



**Store sales  
Time series forecasting.**



# Problem Statement

**Stores across the world face challenges in optimizing their stocks due to lack of sales forecasts and decision-making processes.**

These discrepancies lead to:

- High Costs & Lost Sales (due to overstock or understock)
- Budget Waste (from poor resource allocation)
- Missed Sales Opportunities (due to lack of responsiveness to market shifts)



# Applications and Impact



**Retail & Consumer Goods:** Optimize stocking decisions across all categories to minimize waste and maximize customer satisfaction. This includes:

- Anticipating demand fluctuations.
- Making better decisions for marketing, sales, production, and finance.
- Improving inventory management efficiency.

**Manufacturing:** Accurate forecasts enable manufacturers to plan production efficiently for various products, avoiding material shortages and production delays.



# Literature review

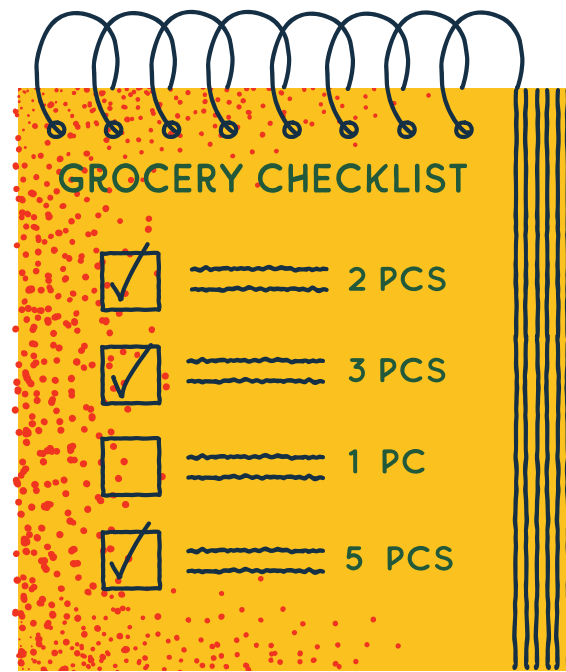
## Sales Forecasting for Retail Chains

Ankur Jain , Manghat Nitish Menon , Saurabh Chandra, @eng.ucsd.edu



<b>Model</b>	<b>RMPSE on Test Set</b>
Mean of each DayOfWeek	0.18968
Linear Regression	0.15672
Random Forest Regression	0.13198
XGBoost	0.10532





## Handling Nonlinear Relationships

XG-Boost captures intricate nonlinear relationships in sales data, crucial for modeling complex sales patterns influenced by various factors.

## Regularization

Incorporating regularization, prevents overfitting, vital for accurate forecasting amidst noisy or outlier-rich data.

## Gradient Boosting

XGBoost iteratively refines predictions, enhancing accuracy by combining weak learners and focusing on previous residuals.

## Feature Importance

XG-Boost reveals feature importance, aiding in identifying key drivers of sales, facilitating informed decision-making.

## Scalability

XGBoost is scalable and efficient, handling large datasets effectively, making it ideal for diverse retail sales forecasting.



Supermarket sales Prediction using regression. (2021). International Journal of Advanced Trends in Computer Science and Engineering, 10(2), 1153–1157.  
<https://doi.org/10.30534/ijatcse/2021/951022021>

#### 4.4 XGBoost Algorithm

XGboost is the one of the most popular and one of the highest accuracy providing machine learning algorithm used in the present day and is implemented regardless on the type of prediction task at hand ie whether it is regression or classification. It is an implementation of decision trees which are gradient boosted which are designed for performance and speed that is competitive machine learning. This algorithm is well known for providing better solutions as compared to other machine learning algorithms. In fact, since it has been developed, it has become the "state-of-the-art" machine learning algorithm to deal well-structured data.

ALGORITHMS	ACCURACY
Linear Regression Algorithm	56.0
Ridge Regression Algorithm	46.0
Lasso Regression Algorithm	55.0
Decision Tree Algorithm	62.0
Random Forest Algorithm	61.0
XGBoost Algorithm	88.51

**Table 1:** Algorithm and Accuracy

Journal of Electronics, Computer Networking and Applied Mathematics (JECNAM) ISSN : 2799-1156. (n.d.). <http://journal.hmjournals.com/index.php/JECNAM>

Predicting Retail Sales: A Comparative Study of Regression Analysis and Artificial Neural Networks". This study compares the accuracy of linear regression with artificial neural networks in predicting retail sales. Findings suggest that while linear regression provides reliable predictions, neural networks may offer advantages in capturing complex non-linear relationships.

Predictive Analytics in E-commerce: A Case Study Using Linear Regression. This case study delves into the practical application of linear regression in e-commerce sales prediction. It provides insights into how businesses can leverage historical data to optimize marketing strategies and inventory management through accurate sales forecasts.



**Elmasdotter, A., & Nyströmer, C. (2018). A comparative study between LSTM and ARIMA for sales forecasting in retail (Dissertation). Retrieved from <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-229747>**

In the first scenario the models predict sales for one day ahead using given data, while they in the second scenario predict each day for a week ahead. Using the evaluation measures RMSE and MAE together with a t-test the results show that the difference between the LSTM and ARIMA model is not of statistical significance in the scenario of predicting one day ahead. However when predicting seven days ahead, the results show that there is a statistical significance in the difference indicating that the LSTM model has higher accuracy. **This study therefore concludes that the LSTM model is promising in the field of sales forecasting in retail and able to compete against the ARIMA model.**





# About the Dataset



**Corporación Favorita, a large Ecuadorian-based grocery retailer.**

**54 stores, 33 product families across 22 cities.**

**Total 3008280 entries in training dataset**

**The time serie starts from 2013-01-01 and finishes in 2017-08-31.**

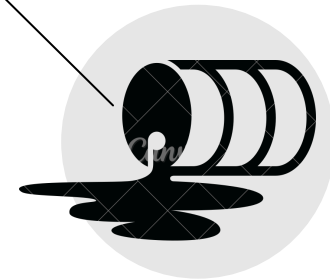
**This is a multivariate Time-series data.**



# Kaggle competition- Data provided.

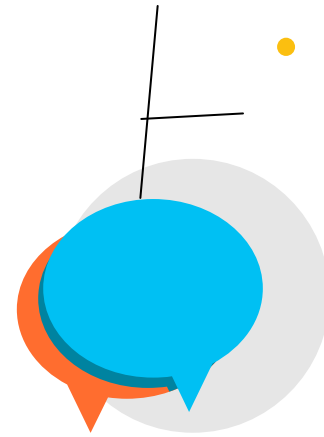
## Oil.csv

date,dcoilwtico



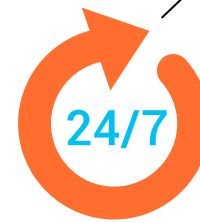
## Stores.csv

store\_nbr,city,state,type,cluster



## Train.csv

id,date,store\_nbr,family,sales,on  
promotion



## Transactions.csv

date,store\_nbr,transactions



## Test.csv

id,date,store\_nbr,family,  
onpromotion

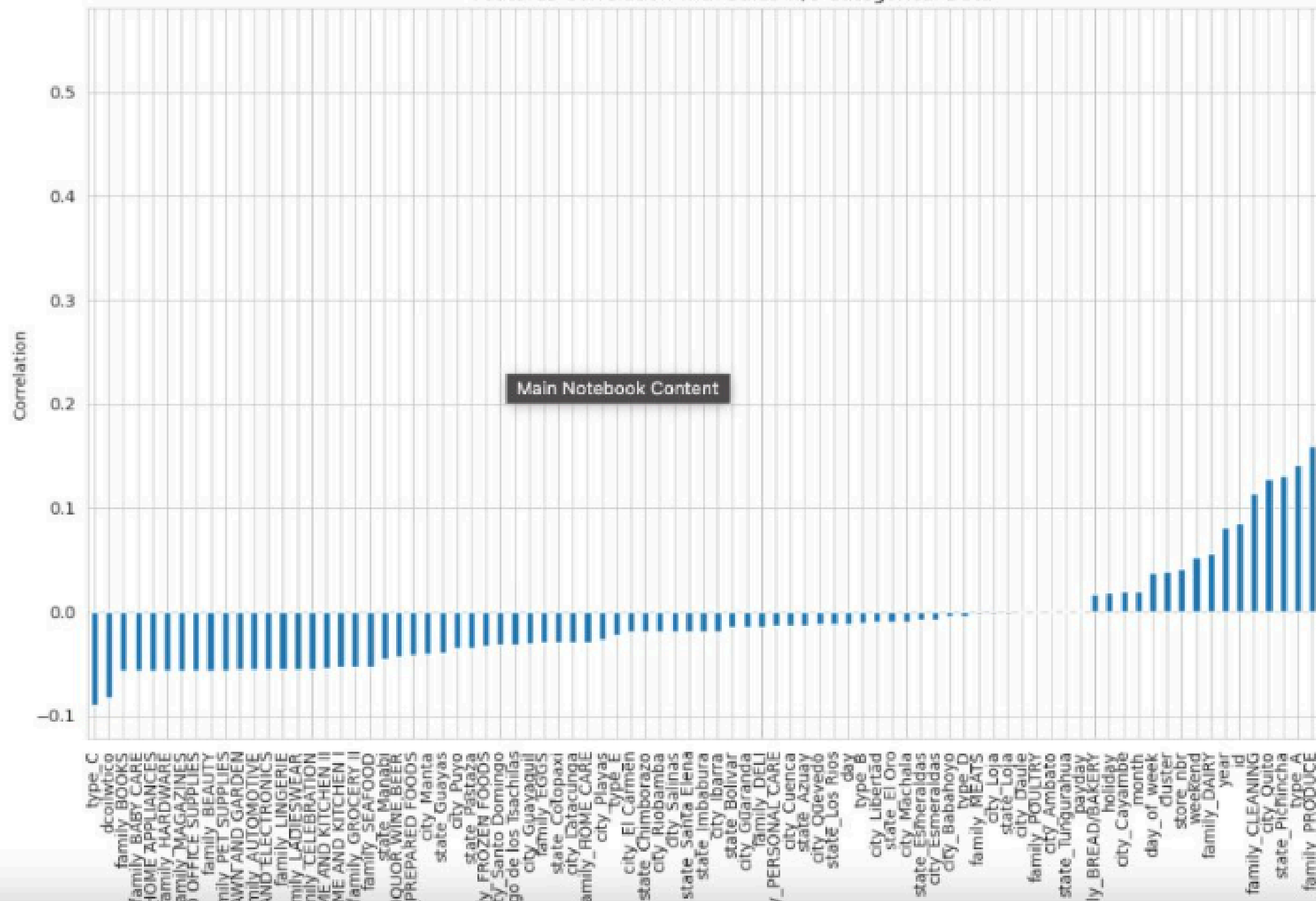


## Holiday events.csv

date,type,locale,locale\_name,  
description,transferred

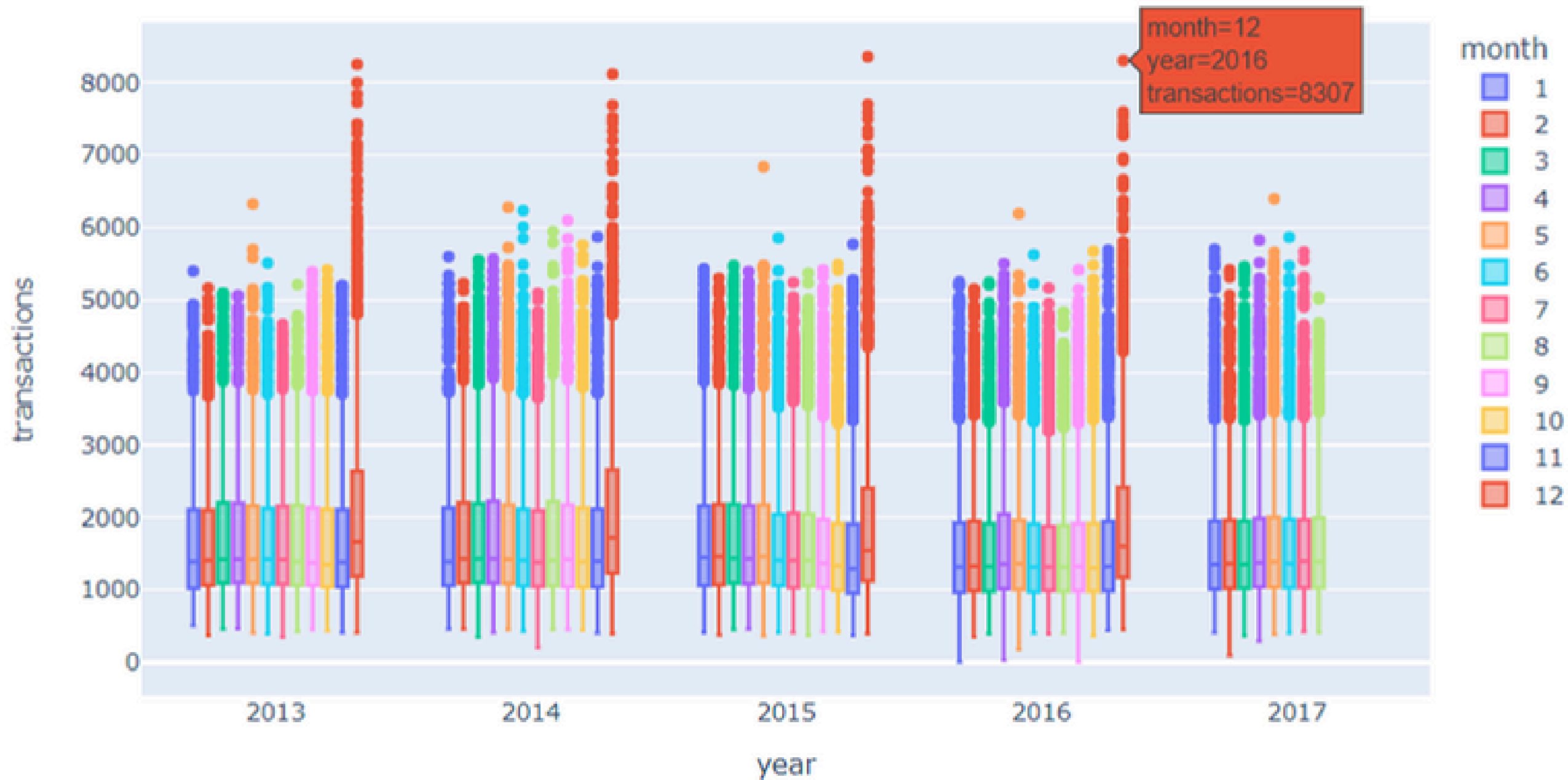


Features Correlation with Sales w/o Categorical Data



**Most of the features having weak correlation(-0.3, 0.3) with target variable is part of solution set and can't be removed**

## Transactions

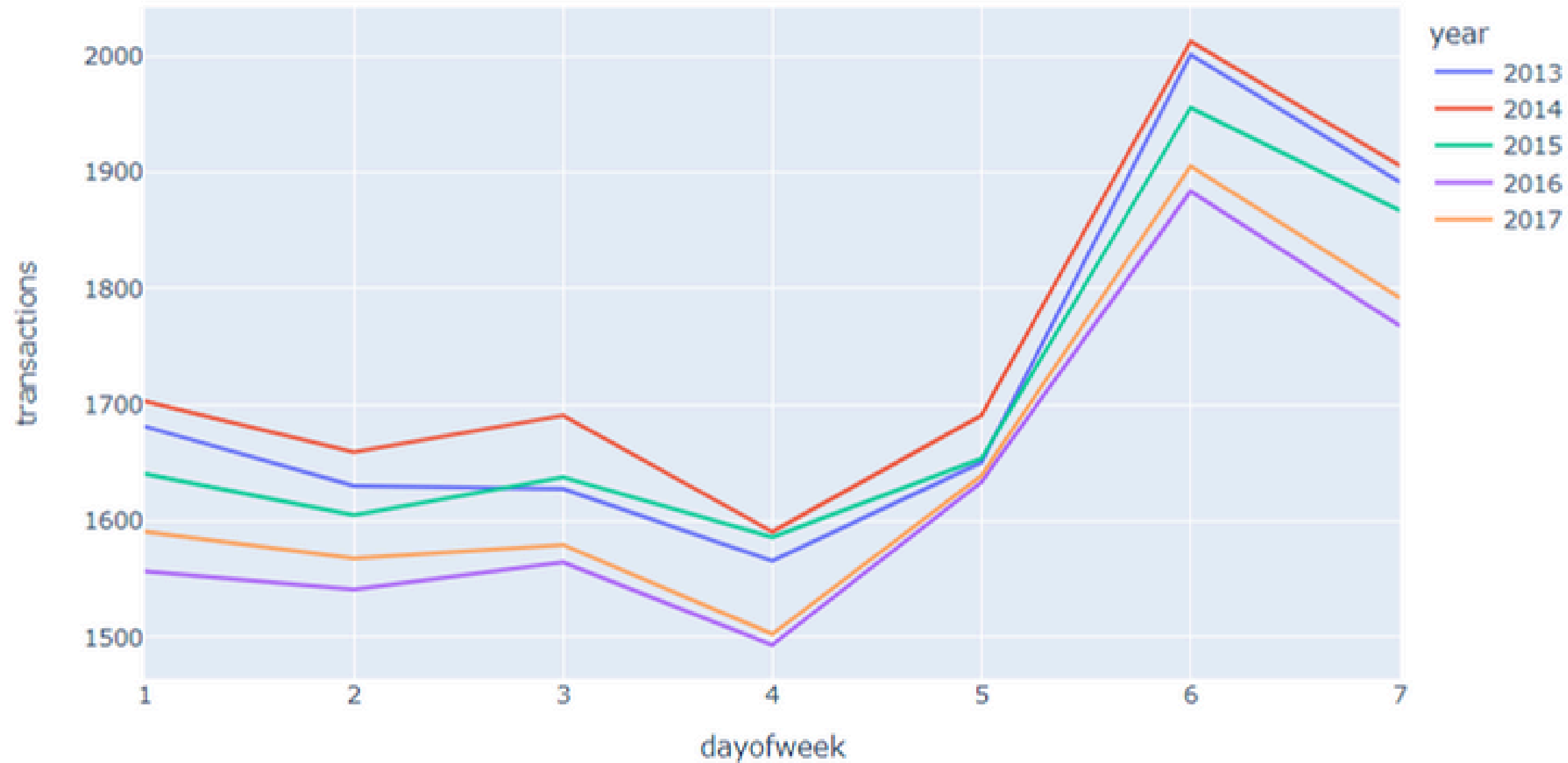


**Correlation between  
Total Sales and  
Transactions: 0.8175**

**All months are similar except December from 2013 to 2017 by boxplot.  
Store sales had always increased at the end of the year.**



## Transactions



Happy 😊  
SHOPPING

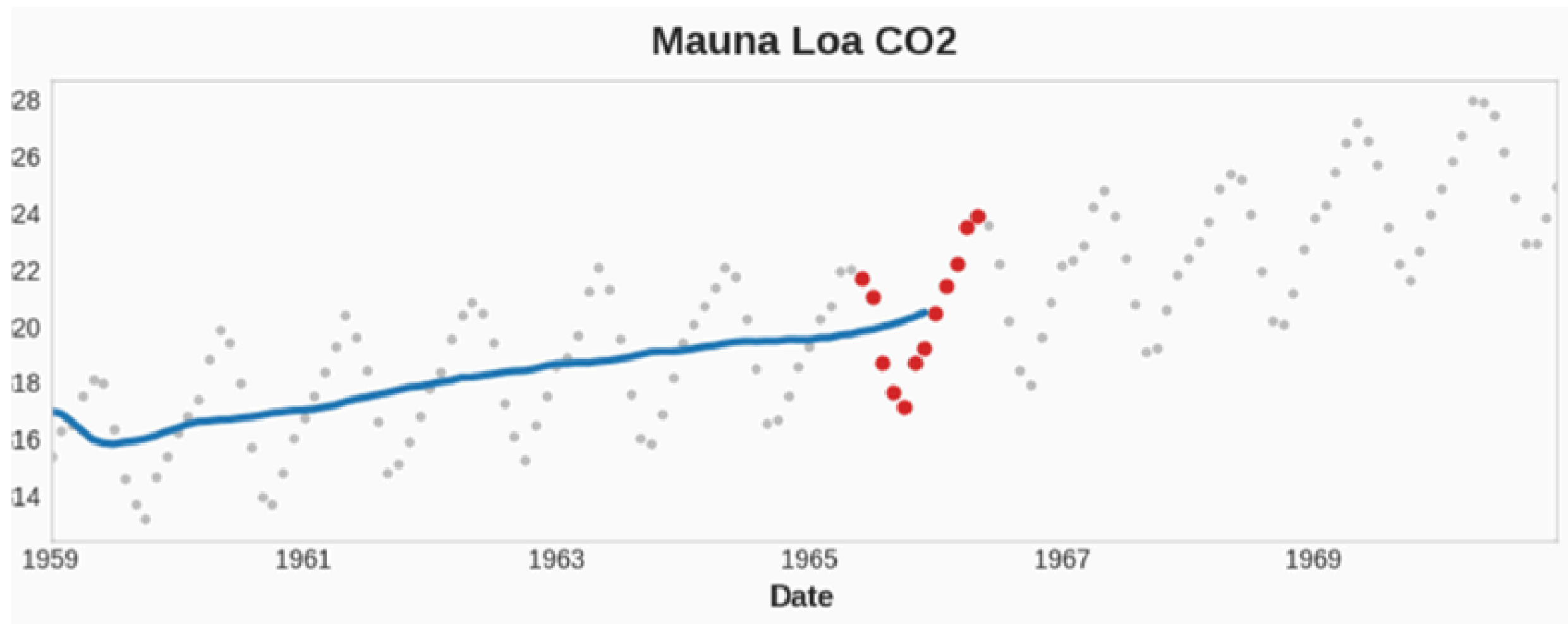


**Stores make more transactions at weekends. The patterns is almost same for 2013 to 2017, Saturday is the most important day for shopping.**



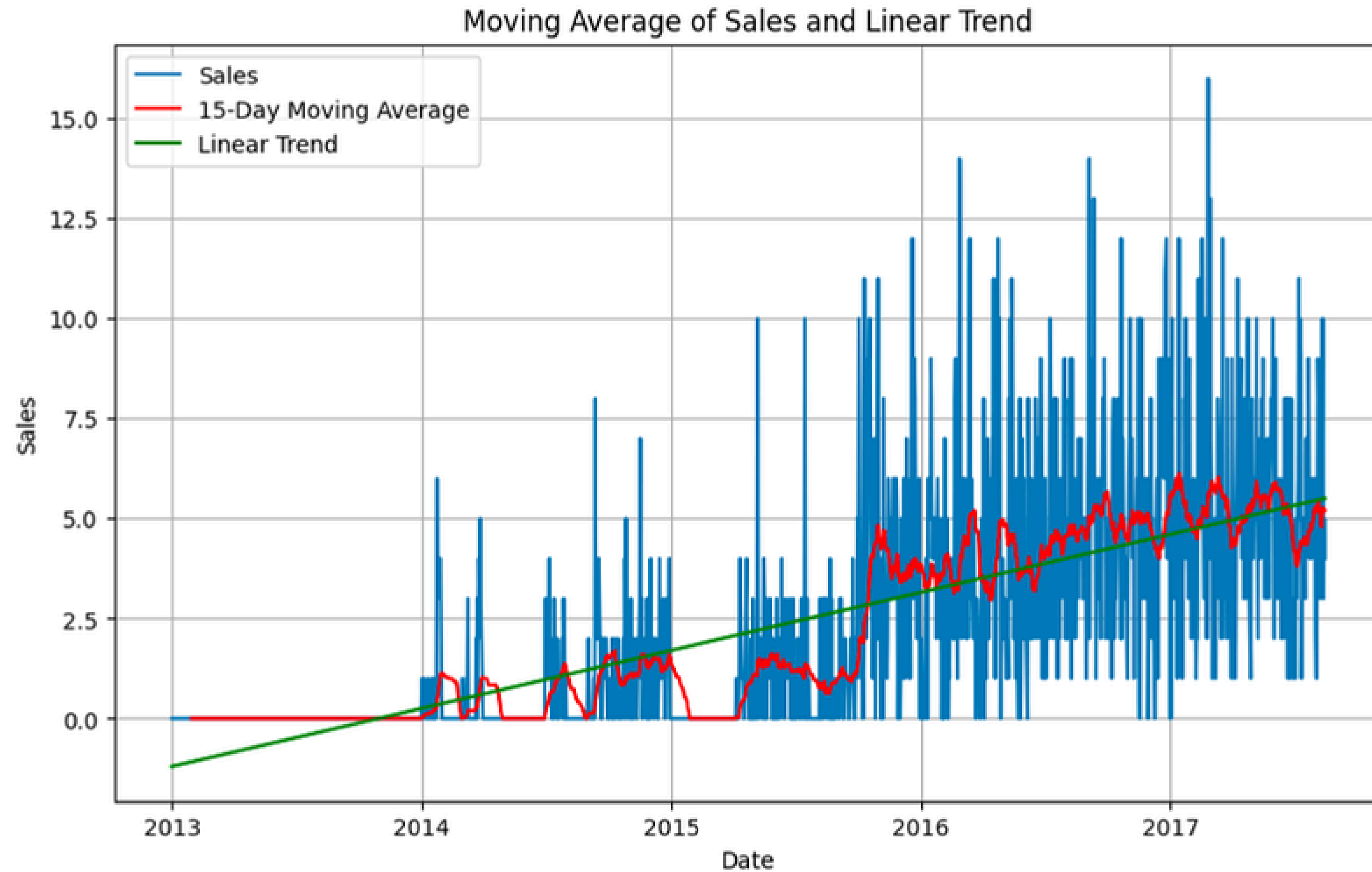
# Feature engineering to model the major time series components

- Trend
- Seasonality
- Cycle



**Moving average is used to find the trend in a dataset.**

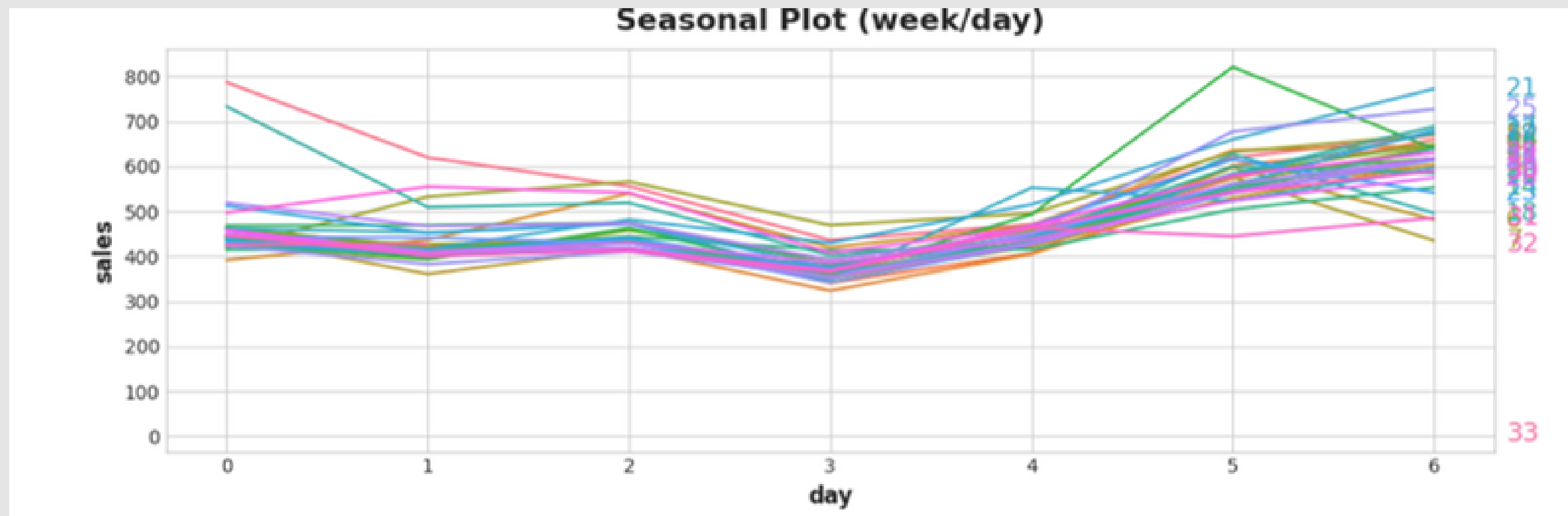
*We computed the moving average of sales data it showed linear trends.*



$$\text{target} = b * \text{time} + c$$

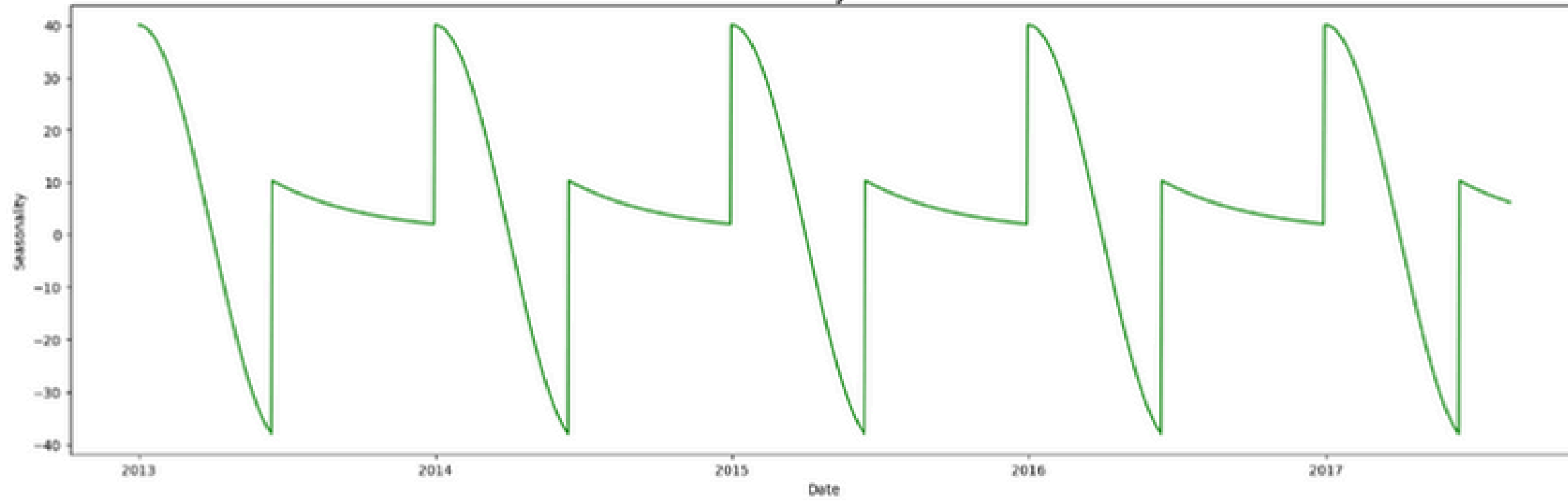
# Seasonality.

Seasonal indicators are binary features that represent seasonal differences in the level of a time series.

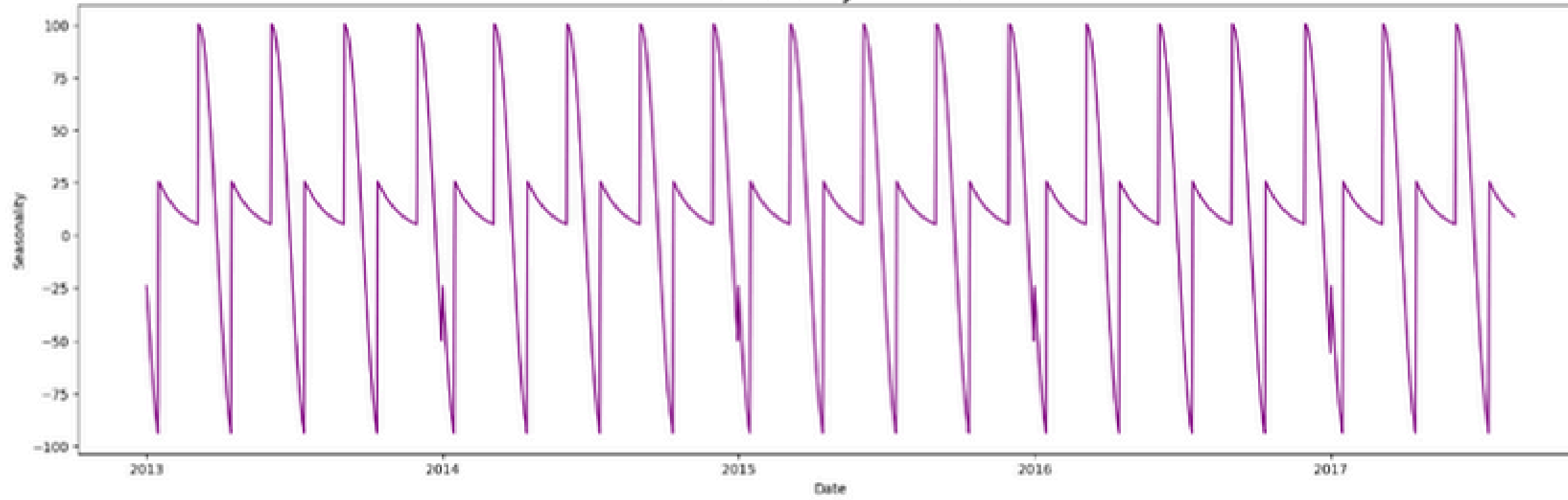




Seasonality Plot - 1



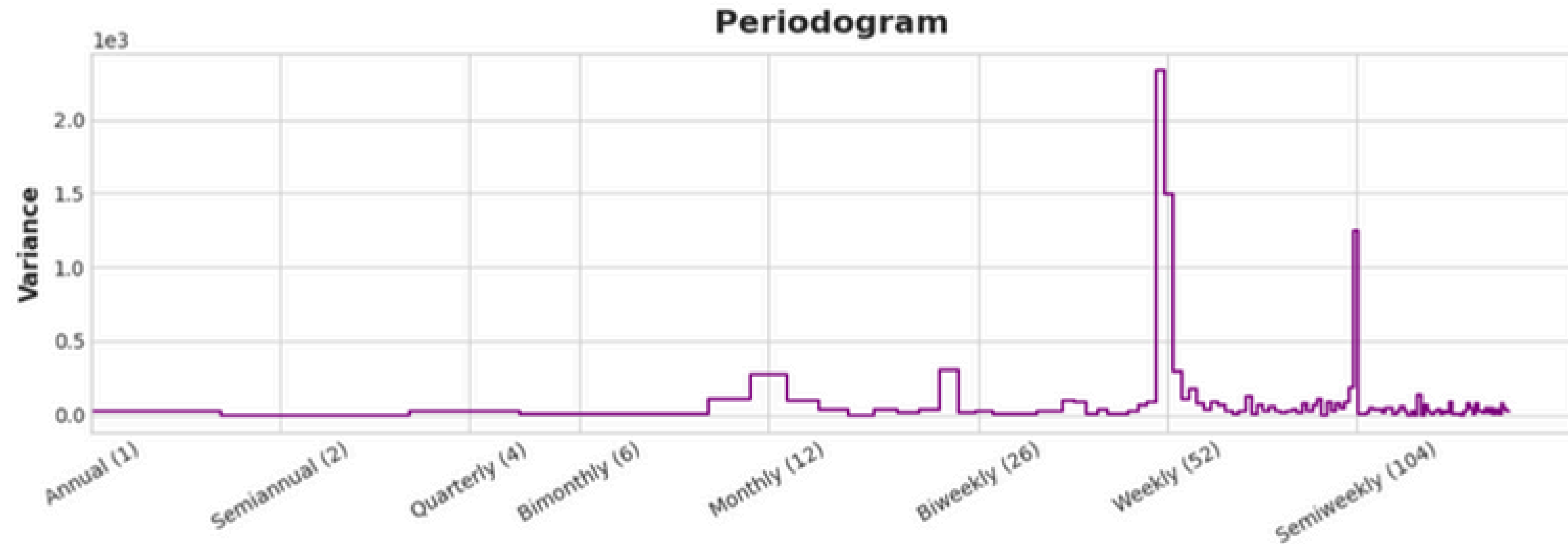
Seasonality Plot - 2



**Annual and Quarterly seasonal Plot**

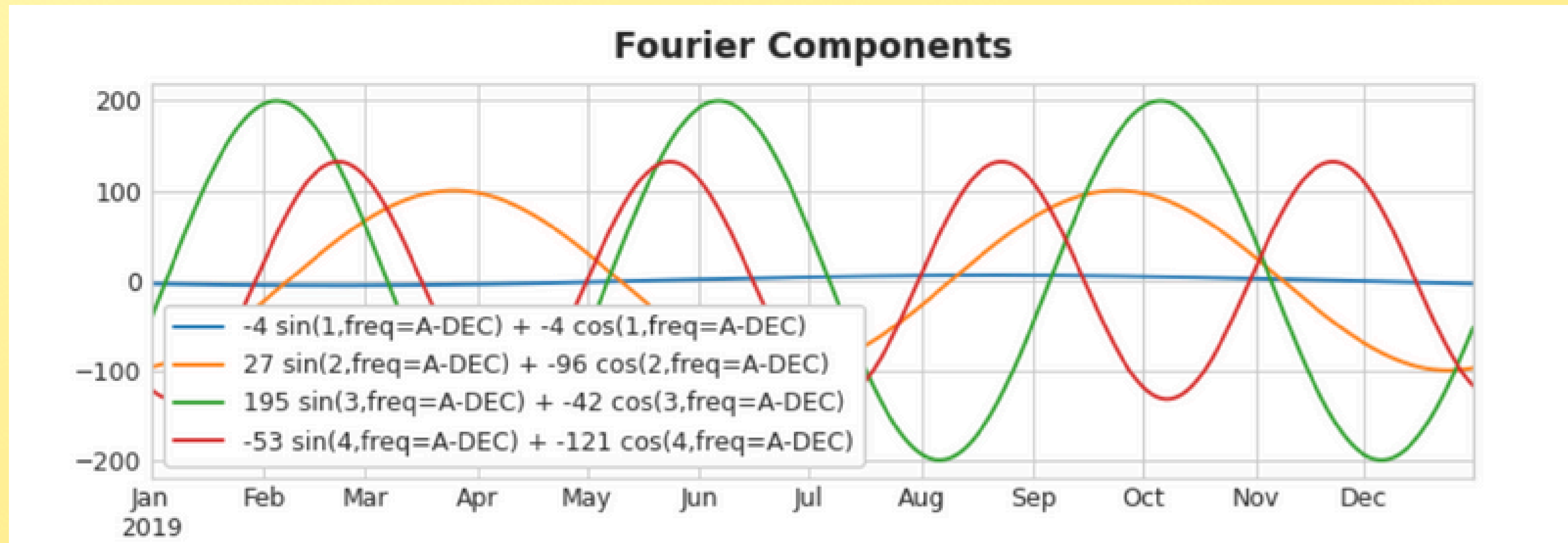
**All Yearly, quarterly, and weekly showed some sort of seasonality.**

**Periodogram shows how the variance of the time series data is distributed across different frequencies**



**So to capture these variance we use Fourier Features.**

**Fourier features try to capture the overall shape of the seasonal curve with just a few features.**



**Cycles**



**common way for serial dependence to manifest is in cycles**



**Lag Feature is used to cover the dependence on the previous steps.**



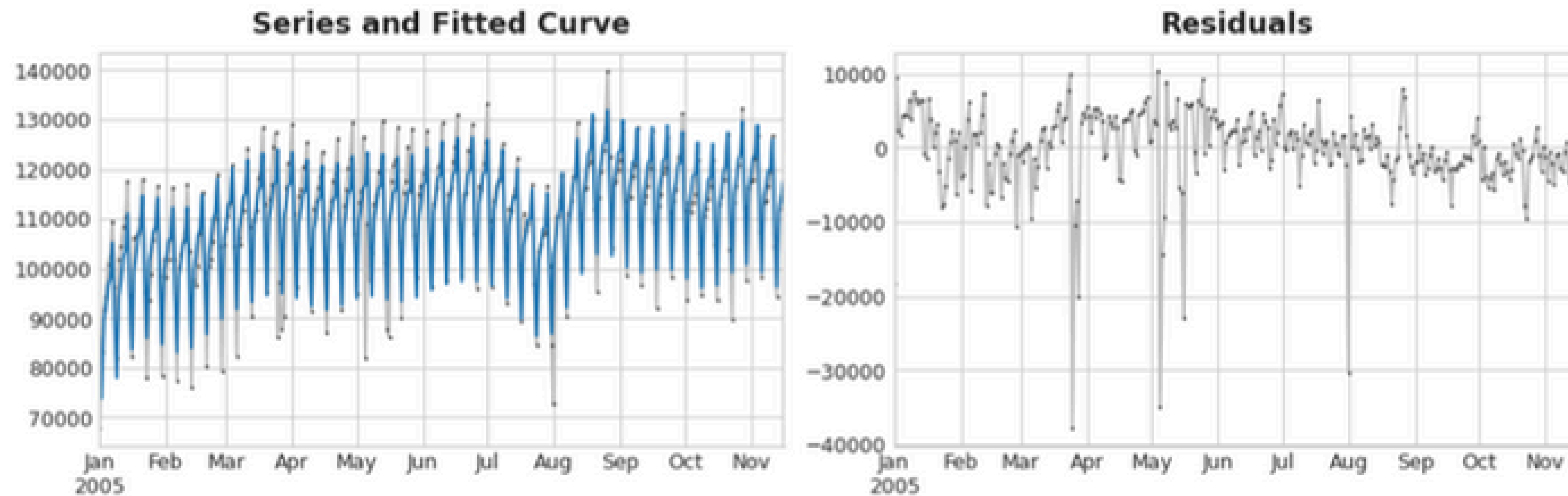
**In our data set: Not many products showed cycles**

# ML- Methodology

- Many time series can be closely described by an additive model of just these three components plus some essentially unpredictable, entirely random *error*:  
$$\text{series} = \text{trend} + \text{seasons} + \text{cycles} + \text{error}$$
- Once we found these features we used linear regression to assign weights and find the accurate series and then predict that series.

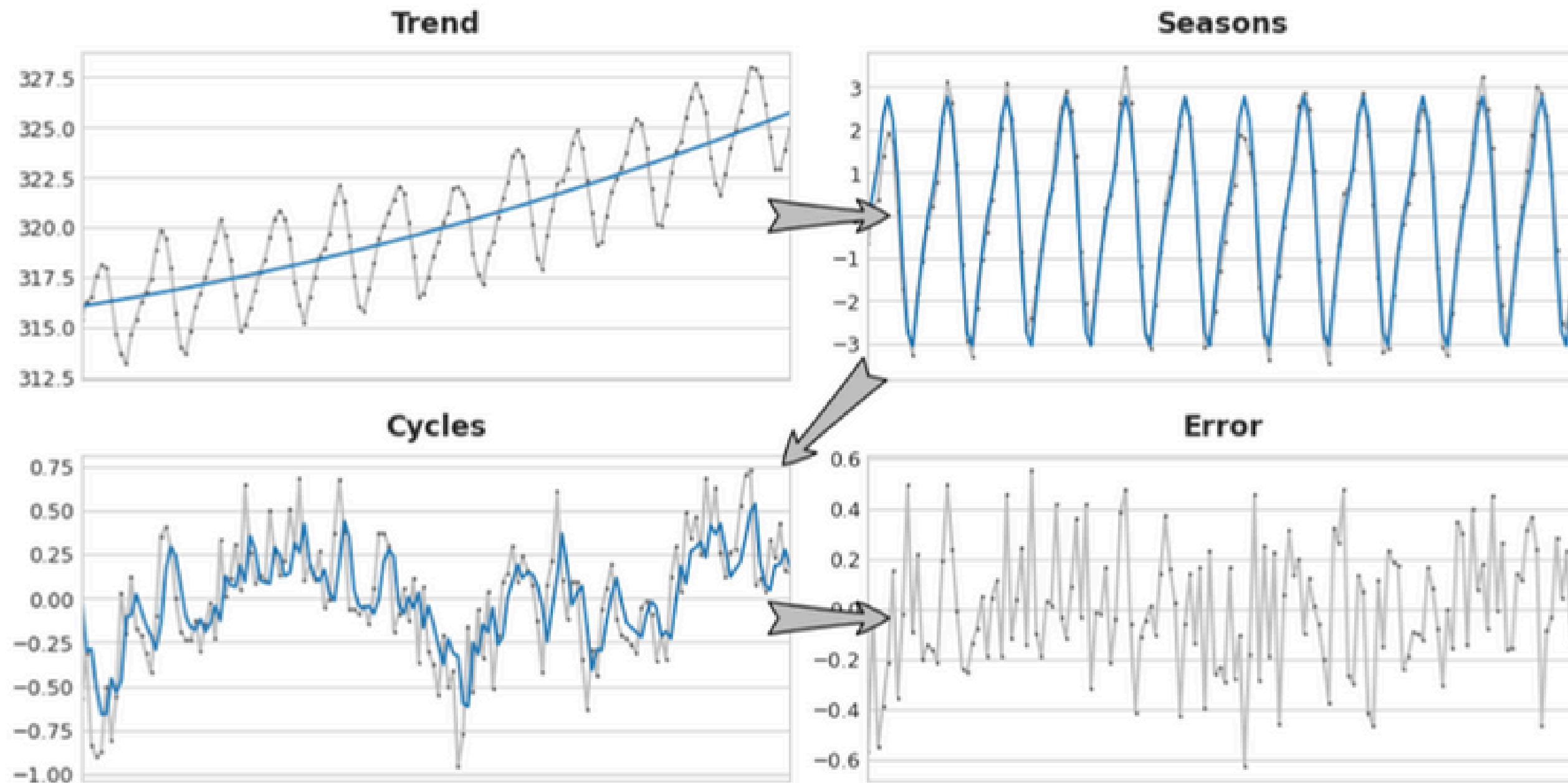


**Residuals are the difference between series and fitted curve.  
Aim is to minimise the residual which we do through regression techniques.**



*The difference between the target series and the predictions (blue) gives the series of residuals.*

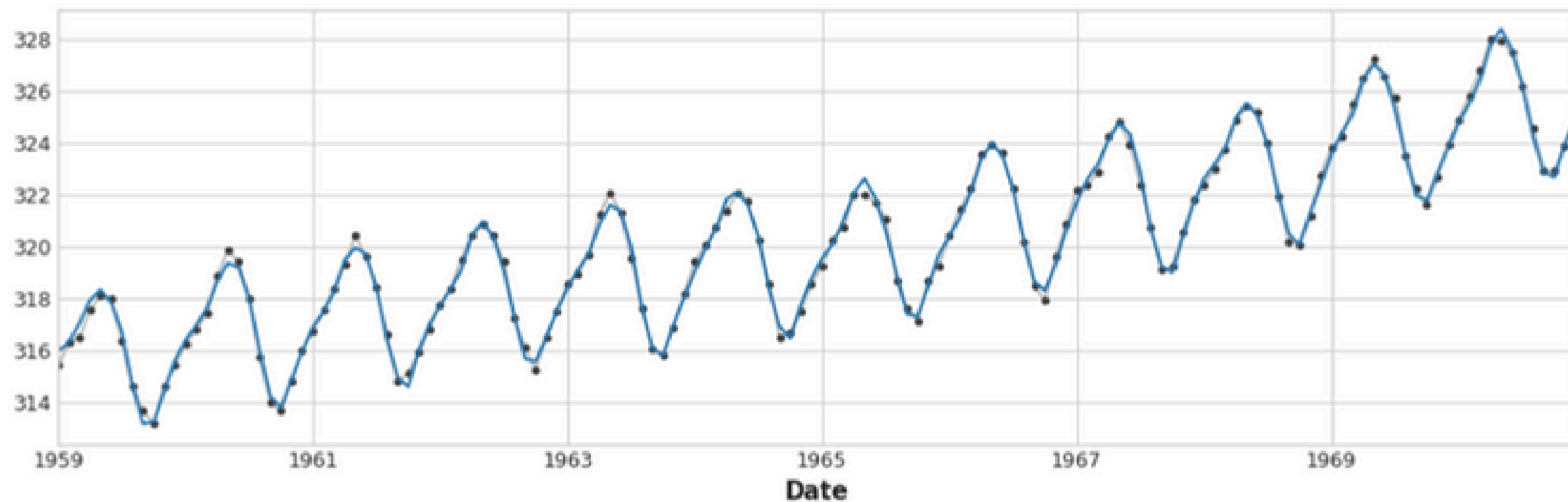
**We could imagine learning the components of a time series as an iterative process.**



**We can use one algorithm to do all this. But some algorithms are good at learning something. This is the idea of hybrid algorithm.**

**Use one algorithm for some of the components and another algorithm for the rest.**

**This way we can always choose the best algorithm for each component.**



*Add the learned components to get a complete model.*

**To do this, we use one algorithm to fit the original series and then the second algorithm to fit the residual series. Combine the two to get the complete model.**





## In detail, the process is this:

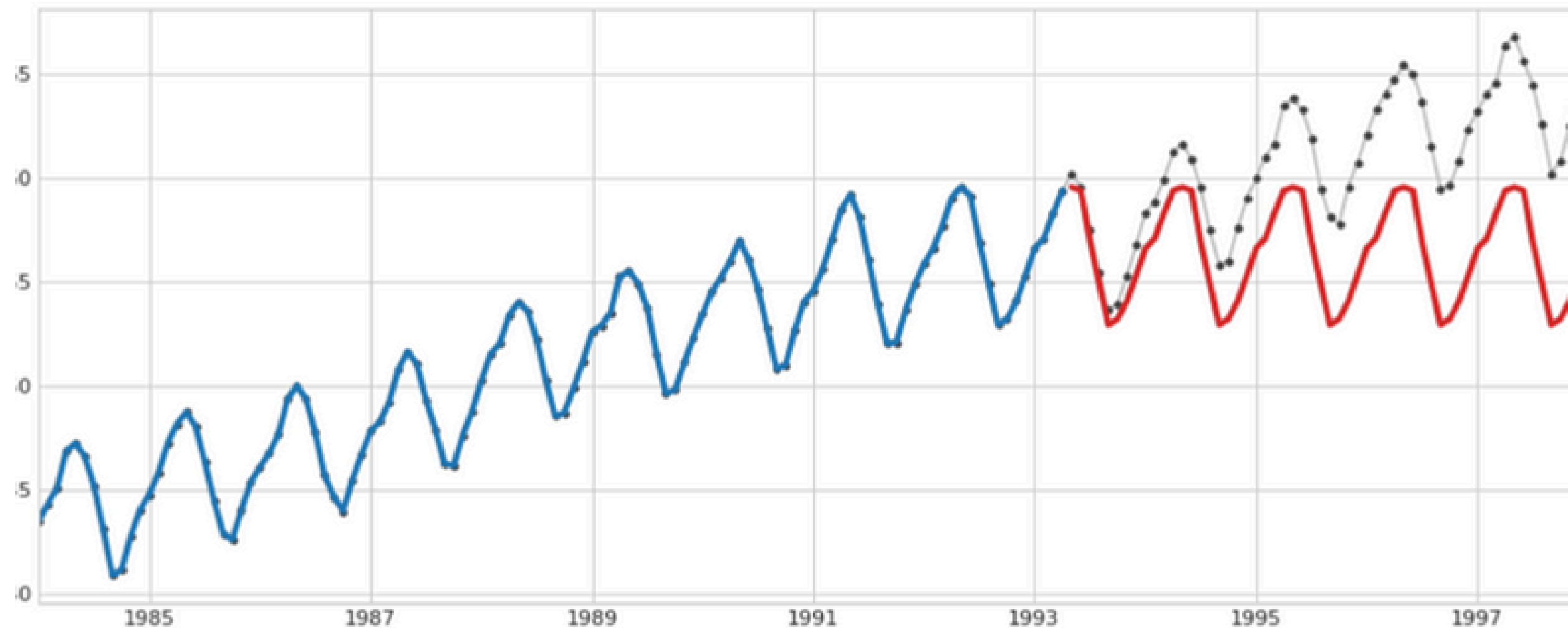
```
# 1. Train and predict with first model
model_1.fit(X_train_1, y_train)
y_pred_1 = model_1.predict(X_train)

# 2. Train and predict with second model on residuals
model_2.fit(X_train_2, y_train - y_pred_1)
y_pred_2 = model_2.predict(X_train_2)

# 3. Add to get overall predictions
y_pred = y_pred_1 + y_pred_2
```

**The most common strategy for constructing hybrids is combining a simple (usually linear) learning algorithm followed by a complex, non-linear learner like XGBoost, MLP Regressor. So we used a hybrid of :**

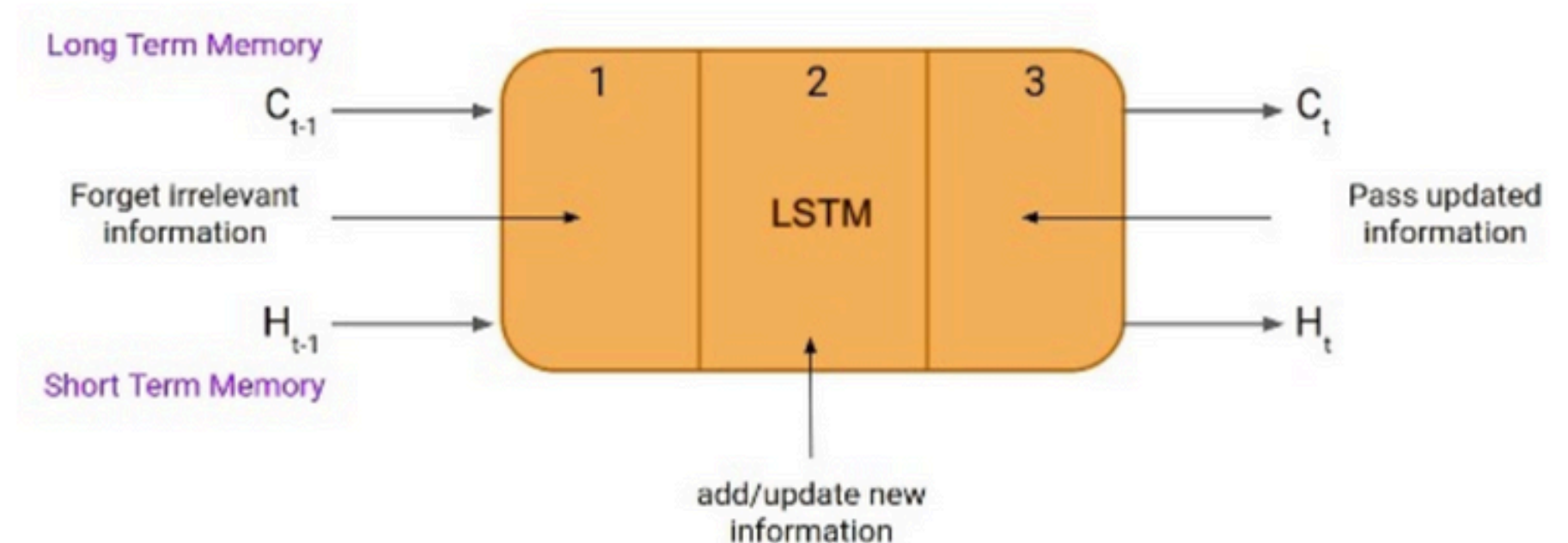
**Linear Regression and XGBoost  
MLP Regressor and XGBoost**



**Hybrid models can extrapolate target values beyond the training set given appropriate features as inputs and can lead to overfitting.**

# LSTM

- Utilized data from the previous 50 days.
- Incorporated lag features "on\_promotions" and "dcoilwtico".
- Employed two stacked LSTM layers.
- Integrated dropout layers after each LSTM layer.
- Dropout is a regularization method to counter overfitting by randomly nullifying a fraction of input units during training.
- Utilized a dropout rate of 0.2, meaning 20% of inputs are randomly zeroed during training.



## **Challenges faced.**

- **Complexity in the data set: About 1800 time series data for products sold at different stores.**
- **Passing these 1800 time series data and creating lags and steps was the hurdle.**
- **Computational complexity is too high for deep learning algorithms like LSTM to predict sales of all the product on our laptops.**
- **Used algorithms like linear regression capable of multi-output regression to predict the output of different products.**

# Evaluation

The evaluation metric for this competition is **Root Mean Squared Logarithmic Error**.

The **RMSLE** is calculated as:

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (\log(1 + \hat{y}_i) - \log(1 + y_i))^2}$$

where:

- $n$  is the total number of instances,
- $\hat{y}_i$  is the predicted value of the target for instance (i),
- $y_i$  is the actual value of the target for instance (i), and,
- $\log$  is the natural logarithm.

# *Validation RMSLE (Root Mean Squared Logarithmic Error)*

## **Linear regression**

```
Root Mean Squared Logarithmic Error (RMSLE) for kaggle data set : 0.5488001495051799
```


## **Linear regression and XG boost (hybrid)**

```
y_resid = y_resid.stack().squeeze() # wide to long  
Root Mean Squared Logarithmic Error (RMSLE): 0.015034277278826332
```


## **MLP regression and XG boost (hybrid)**

```
Root Mean Squared Logarithmic Error (RMSLE): 0.018003350764466144
```


# Test $RMSLE$ (Root Mean Squared Logarithmic Error)

	<b>submission.csv</b> Complete · now	<b>0.75677</b>
----------------------------------------------------------------------------------	-----------------------------------------	----------------

**LSTM**

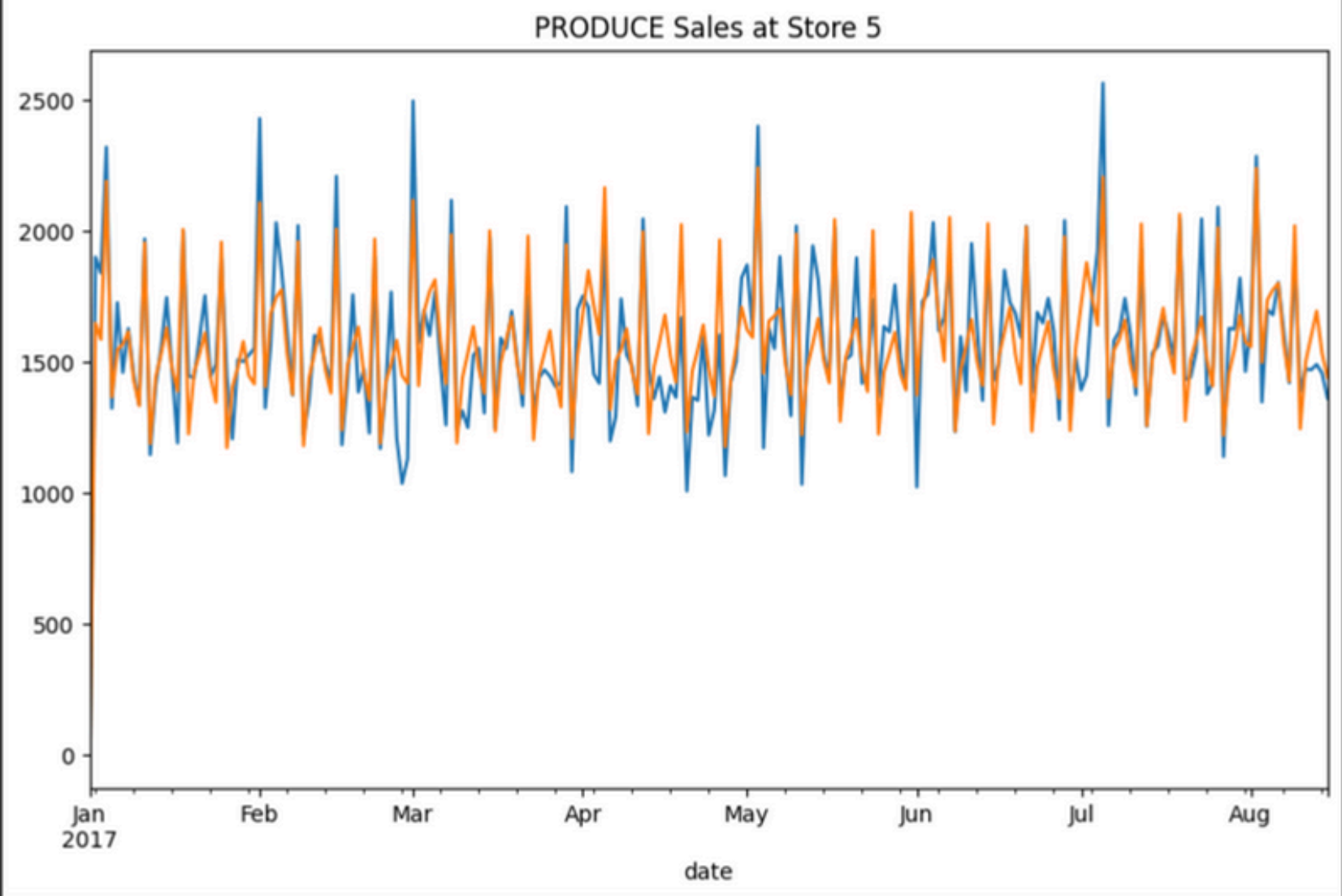
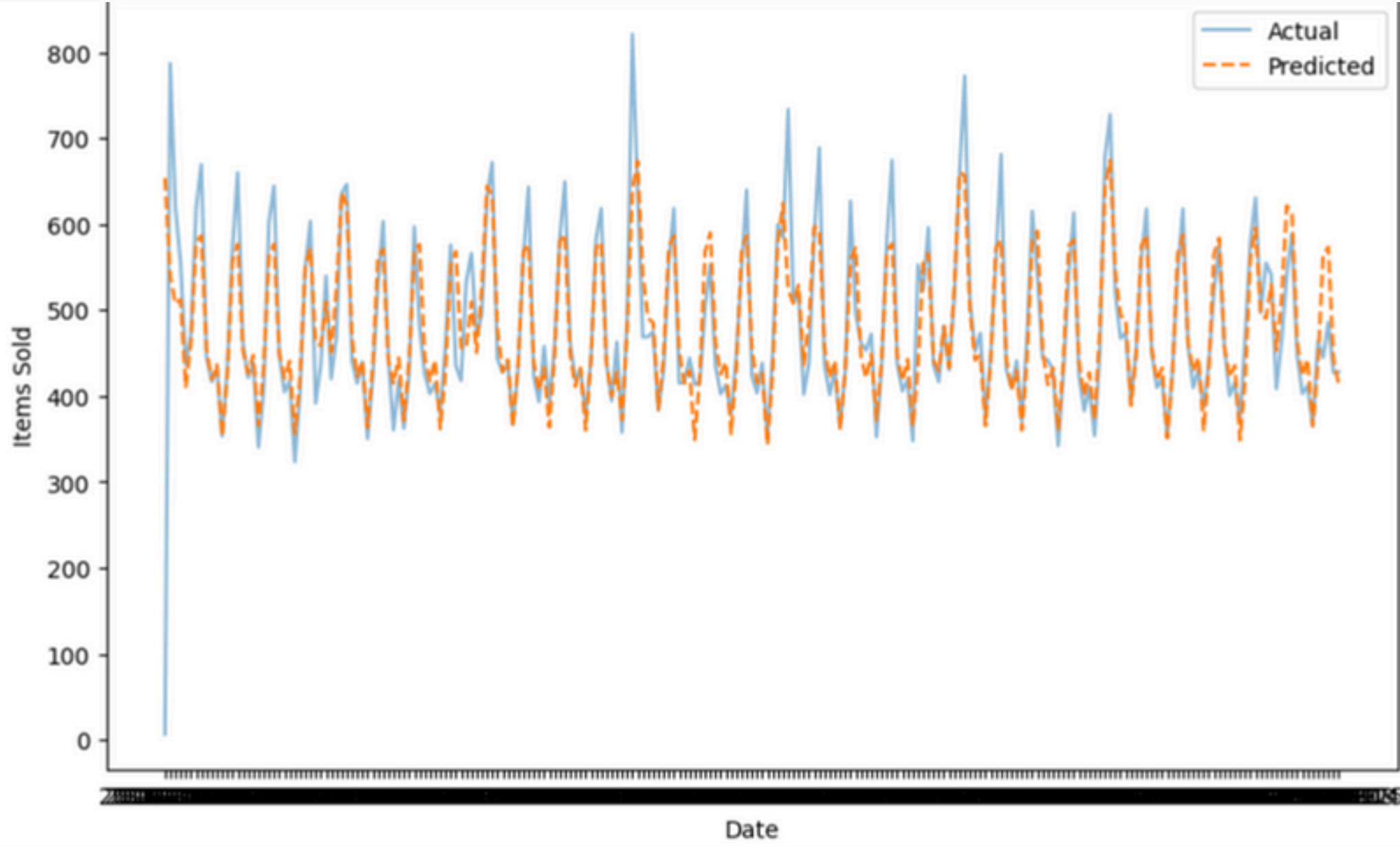
	<b>submission.csv</b> Complete · 7h ago	<b>0.51582</b>
-------------------------------------------------------------------------------------	--------------------------------------------	----------------

**Linear regression and  
XG boost (hybrid)**

	<b>submission.csv</b> Complete · 2d ago	<b>0.51090</b>
-------------------------------------------------------------------------------------	--------------------------------------------	----------------

**Linear Regression.**

# Prerdiction by linear Regression model.





# Uncle Tonnies: Sales prediction

## Sales data

April sales data.

984

No of Data instances

## 3 Features

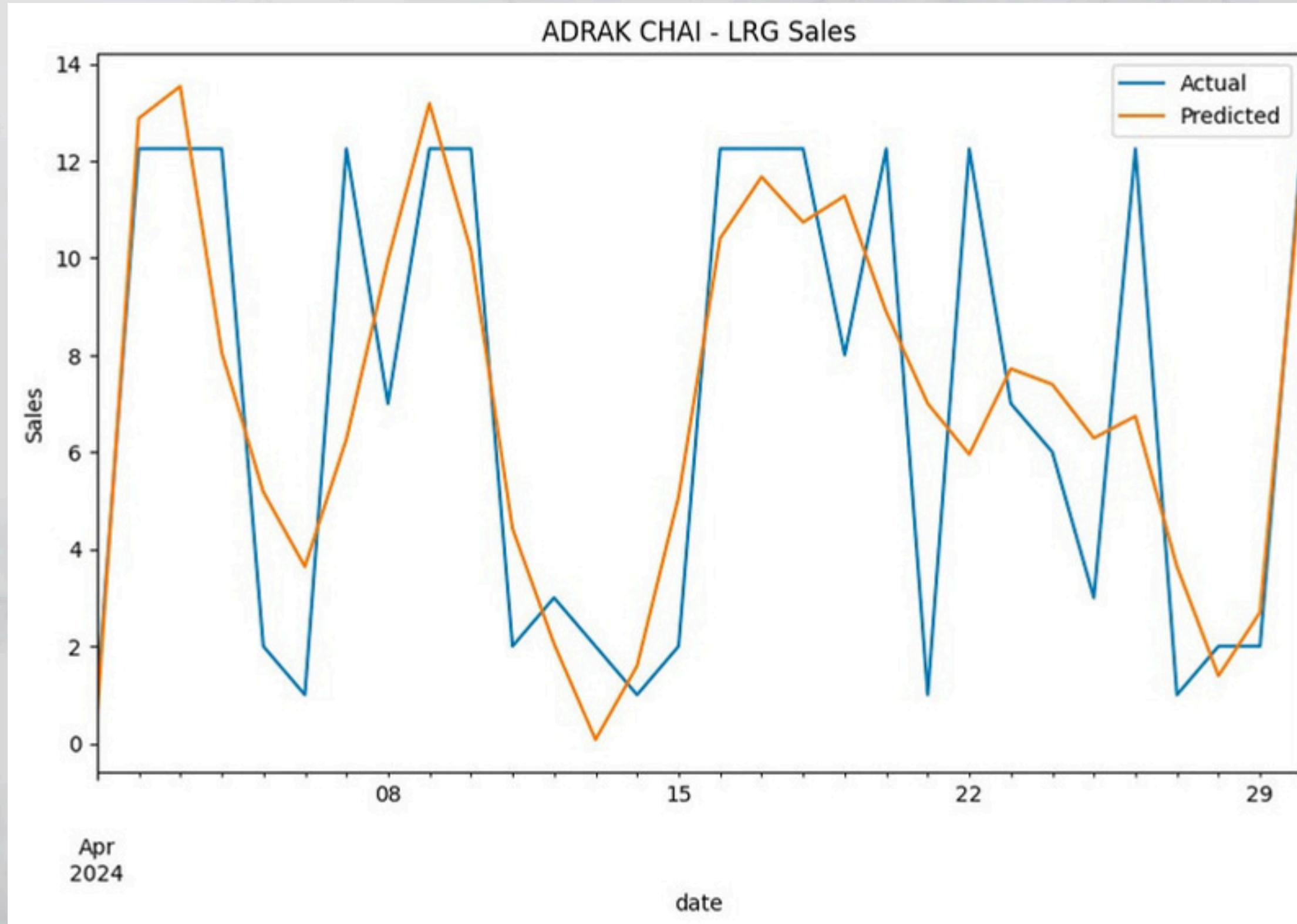
Item\_name, Qty, Price

## Sales prediction

for the next 15 days.



Root Mean Squared Logarithmic Error (RMSLE) for real world data : 0.5173962411934679



**Predicted curve, fitting on the actual curve of sales of Adrak chai**

## References:

- Sales Forecasting for Retail Chains Ankur Jain<sup>1</sup> , Manghat Nitish Menon<sup>2</sup> , Saurabh Chandra, @eng.ucsd.edu
- Agrawal, D.; Schorling, C. 1997. Market share forecasting: an empirical comparison of artificial neural networks and multinomial logit model, *Journal of Retailing* 72(4): 383–407. [https://doi.org/10.1016/S0022-4359\(96\)90020-2](https://doi.org/10.1016/S0022-4359(96)90020-2)
- Ahmed, N. K.; Atiya, A. F.; Gayar, N. E.; El-Shishiny, H. 2010. An empirical comparison of machine learning models for time series forecasting, *Econometric Reviews* 29(5–6): 594–621.
- Supermarket sales Prediction using regression. (2021). *International Journal of Advanced Trends in Computer Science and Engineering*, 10(2), 1153–1157. <https://doi.org/10.30534/ijatcse/2021/951022021>
- Elmasdotter, A., & Nyströmer, C. (2018). A comparative study between LSTM and ARIMA for sales forecasting in retail (Dissertation). Retrieved from <https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-229747>
- Canova, F.; Hansen, B. E. 1995. Are seasonal patterns constant over time? A test for seasonal stability, *Journal of Business & Economic Statistics* 13(3): 237–252.
- Box, G. E. P.; Jenkins, G. 1970. *Time series analysis, forecasting and control*. San Francisco: Holden-Day, CA.



Thank You

