Time Series Forecasting

Overview and Discussion

Agenda

- Data Preprocessing
- Baselining
- Machine Learning (Global Forecast Model)
- Deployment

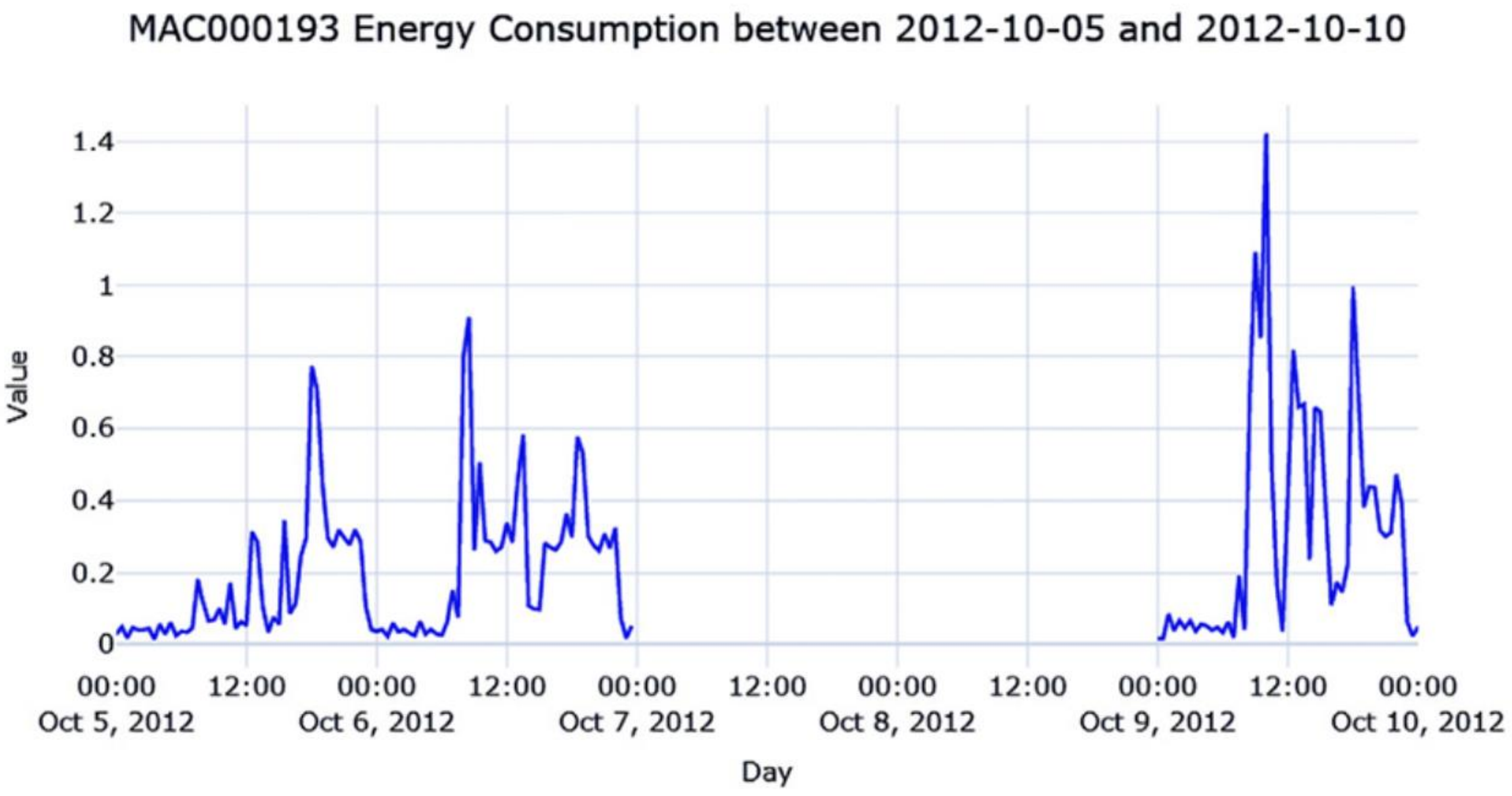
Data Preprocessing

- Missing data
- Visualization
- Decomposition
- Outlier handling

Handling missing data – few timepoints

- Simpler techniques
  - Last Observation Carried Forward or Forward Fill
  - Next Observation Carried Backward of Backward Fill
  - Nearest Interpolation
  - Mean Value Fill
- Interpolation
  - Linear Interpolation
  - Spline, Polynomial, and Other Interpolations

Handling bigger gaps

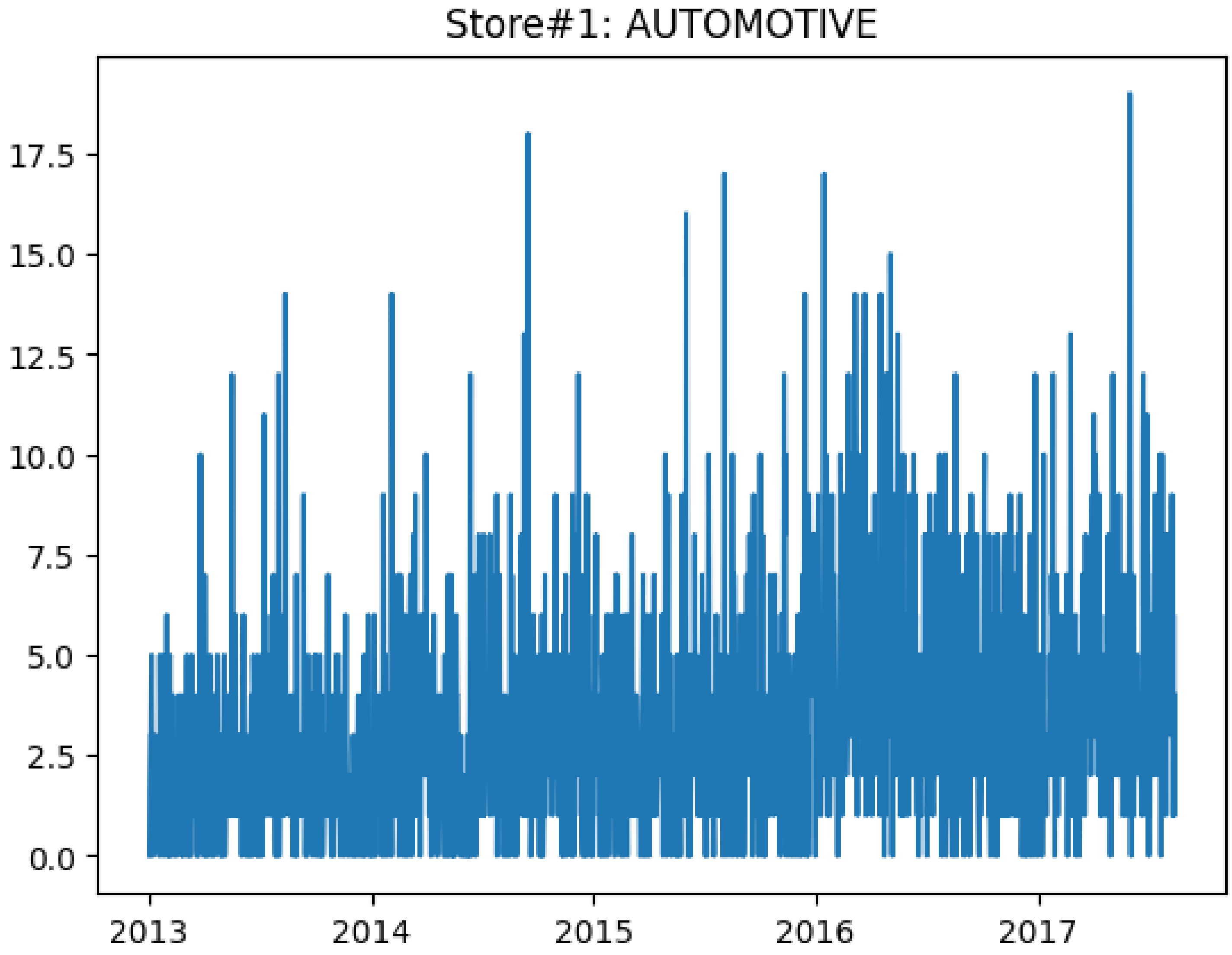


Ref: Modern Time Series Forecasting with Python, Second Edition, Manu Joseph, Jeffrey Tackes

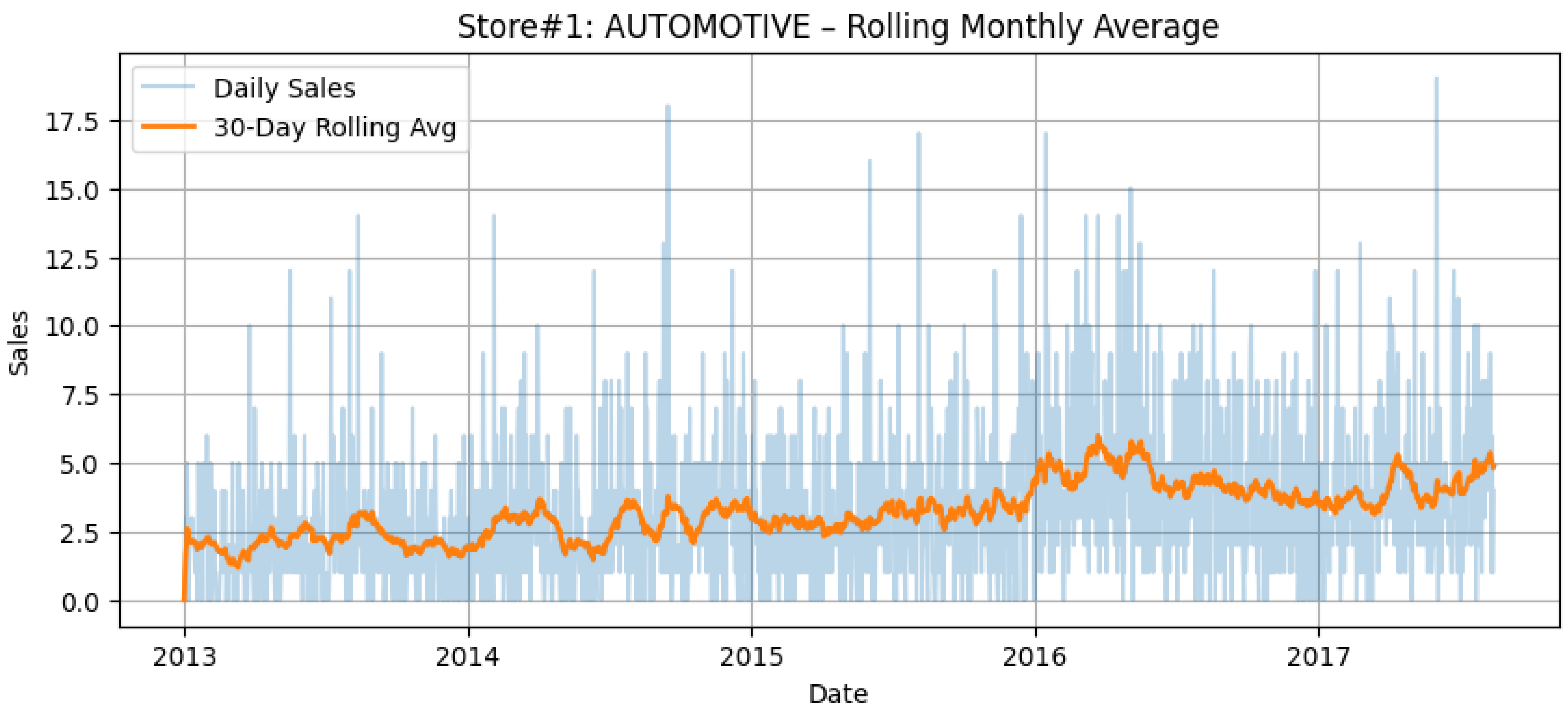
Handling bigger gaps

- Imputing with the previous day
- Hourly average profile
- The hourly average for each weekday
- Seasonal interpolation
  - Fills missing data by temporarily removing repeating patterns, filling gaps smoothly, and then restoring those patterns.

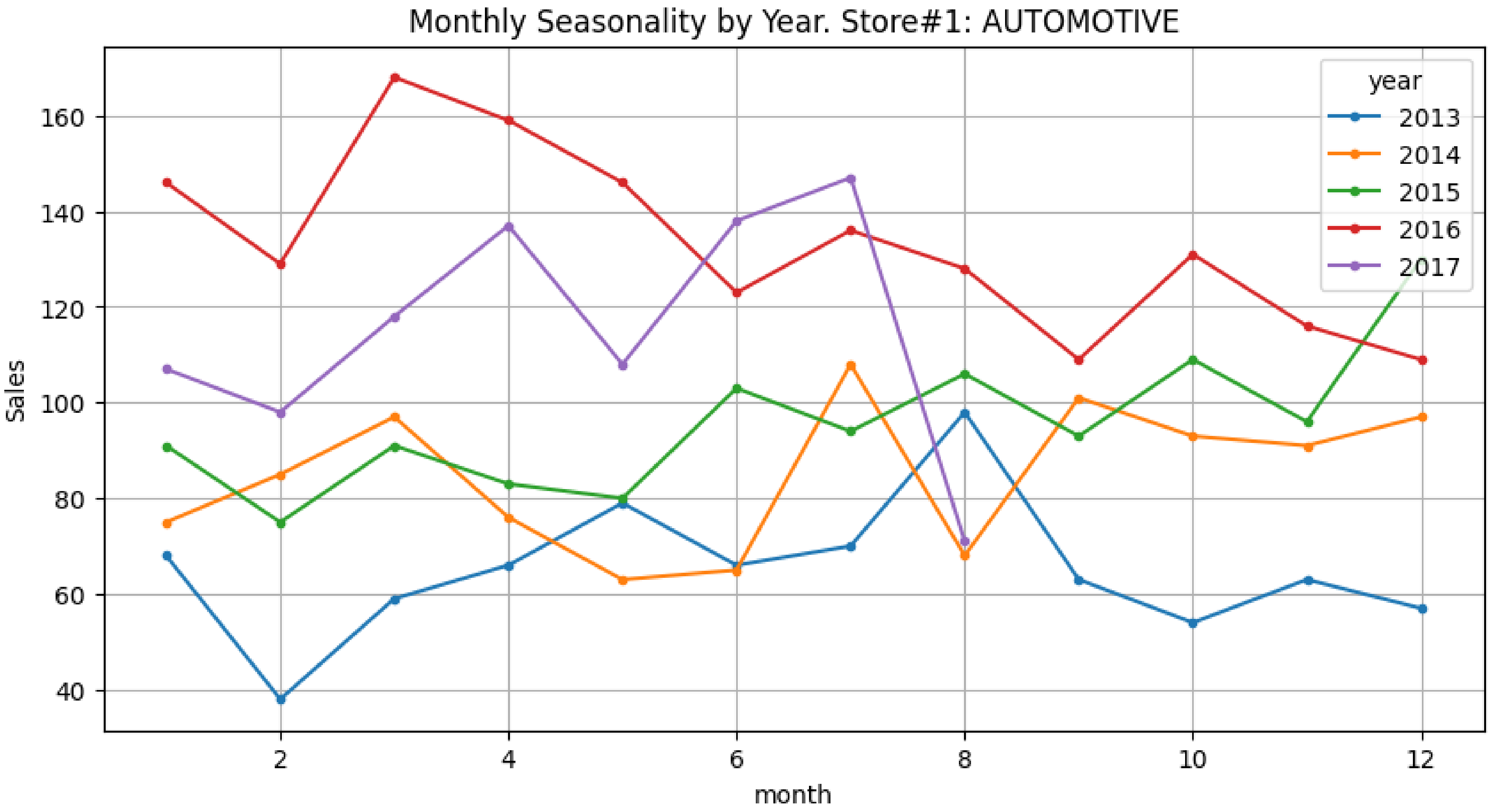
Visualization



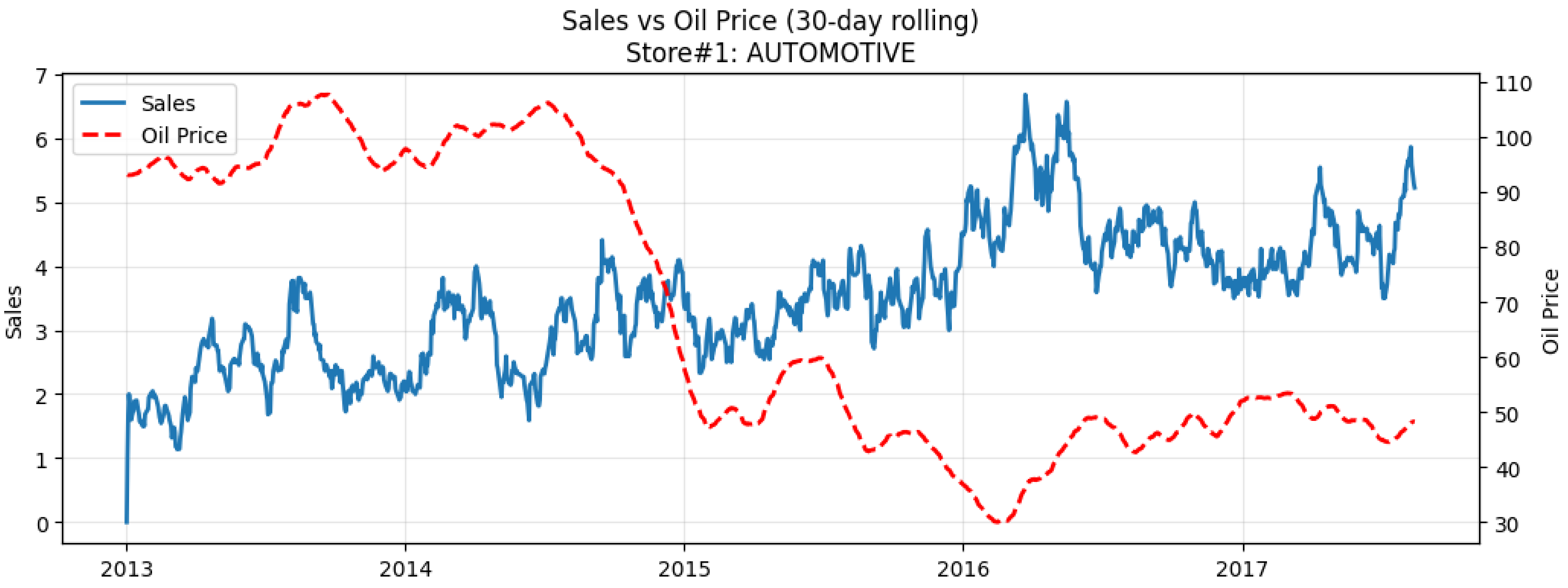
Visualization: Macro Patterns



Visualization: Seasonality



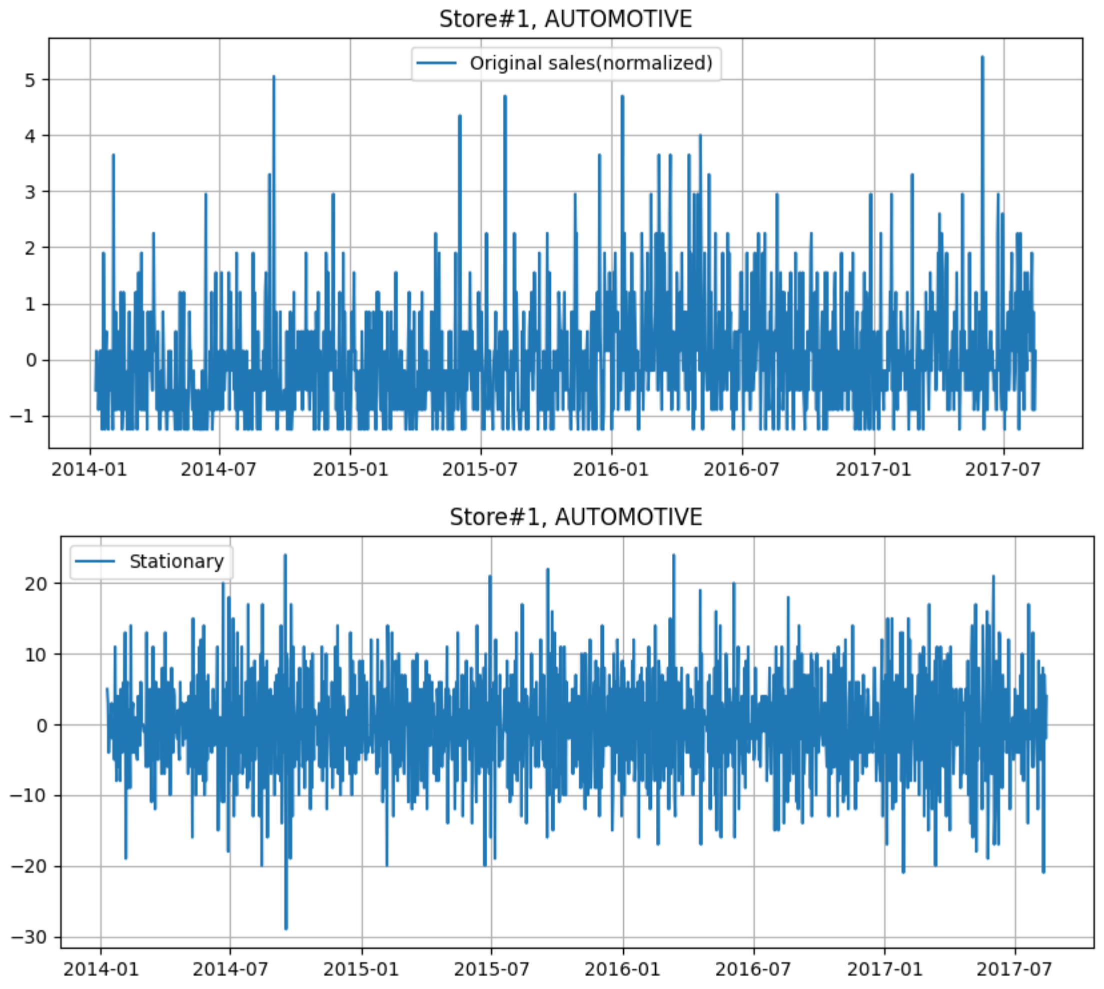
Visualization: Correlation



Decomposition

- Time series data = trend + seasonality + residual
- Remove trend
  - $diff(t) = x(t) - x(t-1)$
- Remove seasonality
  - Weekly
    - $x(t) - x(t-7)$
  - Yearly
    - $x(t) - x(t-365)$

Decomposition



Outlier handling

- Detection:
  - Standard deviation
  - IQR
  - Isolation Forest
- Removal:
  - Do not blindly remove outliers as they may contain information
  - Replace with a heuristic such as the maximum, minimum, and 75th percentile
  - Consider the outliers as missing data
- Modern algorithms are capable of handling outliers

Baselining

- Setting up a test harness
- Generating strong baseline forecasts (classical and statistical methods)
- Assessing the forecastability of a time series

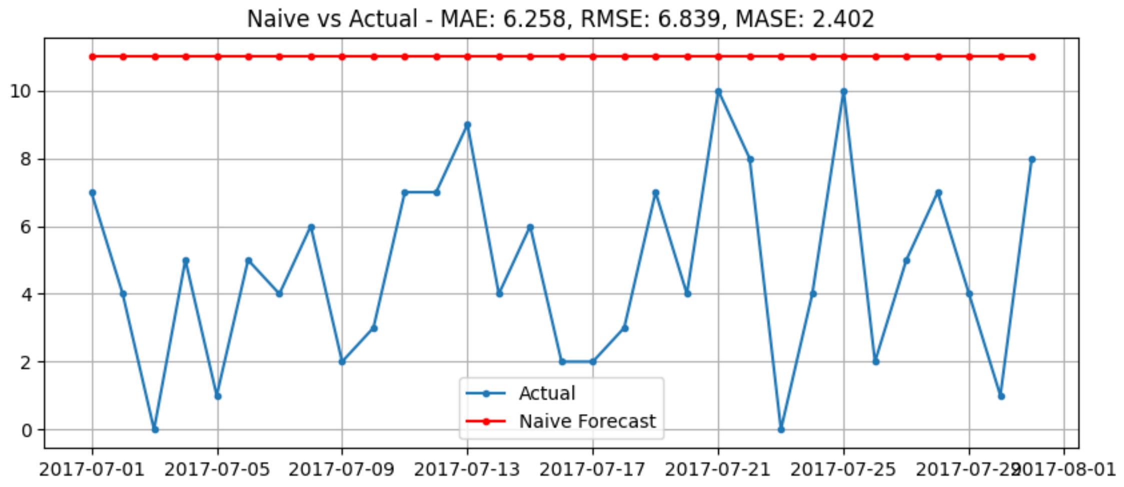
Setting up a test harness

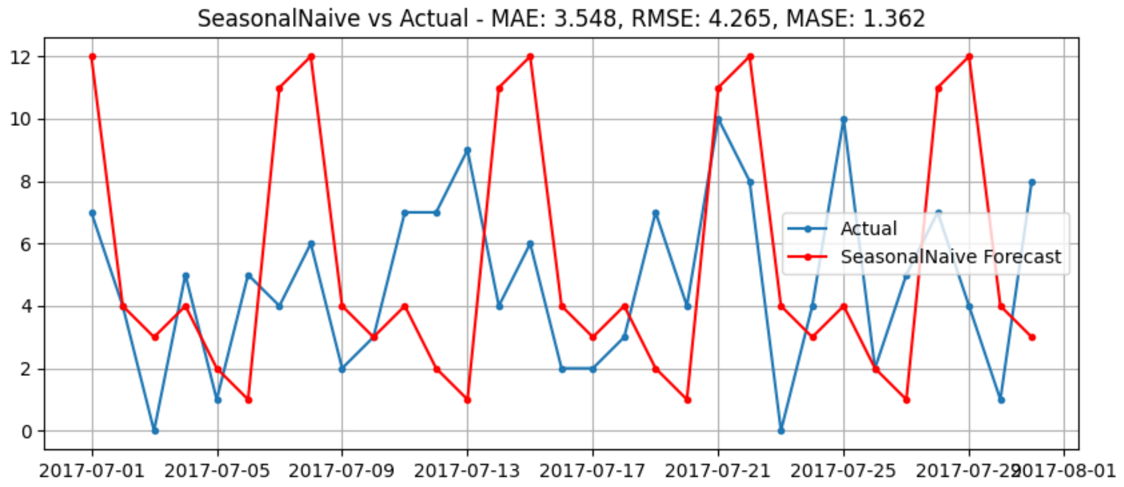
- Train, Validation and Test
- Evaluation metric
    - Mean Absolute Error (MAE)
    - Mean Square Error (MSE)
    - Mean Absolute Scaled Error (MASE)
    - Forecast Bias (FB)

Generating strong baseline forecasts

- Naïve Forecast
- Seasonal Naïve Forecast

Generating strong baseline forecasts

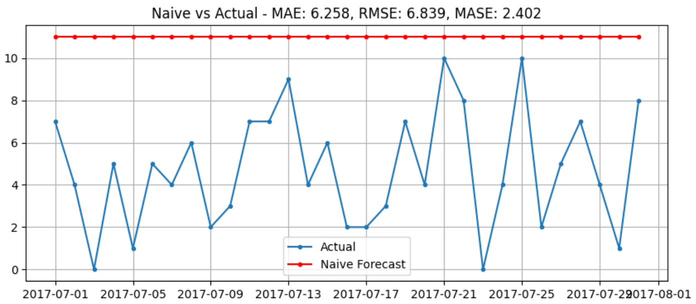


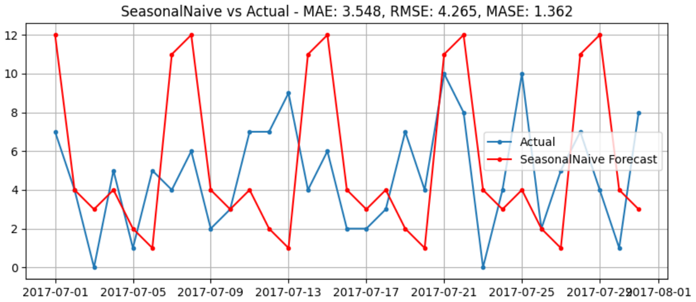


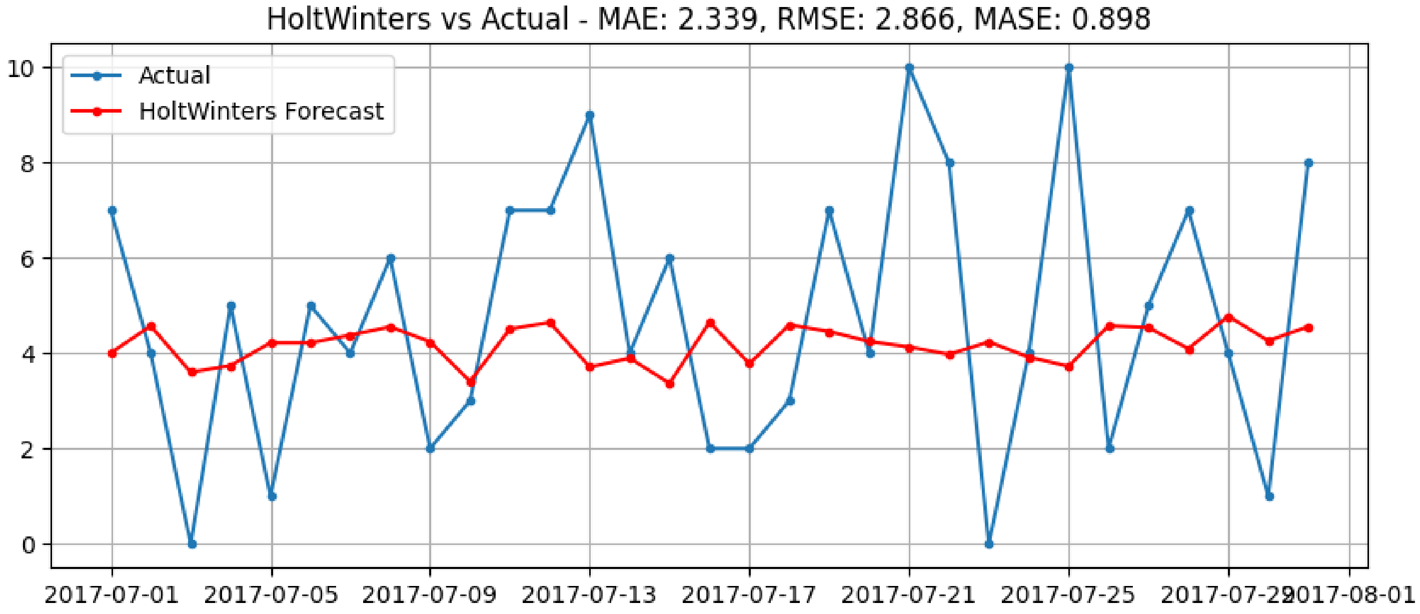
Generating strong baseline forecasts

- Naïve Forecast
- Seasonal Naïve Forecast
- Exponential smoothing
  - Calculate rolling mean with decreasing weights as we look farther back
  - Modeled around trend and seasonality

Generating strong baseline forecasts



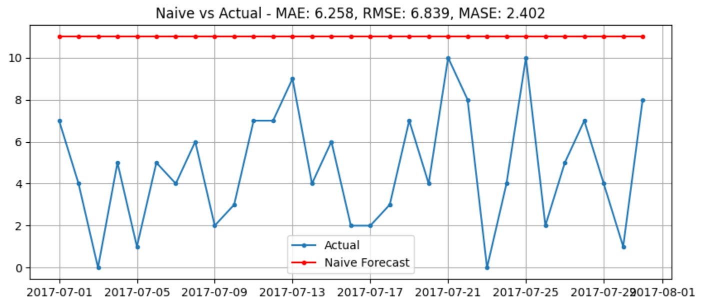


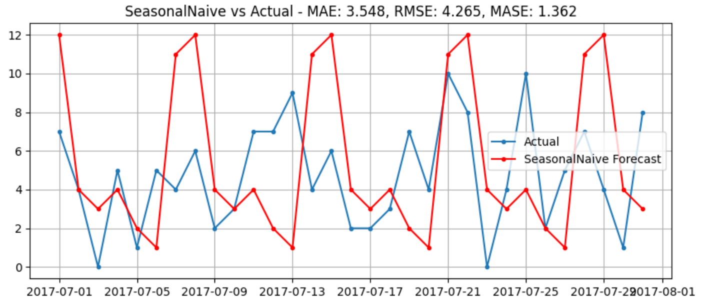


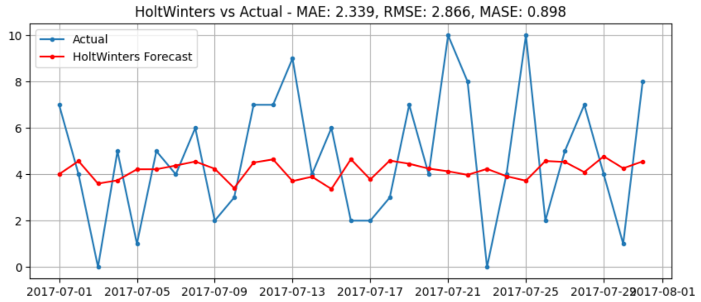
Generating strong baseline forecasts

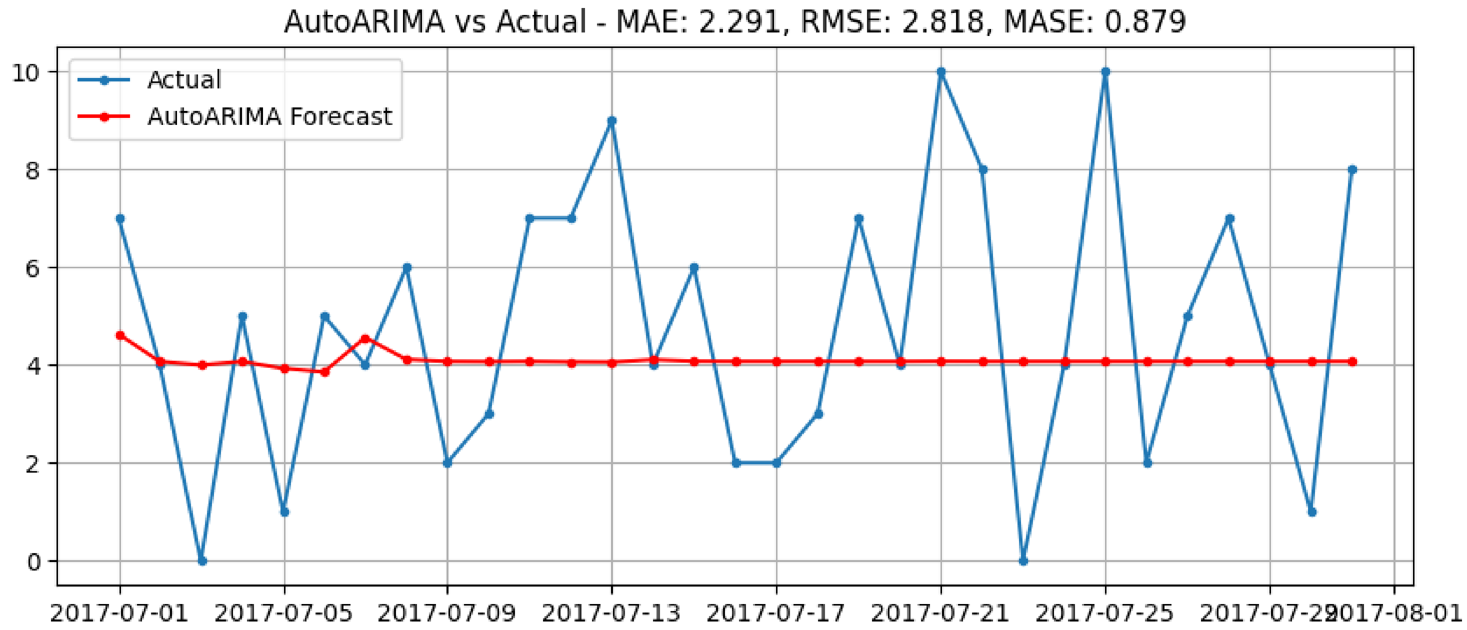
- Naïve Forecast
- Seasonal Naïve Forecast
- Exponential smoothing
- AutoRegressive Integrated Moving Average (ARIMA)
  - relies on autocorrelation (the correlation of y(t) with y(t-1), y(t-2), and so on)

Generating strong baseline forecasts









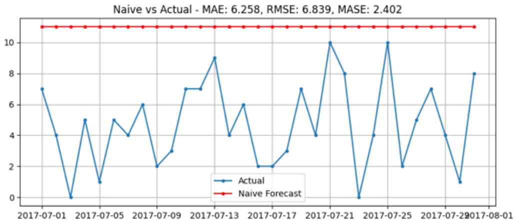
Flat Predictions Because

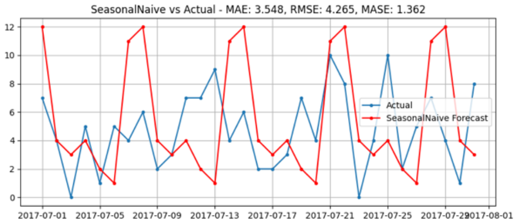
- AutoARIMA is not good for short spikes (daily sales)
- Settles to seasonal mean

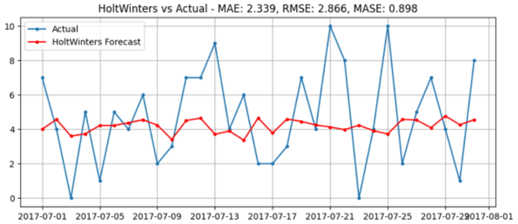
Generating strong baseline forecasts

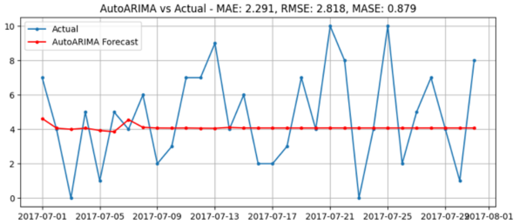
- Naïve Forecast
- Seasonal Naïve Forecast
- Exponential smoothing
- AutoRegressive Integrated Moving Average (ARIMA)
- Theta Forecast

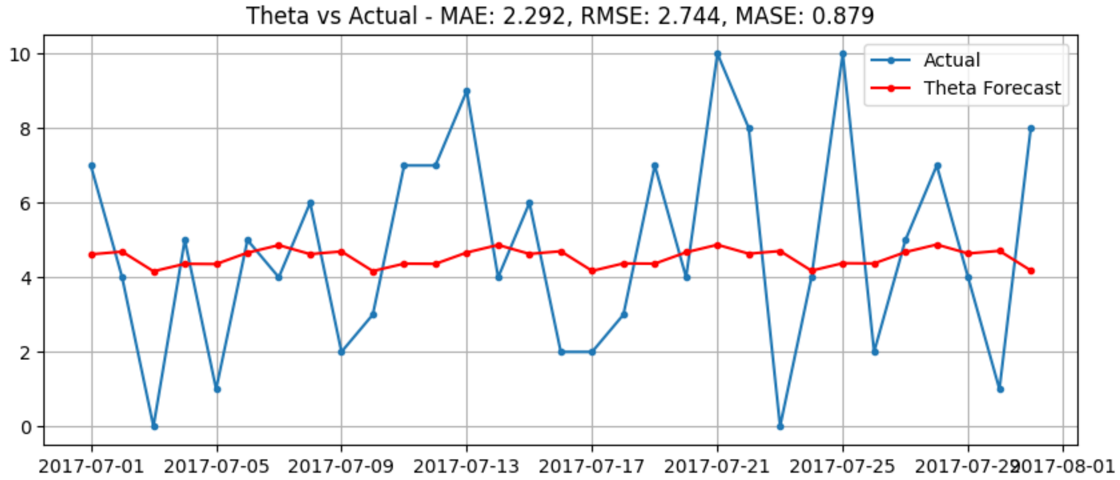
Generating strong baseline forecasts









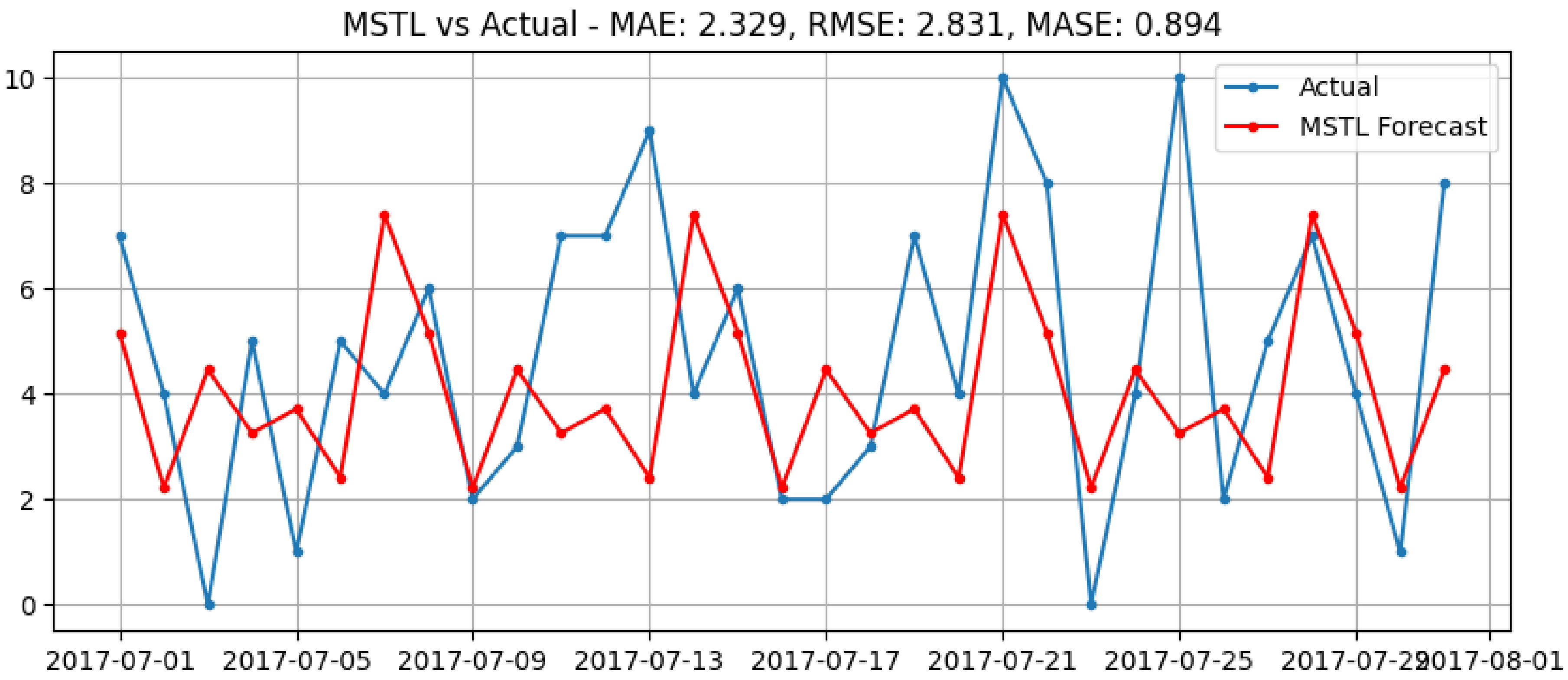


Captures the pattern
but not the peaks

Generating strong baseline forecasts

- Naïve Forecast
- Seasonal Naïve Forecast
- Exponential smoothing
- AutoRegressive Integrated Moving Average (ARIMA)
- Theta Forecast
- Multiple Seasonal-Trend decomposition using LOESS (MSTL)
  - Use regression technique to extract Trends, Cycles, Seasonality and Irregularity
  - Employs non seasonal model to forecast and then add the components

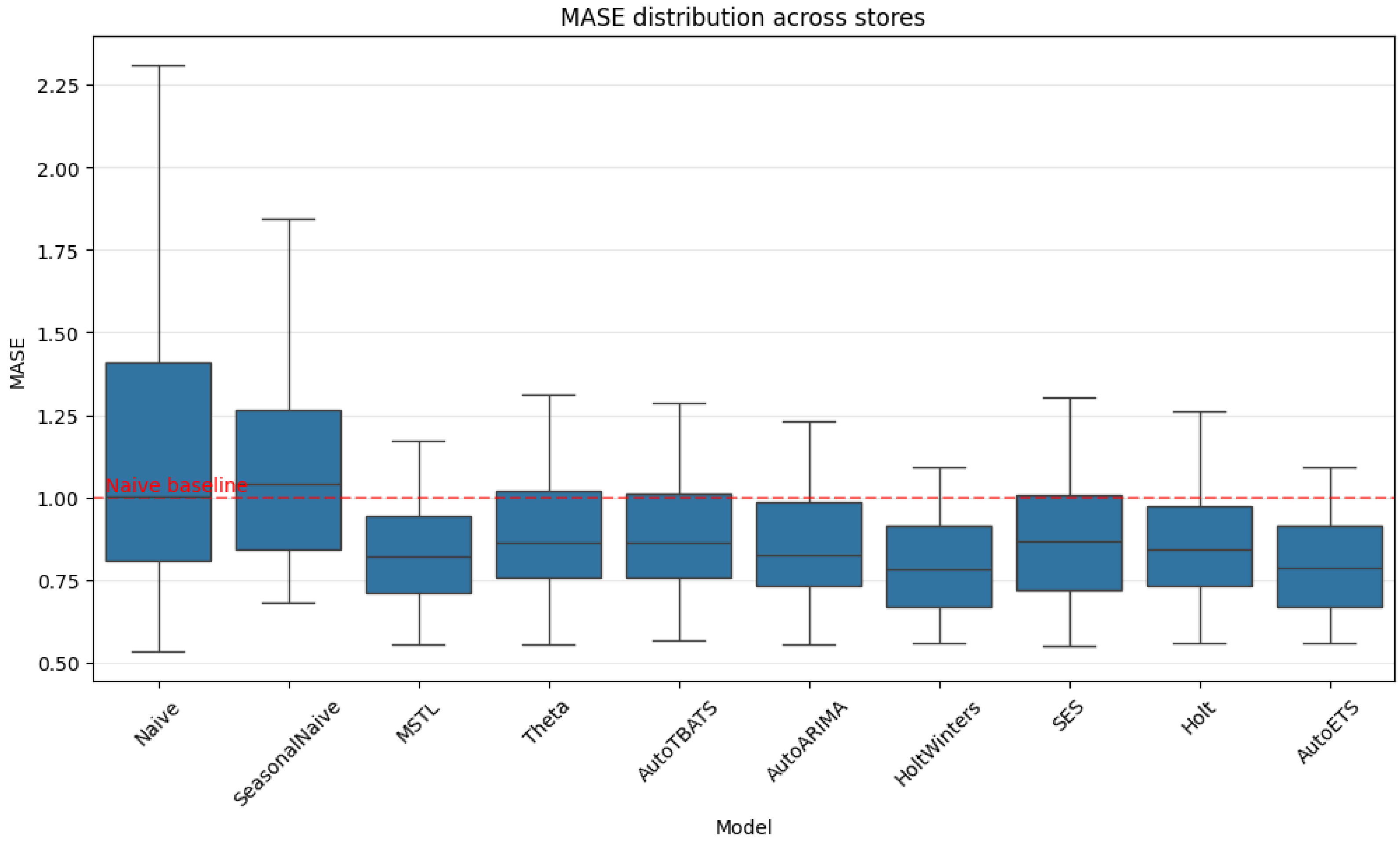
Generating strong baseline forecasts



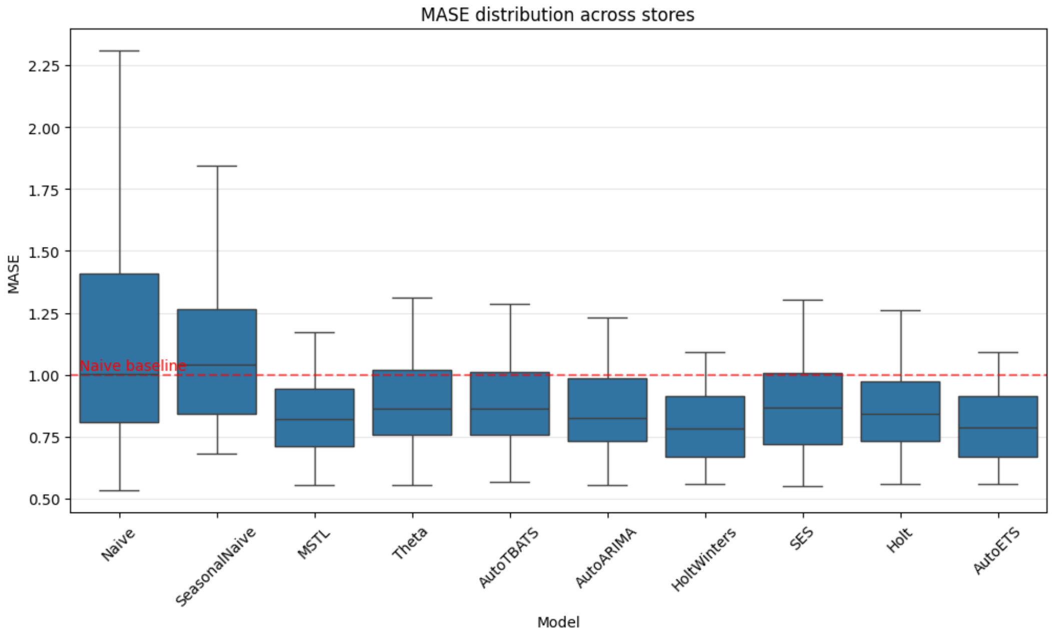
Model Trials: for a specific product family

```
for store_nbr in list(df.store_nbr.unique()):
    df_store_nbr=df[df['store_nbr']==store_nbr]
    for model_name, model in models_dict.items():
        metrics = evaluate_model(
        model=model,
        model_name=model_name,
        ts_train=ts_train,
        ts_val=ts_val,
        _ts_train=_ts_train,
        _ts_val=_ts_val,
        freq='D',
        seasonality=7
        )
```

Model Trials: for a specific product family across all stores



Model Trials: for a specific product family across all stores

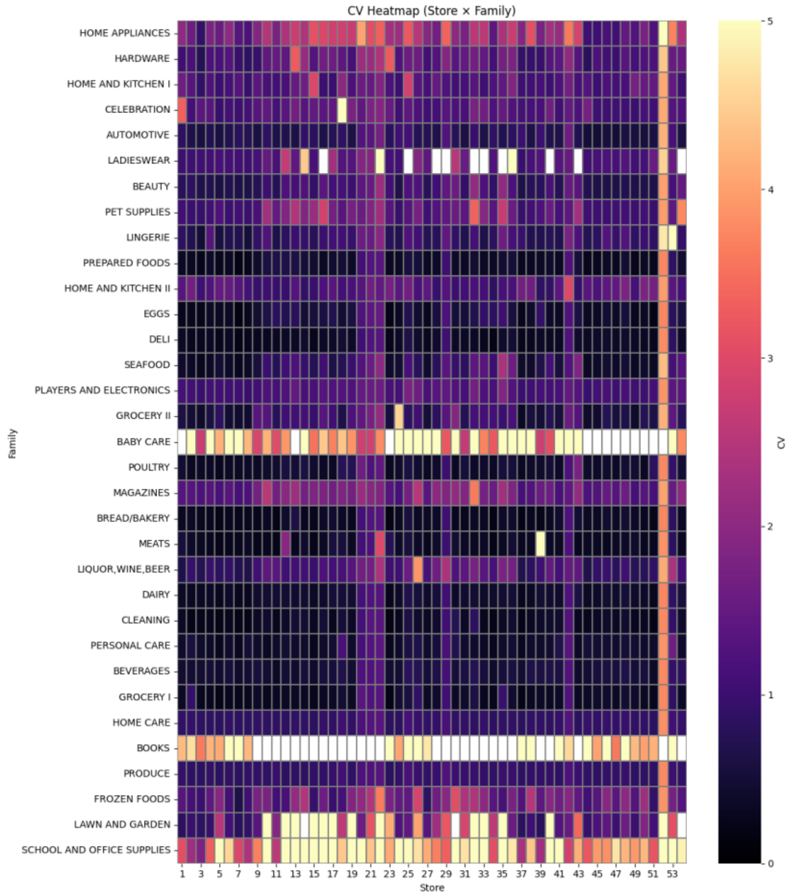


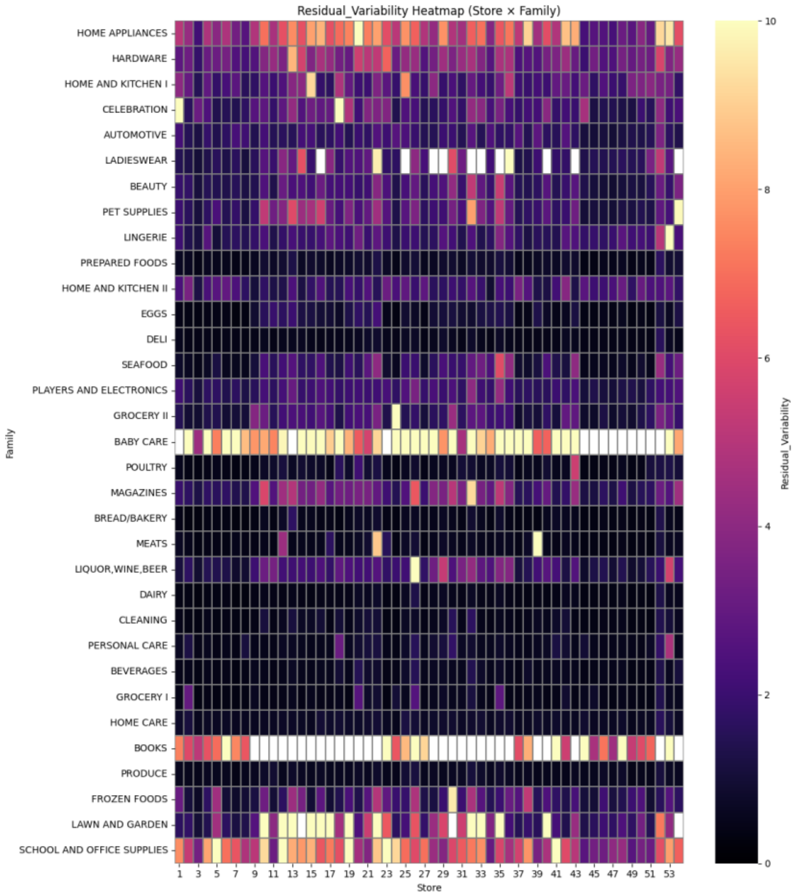
Model	Num_Wins
HoltWinters	**19**
MSTL	**12**
SES	8
AutoETS	5
Naive	3
AutoTBATS	2
Theta	2
Holt	1
AutoARIMA	1
SeasonalNaive	1

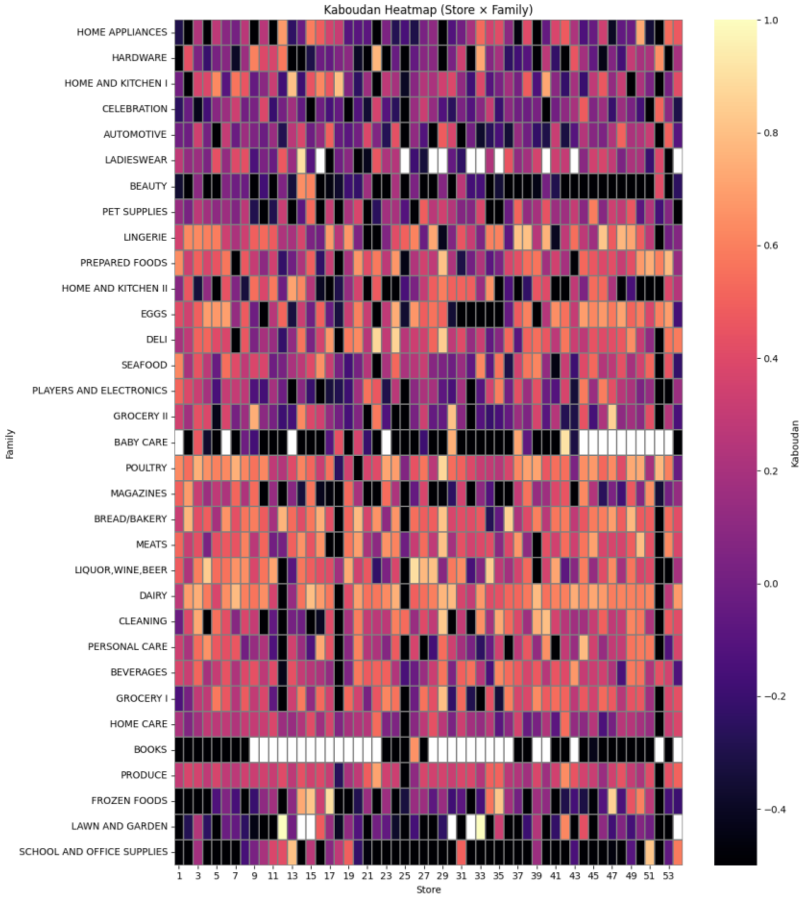
Forecastability

- Coefficient of variation
  - How variable the series is relative to its average level?
- Residual variability
  - How much randomness remains after removing trend and seasonality?
- Spectral entropy
  - How spread out the signal's energy is across frequencies?
- Kaboudan metric
  - How much better a model forecasts the real series compared to a shuffled version?

Forecastability

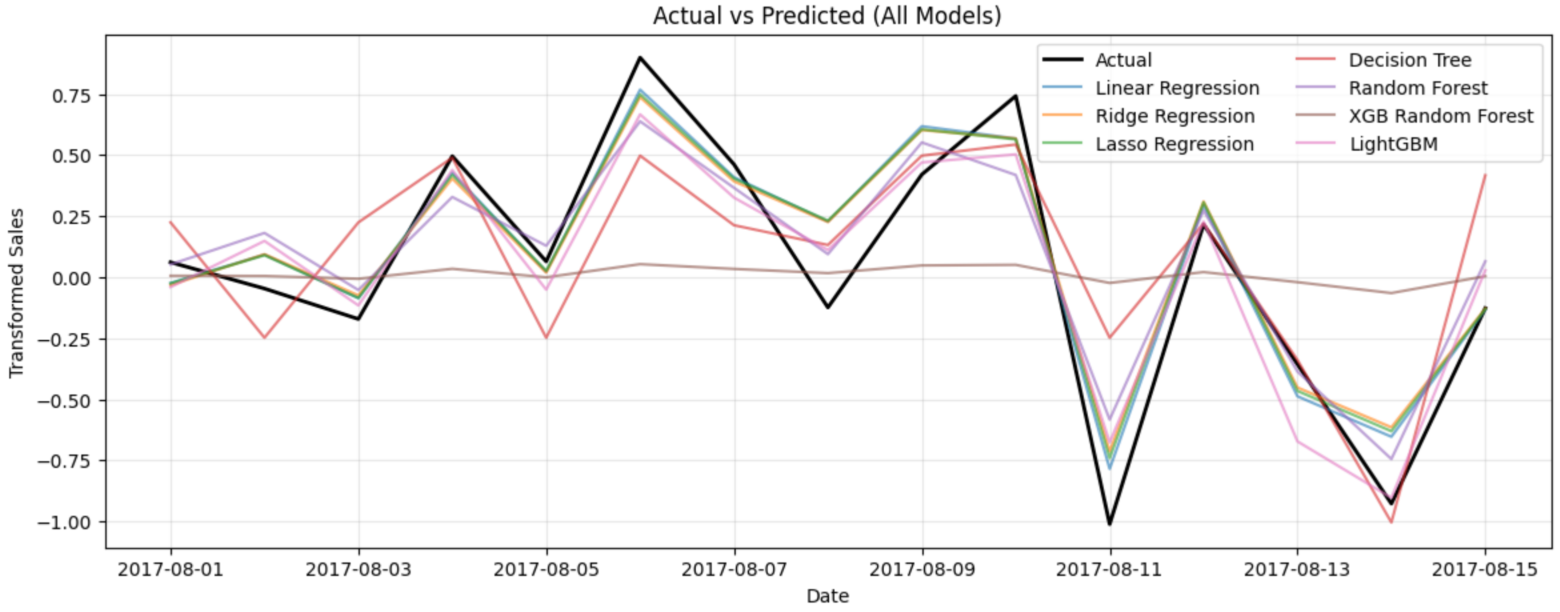




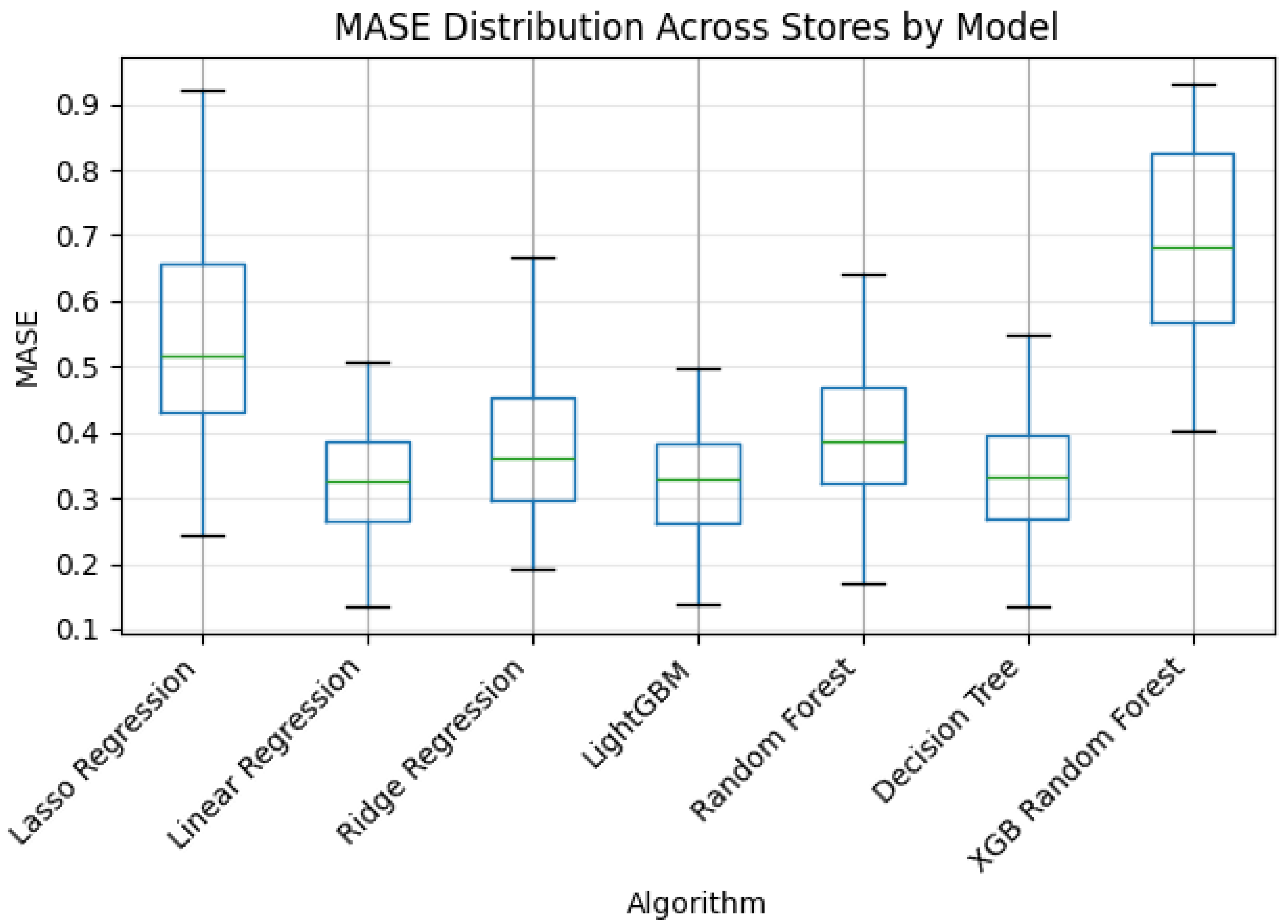


Machine Learning For Time Series

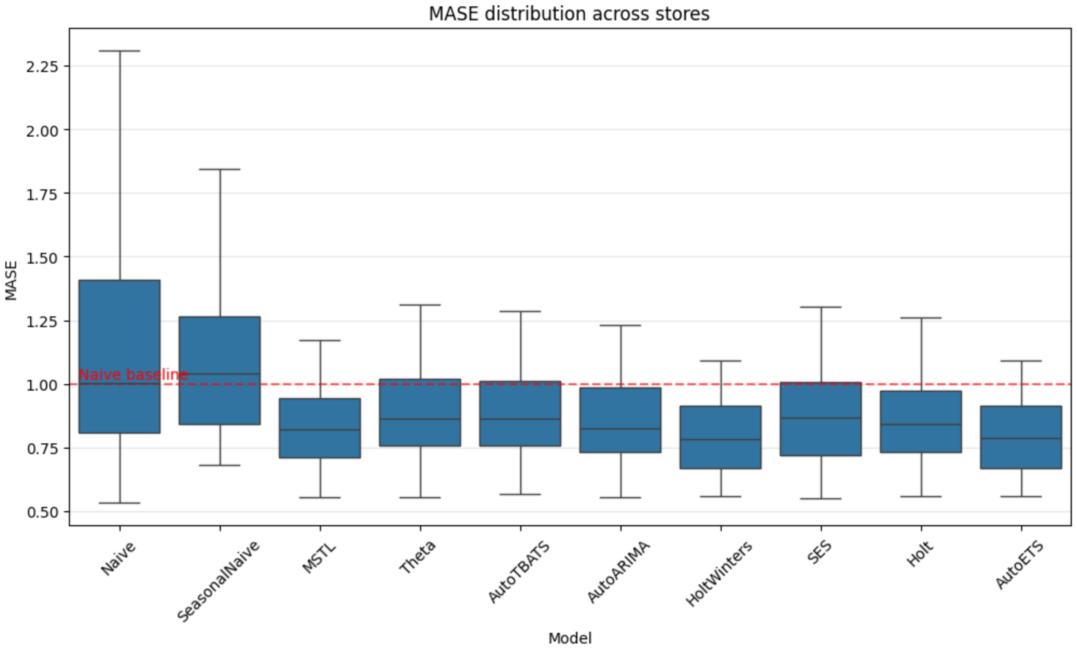
- Target transformations (log, detrend, deseasonalize)
- Time-delay features (lags, rolling stats)
- Temporal features (calendar, Fourier)
- Regression / ML models

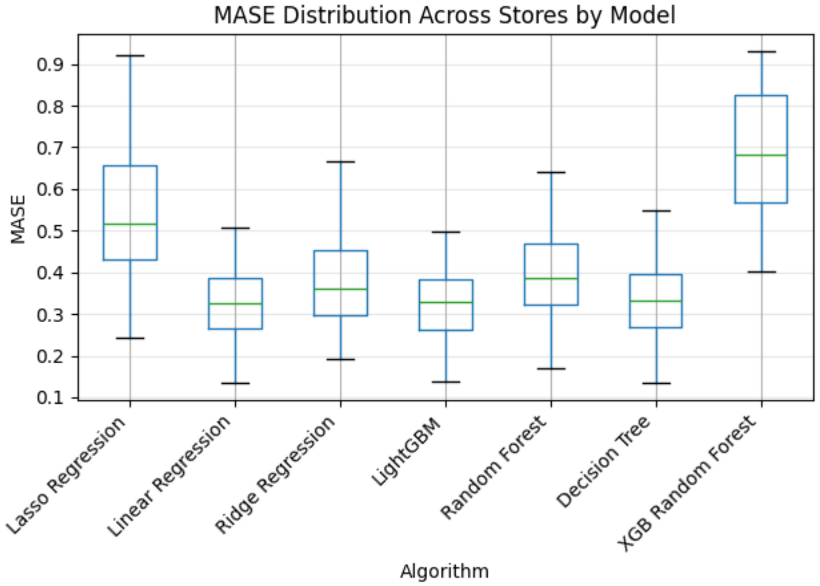


ML: for a specific product family across all stores



Statistical vs ML methods

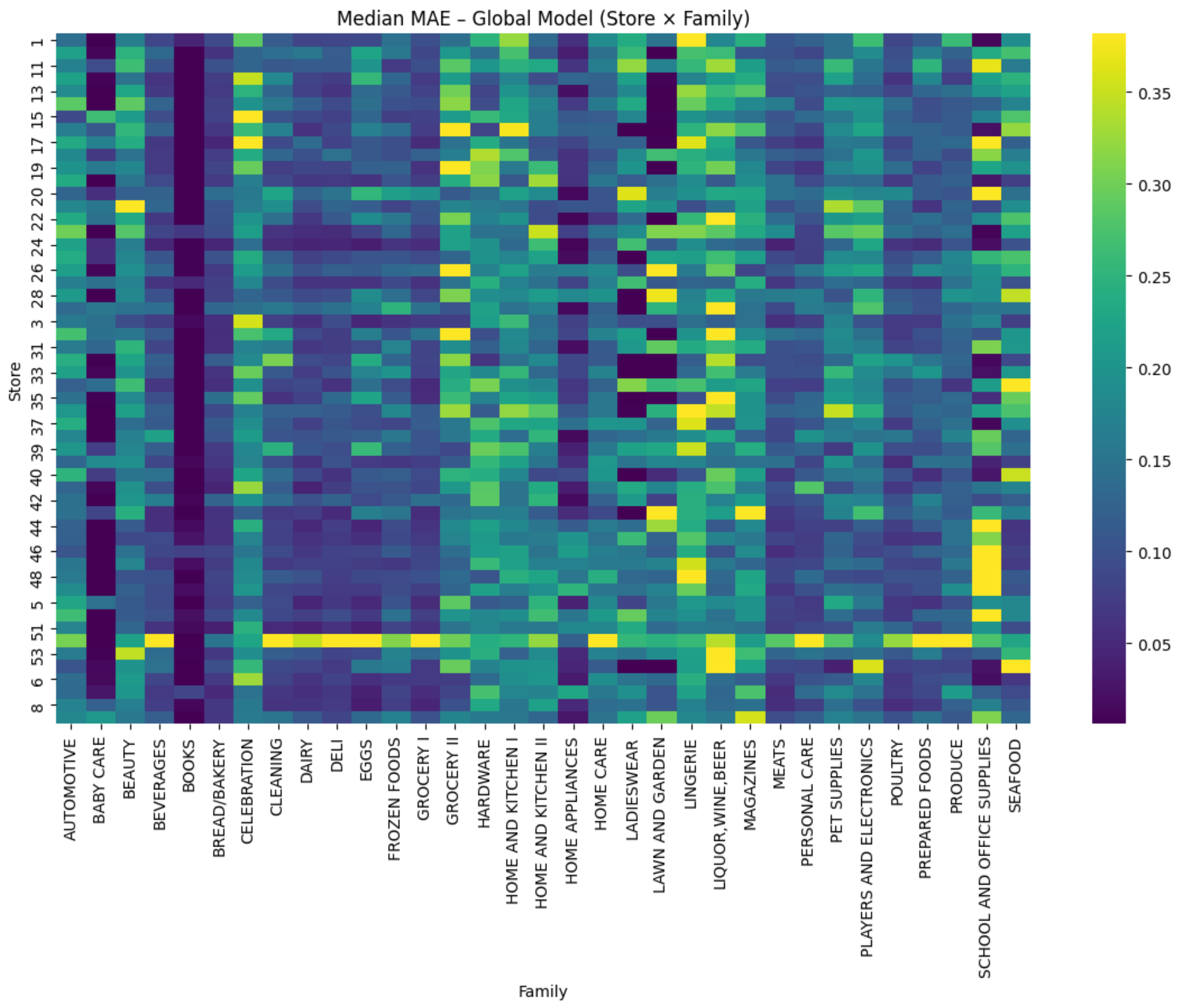




Global Forecast Model

- Training one model per store-product combination is inefficient
- Not able to take advantage of the entire time series (data quantity)
- All related time series share a common underlying structure, even if their levels differ.
- Instead of fitting N models for N series, we fit one model across all series.

Global Forecast Model



Deployment

Training System

- Builds / updates models
- Runs offline (daily / weekly)

Inference system

- Generates forecasts
- Must be reliable, fast, reproducible
- Often batch (not real-time)

Monitoring & retraining system

- Detects drift / degradation
- Check model health and retrain

Deployment: Retraining

- Scheduled retraining
- Rolling window retraining
- Performance-triggered retraining

Reference Materials Used

- Modern Time Series Forecasting with Python, Second Edition, Manu Joseph, Jeffrey Tackes
- https://github.com/Nixtla/statsforecast
- https://github.com/Nixtla/mlforecast
- https://www.kaggle.com/competitions/store-sales-time-series-forecasting/overview