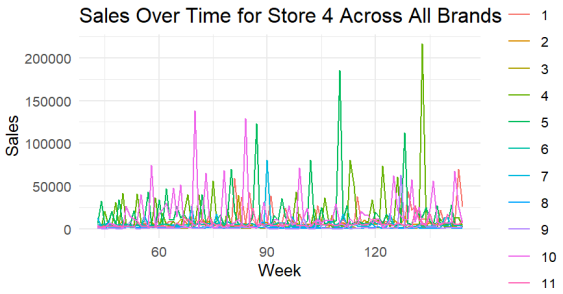

Figure 12:

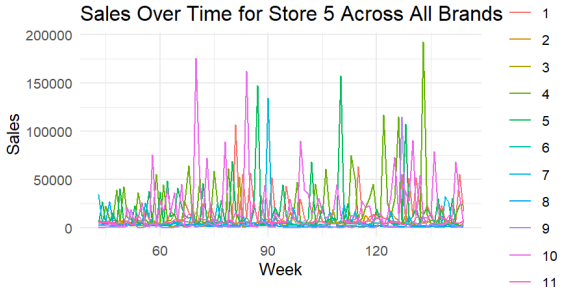

Figure 13:

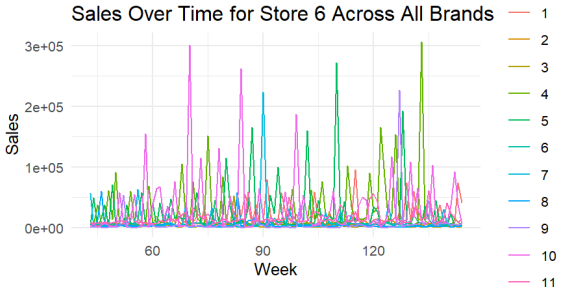

Figure 14:

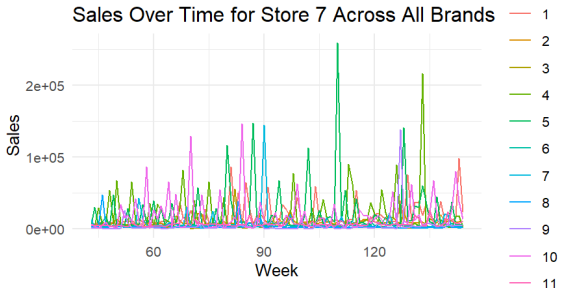

Figure 15:

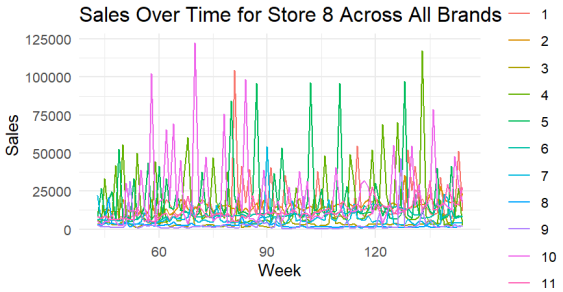

Figure 16:

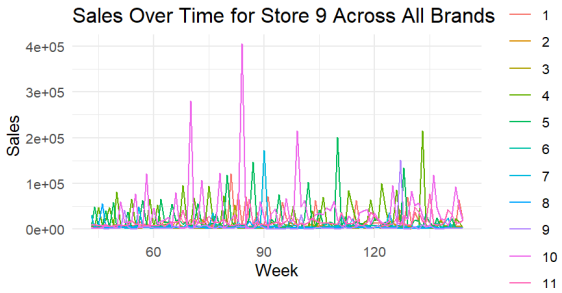

Figure 17:

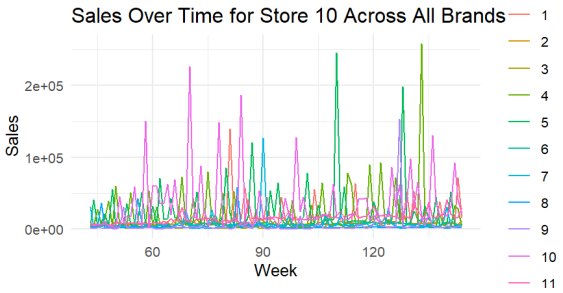

Figure 18: