A blue and white logo

Description automatically generated  
UNIVERSITY OF HERTFORDSHIRE  
School of Physics, Engineering and Computer Science

MSc Data Science and Analytics Masters Project 7COM1075-0109-2023

August 2024

Enhancing Retail Sales Predictions with LightGBM

Name: Sahil Hamid Mulani

Student ID: 21078559

Supervisor: Phillip Gainley

# Abstract

Accurate demand forecasting is crucial for effective inventory and supply chain management. This study explores the application of feature engineering on LightGBM, a gradient boosting algorithm, for sales forecasting using Walmart’s M5 competition dataset.

The dataset features hierarchical sales data categorised by products, stores, departments, and geographic regions, posing substantial challenges due to its complexity.

The research began with Exploratory Data Analysis to understand sales patterns across different regions and products. It revealed significant regional variations and intermittent sales behavior. The initial forecasting model based on recursive LightGBM was applied to a specific product from the Hobbies category where challenges with intermittent sales and peak detection were observed. Subsequent iterations incorporated advanced feature engineering, including lagged features, rolling window statistics, and calendar-based variables. Despite these improvements, limitations were noted, particularly with capturing relationships in highly intermittent data.

In chapter 3 and 4, a continuous time series was selected from the food category aimed at forecasting 226 days and 28 days ahead. Incremental training and refitting strategies were used, resulting in a minimum mean squared error during iteration 1 of chapter 3 and iteration 1 of chapter 4. This highlights LightGBM’s efficacy in managing time series data and demonstrated the importance of iterative refinement and feature engineering in enhancing forecast accuracy.

Additionally, in Chapter 5, all the products for a particular store were forecasted and added in a hierarchical manner to mimic the real-world problem. This resulted in poor forecasts at the store level during the first iteration but led to considerable improvements in performance after incorporating features into the model in the second iteration. Further work needs to be carried out in this domain.

The study underscores LightGBM’s potential for complex forecasting tasks and provides insights into optimizing feature selection, feature engineering and model parameters for single stock keeping unit predictions. Additionally, it also highlights the effectiveness of the skforecast library to generate forecasts with relative ease.

# Acknowledgements

I would like to express my sincere gratitude to all those who have contributed to the successful completion of this research project.

First and foremost, I extend my deepest thanks to my research advisor, Phillip Gainley, whose guidance, support, and expertise were invaluable throughout this study. His guidance on how to approach a MSc project in various areas such as literature review, structuring and references helped in providing the foundations which accelerated this project.

I am also grateful to the University of Hertfordshire for providing the resources and facilities necessary for this research. The access to computational resources was essential for conducting the experiments and analyses required for this study.

I would also like to acknowledge the M5 Competition organizers for making their dataset publicly available. Their contribution to the field of time series forecasting has provided a valuable foundation for this research.

Additionally, I appreciate the support and encouragement of my family and friends. Their patience and understanding throughout the research process have been a great source of motivation.

Thank you all for your unwavering support and contributions.

# MSc Final Project Declaration

This report is submitted in partial fulfilment of the requirement for the degree of Master of Science in Data Science and Analytics Masters Project at the University of Hertfordshire (UH).

It is my own work except where indicated in the report.

I did not use human participants in my MSc Project.

I hereby withhold permission for the report to be made available on the university website provided the source is acknowledged.

Table of Contents

[Abstract 2](#_Toc174825428)

[Acknowledgements 3](#_Toc174825429)

[MSc Final Project Declaration 4](#_Toc174825430)

[Table of Figures 6](#_Toc174825431)

[Introduction 9](#_Toc174825432)

[Specification 12](#_Toc174825433)

[Literature Review 13](#_Toc174825434)

[Dataset 19](#_Toc174825435)

[Chapter 1: Exploratory Data Analysis: 22](#_Toc174825436)

[Chapter 2: Light Gradient boosting machine on HOBBIES\_1\_018\_CA\_1\_validation 29](#_Toc174825437)

[Iteration 1: 29](#_Toc174825438)

[Iteration 2: 32](#_Toc174825439)

[Chapter 3: Light Gradient Boosting Machine on FOODS\_3\_555\_CA\_1\_validation for predicting the next 7.5 months 35](#_Toc174825440)

[Iteration 1: 35](#_Toc174825441)

[Iteration 2: 37](#_Toc174825442)

[Iteration 3: 38](#_Toc174825443)

[Iteration 4: 39](#_Toc174825444)

[Iteration 5: 40](#_Toc174825445)

[Chapter 4: Light Gradient Boosting Machine on FOODS\_3\_555\_CA\_1\_validation for predicting the next 28 days 44](#_Toc174825446)

[Iteration 1: 44](#_Toc174825447)

[Iteration 2: 45](#_Toc174825448)

[Iteration 3: 46](#_Toc174825449)

[Chapter 5: Forecasting sales across various levels of hierarchies 48](#_Toc174825450)

[Iteration 1: 48](#_Toc174825451)

[Iteration 2: 54](#_Toc174825452)

[Results 62](#_Toc174825453)

[Conclusion 65](#_Toc174825454)

[References 67](#_Toc174825455)

[Appendix 70](#_Toc174825456)

# Table of Figures

[Figure 1: Illustration of the time series of an item (Fig. 1 from Tian, Wang, & E (2021) 14](#_Toc174729779)

[Figure 2: Illustration of how recursive forecasting technique is applied in skforecast 17](#_Toc174729780)

[Figure 3: Illustration of the hierarchical nature of the Walmart data (Fig. 1 from Tian, Wang, & E (2021) 19](#_Toc174729781)

[Figure 4: Breakdown of the number of series as per the aggregation level by (Hallam, J) 19](#_Toc174729782)

[Figure 5: Illustration of the sales for FOODS\_1\_001 across California and Texas regions 22](#_Toc174729783)

[Figure 6: Illustration of sales for FOODS\_1\_001 around Christmas for California and Texas regions 23](#_Toc174729784)

[Figure 7: Illustration of the sales for Wisconsin and Texas region for product FOODS\_1\_001 24](#_Toc174729785)

[Figure 8: Illustration of the Christmas period sales for Wisconsin and Texas region for product FOODS\_1\_001 24](#_Toc174729786)

[Figure 9: Illustration of the sales for California and Texas region for product FOODS\_1\_001 25](#_Toc174729787)

[Figure 10: Illustration of the Christmas period sales for California and Texas region for product FOODS\_1\_001 26](#_Toc174729788)

[Figure 11: Illustration of the latest year’s sales pattern of FOODS\_3\_555 product in a specific store in California 27](#_Toc174729789)

[Figure 12: Illustration of latest year’s sales pattern of FOODS\_3\_555 product in a specific store in Texas 27](#_Toc174729790)

[Figure 13: Illustration of latest year’s sales pattern of FOODS\_3\_555 product in a specific store in Wisconsin 28](#_Toc174729791)

[Figure 14: Illustration of the intermittent nature of the HOBBIES\_1\_018\_CA\_1\_validation sales data where x-axis denotes the number of days 30](#_Toc174729792)

[Figure 15: Illustration of the actual and predicted values for HOBBIES\_1\_018\_CA\_1\_validation product 32](#_Toc174729793)

[Figure 16: Illustration of the actual and predicted values for HOBBIES\_1\_018\_CA\_1\_validation product 33](#_Toc174729794)

[Figure 17:Illustration of the actual and predicted values for HOBBIES\_1\_018\_CA\_1\_validation product 34](#_Toc174729795)

[Figure 18: Illustration of the movement of the time-series for product FOODS\_3\_555\_CA\_1\_validation for the first 50 days 35](#_Toc174729796)

[Figure 19: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation 36](#_Toc174729797)

[Figure 20: Illustration of the autocorrelation of FOODS\_3\_555\_CA\_1\_validation sales feature up to 30 lags 37](#_Toc174729798)

[Figure 21: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation 38](#_Toc174729799)

[Figure 22: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation 39](#_Toc174729800)

[Figure 23: Illustration of the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation 40](#_Toc174729801)

[Figure 24: Illustration of the partial autocorrelation on FOODS\_3\_555\_CA\_1\_validation sales feature 41](#_Toc174729802)

[Figure 25: Illustration of the partial autocorrelation on FOODS\_3\_555\_CA\_1\_validation sales feature after de-trending 42](#_Toc174729803)

[Figure 26: Illustration of the partial autocorrelation on FOODS\_3\_555\_CA\_1\_validation sales feature after de-trending and de-seasonalising 42](#_Toc174729804)

[Figure 27: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation 43](#_Toc174729805)

[Figure 28: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation 45](#_Toc174729806)

[Figure 29: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation 46](#_Toc174729807)

[Figure 30: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation 47](#_Toc174729808)

[Figure 31: Figure illustrating different error metrics across departments, category and store levels 48](#_Toc174729809)

[Figure 32: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_1 department 49](#_Toc174729810)

[Figure 33: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_2 department 49](#_Toc174729811)

[Figure 34: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_3 department 50](#_Toc174729812)

[Figure 35: Figure illustrating the difference between the actual sales and predicted sales for HOBBIES\_1 department 50](#_Toc174729813)

[Figure 36: Figure illustrating the difference between the actual sales and predicted sales for HOBBIES\_2 department 51](#_Toc174729814)

[Figure 37: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD\_1 department 51](#_Toc174729815)

[Figure 38: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD\_2 department 52](#_Toc174729816)

[Figure 39: Figure illustrating the difference between the actual sales and predicted sales for FOODS Category 52](#_Toc174729817)

[Figure 40: Illustrating the difference between the actual sales and predicted sales for HOBBIES Category 53](#_Toc174729818)

[Figure 41: Illustrating the difference between the actual sales and predicted sales for Household Category 53](#_Toc174729819)

[Figure 42: Figure illustrating different error metrics across departments, category and store levels 54](#_Toc174729820)

[Figure 43: Figure illustrating different error metrics across departments, category and store levels 55](#_Toc174729821)

[Figure 44: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_1 department 56](#_Toc174729822)

[Figure 45: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_2 department 56](#_Toc174729823)

[Figure 46: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_3 department 57](#_Toc174729824)

[Figure 47: Figure illustrating the difference between the actual sales and predicted sales for HOOBIES\_1 department 57](#_Toc174729825)

[Figure 48: Figure illustrating the difference between the actual sales and predicted sales for HOOBIES\_2 department 58](#_Toc174729826)

[Figure 49: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD\_1 department 58](#_Toc174729827)

[Figure 50: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD\_2 department 59](#_Toc174729828)

[Figure 51: Figure illustrating the difference between the actual sales and predicted sales for FOODS category 59](#_Toc174729829)

[Figure 52: Figure illustrating the difference between the actual sales and predicted sales for HOOBIES category 60](#_Toc174729830)

[Figure 53: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD category 60](#_Toc174729831)

[Figure 54: Figure illustrating the difference between the actual sales and predicted sales for Store CA\_1 based in California 61](#_Toc174729832)

[Figure 55: Figure illustrating different error metrics across departments, category and store levels for Chapter 5, Iteration 1 63](#_Toc174729833)

[Figure 56: Figure illustrating different error metrics across departments, category and store levels for Chapter 5, Iteration 2 63](#_Toc174729834)

# Introduction

The theory of forecasting underpins that current and past knowledge can be leveraged to estimate future time points. Furthermore, identifying patterns in past data and utilising those patterns in the process of prediction can be fruitful. Predicting future values with exact accuracy is not expected rather options such as an expected value, a prediction interval, a percentile, and an entire prediction distribution can be collectively said to be called a ‘forecast.’ A forecasting method is defined as a certain sequence of steps taken that leads to the production of future time points (Petropoulos et al., 2022).

Time series forecasting has traditionally been a fundamental tool in industries such as finance, healthcare, energy, and retail (Box et al., 2015). Accurate sales forecasting is a critical component of effective supply chain management, particularly in the retail sector, where it significantly impacts inventory control, pricing strategies, and overall financial performance (Chopra and Meindl, 2016).

The retail industry must deal with significant fluctuations in the demand and inaccurate forecasts can further amplify the situation which can result in lost revenue to the business (Fildes, Goodwin, and Lawrence, 2006). Inaccurate forecasts can lead to overstocking, resulting in increased holding costs or stockouts, which can result in lost sales and customer dissatisfaction (Fildes, Goodwin, and Lawrence, 2006). Thus, improvements in forecasting accuracy can significantly enhance the businesses processes and improve profitability.

The significant rise of machine learning techniques has created a huge impact in time series forecasting providing more precise predictions which are vital for operation success of the business (Makridakis, Spiliotis, and Assimakopoulos, 2018). Machine learning models, particularly ensemble methods like gradient boosting, have emerged as robust alternatives (Dietterich, 2000). LightGBM, developed by Microsoft, is one such model that has gained widespread popularity for its speed and accuracy (Ke et al., 2017). It builds upon traditional gradient boosting by incorporating novel techniques such as Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which make it an ideal candidate for large datasets with high dimensionality (Ke et al., 2017). These features allow LightGBM to outperform other models like XGBoost in many scenarios, particularly in competitions such as the M5 Forecasting Competition, where participants predict the sales of thousands of products over an extended horizon (Makridakis et al., 2020).

Despite advances in machine learning, significant challenges remain in time series forecasting. One major issue is dealing with intermittent or sparse data, which is common in retail sales, where many products do not sell consistently, leading to numerous zero-sales days (Syntetos, Boylan, and Croston, 2005). Traditional forecasting methods often perform poorly with such data as they struggle to model the underlying distribution of sales (Zhang, 2003). Moreover, the scalability of models to manage increasingly large datasets—both in terms of features and observations—is an ongoing concern (Hastie, Tibshirani, and Friedman, 2009). LightGBM addresses these challenges by being computationally efficient and capable of handling large datasets with numerous features (Ke et al., 2017). However, its implementation demands careful preprocessing, feature engineering, and hyperparameter tuning to achieve optimal performance (Prokhorenkova et al., 2018).

Another pressing issue is the interpretability of machine learning models. While models like LightGBM can produce highly accurate forecasts, understanding how they arrive at their predictions can be challenging (Rudin, 2019). This lack of transparency can be problematic, particularly when forecasts are used to guide critical business decisions. In the retail context, understanding the drivers of demand is as important as the forecast itself, highlighting the need for interpretable models in this project (Doshi-Velez and Kim, 2017).

The application of machine learning models in sales forecasting raises several legal, social, and ethical concerns. One significant issue is data privacy. Retailers often have access to vast amounts of customer data, which can be used to enhance sales forecasts. However, the use of such data must comply with legal frameworks like the General Data Protection Regulation (GDPR) in Europe, which sets strict guidelines on data collection, processing, and storage (Voigt and Bussche, 2017). Ethical considerations also emerge, particularly in avoiding biases that can arise from historical data, which may lead to unfair or discriminatory practices if not carefully managed (Barocas, Hardt, and Narayanan, 2019).

On the professional front, the growing reliance on machine learning models in decision-making processes raises questions about accountability and transparency (Mittelstadt et al., 2016). It is crucial that businesses implementing these models understand their benefits and limitations. Ensuring that models are interpretable and that their predictions can be explained to stakeholders is essential in maintaining trust and upholding ethical standards (Lipton, 2018).

Accurate sales forecasting has a profound economic impact, particularly in the retail sector. Poor forecasts can lead to overstocking or stockouts, both of which have direct financial consequences (Chen, Yu, and Shih, 2020). Overstocking ties up capital and increases storage costs, while stockouts result in lost sales and damaged customer relationships. By enhancing the accuracy of sales forecasts, businesses can optimize inventory levels, reduce costs, and increase profitability (Chase, 2013).

However, implementing advanced machine learning models involves commercial risks. These models require substantial computational resources for training and ongoing operation, which can be costly (Goodfellow, Bengio, and Courville, 2016). Additionally, there is a risk that the models may not perform as expected when deployed in real-world settings, particularly if the training data does not fully capture the complexities of the business environment (Dietterich, 1995). To mitigate these risks, it is essential to implement robust validation and backtesting processes and continuously monitor the model’s performance, making necessary adjustments (Arlot and Celisse, 2010).

Moreover, there is the risk of overfitting, where the model becomes overly tailored to the training data and performs poorly on unseen data. This risk is particularly high in time series forecasting, where future events can be unpredictable and driven by factors not present in the historical data (Yao et al., 2017). Careful model tuning and regular updates with new data can help mitigate this risk.

This project addresses a critical challenge in the retail sector: improving the accuracy of sales forecasts using advanced machine learning techniques. By focusing on LightGBM, a state-of-the-art model that has demonstrated superior performance in large-scale forecasting tasks, this research aims to contribute valuable insights to the field of time series forecasting. The project is grounded in the realities of the modern retail environment, where accurate forecasting is essential for operational efficiency and financial success. It also considers the broader legal, ethical, and commercial risks associated with the use of machine learning models, ensuring that the solutions developed are not only effective but also responsible and sustainable.

This report is structured to provide a comprehensive and systematic overview of the research project, progressing from foundational exploration to a conclusive evaluation. The following components are included:

* **Introduction:** The introduction describes sales forecasting using machine learning and time series analysis, focusing on the LightGBM model. It addresses challenges like model interpretability and data privacy, emphasizing the need for ethical standards. It highlights the commercial impact of forecasting accuracy and gives a general overview of the project domain.
* **Specification:** The specification section describes the scope of the project, the aim, and objectives the research is trying to achieve along with the research question that is trying to be answered.
* **Literature Review:** Offers a critical analysis of existing research and methodologies relevant to sales forecasting and machine learning, with a particular focus on the theoretical underpinnings and practical applications of LightGBM.
* **Methodology:** Provides a detailed account of data preparation, exploratory data analysis, feature engineering, and the implementation of the LightGBM model. The evaluation methods employed to assess model performance are also outlined.
* **Results:** Presents the outcomes of the model evaluation, supported by quantitative metrics, comparative analyses, and visual representations.
* **Conclusion:** Summarizes the key contributions of the research, reflecting on the project's achievements and their significance within the field of sales forecasting.
* **References:** Provides a comprehensive list of cited sources adhering to the Harvard referencing style.
* **Appendices:** Includes supplementary materials that support the main text, such as detailed data visualizations, code excerpts, and extended results.

This structured approach guides the reader through the research journey, from understanding the problem to evaluating the solutions, ensuring a clear and coherent presentation of the work undertaken.

# Specification

This research project focuses on improving sales forecasting accuracy using the M5 competition dataset, which includes sales data for over three thousand products from various retail sectors. The dataset’s complexity arises from its enormous size, intermittent sales patterns, and the influence of various external factors such as promotions, holidays, and store-specific attributes. The primary goal is to develop a robust forecasting model using Light Gradient Boosting Machine (LightGBM), a state-of-the-art algorithm known for its efficiency and effectiveness in handling large datasets.

The research aimed to create a real-world machine learning model development process suitable for time series forecasting which undertook different steps such as loading the dataset, data-preprocessing, feature engineering, model development and model evaluation. The process includes experimenting with feature engineering where various date time, lags and statistical features were incorporated into to the model to assess whether adding features added value to the forecast results or diminished it. Furthermore, during the model initialisation process, different combination of model parameters were tested to asses which combinations contribute to optimal forecast performance. The model evaluation process utilised error metrics such as mean-squared error to understand the difference between the actual and predicted value.

The research question was formulated as: **‘How the LightGBM model development process of loading the data, data-preprocessing, feature engineering, model development and evaluation be enhanced to improve time-series forecasting performance?’**

The objectives which this research wanted to achieve were:

1. Performing exploratory data analysis
2. Developing a real-world machine learning model development process suitable for time series forecasting which included the following processes:

* Loading the dataset
* Data-preprocessing
* Feature engineering
* Model development
* Model Evaluation

1. Experimenting with different combinations of feature engineering techniques and hyperparameter tuning.

**Novelty**: The uniqueness of the research lies in experimenting with various combinations of feature engineering, hyperparameter tuning and model evaluation processes. Additionally, this research through exploratory data analysis aims to understand the behaviours of sales patterns for a particular product across different regions and stores to understand if they have any commonalities or differences between them and to understand if a given product is seasonal during a particular period of time such as Christmas or national holidays.

# Literature Review

The retail industry is considered one of the key drivers of the global economy with millions of enterprises and retailers offering goods and services to billions of customers. It captures a wide range of industries such as food, motor vehicles, apparel, and electronics where transactions occur through distribution channels such as in-store and e-commerce.

In the United States, Walmart, Amazon, and Costco are the top three retail companies in the world where the total retail market has reached a revenue of more than seven trillion U.S. dollars. China, being a dominant market in the consumer goods and retail industry reached a revenue of more than two trillion U.S. dollars, followed by India with a revenue of 1.4 trillion U.S. dollars.

In Europe, The United Kingdom and Germany are the leaders in the retail market with the former generating sales of close to 510 billion pounds and later worth 650 billion euros (Statistica, 2024).

Large retail organisations such as Walmart, Costco, Amazon, and Target work on a business model where they enable the selling of their products as well as those of their competitors (Hassan et al., 2022).

In retail, SKU (Stock keeping unit such as a shampoo of size X) is considered the smallest unit important in operations such as daily stock replenishment and distribution. The number of these SKUs can be in the thousands depending on the retail chain. Nowadays, a typical supermarket or drugstore has tens of thousands of SKU items. Walmart, being the biggest retailer in the world, deals with more than one billion SKU and store combinations. Fashion chains such as Zara maintain SKU items in tens of thousands (Fildes, Ma, & Kolassa, 2022).

Considering the sheer number of unique items, it becomes immensely important for the retailer to predict the demand for their unique products. Successfully predicting the demand results in better inventory management and better distribution, thus minimizing loss and improving sales and customer satisfaction (Jain, Menon, & Chandra, 2015), (Gordon & Berry, 2004).

Furthermore, many external events can affect demand. Some of them are competition, weather, and seasonal trends. In addition to external factors, internal events such as promotions, sales events, and pricing play a significant role in the changing demand (Jain, Menon, & Chandra, 2015).

To prevent customer service issues and high inventory cost supermarkets heavily rely on forecasts to aid in making strategic and tactical decisions as well as demand and supply planning (Fildes, Ma, & Kolassa, 2022).

The above factors make forecasting the demand of retail sales an important tool to add value to the business (Hassan et al., 2022). Furthermore, forecasting for the sake of forecasting has no value. Optimistic forecasts can lead to issues of overstocking leading to extra costs whereas pessimistic forecasts can lead to lost sales because of insufficient stocking. Thus, it is pivotal that our forecast has the highest accuracy.

Demand can be differentiated into four unique categories: intermittent, lumpy, smooth, and erratic (Syntetos, Boylan, & Croston, 2005), (Tian, Wang, & E, 2021). Intermittent demand occurs when there is a high proportion of zero values which is commonly observed in the retail industry.

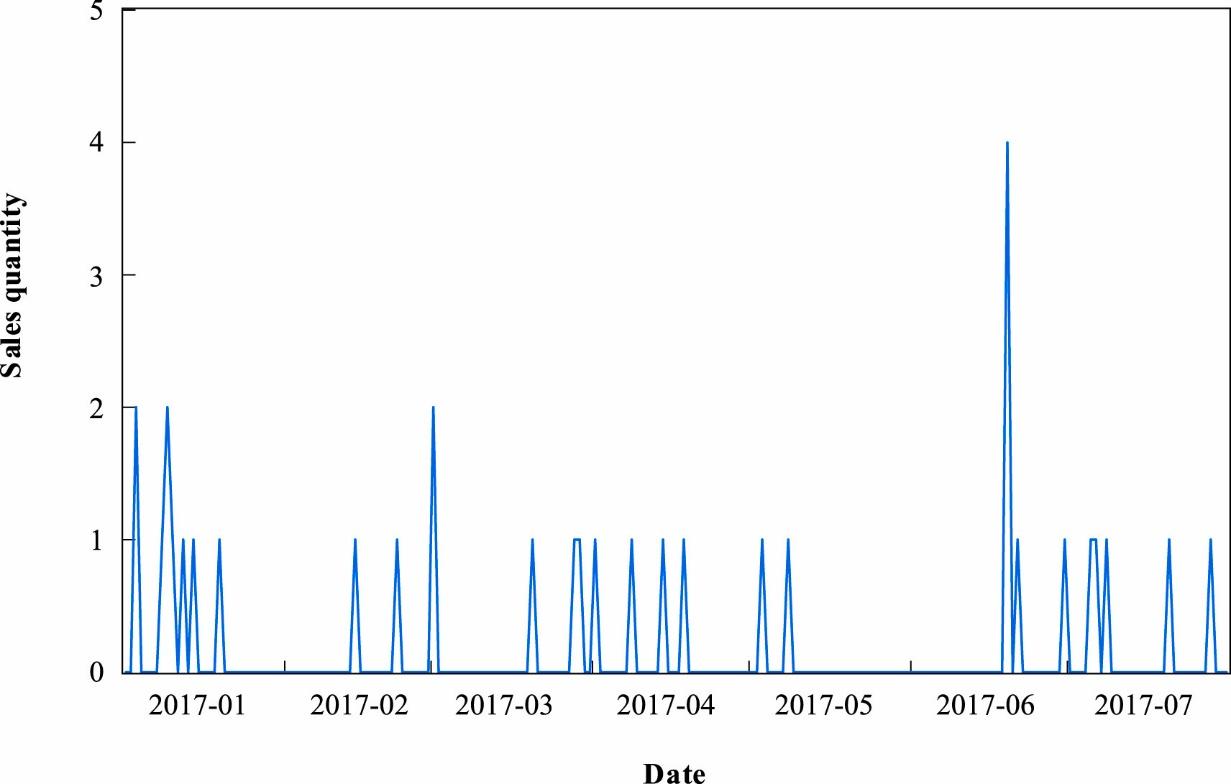


Figure 1: Illustration of the time series of an item (Fig. 1 from Tian, Wang, & E (2021)

The above figure is an example an item displaying intermittent demand from the Tmall.com retail platform. For 87% of the time, the item has no demand which can lead to inconsistencies in the stock levels and delivery center.

Forecasting of intermittent demand is considered as a challenging venture where two important difficulties need to be acknowledged: 1) actual demand is irregular or random and 2) difficult to predict the occurrence or timing of demand (Nikolopoulos, 2021). Thus, it becomes vital to predict the quantity and time of demand occurrence.

Past decades of research have contributed to many models for demand prediction and forecasting that focus on fast-moving time-series, but limited research has been done to forecast intermittent demand. Another reason for the lack of research is the difficulty as forecasting intermittent data is considered challenging compared to the traditional forecasting problem (Nikolopoulos, 2021).

Although contribution to intermittent demand forecasting has been limited, various researchers have developed solutions. The Croston method is the first original work conducted on intermittent demand forecasting which has been practically successful (Croston, 1972). Syntetos and Boylan (2001) found bias in the Croston method and proposed SBA method which deals with the bias and is considered an improvement to the Croston method. Prestwich et al. (2014) proposed an unbiased model which is the combination of Croston method and Bayesian inference. The problem of outdating items in intermittent demand forecasting has also the attracted attention of many scholars (Babai et al., 2019), (Prestwich et al., 2014), (Teunter et al., 2011).

Furthermore, machine learning techniques such as neural networks have been widely used for intermittent demand forecasting. Lolli et al. (2017) utilized feedforward single-hidden layer neural network for different aggregation levels and Kourentzes (2013) proved neural networks to be effective for forecasting intermittent demand.

The M competitions have been organized since the last four decades to improve forecast accuracy by studying and evaluating state of the art forecasting methods ([Makridakis et al., 1982](https://www.sciencedirect.com/science/article/pii/S0169207021001874" \l "b27), [Makridakis et al., 1993](https://www.sciencedirect.com/science/article/pii/S0169207021001874" \l "b30), [Makridakis and Hibon, 2000](https://www.sciencedirect.com/science/article/pii/S0169207021001874" \l "b31), [Makridakis et al., 2020c](https://www.sciencedirect.com/science/article/pii/S0169207021001874" \l "b34).). The results from these competitions have had a significant impact on the practice of forecasting by providing valuable knowledge in improving forecast accuracy ([Hyndman, 2020](https://www.sciencedirect.com/science/article/pii/S0169207021001874" \l "b15)). The first four competitions contributed to understanding the potential of combining, automatic forecasting, merits of simplicity and machine learning methods (Makridakis, Spiliotis, & Assimakopoulos, 2022).

The M5 competition was an extension of the previous ones which focused on retail sales forecasting by utilizing real world sales data that was hierarchically structured with intermittent and unpredictable behavior ([Syntetos and Boylan, 2005](https://www.sciencedirect.com/science/article/pii/S0169207021001874" \l "b54), [Syntetos et al., 2005](https://www.sciencedirect.com/science/article/pii/S0169207021001874" \l "b55)). Key findings obtained from this competition are the superior performance of the machine learning methods, value of combining, cross learning, and explanatory variables More interestingly, among all the previous competitions, M5 competition was the first where all the top performing methods were “pure” ML methods (Makridakis, Spiliotis, & Assimakopoulos, 2022).

Wisdom of the crowd highlights that in many cases the answer to a complex question is more reliable when it is supported by a larger audience than by a single expert. Similarly, aggregating the predictions of a group of models/predictors can often lead to better predictions than accepting the prediction of a single model. The group of models is called an ensemble, and the technique is called Ensemble Learning (Géron, 2019, Chapter 7).

Boosting is a type of Ensemble learning which combines several weak models or learners into a strong learner. The idea revolves around the notion that each model is trying to correct its previous models (Géron, 2019, Chapter 7), (Freund and Schapire, 1997).

AdaBoost works on the basic notion that the new model can improve over the previous ones by focusing on the training points that the previous model could not fit (Freund and Schapire, 1997).

Gradient boosting is another popular boosting algorithm which operates by adding models in a consecutive manner. However, in contrast to AdaBoost which modifies an instance’s weight, gradient boosting is unique in its approach of fitting where it fits the latest model to the errors generated by the previous model (Breiman, 1997), (Friedman, 2001).

Gradient boosting decision tree (GBDT) (Friedman, 2001) is renowned in the machine learning domain for its efficiency, accuracy, and interpretability where it has generated significant results in various applications such as multi-class prediction (Li, 2012), click prediction (Richardson, Dominowska and Ragno, 2007) and rank learning (Burges, 2010).

However, the scale of data has expanded with ever increasing number of features and data points as known as big data. This scale of data has posed difficulties on the Gradient boosting decision tree (GBDT) (Ke et al., 2017). The current implementation of GBDT requires continuous scanning of the algorithm and this becomes alarming in terms of dealing with big data giving rise to additional computational and time costs (Ke et al., 2017).

Thus, two new techniques are proposed by (Ke et al., 2017) called the Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) which makes up the LightGBM. Experimental results highlight the prowess of LightGBM in that it is shown to be an improvement over XGBoost and SGB in terms of computational and memory costs (Ke et al., 2017).

LightGBM, a decision tree-based machine learning technique provided high-quality forecasting performance and was used by all the top 50 competitors. Additionally, deep learning techniques like DeepAR and N-BEATS have also shown potential (Makridakis, Spiliotis, & Assimakopoulos, 2022).

Moving averages and exponential smoothing come into the category of traditional forecasting methods where one of their major drawbacks is that they are unable to capture the complex patterns and trends present in the sales data, especially for large-scale data with complexity embedded in it (Hyndman and Athanasopoulos, 2018). Additionally, statistical methods like ARIMA (AutoRegressive Integrated Moving Average) and Exponential smoothing are widely used for forecasting (Hyndman and Athanasopoulos, 2018). However, these methods face difficulty when dealing with large, complex datasets and are limited in their ability to capture and model non-linear relationships (Zhang, 2003).

The motivation to apply LightGBM model to M5 Competition Data comes from the competition’s leaderboard. The M5 competition’s winner utilised an equal weighted combination of different LightGBM models and various date-time features from the calendar data and pricing features from the pricing data were incorporated (Makridakis, Spiliotis and Assimakopoulos, 2022). The second placed method utilized the same equal weighted approach on LightGBM models with external adjustments from forecasts produced by N-BEATS (deep-learning NN for time series forecasting (Oreshkin et al., 2019). The fourth placed generated forecasts for the product-stores series using non-recursive LightGBM models and trained for each store (i.e.10 models) whereas the fifth placed candidate applied recursive LightGBM models and was trained for each of the seven departments. Additionally, among the top 50 performing methods, many candidates leveraged approaches similar to the first placed candidate by training recursive and non-recursive LightGBM models for each store, department, or store-department. The above results highlight the capabilities of the LightGBM model and its ability to process multiple relation time series and leverage exogenous/explanatory variables to improve forecast accuracy (Makridakis, Spiliotis and Assimakopoulos, 2022).

The Python library used for forecasting is skforecast. To train a time series in skforecast we utilised two classes 1) ForecasterAutoreg and 2) Backtesting\_forecaster

Using machine learning models to generate forecasts comes with the prerequisite of transforming the times series into a matrix form where features such as lag and external variables are utilised (Skforecast, 2024). Lags are values of the times series at some time points in the past. This transformation enables the machine learning models to capture patterns and relationships that exist between the past and future values. In this iteration, we will be utilising a recursive multi-step forecasting technique (Skforecast, 2024).

A diagram of a diagram

Description automatically generated

Figure 2: Illustration of how recursive forecasting technique is applied in skforecast

The above figure describes the concept behind recursive forecasting. In the first step, we are training the model on the observed values and predicting one step into the future. In the next step, along with our observed values, we are incorporating the value predicted in the previous step to retrain the model and predict one step ahead and so on. This is the basic concept behind Recursive forecasting which we will be applying using our class ForecasterAutoreg (Skforecast, 2024).

The second concept is the backtesting\_forecaster which is simply to say cross-validation for time series. It is used to assess the accuracy of the model on historical data and is considered a vital process in the generating a robust time-series forecasting model (Skforecast, 2024).

Feature engineering is considered a crucial step in the model development process. Lagging a time series constitutes shifting the value of the time series forward or backward in time by one or more steps. This results in observations which appear to have occurred later (Ryan Holbrook, n.d.). The lagged time series makes its values appear at the same time as the values we are trying to predict and makes them useful features for modelling (Ryan Holbrook, n.d.). Using lagged features is common in time-series analysis and machine learning in that it can reveal time dependencies and trends in the data and provide important insights (Gordon, 2023).

As correlation is used to measure the strength of linear relationship between two variables, Autocorrelation is used to gauge the strength of the linear relationship between the original time series and its past values. For example, if yt is the original time series and yt-1 is the lagged-by-1 version of the time series. Then autocorrelation helps us understand if there is any relationship between our original time series and time series using previous day values (Hyndman and Athanasopoulos, 2021, ch.2.8).

Autocorrelation is instrumental in identifying recurring patterns or trends within a time series. By examining how a variable correlates with its previous values at different lags, analysts can detect cyclic or seasonal patterns in the data. For instance, in economic datasets, autocorrelation might reveal whether certain indicators display regular patterns over specific intervals, such as monthly or quarterly cycles. The Autocorrelation Function (ACF) is a vital tool for modelling time series data. ACF helps pinpoint which lags exhibit significant correlations with the current observation. Understanding the autocorrelation structure is crucial for selecting suitable models in time series modelling (GeeksforGeeks, 2024).

Partial autocorrelation measures the direct correlation between yt and yt-k after removing the correlation introduced by intermediate lags on yt and yt-k. The high partial correlation at lag k indicates that lag k adds additional information which is not accounted for by lags before it. Thus, partial correlation can help us identify lags which can be used as features in our model (Machine Learning Mastery, 2020).

Rolling window features are a widely used technique for feature engineering in time-series analysis and machine learning and it encompasses calculating summary statistics like mean and standard deviation over a sliding window of previous values which aid in capturing trends and patterns in the data providing valuable insights and improving model performance (Gordon, 2023).

DateTime feature engineering involves creating new features based on date and time information to enhance the accuracy of machine learning models. This technique is particularly valuable in time-series analysis, where time-related patterns play a crucial role. For example, extracting features like the day of the week, hour of the day, or month of the year can capture seasonality and temporal trends in the data. An e-commerce sales dataset might exhibit increased sales volumes on weekends or during holidays like Christmas (Gordon, 2023).

Mean Squared Error (MSE) is a measure of the average squared difference between predicted and actual values (Brownlee, 2018). It assesses the quality of a model by penalizing larger errors more than smaller ones. The formula is:

A mathematical equation with numbers and symbols

Description automatically generated

Root Mean Squared Error (RMSE) is the square root of MSE, providing error in the same units as the original data:

A black text with black letters

Description automatically generated

When it comes to interpreting errors, the lower the error the better the accuracy.

# Dataset

The dataset used in this research was taken from the Walmart M5 competition, which provides a comprehensive hierarchical sales data structure. The dataset has both product-level and location-based aggregation, providing detailed analysis at various levels of granularity. This hierarchical nature is important for time series forecasting tasks and allows for the exploration of sales patterns across different levels of the hierarchy.

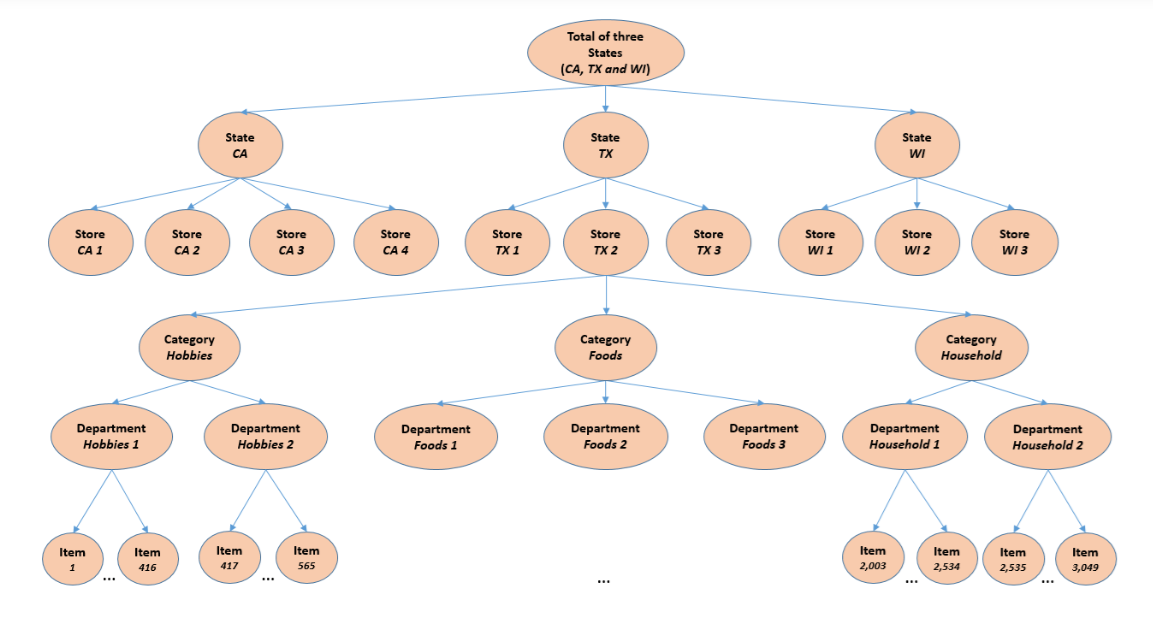


Figure 3: Illustration of the hierarchical nature of the Walmart data (Fig. 1 from Tian, Wang, & E (2021)

The above visualization helps in understanding the complex structure of the dataset and the interrelationships between different aggregation levels.

As illustrated in the figure above: The dataset can be split into two primary dimensions: **Location and Product Type**

**Location Dimension:**

* The geographical data has three states from the United States: California (CA), Texas (TX), and Wisconsin (WI).
* There are 10 stores: 4 in California, 3 in Texas, and 3 in Wisconsin.

**Product Dimension:**

* Products are categorised into three main product categories: Hobbies, Foods, and Household.
* These categories are further divided into seven product departments.
* The dataset includes 3,049 individual products, each belonging to a specific department and category.

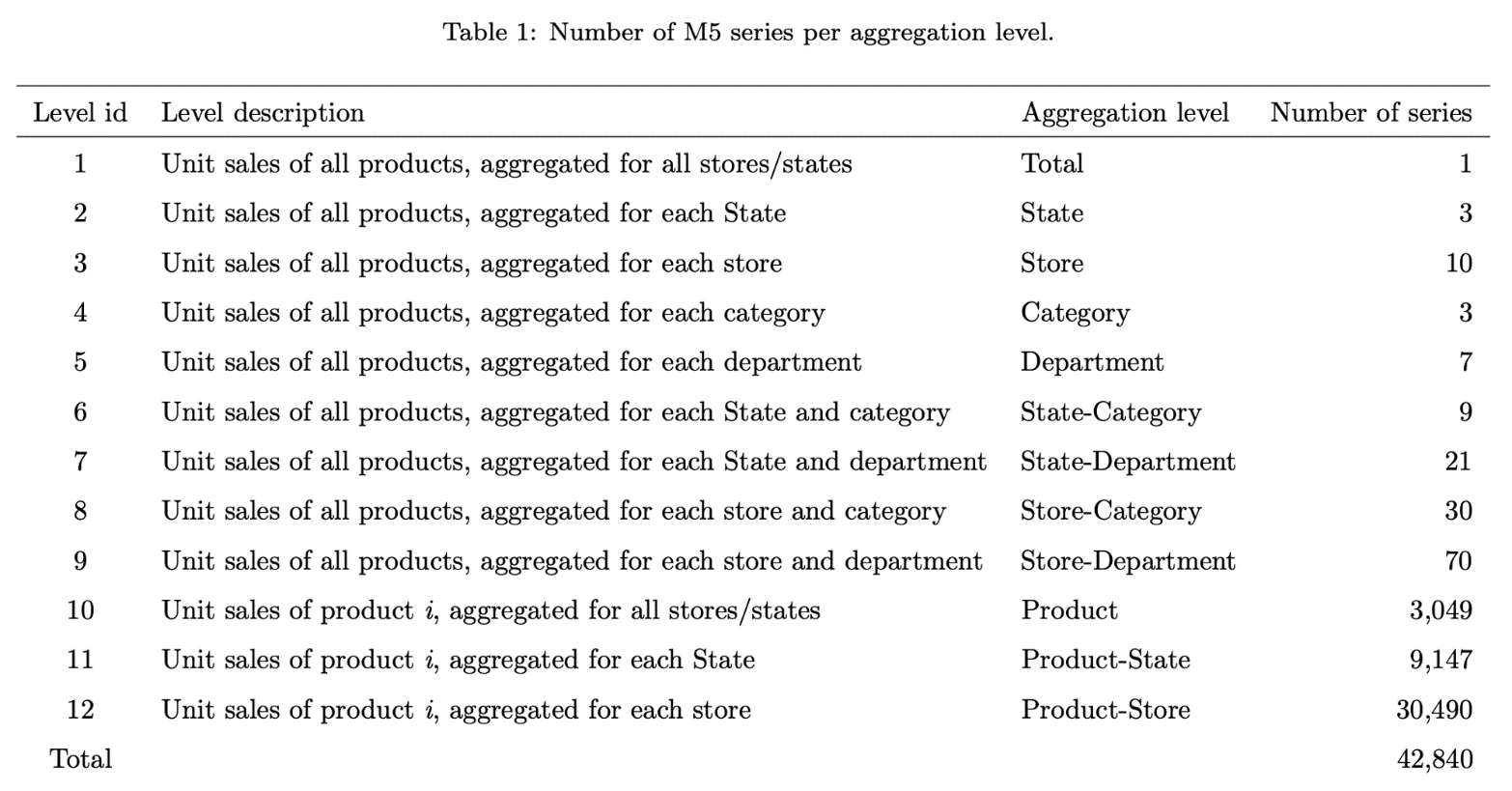


Figure 4: Breakdown of the number of series as per the aggregation level by (Hallam, J)

Here level id represents the position of the hierarchy and number of series represents the number of time-series present in each hierarchy. Level id 1 which represents the total sales of all three states (California, Texas, and Wisconsin). Thus, only one time-series is required to represent that. Level id 2 represents the total sales for each state. Thus, the state of California will have its own time-series which will be the addition of all the sales of all the stores which come under the California. The same is done for Texas and Wisconsin, thus we get three time-series to represent the sales for each state. Now when we talk about Level id 3, it represents all the stores. As there are 10 stores spanning 3 states, there will be 1 time-series per store. Level 10 represents the stores/states hierarchy, where for every state/ store there will consist of 3049 products. So, for example, product liquid0101 will be the same for all the three states. What will differ is its sales pattern across three different states. So, every State/ Store has a sales pattern of how each of its unique 3049 products are doing. Store CA 1 has unique 3049 products under it. Store TX 1 has the same 3049 products under it.

So, to sum it up, Walmart, has 3049 unique products to offer to its the customers. These unique products are delivered in each state and within each state to its respective stores. For example, 100 units of liquid0101 were sold in California, 100 units in Texas, 5 units in Wisconsin. Of these 100 units in California, 30 were sold in Store 1, 40 in Store 2, 30 in Store 3. For Texas, 30 in Store 1, 30 in store 2, and 30 in store 3. For Wisconsin, 5 in store 1 and 0 in store 2 and store 3. The total units sold across 3 states is 205. This gives us the overall picture of the total sales of liquid0101. But the expected total sales of liquid0101 is 300. This where the different levels/ hierarchies are so important. If we divide these total sales according to their respective states, then we see 100 in California, 100 in Texas, 5 in Wisconsin. The 5 in Wisconsin raises concerns regarding low sales of liquid0101 in Wisconsin which leads us to question what’s happening in each store in Wisconsin, what’s happening within each category, department in Wisconsin. This hierarchy leads us to extract and analyse the sales pattern of liquid0101 in all stores in Wisconsin, understand why liquid0101 has low sales, based on the historical pattern create an accurate forecast for the next x number of days, based on the forecast and other indicators, make decisions to decrease or increase inventory for liquid0101 and plan on production accordingly.

This hierarchy helps in understanding the overall business performance, sales patterns and make its easier to process the data and understand performance of each hierarchy at a general level and at a detailed level, which helps in cutting costs, improving profits and business performance.

**Aggregation Levels:**

The table provided lists the number of series at different levels of aggregation within the hierarchy. Here’s a breakdown:

1. **Total Level (1 series)**: This is the overall aggregation of all product sales across all stores and states.
2. **State Level (3 series)**: This aggregates the sales data at the state level (CA, TX, WI).
3. **Store Level (10 series)**: Here, sales data is aggregated at the store level, across all the 10 stores.
4. **Category Level (3 series)**: Sales data is aggregated by the three main product categories (Hobbies, Foods, Household).
5. **Department Level (7 series)**: Aggregation is done at the department level within each category.
6. **State-Category Level (9 series)**: This level combines the state and category dimensions, aggregating sales data across the three states for each of the three categories.
7. **State-Department Level (21 series)**: This aggregates the data at the intersection of state and department.
8. **Store-Category Level (30 series)**: Sales data is aggregated for each store across the three product categories.
9. **Store-Department Level (70 series)**: This aggregates the sales data by both store and department.
10. **Product Level (3,049 series)**: This is the individual product level aggregation across all stores and states.
11. **Product-State Level (9,147 series)**: Aggregates the sales of each product by state.
12. **Product-Store Level (30,490 series)**: This is the most granular aggregation, showing sales for each product in each store.

# Chapter 1: Exploratory Data Analysis:

Data is a very important part of any research; understanding it better through a visual medium enhances our understanding of various aspects such as distributions, trends, and patterns. Exploratory Data Analysis is a crucial step in any data-driven research, particularly when dealing with complex and hierarchical datasets like the one provided by the Walmart M5 competition. The primary goal of EDA is to gain an in-depth understanding of the data, uncover underlying patterns and detect anomalies that can guide subsequent analyses and model development.

To build accurate forecasts, it is vital that the data at hand should be studied, analysed and visualised to understand the different patterns and relationships between variables. Here, Exploratory Data Analysis comes to picture and is considered as an important step in the data science process, especially for projects which work on time series forecasting. EDA comprises of detailed examination of the data at hand and involves identifying patterns, detecting anomalies, and understanding the structure of the data in a detail manner (Tukey, 1977). EDA not only helps in improving our understanding of the data but aids in identifying preprocessing requirements such as dealing with missing data, scaling the data and encoding categorical variables (James et al., 2013).

This step is important because it forms the foundation on which different decisions with regards to the selection of models, feature engineering strategies and data preprocessing techniques are built on (Zhu and Xiong, 2015). In this project, EDA played a crucial role in the model development process, leading to better decision making which in turn led to accurate and reliable forecasts. Once the EDA has been performed, the model development process to build accurate forecasts is commenced.

One of the key questions explored through EDA is whether a single product exhibits similarities or differences in sales behavior across different regions. Furthermore, the product is analysed to check if it possesses any seasonal characteristics across the Christmas and New Year period.

This is particularly important for retail businesses like Walmart, Tesco, Aldi and Asda that operate across diverse markets where consumer behavior can vary significantly based on factors such as region, promotions, weather etc. Understanding these variations can inform inventory management, marketing strategies, and resource allocation, ensuring that each region is operated according to its unique demand patterns.

In this section, we answer some questions regarding time series.

1. Does a single product across two different regions i.e. California and Texas show the same behaviour in its sales patterns?

To answer this question product\_id **FOODS\_1\_001** was analysed across the CA (California) and TX (Texas).

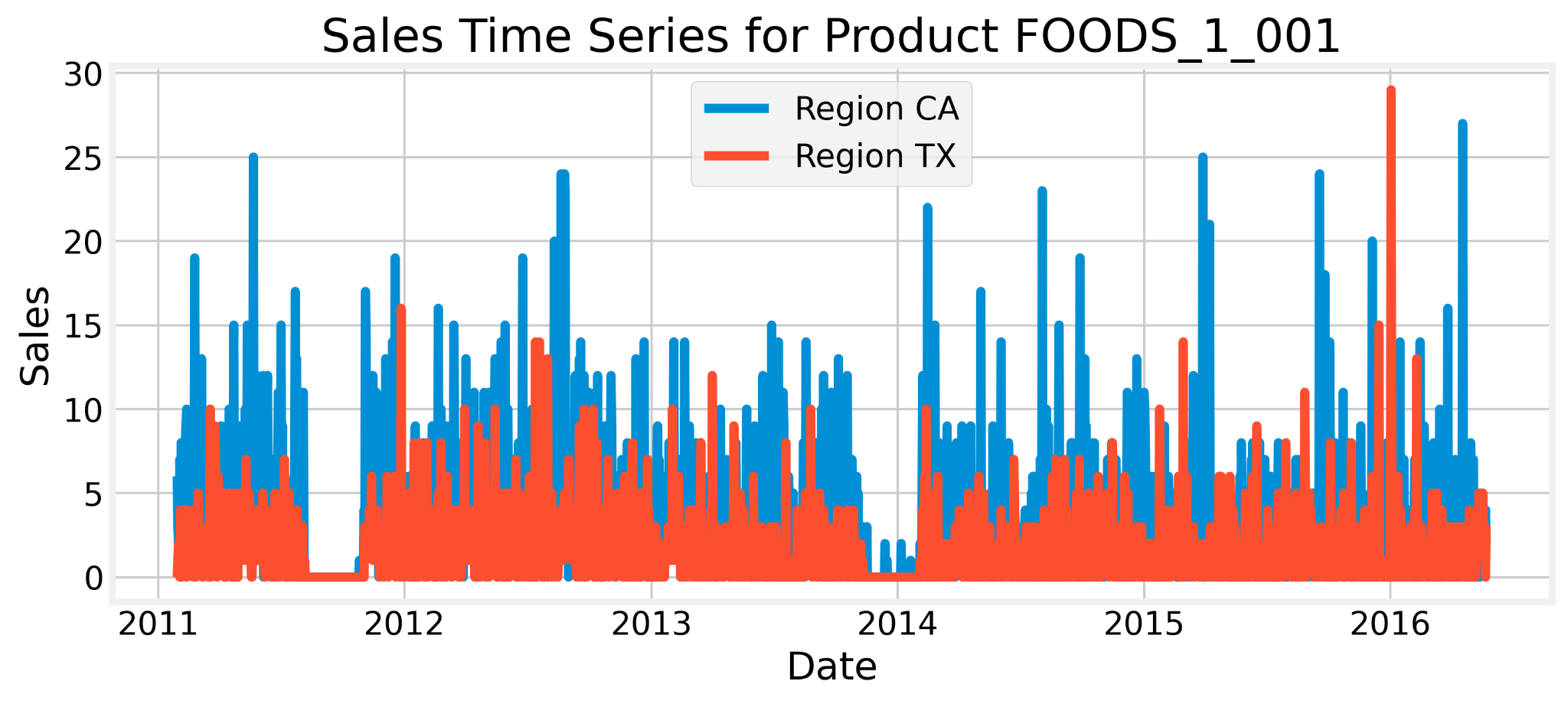


Figure 5: Illustration of the sales for FOODS\_1\_001 across California and Texas regions

From the above figure, it can be observed that the quantity of sales differs for both regions where the sales for California are comparatively higher than sales for Texas. There is one distinct similarity which can be observed at the end of 2012 when the sales for both regions were zero which might indicate shortages or stockouts or having no inventory present for the respective product. This product seems to appear more popular in California as compared to Texas.

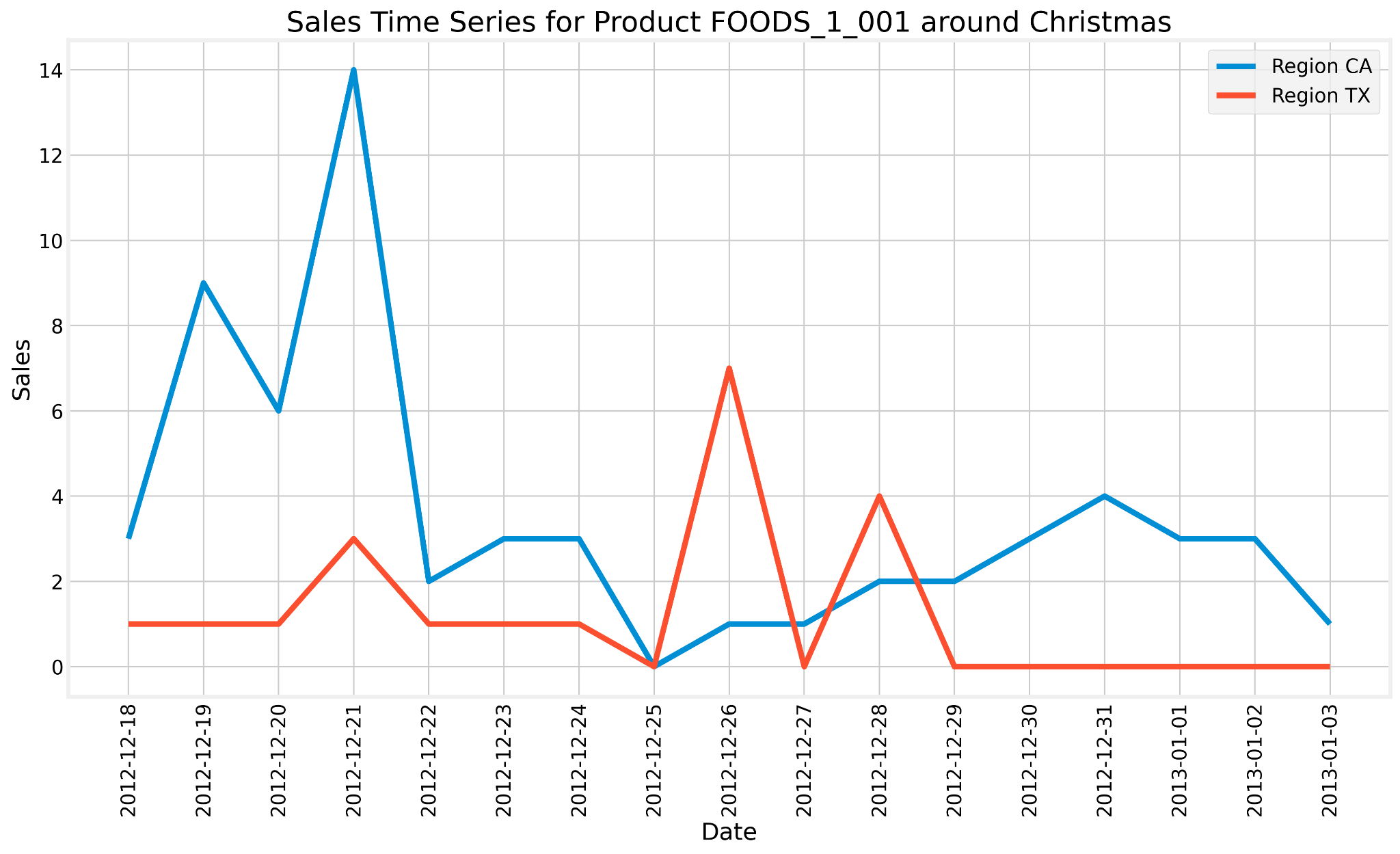


Figure 6: Illustration of sales for FOODS\_1\_001 around Christmas for California and Texas regions

The data was filtered from '2012-12-18' to '2013-01-03' to check the pattern of sales during the Christmas period. When it comes to the Texas region, there was a sudden increase in demand on the 26th of December 2012 whereas in the California region, the sales did not peak as much. The low frequency of sales around the Christmas period indicates that the product does not showcase seasonality around the Christmas period.

1. Does a single product across two different regions i.e. Wisconsin and Texas show the same behaviour in its sales patterns?

Product\_id **FOODS\_1\_001** was analysed across the WI (Wisconsin) and TX (Texas) to answer this question.

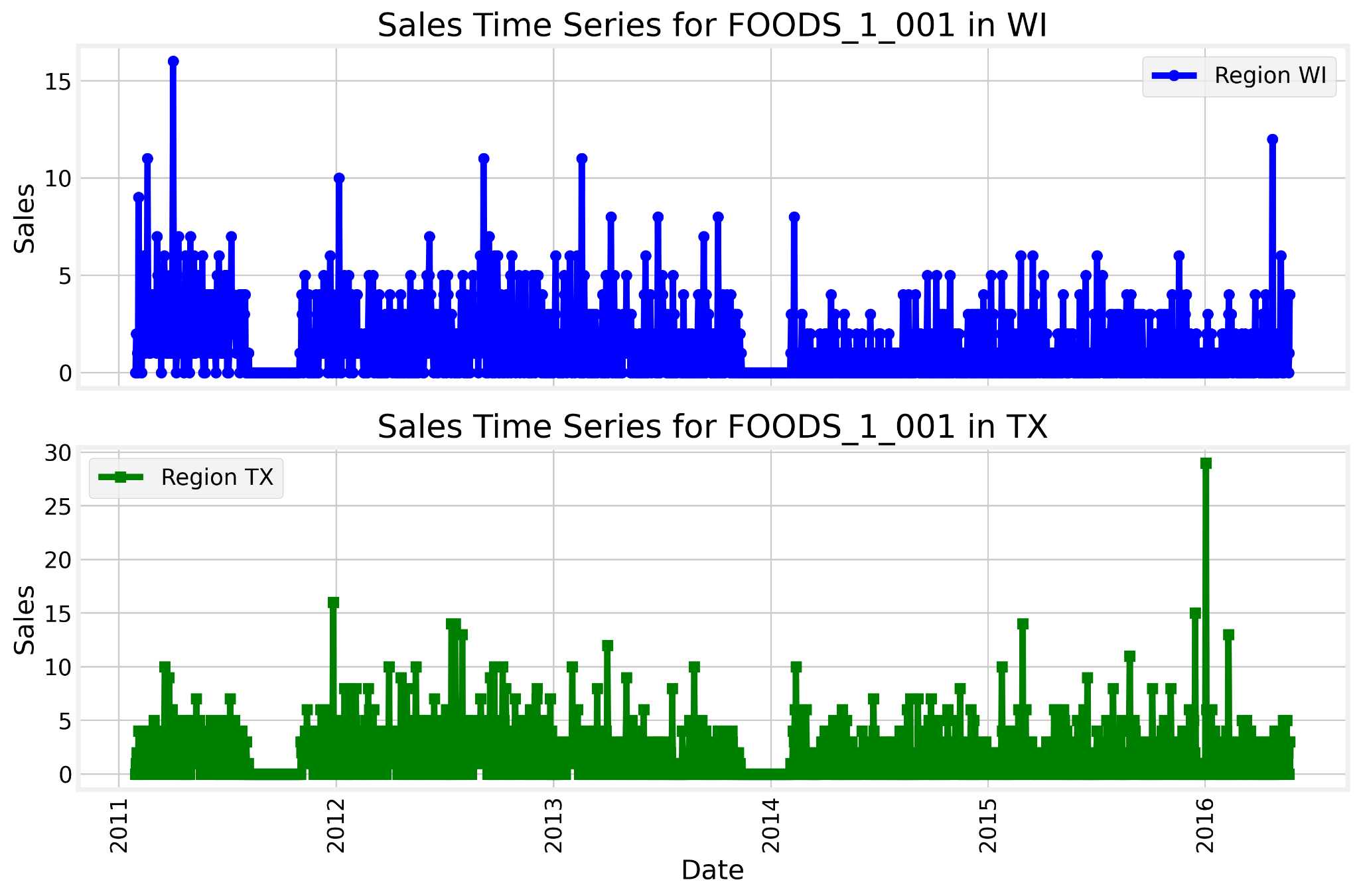


Figure 7: Illustration of the sales for Wisconsin and Texas region for product FOODS\_1\_001

From the above figure, it can be observed that the quantity of sales for FOODS\_1\_001 stays below 15 for both regions with minor surges in the Texas region at certain points in time. Interestingly, it can be observed that at the end of 2012 and the start of 2014, both regions might have witnessed shortages or stockouts for the product.

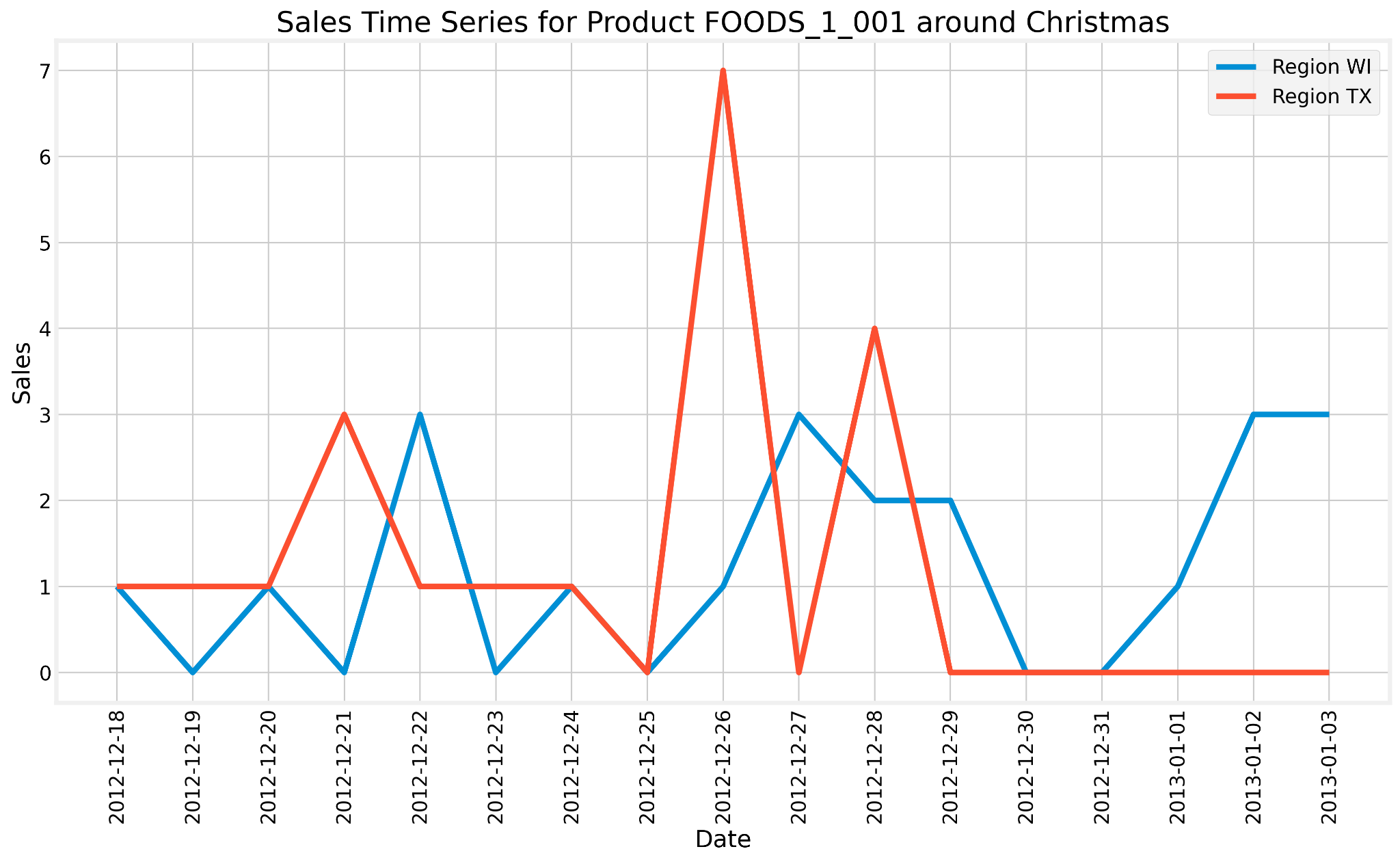


Figure 8: Illustration of the Christmas period sales for Wisconsin and Texas region for product FOODS\_1\_001

The data was filtered from '2012-12-18' to '2013-01-03' to check the pattern of sales during the Christmas period. When comparing both the regions, Texas saw a surge in demand for the product on 26th December whereas the demand for Wisconsin during the same period does not show any significant surge.

1. Does a single product across two different regions i.e. California and Wisconsin show the same behaviour in its sales patterns?

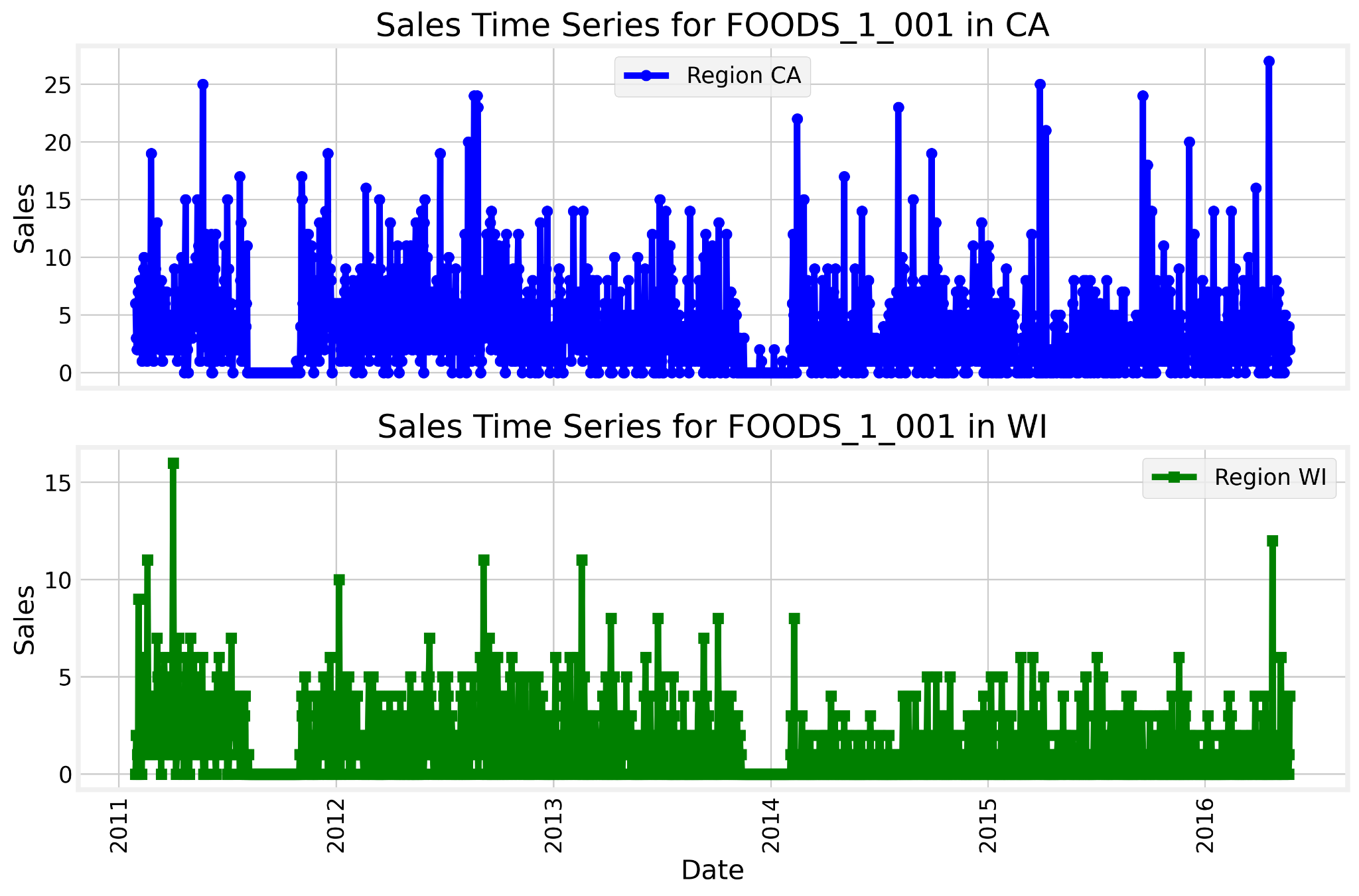


Figure 9: Illustration of the sales for California and Texas region for product FOODS\_1\_001

When observing the time series for the product across California and Wisconsin region it can be observed that the sales for California are higher compared to the Sales for Wisconsin. At certain time points, Sales for California reach as high as 25 units whereas for Wisconsin, they reach as high as 15 units. The maximum number of sales recorded for Wisconsin on a given day is 10 units whereas the maximum number of sales recorded for California on a given day is 24 units

This indicates that the product is more consumed in California than in Wisconsin. One similarity can be observed around the end of 2012 for both the regions where there were no sales recorded. This can be due to shortages or stockouts.

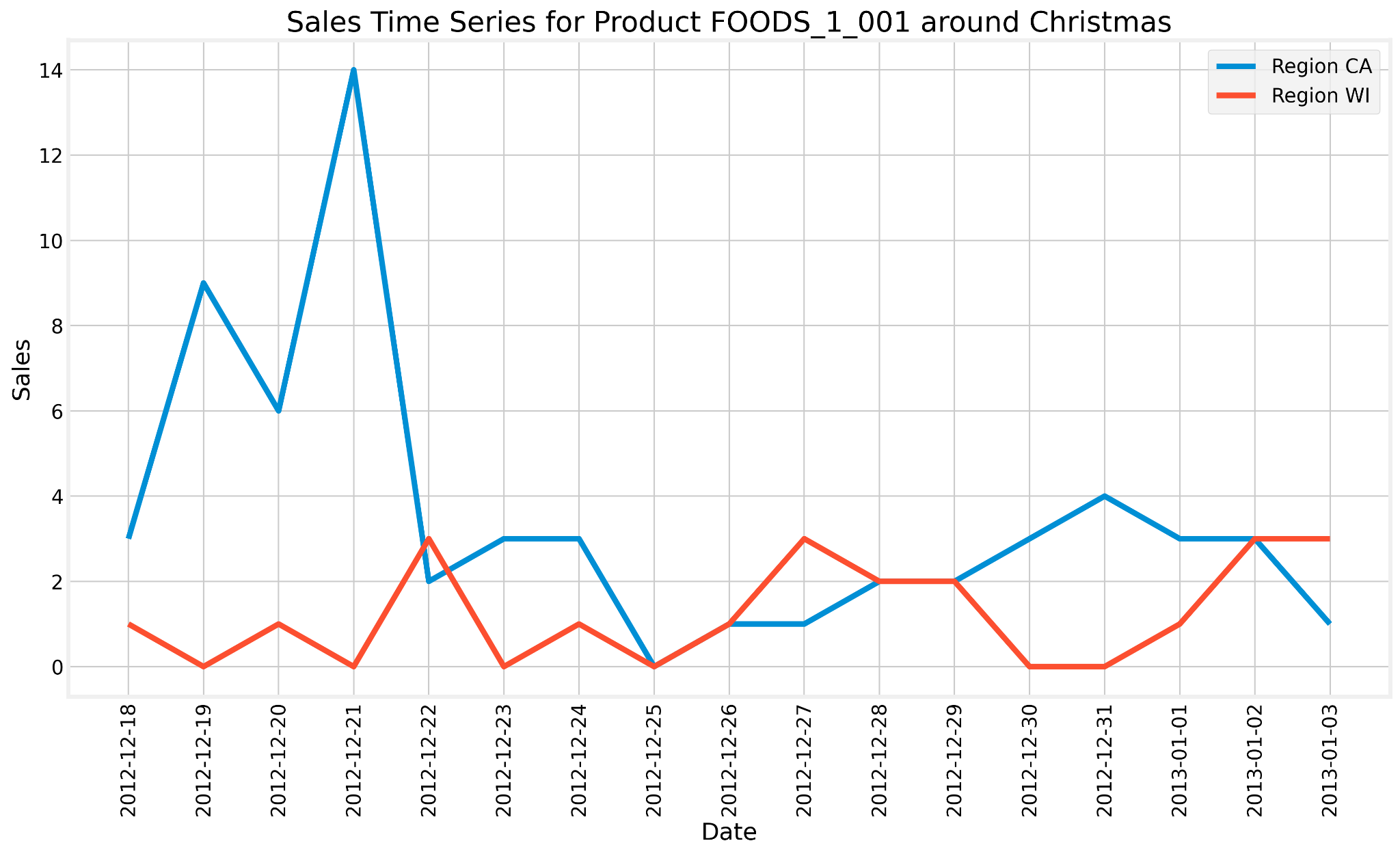


Figure 10: Illustration of the Christmas period sales for California and Texas region for product FOODS\_1\_001

When observing the sales around the Christmas period, it can be observed that there is no significant surge in the demand for the product in California and Wisconsin region.

Below we compare the time series for the product ID FOODS\_3\_555 across stores spanning various regions. The names of the stores are CA\_4 (California), TX\_1 (Texas), and WI\_1 (Wisconsin).

****

Figure 11: Illustration of the latest year’s sales pattern of FOODS\_3\_555 product in a specific store in California

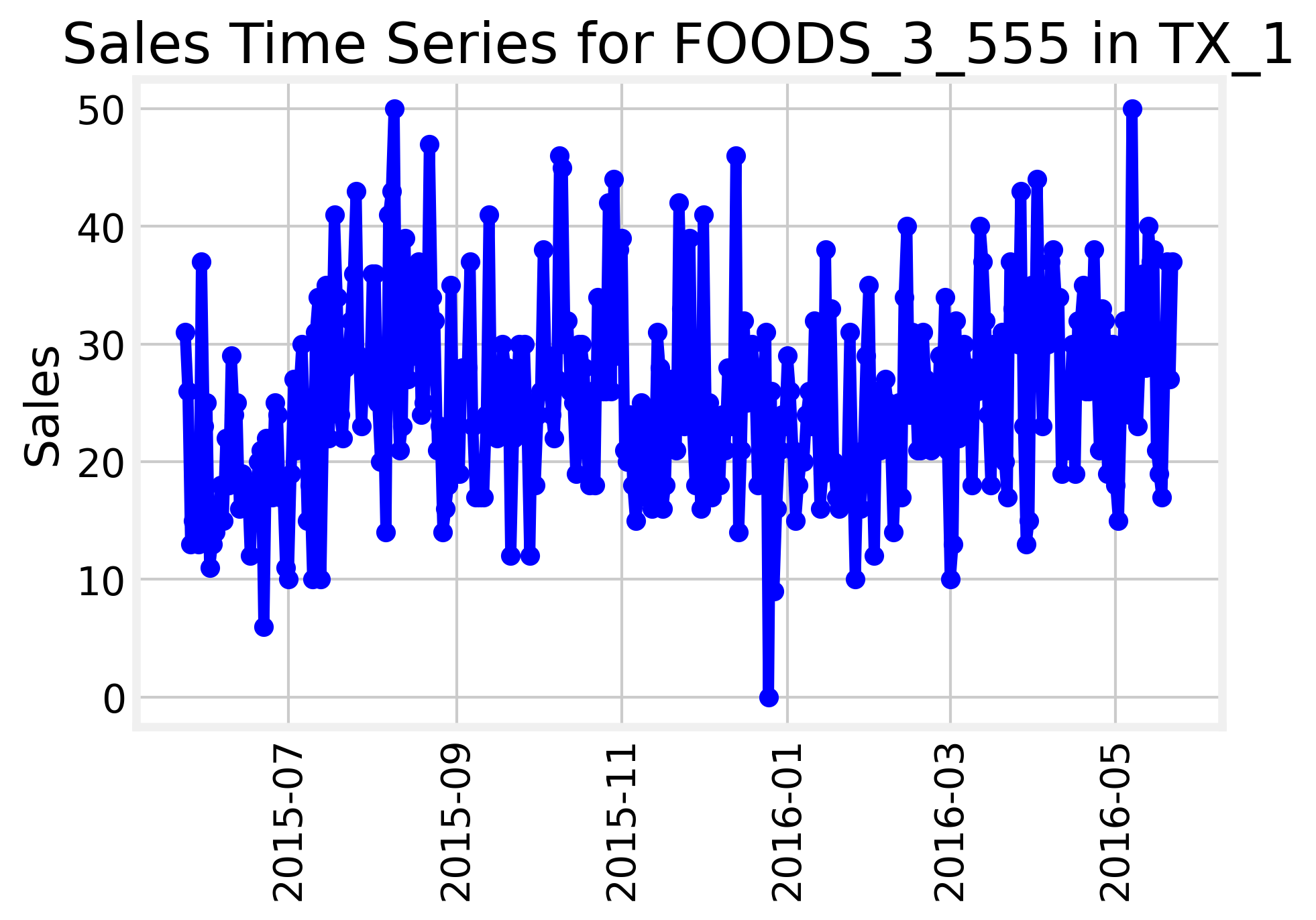
****

Figure 12: Illustration of latest year’s sales pattern of FOODS\_3\_555 product in a specific store in Texas

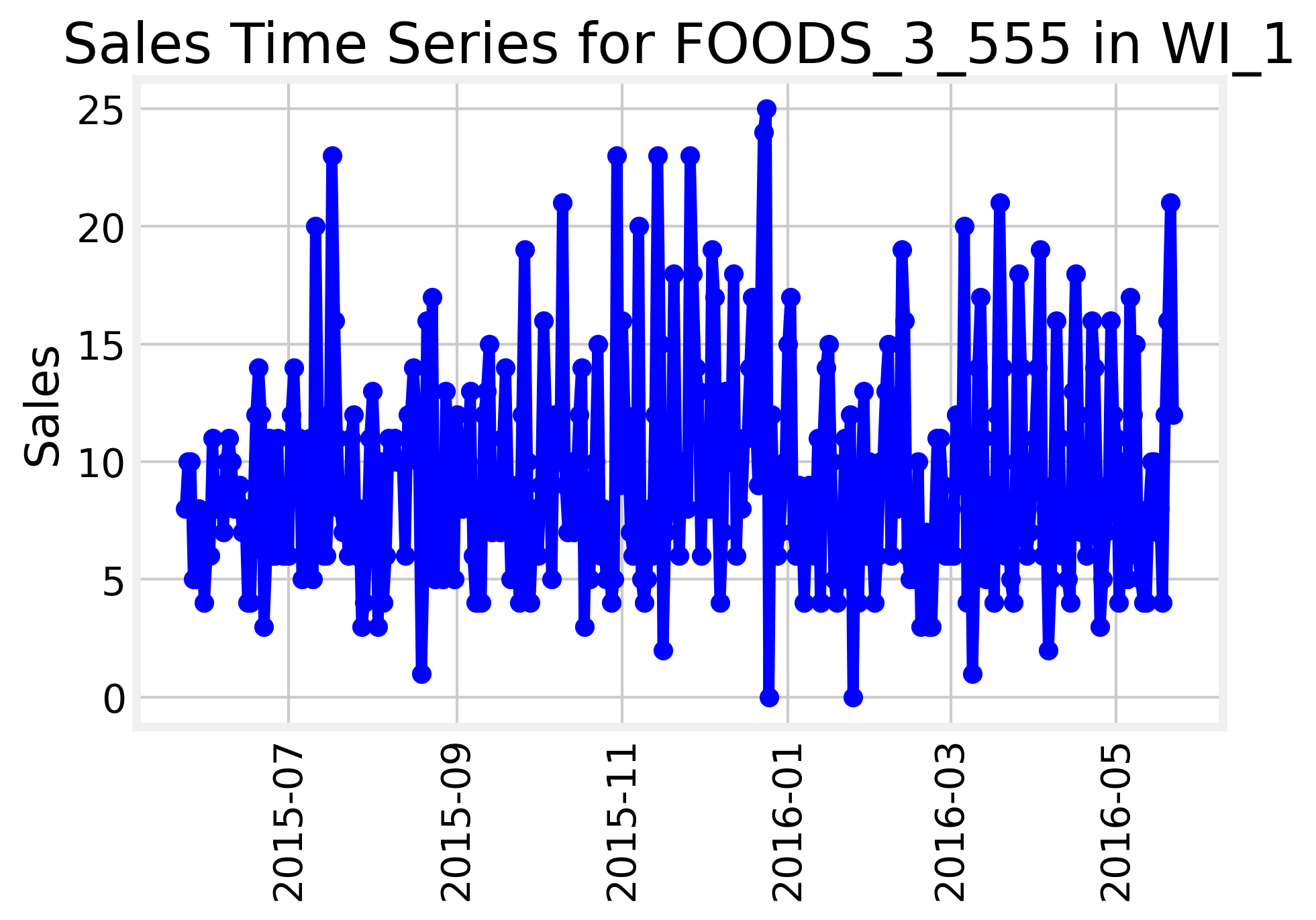
****

Figure 13: Illustration of latest year’s sales pattern of FOODS\_3\_555 product in a specific store in Wisconsin

When comparing the time series for the product FOODS\_3\_555 across different stores in different regions for the latest year recorded, it can be observed that all three-time series exhibit different behaviours. When considering the TX\_1 store based in Texas, the sales of the product reach a maximum of 50 units whereas that of WI\_1 and CA\_4 reach 25 and 17.5. Thus, when just observing the latest year, the product is famous in this particular store in Texas as compared to the other stores in California and Wisconsin. Interestingly, when observing the sales in the California store, it can be observed that the sales for the store seem to be rising at the end of the year.

# Chapter 2: Light Gradient boosting machine on HOBBIES\_1\_018\_CA\_1\_validation

The M5 competition contains 3049 products which represent different departments, categories, stores, and states. Due to the limitations of memory and the complexity of the problem. The initial objective was to forecast the sales of the single item/product/ SKU. The product chosen was HOBBIES\_1\_018\_CA\_1\_validation. This contains product contained data worth 1913 days. The second chapter will be divided into five sections as below:

• Loading the dataset

• Data-preprocessing

• Feature engineering

• Model development

• Model Evaluation

The above five sections are considered as the general model development process. Each section will describe what was implemented. Each iteration consists of these 5 sections what actions were taken within each iteration.

## Iteration 1:

**Loading the data:** The dataset was loaded in the jupyter notebook, and the sales patterns of the data can be observed below.

The plot of the figure is as follows: From the below figure, the intermittent nature of the product can be observed.



Figure 14: Illustration of the intermittent nature of the HOBBIES\_1\_018\_CA\_1\_validation sales data where x-axis denotes the number of days

As observed from the above plot, where the x-axis is the number of days recorded and the y-axis is the sales on a given day. The plot illustrates that the time series contains intermittent sales.

**Data-preprocessing:** The data was preprocessed to reduce memory usage which resulted in successfully reducing memory usage by 86.5%.

**Feature Engineering:** As this was the first iteration only lags of sales up to 24 periods were included as features to the model.

**Model development:**

To model the series, the data was divided into three parts train, validation, and test. The first 1322 data points were allocated for the train, the next 265 for validation and the next 326 for the test. It is important to note that techniques like random shuffle or traditional cross-validation do not work on time-series data as they violate the temporal order.

Inputs to Forecasterautoreg and backtesting\_forecaster are as follows:

forecaster = ForecasterAutoreg(

regressor=LGBMRegressor(

learning\_rate=0.1,

max\_depth=5,

n\_estimators=500,

num\_leaves=32, # Set num\_leaves to be greater than 2^max\_depth

random\_state=123,

force\_row\_wise=True,

# or force\_col\_wise=True if memory is a concern

),

lags=24

)

In the ForecasterAutoreg class, recursive forecasting is implemented where the regressor or our model is chosen to be LGBM Regressor, and all the parameters are initialized.

metric, predictions = backtesting\_forecaster(

forecaster=forecaster,

y=data['sales'],

initial\_train\_size=initial\_train\_size,

fixed\_train\_size=False,

steps=10,

refit=False,

metric='mean\_squared\_error',

verbose=True # Change to True for detailed information

)

In the backtesting\_forecaster we passed our defined forecaster and our target variable along with parameters such as initial\_train\_size, steps, refit and our metric our choice.

**Model Evaluation:**

Upon running the above classes, some of the inconsistencies which we observed were as follows: 1) **The mean squared error: 0.12077583661944795** did not seem representative which raised doubt in the mean-squared-error metric and its appropriateness for the data where most of the sales values were zero.

2) As observed from the figure below, in the test set, the model is unable to capture any peaks or any meaning relationships in the data. This can be observed in the below figure:

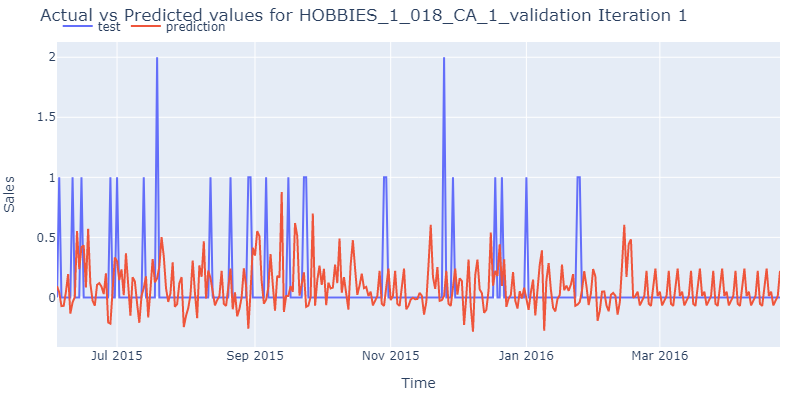


Figure 15: Illustration of the actual and predicted values for HOBBIES\_1\_018\_CA\_1\_validation product

## Iteration 2:

**Loading the data:** The same dataset was loaded as the previous iteration.

**Data-preprocessing:** The data was preprocessed to reduce memory usage which resulted in successfully reducing memory usage by 86.5%.

**Feature engineering:** In the second iteration, along with the sales data we joined the calendar dataset which consisted of different features such as weekday and different event types. A lag of 28 days and rolling windows statistics such as mean and standard deviation for 7 and 28 days were calculated and incorporated as features. Furthermore, time features such as day and month were calculated and incorporated into the model. Feature engineering is important to improve the forecast performance of our model thus experimenting with lag of 28 and rolling window statistics was considered and incorporated as features to our model.

**Model development:**

forecaster = ForecasterAutoreg(

regressor=LGBMRegressor(

learning\_rate=0.1,

max\_depth=5,

n\_estimators=300,

num\_leaves=40,

objective = "tweedie”, # Set num\_leaves to be greater than 2^max\_depth

random\_state=123,

force\_row\_wise=True,

# or force\_col\_wise=True if memory is a concern

),

lags=24,

)

metric, predictions = backtesting\_forecaster(

forecaster=forecaster,

y=data['sales'],

initial\_train\_size=initial\_train\_size,

fixed\_train\_size=False,

steps=10,

refit=True,

metric='mean\_squared\_error',

verbose=True,

exog = exog # Change to True for detailed information

)

Some of the parameters which were incorporated in this iteration are objective = ‘tweedie’ in

the ForecasterAutoReg class and exog = exog in the second class. The exog parameter was passed all the features such as rolling window, lags and date time features. The objective parameter was selected as Tweedie due its properties where it can have mixture of zeros and non-negative data points.

**Model evaluation:**

Upon running this experiment, **the mean-squared error is:**

**0.009742097226445246** and the graph is as follows:

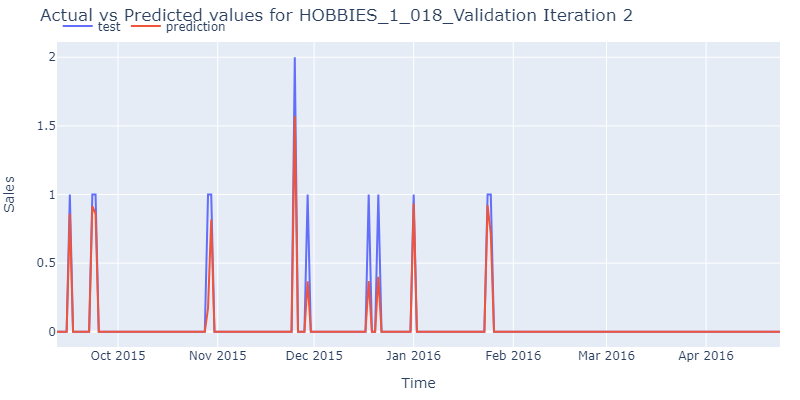


Figure 16: Illustration of the actual and predicted values for HOBBIES\_1\_018\_CA\_1\_validation product

Upon observation, it looks like the model can capture the peaks well. But this is a result of a major error in the backtesting\_forecaster class where in the parameter exog which enables the incorporation of external variables, we are giving target i.e. (y) as feature in exog. Upon removing the target as a feature, the observed graph is as follows:

A graph of a number of people

Description automatically generated with medium confidence

Figure 17:Illustration of the actual and predicted values for HOBBIES\_1\_018\_CA\_1\_validation product

As observed from the above figure, the only improvement from first iteration is that the tweedie distribution has enabled our predictions to stay at zero instead of fluctuating in the negative y-axis. Apart from this, it seems that the model is unable to capture any relationships in the data which raises doubts and emphasizes the need to re-examine the experiment.

In summary, Chapter 2 was unable to provide us with any significant improvements when forecasting sales data which was highly intermittent (contained many zero values). Further work needs to be done in this area.

# Chapter 3: Light Gradient Boosting Machine on FOODS\_3\_555\_CA\_1\_validation for predicting the next 7.5 months

## Iteration 1:

The HOBBIES\_1\_018\_CA\_1\_validation time-series data had many zero values where trend and seasonality patterns could not be observed. The choice of forecasting the next 7.5 months of sales is random and the main motivation was to assess if the model could forecast sales for longer time periods. Thus, we turn our attention to another SKU which has continuous values and tried to model it. The graph of the time series looks as follows:

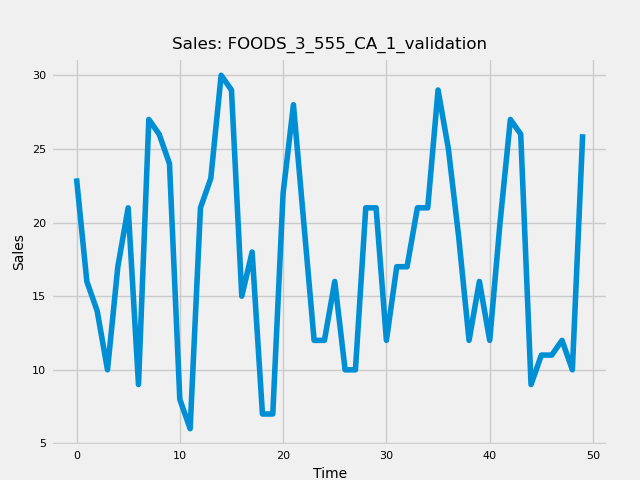


Figure 18: Illustration of the movement of the time-series for product FOODS\_3\_555\_CA\_1\_validation for the first 50 days

**Loading the data:** FOODS\_3\_555\_CA\_1 item was loaded into the jupyter notebook

**Data-preprocessing:** The data was preprocessed to reduce memory usage which resulted in successfully reducing memory usage by 86.5%.

**Feature engineering:** In the first iteration the time-series was modelled by incorporating features such as 'year', 'snap\_CA', 'snap\_TX', 'snap\_WI', 'y\_window\_7\_mean', 'y\_window\_7\_std', 'y\_window\_28\_mean', 'y\_window\_28\_std', 'month', 'day' and lag values till 30. The snap\_CA, snap\_TX and snap\_WI is the Supplemental Nutrition Assistance Program which provides food-benefits to low-income families (USDA, 2024). The y\_window\_7\_mean and y\_window\_7\_std deviation represent the rolling window mean and standard deviation statistical features.

**Model development:**

The parameters for the LGBM regressor are:

learning\_rate=0.1,

max\_depth=6,

n\_estimators=500,

num\_leaves=30,

objective = "tweedie

random\_state=123,

force\_row\_wise=True

**Model evaluation:**

This resulted in the mean squared error of **31.163215312952573**

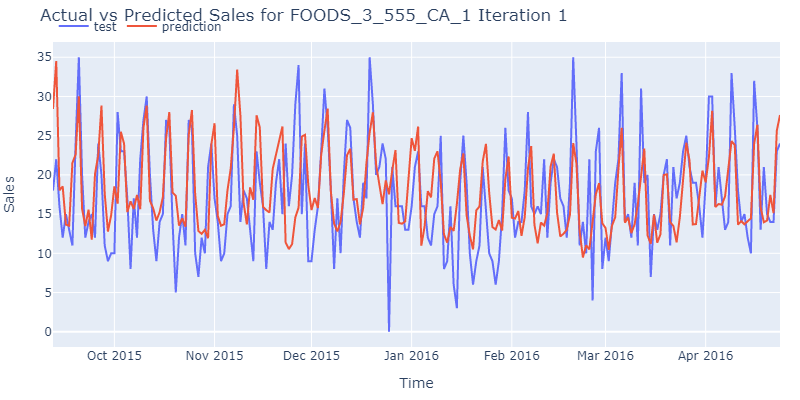


Figure 19: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation

From the above figure, for the first iteration, the model can capture peaks to some extent but not troughs.

Thus, further modelling and feature engineering is required to improve performance.

## Iteration 2:

Iteration 2 had all the similar steps in loading the data, data-preprocessing.

The motivation for iteration 2 was to improve more upon the results of iteration 1. Furthermore, iteration 2 applied autocorrelation to the sales data to identify which lag features might be useful to incorporate into our model.

**Feature engineering:**

A graph with blue lines

Description automatically generated

Figure 20: Illustration of the autocorrelation of FOODS\_3\_555\_CA\_1\_validation sales feature up to 30 lags

Autocorrelation plot of the sales data up to lags 30 were created. The values/lags lying outside the blue bounds can be considered significant and can be experimented with by incorporating them into our models (Pennsylvania State University, n.d.). From the above figure, it appears that lags 1, 6, 7, 8, 10, and 11 are particularly significant. Incorporating the following lags into our model.

Features incorporated are 'year', 'month', 'day' and lags 1, 6, 7, 8, 10, 11.

**Model development:** Model parameters are kept the same as iteration 1 as the focus was to see if the incorporating just year, month and day features and the lag features which the Autocorrelation function calculated to be significant in improving forecast performance.

**Model Evaluation:** This resulted in the **mean squared error of 35.7947901394842**. This highlights that iteration 1 delivered higher forecast performance as compared to iteration 2. But it is important to highlight the fact that iteration 2 was developed with fewer features as compared to iteration 1. Thus, we can conclude on a non-critical level that adding more features can lead to improved forecast performance.

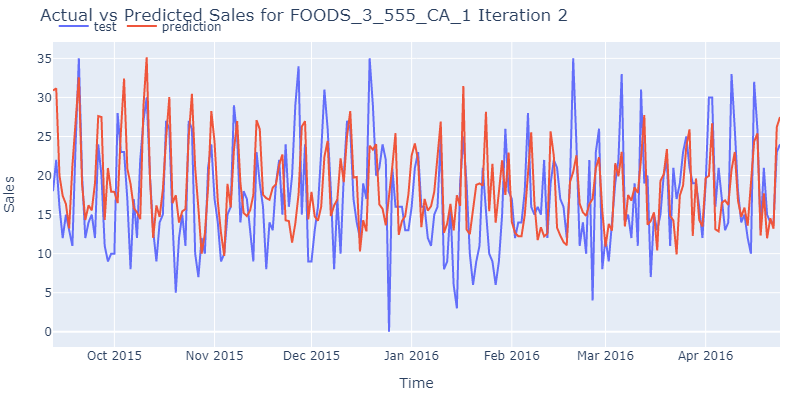


Figure 21: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation

## Iteration 3:

In this iteration, features from iteration 1 and lags from iteration 2 were incorporated to assess if combining features from both the above iterations improved forecast performance.

**Feature engineering:**

Features incorporated are 'year', 'snap\_CA', 'snap\_TX', 'snap\_WI', 'y\_window\_7\_mean', 'y\_window\_7\_std','y\_window\_14\_mean', 'y\_window\_14\_std', 'y\_window\_28\_mean',

'y\_window\_28\_std', 'month', 'day' and lags 1, 6, 7, 8, 10, 11.

**Model development:** Model parameters are kept the same as iteration 2.

**Model evaluation:** This resulted in the **mean squared error of 37.549325358479585**

The model performance declined as compared to iteration 1 and 2.

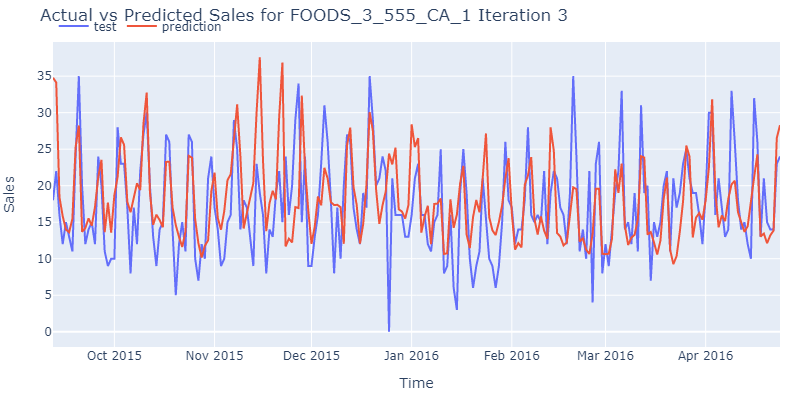


Figure 22: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation

## Iteration 4:

**Feature engineering:** Lags features until 60 were incorporated into our model. There was no practical basis for creating lag features until 60. This was rather an experimenting approach taken to check if model performance could be improved or not.

Features incorporated are 'year', 'snap\_CA', 'snap\_TX', 'snap\_WI', 'y\_window\_7\_mean', 'y\_window\_7\_std','y\_window\_14\_mean', 'y\_window\_14\_std', 'y\_window\_28\_mean',

'y\_window\_28\_std', 'month', 'day'

**Model development:** num\_leaves = 64 parameter of the LightGBM class was updated.

Model evaluation: The **mean squared error** was reduced to **31.664441037541923**

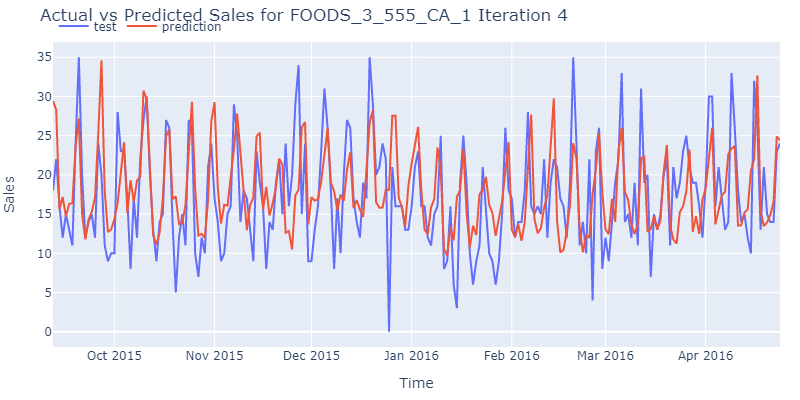


Figure 23: Illustration of the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation

Surprisingly, the forecast performance improved as compared to iteration 3.

## Iteration 5:

In this iteration, we applied autocorrelation and partial autocorrelation functions to our time series. Partial autocorrelation measures the direct correlation of a time series with a lagged version of itself by eliminating the effects of the intermediate lags (Machine Learning Mastery, 2020). This helps us to identify lags which have a direct strong effect on the original time series and can be a strong indication of utilising those lags as our features (Jim, no date), (Cleveland, 1979)  
(Box et al., 2015). Thus, the results from partial autocorrelation will help us decide which lags to incorporate into our model**.**

A graph with blue lines

Description automatically generated

Figure 24: Illustration of the partial autocorrelation on FOODS\_3\_555\_CA\_1\_validation sales feature

The above figure gives significant PACF values at lag 1, 6 and 7. The PACF value at lag7 and its multiples at 14, 21, 28 indicate that there is some weekly seasonality in the data. Furthermore, lag1 and lag6 also look significant and we incorporate them in this iteration.

Furthermore, we calculated PACF without de-trending and de-seasonalising our data as the main assumptions to calculate PACF require the time series to be stationary i.e. mean and variance of the time series should be constant (Shumway and Stoffer, 2017), (Brockwell and Davis, 2016).

First, we calculate PACF by de-trending our data. This can be achieved with a technique called LOWESS (Cleveland, 1979).

After applying LOWESS, the de-trended data is obtained which is illustrated below:

A graph with blue lines

Description automatically generated

Figure 25: Illustration of the partial autocorrelation on FOODS\_3\_555\_CA\_1\_validation sales feature after de-trending

As the original data did not have a strong trend to it, the PACF of the de-trended series is like that of the original series.

Now, PACF is calculated by de-trending (i.e. removing the trend component) and de-seasonalising (i.e. removing the seasonal component from the data). What remains is the residual for which PACF will be calculated and plotted (Hyndman and Athanasopoulos, 2018).

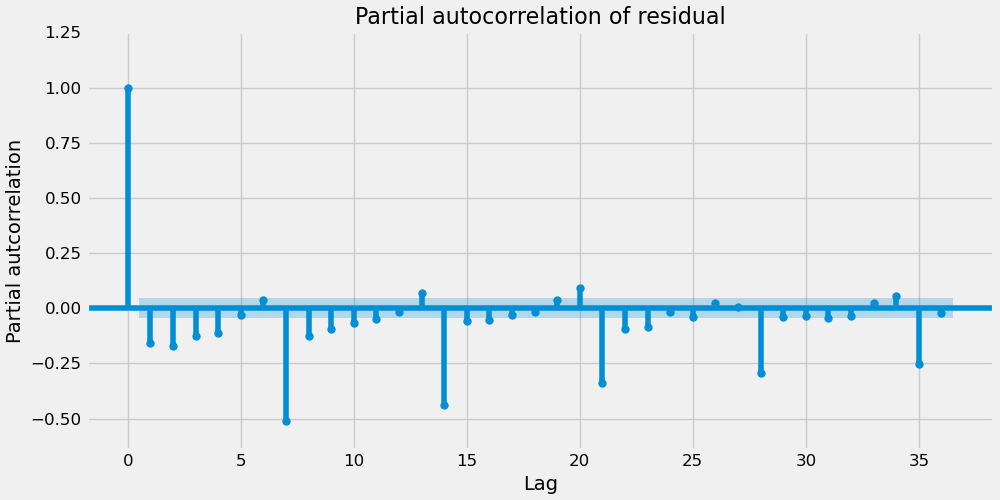


Figure 26: Illustration of the partial autocorrelation on FOODS\_3\_555\_CA\_1\_validation sales feature after de-trending and de-seasonalising

There appears to be some significant lags at multiples of 7 which suggests that some part of the seasonal component is still in the residuals and was not perfectly extracted by STL. Practically speaking from looking at this plot there would not be an additional lag beyond 1 or 2 that we would want to add for feature engineering purposes.

**Feature engineering:** After calculating PACFs, lags1, 6 and 7 were incorporated as features into our model. Features incorporated are 'year', 'snap\_CA', 'snap\_TX', 'snap\_WI', 'y\_window\_7\_mean', 'y\_window\_7\_std','y\_window\_14\_mean', 'y\_window\_14\_std', 'y\_window\_28\_mean',

'y\_window\_28\_std', 'month', 'day'

**Model Evaluation:** **mean squared error: 35.44537758497679**

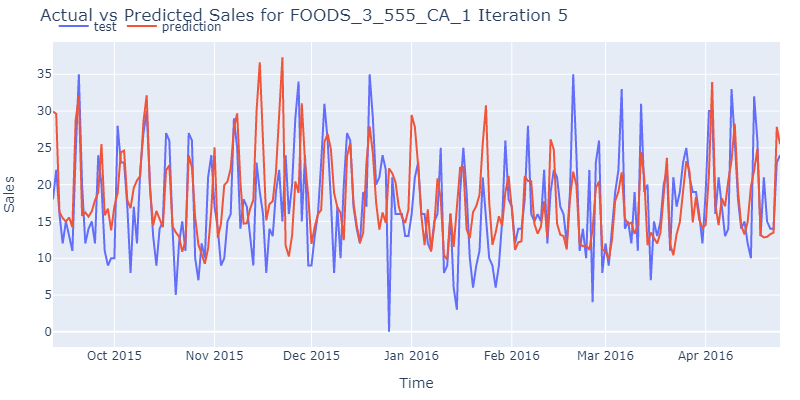
****

Figure 27: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation

# Chapter 4: Light Gradient Boosting Machine on FOODS\_3\_555\_CA\_1\_validation for predicting the next 28 days

## Iteration 1:

**Model development:**

In this iteration, we are trying to replicate the competition’s objective of forecasting for the next 28 days ahead. The first 1885 data points were utilised for training our model and the remaining 28 data points were used for testing or forecast. The significant change in this iteration is the backtesting forecaster parameters. Here backtesting with refit and increasing training size was implemented which was obtained by changing the following parameters:

refit=True*and*fixed\_train\_size=False (SKforecast, 2024).

The LGBM model parameters: learning\_rate = 0.1, max\_depth= 6, n\_estimators =500, num\_leaves=64, objective = ‘tweedie’, random\_state = 123, force\_row\_wise = True

**Feature engineering:** lags [1, 6, 7], day, year, month, snap\_CA, snap\_TX, snap\_WI, y\_window\_7\_mean, y\_window\_7\_std, y\_window\_14\_mean, y\_window\_14\_std, y\_window\_28\_mean, y\_window\_28\_std. std stands for standard deviation.

Number of observations used for initial training: 1885

Number of observations used for backtesting: 28

Number of folds: 3

Number of steps per fold: 10

Number of steps to exclude from the end of each train set before test (gap): 0

Last fold only includes 8 observations.

Fold: 0

Training: 2011-01-29 00:00:00 -- 2016-03-27 00:00:00 (n=1885)

Validation: 2016-03-28 00:00:00 -- 2016-04-06 00:00:00 (n=10)

Fold: 1

Training: 2011-01-29 00:00:00 -- 2016-04-06 00:00:00 (n=1895)

Validation: 2016-04-07 00:00:00 -- 2016-04-16 00:00:00 (n=10)

Fold: 2

Training: 2011-01-29 00:00:00 -- 2016-04-16 00:00:00 (n=1905)

Validation: 2016-04-17 00:00:00 -- 2016-04-24 00:00:00 (n=8)

In Fold 0, the model used 1885 data points to train and predict the next 10 steps in the horizon. In Fold 1, the model incorporated the next 10 data points into training and predicted the next 10 steps on the horizon. As you can see the size of the training set is increasing by 10 at every fold. The model was trained 3 times, where for folds 1 and 2 model incorporated additional 10 data points and was re-trained.

**Model evaluation:** The **mean squared error was 17.82707070314831**

A graph showing a line of sales

Description automatically generated with medium confidence

Figure 28: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation

## Iteration 2:

In the second iteration, the whole process was the same except the two model parameters were changed which were max\_depth=5, num\_leaves = 32

**Feature engineering:** lags [1, 6, 7], day, year, month, snap\_CA, snap\_TX, snap\_WI, y\_window\_7\_mean, y\_window\_7\_std, y\_window\_14\_mean, y\_window\_14\_std, y\_window\_28\_mean, y\_window\_28\_std. std stands for standard deviation. Same features as iteration 1 were incorporated.

Model evaluation: The **mean squared error was 20.371672848336384**

**A graph of a sales report

Description automatically generated with medium confidence**

Figure 29: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation

## Iteration 3:

In the third iteration, the whole process was the same except the 2 model parameters were changed which were max\_depth=4, num\_leaves = 16

Feature engineering: lags [1, 6, 7], day, year, month, snap\_CA, snap\_TX, snap\_WI, y\_window\_7\_mean, y\_window\_7\_std, y\_window\_14\_mean, y\_window\_14\_std, y\_window\_28\_mean, y\_window\_28\_std. std stands for standard deviation. Same features as iteration 1 were incorporated.

Model evaluation: **The mean squared error was 22.092868504350104**

**A graph showing a line of sales

Description automatically generated with medium confidence**

Figure 30: Figure illustrating the difference between the predictions and real values on the product FOODS\_3\_555\_CA\_1\_validation

The major difference between the second chapter and the third chapter was that in the third chapter we were able to model the time series to some extent. As observed from the graphs of actual value and predicted value in iteration 1 and iteration 4, it can be observed that the model can capture peak successfully to some extent but fails to capture the throughs accurately. Furthermore, the Autocorrelation and Partial Autocorrelation functions enabled us to gain more information regarding useful lags which can be incorporated as features in our model. Thus, when engaging in time-series data, autocorrelation and partial autocorrelation can be one of the first techniques from which we might be able to extract significant lags which can potentially aid in the forecast performance.

# Chapter 5: Forecasting sales across various levels of hierarchies

## Iteration 1:

The objective of this chapter was to create forecasts at different levels of hierarchies such as departments, categories, and store level.

The dataset was filtered from the original dataset to improve performance. Only the data for store CA\_1 based in California was filtered.

There are three important functions which achieve different things.

The **forecast\_sku function** for each unique 3049 products fits the LGBMRegressor and returns the predictions for the next 28 days. The lags incorporated into LGBMRegressor were [1, 2, 3, 4, 5, 6, 7, 14, 21, 28].

The **aggregate\_forecast** function and **aggregate\_level** function aggregates the predictions generated by the **forecast\_sku function** at department, category and store levels.

Error metrics such as mean absolute error, mean squared error and root mean squared error were calculated across different department, category and store levels.

A white screen shot of a computer

Description automatically generated

Figure 31: Figure illustrating different error metrics across departments, category and store levels

The figure highlights the errors across departments, category and store levels.

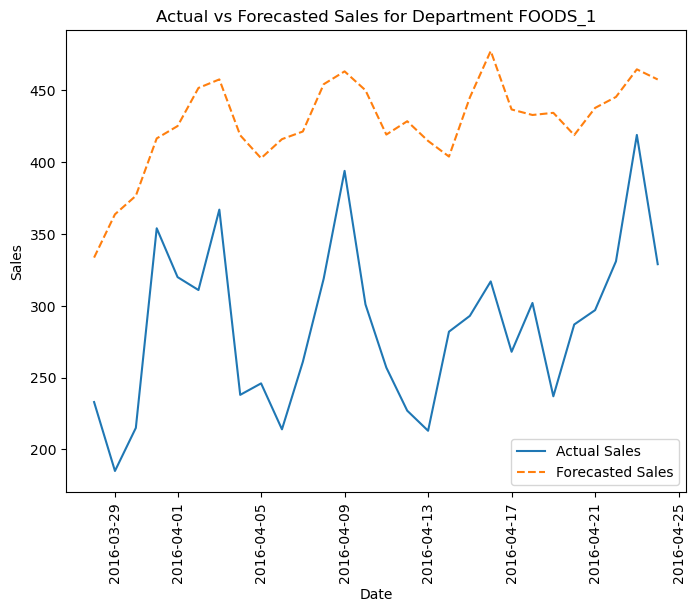


Figure 32: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_1 department

:

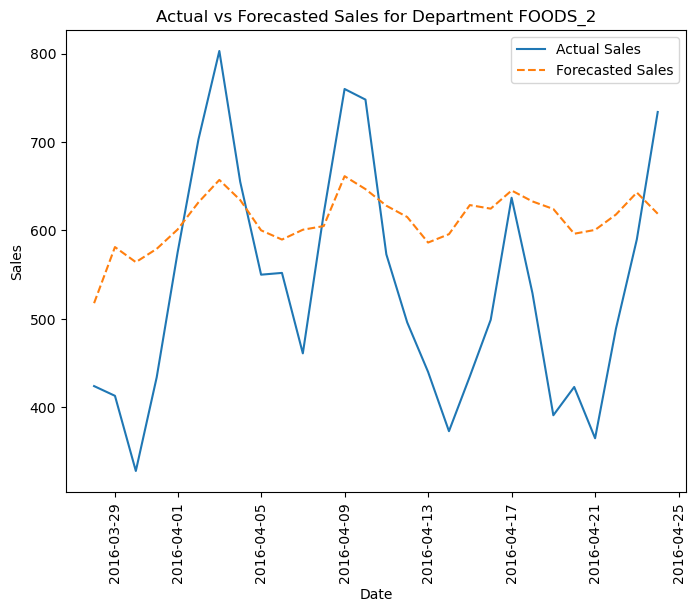


Figure 33: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_2 department

*:*

A graph with blue and orange lines

Description automatically generated

Figure 34: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_3 department

A graph of sales

Description automatically generated

Figure 35: Figure illustrating the difference between the actual sales and predicted sales for HOBBIES\_1 department

*:*

A graph with blue lines and orange lines

Description automatically generated

Figure 36: Figure illustrating the difference between the actual sales and predicted sales for HOBBIES\_2 department

A graph of sales and forecasted sales

Description automatically generated

Figure 37: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD\_1 department

A graph of sales

Description automatically generated

Figure 38: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD\_2 department

A graph of sales and forecasted sales

Description automatically generated

Figure 39: Figure illustrating the difference between the actual sales and predicted sales for FOODS Category

A graph with blue lines and orange lines

Description automatically generated

Figure 40: Illustrating the difference between the actual sales and predicted sales for HOBBIES Category

A graph with blue lines and orange lines

Description automatically generated

Figure 41: Illustrating the difference between the actual sales and predicted sales for Household Category

A graph of sales

Description automatically generated

Figure 42: Figure illustrating different error metrics across departments, category and store levels

From the above figure, it can be observed that for store CA\_1, Household category, foods category, Household\_01 department and FOODS\_3 department are successfully able to capture the seasonal peaks but lacks the proper capturing of throughs. Categories and departments other than these perform very poorly. Thus, more model tuning, feature engineering, and domain specific knowledge is required to improve the forecast accuracy.

## Iteration 2:

In the second iteration, we made some changes to forecast\_sku function in that we enabled the creation of datetime, rolling window, rolling standard deviation features for each product id which was being forecasted. Secondly, in the forecast.fit () method, in addition to passing the y\_train values, we are also passing the created datetime and rolling window features as input to the exog parameter (exogenous variables). This has been done to add more features to our model which in turn can help in the forecast performance.

Furthermore, while using the forecaster.predict() method to forecast the next 28 days, we are passing the datetime and rolling window features as input to the exog parameter as they should be provided in the prediction stage. This creates a forecast environment where given the following features, the model must predict the corresponding sales.

**Feature engineering:** ‘month’, ‘day’, ‘year’, ‘y\_window\_7\_mean’, ‘y\_window\_7\_std’, ‘y\_window\_14\_mean’, ‘y\_window\_14\_std’, lags[1, 2, 3, 4, 5, 6, 7, 14, 21, 28]

**Model evaluation:**

A screenshot of a computer

Description automatically generated

Figure 43: Figure illustrating different error metrics across departments, category and store levels

A graph with blue lines and orange lines

Description automatically generated

Figure 44: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_1 department

A graph of sales and forecasted sales

Description automatically generated

Figure 45: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_2 department

A graph with blue and orange lines

Description automatically generated

Figure 46: Figure illustrating the difference between the actual sales and predicted sales for FOODS\_3 department

A graph of sales and forecasting

Description automatically generated

Figure 47: Figure illustrating the difference between the actual sales and predicted sales for HOOBIES\_1 department

A graph with blue lines and orange lines

Description automatically generated

Figure 48: Figure illustrating the difference between the actual sales and predicted sales for HOOBIES\_2 department

A graph of sales and forecasted sales

Description automatically generated

Figure 49: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD\_1 department

A graph of sales

Description automatically generated

Figure 50: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD\_2 department

A graph with blue and orange lines

Description automatically generated

Figure 51: Figure illustrating the difference between the actual sales and predicted sales for FOODS category

A graph of sales

Description automatically generated

Figure 52: Figure illustrating the difference between the actual sales and predicted sales for HOOBIES category

A graph of sales

Description automatically generated

Figure 53: Figure illustrating the difference between the actual sales and predicted sales for HOUSEHOLD category

A graph with blue and orange lines

Description automatically generated

Figure 54: Figure illustrating the difference between the actual sales and predicted sales for Store CA\_1 based in California

# Results

|  |  |
| --- | --- |
| **LightGBM implemented Product ids** | **Metrics**  **MSE (mean-squared error)** |
| HOBBIES\_1\_018\_CA\_1\_validation\_Iteration\_1  226 days ahead forecast | MSE: 0.12077583661944795 |
| HOBBIES\_1\_018\_CA\_1\_validation\_Iteration\_2  226 days ahead forecast | MSE: 0.009742097226445246 |
| FOODS\_3\_555\_CA\_1\_validation\_Iteration\_1  226 days ahead forecast | MSE: 31.163215312952573 |
| FOODS\_3\_555\_CA\_1\_validation\_Iteration\_2  226 days ahead forecast | MSE: 35.7947901394842 |
| FOODS\_3\_555\_CA\_1\_validation\_Iteration\_3  226 days ahead forecast | MSE: 37.549325358479585 |
| FOODS\_3\_555\_CA\_1\_validation\_Iteration\_4  226 days ahead forecast | MSE: 31.664441037541923 |
| FOODS\_3\_555\_CA\_1\_validation\_Iteration\_5  226 days ahead forecast | MSE: 35.44537758497679 |
| FOODS\_3\_555\_CA\_1\_validation\_Iteration\_1  28 days ahead forecast | MSE: 17.82707070314831 |
| FOODS\_3\_555\_CA\_1\_validation\_Iteration\_2  28 days ahead forecast | MSE: 20.371672848336384 |
| FOODS\_3\_555\_CA\_1\_validation\_Iteration\_3  28 days ahead forecast | MSE: 22.092868504350104 |

A white screen shot of a computer

Description automatically generated

Figure 55: Figure illustrating different error metrics across departments, category and store levels for Chapter 5, Iteration 1

A screenshot of a computer

Description automatically generated

Figure 56: Figure illustrating different error metrics across departments, category and store levels for Chapter 5, Iteration 2

When it comes to the HOBBIES\_1\_018\_CA\_1\_validation in Chapter 2, the LightGBM model was unable to capture relationships and patterns in the sales data. One of the reasons is that the sales data contained many zero values. Even though, the error metrics seems low, the chosen error metric which is the mean-squared error is not the appropriate metric when dealing with data that has many zero values.

However, when we turn our attention to continuous sales data in chapter 3 and 4 where the mean-squared error seems like a reliable metric of choice. When comparing FOODS\_3\_555\_CA\_1\_validation across the first 5 iterations it can be observed that Iteration 1 has the lowest error and thus the highest forecast accuracy among the 5 iterations. Thus, the features incorporated in Iteration 1 do a better job of forecasting sales for the next 226 days. Additionally, Iteration 4 also has mean-squared error like that of iteration 1, thus the features incorporated during iteration 4 can also be helpful.

In chapter 4, when comparing FOODS\_3\_555\_CA\_1\_validation accuracy for forecasting the next 28 days, iteration 1 does a better job as compared to the other two iterations. During the second and third iteration two model hyperparameters namely max\_depth and num\_leaves of Light GBM model were reduced to assess if there was a change in forecast performance keeping all the input data features to the model same across three iterations. It was observed that the forecast performance drop by 14.23% from iteration 1 to iteration 2 and 23.93% from iteration 1 to iteration 3. Thus, decreasing the max\_depth and num\_leaves parameter of the LightGBM model is leading to a decreased forecast accuracy.

When comparing the forecast performance across higher levels of hierarchies in Chapter 5, significant improvements can be observed in certain departments, categories and at the store level. Here, we will be using the root-mean squared error metric to discuss our forecast performance. The major difference between Iteration and Iteration 2 is the feature engineering part. In Iteration 1, only lag features [1, 2, 3, 4, 5, 6, 7, 14, 21, 28] were incorporated into our model. Whereas in Iteration 2, in addition to lags[1, 2, 3, 4, 5, 6, 7, 14, 21, 28], features such as ‘month’, ‘day’, ‘year’, ‘y\_window\_7\_mean’, ‘y\_window\_7\_std’, ‘y\_window\_14\_mean’, ‘y\_window\_14\_std’ were incorporated into our model to assess if adding more features improved forecast performance.

The model parameters to the LightGBM model were kept constant across both iterations. When comparing the Store\_CA\_1 root-mean squared error, there was a 65% reduction in error from Iteration 1 and Iteration 2. Thus a 65% reduction in error translates to a 65% improvement in forecast accuracy. Similarly, For Household category, there was more than a 100% reduction in root-mean squared error from Iteration 1 to Iteration 2 which translates to more than 100% improvement in forecast accuracy. The same improvement can be seen across all departments and categories. Thus, incorporating useful features into our model can lead to significant improvement in forecast performance which answers the question that feature engineering and feature selection can improve the performance of LightGBM model. In Iteration 2, when observing the figures for Store CA\_1 (figure 54), category Household (figure 53), category FOODS (figure 51), department Household 2 (figure 50) , department Household 1 ( figure 49) and department FOODS 3 (figure 46) , it appears that the LightGBM was able to captures patterns successfully.

However, upon observing figures of category HOBBIES (figure 52), department HOBBIES\_1 (figure 47), HOBBIES\_2 (figure 48) and FOODS\_1 (figure 44) , it can be seen that the LightGBM model is performing poorly and thus more work needs to be done for these departments and categories.

# Conclusion

Forecasting being a complex task, the initial problem of fitting the data and understanding the sequence of steps required to generate appropriate forecasts was a challenging feat. Understanding different moving parts such as loading the data, data-preprocessing, feature engineering, model development and evaluation using python and skforecast library required understanding of the documentation and code.

This project’s main aim was to simulate the data science process or forecasting process of loading the data, data-preprocessing, feature engineering, model development and evaluation. The project was successfully able to deliver on all the areas. In data-preprocessing, different operations such as merging the sales data and calendar data, reducing unnecessary memory of the features to improve computational performance were carried out.

The feature engineering part is where majority of time and effort was spent to understand and incorporate different time-series features such as lags, rolling windows and date-time. The different iterations in Chapter 3 and Chapter 5 were performed with the objective of assessing whether incorporating different features and their combinations could help in improving forecast performance. Of the two chapters, Chapter 5 was successfully able to highlight the importance of feature engineering and selection in improving forecast performance.

The project was able to apply LightGBM model successfully but more mathematical and algorithmic understanding of LightGBM model and its parameters is required to enhance the forecast performance. This need can be further highlighted when observing figures 44, 47, 48, and 52 where the LightGBM model was unable to capture the trends in the data.

Furthermore, it is important to highlight the contribution the skforecast library had in this project. The library had easy to use functionalities for feature creation, model development and model evaluation which made the data science process much easier to handle.

Overall, the project was able to achieve all its objectives and was able to deliver on Exploratory Data Analysis in Chapter 1, creating a data science/ forecasting process in Chapter 2, 3, 4, 5 and experimenting with different combinations of feature engineering and feature selection process.

The main contribution of this project is that it highlighted the importance of feature engineering and feature selection in a data science/ machine learning/ forecasting process which can be difference between low quality and high-quality models. Thus, its important to take time to understand the underlying data and derive meaningful features from it to improve forecast performance.

Some of the limitations of this project was the lack of baseline model where the predictions of the baseline model and LightGBM model could be compared to assess if LightGBM is worth implementing. Furthermore, the complexity of the forecasting problem was toned down to focus only on certain products and one hierarchy due to limitations in technical expertise and knowledge. But this limited focus has enabled to understand the forecasting process much better which can now be scaled to forecast for different hierarchy levels. The need to focus on certain products and hierarchy was also due to limited computational resources.

The future work involves applying the knowledge gained to an upcoming forecasting competition, the VN1 Forecasting Accuracy Challenge. In this competition, participants are tasked with accurately forecasting future sales using historical sales, inventory, and pricing data. The goal is to develop robust predictive models that can anticipate sales trends for various products across different clients and warehouses.

# References

Barocas, S., Hardt, M. and Narayanan, A., 2019. Fairness and Machine Learning. fairmlbook.org.

Bontempi, G., Taieb, S.B. and Le Borgne, Y.-A., 2013. Machine learning strategies for time series forecasting. In European Business Intelligence Summer School (pp. 62-77). Springer.

Box, G.E., Jenkins, G.M., Reinsel, G.C. and Ljung, G.M., 2015. Time series analysis: forecasting and control. John Wiley & Sons.

Breiman, L., 1997. *Arcing the edge* (pp. 1-14). Technical Report 486, Statistics Department, University of California at Berkeley.

Brockwell, P.J. and Davis, R.A., 2016. Introduction to Time Series and Forecasting. 3rd ed. New York: Springer.

Burges, C.J., 2010. From ranknet to lambdarank to lambdamart: An overview. Learning, 11(23-581), p.81.

Chase, C.W., 2013. Demand-driven forecasting: A structured approach to forecasting. John Wiley & Sons.

Chen, K.K., Yu, H. and Shih, C., 2020. Inventory management and machine learning: A review. Elsevier.

Chopra, S. and Meindl, P., 2016. Supply Chain Management: Strategy, Planning, and Operation. Pearson.

Cleveland, W.S., 1979. Robust Locally Weighted Regression and Smoothing Scatterplots. Journal of the American Statistical Association, 74(368), pp.829-836.

Dietterich, T.G., 1995. Overfitting and undercomputing in machine learning. ACM Computing Surveys (CSUR), 27(3), pp.326-327.

Dietterich, T.G., 2000. Ensemble methods in machine learning. In International workshop on multiple classifier systems (pp. 1-15). Springer.

Doshi-Velez, F. and Kim, B., 2017. Towards a rigorous science of interpretable machine learning. arXiv preprint arXiv:1702.08608.

Fildes, R., Goodwin, P. and Lawrence, M., 2006. The design, selection and implementation of forecasting support systems. Decision Support Systems, 42(1), pp.351-361.

Freund, Y. and Schapire, R.E., 1997. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, *55*(1), pp.119-139.

Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pp.1189-1232.

Geeks for Geeks (2024) *AutoCorrelation*. Available at: https://www.geeksforgeeks.org/autocorrelation/ (Accessed: 13/07/2024).

GeeksforGeeks, 2024. AutoCorrelation. [online] GeeksforGeeks. Available at: <https://www.geeksforgeeks.org/autocorrelation/> [Accessed 31 July 2024].

Geron, A. (2019) *Hands-on Machine Learning with Scikit-Learn, Keras and Tensorflow*. 2nd Edition. USA: Oreily Media

Goodfellow, I., Bengio, Y. and Courville, A., 2016. Deep Learning. MIT Press.

Gordon, J., 2023. Practical Guide for Feature Engineering of Time Series Data. [online] dotData. Available at: <https://dotdata.com/blog/practical-guide-for-feature-engineering-of-time-series-data/> [Accessed 31 July 2024].

Hastie, T., Tibshirani, R. and Friedman, J., 2009. The elements of statistical learning: data mining, inference, and prediction. Springer Science & Business Media.

Hyndman, R.J., & Athanasopoulos, G. (2021) *Forecasting: principles and practice*, 3rd edition, OTexts: Melbourne, Australia. OTexts.com/fpp3. Accessed on: 13/07/2024.

Jim, S., no date. Autocorrelation and Partial Autocorrelation in Time Series Data. [online] Statistics By Jim. Available at: <https://statisticsbyjim.com/time-series/autocorrelation-partial-autocorrelation/> [Accessed 31 July 2024].

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q. and Liu, T.Y., 2017. Lightgbm: A highly efficient gradient boosting decision tree. Advances in neural information processing systems, 30.

Li, P., 2012. Robust logitboost and adaptive base class (abc) logitboost. arXiv preprint arXiv:1203.3491.

Machine Learning Mastery (2020) *A Gentle Introduction to Autocorrelation and Partial Autocorrelation* Available at: https://machinelearningmastery.com/gentle-introduction-autocorrelation-partial-autocorrelation/ (Accessed: 13/07/2024).

Makridakis, S., Spiliotis, E. and Assimakopoulos, V., 2022. M5 accuracy competition: Results, findings, and conclusions. *International Journal of Forecasting*, *38*(4), pp.1346-1364.

Oreshkin, B.N., Carpov, D., Chapados, N. and Bengio, Y., 2019. N-BEATS: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*.

Pennsylvania State University (n.d.). *Autocorrelation and Time Series Methods*. Available at: https://online.stat.psu.edu/stat462/node/188/ (Accessed: August 5, 2024).

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M.Z., Barrow, D.K., Taieb, S.B., Bergmeir, C., Bessa, R.J., Bijak, J., Boylan, J.E. and Browell, J., 2022. Forecasting: theory and practice. *International Journal of Forecasting*, *38*(3), pp.705-871.

Richardson, M., Dominowska, E. and Ragno, R., 2007, May. Predicting clicks: estimating the click-through rate for new ads. In Proceedings of the 16th international conference on World Wide Web (pp. 521-530).

Shumway, R.H. and Stoffer, D.S., 2017. Time Series Analysis and Its Applications: With R Examples. 4th ed. New York: Springer.

Skforecast, 2024. Backtesting. [online] Available at:<https://skforecast.org/> [Accessed 22 July 2024].

Skforecast, 2024. Introduction to forecasting. [online] Available at:<https://skforecast.org/> [Accessed 22 July 2024].

Skforecast, 2024. Recursive multi-step forecasting. [online] Available at:<https://skforecast.org/> [Accessed 22 July 2024].

SKforecast (2024) Backtesting. Available at: https://skforecast.org/0.13.0/user\_guides/backtesting.html (Accessed: 31 July 2024).

USDA . Supplemental Nutrition Assistance Program (SNAP). [online] Available at: <https://www.fns.usda.gov/snap/supplemental-nutrition-assistance-program> [Accessed 31 July 2024].

Brownlee, J. (2018) *Machine learning mastery*. Available at: <https://machinelearningmastery.com/> (Accessed: 17 August 2024).

# Appendix

GitHub repository: <https://github.com/SahiltheRed/Masters_Project_21078559.git>

Skforecast library: <https://skforecast.org/0.12.1/index.html>

M5 Competition data: <https://www.kaggle.com/competitions/m5-forecasting-accuracy/data>

Time series: <https://www.geeksforgeeks.org/time-series-analysis-and-forecasting/?ref=header_outind>

Kaggle notebook reference: <https://www.kaggle.com/code/konradb/ts-4-sales-and-demand-forecasting>

Train in data (course undertaken to study time-series forecasting): <https://www.trainindata.com/>