FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO



Comparing the effect of concurrent promotions over demand with interpretable deep learning

Micael Pires Gomes

Mestrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Luís Paulo Gonçalves dos Reis

June 26, 2023

© Micael Pires Gomes, 2023

Abstract

Accurately predicting customer demand is challenging but crucial for businesses and the environment. Numerous factors can impact demand, and promotions are of particular interest as they are known to alter customer behavior. Often, multiple promotional events occur at the same time, making it difficult to assess the individual impact of each one. However, understanding the effect of individual promotions on demand can help retail business managers to make more informed decisions.

Generally, estimates of customer demand are based on time series (historical data), which requires specialized analysis techniques ranging from pre-processing to forecasting and evaluation. To overcome these difficulties, this study introduces time series and forecasting methods. It discusses the use of interpretable deep learning models for demand forecasting and suitable assessment measures, with the intent of better understanding the efficacy of various marketing techniques.

Traditional time series forecasting approaches and machine learning techniques face several limitations when trying to capture these effects adequately. Some are hard to interpret, making comprehending how these make forecasts challenging. Furthermore, adequate evaluation metrics are critical for measuring the correctness of the deployed model and ensuring that the findings are accurately understood. Due to the usage of several series on widely diverse scales, scale-free measures are frequently used in retail forecasting.

To the better of our knowledge, there is a need to extend research integrating the review of various promotions with interpretable deep learning models in the current state-of-the-art, which is necessary for building future business decisions, and marketing activities and evaluating the influence of different promotional events on customer behaviour. These interpretable deep learning models can increase transparency in the demand forecasting process, providing a better grasp of the underlying mechanisms at work and allowing for more informed decision-making.

Hence, this research aims to improve demand forecasting accuracy in the presence of promotions and gain a deeper understanding of the influence of different marketing techniques on customer behaviour by considering all these elements. To this end, deep learning algorithms will be applied and compared to a database containing promotional information, with their performance assessed in accuracy and interpretability. Additionally, the most critical factors in predicting the effect of concurrent promotions on demand will be identified.

SARIMAX is used as the benchmark, while Long Short-Term Memory (LSTM) models and XGBoost are trained and tested, taking into account available promotional information. Subsequently, SHAP is employed to interpret the predictions obtained by these models. Furthermore, Temporal Fusion Transformer (TFT), an interpretable deep learning model, is also utilized to compare its performance against SARIMAX, LSTM, and XGBoost. Python will be employed for its ease of programming and machine learning capabilities, leveraging various libraries for data processing and analysis.

Retail Consult provided the data used for this study, and it is obtained from a real-world retailer whose information cannot be shared due to confidentiality purposes. This data encompasses approximately two years' worth of sales records. It includes the daily unitary sales for each item available at all the retailer's stores, along with the details of each promotional event for each item. After preprocessing steps and data cleaning, the proposed models were trained and tested on the available data using a sliding window method of 7 days of historical data employed on the proposed models to predict the sales for the following day. This was done to evaluate the forecasting accuracy of each of these models using metrics such as the Mean Absolute Error (MAE), Mean Absolute Scaled Error (MASE), and Root Mean Square Error (RMSE).

Assessing the models was accomplished through their predictive accuracy, using metrics like MAE, RMSE, and MASE. The TFT model repeatedly surpassed SARIMAX, XGBoost, and LSTM when tested against MAE and MASE measures, recording average values of 1.83 and 0.71, respectively. Nonetheless, the LSTM model took the lead in the RMSE measure, with mean and median values of 2.49 and 1.77, respectively. This comprehensive appraisal points towards the TFT model being the most trustworthy and precise in forecasting, offering an essential tool for efficient retail inventory management and increased profit margins. In order to gain a deeper understanding of the factors that influenced the target variable for each model, an interpretability analysis was conducted.

Hence, this analysis was performed at three different levels. The first breakdown was conducted globally, considering the all-time series from all stores. While this provided a general overview of each model, it didn't offer specific insights into the effects of promotions, particularly at an individual level. The key findings suggest that promotions are essential determinants of unit sales for all models under consideration.

Furthermore, the second analysis was conducted at a category level, which can be beneficial as items within the same category often exhibit similar characteristics and behaviours. This allowed for a more detailed understanding of the features most significantly impacted sales within each category. Regarding the Dairy and Deli categories, various promotions, including circular, discount, front-page everyday low-price, and buy-one-get-one, substantially influence sales. In contrast, the Frozen and General Merchandise categories underscore the importance of circular and discount promotions as crucial sales drivers.

Also, the Grocery category reveals uneven behaviour of discount and circular promotions, with different results emerging depending on the feature values in the XGBoost model. LSTM suggests that higher forecasted unit sales are detected when discounts are inactive, indicating that this promotional activity may not particularly affect sales in this category. The inactivity of circular promotions results in negative SHAP values, indicating a potential adverse impact.

In the Health, Beauty, and Care category, discount and circular promotions are the most influential, while buy-one-get-one advertisements tend to reduce the model's predictions. For the Meat category, front-page discounts positively affect predictions, with less significant but noticeable influence from circular, front-page everyday low-price, and buy-one-get-one promotions. The Produce category, however, exhibits erratic responses to promotional activities.

However, the categories in the given dataset were still too general, and the results could be further refined by considering a more restricted category description. The final analysis was conducted at an item level, considering just one item at one store. This approach yielded promising results as it allowed us to identify which promotions were strategically beneficial in boosting unit sales and which were not impactful or even negatively affected sales. This item-level analysis was conducted on the stores' three most heavily promoted items. Considering the item-level analysis, buy-one-get-one promotions negatively impact model predictions for the top promoted item, while circular and front-page everyday low-price promotions have a positive effect, especially the

former. The second most advertised item follows a similar trend, with the inefficacy of buy-one-get-one and discount promotions being counterbalanced by the positive influence of front-page everyday low-price and circular everyday low-price promotions. For the third most promoted item, discount promotions show inconsistent patterns, while front-page, circular, and circular everyday low-price promotions positively influence sales.

These findings provide clear insights to inform the development of more effective and targeted promotional strategies. Despite the variance in how different models perceive promotional activities, promotions play a substantial role in sales forecasting. The insights can offer a pathway toward improved sales performance and enhanced profitability.

Keywords: Promotions, Deep Learning, Interpretability, Time Series Forecasting

Resumo

Prever com precisão a procura dos clientes é um desafio crucial para as empresas e o meio ambiente. Inúmeros fatores podem impactar a procura, e as promoções são de particular interesse, pois são conhecidas por alterar o comportamento do cliente. Frequentemente, vários eventos promocionais ocorrem ao mesmo tempo, tornando difícil avaliar o impacto individual de cada um. No entanto, compreender o efeito individual das promoções na procura pode ajudar os gerentes de negócios a tomar decisões mais informadas.

Geralmente, as estimativas de procura do cliente são baseadas em séries temporais (dados históricos), o que requer técnicas especializadas de análise que vão desde o pré-processamento até à previsão e avaliação. Para superar essas dificuldades, este estudo introduz métodos de séries temporais e previsão. Discute-se o uso de modelos de deep learning interpretáveis para previsão da demanda e medidas de avaliação adequadas, com o objetivo de entender melhor a eficácia de várias técnicas de marketing.

As abordagens tradicionais de previsão de séries temporais e técnicas de machine learning enfrentam várias limitações ao tentar capturar esses efeitos adequadamente. Algumas são difíceis de interpretar, tornando desafiador compreender as previsões geradas. Além disso, as métricas de avaliação adequadas são críticas para medir a correção do modelo implementado e garantir que os resultados sejam compreendidos com precisão. Devido ao uso de várias séries em escalas amplamente diversas, medidas livres de escala são frequentemente usadas em previsões ligadas ao retalho.

Para o melhor do nosso conhecimento, há uma necessidade de ampliar a pesquisa integrando a revisão de várias promoções com modelos de deep learning interpretáveis no estado atual da arte, o que é necessário para construir decisões empresariais futuras e atividades de marketing e avaliar a influência de diferentes eventos promocionais no comportamento do cliente. Estes modelos podem aumentar a transparência no processo de previsão da procura, fornecendo uma melhor compreensão dos mecanismos subjacentes em funcionamento e permitindo uma tomada de decisão mais informada.

Assim, esta pesquisa tem como objetivo melhorar a precisão da previsão da procura na presença de promoções e obter uma compreensão mais profunda da influência de diferentes técnicas de marketing no comportamento do cliente, considerando todos estes elementos. Para tal, algoritmos de deep learning serão aplicados e comparados a bases de dados relevantes contendo informações promocionais, com o desempenho avaliado em precisão e interpretabilidade. Além disso, os fatores mais críticos na previsão do efeito de promoções simultâneas na procura serão identificados.

O SARIMAX é utilizado como referência, enquanto os modelos Long Short-Term Memory (LSTM) e XGBoost são treinados e testados, tendo em conta as informações promocionais disponíveis. Posteriormente, o SHAP é utilizado para interpretar as previsões obtidas por esses modelos. Além disso, o Temporal Fusion Transformer (TFT), um modelo de deep learning interpretável, é também utilizado para comparar o seu desempenho com o SARIMAX, LSTM e

XGBoost. O Python será utilizado pela sua facilidade de programação e capacidades de aprendizagem automática, tirando partido de várias bibliotecas para processamento e análise de dados.

A Retail Consult forneceu os dados usados para este estudo, sendo obtidos de um retalhista real cujas informações não podem ser partilhadas por motivos de confidencialidade. Estes dados abrangem aproximadamente dois anos de registos de vendas. Incluem as vendas unitárias diárias para cada item disponível em todas as lojas do retalhista, juntamente com os detalhes de cada evento promocional para cada item.

Após as etapas de pré-processamento e limpeza de dados, os modelos propostos foram treinados e testados nos dados disponíveis usando um método de janela deslizante de 7 dias de dados históricos nos modelos propostos para prever as vendas do dia seguinte. Isso foi feito para avaliar a precisão das previsões de cada um desses modelos usando métricas como o Erro Absoluto Médio (MAE), o Erro Absoluto Médio Escalado (MASE) e o Erro Quadrado Médio Raiz (RMSE).

A avaliação dos modelos foi realizada através da sua precisão de previsão, utilizando métricas como MAE, RMSE e MASE. O modelo TFT superou repetidamente o SARIMAX, XGBoost e LSTM quando testado contra as medidas MAE e MASE, registando valores médios de 1,83 e 0,71, respectivamente. No entanto, o modelo LSTM liderou na medida RMSE, mostrando valores médios e medianos de 2,49 e 1,77, respectivamente. Esta avaliação abrangente aponta para o modelo TFT como sendo o mais confiável e preciso nas previsões, oferecendo uma ferramenta essencial para uma gestão eficiente do inventário de retalho e aumento das margens de lucro. Para obter uma compreensão mais profunda dos fatores que influenciaram a variável alvo de cada modelo, foi realizada uma análise de interpretabilidade.

Assim, esta análise foi realizada em três diferentes níveis. A primeira análise foi conduzida globalmente, considerando todas as séries temporais de todas as lojas. Embora isto tenha fornecido uma visão geral de cada modelo, não ofereceu informações específicas sobre os efeitos das promoções, particularmente ao nível individual. As principais descobertas sugerem que as promoções são determinantes essenciais das vendas unitárias para todos os modelos em consideração.

A segunda análise foi realizada ao nível da categoria, o que pode ser benéfico, pois os itens dentro da mesma categoria muitas vezes exibem características e comportamentos semelhantes. Isto permitiu uma compreensão mais detalhada das características que mais significativamente impactaram as vendas dentro de cada categoria. No que diz respeito às categorias de Dairy e Deli, várias promoções, incluindo circular, desconto, primeira página de preços baixos todos os dias, e compre um leve dois, influenciam substancialmente as vendas. Em contraste, as categorias Frozen e General Merchandise destacam a importância das promoções circulares e de desconto como impulsionadores cruciais das vendas.

Além disso, a categoria Grocery revela um comportamento irregular das promoções de desconto e circular, com diferentes resultados a surgir dependendo dos valores das características no modelo XGBoost. O modelo LSTM sugere que as vendas unitárias previstas mais altas são detetadas quando os descontos estão inativos, indicando que esta atividade promocional pode não afetar particularmente as vendas nesta categoria. A inatividade das promoções circulares resulta em valores SHAP negativos, indicando um potencial impacto adverso.

Na categoria Saúde, Beleza e Cuidados, as promoções de desconto e circulares são as mais influentes, enquanto as publicidades de compre um leve outro tendem a reduzir as previsões do modelo. Para a categoria Meat, os descontos na primeira página afetam positivamente as previsões, com uma influência menos significativa, mas notável, das promoções circulares, da primeira página com preços baixos todos os dias, e de compre um leve outro. A categoria Produce, no entanto, exibe respostas erráticas às atividades promocionais.

No entanto, as categorias no conjunto de dados fornecido eram ainda muito gerais, e os resultados poderiam ser mais refinados considerando uma descrição de categoria mais restrita. A análise

final foi conduzida ao nível do item, considerando apenas um item numa loja. Esta abordagem produziu resultados promissores, pois permitiu-nos identificar quais promoções eram estrategicamente benéficas para aumentar as vendas unitárias e quais não eram impactantes ou até afetavam negativamente as vendas. Esta análise ao nível do item foi realizada nos três itens mais promovidos pelas lojas.

Considerando a análise ao nível do item, as promoções de compre um leve dois impactam negativamente as previsões do modelo para o item mais promovido, enquanto as promoções circulares e da primeira página com preços baixos todos os dias têm um efeito positivo, especialmente a primeira. O segundo item mais anunciado segue uma tendência semelhante, com a ineficácia das promoções de compre um leve dois e de desconto sendo contrabalançada pela influência positiva das promoções da primeira página com preços baixos todos os dias e circulares com preços baixos todos os dias. Para o terceiro item mais promovido, as promoções de desconto mostram padrões inconsistentes, enquanto as promoções da primeira página, circulares, e circulares com preços baixos todos os dias influenciam positivamente as vendas.

Estas descobertas fornecem informações clarass para informar o desenvolvimento de estratégias promocionais mais eficazes e direcionadas. Apesar da variação na forma como diferentes modelos percebem as atividades promocionais, as promoções desempenham um papel substancial na previsão de vendas. Estes insights podem oferecer um caminho para melhorar o desempenho de vendas e aumentar a rentabilidade.

Palavras-chave: Promoções, Deep Learning, Interpretabilidade, Previsão de Séries Temporais

Acknowledgements

This project marks the realization of my academic journey, and I owe immense gratitude to those who have guided me through this process.

Firstly, I express my sincerest appreciation to my Second Supervisor, PhD Alexandra Oliveira, whose unwavering support and guidance have been indispensable from the beginning to the end of this project. My thanks also extend to João Miguel and José Borges from the innovation team. Their constant availability, dedication, and emphasis on teamwork significantly contributed to this endeavour.

I am also thankful to my Supervisor, PhD Luís Paulo Reis, whose insightful guidance was instrumental in developing this project.

Lastly, I dedicate this work to my parents, grandparents, and girlfriend. Their belief in my abilities and unwavering support has always been a source of motivation. I sincerely appreciate their investment in my pursuits.

This project stands as a testament to their collective contributions—my heartfelt thanks to all.

Micael Gomes

This work has been partially supported by the project "IBPS - Intelligent Batch Processing System" , with the reference POCI-01-0247-FEDER-069998, co-financed by the European Regional Development Fund (ERDF), through the Operational Programme for Competitiveness and Internationalization (COMPETE 2020), under the PORTUGAL 2020 Partnership Agreement.

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Motivation . . . 2
- 1.3 Objectives . . . 2
- 1.4 Thesis Contributions . . . 3
- 1.5 Document Structure . . . 4

2 Background **5**
- 2.1 Time series . . . 5
- 2.2 Pre-processing . . . 7
- 2.3 Forecasting . . . 8
  - 2.3.1 Parametric Models . . . 8
  - 2.3.2 Machine Learning Models . . . 12
  - 2.3.3 Deep Learning . . . 15
- 2.4 Interpretability in Machine Learning . . . 21
- 2.5 Demand Influencing Factors . . . 23
- 2.6 Evaluation Metrics . . . 24
  - 2.6.1 Scale Dependent . . . 24
  - 2.6.2 Percentage errors/ Scale independent . . . 25
  - 2.6.3 Relative errors . . . 26
  - 2.6.4 Scaled errors . . . 27

3 State of the Art **29**
- 3.1 Promotion Influence Factors . . . 29
- 3.2 Demand Forecasting methods in the presence of promotions . . . 30

4 Methodology and Experiments Setup **37**
- 4.1 Dataset Overview . . . 37
- 4.2 Dataset Preparation and Cleaning . . . 39
- 4.3 Exploratory Data Analysis . . . 41
- 4.4 Feature Engineering . . . 57
  - 4.4.1 Categorical Variables Encoding . . . 57
  - 4.4.2 Scaling . . . 58
  - 4.4.3 Lags . . . 59
- 4.5 Models and Evaluation . . . 60
  - 4.5.1 SARIMAX . . . 60
  - 4.5.2 XGBoost . . . 60
  - 4.5.3 LSTM . . . 61

4.5.4 Temporal Fusion Transformer 62
4.5.5 Evaluation Metrics 62
4.5.6 Interpretability 63

5 Tests and Results **65**
5.1 Forecasting Accuracy 65
5.2 Interpretability 67
5.2.1 Global 68
5.2.2 Category Level 71
5.2.3 Item Level 98

6 Conclusions **113**

A Additional Literature review **117**

References **121**

List of Figures

2.1 Sliding and Expanding window visual representation . . . 6
2.2 High-level architecture with individual components of TFT model. . . . 18
2.3 N-BEATS architecture concept. . . . 20

4.1 Count of time series grouped into three categories based on their median values. . 41
4.2 Promotional Events . . . 42
4.3 Unit sales per category of products . . . 44
4.4 Time series analysis for item 1260 from store 88 . . . 45
4.5 Time series analysis for item 1860 from store 548 . . . 46
4.6 Unit sales by promotion type . . . 47
4.7 Unit sales by category . . . 47
4.8 Unit sales by an interval of discount . . . 48
4.9 Unit sales by an interval of discount on front-page advertisements . . . 48
4.10 Unit sales by promotion on DELI category . . . 49
4.13 Unit sales by promotion on GENERAL MERCH category . . . 49
4.11 Unit sales by promotion on DAIRY category . . . 50
4.14 Unit sales by promotion on GROCERY category . . . 50
4.15 Unit sales by promotion on HBC category . . . 50
4.12 Unit sales by promotion on FROZEN category . . . 51
4.16 Unit sales by promotion on MEAT category . . . 51
4.17 Unit sales by promotion on PRODUCE category . . . 52
4.18 Time series, Autocorrelation (ACF), and Partial Autocorrelation (PACF) for item 11142 from store 548 . . . 53
4.20 Time series, Autocorrelation (ACF), and Partial Autocorrelation (PACF) for item 57386 from store 6100 . . . 53
4.19 Time series, Autocorrelation (ACF), and Partial Autocorrelation (PACF) for item 47157 from store 548 . . . 54
4.21 Time series, Autocorrelation (ACF), and Partial Autocorrelation (PACF) for item 909701 from store 548 . . . 54
4.22 Entropy . . . 55
4.23 Fourier Entropy . . . 56
4.24 Skewness . . . 56
4.25 Example of One-Hot Encoding to represent promotional activities . . . 58

5.1 Feature importance for XGBoost, considering all time-series . . . 69
5.2 SHAP results for XGBoost, considering all time-series . . . 70
5.3 SHAP results for LSTM, considering all time-series . . . 71
5.4 Interpretability results for TFT, considering all time-series . . . 72

5.5 Feature importance for XGBoost, considering dairy category 73
5.6 SHAP results for XGBoost, considering dairy category 73
5.7 SHAP results for LSTM, considering dairy category 74
5.8 Interpretability results for TFT, considering dairy category 75
5.9 Feature importance for XGBoost, considering deli category 76
5.10 SHAP results for XGBoost, considering deli category 76
5.11 SHAP results for LSTM, considering deli category 77
5.12 Interpretability results for TFT, considering deli category 78
5.13 Feature importance for XGBoost, considering frozen category 79
5.14 SHAP results for XGBoost, considering frozen category 80
5.15 SHAP results for LSTM, considering frozen category 81
5.16 Interpretability results for TFT, considering frozen category 82
5.17 Feature importance for XGBoost, considering general merchandise category . . . 83
5.18 SHAP results for XGBoost, considering general merchandise category 83
5.19 SHAP results for LSTM, considering general merchandise category 84
5.20 Interpretability results for TFT, considering general merchandise category 85
5.21 Feature importance for XGBoost, considering grocery category 86
5.22 SHAP results for XGBoost, considering grocery category 87
5.23 SHAP results for LSTM, considering grocery category 88
5.24 Interpretability results for TFT, considering grocery category 89
5.25 Feature importance for XGBoost, considering HBC category 90
5.26 SHAP results for XGBoost, considering HBC category 90
5.27 SHAP results for LSTM, considering HBC category 91
5.28 Interpretability results for TFT, considering HBC category 92
5.29 Feature importance for XGBoost, considering Meat category 93
5.30 SHAP results for XGBoost, considering HBC category 94
5.31 SHAP results for LSTM, considering Meat category 94
5.32 Interpretability results for TFT, considering Meat category 96
5.33 Feature importance for XGBoost, considering Produce category 97
5.34 SHAP results for XGBoost, considering Produce category 97
5.35 SHAP results for LSTM, considering Produce category 98
5.36 Interpretability results for TFT, considering Produce category 99
5.37 Feature importance for XGBoost, considering the most promoted item 100
5.38 SHAP results for XGBoost, considering the most promoted item 101
5.39 SHAP results for XGBoost, considering the most promoted item 102
5.40 SHAP results for LSTM, considering the most promoted item 103
5.41 SHAP results for LSTM, considering the most promoted item 104
5.42 Interpretability results for TFT, considering the most promoted item 105
5.43 Feature importance for XGBoost, considering the second most promoted item . . 106
5.44 SHAP results for XGBoost, considering the second most promoted item 106
5.45 SHAP results for XGBoost, considering the second most promoted item 107
5.46 SHAP results for LSTM, considering the second most promoted item 107
5.47 Force Plot for multiple predictions for LSTM, considering the second most promoted item 108
5.48 Interpretability results for TFT, considering the second most promoted item . . . 109
5.49 Feature importance for XGBoost, considering the third most promoted item . . . 110
5.50 SHAP results for XGBoost, considering the third most promoted item 110
5.51 SHAP results for XGBoost, considering the third most promoted item 111

5.52 SHAP results for LSTM, considering the third most promoted item 111
5.53 Force Plot for multiple predictions for LSTM, considering the third most promoted item 112
5.54 Interpretability results for TFT, considering the third most promoted item 112

A.1 Additional studies comparing forecasting approaches 118
A.2 Additional studies comparing forecasting approaches 119
A.3 Additional studies comparing forecasting approaches 120

List of Tables

4.1 Summary statistics of selected time series: Median and Interquartile Range (IQR) 40
4.2 Summary statistics of sales data: The dataset consists of approximately 9.76 million observations. On average, unit sales are around 3.36, with a standard deviation of 11.87. The maximum number of unit sales recorded is 1374. 42

5.1 Comparative summary statistics of forecast performance metrics, including MAE, RMSE, and MASE for SARIMAX, XGBoost, LSTM, and TFT. Metrics are shown across various data distribution points: Maximum, Upper Fence, Third Quartile (Q3), Mean, Median, First Quartile (Q1), Lower Fence, and Minimum. 65

Abbreviations

ANN	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
ARIMAX	Autoregressive Integrated Moving Average with eXogenous variables
CoV	Coefficient of Variation
DLR	Direct Linear Regression
ETS	Error, Trend, Seasonality model
ETSX	Error, Trend, Seasonality model with eXogenous variables
FNN	Feedforward Neural Network
FMCG	Fast-Moving Consumer Goods
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
MdAE	Median Absolute Error
MdAPE	Median Absolute Percentage Error
MdASE	Median Absolute Scaled Error
ME	Mean Error
MLP	Multilayer Perceptron
MPE	Mean Percentage Error
MSE	Mean Squared Error
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RF	Random Forest
RMSPE	Root Mean Square Percentage Error
RMSE	Root Mean Squared Error
RMSSE	Root Mean Squared Scaled Error
RNN	Recurrent Neural Network
SARIMA	Seasonal Autoregressive Integrated Moving Average
SARIMAX	Seasonal Autoregressive Integrated Moving Average with eXogenous variables
SHAP	SHapley Additive exPlanations
SKU	Stock Keeping Unit
SVR	Support Vector Regression
SVM	Support Vector Machine
TFT	Temporal Fusion Transformer
XGBoost	eXtreme Gradient Boosting

Chapter 1

Introduction

1.1 Context

In the retail industry, product demand forecasting intends to generate predictions over a given period (forecast horizon) for different products, using a large amount of time-related data known as time series. Accurate forecasts of the products sold in stores are essential for the growth of these retail chains since that stock management and pricing of a product are directly dependent on these demand forecasts [1, 2].

Hence, stock management plays a decisive role since it is necessary to avoid getting the inventory level critically low, stock-outs, and, consequently, loss of sales, or contrarily, overstocking and, thus, high inventory costs [1].

Overstocking can be minimized with advertisements and promotions, which are marketing events that impact consumers' purchases. Advertisement work as a significant feature that can influence customer intention to buy a product. These promotional disclosures can occur in two ways: in-store and out-store advertisements. Thus, we must consider the influence these two advertising forms have on consumers [3].

Promotions can also be of different types, such as buy-one-get-one-free, 50% discount, or free coupons having different outcomes on inventory and future sales. Promotion on one item may decrease its own afterwards sales, the sales of other products, so-called cannibalization, or oppositely, increase the sales of another product- complementarity. Although, modelling these last two effects requires considerable effort when identifying the affected products [3].

Therefore, with the traditional demand forecasting approaches, it is not possible to predict the effect of promotions on the sales of the related products either through complementarity or cannibalization [1]. Moreover, the traditional approaches cannot separate each promotion effect in the presence of multiple promotions. Consequently, regular and price discounts can be integrated into interpretable deep learning to improve their performance and to evaluate the different effects caused by different promotion types. Also, it is essential to increase trust in these methods and ensure that they make proper and unbiased decisions [4].

1.2 Motivation

Accurate retail sales forecasting has a significant impact on retail operations management. It helps businesses to make more informed decisions about inventory management, pricing, and promotions, which may lead to increased profitability and consumer engagement.

However, retail sales time series usually display strong trends and seasonal variations, making it challenging to build appropriate forecasting models. Because of the short life spans of products and the increasing quantity of marketing activities such as promotions, forecasting sales at the Stock Keeping Unit (SKU) level in a given shop is complex. Furthermore, there has been a lack of information on how promotions impact demand volatility and how they might be included in forecasting models to enhance supply chain performance [5, 6].

Besides that, time series for daily perishable food sales at retail stores exhibit significant volatility and skewness, which can be influenced by external variables such as day-of-week and/or month-of-year seasonality, holidays, discount offers, and weather [5, 6].

On the other hand, traditional demand forecasting methodologies have limitations, such as the inability to adequately assess the impact of promotions on sales or determine which goods may be influenced by promotion through cannibalization or complementarity.

As a result, employing interpretable deep learning algorithms to overcome these constraints and accurately compare the effects of various promotional events on demand is expected to assist retailers in making better-informed promotional strategy decisions while avoiding the costly consequences of overstocking or understocking products. Interpretability is an essential component of machine learning because it may assist retailers in understanding the characteristics driving their sales and making informed decisions based on this understanding.

Aside from practical benefits for retailers, the effort seeks to improve the science of demand forecasting by encouraging the adoption of interpretable deep learning techniques and improving knowledge of the impact of promotions on demand.

1.3 Objectives

The main purpose of this dissertation is to apply, fine-tune and compare interpretable deep learning models to accurately forecast demand in the retail industry, considering the impact of promotions and other external factors such as the type and method of promotion. The dissertation will also aim to measure and evaluate the effects of different promotional events and types on demand as well as to identify what products are the most impacted by promotions. To achieve these goals, the following steps will be undertaken:

- perform data pre-processing on real datasets to infer their main characteristics and evaluate their quality to apply the forecasting models;

- determine the proper forecasting window;

- apply and fine-tune different forecasting models, including deep learning algorithms, to these datasets and compare their performance in terms of accuracy and interpretability;
- compare the results obtained using the different algorithms with each other, as well as with traditional statistical methods.
- identify the factors that are most important in predicting the demand;
- analyze the effect of different promotions on demand;
- identify how the features of the models affect the values of the demand.

1.4 Thesis Contributions

This thesis provides several key contributions to the field of time series forecasting, particularly in the context of retail sales. One of the most significant contributions is comparing and applying different forecasting models, both global (including XGBoost, LSTM, and Temporal Fusion Transformer) and local (specifically SARIMAX). By comparing these models in a practical retail setting, the research offers insights into the strengths and weaknesses of each approach, contributing to a better understanding of the suitable contexts and scenarios for their application.

Moreover, the study provides an extended application of the emerging Temporal Fusion Transformer (TFT) model, firstly introduced in 2020. The application of TFT to predict a large volume of time series data in the retail context is groundbreaking as it needs more research to strengthen the evidence of the accuracy of the model in the different data sets. This contribution pushes the frontier of understanding how such deep learning models can be utilized in retail environments and their potential advantages in dealing with real-world, high-volume, and complex data.

Beyond the application of models, this research also contributes significantly to the field by exploring model interpretability. The study employs agnostic interpretability tools to better understand and explain the predictions made by these models. This contribution is particularly noteworthy in the era of "black box" machine learning models, where understanding the reasoning behind predictions is often difficult yet crucial for practical implementation.

Finally, the research extends the application of interpretability analysis at various levels within the retail environment. From a global perspective to an item-level viewpoint, this layered approach facilitates a granular understanding of promotional events and their impacts on sales. This is a significant contribution to the literature as it provides a method for dissecting and understanding complex sales dynamics in retail environments, with implications for strategic decision-making and resource allocation in real-world retail operations.

Together, these contributions advance state-of-the-art time series forecasting and provide actionable insights for retailers, data scientists, and decision-makers in understanding and leveraging promotional activities.

1.5 Document Structure

The final report is divided into several chapters, each with its focus and purpose.

Chapter 2 provides an overview of the relevant concepts and theories related to demand forecasting in the presence of promotions. The chapter is divided into several sections, including a section on time series analysis, 2.1, a section on time series pre-processing, 2.2, and a section on various forecasting methods, 2.3, including parametric models, machine learning models, and deep learning. Additionally, 2.4 discusses interpretability in machine learning. The chapter also includes a section on demand influencing factors as external variables, 2.5, and a section on promotion influence factors, 3.1. Finally, the chapter concludes with a section on evaluation metrics, 2.6, including scale-dependent, percentage errors/scale-independent, relative errors, and scaled errors.

Chapter 3 reviews the current state-of-the-art research on demand forecasting methods in the presence of promotions. This chapter aims to identify gaps in the existing literature and provide a foundation for the research proposed in the dissertation.

Chapter 4 presents this study's methodology and experimental setup. This chapter covers the data sources, data pre-processing, and the different models and algorithms utilized to evaluate the performance of the proposed approach.

Chapter 5 presents the experiments' results to evaluate the proposed approach's performance. This chapter includes a detailed analysis of the experimental results, including comparisons with state-of-the-art methods and discussions of the strengths and weaknesses of the proposed approach.

Finally, Chapter 6 summarizes the main findings and implications of the research and provides recommendations for future work in this area.

Chapter 2

Background

2.1 Time series

A time series is a chronological succession of observations on a variable of interest. This variable is measured at equally spaced periods, usually daily, weekly, monthly, quarterly, or annually [7].

Moreover, if a series of values is generated using a mathematical function, it is referred to as deterministic. However, if a random element is included in the series through the use of a mathematical time function, the series becomes stochastic or non-deterministic [2].

Time series plots can demonstrate patterns such as random, trends, cycles, or a combination of patterns [7]. As a result, the identification of patterns associated with time series uses decomposition techniques. These techniques can detect changes in data over a long period of time, such as trends (which can follow various patterns like linear, polynomial, or exponential), and are useful for predicting the typical range of values that a variable would assume if these changes didn't exist; identify cyclical patterns of variation that repeat at regular intervals, known as seasonality (for example, in the retail industry, sales tend to increase on weekends and in the summer and decrease in the winter). Additionally, these can detect short-term, unpredictable fluctuations, often caused by unexpected events or weather, known as residues, obtained by eliminating trend and seasonality from the time series [1, 7, 2]. Also, stationarity is another important aspect of a time series. A stationary series maintains the average, variance, and constant auto-correlation over time. Meaning that the probability distribution doesn't alter with time [7].

Finally, for time series problems, depending on the number of inputs and outputs, a univariate time series problem comprises one time series as an input and as an output, this approach is recommended if the input time series does not have any dependencies on others. Contrarily, multivariate time series problems have multiple series as input and as output, although the number of inputs can be different from the number of outputs. Therefore, when there is a correlation between multiple series a multivariate with the same number of inputs as outputs approach is considered, and otherwise when additional information is necessary from another series [8]. Also, the selection of proper input series type and step size is critical in time series analysis for constructing a trustworthy and effective model. When considering the training and validation process, there are

two primary classifications to consider: fixed and variable length input sequences, as shown in Fig. 2.1. Sliding window time series problems contain fixed-length input sequences that have a fixed length, such as one hour, one day, or one year. Variable-length input sequences, on the other hand, expand in length with time. Together with the type of input sequence, the step size is an essential issue to consider. The amount of accessible training samples can be changed by moving or increasing the sequence window one or a few timesteps at a time. A lower step size results in more training samples, which can considerably improve the model's accuracy. As a result, while determining the input sequence type and step size, it is critical to examine the problem at hand as well as the available data. This consideration is critical in generating a reliable time series model [8].

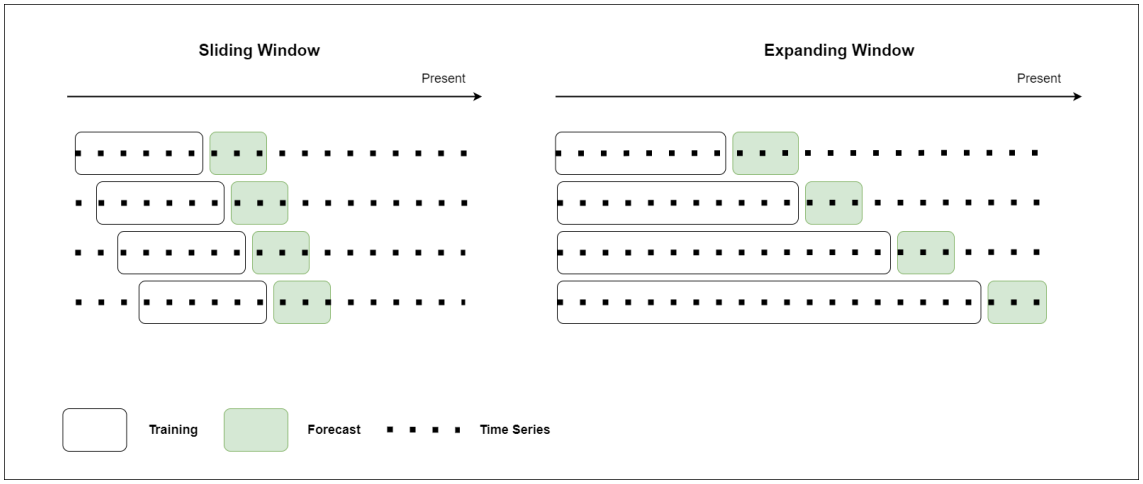


Figure 2.1: Sliding and Expanding window visual representation

Additionally, considering the duration of output sequences, there are single-time steps or multiple-time steps into the future. Time series issues can be classed as single-step or multistep, with the latter segmented further into single-shot and recursive forecasts. Single-step time series problems are a straightforward task that requires forecasting a single time step into the future. In contrast, multistep time series problems are more complex, requiring forecasting multiple time steps into the future. The reliability of predictions typically decreases as forecasts go further into the future. Recursive multistep output sequence is another method for predicting future values, in which we forecast one time step multiple times. However, this approach carries forward the previous prediction's error. It is essential to note that although forecasting a single time step is more reliable than forecasting multiple time steps at once, cautiousness must be taken when employing recursive multistep prediction [8].

Overall, a structured methodology that considers the input sequence type, step size, and output sequence length is essential for successful time series analysis. By carefully selecting the most appropriate input sequence type and step size, analysts can optimize their model's accuracy, leading to more reliable forecasts.

2.2 Pre-processing

Data mining aims to extract useful information from time series, such as patterns. This strategy depends on the quality of the data, usually in its raw structure, which can have outliers, missing values, noisy, incomplete and unreliable data because of the size of the datasets, multiple resources, and the gathering methods. Therefore, the data has to be pre-processed, which comprises the preparation and transformation of the dataset into a suitable form to be mined.

Pre-processing aims to reduce the data size, uncover links between data, normalize data, remove outliers, and extract features from data.

Firstly, data cleaning involves identifying and correcting errors or inconsistencies, such as missing values, duplicates, or incorrect data types otherwise, it would conduct erratic and poor outputs [9].

Hence, when a missing value is detected, it can be substituted manually, which requires effort, it's time-consuming, and it's not appropriate for extensive datasets, using a global constant to fill the missing value with an equivalent attribute for all records, employing the average value for that attribute or the average value for records belonging to the same class or utilizing the most likely values, this approach recurs to inference based regressions [9].

Moreover, noisy data is a random error or variance in a recorded variable that deviates from the real value. Binning is a method that works on smoothing stored data established on closer observations, executing local smoothing. Regression also works on smoothing by fitting data into a function, for example, using linear regression to decide the most suitable line to fit two variables so that each attribute is used to predict the other. Also, clustering groups sets of points according to their distance from each other. This technique can detect outliers since identical points are in the same cluster and the ones that don't belong to any grouping are considered outliers [9].

Additionally, the process of altering data into forms applicable to the mining process is called data transformation and it comprises the following approaches:

- Smoothing- Accountable for clearing noise from the data; it contains techniques such as clustering, regression, and binning [9];
- Aggregation- Applies statistical metrics to summarize the data and the resultant aggregated data is used in data mining algorithms [9];
- Generalization - Replaces lower-level data with higher-level using hierarchical concepts [9];
- Normalization- Alters data values into a specific range of values. This is useful for techniques such as classification, artificial neural networks, and clustering algorithms [9];
- Concept hierarchy generation- In data generalization, low-level or primitive data is replaced with higher-level notions using concept hierarchies. For example, categorical qualities are extended to higher-level concepts [9].

Then, data reduction reduces the presentation of the dataset in smaller volumes preserving the information of the original data. This includes distinct approaches such as dimensionality and

numerosity reduction and its intent is to enhance storage efficiency while reducing data storage and analysis costs [9]. Dimensionality reduction or data compression seeks to minimize the number of features. Data is replaced or approximated for numerosity reduction utilizing alternative and smaller data representations such as parametric models [9].

2.3 Forecasting

Forecasting is a crucial problem that traverses multiple domains including business and industry, government, economics, environmental sciences, medicine, social science, politics, and finance [7]. It's about anticipating the future as accurately as possible, given all of the knowledge available, including recorded data and understanding of any future events that might impact the forecasts [10].

Forecasting can be classified as short-term, medium-term, and long-term. Short-term forecasting problems concern predicting events only a few time periods (days, weeks, and months) into the future and are required for the scheduling of personnel, production, and transportation, for example [7, 10]. Medium-term forecasts spread from 1 to 2 years into the future and are utilized to define future resource necessities, in order to buy raw materials, employ personnel or buy machinery and equipment. Additionally, long-term problems can extend further by numerous years and are employed in strategic planning [7, 10].

Prediction of coming occasions is a crucial input into considerable types of planning and decision-making processes, with application to areas such as operations management where organizations use forecasts of product sales or need for services in order to plan production, manage inventories, supervise the supply chain and determine staffing necessities [7].

If there is little or no historical data on which to base the forecast, then qualitative forecasting methods must be employed, these are subjective and demand judgment from experts. Otherwise, quantitative forecasting techniques make proper use of recorded data and a forecasting model. The model formally outlines patterns and describes a statistical connection between prior and current values of the variable [7, 10].

There are two categories of forecasting techniques: parametric and non-parametric. Parametric methods involve using exponential smoothing, autoregression, or moving averages, while non-parametric techniques involve describing the characteristics of the data without any prior knowledge of the data's distribution and can be either global or local [2].

2.3.1 Parametric Models

Parametric or statistical methods require prior knowledge regarding data distribution to build the forecasting models once it relies on a set of parameters to obtain optimized prediction results. These models can be subdivided into two groups: Exponential Smoothing and Autoregressive Integrated Moving Average (ARIMA) models [2].

Exponential smoothing and ARIMA methods are the two most widely used to forecasting demand for products in the retail area, these can model trends and seasonal fluctuations usually

contained in aggregate retail sales time series [11]. However, these linear models are deemed insufficient when estimating aggregate retail sales, recognizing signs of non-linearity, and volatility [1].

The exponential smoothing model forecast values of a variable based on a weighted combination of their past observations, these weights exponentially decay over time, and from the integration of an additive or multiplicative structure to recreate the smoothed components [2].

Simple exponential smoothing (SES) is appropriate for forecasting data with no trend or seasonal patterns or it can provide predictions that underestimate or overestimate the actual values [10, 2]. This form of simple exponential smoothing is given by the forecast and level equations, respectively:

$$\hat{y}_{t+h|t} = \ell_t \tag{2.1}$$

Where:

$$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1} \tag{2.2}$$

Where:

- ℓ_t is the level of the series at time t;
- α is the weight allocated to the recorded values;
- y_t corresponds to the last observed value.

Holt's Exponential Smoothing results from an extension of the simple exponential smoothing to allow the forecasting of data with a trend. A forecast equation and two smoothing equations are used in this approach (one for the level and one for the trend) [10]:

$$\hat{y}_{t+h|t} = \ell_t + hb_t \tag{2.3}$$

Where:

$$\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{2.4}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{2.5}$$

Where:

- $b_t$represents an estimation of the trend of the series at time t;
- α is the smoothing parameter for the level;

- β^* is the smoothing parameter for the trend.

The trend equation reveals that b_t is a weighted average of the estimated trend at time t based on $\ell_t - \ell_{t-1}$ and b_t, the previous estimate of the trend.

Holt-Winters seasonal methods comprises the forecast equation and three smoothing equations-one for the level ℓ_t, one for the trend b_t and one for the seasonal component s_t, with the corresponding smoothing parameters α, β^* and γ and m, that denotes the period of the seasonality [10].

Moreover, these can be divided into multiplicative and additive and their choice depends on the seasonality of the series [10].

If the seasonal fluctuations in demand become more intense as the overall average demand increases, then multiplicative models are the better option. On the other hand, if the difference between the highest and lowest demand remains fairly constant, then additive methods are preferred [10, 2].

The component form for the additive method is:

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)} \tag{2.6}$$

Where:

$$\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{2.7}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{2.8}$$

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, \tag{2.9}$$

The trend equation is identical to Holt's linear method.

The seasonal equation shows a weighted average between the current seasonal index, and the previous seasonal index of the same season.

The multiplicative method performs better when the seasonal variations are changing proportionally to the level of the series.The component form for the multiplicative method is:

$$\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t+h-m(k+1)} \tag{2.10}$$

Where:

$$\ell_t = \alpha\frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{2.11}$$

$$b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{2.12}$$

$$s_t = \gamma\frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m}. \tag{2.13}$$

Contrarily, ARIMA models of order (p, d, q) bring together three statistical techniques: autoregression (AR(p)), which captures the relationship between observations; integration (d), which

determines the number of differences needed for stationarity; and moving average (MA(q)), which includes new factors [2].

Therefore, moving average (MA) models use arithmetic averages concerning the last r values of the time series, r being the order of the model, to predict the future value. These models tend to have limitations when time series present trends and seasonality [2].

The full model of ARIMA (p, d, q) can be written as:

$$y'_t = c + \phi_1 y'_{t-1} + \cdots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{2.14}$$

Where y'_t is the differenced series. The right-hand side includes both lagged values of y_t and lagged errors [10]. These can model homogeneous non-stationary series as well as stationary series [2].

SARIMA, seasonal auto-regressive integrated moving average, is used when the series presents both components of trend and seasonality noted as SARIMA(p, d, q)$(P,D,Q)_m$ [2]. The seasonal part of the model consists of terms that are similar to the non-seasonal components of the model, but involve backshifts of the seasonal period [10].

From a time plot, it's not possible to determine the values of p and q that are suitable for the data. However, it is sometimes possible to use the Autocorrelation function (ACF) plot, and the closely related Partial autocorrelation function (PACF) plot, to determine appropriate values for p and q [10].

ACF plot shows how the present value of a given time series is correlated with the past values. However, then y_t and y_{t-2} might be correlated, simply because they are both connected to y_{t-1}, rather than because of any new information contained in y_{t-2} that could be used in forecasting y_t [10].

Therefore, PACF explains the partial correlation between the series and lags itself. So the first partial autocorrelation is identical to the first autocorrelation because there is nothing between them to remove [10].

Da Veiga et al. [12] compared ARIMA and Holt-Winters models to forecast the demand for a group of 50 perishable products with a short shelf life in the period from 2005 to 2013. The performance of both models was compared using MAPE and Theil inequality index (U-Theil). According to these evaluation metrics, the Holt-Winters method outperformed ARIMA because this last method assumes that the historical pattern of the data remains unaltered during the forecast horizon.

Nonetheless, the horizon of predictions produced with the Holt-Winters method shouldn't overextend the seasonal cycle of the series as it affects the model accuracy. Despite that, the Holt-Winters model is simpler and can deliver higher accurate forecasting results than those obtained with more complex approaches [12].

2.3.2 Machine Learning Models

Machine Learning is a subgroup of artificial intelligence where the algorithm is able to make predictions without being explicitly programmed, it learns from raw data to generate predictive models based on predesigned techniques [1, 2]. Supervised and unsupervised learning are two categories of learning algorithms. Firstly, supervised learning uses labeled data, both input and output, to uncover an adequate relationship between them [13]. Unsupervised learning, on the other hand, only uses input data without the corresponding output [13]. When series are complex and highly non-linear these techniques are able to surpass parametric models. Time series are rearranged into subsequences where the nearest values to the current one are combined using approximation functions to anticipate future values [1, 2].

2.3.2.1 Random Forest

Random Forests (RF) are an ensemble learning method that combines multiple decision trees to make predictions. They can be used for time series forecasting by training a set of decision trees on historical data and using them to predict future values [14]. These models are less susceptible to overfitting as Decision Trees models tend to be [15].

Additionally, attending to forecasting problems, RF models generate output based on the average of the results from all of the trees in the model. To properly use the model for this purpose, the data must be transformed into a supervised learning problem, which involves creating new columns using previous observations to predict future ones [16].

Random Forests can be expressed as:

$$\hat{f} = \frac{1}{K} \sum_{k=1}^{K} f_k(X') \tag{2.15}$$

Hyperparameters have a key role in the performance of these models when it comes to forecasting outcomes [17]. Therefore, the number of trees used in the prediction averaging is determined by $n_{estimators}$, the minimum number of samples needed at a node is specified by min samples leaf, and the maximum depth of each tree is determined by max depth. If the value for max depth is too high, the model may overfit the training data. These main hyperparameters impact a random forest regressor model [17].

2.3.2.2 XGBoost

Gradient boosting is an ensemble machine learning method that constructs a high-performing model by aggregating the results of numerous simpler models, or base learners, typically decision trees. Unlike other boosting methods, gradient boosting sequentially creates and trains models to improve output prediction. It develops new base learners to minimize a loss function tied to the entire ensemble, assigning higher weights to instances mispredicted in earlier stages, thus focusing on refining these predictions and learning from past errors [18].

Extreme Gradient Boosting (XGBoost) is a specific variant of gradient boosting that brings enhanced power and performance. It differentiates itself through advanced regularization, contributing to improved model generalization and reduced overfitting. XGBoost also computes second-order gradients of the loss function, offering additional insight into the gradient's direction and simplifying the task of minimizing the loss function. Furthermore, it can manage missing values and supports parallel processing, compensating for increased computational requirements. XGBoost focuses on defining and optimizing the objective function using gradient descent optimization and any differentiable loss functions, thereby reducing the loss function by incorporating weak learners [18].

As follows, XGBoost seeks to minimize the regularized goal:

$$obj(\theta) = \sum_i L(\hat{y}_i, y_i) + \sum_k \Omega(f_k), \quad f_k \in \mathscr{F} \tag{2.16}$$

where L is the training loss function that assesses the difference between the model's projected value $\hat{y}_i$ and the actual value y_i. Ω is the regularization function that gauges the model's complexity and helps to prevent overfitting. f_k is a function in the functional space $\mathscr{F}$, which contains all conceivable regression trees. XGBoost employs a greedy search technique to build an ideal tree structure to minimize the objective function $obj(\theta)$ [19].

2.3.2.3 Support Vector Machines

Support Vector Machines is based on statistical learning and intends to minimize structural risk to obtain a lower training error and reduce the upper bound on the model's generalization error [20, 2]. In the context of binary classification, the generalization concept involves finding a hyperplane that separates a set of points into two classes by placing the greatest number of points from one class on one side of the hyperplane and maximizing the distance between the two classes and the decision surface [2].

The line with the maximum margin, denoted as the optimal separation hyperplane, is acquired during the model training. However, if the samples cannot be separated by a straight line, the solution will instead focus on transforming the input data into a higher-dimensional space using a kernel function. The most common kernel functions used in practice are linear, polynomial, and Radial Basis Function (RBF). Creating an SVM involves finding a solution to a quadratic problem with linear constraints based on the input data set, parameters, and margin of separation [20, 2].

2.3.2.4 Artificial Neural Networks

Artificial Neural Networks (ANN) have processing units organized into input, hidden, and output layers. Each layer's nodes are linked to nodes in adjacent layers, and each link is weighted. The inputs are then multiplied by the weights and added to each unit.

Following that, the sum is transformed depending on the activation function, which is usually a sigmoid function, tan hyperbolic, or rectified linear unit (ReLU) due to their mathematical derivative, which makes it easier to compute partial derivatives of the error delta involving individual weights. The sigmoid and tanh functions convert the input into a narrow range of output values. In contrast, ReLu exhibits both saturating and non-saturating tendencies [21].

Moreover, ANNs can be typed as, regarding times series prediction:

- Feedforward Neural Network;
- Recurrent Neural Network (RNN);
- Radial Basis Function Neural Network.

Among ANN, the Perceptron is the most straightforward approach used particularly to classify linearly separable classes [2].

The Perceptron consists of a single neuron that contains i data inputs, $x_i \in \mathrm{X}$, and predicts an outcome. The linear combination of inputs with weights added a bias, $\mathrm{b} \in \mathfrak{R}$, resulting in the net value, and it's presented as:

$$\sum_{i=1}^{l} w_i x_i + b \tag{2.17}$$

that it's sent to an activation function f that sets the output y of the neuron. The learning process for the Perceptron involves adjusting its synaptic weights through a limited number of iterations [2].

Additionally, a Multilayer Perceptron (MLP) can have more than one layer of neurons. These layers operate as combiners of attributes and with suitable links between the inputs and the intermediate layers, the model that will produce the accurate mapping between the data input and the outcomes is obtained [2]. Backpropagation uses gradient descent to reduce error by changing the weights depending on the partial derivative of the error for each weight, allowing ANN to have more than two layers [21].

The output of the model of a three-layered MLP and a single neuron in the output layer can be represented as follows:

$$y = f(\sum_{j=1}^{n} w_j f(\sum_{i=1}^{l} w_{ij} x_i + b_{0j}) + b_0) \tag{2.18}$$

Multilayer Perceptron models, despite their complexity, can operate with substantial amounts of data and still have an appropriate generalization. Both Perceptron and MLP algorithms are feed-forward neural networks since their neuron-to-neuron signals flow only from input to output [2, 21].

2.3.3 Deep Learning

Deep learning involves using neural networks to analyze raw data and solve complex problems by utilizing multiple hidden layers in the network for processing [22]. These models are able to extract important, unique information from a raw dataset without requiring any input or guidance from a human [23].

2.3.3.1 Recurrent Neural Network

Recurrent Neural Networks (RNN), in contrast to MLP, establish a cycle in which the output of one layer becomes the input to the next layer, and the output of the layer becomes the input to itself, resulting in a feedback loop [24]. This recursion reveals a short-term memory allowing the storage of complex signals for longer periods [2]. Recurrent Neural Networks are able to effectively represent the chronological order of time series data and have fewer parameters that need to be adjusted during the fitting process [24].

A representation of an RNN driven by an external signal x^t, to model the values of the hidden units that are neither of the two layers (input and output), it's given by:

$$h^t = f(h^{(t-1)}, x^t, \theta) \tag{2.19}$$

Where h^t is the state of the system at step t and θ is a parameter of a transit function f [24]. Nevertheless, RNNs have a recursive characteristic that can sometimes lead to optimization problems that are difficult to solve, known as vanishing or exploding gradients [24].

There are different types of architectures for RNN models depending on their length. These include: a one-to-one architecture, where a single output corresponds to a single input; a one-to-many architecture, where a single input leads to multiple outputs; a many-to-one architecture, where multiple inputs produce a single output; and a many-to-many architecture, where multiple inputs result in multiple outputs [16]

Moreover, a deep neural network comprises multiple layers of nodes. These techniques only require minimal data preparation before utilizing an end-to-end model to address the forecasting issue [24]. Thus, the most standard architectures of deep neural networks are presented below:

- Convolution Neural Network- established on the human visual cortex and is the neural network most employed in computer vision [21];
- Autoencoder learns the representation of the input data set for dimensionality reduction and to regenerate the starting data set using an unsupervised technique. Backpropagation is used to implement the learning algorithm [21];
- Restricted Boltzmann Machine (RBM) uses unsupervised learning algorithms to obtain non-linear generative methods from unlabeled data, to prepare the network to improve a function of the probability of vector in the visible units so it can reconstruct the input. It learns the probability distribution over its inputs [21];

- Long Short-Term Memory (LSTM) stores information learned in previous states. This technique overcomes the restriction of vanishing gradients by allowing gradients to pass unchanged while keeping track of long-term dependencies. As a result, they are preferred for learning from sequence input data to develop models that rely on context and previous states [21].

2.3.3.2 Long-Short Term Memory

Long-Short Term Memory (LSTM) is derived from RNN models which are able to understand the temporal sequence and dependencies between data points in a sequence. The LSTM block includes a self-connected memory cell which is controlled by three gates, allowing the LSTM model's memory to retain information for extended periods of time [25].

These gates are described as :

- Forget gate- determines the amount of information from previous timesteps that should be discarded or remembered, and it can be represented as follows:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{2.20}$$

- Input gate- determines which information is necessary to be added and updated from the current timestep to the next prediction as:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{2.21}$$

- Output gate- obtains the future predicted value and it can express:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.22}$$

where σ is the sigmoid activation function and h_{t-1} describe the output of the last time $step_{t-1}$. The bias is characterized as b and W denotes the weights.

2.3.3.3 DeepAR

DeepAR is a model based on autoregressive recurrent neural networks proposed by Salinas et al [24]. Instead of predicting individual or small sets of time series, this approach is trained on a global model that includes data from all recorded time series [24].

The suggested model employs an LSTM (long-term memory) architecture to address the issue of explosive or vanishing gradients. This architecture allows for the retention of seasonal patterns and dependencies on relevant factors over time, while requiring minimal manual input for capturing complex, group-dependent behavior [24]. This is able to predict the future behavior of items that have little or no past data available by analyzing the behavior of similar items. It generates

probabilistic forecasts using Monte Carlo samples, which can be used to calculate quantile estimates for different time periods in the forecasting horizon. This model does not assume that the data follows a Gaussian distribution, but rather offers a range of likelihood functions for the user to select based on the statistical characteristics of the data [24].

Salinas et al. [24] evaluated their method using five datasets: a dataset of monthly sales data for various items sold by a US automobile company (parts), a dataset of hourly electricity consumption data for 370 customers (electricity), a dataset of hourly occupancy rates for 963 car lanes in the San Francisco bay area (traffic), and two datasets of weekly sales data for items sold on Amazon, including both fast-moving and slow-moving items, as well as new products introduced in the weeks leading up to the forecast time (ec and ec-sub).

Several methods, including the Croston method [26], the negative-binomial auto-regressive method [27], an innovative state space model with covariate features, and an ETS method, proposed by Öller et al. [28], were compared to two baselines RNN techniques using the ρ-risk metric. The Deep AR model demonstrated the ability to learn from relevant time series data, adjust to changes in scale, make accurate predictions, and recognize patterns like uncertainty and seasonality. However, the forecasted data did not consider factors like promotional strategies and marketing efforts that can affect demand.

2.3.3.4 Temporal Fusion Transformer

Multi-horizon forecasting is a crucial problem in time series machine learning, where the prediction of variables of interest is made at multiple future time steps, rather than just one. This approach allows to optimize the actions at multiple steps in the future.

However, practical multi-horizon forecasting applications often have access to a variety of data sources, which can make the task challenging. Deep neural networks (DNNs) are effective in multi-horizon forecasting, with strong performance improvements over traditional time series models. However, current DNN architectures for multi-horizon forecasting do not take into account the heterogeneity of common inputs, and are often 'black-box' models, making it difficult to understand how the model arrived at its predictions, and for users to trust the model's outputs [29].

Let there be I unique entities in a given time series dataset, such as different stores in retail or patients in healthcare. Each entity i is associated with a set of static covariates s_i and inputs $\chi_{i,t}$ and scalar targets $y_{i,t}$ at each time-step t. The input features are divided into two categories: observed inputs $z_{i,t}$ which can only be measured at each step and are unknown beforehand, and known inputs $x_{i,t}$ which can be predetermined (e.g. the day-of-week at time t) [29].

To optimize decisions and risk management, the provision for prediction intervals is useful, which yields an indication of likely best and worst-case values that the target can take. For this, quantile regression is adopted for the multi-horizon forecasting setting. The equation for the quantile forecast is given as:

$$\hat{y}_i(q,t,\tau) = f_q(\tau, y_{i,t-k:t}, z_{i,t-k:t}, x_{i,t-k:t+\tau}, s_i) \tag{2.23}$$

where $\hat{y}_{i,t+\tau}(q,t,\tau)$ is the predicted qth sample quantile of the τ-step-ahead forecast at time t, and $f_q(.)$ is a prediction model. All past information is incorporated within a finite look-back window k, using target and known inputs only up till and including the forecast start time t and known inputs across the entire range [29].

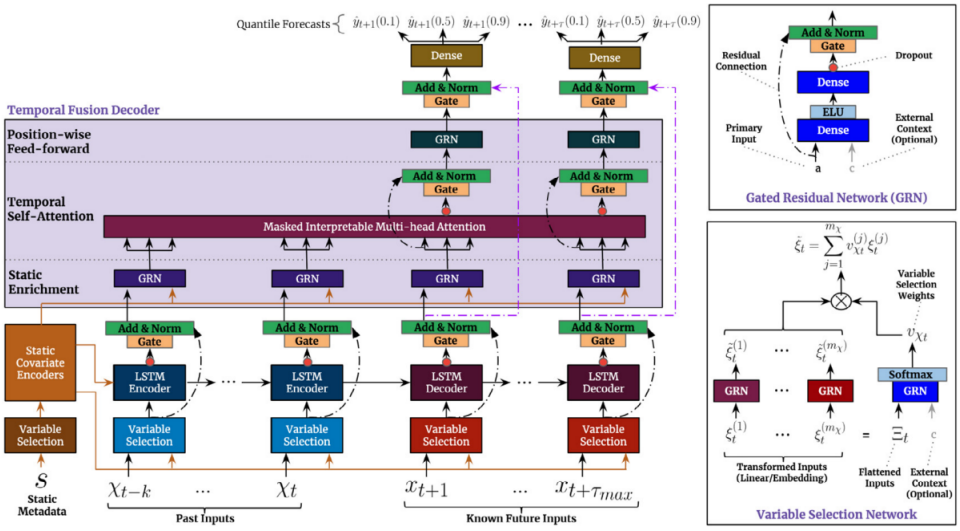


Figure 2.2: High-level architecture with individual components of TFT model. With time-varying past inputs, time-varying a priori known future inputs, and static metadata as inputs, TFT architecture is intended for multi-horizon forecasting. Time-dependent processing is based on LSTMs for local processing and multi-head attention for integrating information from any time step. Variable selection is used to select the most salient features based on input. Gated Residual Network blocks are utilized to allow effective information flow with skip connections and gating layers[29].

Lim et al. [29] selected four datasets: the Electricity and Traffic datasets, which are used in prior academic work and focus on simpler univariate time series; the Retail dataset, which is used to benchmark the model using a full range of complex inputs; and the Volatility dataset, which is used to evaluate the robustness of the model on smaller, noisy datasets. The datasets include information on electricity consumption, traffic occupancy rates, grocery sales, and stock index volatility.

- Lim et al. [29] employ the previous week (i.e., 168 hours) to forecast for the following 24 hours using a UCI Electricity Load Diagrams Dataset, which contains the electricity usage of 370 consumers - aggregated on an hourly level;

- The UCI PEM-SF Traffic Dataset lists the highway occupancy rates for 440 in the Sf Bay Area. It is also hourly aggregated, with the same prediction horizon and look-back window as the power dataset;

- Favorita Grocery Sales Dataset from a Kaggle competition, which incorporates information for various items and retailers, as well as additional exogenous time-varying inputs collected on a daily basis. Using 90 days of historical data, it's anticipated log product sales for the next 30 days;

- The OMI realized library includes daily realized volatility values and daily returns for 31 stock indexes computed from intraday data. Then estimated for the following week based on data from the previous year (i.e. 252 business days).

As a result, all-time series data are separated into three sets: a training set for learning, a validation set for adjusting hyperparameters, and a hold-out test set for performance evaluation. For each dataset, hyperparameter optimization is carried out via a random search with a defined number of iterations. To retain explainability, Lim et al. [29] uses a single interpretable multi-head attention layer, the same constant stack size and a number of heads, and treats kernel sizes as a hyperparameter, as they are observed to be dataset-dependent.

Lim et al. [29] employs a variety of models, including direct and iterative techniques, with an emphasis on deep learning models that provide predictions at future horizons directly. Simple sequence-to-sequence models with global contexts (Seq2Seq), Multi-horizon Quantile Recurrent Forecaster (MQRNN), multi-layer perceptron (MLP), and linear quantile regression with L2 regularization (Ridge) are among the direct techniques. DeepAR, Deep Semantic Similarity Model (DSSM), and the Transformer-based architecture of Li et al. [30] with local convolutional processing (ConvTrans) are among the iterative approaches. Random search across a pre-defined search space is used for hyperparameter optimization, with the same number of iterations used across all benchmarks for a particular dataset.

As a consequence, TFT surpasses all benchmarks across a wide range of datasets, highlighting the advantages of matching the architecture with the broad multi-horizon forecasting problem. When compared to the next best model, the TFT model produces 7% lower P50 and 9% lower P90 losses on average, and the benefits are statistically significant at a 95% confidence level. Lim et al. [29] also emphasize the need to account for observed inputs, citing ConvTrans's lower performance on datasets where observed input imputation is necessary and the advantages of quantile regression when targets are not adequately covered by Gaussian distributions.

There are also three interpretability use cases presented: (1) determining the relevance of each input variable in prediction, (2) displaying persistent temporal patterns, and (3) recognizing any regimes or events that cause major changes in temporal dynamics. Lim et al. [29] emphasizes that these approaches focus on ways to aggregate the patterns over the full dataset, deriving generalized insights about temporal dynamics.

2.3.3.5 N-BEATS

Oreshkin et al. [31] developed a deep neural network for univariate point forecasting based on backward and forward residual links and a deep stack of fully connected layers- N-BEATS. This

architecture can produce human-interpretable outputs and does not depend on time-series-specific feature engineering or input scaling.

According to the architecture depicted in Fig. 2.3, N-BEATS is formed of blocks that obtain an input x_ℓ and output two vectors, $\hat{x}_\ell$ and $\hat{y}_\ell$: the block's forward forecast of length H and the block's most suitable estimation of its input [31].

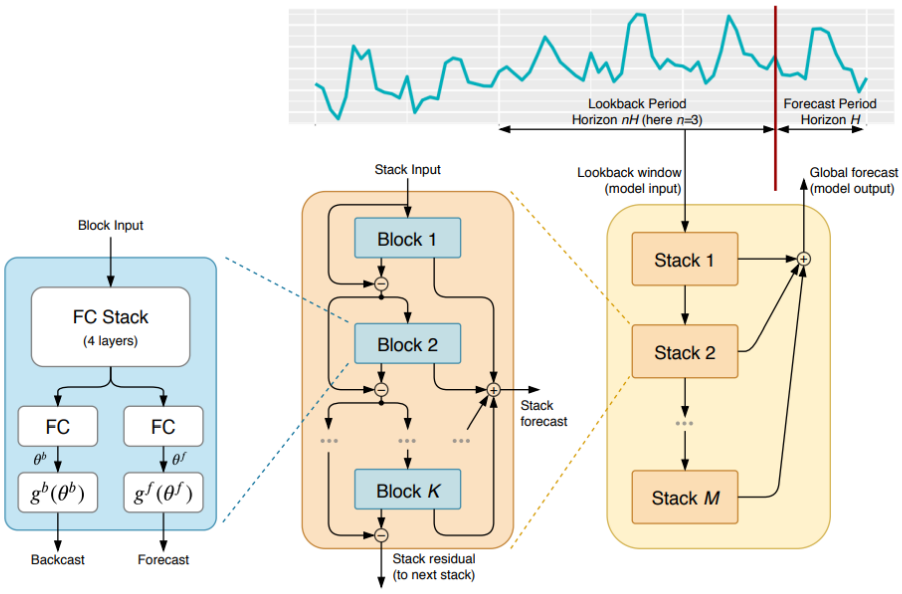


Figure 2.3: Architecture concept. A multi-layer, fully connected network with RELU nonlinearities serves as the foundation. It forecasts basis expansion coefficients both forward, θ^f, and backwards, θ^b, (backcast). The doubly residual stacking concept is used to arrange blocks into stacks. A stack may have levels that share g^b and g^f. Forecasts are arranged in a hierarchical order. This allows for the construction of a very deep neural network with interpretable outputs [31].

Additionally, the internal basic building block of the model comprises an entirely connected network that produces forward and backward expansion coefficient forecasts and backward and forward basis layers that receive these coefficients. Also, the model employs two residual branches to handle backcasting and forecasting. These produce a partial forecast that is firstly aggregated at the stack level and then at the overall network level, delivering a hierarchical decomposition. The final forecast is the sum of all partial forecasts. The equations for the ℓ-th block of N-BEATS are represented as follows:

$$\mathbf{h}_{\ell,1} = FC_{\ell,1}(x_\ell), \mathbf{h}_{\ell,2} = FC_{\ell,2}(h_{\ell,1}), \mathbf{h}_{\ell,3} = FC_{\ell,3}(h_{\ell,2}), \mathbf{h}_{\ell,4} = FC_{\ell,4}(h_{\ell,3}) \tag{2.24}$$

$$\theta^b_\ell = LINEAR^b_\ell(h_{\ell,4}), \qquad \theta^f_\ell = LINEAR^f_\ell(h_{\ell,4}) \tag{2.25}$$

The two output vectors of the ℓ-th block can be expressed as:

$$\hat{y}_\ell = \sum_{i=1}^{dim(\theta_\ell^f)} \theta_{\ell,i}^f v_i^f \tag{2.26}$$

$$\hat{x}_\ell = \sum_{i=1}^{dim(\theta_\ell^b)} \theta_{\ell,i}^b v_i^b \tag{2.27}$$

where $\hat{y}\ell$ is the forward forecast of length H and $\hat{x}\ell$ is the block's most suitable estimation of its input. Both functions g_ℓ^b and g_ℓ^f are learnable and deliver forecast and backcast basis vectors such that their respective outputs can be represented adequately via varying expansion coefficients θ_ℓ^f and θ_ℓ^b [31].

The residual branches of the model are described by the following equations:

$$x_\ell = x_{\ell-1} - \hat{x}_{\ell-1} \tag{2.28}$$

$$\hat{y} = \sum_\ell \hat{y}_\ell \tag{2.29}$$

The hierarchical doubly residual topology proposed in N-BEATS allows for fluid gradient backpropagation, and the final forecast $\hat{y}$ is the sum of all partial forecasts. This structure enables a hierarchical decomposition of the output [31].

Moreover, the final forecast $\hat{y}$ results from the sum of all partial forecasts. In a generic model context, when stacks are allowed to have arbitrary g_ℓ^f and g_ℓ^b, the model can be seen as a universal function approximator that can simulate any continuous function. However, when g_ℓ^f and g_ℓ^b are chosen based on specific problem properties, the model can be optimized more efficiently, decreasing the risk of overfitting and improving generalization capabilities [31].

2.4 Interpretability in Machine Learning

Interpretability is crucial in machine learning, especially when applied to time series data since it helps us to comprehend and explain how a machine learning model makes predictions or judgments. This is especially crucial in medicine or autonomous driving industries, where the model's decisions might have serious repercussions and require transparency. Explainable Artificial Intelligence (XAI) methods are intended to make complex machine learning models more interpretable [32], which is important for increasing trust in the model and ensuring that it makes fair and unbiased decisions, especially in highly regulated domains or situations where the model's decisions have a significant social impact [4].

There are three main categories of interpretability methods in machine learning: pre-model methods, which are applied to the data itself and focus on exploring and understanding the data before selecting a model; in-model methods, which involve building an inherently interpretable model; and post-model methods, which aim to improve the interpretability of a model after it has

been built. Pre-model interpretability techniques include exploratory data analysis and data visualization methods. In-model interpretability involves building models with constraints or other features that make them inherently interpretable, while post-model interpretability refers to methods that improve the interpretability of a model after it has been built (post hoc) [4].

One such method is SHapley Additive exPlanations (SHAP), a concept drawn from cooperative game theory [33]. It is a unified process that provides insights into the predictions generated by machine learning models.

Shapley values provide accurate explanations by assigning a priority value to each feature for a specific prediction. Shapley values give a reasonable measure of feature relevance across multiple models and can provide valuable insights into machine learning models' decision-making process [34].

This importance value is referred to as a SHAP value, represented as $\phi_i, \in \mathbb{R}$, and it provides a suitable way to attribute the contribution of features additively, and it can be obtained as follows:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F|-|S|-1)!}{|F|!} \left[f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S\left(x_S\right) \right] \tag{2.30}$$

F illustrates the collection of input features, while S is a subset of these input features, and $M = |F|$ represents the total count of input features. The given formula estimates the importance of each feature by evaluating its contribution to the prediction when it's included and then subtracting its contribution when it's not included [34]. The function $f_{S\cup i}\left(x_{S\cup i}\right)$ gives the model's output when the i^{th} feature is included. Conversely, $f_S(x_S)$ provides the model's output when the i^{th} feature is not included. Finally, $\sum_{S\in F\setminus i} \frac{|S|!(M-|S|-1)!}{M!}$ is a mathematical expression that calculates a weighted average over all potential subsets of S in F.

There are several techniques for calculating SHAP values. Model-agnostic approaches such as Shapley sampling values and Kernel SHAP are examples of model-specific methods such as Max SHAP and Deep SHAP [34].

Therefore, intrinsic interpretability refers to models that are interpretable by themselves, often through the imposition of constraints such as sparsity or monotonicity. Post hoc interpretability involves using explanation methods that are applied after the model has been trained and can be applied to both interpretable and non-interpretable models. Intrinsic interpretability is also known as transparency and deals with how the model works, while post hoc interpretability focuses on what else the model can tell [4].

Moreover, two main classes of interpretability approaches are established on their model specificity: model-specific and model-agnostic. Model-specific methods are limited to certain model classes and rely on understanding the internal workings of the model, such as weights in a linear model. These methods are typically applied to intrinsically interpretable models.

There are several approaches to evaluating the explanations provided by XAI methods applied to time series data. One approach is to use qualitative methods, such as expert evaluations or user studies, which involve humans assessing the quality and usefulness of the explanations. Another approach is to use quantitative methods, such as metrics that measure the fidelity or consistency

of the explanations [32]. There are limitations to both of these approaches. Qualitative methods can be subjective and may not be consistent across different evaluators. Quantitative methods may not fully capture the complexity and nuance of time series data and the explanations provided by XAI methods [32]. There is also a need for methods that can increase the confidence, stability, and robustness of XAI explanations for time series data. This can involve techniques such as sensitivity analysis, which tests the robustness of the explanations to changes in the data or model, or methods that can identify and mitigate the influence of outlier data points.

Overall, the state of the art in explainability methods applied to time series data is still in its early stages, and there are many potential avenues for future research. Some potential areas of focus include developing more effective methods for evaluating explanations, improving the stability and robustness of XAI methods for time series data, and exploring the use of interactive or multi-modal explanations for time series data [32].

2.5 Demand Influencing Factors

Demand can be influenced by several characteristics that can be part of the retailer's marketing strategy or driven by external factors over which the retailer doesn't have command [1]. Therefore, these demand influencing factors can be classified as controllable, which include price and product attributes, or uncontrollable factors such as events, weather, seasonality, and customer visits. Substitution and cannibalization are partially controllable as they depend on the price and product characteristics [35].

- Calendar Events: These comprise regular holidays, festivals, and special activities such as school vacations, for example [35]. Certain holidays occur at regular intervals and may thus be considered seasonal, whilst others float around in the calendar and thus cannot be considered seasonal [1];

- Weather: The demand for some retail products is likewise affected strongly by air temperature, precipitation, snow cover, sunshine duration, wind speed, and relative humidity [35]. There is also the chance of differing impacts as the year progresses, which might have a greater influence than identical weather later in the year [1];

- Seasonality: At regular time periods, cyclical patterns of variation are repeated. Retail product sales data is highly cyclical, with many seasonal cycles of varying durations. Day of the week, of the month, or of the year, yearly seasons, and yearly quarters are the categories considered [35]. Sales are usually more heightened during the weekend and lower on weekdays, higher in the summer and lower in the winter, and higher around Christmas [1];

- Customer visits: Customers who visit a store can be grouped into three categories. Frequent visitors who visit regularly, irregular customers who visit sporadically, or occasional visitors such as tourists [35];

- Product characteristics: include how fresh and appealing it appears, the size of the packaging, the overall quality, and the amount of time it can remain on the shelf after being placed in the store [35];
- Price: This is classified into the normal price and reduced price, either by discounts or promotions. Price reductions of the selected products may sometimes produce a cannibalization effect on similar types of products in the same store and/or in competitive stores [35]. Additionally, distinct promotional types (e.g. buy-one-get-one-free (BOGOF) or 50% price reduction) can lead to cannibalization or a substitution effect. Similarly, when demand exceeds available stock, customers may seek to buy substitutes, increasing sales of the replacement items while underestimating need for them [1].

Additionally, forecast accuracy, limited by the time series' properties, depends on data availability, data quality, and prediction horizon [35, 36].

- Data Availability: This factor allows the identification and understanding of the external factors that impact sales [36];
- Data Quality: The prediction accuracy relies on the quality of both the training and the testing set. Also, estimation-period data represents a critical role in the accuracy of forecasts [36];
- Forecast Horizon: The longer the horizon the higher the uncertainty meaning a lower accuracy of the model [36].

Consequently, these low-accurate predictions and not considering the demand influencing factors can result in overstocking errors in sales, leading to inadequate shelf space, shrinkage, price deductions, and food wastage, or under-stocking, resulting in loss of sales, lower customer confidence, and deterioration of the market image[35, 36, 1].

2.6 Evaluation Metrics

Y_t corresponds to the observation at time t and F_t represents the forecast of Y_t. Defining the forecast error as $e_t = Y_t - Ft$ and the percentage error (PE_t) as $\frac{e_t}{Y_t} \times 100$.

Therefore, different evaluation metrics can be grouped based on their scale dependency.

2.6.1 Scale Dependent

- Mean Error (ME)

$$\frac{1}{n}\sum_{t=1}^{n} e_t \tag{2.31}$$

- Mean Absolute Error (MAE)

$$\frac{1}{n}\sum_{t=1}^{n} |e_t| \tag{2.32}$$

- Median Absolute Error (MdAE)

$$median(e_t) \tag{2.33}$$

- Mean Squared Error (MSE)

$$\frac{1}{n}\sum_{t=1}^{n} e_t^2 \tag{2.34}$$

- Root MSE

$$\sqrt{\frac{1}{n}\sum_{t=1}^{n} e_t^2} \tag{2.35}$$

If objects with different scales or magnitudes are employed for the forecast task then absolute error measures could not be applied [37].

RMSE is chosen for the MSE as it is on the same scale as the data. Nonetheless, they are more susceptible to outliers than MAE or MdAE [38].

Additionally, absolute error measures provide conservative values if data contain outliers with maximal values, which is a common phenomenon for real-world applications [37].

2.6.2 Percentage errors/ Scale independent

- Mean Percentage Error(MPE)

$$\frac{1}{n}\sum_{t=1}^{n} PE_t \tag{2.36}$$

- Mean Absolute Percentage Error (MAPE)

$$\frac{1}{n}\sum_{t=1}^{n} |PE_t| \tag{2.37}$$

- Median Absolute Percentage Error (MdAPE)

$$median(PE_t) \tag{2.38}$$

- Root Mean Square Percentage Error (RMSPE)

$$\sqrt{\frac{1}{n}\sum_{t=1}^{n} PE_t^2} \tag{2.39}$$

- Root Median Square Percentage Error (RMdSPE)

$$\sqrt{median(PE_t^2)} \tag{2.40}$$

Percentage errors are calculated based on the value of PE_t and are commonly used in the forecasting domain [37]. These measures have the disadvantage of being infinite or undefined if $Y_t = 0$ for any t, and a skewed distribution when any value of Y_t is near to zero, and therefore changes such as logarithms can help them become more stable [38, 37].

Consequently, MAPE is often substantially larger than the MdAPE [38]. Otherwise, MdAPE is more unsusceptible to outliers [37]. Another problem of percentage error approaches is that they presume a meaningful zero. As a result, when predicting errors for temperatures on the Fahrenheit or Celsius scales, percentage errors are not appropriate [37].

Moreover, scale-independent measures are non-symmetrical, which means that the error values differ whether the predicted value is bigger or smaller than the actual [37].

Outliers significantly impact the result, particularly if these have a value higher than the maximum values in normal cases. The error measures are biased. This can conduct in an inaccurate evaluation of the model performance [37].

2.6.3 Relative errors

This category comprises measures estimated as the ratio of mentioned above error measures acquired by estimated forecasting models and benchmark models [37].

- Relative Mean Absolute Error (RelMAE)

$$\frac{MAE}{MAE^*} \tag{2.41}$$

  Where MAE and MAE^* are the mean absolute error for the employed forecasting model and the reference method, respectively [37].

- Relative Root Mean Square Error (RelRMSE)

$$\frac{RMSE}{RMSE^*} \tag{2.42}$$

  Where RMSE and $RMSE^*$ are the Root MSE for the analyzed forecasting model and the benchmark model, similarly as 2.41.

- U2, also known as Theil's U statistic, is a measure of the relative error of one-step forecasts made using a random walk as the benchmark method. It is also referred to as the relative measure of RMSPE [38].

Moreover, equivalent measures can be specified using MdAE, MAPE, etc [38].

Relative MAE calculates the possible improvement from the supplied forecast technique relative to the benchmark prediction method. When RelMAE is less than one, the suggested technique

is better than the benchmark method; when RelMAE is greater than one, the proposed way is worse than the benchmark method [38]. Therefore, these evaluation metrics are easy to interpret [38, 37].

Regardless, they require several predictions for the same series to calculate MAE or MSE, and these metrics are not appropriate for measuring out-of-sample forecast accuracy at a single forecast horizon over several series [38].

Division by zero error appears if the reference forecast error is equal to zero [37]. Additionally, these measures specify the number of cases when the forecasting model is preferred to the base but do not evaluate the value of difference [37].

2.6.4 Scaled errors

Relative errors have an undefined mean and an unlimited variance. Because relative measures are computed when there are several forecasts on the same series, they cannot be used to quantify out-of-sample forecast accuracy at a single forecast horizon. Relative and based on relative error measurements attempt to eliminate the magnitude of the data by comparing forecasts to those derived from some benchmark forecast technique [38].

Hyndman and Koehler [38] proposed a new approach that scales the error based on the in-sample MAE from the naïve (random walk) forecast approach. Hence, a scaled error is defined as

$$q_t = \frac{|e_t|}{\frac{1}{m-1}\sum_{t=2}^{m}|Y_t - Y_{t-1}|} \tag{2.43}$$

which is independent of the scale of the data. A scaled error is less than one if it derives from a more suitable forecast than the average one-step naïve forecast computed in-sample. Moreover, if the forecast is worse than the average one-step naive forecast computed in-sample, it is bigger than one [38].

The Mean Absolute Scaled Error is MASE as follows:

$$mean(|q_t|) \tag{2.44}$$

Similar measurements, such as Root Mean Squared Scaled Error (RMSSE) and Median Absolute Scaled Error (MdASE), can be obtained similarly.

- MdASE

$$median(q_t) \tag{2.45}$$

- RMSSE

$$\sqrt{mean(q_t^2)} \tag{2.46}$$

However, when all of the recorded observations are equivalent, these measurements might be undefined. MASE is slightly vulnerable to outliers, although it is easier to evaluate than RMSSE and has less variability on smaller samples than MdASE [38].

Overall, other measures might be preferred as it's the case of MAE, if all series are on the same scale, because these are easier to interpret and MAPE, if all data is positive and considerably greater than zero, again for simplicity [38]. Regardless, MASE is more appropriate when there are different scales, including data, that are near zero or negative [38].

Chapter 3

State of the Art

3.1 Promotion Influence Factors

Promotions are a common phenomenon in retail operations. When these occur, a price discount is offered to customers, during specific periods, as well as a variety of additional actions to increase the notability of the promoted product or service [39]. These events lead to an uplift in sales and significantly impact consumer behavior and market demand, making it volatile not only over the promotional periods but also over the entire demand series [40, 6]. Consumers frequently stockpile goods when they are on promotion; this is especially true for less perishable products, impacting future sales of the marketed products [39].

Therefore, promotions, for certain categories of products, lead to a reduction in future demands known as the post-promotion dip effect. This happens because consumers tend to stockpile when the product is under promotion for future consumption. Additionally, customers construct a reference price for the products, affected by the different observed prices throughout time, which influences demand. If the price is higher than the reference price, demand decreases; otherwise, it increases [41].

Commonly, in the retail industry, promotional strategies are set by suppliers and retailers in advance of their happening, usually four weeks before their beginning. However, these can be altered while their occurrence. Promotions can be extended if they're performing well, adding extra promotions to induce sales for stock that is close to expiration or altering the initial plan due, for example, to inclement weather [39, 41].

According to Abolghasemi et al.[39], in the Fast-Moving Consumer Goods (FMCG), sales promotions are the leading reason for forecast adjustments and also the most difficult to forecast. FMCG are products that are consumed faster. And for that, price reductions represent an important role in these products as a significant proportion of sales are made on promotion [41]. These events are considered systematic events, and their impacts should be incorporated into statistical models to deliver highly accurate forecasts; judgment is useful when little or no historical data is available, that is, the case of either new products or new promotional campaigns, to achieve lower forecasting errors [39].

Van Donselaar et al. [42] evaluates promotions for perishable articles in a retail background. The dataset for this investigation includes 407 perishable goods from four categories from a Dutch supermarket retail chain. This is statistics from week 2 of 2010 through week 35 of 2011. The Dairy Drinks category has the most goods (182 Stock Keeping Units), while the Cold Cuts category has the fewest (48 SKUs). Overall, 1447 promotions were run over 86 weeks for 407 perishable SKUs. During promotional events, the median price decrease was around 30%, although it might range from 0% to 60%. The promotion's performance was assessed using the Lift Factor per item, defined as sales during a promotion divided by baseline sales, which correspond to the average weekly sales of non-promotional weeks in the five weeks prior to a promotion. To attain insight into this effect, five dummies were implemented as independent variables in the regression model: 0–10%, 10–20%, 30–40%, 40–50%, and 50–60%.

This research does not provide conclusive proof for the existence of a threshold, a minimum value of a temporary price discount required to change the consumer's purchases [43], and saturation levels, from which the consumer no longer increases its purchases if the discount is further increased, for the four perishable product categories. Despite this, items with a shelf-life of one and a half weeks sell about 14 weeks of average regular sales during promotional events, indicating that the effectiveness of promotions is defined by the substitution effect when consumers switch among different products driven by the price discount, and other aspects other than the stockpile constraint imposed by the short time to consume [42].

Apart from price effects, advertisement work as an important feature that can influence customer intention to buy a product. These promotional disclosures can occur in two ways: in-store and out-store advertisements. Thus, we must consider the influence that these two forms of advertising have on consumers [3]. Advertising activity is also one of the main reasons behind the judgmental adjustment of statistical predictions [40].

Out-of-store advertising influences the consumer's attitude towards the advertised brand, then on the product disclosed, and, finally, its purchase intention. On the other hand, in-store advertising has a direct impact on the consumer's purchase intention, then on the brand, and, finally, on the advertised product; however, the attitude towards the advertised product is the same regardless of the type of advertisement [3].

3.2 Demand Forecasting methods in the presence of promotions

The current state of the art in demand forecasting algorithms in the presence of promotions is provided in this section of the project. Promotions have a huge influence on consumer demand and can significantly impair forecasting model accuracy. To make the most accurate estimates, it is critical to include the presence of promotions while estimating demand.

Accordingly, Ramos et al. [11] compared the forecasting performance of state space and ARIMA methods using the retail monthly sales of five different categories of women's footwear of the brand Foreva, from January 2007 to April 2012, with 64 observations. The categories that comprise the data refer to Boots, Booties, Flats, Sandals, and Shoes. The first two are sold

mostly during the winter, contrarily, Flats and Sandals sell most during summer, and Shoes are sold throughout the entire year. For the forecasting evaluation, it was used three scaled-dependent statistics: Mean Error, MAE, RMSE, MPE, and MAPE. With the exception of Flats series all the other series show a substantial seasonal pattern and are non-stationary.

Once updated information is incorporated, multi-step predictions outperform one-step forecasts according to these measures. Despite this, both approaches are capable of anticipating trends and seasonal changes [11].

Moreover, Falatouri et al. [44] conducted a study comparing the performance of SARIMA (seasonal ARIMA) and LSTM. For this purpose, the data focused on the sales levels of the four most purchased perishable items (tomato, potato, salad, and cucumber) from a retailer in Austria, sold in more than 90 stores, over three years, starting in January 2017 to December 2019. MAPE and RMSE were the evaluation metrics employed to determine the accuracy of the proposed models. Attending to the cashier data, a 25% price reduction three days in a row was considered as promotion days.

Therefore, the results showed that the SARIMA and LSTM models had similar accuracy for the salad and tomato products, but the SARIMA model had significantly better performance for cucumber and the LSTM model had better performance for potato. The use of the SARIMAX model, which considered promotion days as an external factor, led to significant improvements in the accuracy of the forecasts for salad and tomato but only moderate improvements for potatoes and a small improvement for cucumber. The differences in the performance of the models may be attributed to the historical sale behavior and fluctuating patterns of the different products [44].

Despite that, SARIMA methods require more statistical background additionally, SARIMAX requires a deeper understanding of the external inputs to recognize special days and their duration effect.

Additionally, Ali et al. [45] employed regression trees with explicit features to forecast the demand for a product category with four subcategories (pasta, sauces, noodles, and ready-made meals) from four stores. The dataset consisted of 11936 store-SKU-week combinations across 76 weeks, with 51 weeks used for training and the remaining for testing.

There were a total of 168 store-SKU combinations in the dataset, with store 1 having 41 SKUs, store 2 with 38 SKUs, store 3 with 44 SKUs, and store 4 with 45 SKUs. The promotional periods were divided into three categories: 8 advertised on TV, 28 with window sheets in the stores, and 44 announced on national radio. The demand for the SKUs was forecasted using support vector regression (SVR) with three different kernel functions (RBF, polynomial degree 1, and polynomial degree 2) and the regression tree technique, with Exponential Smoothing as the benchmark model. MAE was applied to assess the performance of the methods [45].

The results showed that the regression trees with explicit features outperformed the SVR methods and Exponential Smoothing, although they required more data preparation. For the promotional periods, the dataset was split into two parts, with 384 store-SKU-weeks with promotions and 3786 without promotions. The Exponential Smoothing method performed better for the non-promotional periods, while the regression tree with explicit features had an improvement of

65.17% compared to the benchmark method for the promotional periods. The authors also found that pooling data from multiple stores and subcategories often improved model performance. They concluded that advanced techniques and sophisticated input are only beneficial in demand forecasting for SKUs in the presence of promotions when advanced techniques are used [45].

Furthermore, Birim et al. [46] utilizes five machine learning models, including support vector regression (SVR), decision tree regressor (DTR), random forest regressor (RFR), ANN, LSTM, to investigate the effectiveness of ML techniques for predicting demand in the context of a television manufacturing company. This study uses a time series dataset of daily information from a television manufacturing company to investigate the effectiveness of machine learning techniques for predicting demand. The dataset includes 17 features, such as demand, sales, advertising expenses, and gross rating points, and has 2614 daily observations from 2010 to 2017. The goal of the study is to use the data to forecast demand, which is the target variable for the machine learning models being tested [46].

Along with that, the dataset is divided into two groups based on low and high levels of advertising expenses. The data is first sorted by total advertising expenses and then divided into four quartile groups. The highest and lowest quartile groups are used to represent days with high and low expenses, respectively. A t-test is then used to determine if there are significant differences between the low and high-advertising groups. After identifying the low and high advertising groups, the corresponding days in each group are selected and transformed into time-series data. Demand forecasting is then performed using the proposed machine learning algorithms to see the effect of advertising on the performance of the models [46].

The accuracy of the demand forecasting predictions is evaluated using RMSE and MAE metrics. The best hyperparameter combination for each model is used to calculate the RMSE and MAE values. In order to understand the effect of advertising on the demand prediction performance of the models, the RMSE and MAE values for the low and high advertising expenses groups are also reported separately [46].

Thus, the results showed that all ML models had improved prediction performance when advertising expenses were high, with improvements ranging from 19% to 43% for the RMSE metric. LSTM showed the highest overall accuracy in predicting demand, performing well in both the low and high advertising expense groups, possibly due to its ability to store time-dependent information for long sequences. RFR was found to perform better when advertising expenses were high, while LSTM performed well regardless of advertising expense level. Overall, the study demonstrated the effectiveness of using neural network models, particularly LSTM, for demand forecasting in the context of advertising expenses [46].

Limitations of the study include the use of a dataset from a single TV manufacturer, a focus on ML techniques for demand forecasting, and the lack of comparison with traditional statistical methods [46].

Also, Di Pillo et al. [47] discusses the use of machine learning techniques, including artificial neural networks (ANNs), radial basis function (RBF) neural networks, and support vector machines (SVMs), for sales forecasting in retail stores. The authors compare the performance of

these machine learning methods to traditional statistical techniques, such as time series models, using real-world data from a retail store. The focus of the paper is on the effects of promotions on sales and how different input attributes can impact the performance of the machine learning models [47].

Accordingly, this study used data from two retail stores to compare the performance of machine learning techniques, including artificial neural networks (ANNs), radial basis function (RBF) neural networks, and support vector machines (SVMs), for sales forecasting. Store #1 has poor storage management and suffers from frequent stock-outs, leading to inconsistent sales data, store #2, on the other hand, has excellent storage management and experiences very few stock-outs. The input data included 13 attributes: 9 calendar attributes related to the specific day of the output (month, day of the month, and 7 boolean attributes representing the day of the week), a boolean attribute indicating whether there was a promotion on that day, the amount of time the store was open that day, and the daily cost of the product. The output data was the daily sales of a particular brand of pasta. The input-output data covered three years: 2007 and 2008 were used for training and validation, while 2009 was used for forecasting and testing [47].

Di Pillo et al. [47] also used a support vector machine to forecast the number of receipts per day, using 11 input attributes, including the 9 calendar attributes and the number of hours the store was open, as well as a final attribute indicating whether high or low sales were expected on that day. The authors carried out three experiments with different sets of input factors: the first experiment included four input factors (promotion, number of opening hours, price of the product, and forecasted number of daily receipts), the second experiment included 12 input factors (the four mentioned above plus nine calendar attributes), and the third experiment included all of the input factors from the previous two experiments, totaling 13 input factors. Then, compared the results to traditional time series models including ARIMA, exponential smoothing (ES), and Holt-Winter exponential smoothing (HWES) [47].

The results, measured in terms of MAPE, showed that among the machine learning methods, radial basis function neural networks with 4 input attributes performed the best. The authors also found that the machine learning methods generally outperformed the statistical methods, particularly for store #1, which had a history of poor stock management and irregular sales patterns. The results also indicated that the machine learning methods, particularly radial basis function neural networks, performed particularly well during promotion periods [47].

In addition, for store #2 the results showed that among the machine learning methods, support vector machines with 4 input attributes performed the best, while among the statistical methods, Holt-Winter exponential smoothing performed the best. The MAPE values for the machine learning methods were generally lower for this store than for the first store, due to the store's better stock management and more regular sales patterns. The authors also found that the machine learning methods, particularly support vector machines, performed particularly well during promotion periods [47].

Overall, Di Pillo et al. [47] found that both support vector machines and radial basis function neural networks performed better than other methods in 7 out of 20 tests, while multilayer artificial

neural networks performed better in 5 out of 20 tests. Statistical methods were found to be the best in only 1 out of 20 tests. Additionally, all three machine learning methods performed well, and the choice of input attributes was important in determining the accuracy of the forecasts [47].

Correspondingly, Abolghasemi et al. [6] aims to understand how demand volatility affects the effectiveness of forecasting models and how it can be modeled in the supply chain. The study uses data from a fast-moving consumer goods (FMCG) food company to evaluate the performance of various forecasting models in predicting volatile demand. The research focuses on the impact of price as a key explanatory variable on demand and proposes a hybrid regression time series model (HR-ARIMA) as well as other models from well-established methods in the literature, including ARIMAX, ETSX, DLR, ANN, and SVR. The goal is to identify the most suitable forecasting model for different levels of demand volatility [6].

Hence, to do this, Abolghasemi et al. [6] decomposed demand into two main factors: baseline demand, which is estimated using an ARIMA model, and promotional demand, which is the difference between total demand and baseline demand. The authors found that there is a strong linear relationship between promotional demand and price, and developed a piecewise regression model to capture this relationship. The model consists of two components: an ARIMA model to forecast baseline demand and a piecewise regression model to forecast demand uplifts during promotions. The piecewise regression model uses different coefficients for different promotional prices and assumes that the promotion price is known in advance [6].

The data includes consumer demand and price information for 112 weeks and covers 843 different stock-keeping unit locations (SKULs). The demand data shows significant differences between promotional and non-promotional periods. MASE is the criterion used to evaluate the accuracy of models employed [6].

ARIMAX performed well for low and relatively volatile products, DLR and SVR had a similar and robust performance across different values of CoV, ANN worked well for low volatile demand series but poorly for highly volatile demand series, and simple statistical models like ETS outperformed some of the more sophisticated models when the series was highly volatile [6].

Finally, Kumar et al. [48] suggested forecasting method incorporates big data and machine learning to estimate future demand and sales through a four-step procedure utilizing a fuzzy artificial neural network classifier. This assists forecasting managers in optimizing revenue, managing large amounts of data, increasing forecasting accuracy, analyzing the effect of advertising on sales, connecting demand and supply data, and optimizing marketing investment based on promotional activities and other marketing factors [48].

The first step involves collecting demand signals from collected downstream demand data and associating them with promotional marketing data in order to influence future demand using fuzzy ANN and comparing it against ARIMA, Multi Linear Regression, Random Forest, SVM, and ANN [48]. Second, the marketing effectiveness of each advertisement medium is determined using market-mix modeling (MMM) to shift demand based on a variety of capacity restrictions and other variables related to sales, marketing, and operations [48].

A statistical model is then created using sales as the dependent variable and marketing channels, advertising budget, seasonality, and macro factors as independent variables. The final demand response model is developed using what-if analysis and multi-tiered causal analysis to influence and sense the unconstrained demand (MTCA). MTCA is a decision support system that simulates the push-pull impact in a supply chain by linking different regression models with the common variable "sales." This system integrates supply and point-of-sale data, as well as advertising plans, into a unified framework. It connects demand and supply data employing two separate models. The first model anticipates sales based on information about advertising expenses and promotional activities, whereas the second model estimates shipments based on information about the time delay between point-of-sale data and other promotional data. After that, MTCA integrates demand with shipping or supply data to construct a demand forecasting model by including advertising campaigns, marketing channels, and seasonal data into the shipment forecasting process [48].

Additionally, the dataset comprises daily TV sales over a period of 372 weeks from 2010 to 2017. It includes information on demand, sales, supply, point of sale, advertising expenditure, and marketing effectiveness (measured in Gross Rating Points (GRPs)) for multiple channels, such as mobile SMS, newspaper ads, radio, TV, posters, and the internet. The dataset includes 2614 observations and 17 variables. GRPs are a measure of the reach and frequency of an advertisement campaign. GRPs are computed by multiplying the percentage of a target audience exposed to an advertisement by the number of times that the advertisement is aired. Results are shown in terms of MSE, MAPE, and MAD [48].

Finally, the suggested technique provides more suitable inferences for this particular big data-driven demand-driven prediction method than the benchmarked approaches and outperforms them when demand is linked to supply and demand shaping effects are taken into account. The suggested approach can assess huge data sets rapidly and anticipate future demand based on advertising budget, omnichannel, promotion, and sales data [48].

As a result, the presented studies employ parametric and non-parametric models to forecast sales under promotional or advertising events. However, the current state-of-the-art lacks studies that evaluate the effects of different promotions alongside deep interpretable models. This is a significant gap in the literature, as it is essential to understand the effectiveness of different promotional strategies to make informed decisions about marketing efforts. Additionally, using interpretable models or methods would allow for a better understanding of the underlying mechanisms driving the sales response to promotions, which could inform the design of future marketing campaigns.

Nonetheless, deep learning approaches such as LSTM and fuzzy neural networks are favored models for predicting sales with promotional offers. These models can accurately evaluate large and complex datasets, detect trends and patterns and generate reliable predictions about future sales. LSTMs are highly effective at interpreting time series data and can identify long-term correlations, making them well-suited for forecasting sales over extended periods. On the other hand, fuzzy neural networks can deal with ambiguity and imprecision in data, which is common

in sales forecasting situations.

Chapter 4

Methodology and Experiments Setup

4.1 Dataset Overview

The data analysed belong to the real demand observed in a chain of supermarkets for 2 years. It has multiple stores and a large volume of products belonging to different categories.

Additionally, the listed data sets provide insights into diverse categories of retail promotional activities.The information furnished consists of crucial details, including date, product code, store locality, and promotion category, effectively explaining how promotions affect overall sales figures. In essence, these datasets furnish an all-encompassing overview of the array of promotions and marketing events.

The data files can be described as follows:

- **"dpos"**: This file contains daily point-of-sale (POS) data, which includes the date, the SKU (Stock Keeping Unit) or product code, the store where the sale occurred, and the unit sales;

- **"pvarXLBOGO"**: retains information about Buy One Get One (BOGO) promotions, which offer customers a free item when they purchase another item. The file includes the date, the SKU or product code, the store where the promotion is running, and a Boolean value (true/false) indicating whether the promotion is currently active;

- **"pvarXLCIRC"**: This file includes data related to circular promotions, which are sales that are advertised in a store's weekly circular or flyer. The file consists of the date, the SKU or product code, the store where the promotion is running, and the discount value or percentage off;

- **"pvarXLCIRE"**: comprises data related to Circular EDLP (Everyday Low Price) promotions, which are promotions where a product is advertised at a low price for an extended period. The file includes the date, the SKU or product code, the store where the promotion is running, and a Boolean value (true/false) indicating whether the product is currently being offered at the advertised price;

- **"pvarXLCLCP"**: This file includes data related to clipping coupon promotions, which are promotions that require customers to present a coupon to receive a discount. The file consists of the date, the SKU or product code, the store where the promotion is running, and a Boolean value (true/false) indicating whether the promotion is currently active;
- **"pvarXLDISC"**: comprises data related to discount promotions, which offer a price reduction on a product for a limited time. The file includes the date, the SKU or product code, the store where the promotion is running, and the discount value or percentage off;
- **"pvarXLFPE"**: holds data related to Front Page EDLP (Everyday Low Price) promotions, which are promotions where a product is advertised at a low price on the front page of a store's circular or flyer. The file includes the date, the SKU or product code, the store where the promotion is running, and a Boolean value (true/false) indicating whether the product is currently being offered at the advertised price;
- **"pvarXLFRPG"**: This file comprises data related to Front Page promotions, which are sales that are advertised on the front page of a store's weekly circular or flyer. The file includes the date, the SKU or product code, the store where the promotion is running, and the discount value or percentage off;
- **"pvarXLGAS"**: data related to gas promotions, which offer customers discounts on gasoline when they purchase at a store. The file includes the date, the SKU or product code, the store where the promotion is running, and a Boolean value (true/false) indicating whether the promotion is currently active;
- **"pvarXLMB"**: presents data related to Must-Buy promotions, which require customers to purchase a specific product to receive a discount on another product. The file includes the date, the SKU or product code, the store where the promotion is running, and a Boolean value (true/false) indicating whether the promotion is currently active.

Furthermore, the store information is presented in the "loc" file, which contains a set of columns describing each store. These columns include the following attributes, which provide information about the stores:

- **str**: This column contains a unique numeric identifier for each store within the dataset;
- **str_label**: This label provides additional information about the store, such as its location;
- **dstr**: Comprises a code for the district or area where the store is located;
- **dstr_label**: Contains a readable label for the area where the store is located;
- **rgn**: This column presents a code or identifier for the region where the store is located;
- **rgn_label**: Contains a readable label for the region where the store is located;

- **div**: This column contains a code for the division where the store is located, a higher-level administrative unit that encompasses several regions or districts;
- **div_label**: Comprises a readable label providing additional information about its organizational structure;
- **bnnr**: This column contains a code for the retail banner that the store belongs to. The flag may be a brand or a franchise that operates multiple stores across different regions or divisions;
- **bnnr_label**: Provide additional information about the banner;
- **cmpn**: Company that owns or operates the store;
- **cmpn_label**: Readable brand or name for the company that owns or operates the store;
- **stt**: A code for the state where the store is located;
- **stt_label**: A readable brand or name for the state or province where the store is located.

Ultimately, the description of the items can be obtained from the file "prod", which encapsulates detailed information about each item available in the stores:

- **item**: Comprises a unique numeric identifier,SKU, for each item;
- **item_label**: Holds a comprehensive description for every item;
- **cat**: Delineates the category to which each item belongs;
- **brnd**: Encompasses an identifier specific to an item's brand.

4.2 Dataset Preparation and Cleaning

The data preparation and cleaning process played a crucial role in generating an accurate and comprehensive dataset suitable for analyzing the impact of promotional events on sales. This section outlines the various steps to preprocess and clean the data, ultimately producing a final data frame containing daily sales for each store item and the corresponding promotion information.

Initially, the individual files containing information about the different promotions were merged into a single data frame. This consolidated data frame enables more efficient analysis of the diverse promotional activities and their relationship with sales transactions. Each row in this combined dataset now represents a unique combination of date, store, product, and category, along with the respective active promotions for that particular day. To identify each time series, a new variable, 'unique_id', was created by combining the item and store numbers, allowing for a more organized and efficient dataset analysis.

Moreover, to enhance the dataset further, a new binary variable, 'no_promo', was introduced to signify whether any promotion was active on a specific day for a given item. This variable

allows for more intuitive analysis of the effect of promotions on sales, as it helps differentiate between days with and without active promotions. Since the initial raw data only included sales transactions, it did not account for days when a product was not sold. To address this limitation, missing dates were added to the dataset, assuming sales on those days were zero and no promotion was active. The resulting dataset spans from 23-01-2021 to 22-02-2023 (761 days), ensuring that all time series have the same data points.

The median and interquartile deviation were calculated for all time series to avoid using time series that include items with low sales throughout the time scope. The median serves as a measure of central tendency, providing insight into the middle value of a time series. In this context, a median equal to zero indicates that an item did not sell much during 50% of the period under consideration. The interquartile range (IQR) is also an essential measure in this case, as it provides information about the variability of the sales data within each time series. By calculating the IQR, we can identify the spread of sales values around the median.

Consequently, time series with a median equal to zero and an IQR up to a maximum of 2 were eliminated from the dataset, ensuring that the analysis focuses on items with more significant sales figures. Initially, there were 168,647 time series in the dataset; after removing the ones with a median equal to zero, we ended up with 12,830 time series, providing a more relevant and meaningful dataset for analysis.

According to Figure 4.1 and Table 4.1, the majority of the time series (155,817) have a median equal to 0 and an IQR up to a maximum of 2, while a much smaller number have a median between 0 and 5 (11,854), between 5 and 10 (543), or higher than 10 (433). This suggests that a significant portion of the time series data may not have many variations in their unit sales over time. On the other hand, there are also some time series with higher median values, which may indicate more variability or trends in their sales patterns.

	item_nbr	store	median	IQR
24	27	6100	7	4
41	36	1	1	2
45	36	6100	1	2
70	55	1	1	2
72	55	548	1	3
...	...	...	...	...
168505	900094131	88	0.449	2.210
168506	900094131	548	1.639	3.710
168524	900094261	1	1	2
168525	900094261	88	1	2
168593	900094664	1	2.769	4.279

Table 4.1: Summary statistics of selected time series: Median and Interquartile Range (IQR)

Finally, it is essential to note that the final dataset does not incorporate any additional information about store locations or product characteristics. Although these factors can significantly

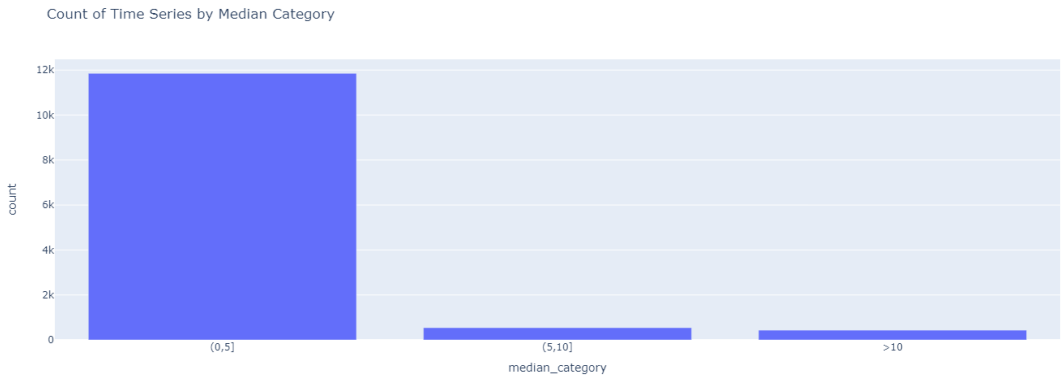


Figure 4.1: Count of time series grouped into three categories based on their median values.

impact consumer demand, the primary focus of this analysis is to understand the influence of promotional events on sales transactions. By concentrating on the relationship between promotions and sales, we can extract valuable insights into the effectiveness of various promotional activities and their role in driving sales performance.

4.3 Exploratory Data Analysis

This section will delve into the Exploratory Data Analysis (EDA) process to better understand the information contained in the dataset. EDA is essential in any data analysis project, as it helps uncover patterns, detect anomalies, and identify relationships between variables. By examining the dataset through various perspectives and employing diverse visualization techniques, we can gain valuable insights to guide subsequent analysis and modeling efforts.

First, we will examine the summary statistics of the dataset's key variables, such as sales and promotions. This will provide an initial understanding of the overall data structure and help reveal potential outliers, data entry errors, or unexpected trends.

In addition to the summary statistics in the table 4.2, a pie chart is provided to visualize the percentage of occurrences of various promotions over the entire dataset in figure 4.2. The chart shows that discount promotions occur 44.8% of the time, circular promotions occur 28.9%, and front-page discounts occur 10%. Circular EDLP promotions are active at 6.73% of the time, and front-page EDLP promotions occur at 4.11%. The remaining promotions, such as buy-one-get-one, gas promo, and coupons, make up 5.43% of occurrences. Also, based on the dataset analysis, it can be concluded that there are no instances of multiple promotions occurring for the same item on the same day. This means that only one promotion occurs on a given day for a specific item. As a result, each item is exclusively featured with a single promotion at any given time.

	unit_sales	circular_discount	discount_value	fp_discount
count	9.763e+6	9.763e+6	9.763e+6	9.763e+6
mean	3.358	0.014	0.017	0.006
std	11.865	0.062	0.062	0.045
min	0	0	0	0
25%	0	0	0	0
50%	1	0	0	0
75%	3	0	0	0
max	1374	0.751	0.947	0.716

Table 4.2: Summary statistics of sales data: The dataset consists of approximately 9.76 million observations. On average, unit sales are around 3.36, with a standard deviation of 11.87. The maximum number of unit sales recorded is 1374.

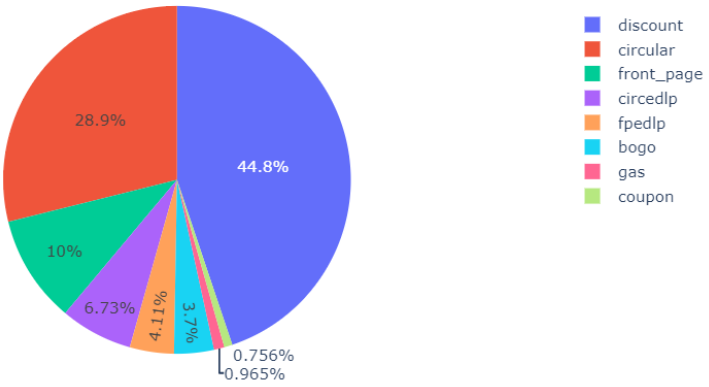


Figure 4.2: Promotional Events

Moreover, the dataset encompasses a variety of product categories, each representing a specific range of products in the stores:

DELI : This category includes ready-to-eat foods such as meats, cheeses, sandwiches, salads, and various side dishes;

FROZEN : This category encompasses all items stored in the freezer section. It includes various products, from frozen fruits and vegetables to meals, pizzas, desserts, and ice cream;

GENERAL MERCH : This category typically includes non-food items such as cleaning supplies, paper products, and clothing;

HBC : Standing for "Health and Beauty Care", this category includes personal care products like shampoo, soap, skincare items, cosmetics, and wellness products;

MEAT : This category represents all kinds of fresh meat products;

PRODUCE : The produce category encapsulates fresh fruits, vegetables, and pastry products.

These categories reflect the variety of items available in the stores, showcasing a broad offering to diverse customer needs and preferences. Figure 4.3 presents a clear view of unit sales across various product categories. Notably, DELI, FROZEN, GENERAL MERCH, and HBC products share a median unit sale of one per day. Meanwhile, MEAT products exhibit a slightly higher median, with two units sold daily. The most popular category, however, is PRODUCE, with a standout median sale of three units per day. This pattern suggests that PRODUCE items are the most frequently sold products across the stores, reflecting customers' preferences or demands.

Next, we analyze the temporal patterns in the data. By visualizing the time series of daily sales, we can investigate seasonality, trends, and any significant fluctuations associated with promotional events, illustrated in figures 4.4 and 4.5.

The decomposed time series in both figures showcase an evident seasonal pattern, repeating every seven days. This seven-day cyclical pattern in retail sales often suggests a weekly pattern in customer purchasing behaviors. For instance, consumers might make most of their purchases on weekends due to free time or on weekdays due to promotional events. A regular weekly cycle such as this can guide retailers to plan their stock levels, sales strategies, and promotional events accordingly.

Furthermore, our time series decomposition does not hint at any specific trend in the data. The absence of a trend means that overall sales do not show a discernible long-term increase or decrease, suggesting a stable demand for the product category or that the effect of any growth factors (such as an increase in customer base or product popularity) is being offset by shrinkage factors (such as seasonal slumps or increased competition). It could also mean that any potential trend is too weak or erratic to capture by our decomposition model.

Additionally, we explore the relationship between the different promotion types and their respective impact on sales. This involves examining the distribution of deals across various promotion categories and assessing each promotion type's effectiveness in driving sales performance.

Therefore, the boxplot analysis in figure 4.6 reveals that promotions, in general, have a positive impact on sales. Particularly, the buy-one-get-one, coupons, front page EDLP, and gas promotions show a noticeable effect on sales despite being the promotional events that occur less frequently than other promotional events. These promotions seem to be more effective at attracting customers and boosting sales. On the other hand, discount and circular discount promotions appear to have a less evident impact on sales. Although these promotions occur more frequently than other promotional events, their effect on sales is not as pronounced. This finding could suggest that consumers might not find these discounts appealing, or they may not provide a sufficient incentive for increased purchasing behavior.

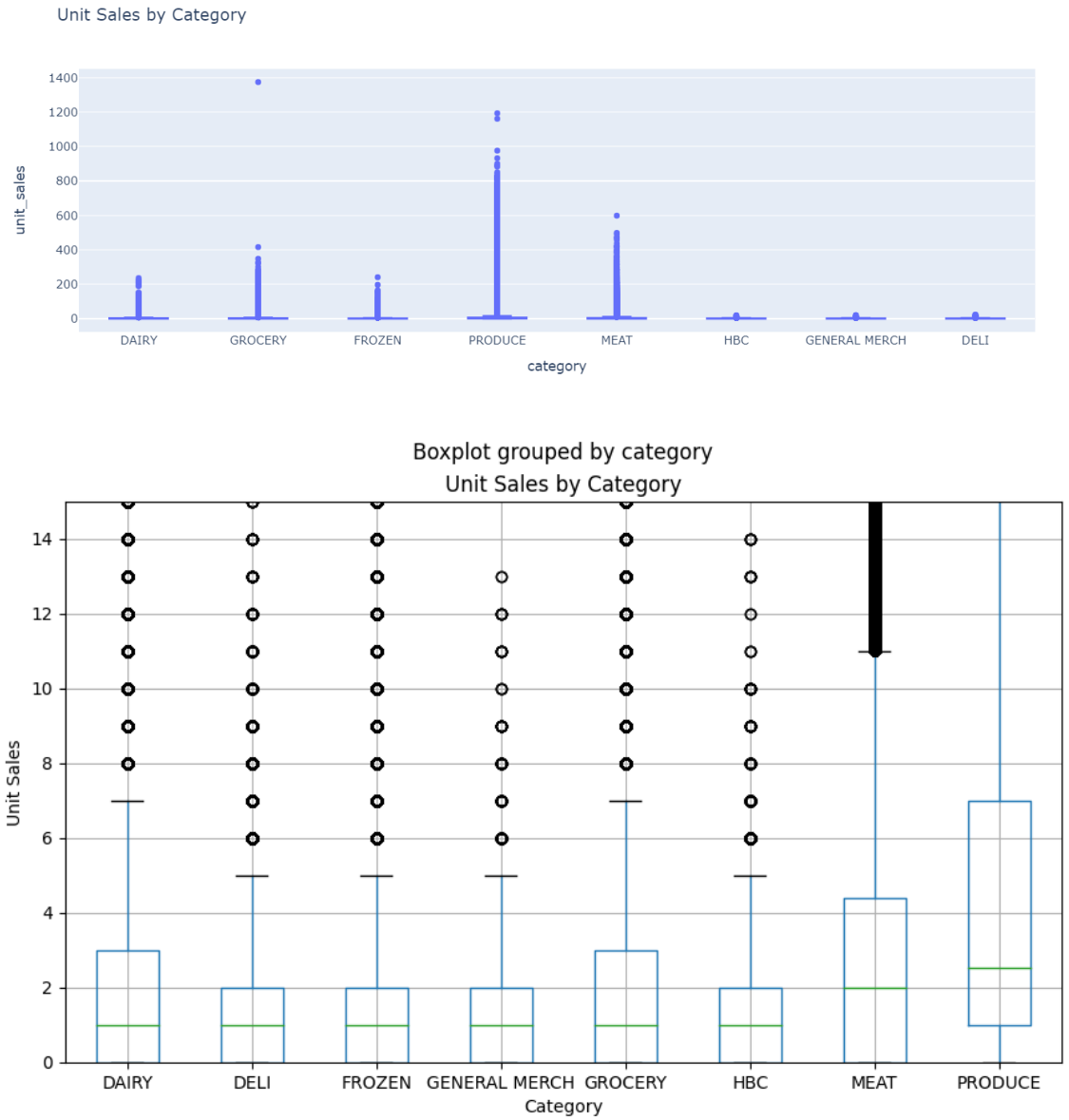
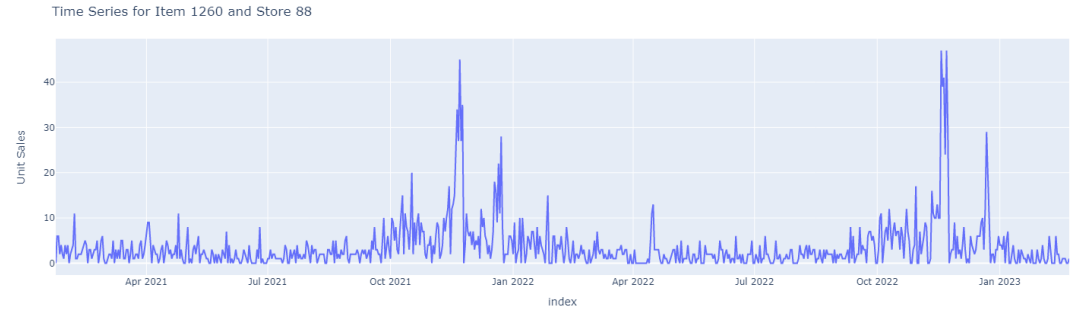
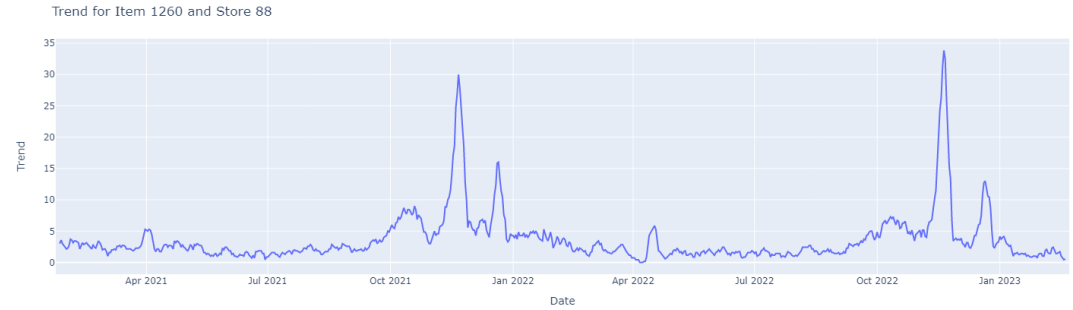


Figure 4.3: Unit sales per category of products

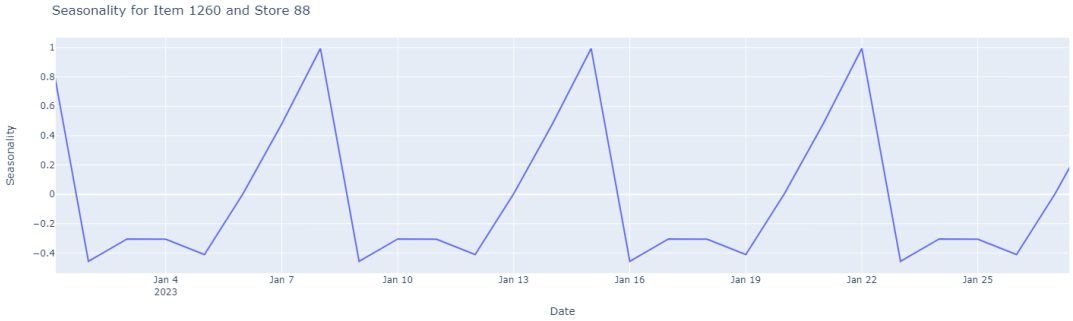
Figure 4.7 depicts the distribution of promotional events across various product categories over two years. The Grocery category stands out, amassing approximately 1.2 million promotions, marking it as the most promoted category within the given timeframe, with the overall promotional strategy in this category being discounts. Dairy, Produce, Meat, and Frozen categories are followed by the Grocery category. These categories also exhibit a significant number of promotions, with discounts once again being the predominant promotional event. In contrast, the Deli, General Merchandise, and Health and Beauty Care (HBC) categories were found to have the least promotional activity. This discrepancy might be attributable to the nature of these categories or consumer purchasing behaviors that differ from those associated with the heavily promoted



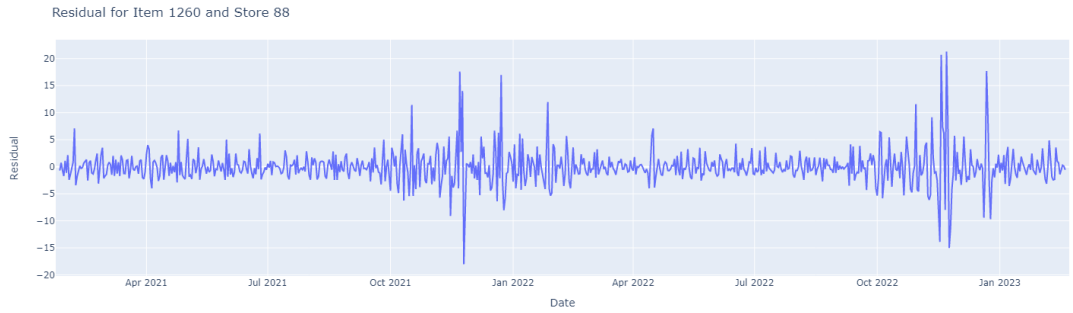
(a) Time series for item 1260 from store 88



(b) Trend



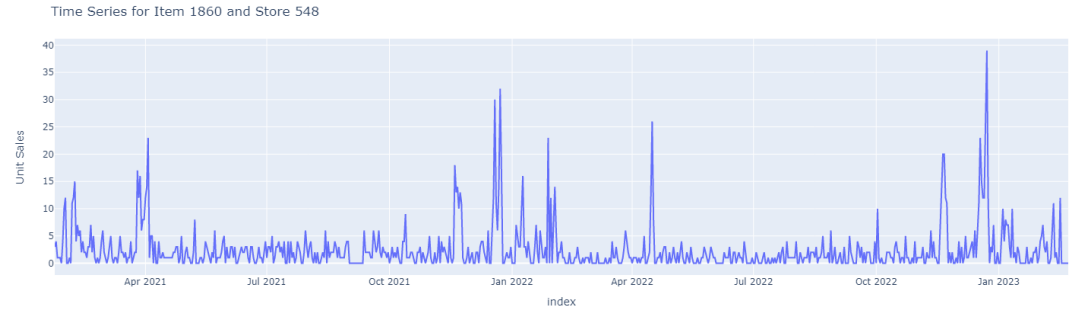
(c) Seasonality



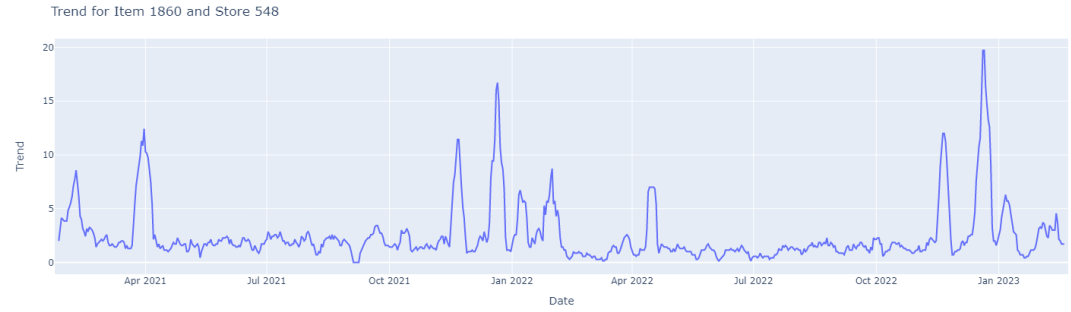
(d) Residuals

Figure 4.4: Time series analysis for item 1260 from store 88

categories.



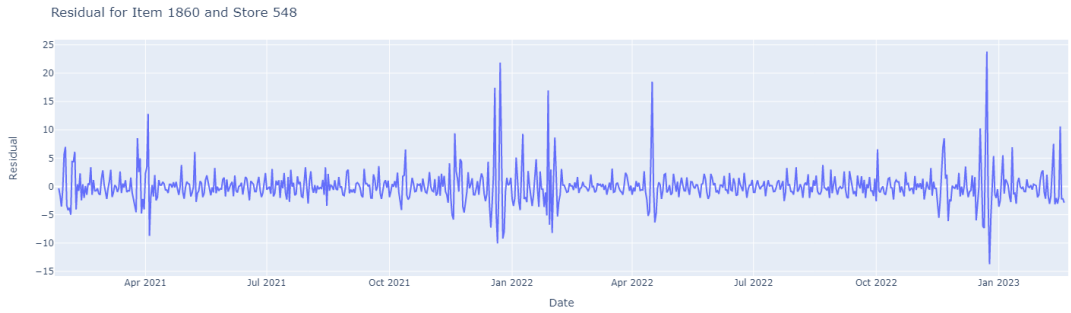
(a) Time series for item 1860 from store 548



(b) Trend



(c) Seasonality



(d) Residuals

Figure 4.5: Time series analysis for item 1860 from store 548

Additionally, Figure 4.8 illustrates the impact of different discount ranges on overall sales; discounts within the 50% to 80% range tend to impact sales more than other intervals positively. This

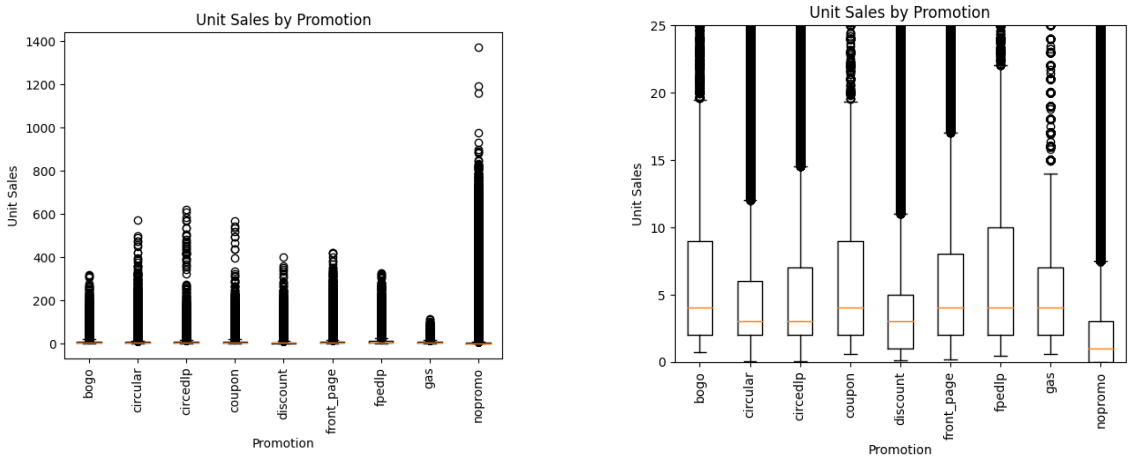


Figure 4.6: Unit sales by promotion type

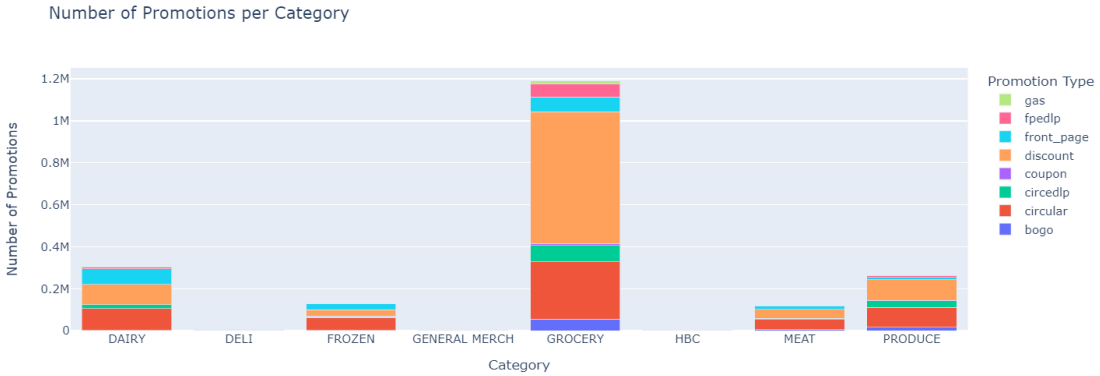


Figure 4.7: Unit sales by category

suggests that values within this range make consumers more likely to be enticed. Furthermore, circular promotions with discounts between 60% to 80% are more effective in driving sales than promotions with other discount values. Finally, in Figure 4.9, we observe the impact of different discount ranges on the effectiveness of front-page advertisements- discounts within the 50% to 70% range impact sales more than lower discount values. However, the most substantial positive impact is observed for front-page promotions with discounts in the 70% to 80% range.

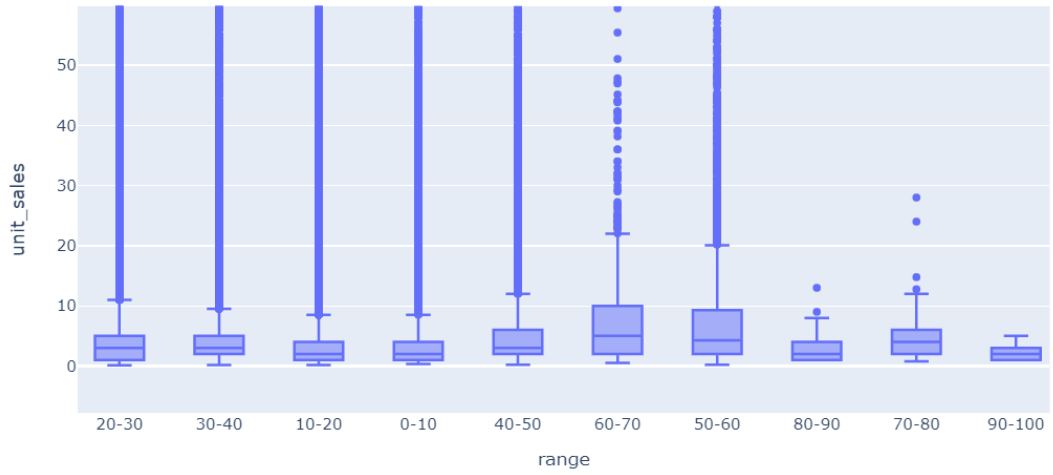


Figure 4.8: Unit sales by an interval of discount

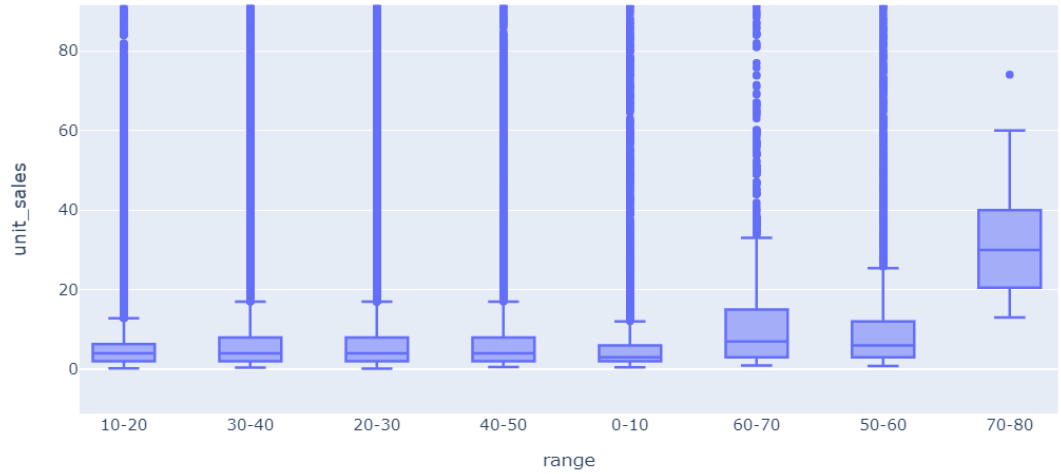


Figure 4.9: Unit sales by an interval of discount on front-page advertisements

From Figure 4.10 through Figure 4.17, we systematically examine the influence of various promotional methods on unit sales across different product categories. This category-by-category analysis provides a competitive advantage in terms of granular understanding and precision targeting of marketing efforts tailored to the distinct characteristics of each category.

Figure 4.10 shows that three promotional strategies, namely front page everyday low price, circular everyday low price, and buy one get one, have a noticeably more significant impact on unit sales in the Deli category than other promotional mechanisms.

The Dairy category, as displayed in Figure 4.11, shows distinctive promotional preferences. In this category, coupon and buy-one-get-one promotions notably outperform the others, whereas, interestingly, discount-based promotions are less impactful in this category.

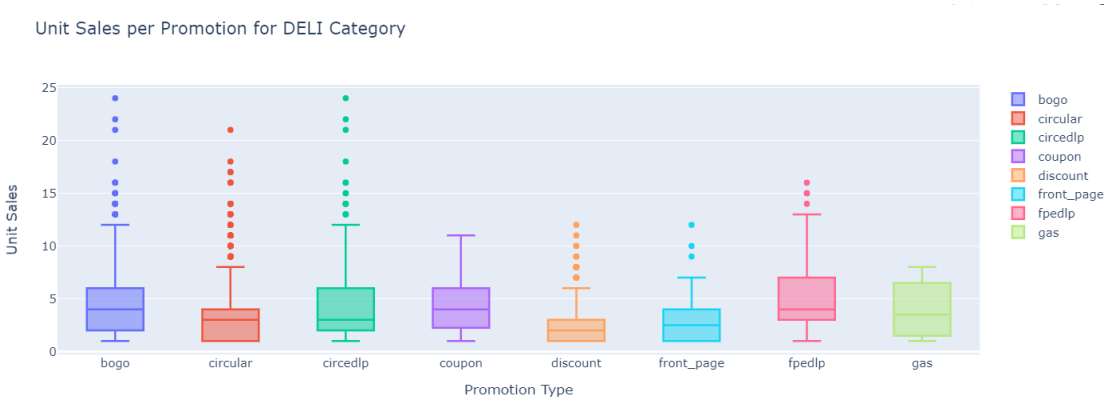


Figure 4.10: Unit sales by promotion on DELI category

Figure 4.12 highlights the Frozen category, where gas promotions yield the most substantial sales impact, followed by front page and coupon promotions.

Moreover, fewer promotional events are deployed in Figure 4.13, dedicated to General Merchandise. Specifically, the buy one gets one, circular, circular everyday low price, discount, and front page promotions are employed, with the latter having the most significant impact on unit sales.

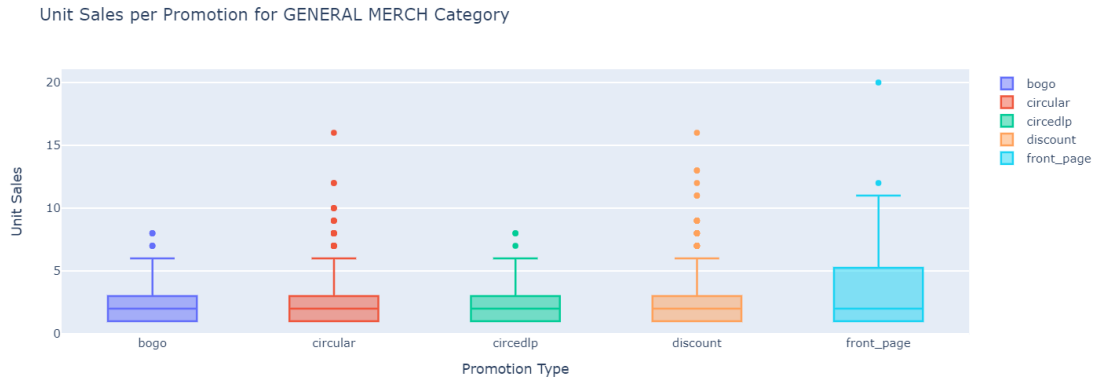


Figure 4.13: Unit sales by promotion on GENERAL MERCH category

Turning to the Grocery category in Figure 4.14, the most promoted category across the stores, we observe that all promotional activities similarly impact sales. Nevertheless, the front page, front page everyday low price, and gas promotions stand out, having a marginally higher impact on sales in this category.

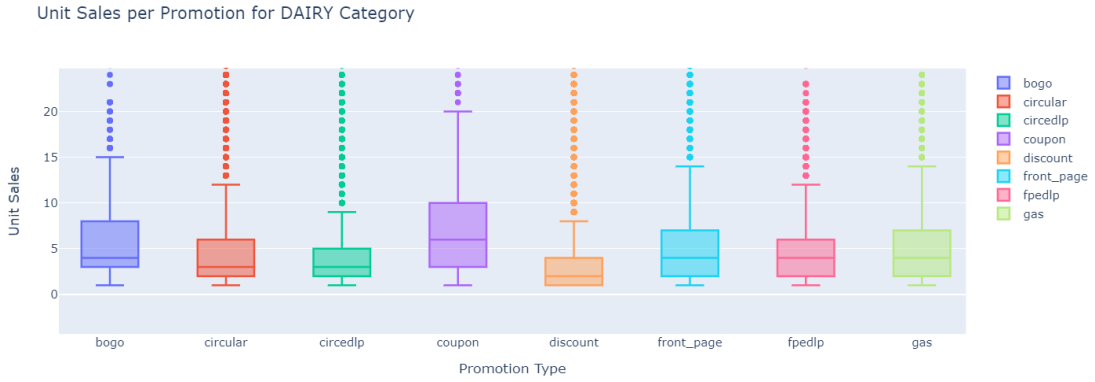


Figure 4.11: Unit sales by promotion on DAIRY category

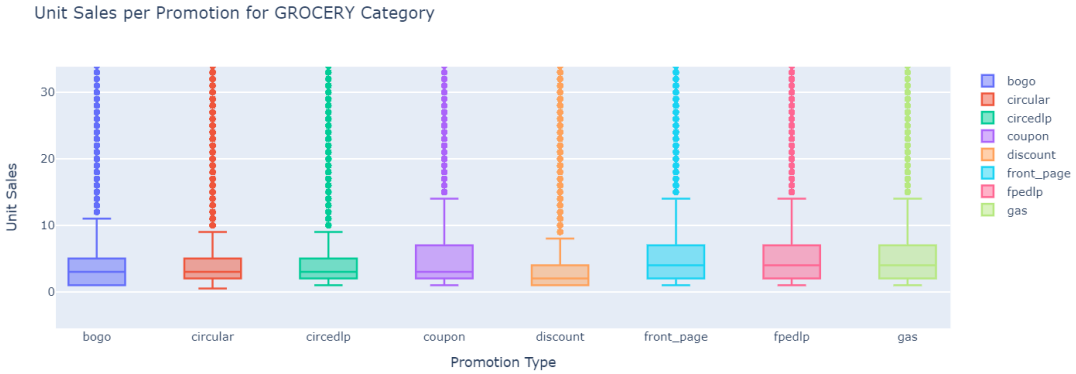


Figure 4.14: Unit sales by promotion on GROCERY category

In Figure 4.15, which corresponds to the Health and Beauty Care category, it is seen that among all promotions, the front page everyday low-price promotion has the most significant impact on unit sales.

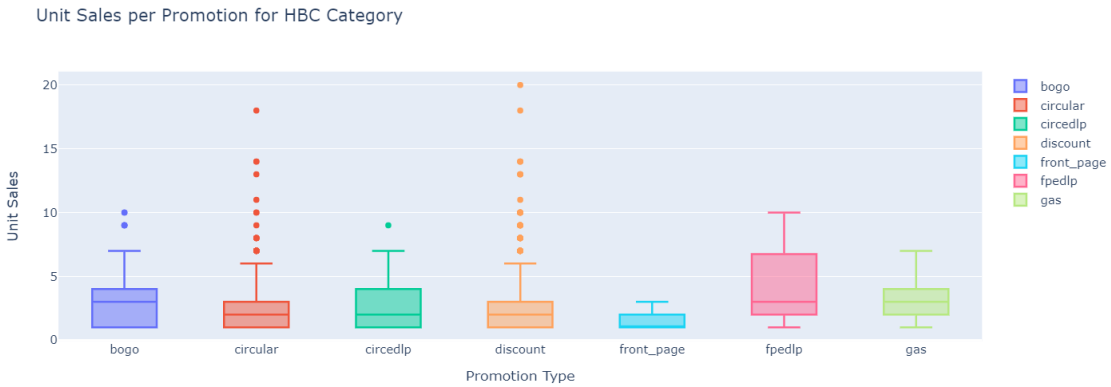


Figure 4.15: Unit sales by promotion on HBC category

Figure 4.16 offers insights into the efficacy of promotional events in the Liquor category, where

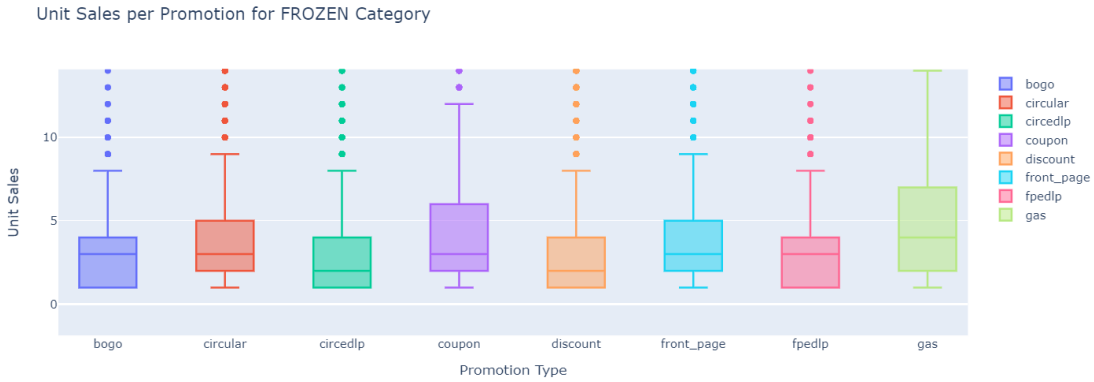


Figure 4.12: Unit sales by promotion on FROZEN category

coupon and front-page promotions register the most positive impact on sales, while, interestingly, discount promotions show the least influence among the promotional events considered.

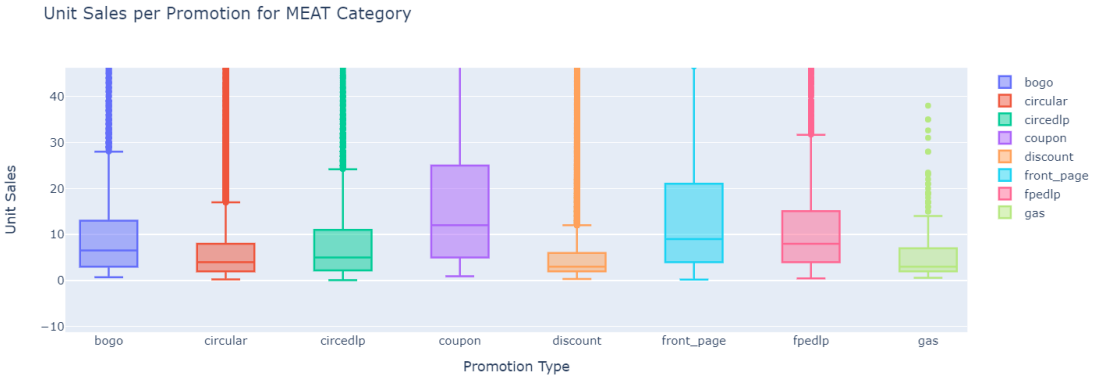


Figure 4.16: Unit sales by promotion on MEAT category

Finally, Figure 4.17 presents the Meat category, where the coupon promotion holds the top impact on unit sales, boasting a median value of 50 unit sales. The buy one closely follows this get one and front page everyday low-price promotions, demonstrating a similar effect on consumer behavior.

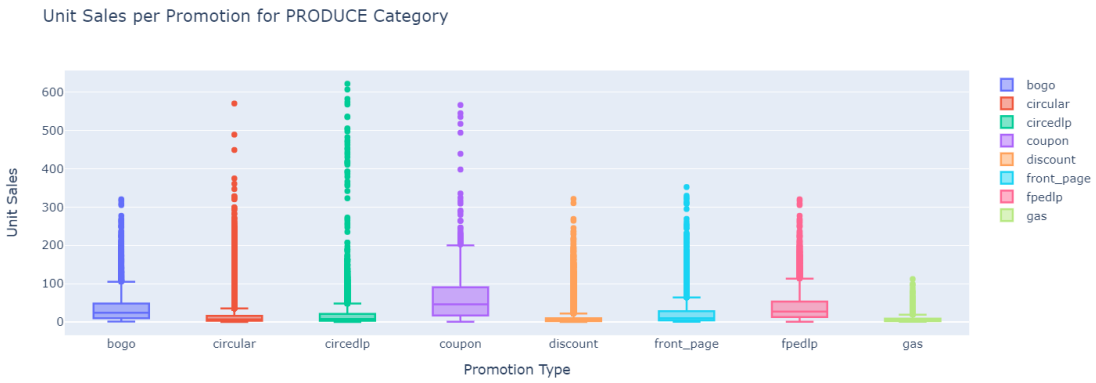


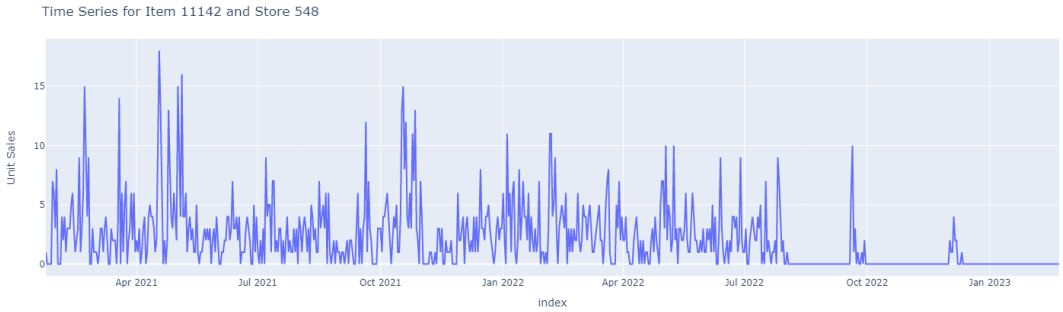
Figure 4.17: Unit sales by promotion on PRODUCE category

Moreover, we also examined the autocorrelation function (ACF) and partial autocorrelation function (PACF) of the time series data, as illustrated in figures from 4.18 to 4.19. Plotting the ACF and PACF is an essential step in understanding the structure and dynamics of the data, mainly when working with time series. The ACF calculates the correlation between data points at different time lags, revealing any underlying patterns or trends in the data. Meanwhile, PACF measures the correlation between data points at different time lags while controlling for the effect of other time lags. This helps to identify which time lags are most influential in predicting future values. Analyzing the ACF and PACF plots is crucial for several reasons:

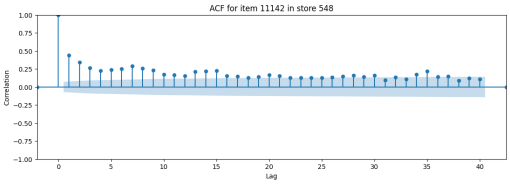
- It helps to identify the presence of seasonality and cyclical patterns in the data, which can be helpful in forecasting and improving sales strategies;

- It assists in determining the appropriate order of autoregressive (AR) and moving average (MA) components for time series modelings, such as ARIMA or SARIMA models;

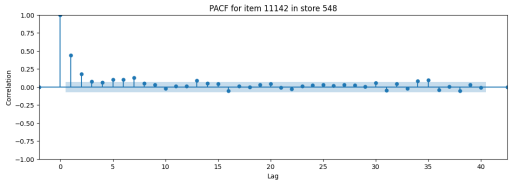
- Understanding the correlation structure in the data can help detect potential multicollinearity issues and inform model selection and evaluation.



(a) Time series for item 11142 from store 548

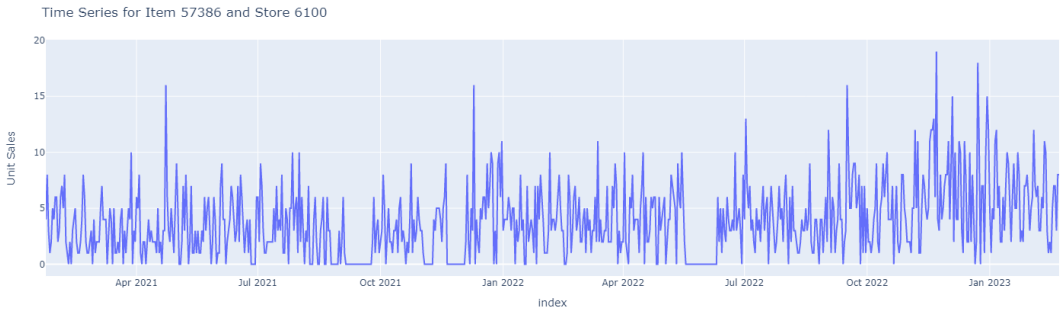


(b) ACF

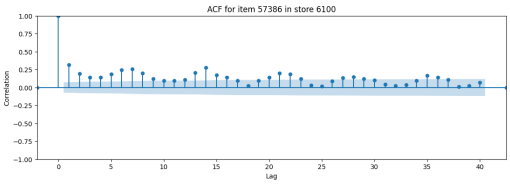


(c) PACF

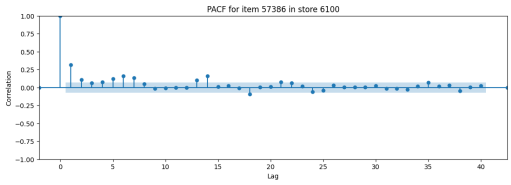
Figure 4.18: Time series, Autocorrelation (ACF), and Partial Autocorrelation (PACF) for item 11142 from store 548



(a) Time series for item 57386 from store 6100

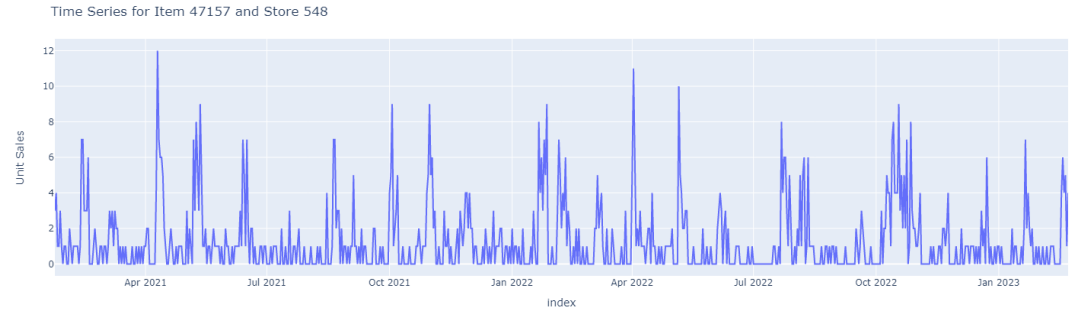


(b) ACF

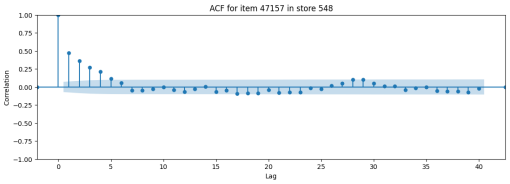


(c) PACF

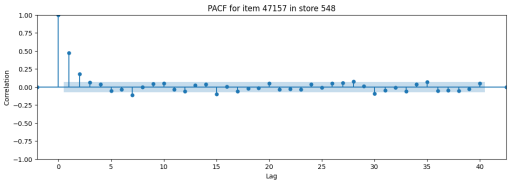
Figure 4.20: Time series, Autocorrelation (ACF), and Partial Autocorrelation (PACF) for item 57386 from store 6100



(a) Time series for item 47157 from store 548

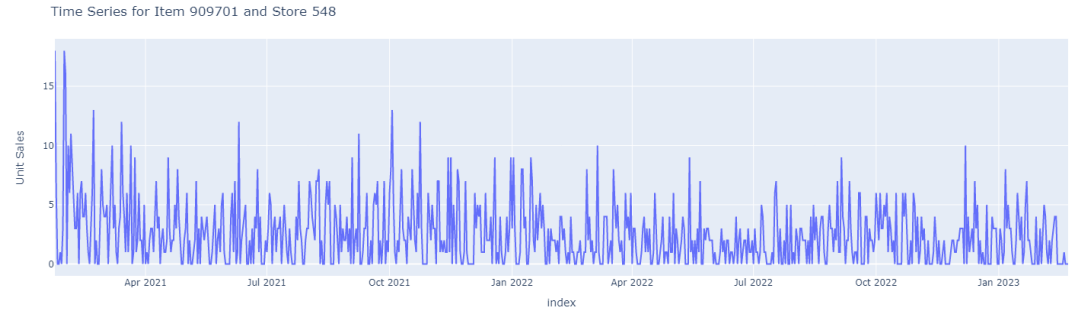


(b) ACF

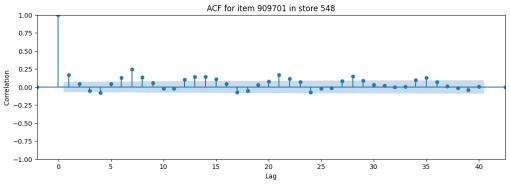


(c) PACF

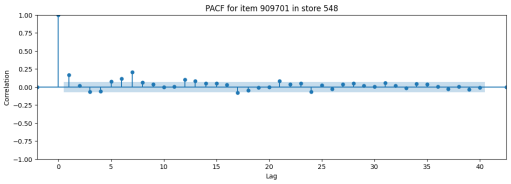
Figure 4.19: Time series, Autocorrelation (ACF), and Partial Autocorrelation (PACF) for item 47157 from store 548



(a) Time series for item 909701 from store 548



(b) ACF



(c) PACF

Figure 4.21: Time series, Autocorrelation (ACF), and Partial Autocorrelation (PACF) for item 909701 from store 548

From the analysis of the ACF and PACF plots, we observed varying lags of significance across different datasets. For instance, in figure 4.18, the ACF cutoff was observed at lag 15 and PACF at lag 3. In figure 4.19, ACF and PACF cutoffs were noticed earlier, at lag 4 and 2, respectively. For figure 4.20, the ACF cutoff is at a much later lag of 21, and the PACF cutoff is at 14. Lastly, in figure 4.21, ACF and PACF exhibit cutoff at lag 7. These observations provided critical insights into

the nature and structure of the various time series datasets. Despite the differences in cutoff points for the multiple plots, a common lag pattern became apparent from the analyses. In particular, a 14-day lag consistently held significant partial autocorrelation across the different datasets.

Therefore, based on this analysis, incorporating a 14-day lag as a feature in our time series models is expected to enhance their predictive performance. This choice is justified by the observed significant partial autocorrelation at the 14-day lag, indicating that there's meaningful information from the past 14 days that could improve our future predictions. This approach acknowledges the inherent autocorrelation within our time series data and allows the models to leverage this information, potentially leading to more accurate and robust predictions.

Finally, three critical statistical measures have been calculated for each series: entropy, Fourier entropy, and skewness. These measures are illustrated in the accompanying figures, and their significance is discussed below.

Entropy is a measurement of the randomness in a time series. A higher entropy value implies that the series is more complex and challenging to forecast, while a lower entropy indicates a more structured and predictable pattern. By calculating the entropy for each series, we can assess the inherent complexity of the time series data and better understand the challenges we might face while forecasting demand in the presence of concurrent promotions. According to Figure 4.22, the maximum value of 0.995 suggests that some of the time series may have highly irregular and unpredictable patterns, whereas the minimum value of 0.352 suggests that some other time series may be more predictable and follow more regular patterns. The median value of 0.904 indicates that most of the time series are moderately complex and have reasonable levels of unpredictability.

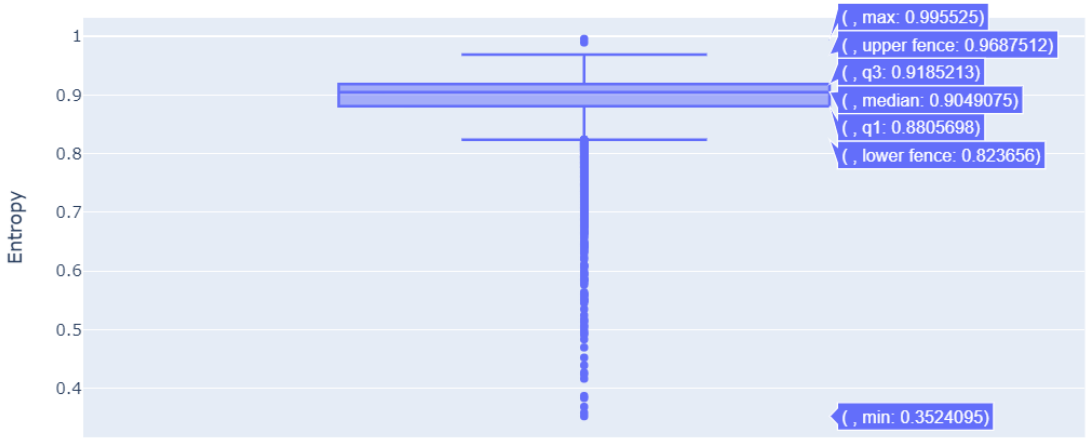


Figure 4.22: Entropy

On the other hand, Fourier entropy focuses on the frequency domain and measures energy distribution across different frequency bands in the time series. This provides insights into the periodicity and cyclical patterns present in the data. A higher Fourier entropy indicates that the series contains a mix of various frequency components, which may complicate the forecasting task. Understanding the frequency characteristics of the time series data is crucial for selecting the

most appropriate deep learning model to capture and forecast these patterns effectively. Hence, as shown in Figure 4.23, the maximum value is 2.229867, the median is 1.480446, and the minimum is 0.090. This suggests that the time series vary widely in complexity and irregularity, with some exhibiting highly irregular patterns and others being more regular and predictable.



Figure 4.23: Fourier Entropy

Ultimately, skewness is a measure of the asymmetry in the distribution of the time series data. A positive skewness value implies that the series has a longer tail on the right side, indicating a higher likelihood of extreme values above the mean. Conversely, a negative skewness value indicates a longer tail on the left side, implying a higher probability of extreme values below the mean. Analyzing the skewness of the series helps us understand the distribution and potential outliers in the data, which can be crucial for accurately modeling and forecasting the impact of concurrent promotions on demand- Figure 4.24

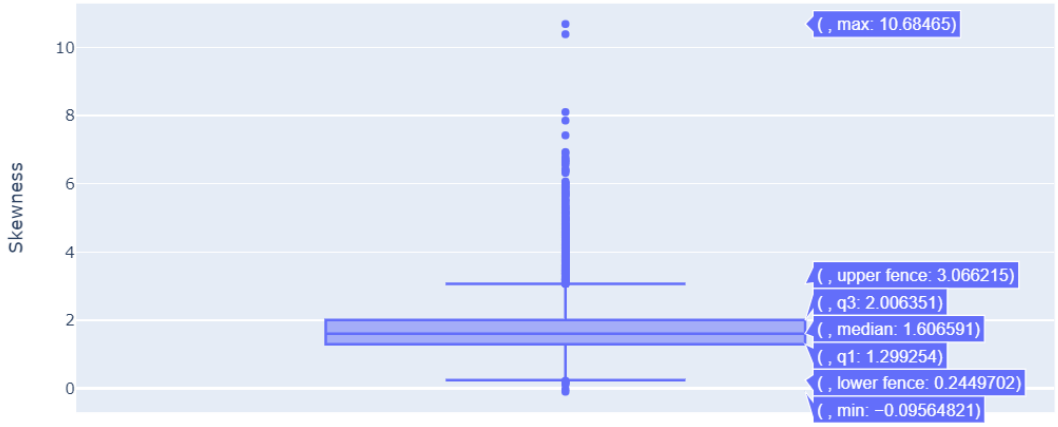


Figure 4.24: Skewness

4.4 Feature Engineering

4.4.1 Categorical Variables Encoding

Some Machine Learning, particularly concerning Deep Learning models, necessitates the numerical representation of all input variables. This means that categorical variables, denoted by labels or strings, must be converted into numerical format. The importance of this process becomes particularly evident when applied to the column referring to promotional events, which has information on the promotions that were active for a specific day for a particular item. These categorical labels provide valuable information for predicting various outcomes, such as sales numbers or customer behavior, which these promotional events could significantly influence. However, by converting these labels into a numerical format, Deep Learning models are able to interpret and use them. This transformation is known as can be accomplished in several ways. In this context, three methods will be highlighted:

- Ordinal Encoding assigns each category a unique number. These numbers often begin with 0 or 1 and may be readily reversed. Ordinal Encoding is appropriate for encoding categorical data having naturally ranked or ordered categories. Such integer correlations are straightforward, and Machine Learning models may easily use them. However, while employing this strategy, care should be used because it might artificially generate associations between unordered categorical data, potentially leading to false assumptions by Machine Learning models;

- One-Hot Encoding: When categorical variables lack intrinsic order or links among their values, ordinal encoding may establish an artificial order accidentally. This has the potential to negatively impact Machine Learning models by allowing them to infer non-existent relationships between categories, resulting in erroneous findings. One-Hot encoding becomes more beneficial in such instances, as it generates a binary variable for each distinct category, with the binary variable corresponding to one category set to 1 and the others set to 0, as shown in Figure 4.25. While this approach works well with unordered types, it can pose problems with high-cardinality categorical variables. This might result in an influx of binary features, which would have an impact on the model's speed and efficiency;

- Dummy Variable Encoding is a One-Hot Encoding modification that reduces redundancy. While One-Hot Encoding generates a binary variable for each category, this causes duplication because one category's binary state may be derived from the conditions of the others. Dummy Variable Encoding avoids this by using C-1 binary variables to represent C categories of a category variable. Despite its advantage in minimizing redundancy, it suffers from the same drawback as One-Hot Encoding when dealing with a large number of distinct categories.

Hence, One-Hot Encoding was employed to represent promotional events. Despite the risk of an increase in binary features, this method was chosen to maintain the presence of all promotional



Figure 4.25: Example of One-Hot Encoding to represent promotional activities

events in the dataset. The main reason for this choice was to facilitate interpretability analysis and conducted post-model training and testing. It's important to note that while interpretability may be enhanced, the drawback of potential model performance issues due to high feature count should be carefully considered and monitored.

4.4.2 Scaling

Being unitary sales of each item at every store as the objective variable, the procedure of scaling this variable, and typically any numeric feature, is a key pre-processing step when it comes to forecasting. Scaling reduces the range of numerical characteristics to a common scale without distorting the differences in value ranges or misplacing information. It is possible to achieve this by approaches such as normalization (scaling to a range between 0 and 1) or standardization (scaling to a mean of 0 and a standard deviation of 1). In this case, scaling 'unit_sales' is important due to several reasons:

- Variable Ranges: The dataset variables have a variety of ranges. On certain days, for example, 'unit_sales' is in the hundreds or thousands. Because variables with greater values may dominate the learning process, this size disparity might bias or inefficient the learning process;
- Algorithm Optimization: Many machine learning algorithms converge faster when features are on an equivalent scale;
- Model Performance: To determine likeness, some models use distance metrics. Without scaled data, bigger values will have a greater influence on the distance between the data points.

- Improved Interpretability: When features are on the same scale, it becomes easier to comprehend their relative importance, and thus interpret the model.

For forecasting models, scaled data is essential as it allows the model to capture trends and patterns in the data more efficiently, preserving the consistency of data interpretation by the model.

After considering the implication of scaling the unit sales variable and taking into account the distinct ranges in this dataset, the Min-Max Scaler method was chosen and applied to the 'unit_sales' column. This scaler is a type of normalization that transforms the data to fall within the range of 0 and 1. This is done by subtracting the minimum value of the feature and then dividing it by the range of the feature.

The Min-Max scaler is helpful in this scenario once it preserves the shape of the original distribution and doesn't reduce the importance of outliers. The impact of the minimum and maximum values becomes more profound, which aligns well with the requirements of several machine learning algorithms that require data on a similar scale. Hence, scaling 'unit_sales' using Min-Max Scaler is a critical step to ensure that the forecasting model can learn efficiently, optimize its performance, maintain consistent interpretation of data, and eventually yield reliable and interpretable forecasts.

4.4.3 Lags

Incorporating lags into a dataset is a critical step in time series analysis and forecasting. This practice accentuates the temporal relationship among observations, enabling to capture of the temporal dynamics that are often exhibited in time series data. Such data often demonstrate temporal dependency, where current observations may be influenced by prior ones. Lagged variables serve to capture recurring patterns or seasonality over time. For instance, sales might peak during weekends or specific seasons of the year. Incorporating lagged variables that span the length of a seasonal period facilitates the learning of these patterns by the model.

Furthermore, lagged variables allow the model to account for the influence of external events, which may not exert an immediate impact, yet could shape future observations. A 14-day lag has been applied in this context. This signifies that for each data point in 'unit_sales', the respective lagged data for the preceding 14 periods have been incorporated as features in the model. These features range from 'unit_sales_lag_1' to 'unit_sales_lag_14'. The selection of a 14-day lag is significant, especially useful when weekly seasonality or recurring patterns over a fortnightly period are observed in the 'unit_sales' data. This can relate to pay cycles, purchasing habits, promotional activities, or other fortnightly business-specific factors. The 14-day lag equips the model to learn from the sales trends of the past two weeks and use this information to predict sales in the current period. It can account for any immediate effects, weekly cyclical effects, and any impacts that might span a fortnight.

This method empowers the forecasting model to accurately understand the data's temporal dependencies and cyclical nature. As a result, the model's ability to generate precise forecasts may be significantly enhanced.

4.5 Models and Evaluation

The analysis embraces numerous attributes of retail items and their related promotional events. Although promotions are strategically planned and vary over time, their historical records are crucial to our data analysis. The development of advanced forecasting models becomes instrumental when integrating the evolving data of these promotional events. By consolidating information across multiple time series corresponding to different promotional events, these models can utilize this dynamic data to predict future customer demand with improved precision.

Such models learn from many time series, leveraging the temporal patterns of sales and promotional data and the intricacies of the ever-changing promotional events. Doing so enhances their predictive capacity, providing a more comprehensive view of sales forecasting. This process paves the way for a more accurate and functional prediction.

By factoring in the dynamic features of promotional history, these forecasting models can derive insights from the temporal dynamics and the evolving characteristics of promotional events. This approach offers a more thorough understanding of the impact of concurrent promotions on customer demand, hence contributing to more informed decision-making in retail.

4.5.1 SARIMAX

The SARIMAX model is used to forecast sales for each time series. The SARIMAX model considers the parameters for the regular ARIMA model (p,d,q) and the seasonal ARIMA model (P,D,Q,s). Moreover, the code incorporates external variables related to promotional activities, which may affect sales, and lags of the last 14 days of sales as exogenous variables. Including these variables improves the model's accuracy in capturing the effects of promotions and other factors on sales. The pmdarima library considers these exogenous variables and automatically selects the best SARIMAX model parameters based on the Akaike Information Criterion (AIC). This approach ensures the model is optimized for each time series and provides accurate forecasts.

4.5.2 XGBoost

The XGBoost technique is an ensemble learning approach that integrates numerous models' forecasts to create a final prediction. Therefore, it constructs a series of decision trees and optimizes them using a technique known as gradient boosting. This includes iteratively adding new models to fix the errors generated by the existing ensemble, hence boosting the model's predictive potential. This comprises prediction trees, model parameters that handle the learning process, and a loss function that computes the difference between the model's prediction and the actual result.

A sliding window method is employed to enhance the model's effectiveness and accuracy in forecasting unit sales. This method uses a 7-day window to make predictions for the subsequent day. The sliding window method treats the dataset as a rolling window, allowing for the dynamic selection of training and testing data based on a specified time window. This way, the model is retrained with the most recent data, making it more adaptive to any changes in the pattern or trend.

By combining the XGBoost technique and the sliding window method, the model will be able to develop a predictive model in the context of anticipating unit sales. It will learn complex mappings from inputs to the target variable using the features in the dataset. This method can identify complicated patterns and relationships, resulting in dependable, accurate forecasts that may be used to drive strategic decision-making.

After generating the model and making the predictions, the next step is determining which features significantly influenced the forecasts. XGBoost has a built-in feature that allows generating of a feature importance plot. This plot ranks the features based on their importance, showing which ones have the most substantial impact on the model's predictions. It's an effective way to gain insight into the model and understand which elements influence the predicted output most.

By comparing the results of the feature importance plot results and the SHAP library, we can gain an in-depth understanding of how different variables interact and influence the model's predictions. This comparison provides a comprehensive perspective, ensuring our model's interpretability and enabling us to make more informed and data-driven decisions.

4.5.3 LSTM

The LSTM model utilized in this project considers the time series data and external variables that may impact sales, such as promotional activities. Additionally, it considers the lags of the last 14 days of sales for each time series as exogenous variables. The model uses a 7-day sliding window method to create observations, and the other columns are added as explanatory variables. The model architecture consists of two LSTM layers, followed by a RepeatVector and TimeDistributed Dense layers, and it is optimized using the Adam optimizer and Mean Squared Error (MSE) loss function. LSTM is particularly advantageous for this problem as it can capture long-term dependencies, critical in forecasting time series data.

After running the LSTM model, the SHAP (SHapley Additive exPlanations) method is used to interpret the results. This method helps explain each feature's contribution to the model's prediction for each data point. First, a SHAP explainer is created using a Kernel Explainer object from the SHAP library, which inputs the wrapped model. Then, SHAP values are calculated for the test set using the explainer's shap_values() method, which returns an array of SHAP values for each feature in each data point. Next, the SHAP summary plot is created using the shap.summary_plot() function from the SHAP library. This plot shows the distribution of SHAP values for each feature across all data points and highlights the elements that impact the model's predictions most. To create a more detailed interpretation of a specific data point, a force plot is created using the shap.force_plot() function. This plot shows the contribution of each feature to the model's prediction for a single data point. The y-axis represents the features, while the x-axis represents the SHAP values. The longer the bar, the greater the impact of the feature on the model's prediction.

4.5.4 Temporal Fusion Transformer

The Temporal Fusion Transformer (TFT) is a robust forecasting model that effectively captures temporal and static patterns in time series data. It is designed to handle multiple input features and combines the advantages of recurrent neural networks, such as LSTMs, with the attention mechanism of transformers. This approach allows the model to focus on relevant features and time steps, enhancing its forecasting capabilities. In this project, the model considers the time series data and external variables that may influence sales, such as promotional activities. The model architecture comprises a series of layers, including gated linear units (GLUs), multi-head attention, and temporal self-attention mechanisms, enabling it to capture both long- and short-term dependencies effectively. It is trained using the Adam optimizer and the MAE loss function. The model's hyperparameters include a hidden size of 128 and six LSTM layers, meaning the recurrent aspect of the model consists of six LSTM layers, each with a hidden size of 128. This structure provides a suitable depth for processing time-dependent information. i After training, one of the significant advantages is its inherent interpretability. Unlike other models, such as LSTM, there is no need to utilize additional methods like SHAP (SHapley Additive exPlanations) for interpretation; its architecture allows it to focus on relevant features and time steps, providing a clear understanding of the relationships between input features and their contributions to the predictions, providing insights into feature importances and interactions through its attention mechanisms. This is particularly noteworthy given that the SHAP library is not directly compatible with PyTorch.

4.5.5 Evaluation Metrics

In order to gauge the accuracy of the model's predictions throughout each iteration, it is critical to compute a metric that quantifies the discrepancy between predicted values and actual demand values. Regarding the analysis in Chapter 2.6, we employed the following metrics with the necessary precautions:

- MASE: This metric is well-suited for evaluating our models using the selected dataset. The magnitudes of individual time series do not influence MASE and are applicable even when dealing with intermittent time series, i.e., those with periods showing zero sales. As a result, it serves as the primary metric in this study for analyzing and comparing the performance of various models;

- RMSE: Originating from MSE, this measure assigns more significant penalties to larger errors while maintaining the same units as the target variable. Owing to these properties, both this metric and the MSE measure are frequently utilized for model training and parameter fitting;

- MAE: We calculated this metric for its ease of interpretation and as an initial analysis during the development of the models. However, given its simplicity and reliance on the magnitudes of the time series, it was not used to assess the performance of the models.

4.5.6 Interpretability

The SHAP library in Python is an effective tool used to interpret predictions made by machine learning models, offering greater transparency in understanding the relationships and interactions between features. This library leverages the concept of Shapley values from cooperative game theory, assigning importance to each feature relying on its contribution to the final forecast.

A significant advantage of the SHAP library is its versatility in dealing with different machine-learning models. It is equipped to interpret tree-based models and neural networks and can also operate in a model-agnostic manner. In our specific case, we applied SHAP to interpret XGBoost, a tree-based model, and LSTM, a neural network. For XGBoost, we utilized the part of the SHAP library designed for tree-based models, while for LSTM, we leveraged the part dedicated to neural networks.

One primary advantage of the SHAP library is its ability to deliver global interpretability, i.e., comprehending the model as a whole, and local interpretability, which involves understanding individual predictions, which is required in cases where decisions based on model predictions must be transparent.

However, it is essential to note that not all models were subjected to SHAP analysis. For instance, we excluded TFT from this analysis as it is already inherently interpretable and because SHAP is incompatible with PyTorch, where this model was implemented.

Understanding feature importance and interaction effects in retail scenarios can be especially valuable for strategic decision-making. For instance, SHAP values could show that sales depend highly on seasonal events, indicating that stock levels should be adjusted accordingly. Furthermore, the SHAP library provides visual support such as SHAP summary plots, dependence plots, and decision plots to enhance comprehension of the feature importance and interaction effects, which are practical when presenting these insights to non-technical stakeholders.

Overall, the SHAP library brings interpretability to complex machine learning models, transforming them from black boxes into transparent and trustworthy decision-making tools.

Chapter 5

Tests and Results

In this next chapter, we exhaustively examine the study's findings. Initially, each predicting accuracy model included in the study will be thoroughly evaluated using a selected set of evaluation measures. This is followed by discussing the interpretability results, beginning with a general view and working down to granular details. The chapter conducts a multi-tiered investigation of interpretability, breaking it down into three unique sections: a global perspective, category-specific insights, and detailed interpretability plots for individual items.

Consecutively, particular attention is given to a select group of items exhibiting diverse behaviors, providing a deeper understanding of item-level interpretability. This chapter is integral to the comprehension of the study, bridging the gap between theoretical models and practical applications while offering valuable insights into the inherent intricacies of retail forecasting models.

5.1 Forecasting Accuracy

In predicting daily unit sales for each item in a retail store's inventory, this study evaluated four distinct models: SARIMAX, XGBoost, LSTM, and TFT. These models' forecast performance was compared across three key metrics: MAE, RMSE, and MASE, and are presented in table 5.1.

	SARIMAX			XGBoost			LSTM			TFT		
	MAE	RMSE	MASE	MAE	RMSE	MASE	MAE	RMSE	MASE	MAE	RMSE	MASE
Maximum	67.62	95.81	11.90	31.38	42.13	1.73	74.69	105.21	3.21	37.08	43.86	1.69
Upper Fence	4.03	5.52	1.32	3.94	5.29	1.01	3.64	4.90	1.23	3.44	4.80	1.22
Q3	2.22	3.02	0.93	2.12	2.89	0.83	2.03	2.70	0.87	1.88	2.70	0.84
Mean	2.14	3.02	0.86	1.87	2.65	0.77	1.84	2.49	0.77	1.83	2.60	0.71
Median	1.45	1.92	0.77	1.32	1.89	0.76	1.33	1.77	0.74	1.22	1.84	0.69
Q1	1.01	1.33	0.66	0.88	1.28	0.70	0.94	1.24	0.63	0.83	1.26	0.56
Lower Fence	0.13	0.15	0.28	0	0	0.52	0	0	0.28	0	0	0.19
Minimum	0.13	0.15	0.07	0	0	0.33	0	0	0	0	0	0

Table 5.1: Comparative summary statistics of forecast performance metrics, including MAE, RMSE, and MASE for SARIMAX, XGBoost, LSTM, and TFT. Metrics are shown across various data distribution points: Maximum, Upper Fence, Third Quartile (Q3), Mean, Median, First Quartile (Q1), Lower Fence, and Minimum.

Therefore, the SARIMAX model exhibited the highest maximum error values across all three metrics(67.62, 95.81, and 11.90 for MAE, RMSE, and MASE, respectively), indicating substantial discrepancies between the forecasts and actual sales. However, the model's mean(2.14 for MAE, 3.02 for RMSE, and 0.86 for MASE) and median(1.45 for MAE, 1.92 for RMSE, and 0.77 for MASE) values were reasonable, suggesting that while the SARIMAX model may produce significant errors occasionally, its general performance in predicting unit sales were satisfactory. On the other hand, the XGBoost model demonstrated comparatively lower maximum error values(31.38, 42.13, and 1.73 for MAE, RMSE, and MASE, respectively), indicating fewer instances of significant forecast errors. Particularly, it achieved zero error at the lower fence and minimum points across all three metrics; this happened because there were instances where certain items did not register any sales for the period corresponding to the test set, and given the sliding window approach utilized by the model, it was capable of capturing this lack of sales activity, resulting in zero forecast errors for these specific instances, the same happened to LSTM and TFT models.

However, its mean(1.87 for MAE, 2.65 for RMSE, and 0.77 for MASE) and median(1.32 for MAE, 1.89 for RMSE, and 0.76 for MASE) errors were slightly higher than those of the LSTM and TFT models. This suggests that while XGBoost can limit large errors, its average and central forecast errors may still be significant. Additionally, the LSTM model, despite displaying the highest maximum values for MAE-74.69- and RMSE-42.13-, had the lowest mean(2.49) and median(1.77) RMSE. This indicates that while the LSTM model can occasionally produce large errors, it typically generates lower root mean square errors than the other models. However, its performance was less consistent across the MAE and MASE metrics, where it had higher error values than the XGBoost and TFT models.

Lastly, the Temporal Fusion Transformer (TFT) model consistently outperformed all metrics. While it did not always achieve the lowest maximum error, its mean and median error values across MAE, RMSE, and MASE were the smallest among the four models. Like the XGBoost model, it also showed instances of perfect forecasts. Thus, based on these comparative summary statistics, the TFT model emerges as the most reliable for forecasting daily unit sales in retail inventory management.

Considering the context of this project, MAE is a particularly suitable and robust metric for predicting daily unit sales for each item in a retail store's inventory due to its unique properties and direct interpretability. It quantifies the average unitary discrepancy between the forecasted and actual sales values. As such, it provides precise insight into the expected average deviation in unit sales, which is essential for inventory management. In retail operations, precision in sales forecasting is paramount. Each error in the unit sales prediction may result in either overstocking or understocking. Overstocking due to forecast overestimation can unnecessarily tie up capital and potentially lead to wastage or forced product markdowns, especially for perishable or out-of-season items.

Conversely, forecast underestimation could result in stock shortages, unmet customer demand, lost sales opportunities, and potential reputational damage due to the unavailability of products. In this context, MAE is a balanced error measure, treating both over- and under-forecasting symmet-

rically. Hence, a lower MAE indicates superior forecast accuracy, assuming that both errors have a comparable impact on operational efficiency and profitability.

Apart from MAE, RMSE and MASE are also robust metrics in evaluating the accuracy of forecasting models in a retail context. RMSE measures the average magnitude of the forecast error, assigning higher weights to more significant errors due to the squaring operation. It is helpful in retail forecasting as it penalizes larger errors more severely than smaller ones. In the retail sector, significant forecast errors can be problematic as they could lead to many overstock or stockout situations, causing substantial financial losses or missed sales opportunities.

Meanwhile, MASE is a relative measure of error, comparing the absolute error of the proposed model to a naïve forecasting model; the advantage is its scale-free nature, which makes it suitable for comparing forecast performances across different datasets or units of measurement. In a retail context, this metric provides a standardized evaluation of forecast accuracy across a wide array of items with possibly differing scales of sales volume, ensuring that the model performance is comparable across all items in the inventory.

Upon analysis of the comparative summary statistics in Table 5.1, it becomes clear that the TFT model outperforms SARIMAX, XGBoost, and LSTM in the majority of categories for the Mean Absolute Error (MAE) and Mean Absolute Scaled Error (MASE). Specifically, the TFT model achieves the lowest mean MAE value of 1.83 and the lowest mean MASE value of 0.71. The model also stands out at the median level, reporting the lowest median MAE and MASE values of 1.22 and 0.69, respectively.

However, in the case of the Root Mean Square Error (RMSE), the LSTM model has shown better performance by achieving the lowest mean and median RMSE values, specifically 2.49 and 1.77, respectively. Additionally, the results show that XGBoost and SARIMAX models present some considerations while not outperforming the TFT and LSTM models regarding the Mean, Median, or Lower Quartile metrics. For example, XGBoost has the lowest maximum MAE(31.38), RMSE(42.13), and MASE(1.73) values, demonstrating its ability to avoid high errors in its worst predictions. This can be particularly relevant when the cost of overstocking or understocking is high.

This comprehensive evaluation of the models suggests that, while the TFT model provides the most accurate and reliable forecasts, the LSTM model might be better if the prediction task focuses on minimizing the RMSE. As such, these insights are valuable for retail inventory management, where both accuracy and reliability are essential for effective decision-making and enhanced profitability.

5.2 Interpretability

In this section, we delve into the interpretability of the predictions generated by various models – XGBoost and LSTM – using the SHAP (SHapley Additive exPlanations) library for Python. This library is a robust tool that helps interpret machine learning models' output. It draws upon the game theory principles to illustrate each feature's contribution to the prediction.

Hence, applying the SHAP library for our selected models led to the generation of force plots for a single prediction, multiple predictions, summary plots, and feature importance plots. These graphical representations are valuable aids in the model interpretation. The force plots offer a clear and intuitive perspective into the contribution of features to a single prediction. These plots help clarify features' positive and negative effects on the predicted output by indicating the feature values that increase or decrease the prediction. Similarly, force plots for multiple predictions provide a broader view of feature influences across many instances, allowing us to identify patterns and consistencies in feature impacts. Summary plots provide an overview of the feature importance across the entire dataset. They display the impact of each feature on the model's output for all instances. This holistic view aids in identifying key features that consistently influence predictions and can be particularly useful when deciding on feature engineering or model improvement steps.

On the other hand, feature importance plots are another way of visually interpreting the model. They permit ranking features based on their importance, or the magnitude of their impact, on the model's predictions. This enables a more focused analysis of the most influential features, leading to potentially valuable insights about the data and model's behavior.

SARIMAX, being a parametric model, presents some ease in interpretation as predictions are based on predefined assumptions and parameters. These parameters, which directly influence the predictions, can be estimated from the data. Therefore, understanding the contribution of different components to the predictions becomes relatively straightforward. SARIMAX served as a benchmark model in this analysis, providing a reference point to compare and interpret the more complex XGBoost and LSTM models.

Lastly, as an inherently interpretable model, TFT was excluded from SHAP analysis, primarily because SHAP is incompatible with Pytorch, the framework used to implement TFT. However, TFT's design includes interpretability features like variable selection and time-varying effects, making it relatively easier to understand the predictions without relying on additional tools such as SHAP.

5.2.1 Global

Beginning with the XGBoost model, the features circular discount, discount value, and front page discounts were identified as critical for prediction due to their high F score in the feature importance plot in figure 5.1. The F score, also known as the Feature Importance score, is a metric that indicates how valuable each feature was in the development of the model's decision trees -the more valuable the feature, the higher the score. The prominence of these specific qualities indicates the importance of discounts in boosting unit sales.

In parallel, the SHAP analysis, at figure 5.2 for XGBoost, emphasizes the influence of 'no_promo'. When 'no_promo' is 0, it suggests that active promotions lead to a higher prediction value, indicating increased sales. The other significant features, like a front-page everyday low price, discount, front-page, and circular promotions, are also consistent with the XGBoost model's intrinsic feature importance. Moving on to LSTM, the SHAP analysis in figure 5.3 again shows 'no_promo' as the most impactful feature. Contrarily, the influence of active promotions in this model is negative,

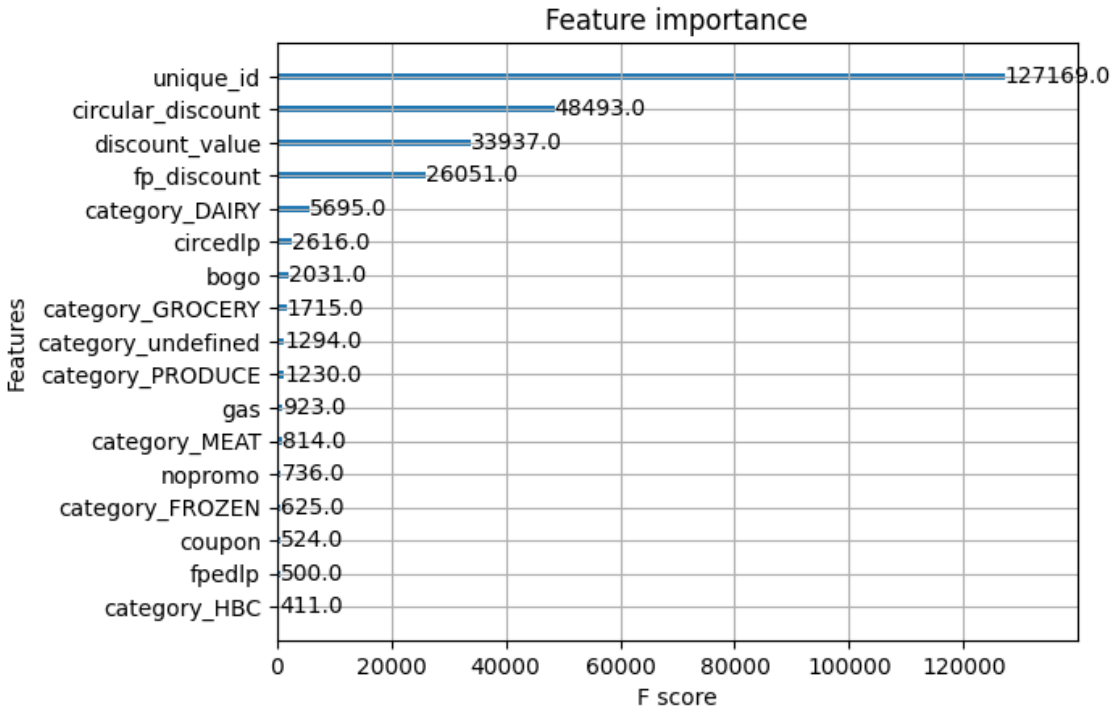


Figure 5.1: Feature importance for XGBoost, considering all time-series

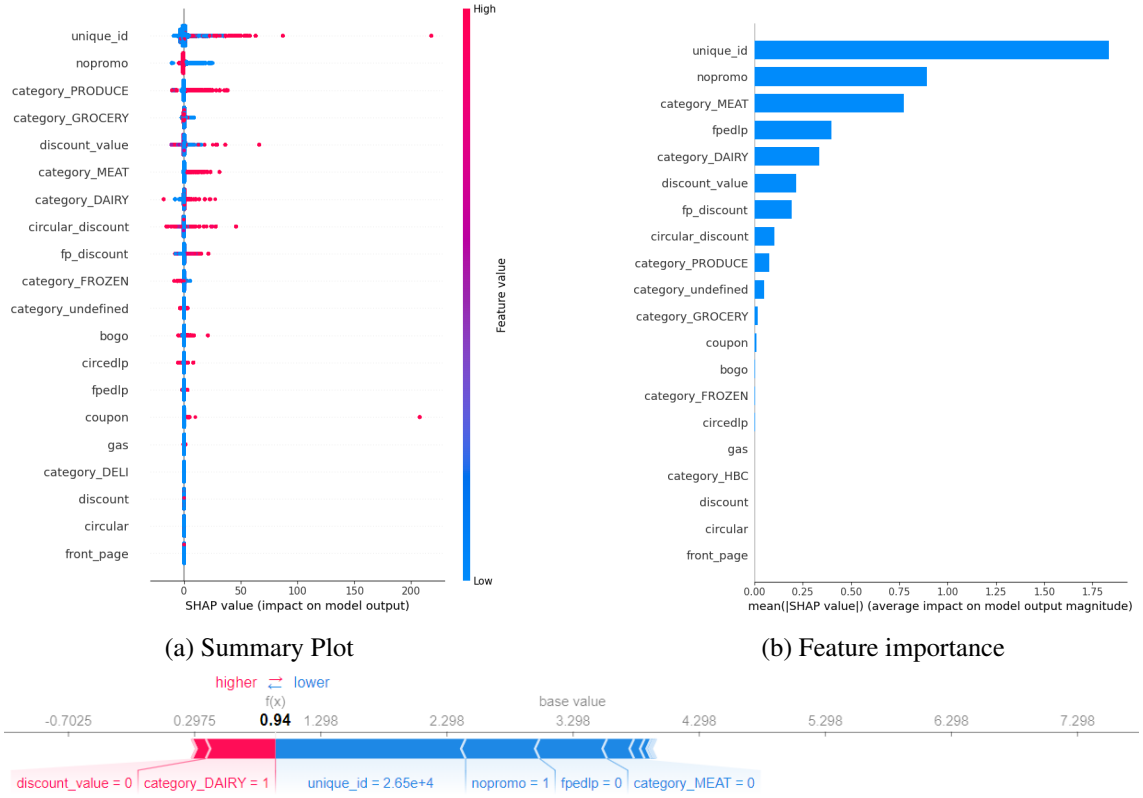
leading to a decrease in predicted sales. This contrasts with the XGBoost model, where promotions were associated with improved sales. LSTM also gives significant importance to lagged unit sales values, suggesting historical data's relevance in forecasting future sales.

Finally, in the Temporal Fusion Transformer model, four distinct interpretability plots are discussed to help understand the internal workings and importance of the variables used. The plots are as follows:

- Attention Diagram (Figure 5.4a): This plot represents the attention distribution across each time step in the input sequence. It helps understand which time steps are given more significance by the model. The higher the attention weight at a particular time step, the more critical it is for the model's prediction.

- Static Variables Importance (Figure 5.4b): This plot exhibits the importance of static (time-invariant) variables. These variables do not change over time in the dataset and can include features like location, category, or other constant attributes. The plot helps identify how significantly these static variables influence the model's predictions.

- Encoder Variables Importance (Figure 5.4c): This plot illustrates the importance of different input variables fed into the encoder component of the Temporal Fusion Transformer. It provides insights into how much these variables contribute to the intermediate representation of data generated by the encoder, which subsequently impacts the model's output forecast.



(a) Summary Plot

(b) Feature importance

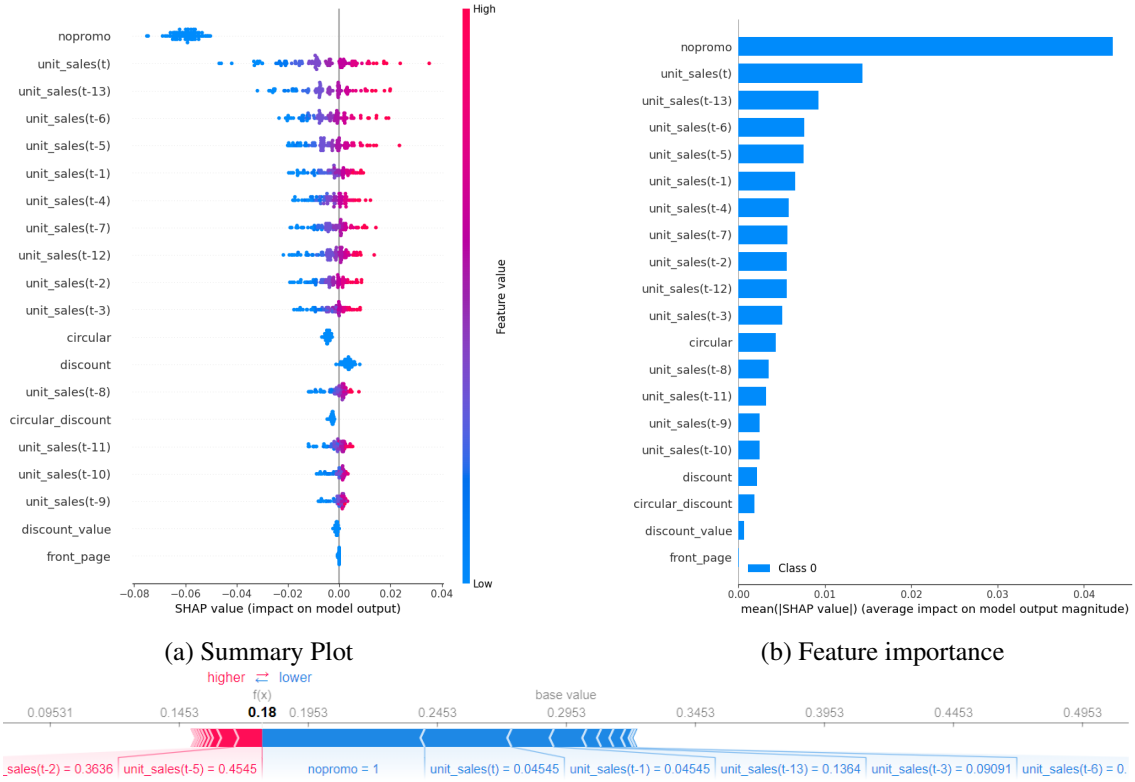
(c) Force Plot

Figure 5.2: SHAP results for XGBoost, considering all time-series

- Decoder Variables Importance (Figure 5.4d): This plot identifies the importance of the variables that the decoder component uses. It shows how much each variable influences the final predictions of the model. These variables can include both the outputs from the encoder and any additional information passed directly to the decoder.

The importance of circular, buy-one-get-one, coupon, and front-page promotions, lagged unitary sales in the encoder, and the predominance of these features in the decoder reaffirms the findings of the other models about their influence.

In comparing the models, we observe that promotions are consistently identified as a critical influencer of unit sales across all models. The impact of the 'no_promo' feature in XGBoost and LSTM and the importance of promotional variables like discounts and coupons in the Temporal Fusion Transformer attests to the deep-seated influence promotional events have on sales. However, a divergence emerges when the direction of the impact is considered. XGBoost associates active promotions with a surge in sales, while LSTM connects them with a drop in sales. This discord could be attributed to how each model interprets and navigates temporal dynamics within the data. LSTM and TFT also highlight past sales data as an integral component in predicting future sales. The emphasis on this feature underscores the pivotal role of historical trends in sales forecasting.



(a) Summary Plot

(b) Feature importance

(c) Force Plot

Figure 5.3: SHAP results for LSTM, considering all time-series

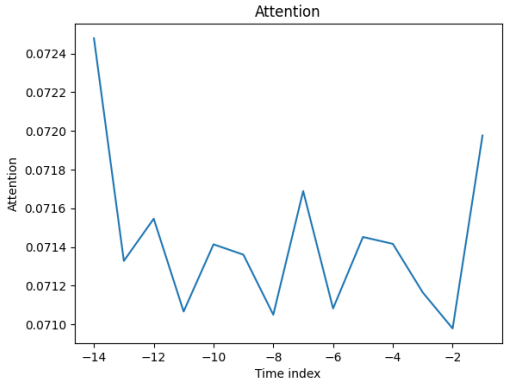
5.2.2 Category Level

5.2.2.1 Dairy

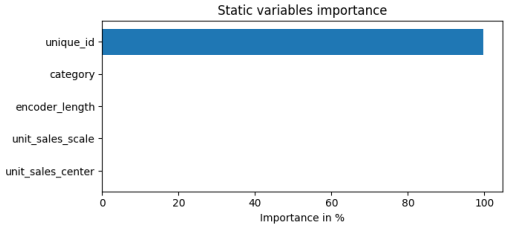
In the context of this research, a more nuanced analysis at the category level offers valuable insights into how each category reacts to promotional activities. This granularity allows for more informed, category-specific sales strategies. It becomes crucial given the different reactions to promotions by categories due to various factors such as customer preference, product seasonality, and product characteristics.

In our focus on the 'Dairy' category, the XGBoost model results manifest that the most significant promotional events are circular, discount, and front-page everyday low-price promotions, according to the model's feature importance plot as illustrated in Figures 5.5 and 5.6b. These findings echo the global analysis regarding the importance assigned to the 'no_promo' feature, per the SHAP feature importance analysis.

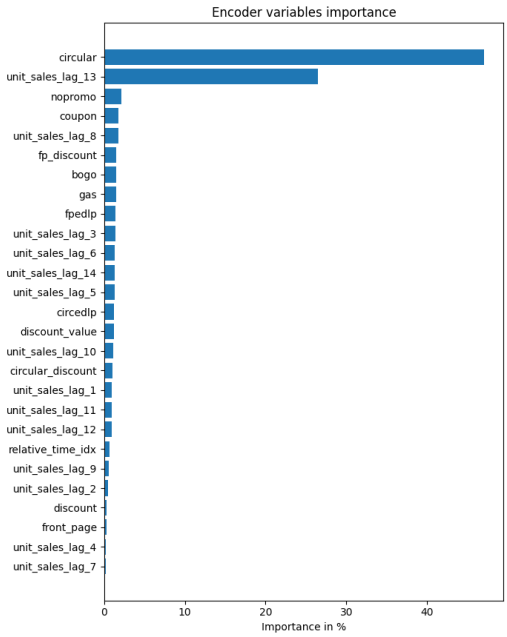
Furthermore, the front page, circular, and discount promotions exhibit significant influence. An active promotion, denoted by 'no_promo' equals 0, positively impacts the model output. This positive correlation between higher values for circular, front-page, and discount promotions and the model output suggests that these promotions effectively augment the Dairy category's sales, illustrated in figure 5.6a.



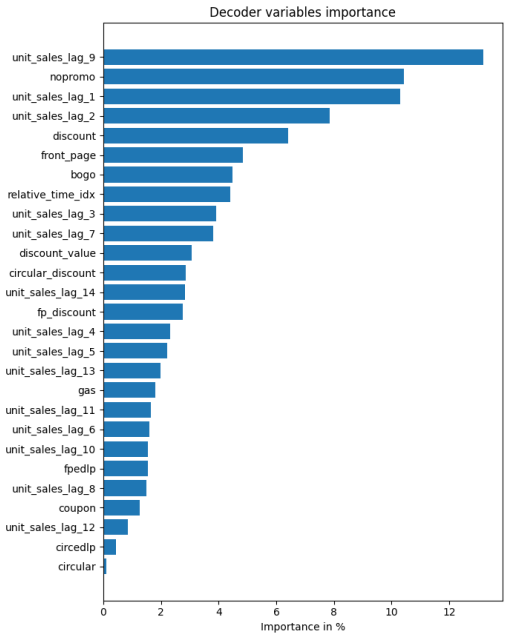
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.4: Interpretability results for TFT, considering all time-series

On the other hand, the LSTM model appears less responsive to promotional activities on sales, gravitating more towards the lagged unit sales, especially those from the day before each prediction, presented in Figure 5.7b. This observation could suggest that historical sales data might serve as a more potent predictor of future sales than promotional activities within the Dairy category. Moreover, it's observed that lower lagged unitary sales negatively influence the model's output, as illustrated in Figure 5.7a. This also could indicate potential limitations of the LSTM model in identifying the intricate influences of promotional activities within this category.

On the contrary, the TFT model resurfaces the prominence of promotional influences. As per the encoder variables' importance, the lagged unitary sales, discounts, and buy-one-get-one promotions are key features- Fig. 5.8c. The decoder variables' importance underscores 'no_promo' as the most significant feature, followed by lagged unit sales values, Fig. 5.8d. In addition, the

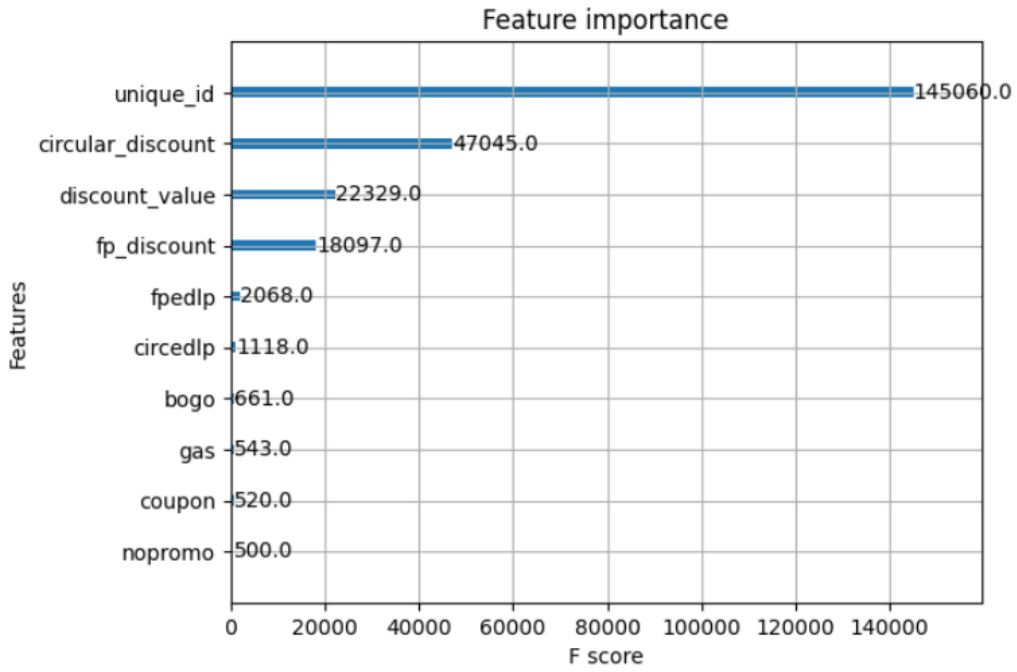
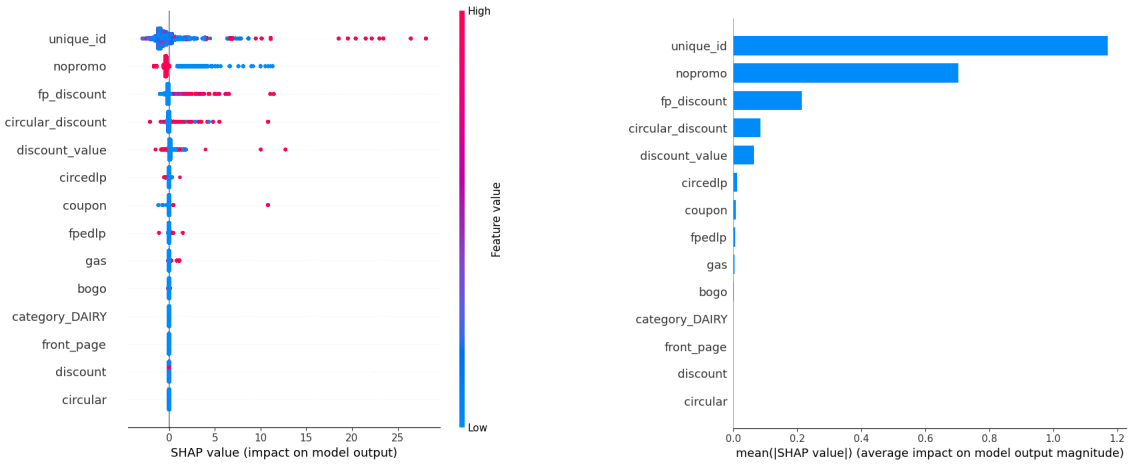


Figure 5.5: Feature importance for XGBoost, considering dairy category



(a) Summary Plot

(b) Feature importance

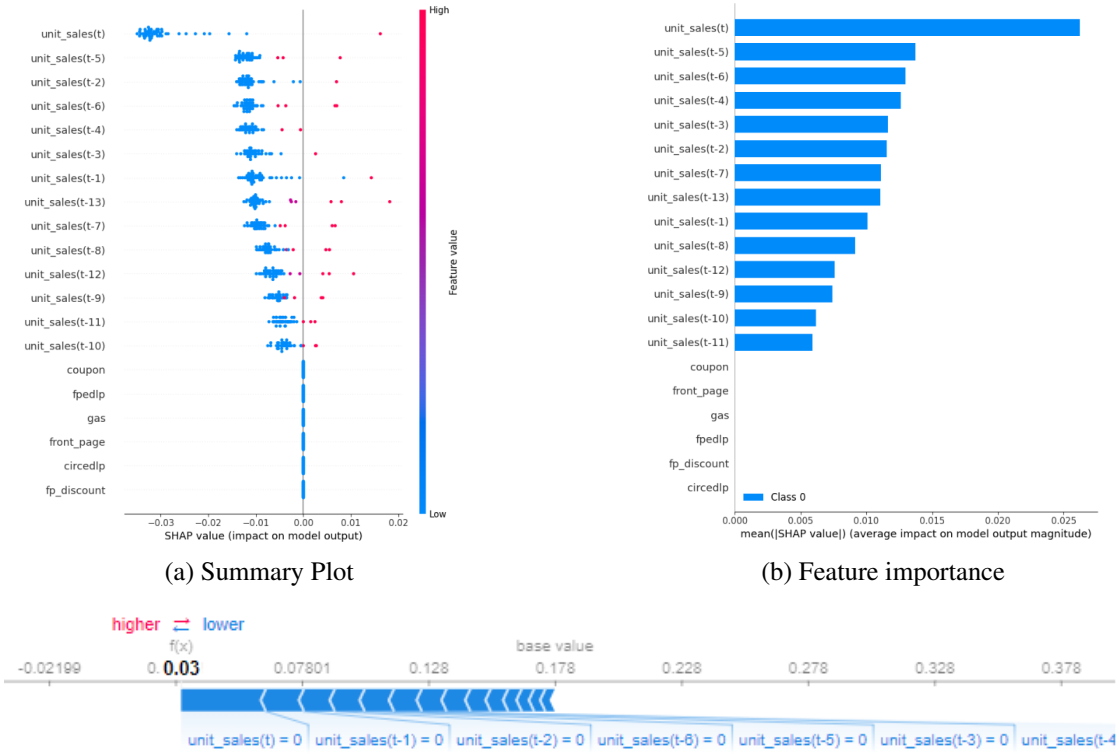


(c) Force Plot

Figure 5.6: SHAP results for XGBoost, considering dairy category

model points to the extent of circular everyday low-price, front-page, circular, buy-one-get-one, and front-page everyday low-price promotions.

These findings provide compelling evidence that, despite the variance in how different models perceive promotional activities, promotions play a substantial role in the Dairy category. The XGBoost and TFT models particularly underscore the importance of circular, discount, front-page everyday low-price, and buy-one-get-one promotions, positing these as key sales levers in this category. The insights gleaned can inform the development of category-specific promotional strategies, offering a potential pathway toward improved sales performance and enhanced profitability in the Dairy category.

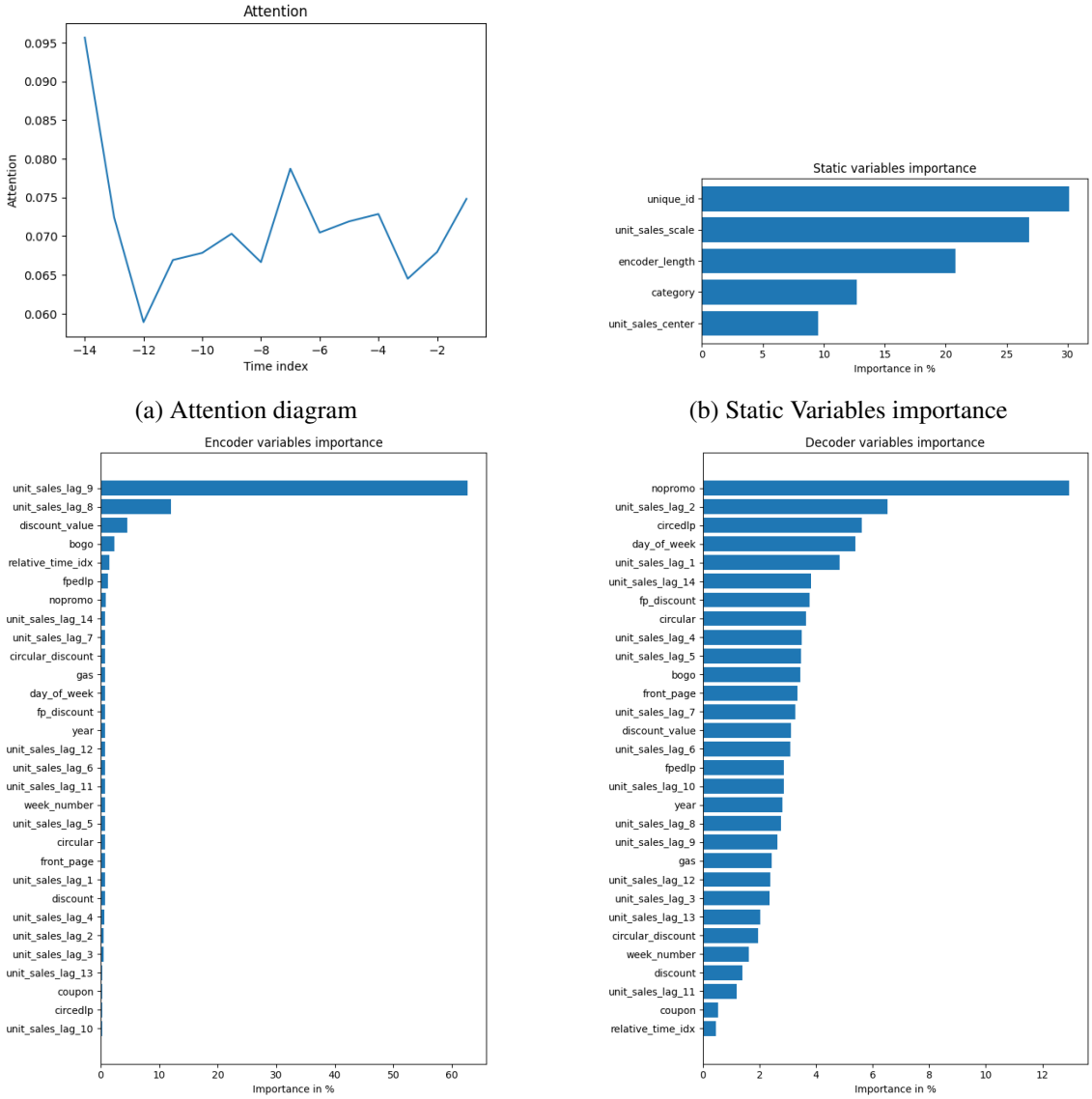


(a) Summary Plot

(b) Feature importance

(c) Force Plot

Figure 5.7: SHAP results for LSTM, considering dairy category



(a) Attention diagram

(b) Static Variables importance

(c) Encoder Variables importance

(d) Decoder Variables importance

Figure 5.8: Interpretability results for TFT, considering dairy category

5.2.2.2 Deli

Upon examining the Deli category, the study reveals intriguing patterns and divergences compared to the Dairy category. The feature importance determined by the XGBoost model in figure 5.9 highlights circular, discount, and circular everyday low-price promotions as the primary influencers on the model's output, followed by front-page, coupon, and buy-one-get-one promotions.

According to that, the SHAP feature importance analysis, in figure 5.10b, also validates these results, underscoring the significant roles of circular, buy-one-get-one, and circular everyday low-price discounts. Nonetheless, a significant deviation is observed concerning the 'no_promo' feature: the XGBoost model ranks it among the least influential features, contrasting the SHAP analysis, which presents it as a substantial determinant of the model's predictions. Further elaboration

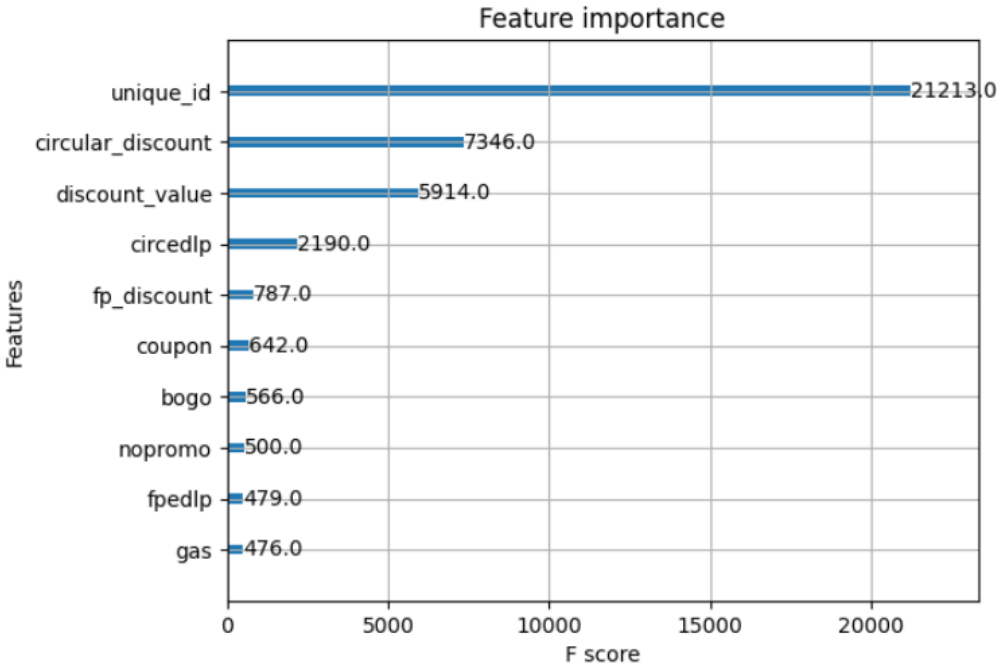
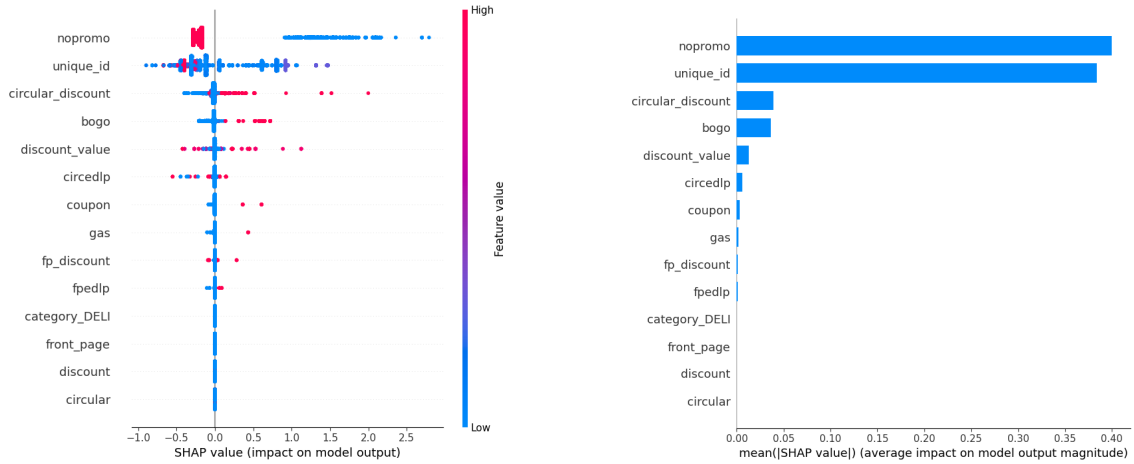


Figure 5.9: Feature importance for XGBoost, considering deli category



(a) Summary Plot

(b) Feature importance



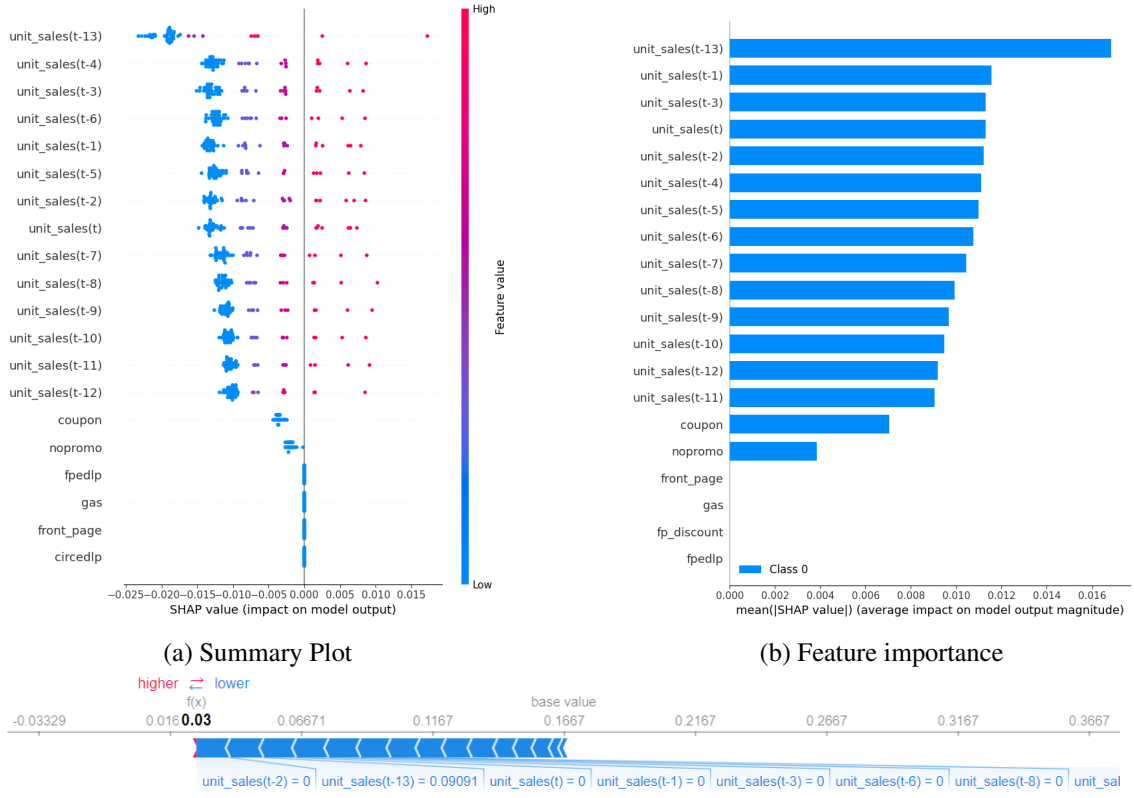
(c) Force Plot

Figure 5.10: SHAP results for XGBoost, considering deli category

on the analysis with the summary plot indicates a positive correlation when 'no_promo' equals 0 and the model output, echoing findings from the 'Dairy' category and global analysis, illustrated in figure 5.10a. This insinuates that active promotions positively influence the model's sales predictions within the 'Deli' category. Similarly, a positive correlation is noted for higher circular,

buy-one-get-one, and discount promotions values. Examining the force plot, in figure 5.10c, for an arbitrary prediction within this category highlights a negative influence on the model's output when 'no_promo' equals one or when the circular promotion is inactive.

On the contrary, the LSTM model offers a differing perspective, emphasizing the significant influence of lagged unitary sales, especially those originating 12 days prior, as shown in figure 5.11b. This pattern could suggest a more robust, enduring response to promotions within the 'Deli' category than 'Dairy.' Alongside lagged sales, the LSTM model also records the influence of coupon promotions and 'no_promo.' In line with the summary plot, in figure 5.11a, smaller values for days before, coupon, and 'no_promo' adversely impact the model output, signaling the importance of recent sales and active promotions in propelling 'Deli' sales.



(a) Summary Plot

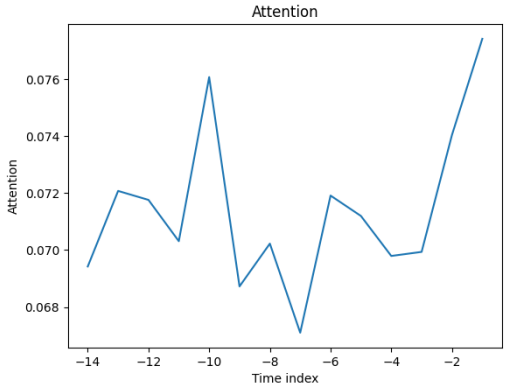
(b) Feature importance

(c) Force Plot

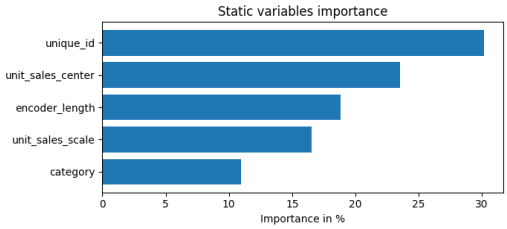
Figure 5.11: SHAP results for LSTM, considering deli category

The interpretability results from the TFT model echo several previous findings while introducing new insights. In descending order of importance, the encoder variables' significance, illustrated in figure 5.12c lists circular everyday low-price, lagged unitary sales, front-page everyday low-price, circular, buy-one-get-one, front-page, and 'no_promo'. Interestingly, this analysis denotes a lesser impact of discount promotions within the Deli category than in the Dairy category. The decoder variables', in figure 5.12d significance exposes front-page everyday low price as the most influential feature, succeeded by 'no_promo', buy-one-get-one, circular promotions, and

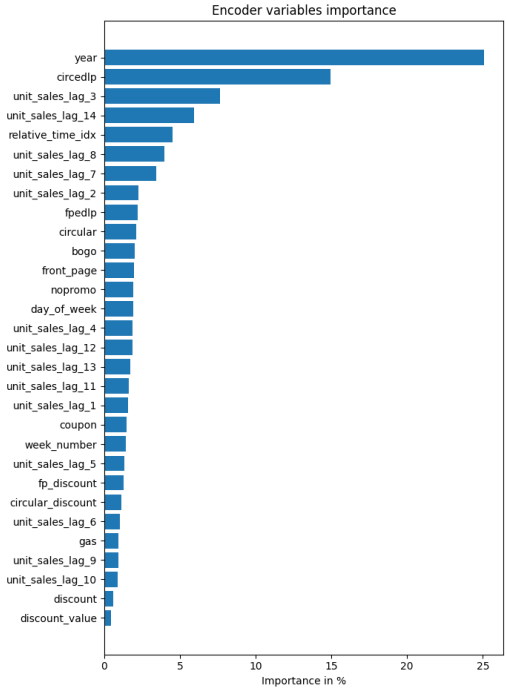
lagged unit sales values. Front-page and discount promotions hold the least importance within this context.



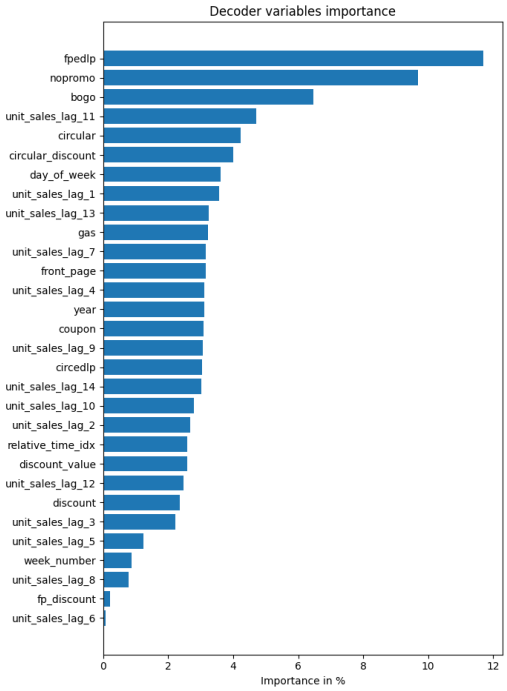
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.12: Interpretability results for TFT, considering deli category

To conclude, the results underline that promotions play a meaningful role in influencing sales within the 'Deli' category, albeit distinctively from the 'Dairy' category. Active promotions and specific types such as circular, buy-one-get-one, and front-page everyday low prices emerge as critical determinants of sales. Moreover, the significance of lagged sales data, particularly those from 12 days prior, suggests unique temporal dynamics within the 'Deli' category, which may be instrumental in enhancing sales forecasting accuracy.

5.2.2.3 Frozen

Breakdown presents intriguing findings for the frozen category, demonstrating similarities and divergences from previous category analyses. According to the feature importance plot generated by the XGBoost model, in figure 5.13, circular, discount, and front-page discount promotions wield the most significant influence on the model's output. The feature importance analysis corroborates this finding carried out using the SHAP library. However, a crucial difference is evident in the case of the 'no_promo' feature, which the SHAP plot identifies as the most significant determinant of the model's predictions, illustrated in figure 5.14b. The summary plot presented in Figure 5.14a illustrates that the activation of a promotion corresponds with an increase in sales, as indicated by the positive SHAP value, thus leading to an upward adjustment in sales forecasts within this category.

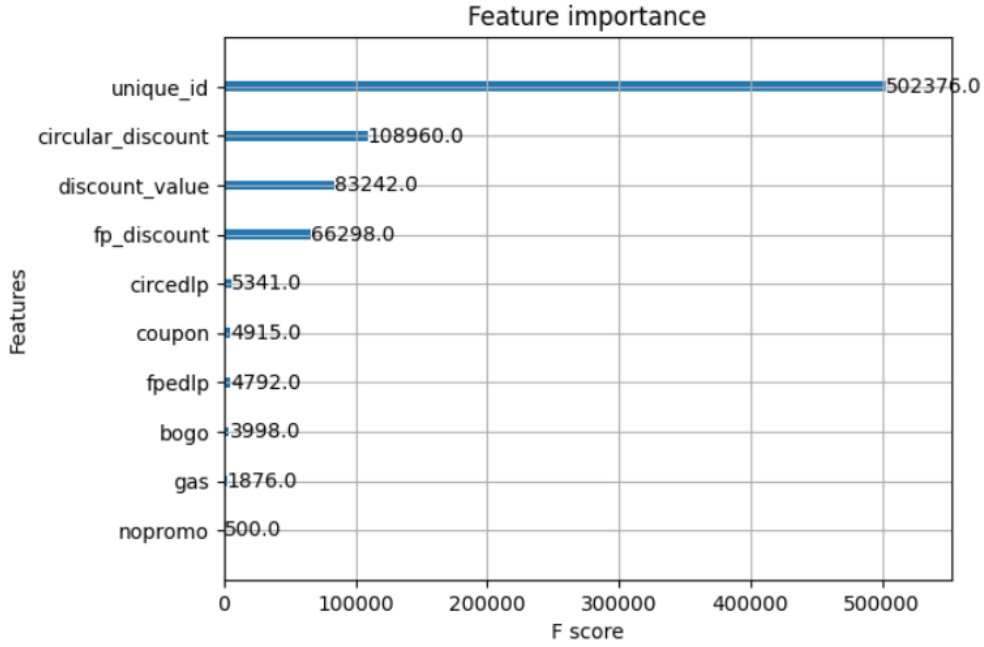
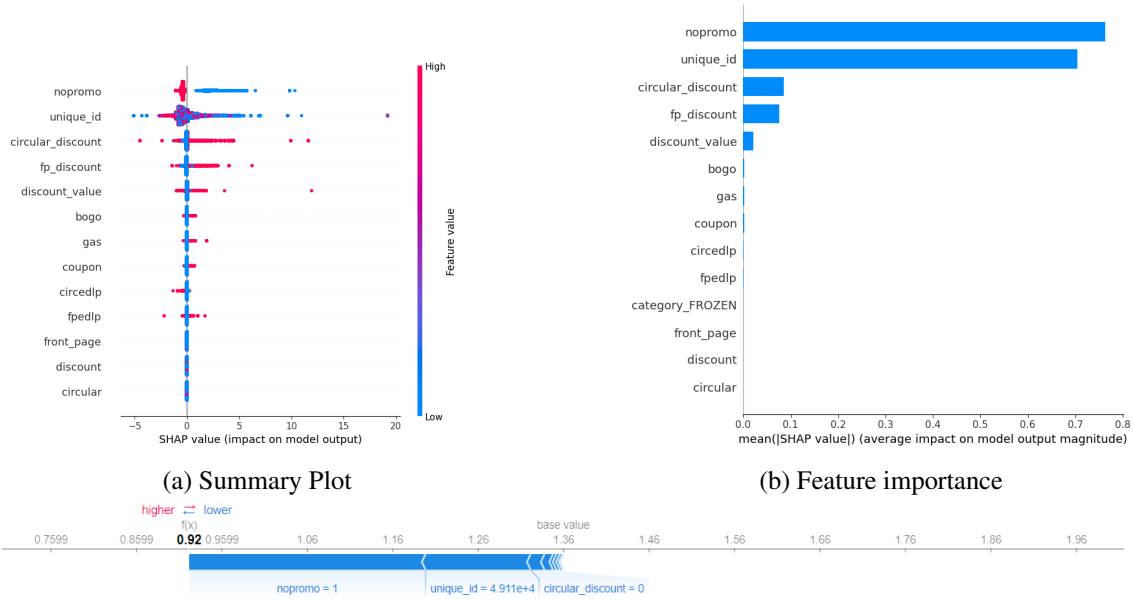


Figure 5.13: Feature importance for XGBoost, considering frozen category

Contrarily, the data does not consistently reflect an increased demand in sales when specific promotions (circular discount, front page, discounts, buy-one-get-one, and gas promotions) are activated. This observation suggests that these promotions do not invariably positively impact the model output, hinting at their inconsistent effectiveness. Nonetheless, it is noteworthy that for most observations, circular, front-page, and discount promotions are perceived by the model to impact sales positively.

The interpretability results of the LSTM model for the 'Frozen' category largely align with those of the XGBoost model. The feature importance plot indicates, in figure 5.15b, 'no_promo' as the most pertinent feature, succeeded by circular and discount promotions. Following these, only lagged unitary sales impact the model output.



(a) Summary Plot

(b) Feature importance

(c) Force Plot

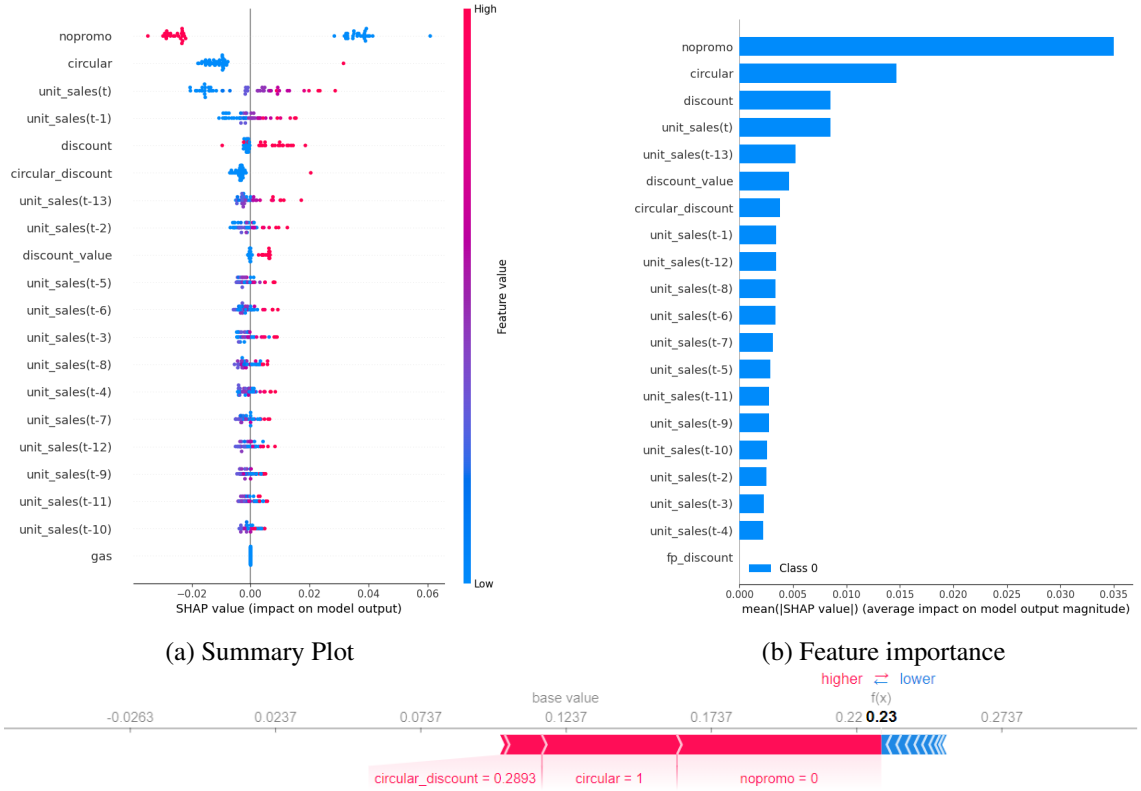
Figure 5.14: SHAP results for XGBoost, considering frozen category

Furthermore, the summary plot (figure 5.15a) reveals that when 'no_promo' is 1, the corresponding SHAP value is negative, indicating a decrease in the model's sales predictions. Conversely, when 'no_promo' is 0, and circular and discount promotions assume higher values, the impact on the model output is positive, thereby forecasting increased sales. This pattern aligns with the observations from other categories, reaffirming the importance of recent sales and active promotions.

Contrastingly, for the encoder variables' importance, the TFT model primarily highlights the impact of unitary sales from seven days prior, as seen in figure 5.16c. As for the decoder variable importance, in figure 5.16d, in descending order, 'no_promo' emerges as the most influential feature. This is followed by lagged values of unitary sales from the day before and two days prior, with circular everyday low-price, coupon, front-page, gas, and discount promotions also being noteworthy. Finally, discount and front-page everyday low-price promotions round up this list. In conclusion, these findings indicate that while there are inconsistencies in the impact of specific promotions, 'no_promo' and circular and discount promotions have consistently emerged as significant determinants of sales in the 'Frozen' category across the models. Furthermore, the crucial role of lagged sales in the TFT model, particularly from seven days before, illuminates unique temporal dynamics that should be considered for effective sales forecasting within this category.

5.2.2.4 General Merchandise

The analysis of the General Merchandise category yields significant insights, highlighting key patterns and relationships. As illustrated by the feature importance chart generated by the XGBoost



(a) Summary Plot

(b) Feature importance

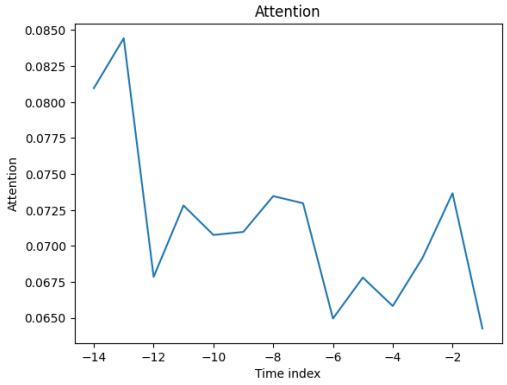
(c) Force Plot

Figure 5.15: SHAP results for LSTM, considering frozen category

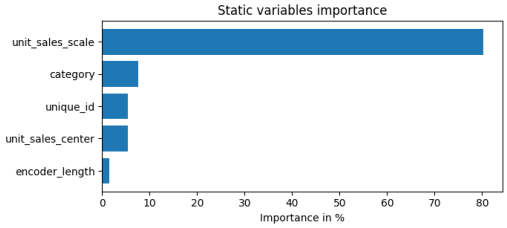
model, in figure 5.17, among the various promotional activities, discounts and circular promotions exhibit the most pronounced impact. This is followed by the 'no_promo' feature. The relevance of these findings is reinforced by the SHAP-generated plots, which underscore the vital role played by discount and circular promotions. However, the SHAP feature importance and summary plot, as demonstrated in Figure 5.18, bring the 'no_promo' feature to the forefront as the most influential factor, succeeded by the promotions previously mentioned.

Moreover, the summary plot elucidates that when 'no_promo' is set to 0, thereby indicating the implementation of active promotion, the SHAP value inclines toward the positive. This trend prompts the model to anticipate an augmentation in sales, a pattern echoing previous category analyses and the all-encompassing global interpretability. Further, high values for discount and circular promotions correspond with positive SHAP values, suggesting their beneficial influence on the category's sales. A force plot created for a random prediction within this category, in figure 5.18c, conveys that the absence of promotional activities leads the model to anticipate a markedly reduced sales value of 0.44 units (a scaled value), contrasting with the expected 1.434 units projected for the time series.

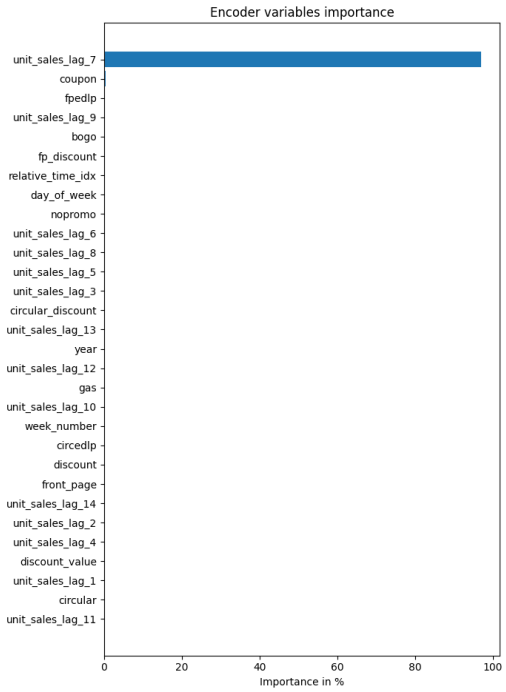
The interpretability results from the LSTM model largely correspond with those derived from the XGBoost model. The feature importance plot, in figure 5.19b, created using SHAP underlines the importance of 'no_promo', discount, and circular promotions. These are followed by lagged



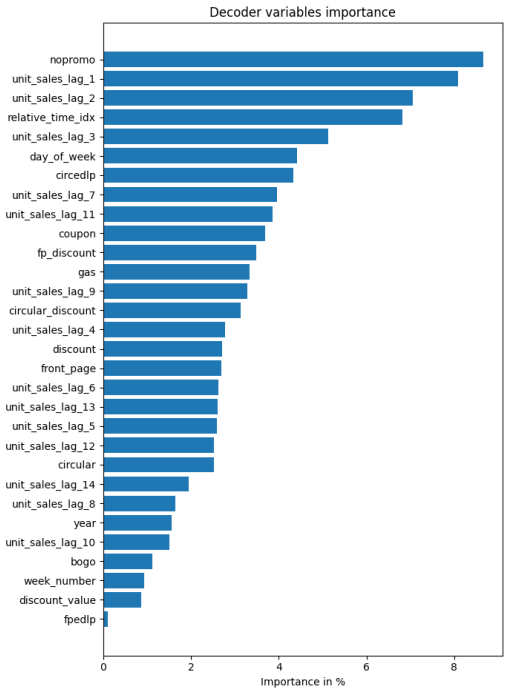
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.16: Interpretability results for TFT, considering frozen category

unit sales values, with the value immediately preceding the forecast exerting the most substantial impact on the model's output.

Additionally, the summary plot, in figure 5.19a, substantiates this trend, showing that when 'no_promo' is 0, it leads to a positive SHAP value, implying an expected increase in unit sales. Moreover, higher values for a discount or circular promotions also yield positive SHAP values, forecasting enhanced unit sales. A force plot for a random prediction within this category illustrates that a discount of 27% has a favorable impact on the model's output. Accounting for all other features depicted in the figure, the model forecasts 0.13 unit sales (a scaled value) compared to the mean value of 0.071, this is illustrated in figure 5.19c

Finally, the TFT model offers unique perspectives. The encoder variables' importance plot,

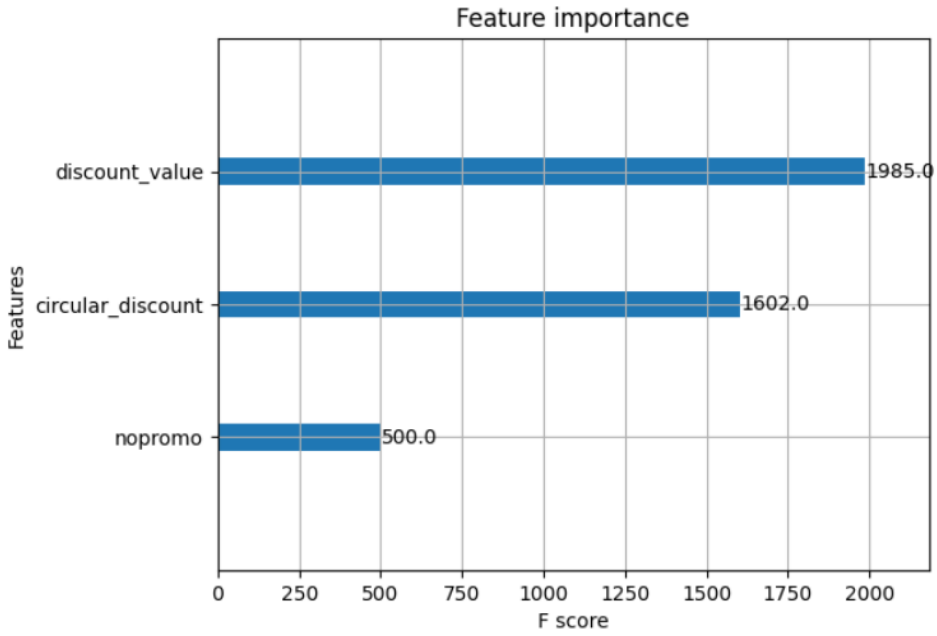
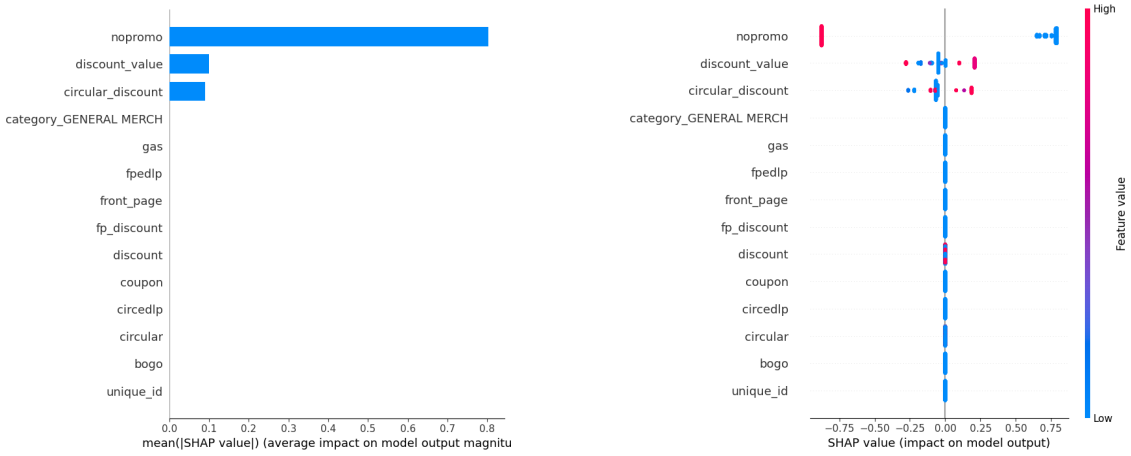
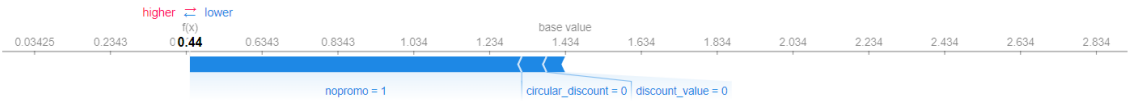


Figure 5.17: Feature importance for XGBoost, considering general merchandise category



(a) Summary Plot

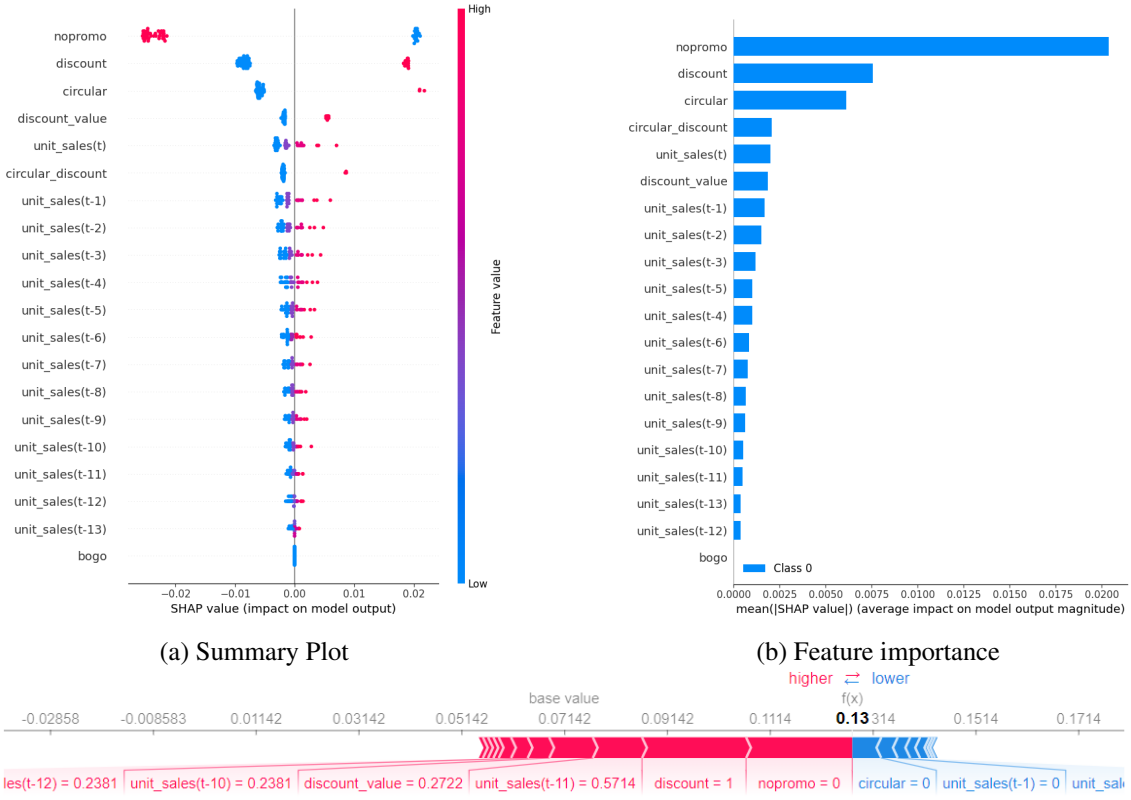
(b) Feature importance



(c) Force Plot

Figure 5.18: SHAP results for XGBoost, considering general merchandise category

in figure 5.20c, denotes a significant influence of lagged unit sales from the past three days and the day-of-the-week feature. However, circular and discount promotions continue to command the most impact among all promotional activities. Interestingly, coupon promotions exert a lesser effect than other promotions. Conversely, the decoder variables' importance plot, in figure 5.20d,



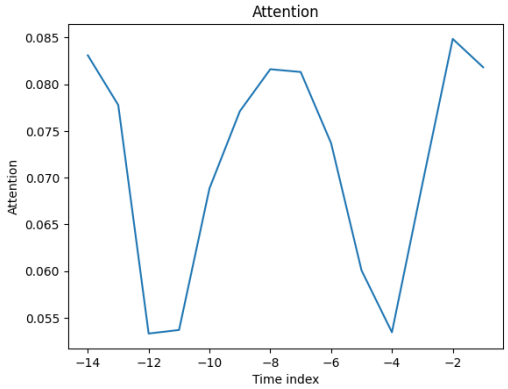
(a) Summary Plot

(b) Feature importance

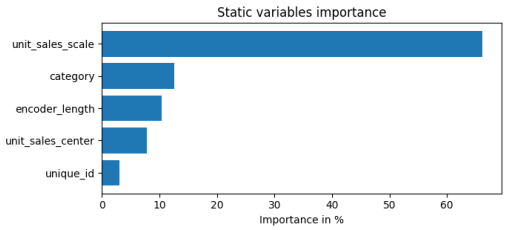
(c) Force Plot

Figure 5.19: SHAP results for LSTM, considering general merchandise category

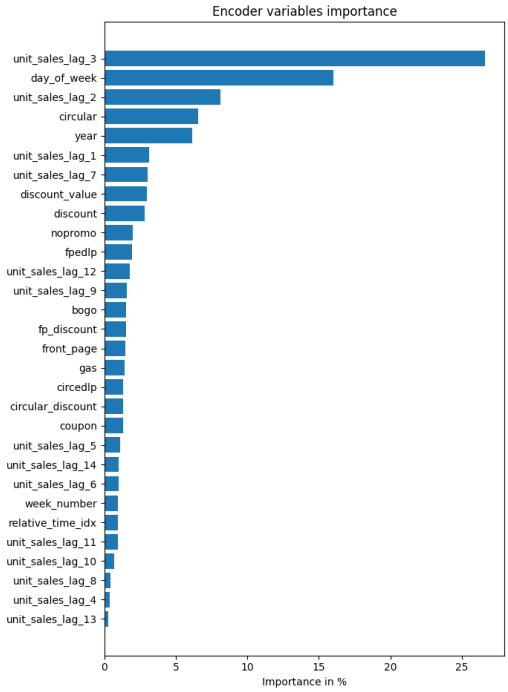
demonstrates that 'no_promo' significantly impacts the model output. This is followed by lagged unit sales for 12, 10, 8, and 2 days before the prediction, succeeded by front-page everyday low price and circular promotions. In conclusion, the models capture similar behaviours in response to promotional activities. Discounts and circular promotions consistently surface as notable influencers on sales. This constancy across models underscores these promotions' critical role in determining sales within the General Merchandise category. The insights gleaned from this analysis provide a promising pathway for enhancing sales forecasting precision. They emphasize incorporating promotional activities, mainly circular and discount promotions, when formulating predictive models.



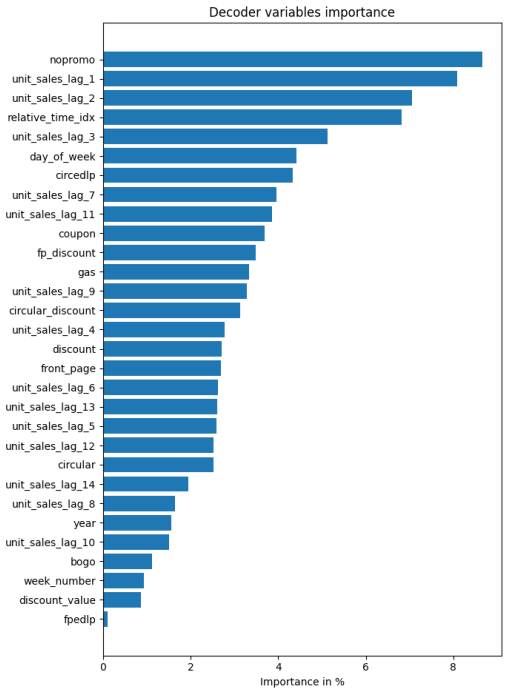
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.20: Interpretability results for TFT, considering general merchandise category

5.2.2.5 Grocery

Now considering the analysis of the Grocery category, the XGBoost model's feature importance chart, in figure 5.21, the promotional activities exerting the most influence on the model's output, in descending order, are circular promotions, discounts, front-page promotions, and circular every-day low-price promotions, as indicated by the F score. In contrast, front-page everyday low-price, gas, and coupon promotions have the least significant impact.

Mirroring these results, the SHAP feature importance plot, in figure 5.22b, identifies the 'no_promo' feature as the most influential on the model's output. This is followed by discounts, circulars, front page everyday low prices, and buy-one-get-one as the most impactful promotional events. The summary plot, in figure 5.22a, further reinforces these findings, revealing a positive

SHAP value when 'no_promo' is 0, leading to a subsequent positive impact on the model's output and an anticipated increase in sales.

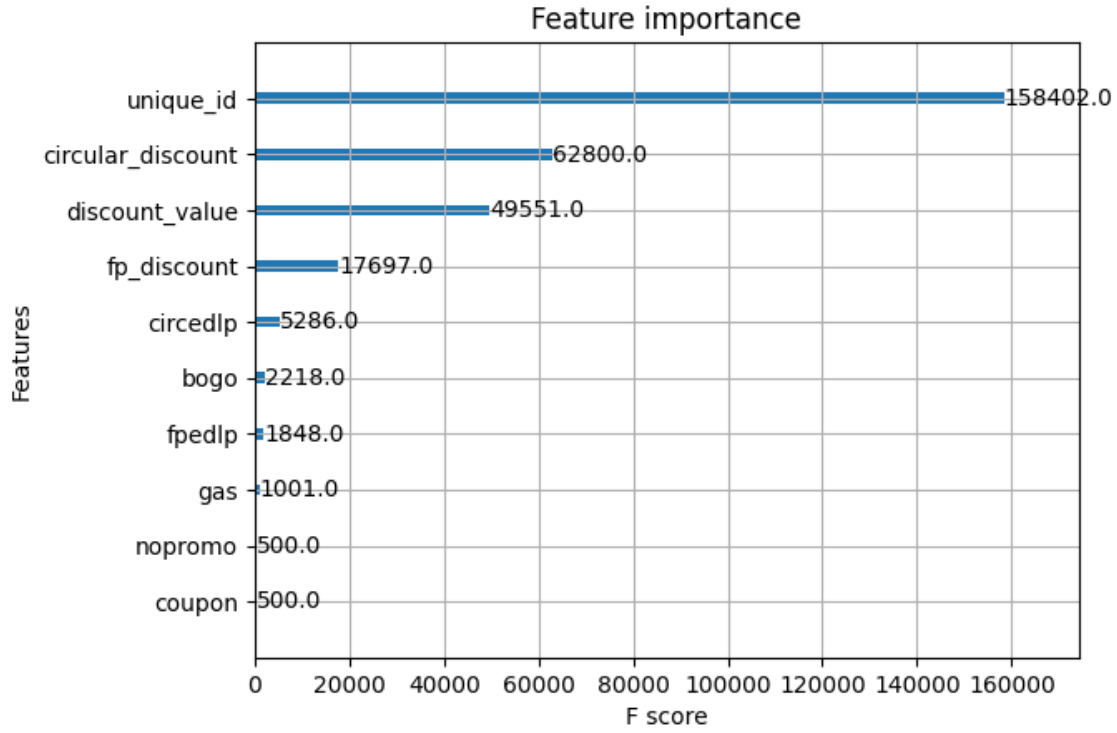
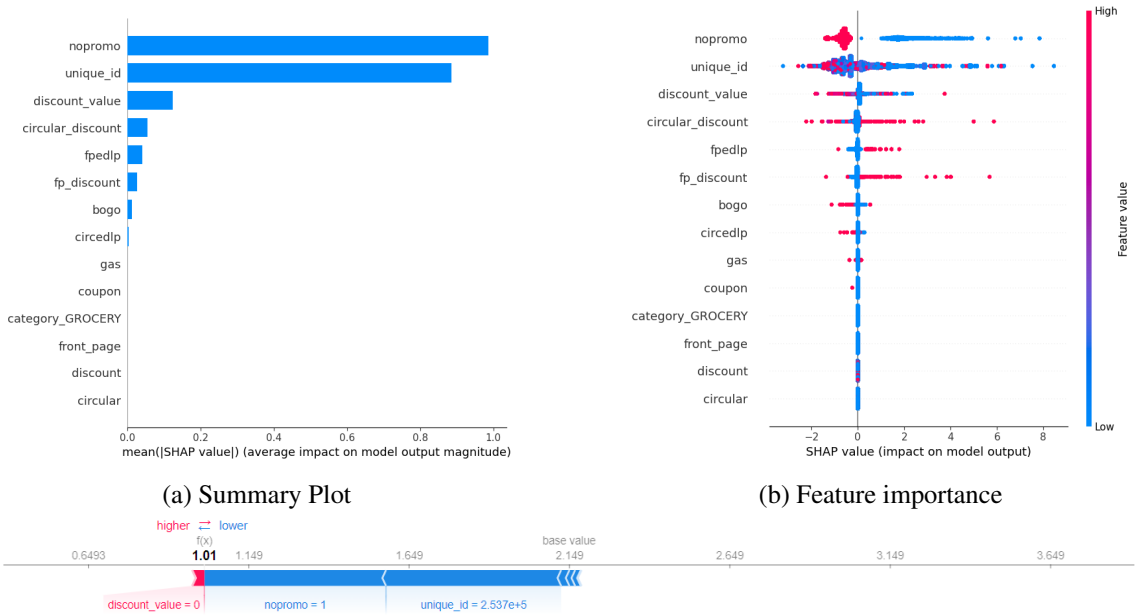


Figure 5.21: Feature importance for XGBoost, considering grocery category

However, discount and circular promotions exhibit inconsistent behaviour as higher values of these features can positively or negatively impact different predictions. Consequently, it is challenging to ascertain the utility of these promotional events in enhancing the retailer's sales for this specific category. Conversely, front-page and front-page everyday low-price promotions positively impact the model's output when their values are high or non-zero. Notably, buy-one-get-one and circular everyday low-price promotions negatively influence the model output when active, suggesting that these promotional events may not be an optimal strategy for this category. As depicted in a forced plot, in figure 5.22c, for a random prediction, when no promotion is active, the predicted value for a random item within this category on a given day has a negative impact, a pattern consistent with previous category analyses.

Similarly, the LSTM model's interpretability results align with the feature importance findings. The 'no_promo' feature is again identified as having the most significant impact on the model output. It also highlights the effects of discount and circular promotions on the model's output. However, this model indicates that lagged unitary sales values significantly influence this category's model output. As illustrated in Figure 5.23b, the lagged values from the day preceding the prediction and 14 days prior emerge as the second and third most impactful features. The summary plot analysis, in figure 5.23a, confirms the expected behavior observed in other categories: higher sales from past values are associated with increased forecasted sales. Interestingly,



(a) Summary Plot

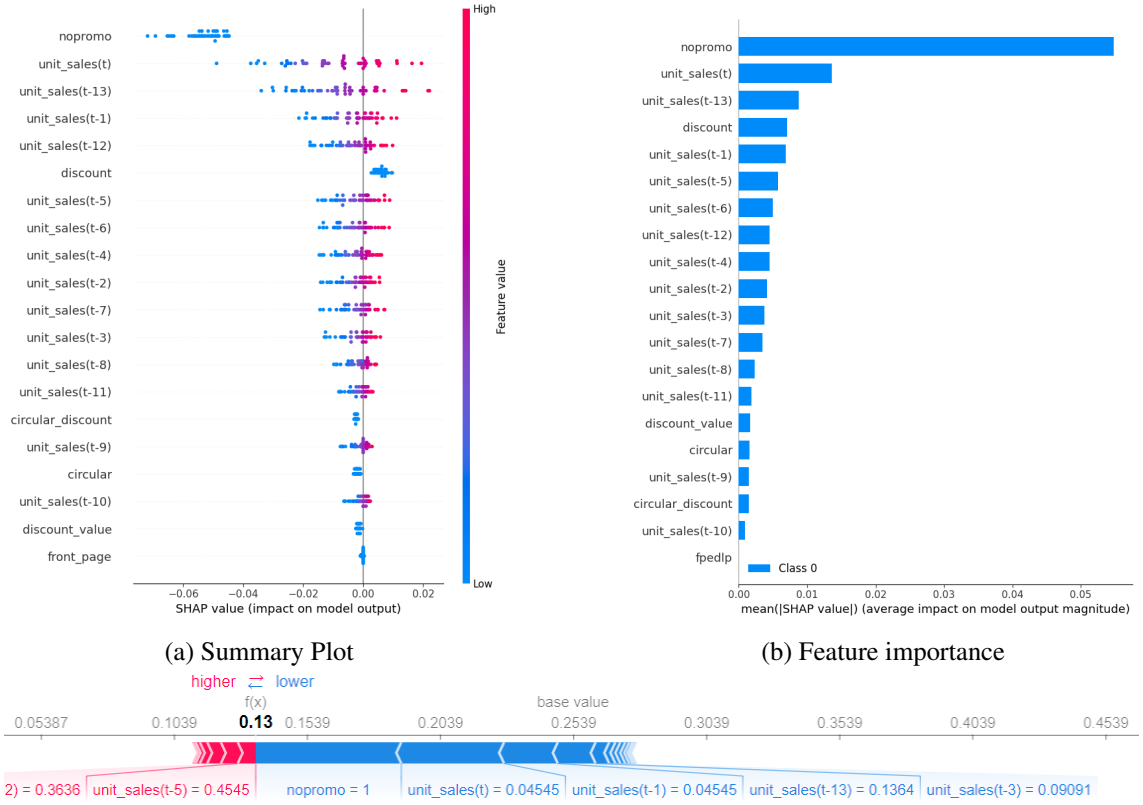
(b) Feature importance

(c) Force Plot

Figure 5.22: SHAP results for XGBoost, considering grocery category

the LSTM model suggests that higher forecasted unit sales occur when the discount is inactive, implying that this promotional activity may not significantly impact sales within this category. In contrast, when circular promotions are inactive, the SHAP values lean negative, indicating a potential negative impact on the model's output.

Lastly, the TFT model's encoder variables' importance plot, illustrated in figure 5.24c, reveals that only the lagged values corresponding to 5 and 12 days before the prediction have a significant impact. Similarly, the decoder variables' importance plot, in figure 5.24d, confirms 'no_promo' as the most impactful feature. Front-page everyday low-price, gas, and circular promotions exert the most substantial impact among promotional activities.



(a) Summary Plot

(b) Feature importance

(c) Force Plot

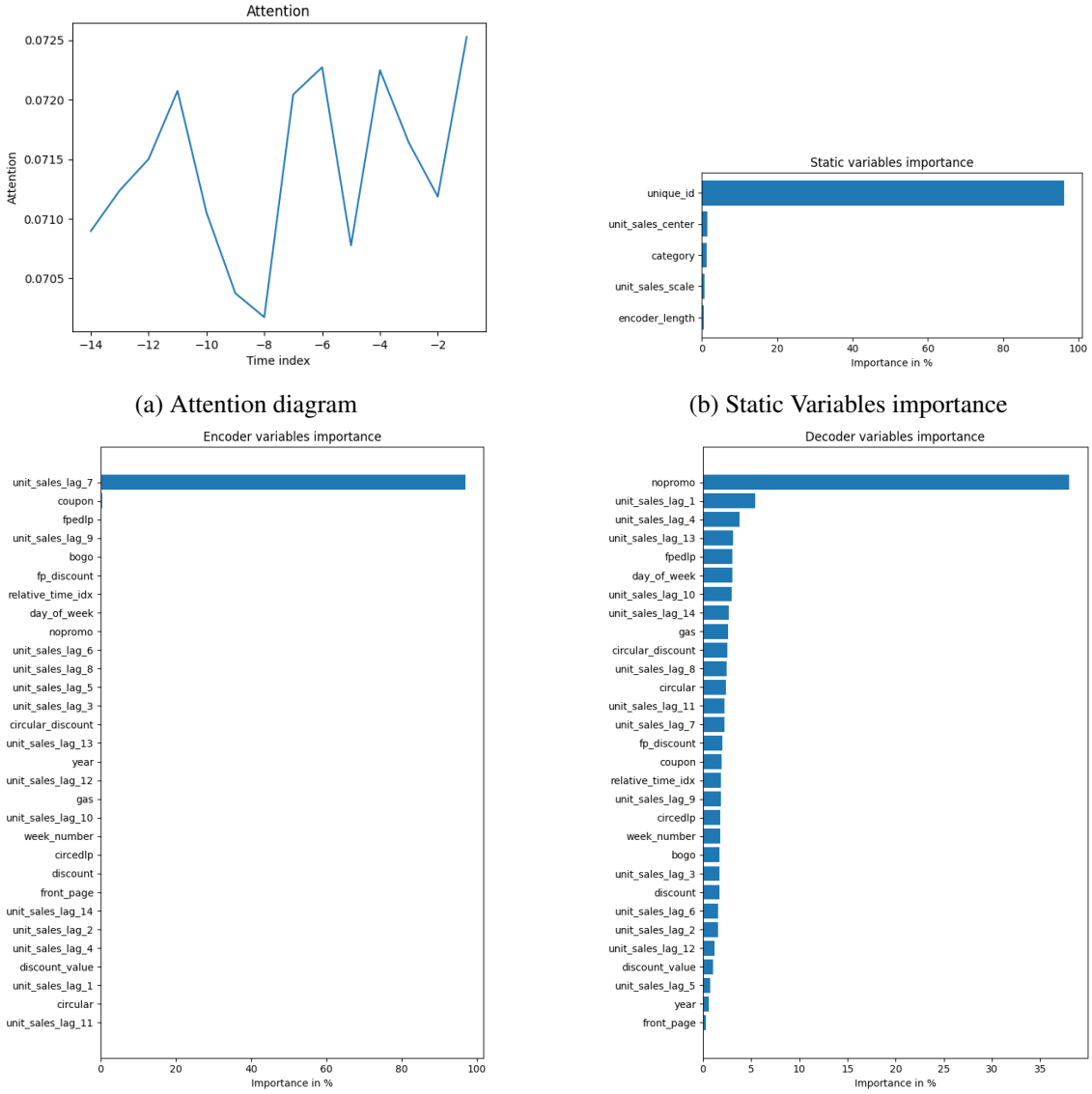
Figure 5.23: SHAP results for LSTM, considering grocery category

5.2.2.6 HBC

Based on the feature importance generated by the XGBoost model for the Health and Beauty Care category, illustrated in figure 5.25, discount and circular promotions emerge as the most impactful on the model's output. These are followed by circular everyday low-price, buy-one-get-one, and front-page everyday low-price promotions. A similar pattern is observed in the SHAP feature importance plot, in figure 5.26b, identifying discount and circular promotions as the most impactful promotional events. They succeeded with buy-one-get-one, front-page everyday low-price, and circular everyday low-price promotions. Consistent with the trends observed in other categories, 'no_promo' stands out as the most significant feature for the model, as per SHAP feature importance.

The summary plot in Figure 5.26a mirrors the patterns noted in other categories. It suggests that when a promotion is active, corresponding to a 'no_promo' value of 0, the model's output increases, as indicated by the positive SHAP values.

Additionally, the plot illustrates that discount promotions negatively impact sales - higher discount values (indicating active promotion) correspond to negative SHAP values, leading the model to predict lower unitary sales for items in this category. Circular promotions show an irregular pattern, with active promotions potentially leading to higher or lower unitary sales predictions.



(a) Attention diagram

(b) Static Variables importance

(c) Encoder Variables importance

(d) Decoder Variables importance

Figure 5.24: Interpretability results for TFT, considering grocery category

Furthermore, the buy-one-get-one promotion does not seem an effective strategy for this category, given that negative SHAP values are observed when the promotion is active, leading to lower predicted unit sales. The force plot for a random prediction in Figure 5.26c supports this analysis, showing that the absence of promotions negatively impacts the model output. Interestingly, when discount promotions are inactive, it positively affects predicted unit sales, indicating that this promotional activity may not fit this category best.

Next, the LSTM model, interpreted using the SHAP library, reaffirms the significant impact of the discount promotion on the model's output. It is followed by 'no_promo', buy-one-get-one, and circular everyday low-price promotions. In addition, the LSTM model highlights the significant role of lagged unit sales values in determining the model's output, as depicted in Figure

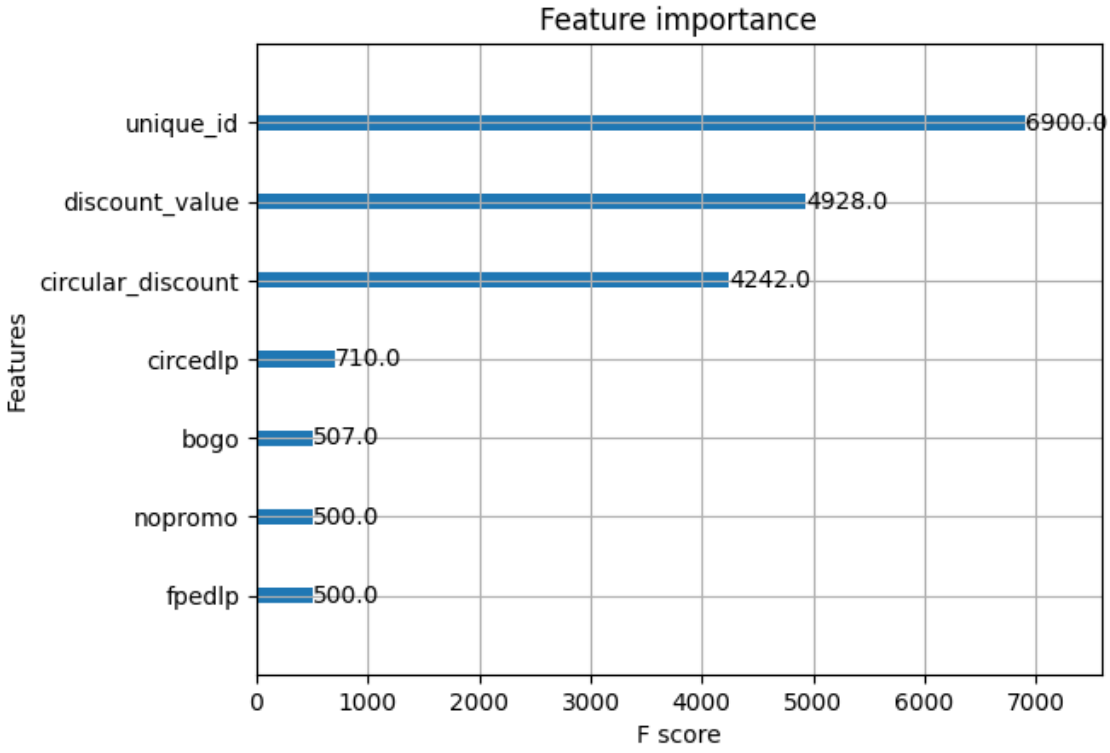
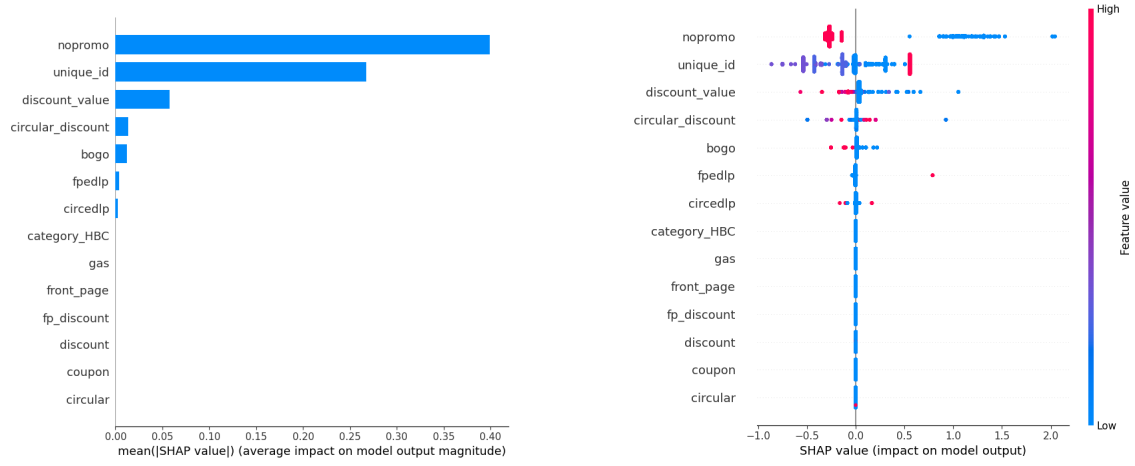
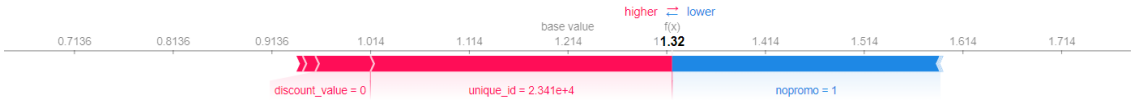


Figure 5.25: Feature importance for XGBoost, considering HBC category



(a) Summary Plot

(b) Feature importance

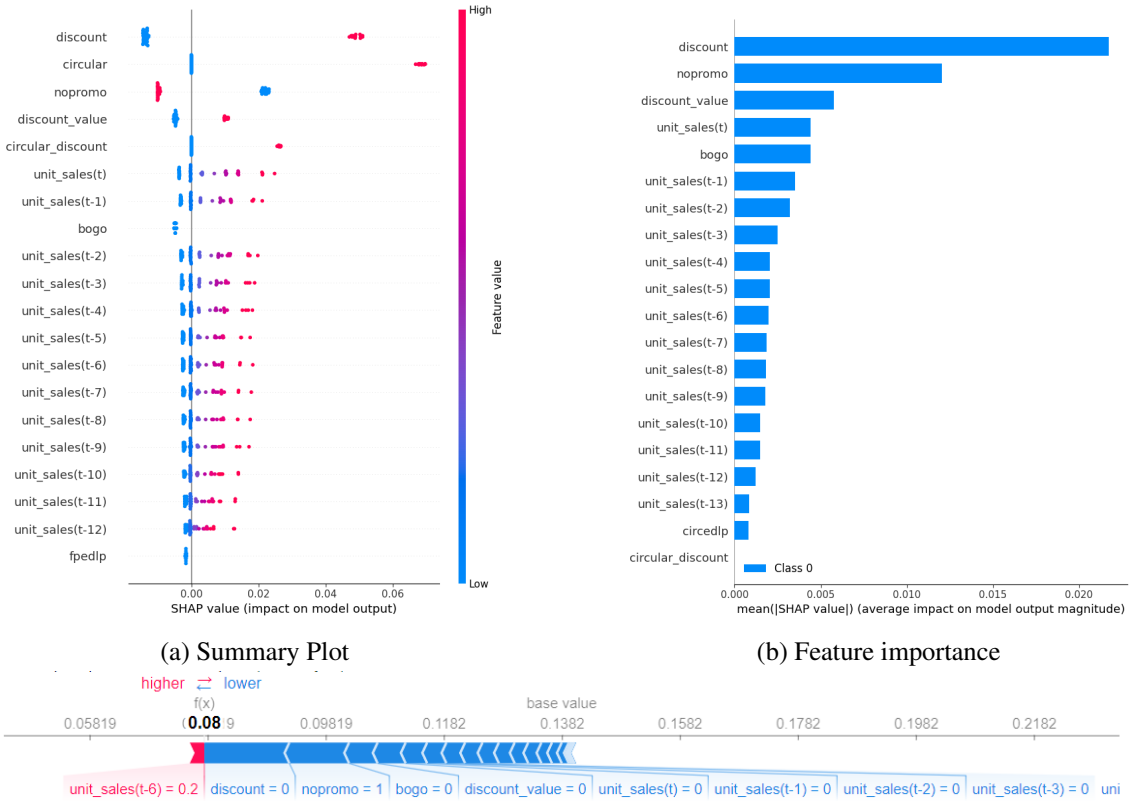


(c) Force Plot

Figure 5.26: SHAP results for XGBoost, considering HBC category

5.27. In contrast to the findings from the XGBoost model, the LSTM model suggests that active discount promotions yield positive SHAP values, thereby enhancing model predictions and poten-

tially boosting sales. Similar behavior is noted for circular promotions. Moreover, lagged unitary sales conform to the patterns observed in other categories, with higher past values positively impacting the model output.



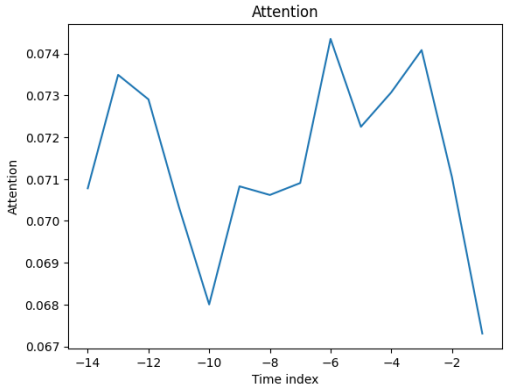
(a) Summary Plot

(b) Feature importance

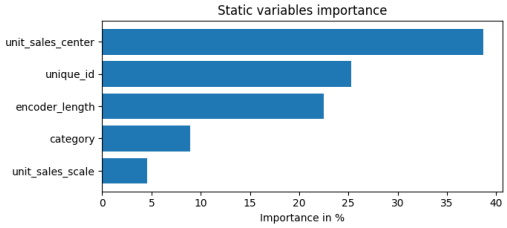
(c) Force Plot

Figure 5.27: SHAP results for LSTM, considering HBC category

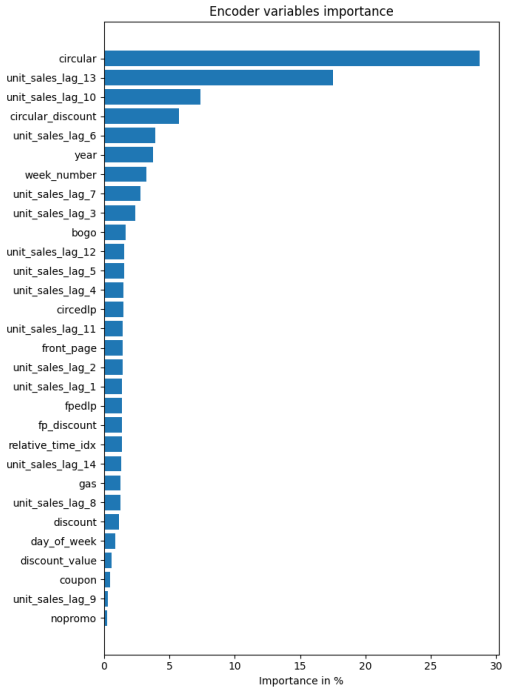
Finally, the TFT model's analysis of encoder variable importance in Figure 5.28c identifies circular promotions as the most impactful feature, followed by lagged unit sales for days 14, 11, and 6 before prediction. Interestingly, 'no_promo 'has negligible impact on encoder importance. When considering decoder variable importance, front page, discount, front-page everyday low price, coupon, buy-one-get-one, and circular promotions emerge as the most impactful, along with lagged unit sales. Lagged unit sales for the six days before the predictions emerged as the most significant feature, illustrated in figure 5.28d.



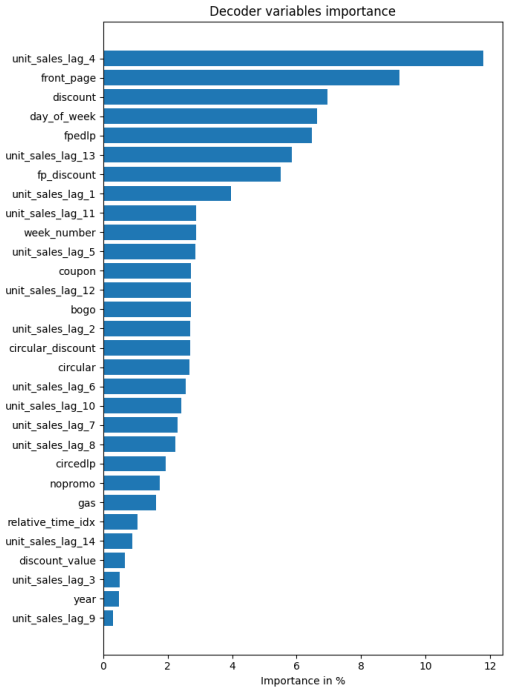
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.28: Interpretability results for TFT, considering HBC category

5.2.2.7 Meat

As illustrated by the feature importance chart generated by the XGBoost model in Figure 5.29, considering the Meat category, circular, discount, and front-page promotions exhibit the most significant influence on the model's predictions for item sales within this category. The 'no_promo' feature closely follows these, while circular everyday low-price, buy-one-get-one, front-page everyday low-price, and coupon promotions are considered less critical. The SHAP feature importance plot reinforces these findings, revealing front-page, discount, and circular promotions as having the most impact on the model. Interestingly, 'no_promo' outperforms the previously mentioned promotional activities in terms of its impact.

The summary plot in Figure 5.30a mirrors the 'no_promo' feature behavior observed in other

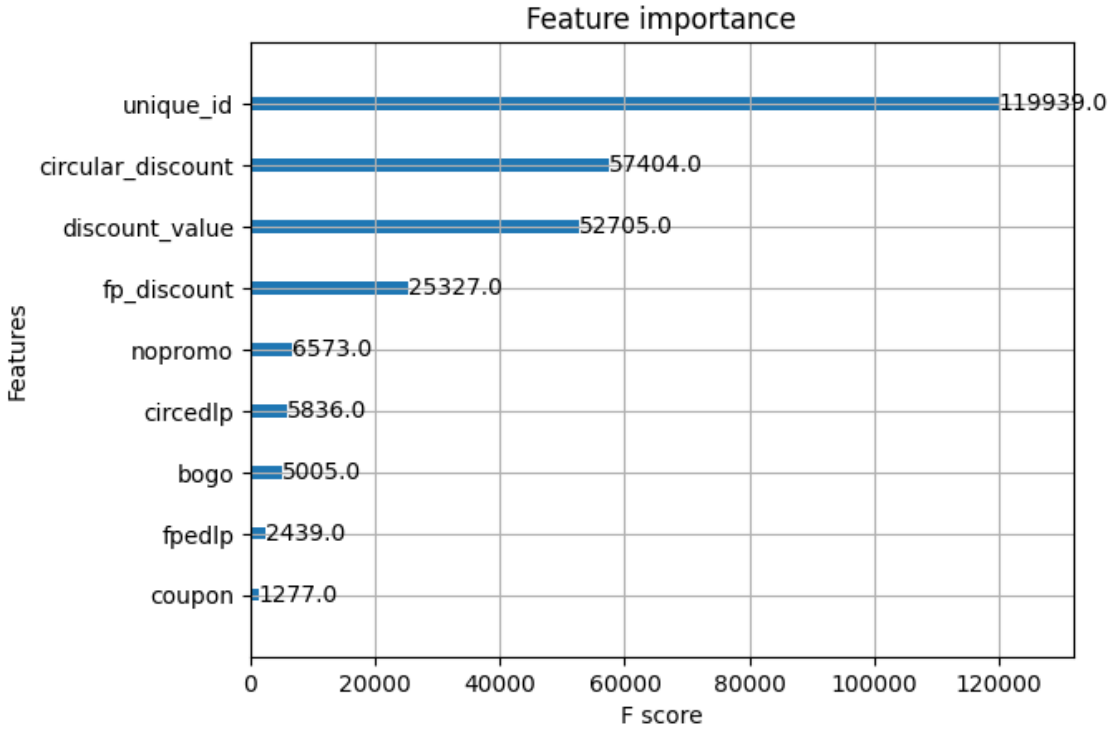
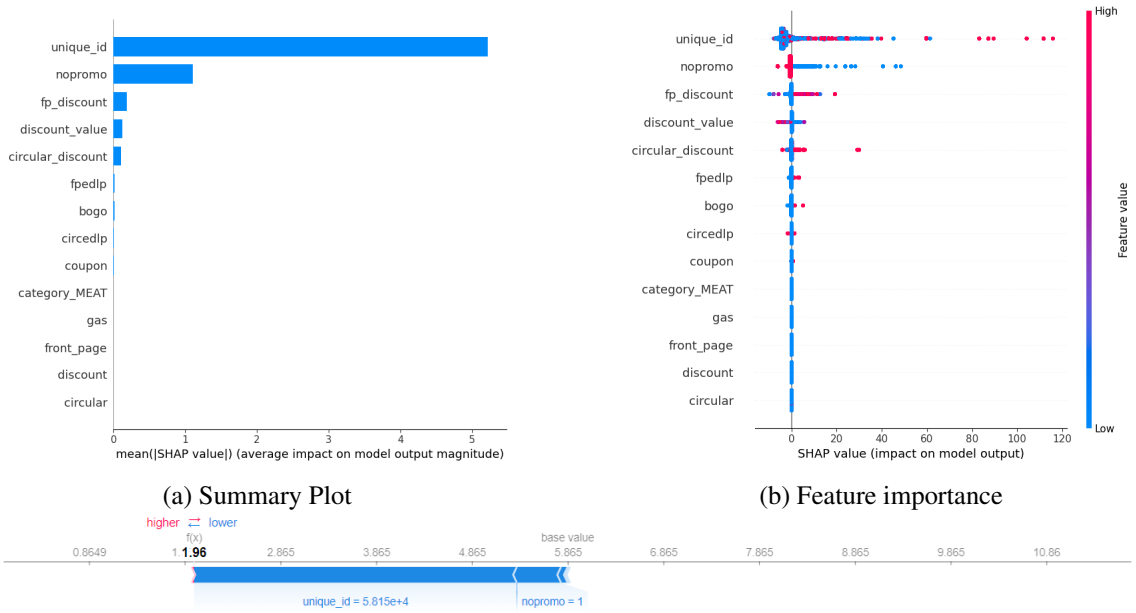


Figure 5.29: Feature importance for XGBoost, considering Meat category

categories. Moreover, front-page discounts appear to positively impact the model's output, as indicated by positive SHAP values when the promotion is active, which subsequently translates into an anticipated boost in sales. This pattern is less pronounced but still present for circular, front-page everyday low-price, and buy-one-get-one promotions. The force plot in Figure 5.30c shows that the absence of active promotions negatively impacts the prediction for a random item within this category, once again highlighting that the lack of promotional activities might result in decreased sales.

Further, the LSTM model's feature importance plot, demonstrated in Figure 5.31b, emphasizes that lagged unit sales for the day before the prediction bears the most significant influence on the model's outcome. This is followed by the 'no_promo' feature. Similarly to the XGBoost interpretability results, the LSTM model indicates that circular and discount promotions substantially impact the model's output among all promotional activities. The model's output is also significantly affected by the lagged unit sales values for various days, a pattern consistent with the LSTM model's behavior across other categories. The summary plot in Figure 5.31a reaffirms the same behavior for unit sales lagged values, i.e., higher past sale values are correlated with higher SHAP values, resulting in a positive impact on the model's output and higher predicted unitary sales. The 'no_promo' feature again behaves similarly to its analysis in other categories. Furthermore, the plot also suggests that circular promotions positively affect the model's output, indicating their desired effect on item sales within this category.

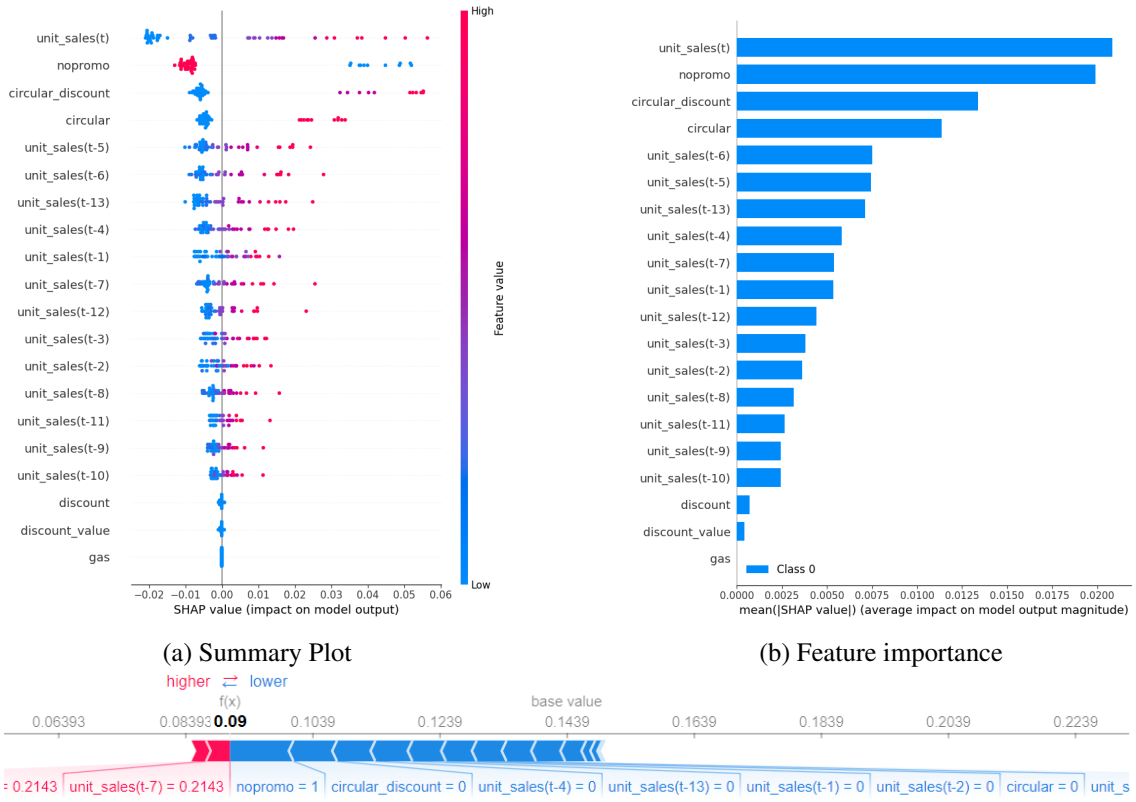


(a) Summary Plot

(b) Feature importance

(c) Force Plot

Figure 5.30: SHAP results for XGBoost, considering HBC category



(a) Summary Plot

(b) Feature importance

(c) Force Plot

Figure 5.31: SHAP results for LSTM, considering Meat category

Finally, the TFT model's encoder variable importance plot, depicted in Figure 5.32c, identifies lagged unitary sales for 13 days before the prediction as the most impactful feature. This is followed by circular everyday low-price and lagged unitary sales for 7 and 10 days prior. As per the decoder variable importance analysis, illustrated in Figure 5.32d, lagged unitary sales for two days before the prediction emerged as the most impactful feature, succeeded by 'no_promo'. The analysis also highlights front-page, gas, buy-one-get-one, and discount promotions as the most significant contributors to the model's output.

These results collectively suggest a close relationship between the promotional activities and the predictive accuracy of the models. The findings further underscore the importance of considering historical sales data in refining the forecast accuracy, a pattern consistent across various product categories.

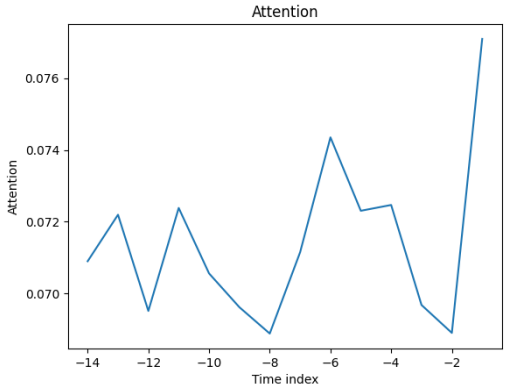
5.2.2.8 Produce

Exploring the Produce category highlights key results regarding the interpretability of the models used for sales predictions. As shown in the XGBoost model's feature importance chart in Figure 5.33, circular, discount, and front-page promotions significantly influence the model's predictions. These are followed by circular everyday low-price, buy-one-get-one, front-page everyday low-price, and coupon promotions, while the 'no_promo' feature is less significant. In contrast, the SHAP feature importance plot in Figure 5.34b highlights 'no_promo' as the most impactful feature, succeeded by circular, discount, and buy-one-get-one promotions.

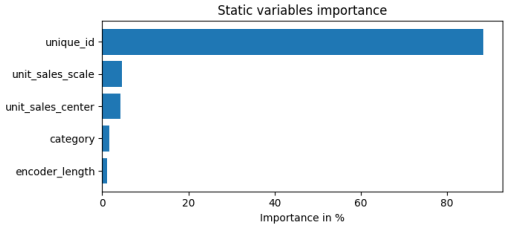
However, the summary plot in Figure 5.34a demonstrates inconsistent behaviour for the 'no_promo' feature, thus making it challenging to predict its effect on the model's output. Similarly, the impact of the most influential promotional activities on the model's output appears irregular, as their activation could lead to either a positive or negative effect. Notably, only circular everyday low-price and coupon promotions consistently positively impact the model's output, suggesting that their activation might prompt higher predicted sales.

The LSTM model's feature importance plot depicted in Figure 5.35b identifies lagged unitary sales as the most impactful features, a pattern that aligns with the findings across other categories. This is followed by 'no_promo', discount, and circular promotions. The summary plot, shown in Figure 5.35a, confirms the anticipated behavior for the lagged unit sales values—higher past sales values correspond to positive SHAP values, prompting an increase in the predicted unitary sales. Uniquely to this category, the 'no_promo' feature results in a positive SHAP value when it equals 0, implying that the presence of any promotion might stimulate an uplift in predicted sales. Furthermore, discount promotions exert a positive influence on predicted sales, with the model forecasting higher sales during the promotion period.

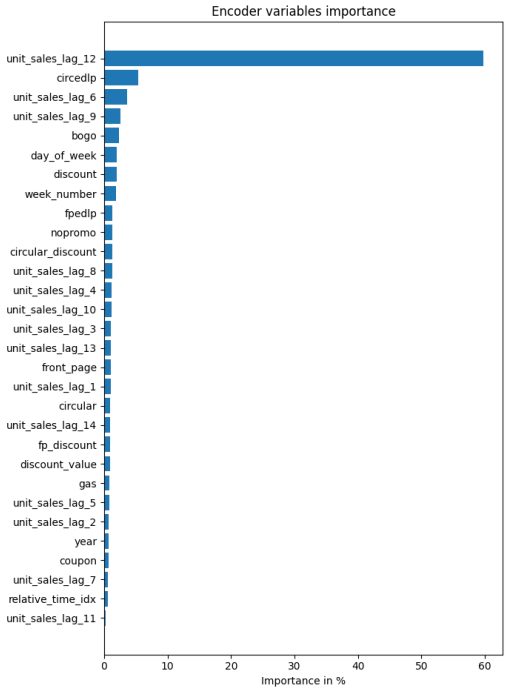
Lastly, as demonstrated in Figure 5.36c, the TFT model's encoder variable importance plot identifies lagged unitary sales for 2 and 13 days before the prediction as the most impactful features. These are closely followed by circular promotions and lagged unitary sales for 7 and 14 days



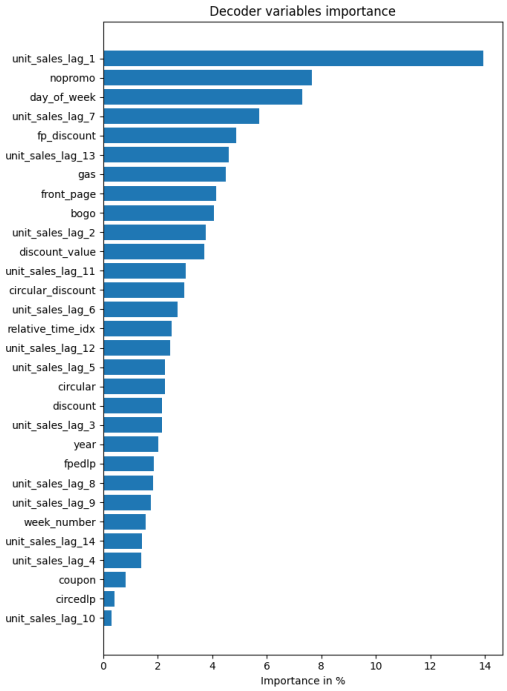
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.32: Interpretability results for TFT, considering Meat category

prior. As for the decoder variable importance, illustrated in Figure 5.36d, the most impactful features are lagged unitary sales, while the most influential promotional activities include discount, circular everyday low-price, buy-one-get-one, and front-page promotions.

In conclusion, these findings collectively underline the essential role of promotional activities and historical sales data in shaping the predictive performance of the models. The influence of different promotions varies across the models and categories, underlining the need for a nuanced understanding of these dynamics for more accurate forecasting. Notably, the observed inconsistencies in the 'no_promo' feature's impact across models further emphasize the importance of considering model-specific insights to formulate effective promotional strategies in the Produce category.

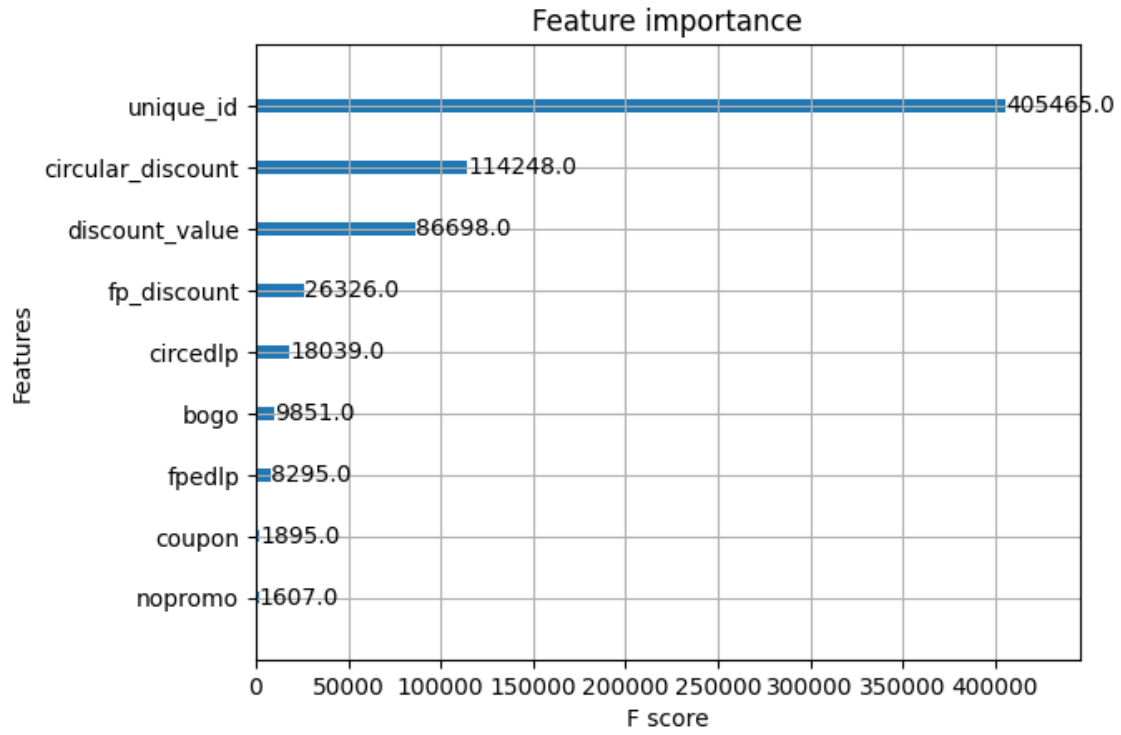
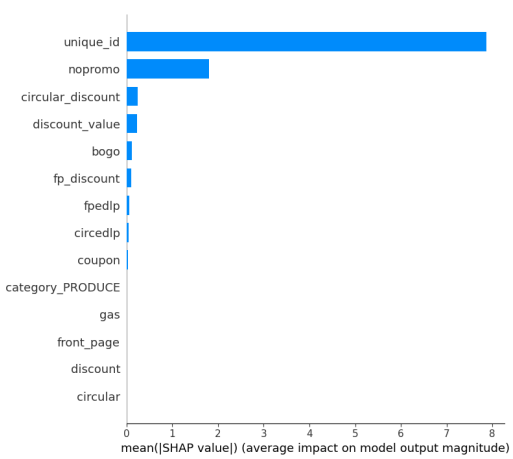
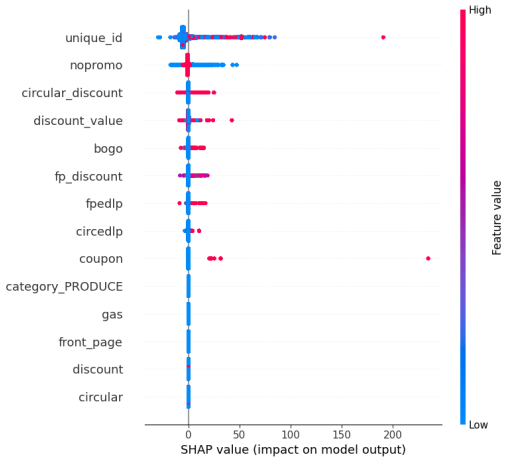


Figure 5.33: Feature importance for XGBoost, considering Produce category



(a) Summary Plot

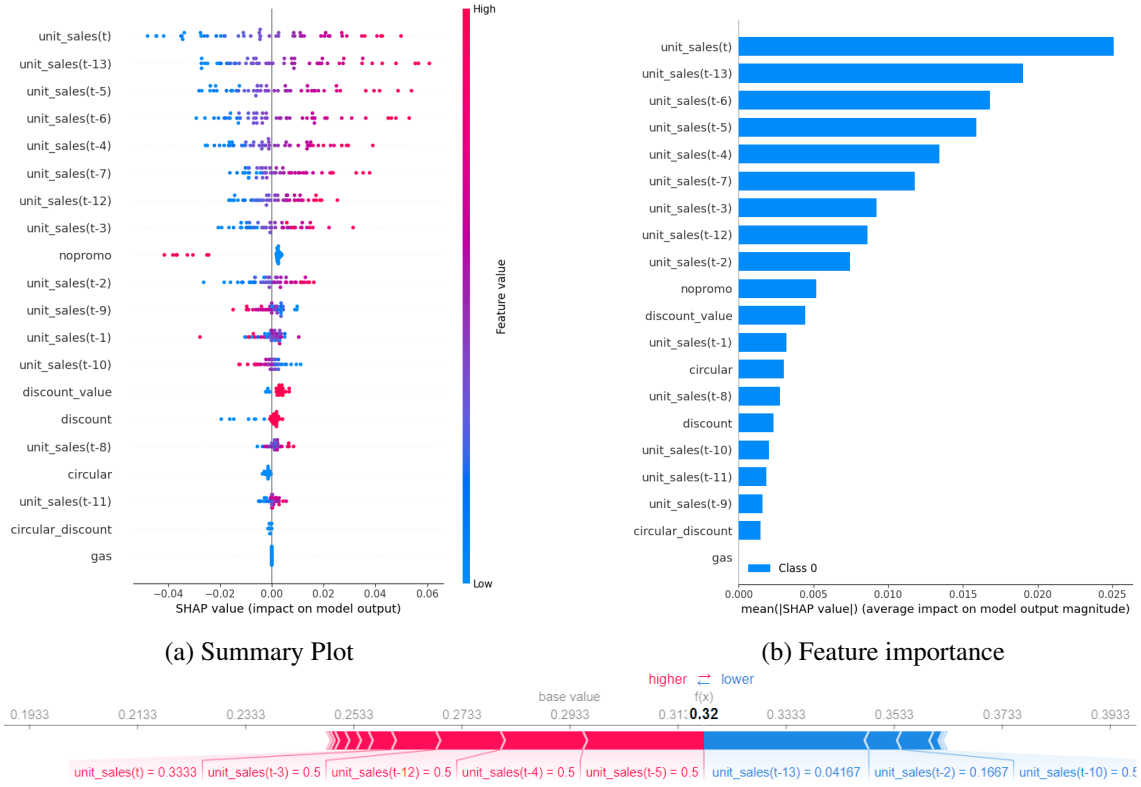


(b) Feature importance



(c) Force Plot

Figure 5.34: SHAP results for XGBoost, considering Produce category



(a) Summary Plot

(b) Feature importance

(c) Force Plot

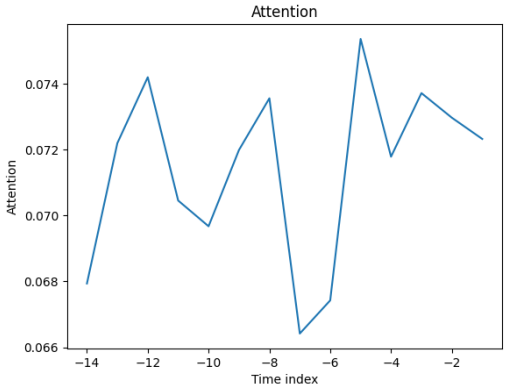
Figure 5.35: SHAP results for LSTM, considering Produce category

5.2.3 Item Level

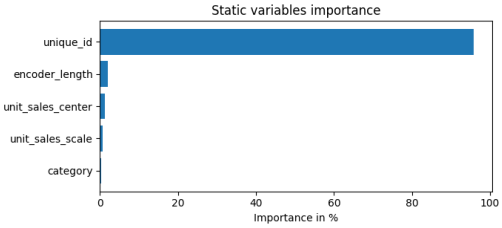
In the subsequent sections of this study, we transition our focus from global and category interpretability analysis towards an investigation of interpretability at the individual time series level. Conducting an item-level interpretability analysis presents an intriguing opportunity for several reasons. It allows us to unearth granular insights, understand the unique behaviour and performance of individual items, and uncover the effectiveness of promotional activities at a more detailed level. This level of analysis offers a more comprehensive picture, helping to craft more targeted and effective sales and promotional strategies for specific products.

5.2.3.1 Most Promoted item

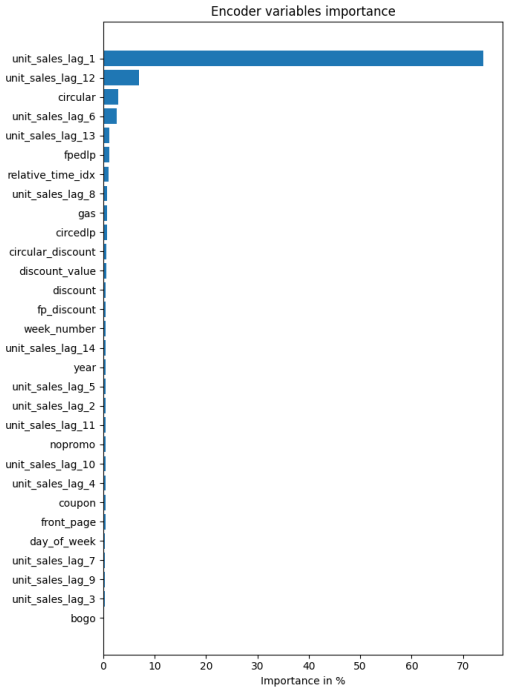
For this initial item-level analysis, we have chosen to study the most promoted item within the Grocery category. This item was promoted for 734 days during the 761 days encompassed by our dataset. Beginning again with the XGBoost feature importance analysis, as depicted in figure 5.37, we find that the circular everyday low-price promotion has the most significant impact among all promotional activities when considering the F score metric. This is followed by front-page everyday low-price, buy-one-get-one, and discount promotions.



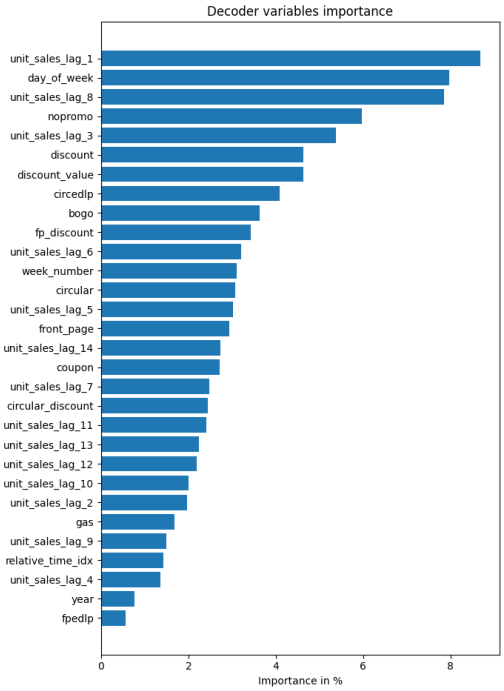
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.36: Interpretability results for TFT, considering Produce category

The SHAP feature importance plot, illustrated in figure 5.38b, identifies the same promotional activities as particularly relevant for this product. However, this plot positions the buy-one-get-one promotion as having the most significant impact on the model. Yet, the summary plot, presented in figure 5.38a, shows that the buy-one-get-one promotion may not be an effective strategy for this item. When this promotion is active, the plot suggests a negative SHAP value, which negatively impacts the model's output. In other words, the model predicts fewer unitary sales on the days when this promotion was active. In contrast, the circular everyday low-price and front-page everyday low-price promotions generate a positive impact when active, with the former promotion proving the most beneficial for sales uplift. As previously suggested, the force plot for a random prediction, shown in figure 5.38c, demonstrates that the buy-one-get-one promotion positively im-

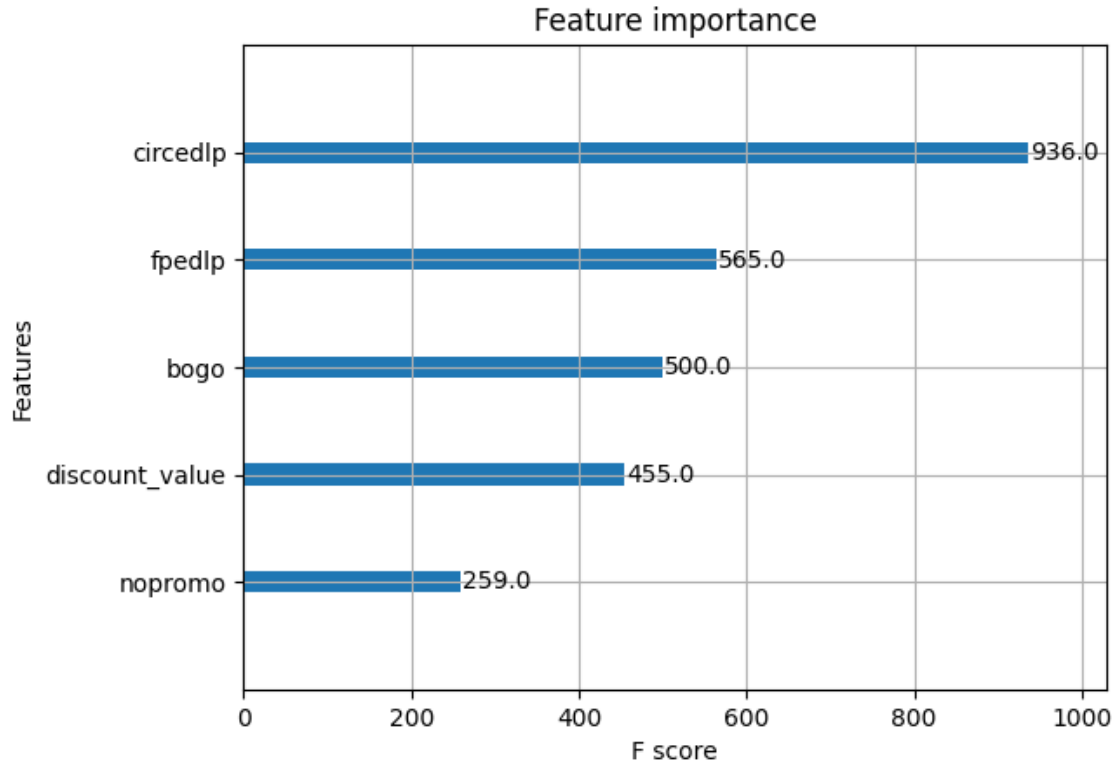
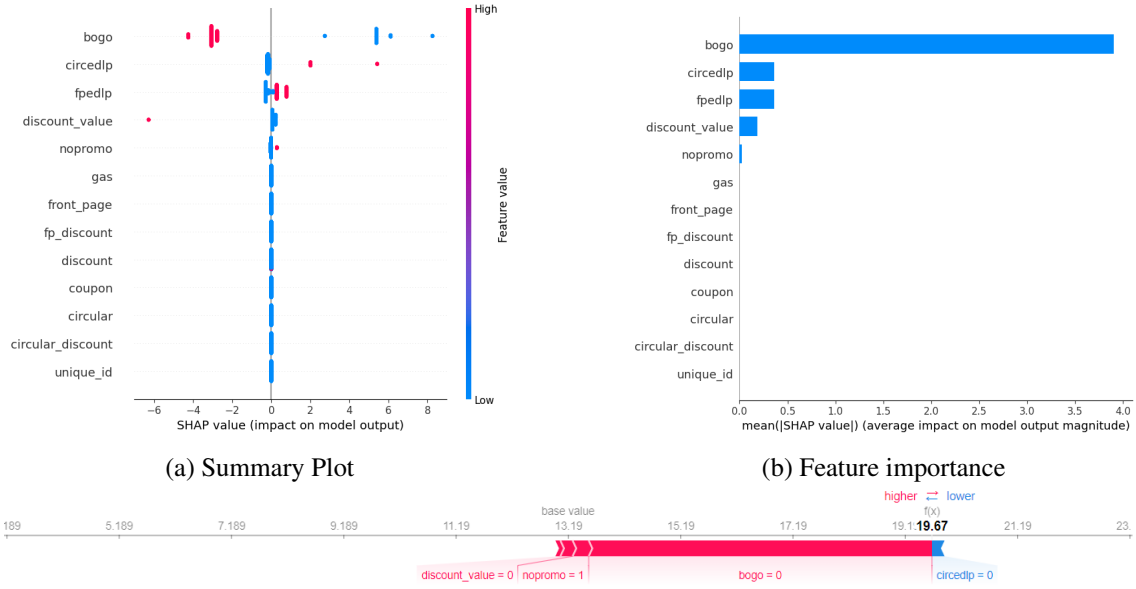


Figure 5.37: Feature importance for XGBoost, considering the most promoted item

pacts the model when it is inactive. Along with other features, for this specific prediction, the base value was 13.19 unit sales (a scaled value), and the model predicted 19.67 unit sales. The primary contributors to this change were the absence of the buy-one-get-one promotion and the fact that no other promotion was active on this day.

In addition to these analyses, we introduce the force plot for multiple days, visualized in figure 5.39a. Like the force plot for a single prediction, this force plot enables us to track changes in the model's output caused by each feature across the test set. When examining each feature individually, the force plot for the buy-one-get-one promotion (figure 5.39b) shows that the model forecasted higher sales than the base value during the initial inactive promotion days. As expected, when this promotion was active, the model forecasted lower values than the base value, reinforcing the conclusion that this promotion does not effectively boost sales for this item.

Furthermore, the plot in figure 5.39c reveals the circular everyday low-price promotion behavior across the test set. It can be observed that this promotion generates a positive impact when active on both the initial and final days of the test set. The same behavior is kept for the front-page everyday low-price promotion. However, it is also apparent that the effects of this promotion can vary over time. As shown in figure 5.39d, despite the positive impact of this promotion, its significance was not substantial compared to the following days when the promotion was active. It is essential to acknowledge that these variations could be attributed to several external factors that drive consumer behavior.



(a) Summary Plot

(b) Feature importance

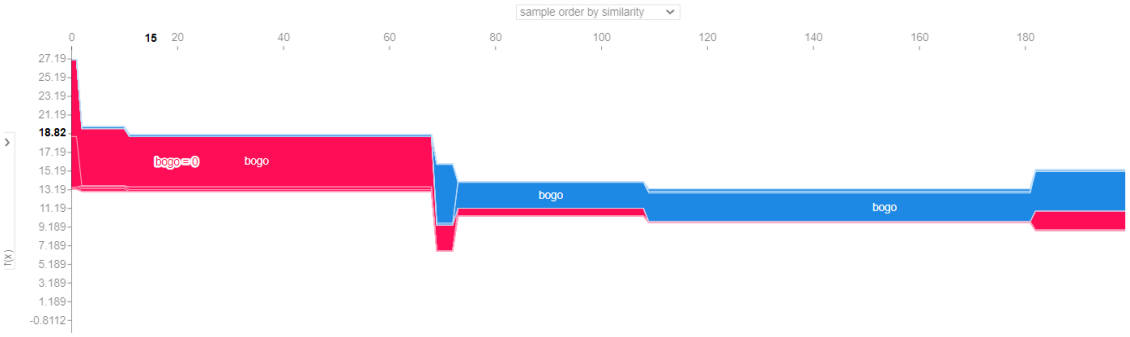
(c) Force Plot

Figure 5.38: SHAP results for XGBoost, considering the most promoted item

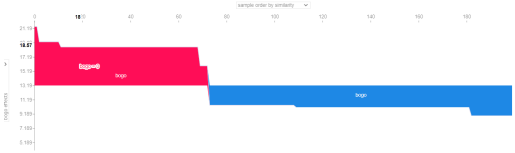
Lastly, as displayed in figure 5.39e, a 12% discount had a negative impact on sales for this item on day 70, even though this was the sole day when this promotion was applied. This result underscores the complexity of promotional strategies and the importance of understanding their impact at an item level to enable more effective decision-making.

In continuation of the analysis, we now apply the same process to the LSTM model. The feature importance plot depicted in figure 5.40b emphasizes the significance of the buy-one-get-one, front-page everyday low-price, and circular everyday low-price promotions, echoing the results obtained from the XGBoost model. This corroboration underscores the pivotal role these promotions play in driving the sales of this item. Further, the summary plot in figure 5.40a highlights a similar pattern for these promotions. It underscores that the buy-one-get-one promotion does not positively impact the sales of this item, as the SHAP values tend to be negative when this promotion is active, thereby predicting a decrease in unit sales. Conversely, the front-page everyday low-price and circular everyday low-price promotions emerge as beneficial, consistent with the insights drawn from the XGBoost model's interpretability analysis. The force plot for one prediction, illustrated in figure 5.40c, demonstrates the positive influence of the front-page everyday low-price promotion when it is active. It also importantly highlights the favorable effect resulting from the absence of the buy-one-get-one promotion.

Transitioning to the force plot for multiple predictions, as introduced earlier in the context of the XGBoost model, figure 5.41 reveals the varying impacts exerted by each feature on different predictions. As this analysis targets a single time series and the projections are arranged chronologically, we can track these effects throughout the testing period. Figure 5.41b focuses on the buy-one-get-one promotion. In the initial days when the promotion was inactive, a positive impact is observed, followed by a negative effect when the promotion is activated - a pattern anticipated



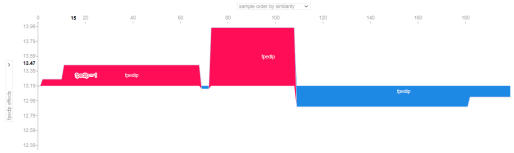
(a) Force Plot for multiple predictions



(b) Buy-one-get-one promotion effects



(c) Circular every day low-price promotion effects



(d) Front-page every day low-price promotion effects



(e) Discount promotion effects

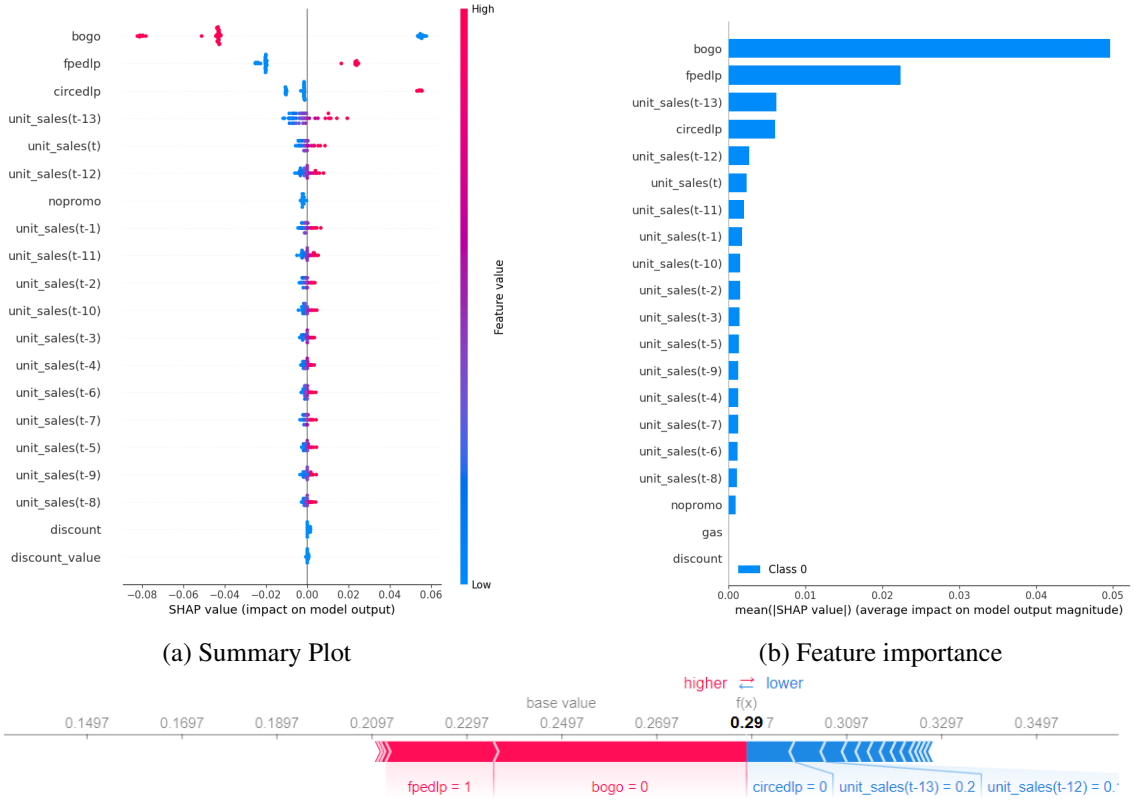
Figure 5.39: SHAP results for XGBoost, considering the most promoted item

from the XGBoost model's analysis. Figures 5.41d and 5.41c, corresponding to the front-page everyday low-price and circular everyday low-price promotions, exhibit a dichotomy of impacts. The prevalence of red indicates a positive influence on the model's output when these promotions are active. Conversely, their inactivity corresponds to a negative impact.

The analysis of the discount promotion, as shown in figure 5.41e, also implies that this promotional strategy is not efficacious for this item. The model output experiences a positive impact when this promotion is not active and a negative impact when it is, mirroring the findings from the XGBoost model analysis.

Concluding the interpretability analysis with the TFT model, it's important to note that the insights gleaned from this model are less comprehensive and explanatory than those derived from the XGBoost and LSTM models. Despite this, significant findings emerge, such as the high impact of the circular everyday low-price promotion and the 'no_promo' feature on the encoder variables' importance, illustrated in the decoder variables importance in figure 5.42d; these results are consistent with the previous analyses for this item.

However, it's worth noting that the buy-one-get-one and front-page everyday low-price promotions did not rank as the most influential promotional activities in this model. Instead, the TFT model identifies gas and front-page promotions as having a more significant impact.



(a) Summary Plot

(b) Feature importance

(c) Force Plot

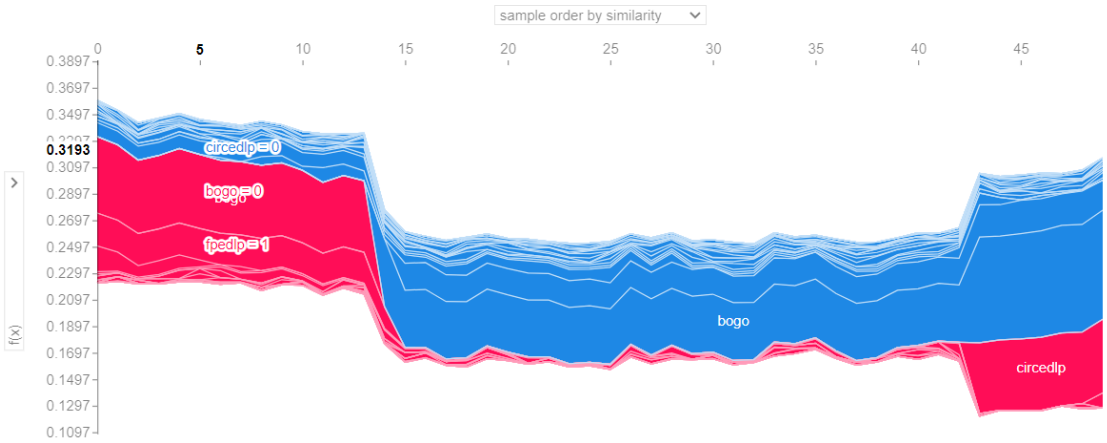
Figure 5.40: SHAP results for LSTM, considering the most promoted item

While these interpretability results might be less illustrative or directly applicable in real-world settings, the TFT model has demonstrated superior performance compared to the other models. This juxtaposition underscores the need to balance model interpretability and prediction accuracy, especially when applying these models in practical scenarios.

5.2.3.2 Second Most Promoted Item

Shifting our focus to the second most promoted item, we reinitiate our analysis with the XGBoost model. As shown in the feature importance plot in figure 5.43, circular everyday low-price promotions hold the most significant influence among all promotional activities. This is followed by front-page everyday low-price, buy-one-get-one, and discount promotions. The feature importance plot, depicted in figure 5.44b, corroborates this finding, identifying buy-one-get-one, front-page everyday low-price, discount, and circular everyday low-price promotions as the most relevant features for this model.

The summary plot, as presented in figure 5.44a, mirrors the ineffective behavior of the buy-one-get-one promotion as found in the analysis for the most promoted item. When this promotion is active, it generates negative SHAP values, negatively impacting the model's output. A similar



(a) Force Plot for multiple predictions



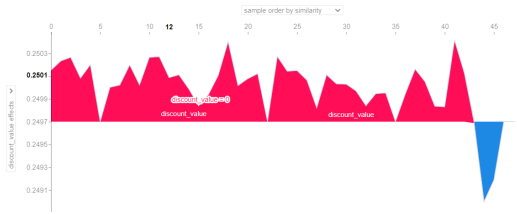
(b) Buy-one-get-one promotion effects



(c) Circular every day low-price promotion effects



(d) Front-page every day low-price promotion effects



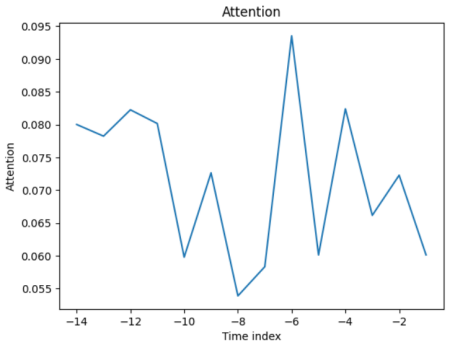
(e) Discount promotion effects

Figure 5.41: SHAP results for LSTM, considering the most promoted item

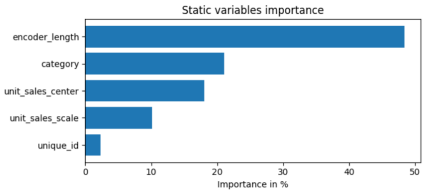
pattern is observed for the discount promotion. On the other hand, both front-page everyday low-price and circular everyday low-price promotions positively influence the predicted unitary sales of this model, according to the same figure.

Examining the force plot for one prediction within the test set, illustrated in figure 5.44c, we can see a clear reflection of the summary plot analysis. The model's output shows the significant impact of not having buy-one-get-one and discount promotions active.

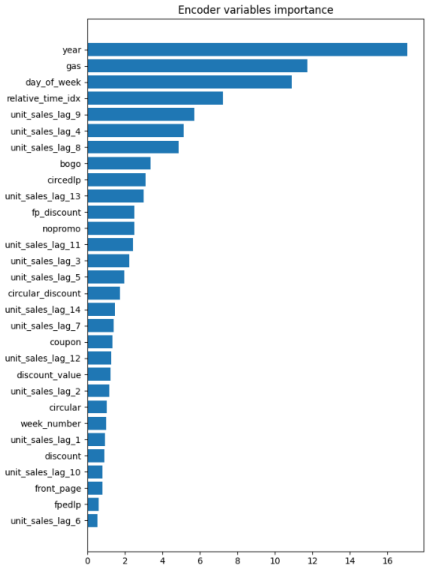
Furthermore, the force plot for multiple predictions, depicted in figure 5.45, provides a visual overview of these effects on the model's output across the test set. The red sections represent periods when the buy-one-get-one promotion was inactive, leading the model to forecast more sales than the base value. A similar pattern is observed for the front-page everyday low-price promotion, except its impact is positive when active. This analysis affords us a comprehensive understanding of the model's learnings. Indeed, this interpretability tool offers a valuable gateway into the workings of the model, making its predictions more accessible to retailers who often lack



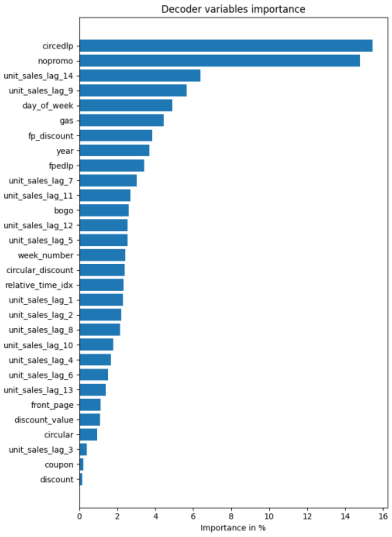
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.42: Interpretability results for TFT, considering the most promoted item

familiarity with these models.

Transitioning to the LSTM analysis, the insights gleaned about promotional events align with those of the XGBoost analysis. The feature importance plot in figure 5.46b identifies buy-one-get-one as the most impactful promotion, followed by front-page everyday low-price and circular everyday low-price. The summary plot in figure 5.46a confirms the same behavior for these promotions.

Concluding with the TFT model, the encoder variables importance plot in figure 5.48c indicates that lagged unitary sales for the preceding 9, 4, 10, and 8 days exert the most significant impact. Gas, discount, circular everyday low-price, and coupon promotions follow this. Moreover, the decoder variables importance plot, presented in figure 5.48d, identifies discount promotion as the most impactful feature, trailed by buy-one-get-one, front-page, and coupon promotions. It is worth noting the slight deviation in TFT results from the other models, underscoring the unique interpretive approach of each model.

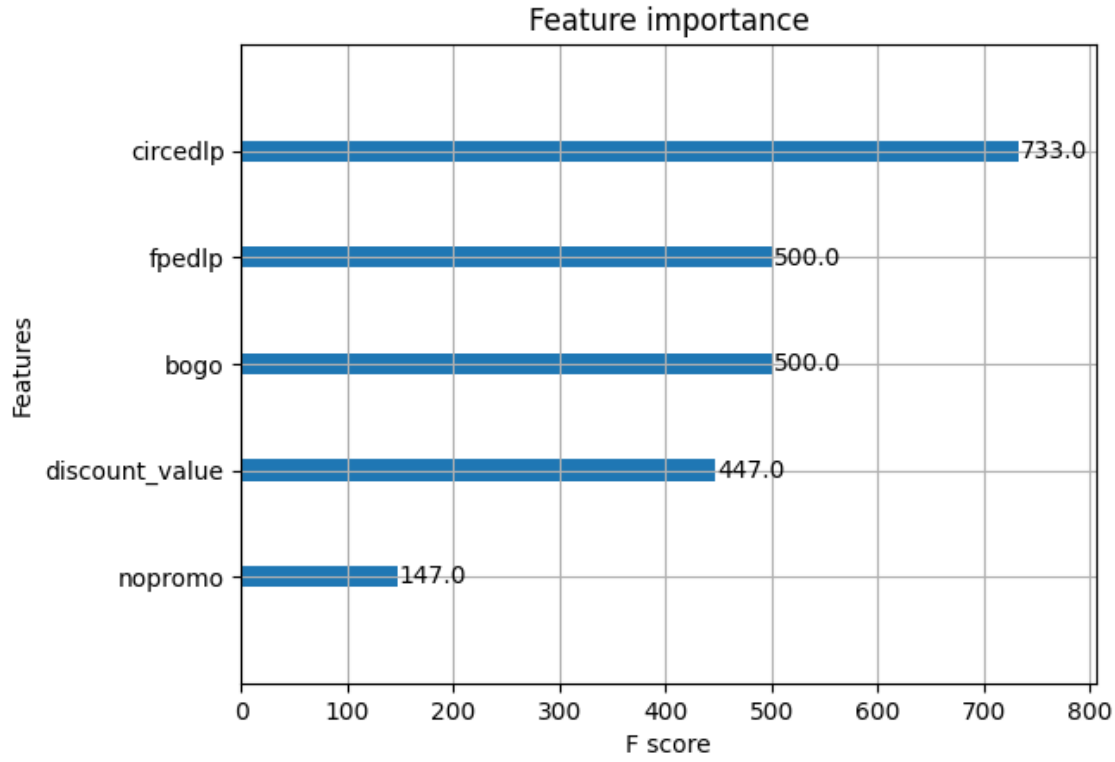
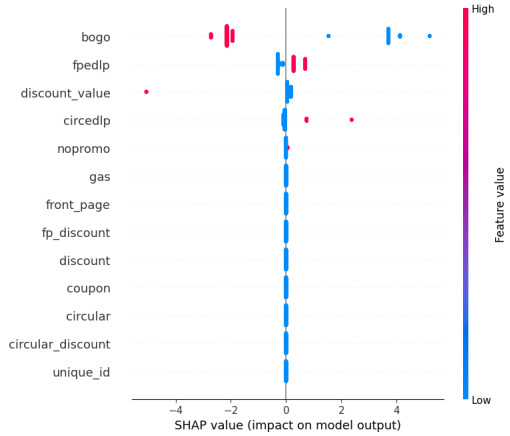
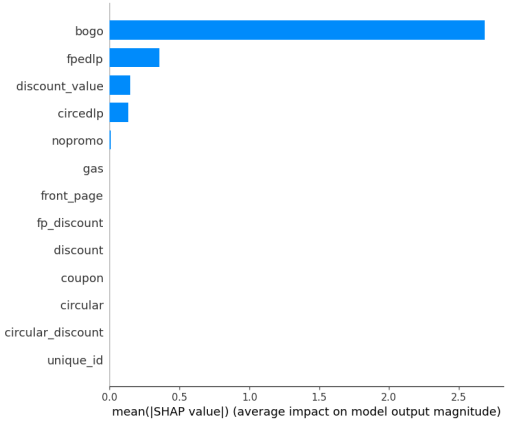


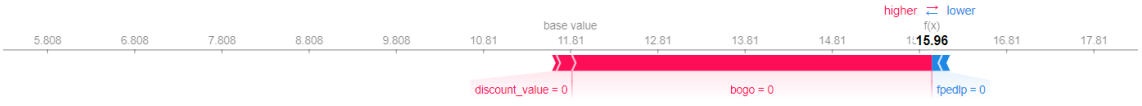
Figure 5.43: Feature importance for XGBoost, considering the second most promoted item



(a) Summary Plot

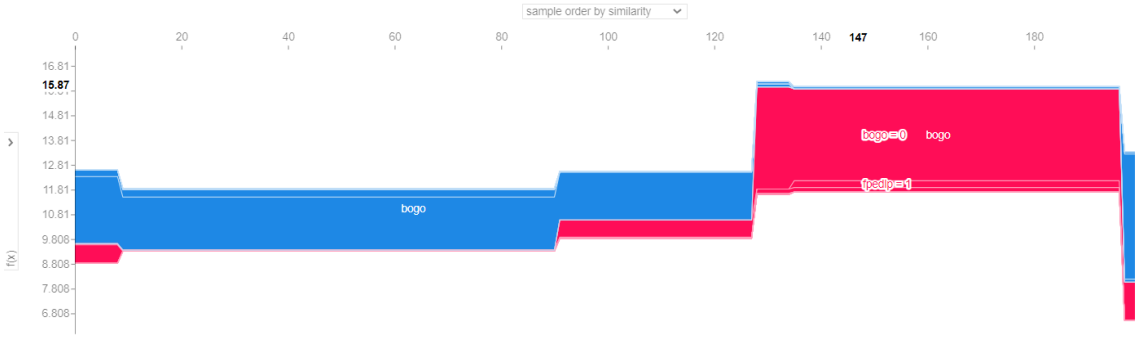


(b) Feature importance

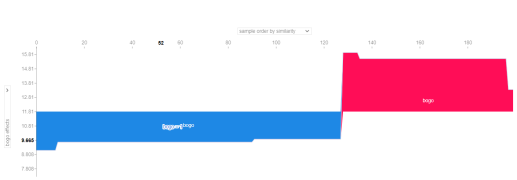


(c) Force Plot

Figure 5.44: SHAP results for XGBoost, considering the second most promoted item



(a) Force Plot for multiple predictions

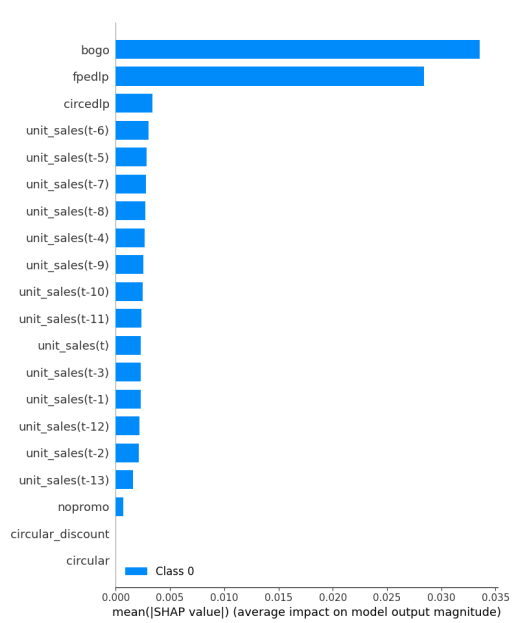


(b) Buy-one-get-one promotion effects

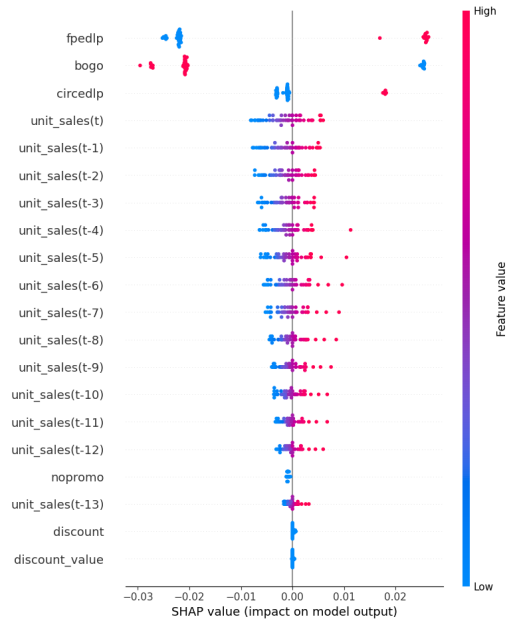


(c) Circular everyday low-price promotion effects

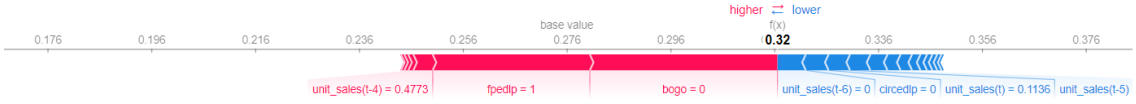
Figure 5.45: SHAP results for XGBoost, considering the second most promoted item



(a) Summary Plot



(b) Feature importance



(c) Force Plot

Figure 5.46: SHAP results for LSTM, considering the second most promoted item

5.2.3.3 Third Most Promoted Item

We conclude the item-level and extensive interpretability analyses as we focus on the third most promoted item. Initiated with the XGBoost analysis, the feature importance plot shown in figure

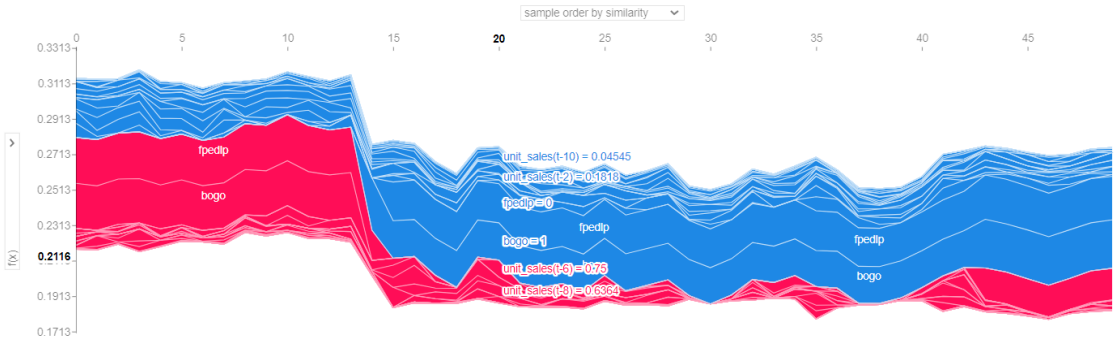


Figure 5.47: Force Plot for multiple predictions for LSTM, considering the second most promoted item

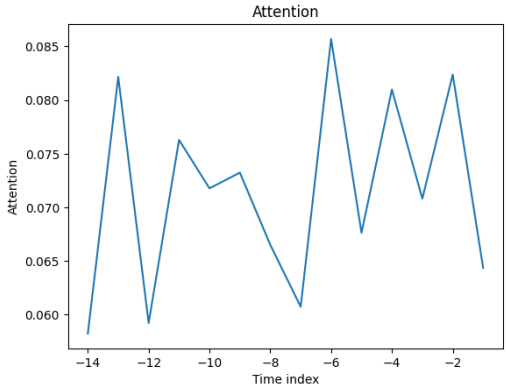
5.49 suggests that the circular promotion is the most impactful event. It is followed by discount, front-page, circular everyday low-price, and buy-one-get-one promotions, the latter being the least impactful. In accordance, the feature importance generated by the SHAP library also identifies these promotions as relevant but in a slightly different order, with placing discounts as the most impactful event, illustrated in figure 5.50b.

The summary plot, in figure 5.50a, reveals an irregular pattern for discount promotions where the same percentage discount can positively or negatively impact the model's output. Conversely, front-page, circular, and circular everyday low-price promotions generally positively impact model outputs when active. The analysis further confirms that the buy-one-get-one promotion does not impact the model's output positively. As these three most promoted items all fall within the grocery category and are the same drink from the same brand, differing only in size, this similarity could contribute to the identical promotional behavior. A more granular category-level analysis might enhance our understanding of the promotional behavior across these items.

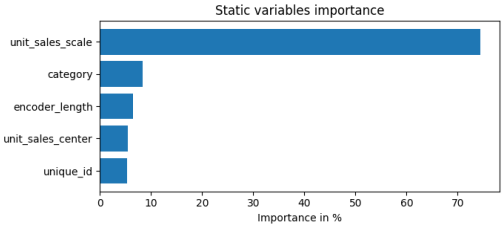
The force plot for a random prediction, shown in figure 5.50c, highlights the favourable impact of having no active discount promotions, suggesting that this may not be an effective promotion for this item. Notably, the absence of any active promotion resulted in a significantly negative impact, implying that active promotions could boost sales. Considering the previous analysis, front-page, circular, or circular everyday low-price promotions could potentially elevate sales figures.

Additionally, the force plot for multiple predictions support these conclusions about this item's most effective promotional events, as illustrated in figure 5.51. For example, on day 28 of the test set, a 12% discount resulted in a lower predicted value than the expected base value. Conversely, on day 1, a 30% front-page promotion positively impacted the model output, leading to a prediction higher than the base value of 13.36 unitary sales; this value is scaled.

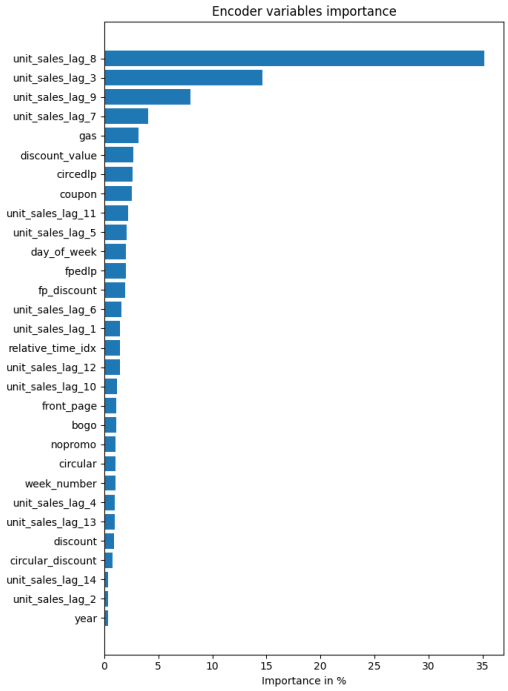
Turning to the LSTM analysis for the same item, the feature importance plot in figure 5.52b reveals that circular, discount, buy-one-get-one, and front-page everyday low price discounts were the most influential overall promotional events, along with lagged unitary sales values. The summary plot, portrayed in figure 5.52a, asserts that active circular promotions had the most significant positive effect on the model output among all identified important features. Similarly impactful



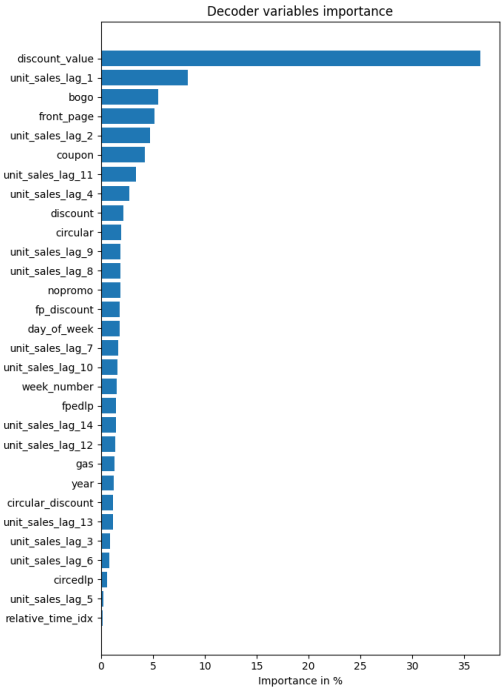
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.48: Interpretability results for TFT, considering the second most promoted item

were discount, circular, and discount promotions, albeit with less significant effects. The force plot for a random prediction, shown in figure 5.52c, reveals that an active discount promotion had a positive impact on this prediction.

Hence, the force plot for multiple predictions, represented in figure 5.53, confirms this behavior throughout the test set period for all analyzed features. Of note, on day 13, a 26% circular discount had a positive impact on the model output. These findings are consistent with those obtained for the XGBoost model.

Lastly, the TFT model's encoder variables importance plot in figure 5.54c indicates that discount, front-page, coupon, and gas promotions had a significant impact alongside lagged unitary sales values. As for the decoder variables' importance, in figure 5.54d, the feature relative to the

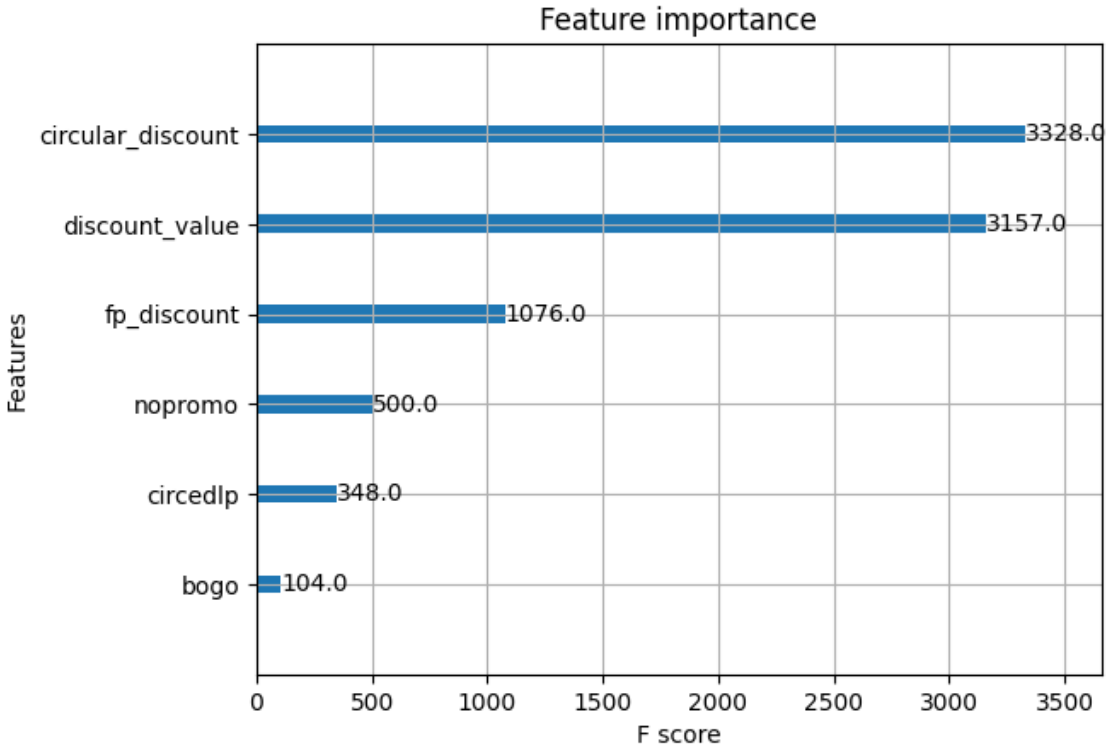
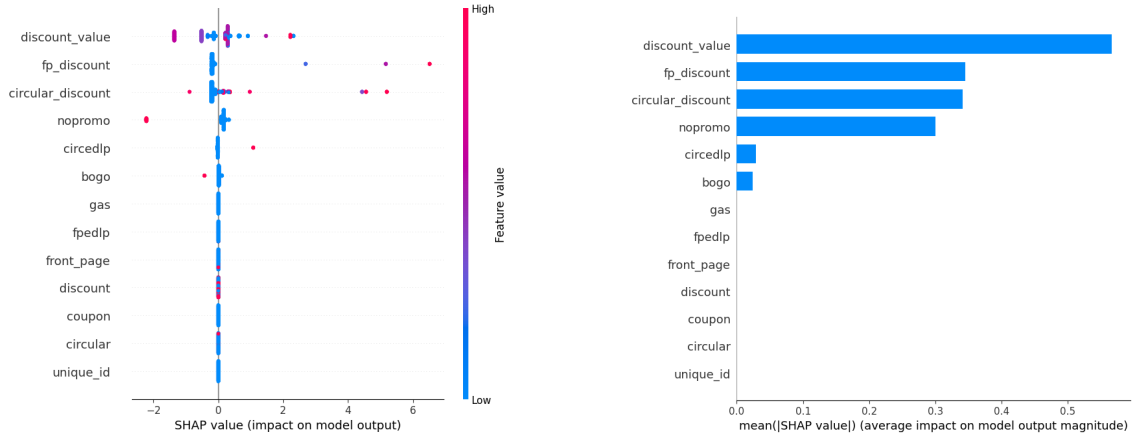


Figure 5.49: Feature importance for XGBoost, considering the third most promoted item



(a) Summary Plot

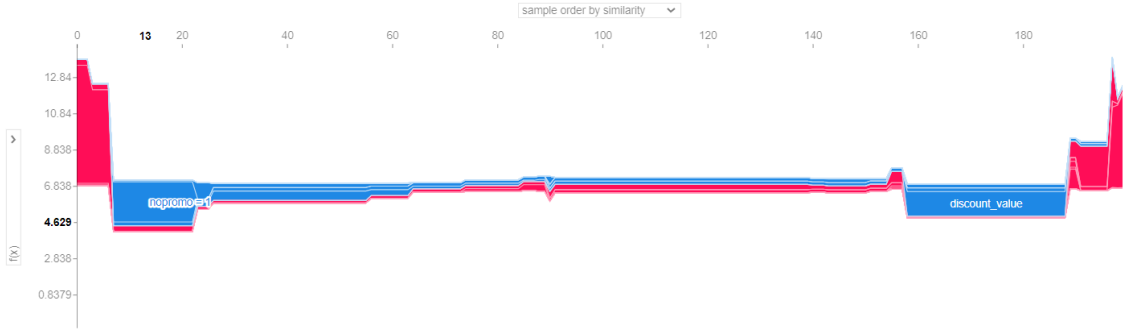
(b) Feature importance



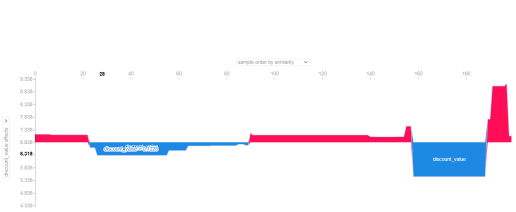
(c) Force Plot

Figure 5.50: SHAP results for XGBoost, considering the third most promoted item

year was the most impactful, followed by front-page everyday low-price, front-page, and circular everyday low-price promotions. These insights accentuate the parallels and divergences with the



(a) Force Plot for multiple predictions

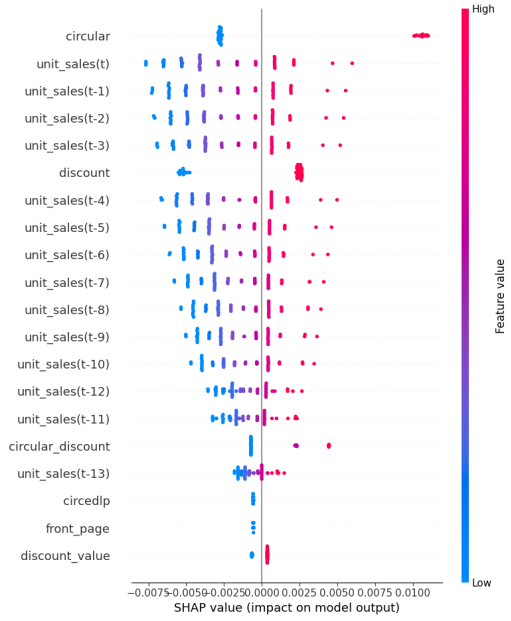


(b) Buy-one-get-one promotion effects

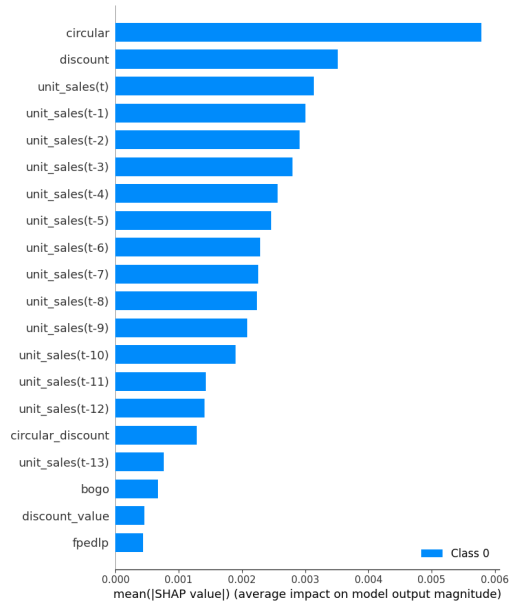


(c) Circular everyday low-price promotion effects

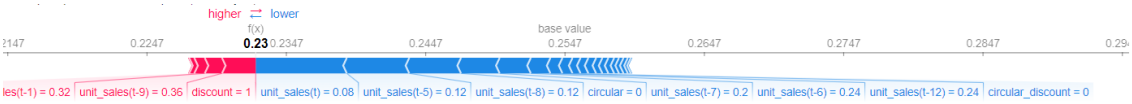
Figure 5.51: SHAP results for XGBoost, considering the third most promoted item



(a) Summary Plot



(b) Feature importance



(c) Force Plot

Figure 5.52: SHAP results for LSTM, considering the third most promoted item

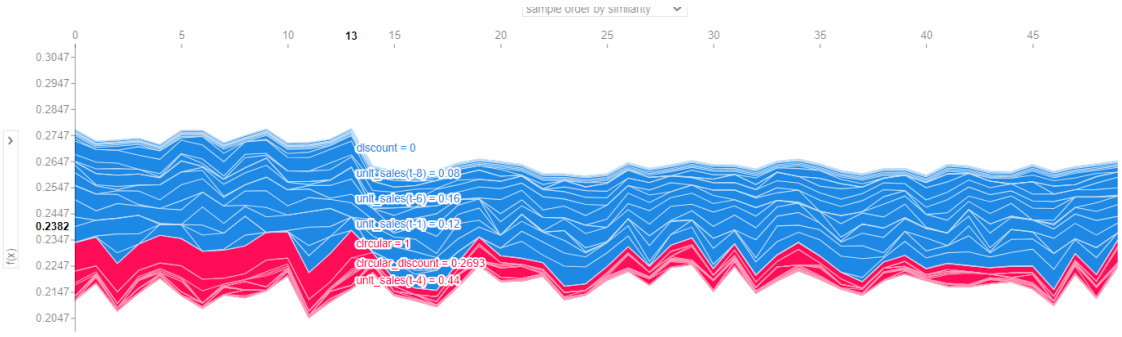
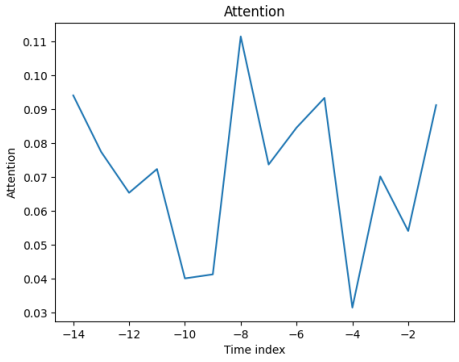
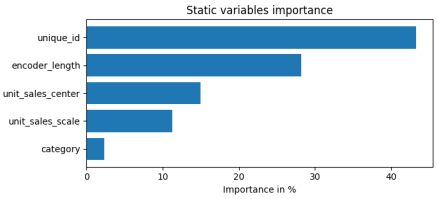


Figure 5.53: Force Plot for multiple predictions for LSTM, considering the third most promoted item

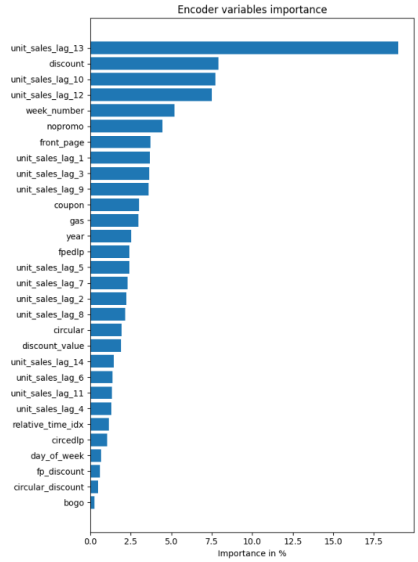
findings from the other models.



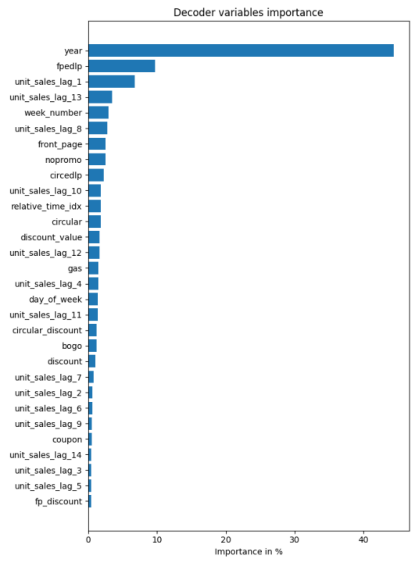
(a) Attention diagram



(b) Static Variables importance



(c) Encoder Variables importance



(d) Decoder Variables importance

Figure 5.54: Interpretability results for TFT, considering the third most promoted item

Chapter 6

Conclusions

In sum, promotions on a product affect its immediate sales and can affect the immediate sales of other products by complementarity or cannibalization, as well as the promoted product's future sales. That being the case, these complexities and promotional events happening simultaneously prevent highly accurate demand forecasts. Traditional approaches cannot separate and predict different individual promotion effects [3, 1]. Hence, the need to employ interpretable deep learning models to forecast demand capable of understanding the effects of various promotional events comes. Hence, considering state-of-the-art, the analyzed sales forecasting studies often use parametric and non-parametric models to make predictions considering the effects of promotional or advertising events. However, there is a lack of research that combines the evaluation of different promotions with interpretable deep learning models, which is important for understanding the effectiveness of different marketing strategies and designing future campaigns.

On that account, this study covers pre-processing methods, time series, and forecasting concepts. Thorough preparation and cleaning of the database proved critical in ensuring the successful implementation of subsequent steps in this project. Downcasting was performed, enabling the condensation of information from multiple files into a single file. This strategy optimized memory usage and improved the feature engineering process, wherein valuable features were developed from the existing dataset variables. These features were then fed into the Machine Learning models. The process was expedited significantly, enhancing the generation of graphics and plots for variable analysis.

In addition, the data was reformatted from its original state by melting the sales/demand data. This allowed for more straightforward utilization of built-in methods from Python libraries whenever data restructuring was necessary, thus facilitating data analysis. Efficient data reorganization is critical in aggregating sales of different products - by category or store. Cleaning the data involved replacing missing values in the numeric and categorical variables of the datasets. In this particular case, missing dates for each time series were added, with zero unit sales and no active promotional events being considered for these days. This substitution is crucial as many Machine Learning models do not accept null values in numerical variables. After cleaning, a comprehensive dataset analysis was conducted, revealing critical insights into the data's behavior. This analysis

was crucial to understanding which dataset variables significantly impact the target variable and are suitable as features in forecasting models. Moreover, it informed the steps to be taken during Feature Engineering to create valuable features from the selected variables. As a result, lagged unit sales of 14 days were added to the dataset. This dataset encompasses five stores, totalling 12,830 time series with daily unit sales over approximately two years (761 data points).

Moreover, a diverse set of both parametric and non-parametric models were examined. Statistical models were initially considered, making inferences based on the data's distribution. In contrast, machine learning methods were also utilized to make forecasts without explicit programming. These methods learn from raw data to generate predictive models. To establish a benchmark for our time series prediction tasks, SARIMAX, a traditional statistical method, was used. It is a model that can handle seasonal patterns, autocorrelation, and non-stationarity and, as such, is a widely recognized technique for time series forecasting. However, machine learning models were also considered, specifically XGBoost, LSTM (Long Short-Term Memory), and Temporal Fusion Transformer (TFT). Additionally, for the test set, a sliding window method was employed, wherein a span of 7 days was used to forecast one day ahead. This technique provided an efficient approach to future predictions while leveraging the advantages of both statistical and machine learning models.

Given that, traditional methods can be limited in their ability to capture non-linear relationships in time series data and may not be effective for large data sets that do not meet certain assumptions, such as linearity and normality [48]. Also, due to their complexity, machine learning methods can be challenging to fully understand how these make predictions, making interpretability an essential aspect of their use. Model evaluation was conducted based on their forecasting accuracy, employing MAE, RMSE, and MASE. The TFT model consistently outperformed SARIMAX, XGBoost, and LSTM when assessed with MAE and MASE metrics, with respective mean values of 1.83 and 0.71. However, for RMSE, LSTM outpaced other models with mean and median values of 2.49 and 1.77, respectively. This extensive evaluation suggests that the TFT model offers the most reliable and accurate forecasts, providing a valuable tool for effective retail inventory management and enhancing profitability.

Consequently, interpretability in machine learning enables the understanding of how the model generates predictions or decisions, mainly when used on time series data. This understanding allows us to explain the model's behavior [32], as there are, additionally to the complexity of the methods, several characteristics that can be part of the retailer's marketing strategy, such as promotions and their effects on consumers behavior, or driven by external factors over which the retailer doesn't have command. Not considering the demand influencing factors can result in low-accurate predictions.

Therefore, the interpretability of the predictions produced by various models, specifically XGBoost and LSTM, was explored using the SHAP (SHapley Additive exPlanations) library for Python. This library is a robust tool for interpreting the output of machine learning models. Application of the SHAP library to the selected models enabled the generation of force plots for a single prediction, multiple predictions, summary plots, and feature importance plots. These graphical

representations serve as valuable aids in interpreting the model.

Interpretability results were then scrutinized, starting from a general view and delving into granular details. Concerning global category interpretability outcomes, promotions are a crucial driver of unit sales across all models. Nonetheless, a disparity surfaces when considering the impact's direction. XGBoost correlates active promotions with a sales upswing, while LSTM links them with a sales dip. This incongruity could result from how each model processes and navigates temporal data dynamics.

Regarding category-specific interpretability, the prominence of circular, discount, front-page everyday low-price, and buy-one-get-one promotions are vital sales influencers within the Dairy category. The Deli category also reveals circular, buy-one-get-one, and front-page everyday low prices as critical sales catalysts, demonstrating a positive impact when active.

Additionally, the Frozen category emphasizes the relevance of circular and discount promotions as key sales indicators across models. Similarly, discounts and circular promotions consistently appear as significant sales drivers in the General Merchandise category. This uniformity across models highlights the pivotal role of these promotions in determining sales.

Conversely, in the Grocery category, the behaviour of discount and circular promotions is inconsistent, as higher feature values can yield different results, considering XGBoostoost outcomes. LSTM model suggests that elevated forecasted unit sales are observed when the discount is inactive, implying that this promotional activity might not substantially influence sales in this category. When circular promotions are inactive, negative SHAP values are observed, suggesting a potential negative impact on the model's results.

For the Health, Beauty, and Care category, discount and circular promotions are the most impactful promotional activities. Notably, buy-one-get-one promotions are ineffective, as the model's predictions drop when this promotion is active. The absence of promotions also adversely affects sales predictions.

For the Meat category, front-page discounts positively influence predictions, with a less pronounced but still present pattern for circular, front-page everyday low-price, and buy-one-get-one promotions. Finally, in the Produce category, the influence of promotional activities on the model's output is erratic, with activation potentially resulting in either positive or adverse effects. Only circular everyday low-price and coupon promotions consistently impact the predictions positively. (In summary, while there are variations across categories, commonalities can be observed in the global analysis, particularly in the consistent influence of specific promotional strategies on sales across multiple categories.)

Moving onto item-level analysis, we interpret the forecasts of the three most promoted items. Given that these items are from the same category and brand and differ only in size, this could lead to identical promotional behaviours. A more in-depth category-specific analysis could provide further insights into the promotional behaviours across these items.

For the top promoted item, buy-one-get-one promotions are ineffective and negatively impact model predictions. In contrast, circular and front-page everyday low-price promotions have a positive effect when active, with the former being the most beneficial for sales uplift.

Similarly, the second most promoted item shows the inefficacy of buy-one-get-one promotions, with a comparable pattern observed for discount promotions. Conversely, both front-page everyday low-price and circular everyday low-price promotions positively influence the predicted unit sales.

Finally, for the third most promoted item, an irregular pattern emerges for discount promotions where the same percentage discount can positively or negatively impact predictions. On the other hand, front-page, circular, and circular everyday low-price promotions effectively boost sales of this product. This information is critical for making informed decisions about promotional strategies in a retail context.

Appendix A

Additional Literature review

Title	Focus	Data	Variables	Forecast horizon and evaluation	Baseline Methods	Issues and Limitations	Conclusion
Characterizing retail demand with promotional effects for model selection	Selection of forecasting models that perform best on weekly data with promotional information	988 SKUs from 209 categories, with intense promotional activity that are seasonal and non-seasonal	Seasonal data, holidays, promotional events	Rolling forecast origin with 1-step ahead Evaluation: MAPE, MdAPE, MRMAE, MdRMAE, GMRMAE, MRRMSE, MdRRMSE, GMRRMSE, MASE, MdASE	ETS, TBATS, LASSO, ARIMA Fourier	Individual selection has potential to improve promotional events	Univariate models perform worse than correspondent multivariate models; TSxTSX is the best model based on the average rank
Improved supply chain management based on hybrid demand forecasts	Show improvements using hybrid systems; hybrid intelligent system combining Auto- regressive Integrated Moving Average (ARIMA) model and neural networks for demand forecasting	5000 SKUs 6 out of the 50 best-selling SKUs in the supermarket	Past sales with lag k Binary variables to characterize special days Price variables	MAPE, NMSE	ARIMA, Neural Networks	-	Neural networks outperformed ARIMA model's and the proposed additive hybrid approach gave best results
Measuring forecasting accuracy: The case of judgmental adjustments to SKU-level demand forecasts	How to measure the accuracy of adjustments of a statistical forecast of demand to take into account that are excluded from the statistical calculation	data collected from a company specialising in the manufacture of fast-moving consumer goods	-	MAPE, MASE, MdAPE,RMSPE,MRAE, MdRAE,GMRAE,RelGMSE Recommended: -AvgRelMAE	-	Limitations are referred to each of the evaluation methods presented on the paper	Use of percentage errors is not advisable because of the considerable proportion of low actual values; MAPE and MdAPE do not provide sufficient indication of the effectiveness of adjustments, in terms of a linear loss function; Median can also induce biases and should not be used as the arithmetic mean to average relative quantities; It is recommended that an average relative MAE be used which is calculated as the geometric mean of relative MAE values
Another look at measures of forecast accuracy	compare measures of accuracy of univariate time series forecasts	M3 series (472), Disney monthly log return, Sales of lubricant	-	MAPE, MdAPE, sMAPE, sMdAPE, MdRAE, GMRAE, MASE	Average Root Mean Squared Sales Prediction Error (ARMSE)	(1) historical mean using data up to the most recent observation; (2) the naive method based on the most recent observation; (3) simple exponential smoothing; (4) Holt's method	Intermittent demand data often contain zeros and many time series of interest to forecasters contain negative observations; Frequent occurrence of consecutive observations which take the same value
Accommodating heterogeneity and non linearity in price effects for predicting brand sales and profits	hierarchical Bayesian semi parametric approach to account simultaneously for heterogeneity and nonlinear flexibility in store sales models	weekly store-level scanner data for eight brands of orange juice	The lowest price of a competing national (premium) brand in stores s and week t and 11 covariates:. store size, education, family size, income, distances to nearest competing supermarkets or warehouses.	-	extended Flexible heterogeneous store sales model(EFHetM) and several other models (nested in EFHetM)	1 validation method	model accommodating both heterogeneity and fluctional flexibility outperforms competing models in predictive validity for substantial increases in expected profits for those brands

Figure A.1: Additional studies comparing forecasting approaches

An application of learning machines to sales forecasting under promotions	assess the relative effectiveness of three kinds of learning machines in sales forecasting and comparison with time series based methods, using the real data of a retail store. Capturing the effects of promotion policies on the sales	2 input-output time series(from store #1 and #2) of the daily sales receipts of a particular kind of pasta during a 3-years period	- sub-sequences of 13 attributes: 9 calendar attributes, 4 products specific attributes	MAPE	Multilayer artificial neural networks; Neural networks of radial basis functions; Support vector machines	-	machine learning methods, particularly support vector machines, performed particularly well during promotion periods
Analysis and forecasting of demand during promotions for perishable items		Dutch grocery retail chain. sales data are also weekly data (from week 2 in 2010 till week 35 in 2011). The data were weekly SKU-data from a supermarket chain. 407 different perishable items	data on sales quantities, prices, product information, weight or volume per consumer unit, timing of promotions, additional promotion action, and information whether the product was displayed in-store or included in the store flyer.				
Performance of state space and ARIMA models for consumer retail sales forecasting	This work compares the forecasting performance of state space models and ARIMA models. The forecasting performance is demonstrated through a case study of retail sales of five different categories of women footwear: Boots, Booties, Flats, Sandals and Shoes	monthly sales of the five categories of women footwear of the brand Forever Young: Boots, Booties, Flats, Sandals and Shoes, from January 2007 to April 2012 (64 observations)	-	ME, MAE, RMSE, MPE, MAPE	state space and ARIMA	compare the performance of this 'dummy model' with two alternative models: the 'linear model' and the 'quadratic model'	MAPE is having huge values when data contain very small numbers, as is the case of footwear models for the Boots, Booties and Sandals retail series are explained by the fact that during the out-of-sample period there are some months with almost no sales (close to zero).	both ETS and ARIMA models have the capability to forecast the future behavior of seasonal fluctuations fairly well. As expected, the exceptional increase in the sales of flats observed in March and April 2012 was not predicted by both models which underforecasted this situation
Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion	decompose demand to the non-promotional (baseline) and promotional demand and propose a hybrid model that uses a piecewise regression model and a time series model to forecast the promotional and baseline demand, find the most suitable forecasting model for demand series with different levels of volatility.	data from a food manufacturing company producing hundreds of FMCG in Australia. 112 weeks: 843 stock keeping unit locations (SKUs).	price of all products	coefficients of variation (CoV), MASE	ARIMAX, ETSX, DLR, ANN, and SVR. hybrid regression time series model, called HR-ARIMA	-	volatility significantly impacts ETSX model has a poor performance and is not recommended to use for forecasting volatile time series during promotions. ARIMAX has a superior performance for low and moderate volatile products DLR seems to be promising and SVR generates robust forecasts across different values of CoV. ANN generates better results for a low volatile.
Predictive Analytics for Demand Forecasting – A Comparison of SARIMA and LSTM in Retail SCM	provide an overview of approaches in retail SCM, emphasizes on perishable products and more precisely on fruits and vegetables and compare the quality of SARIMA and LSTM	four perishable products sold in more than 90 stores of a selected region of a leading retailer in Austria. The sale levels of the most purchased vegetables in the stores, namely potato, salad and cucumber. Three years from January 2017 to December 2019	Promotions	MAPE, RMSE	SARIMAX, LSTM		only focusses on the vegetable products of one retailer in Austria -used forecasting methods for selected items of product groups. This simplification could underrate the importance of cannibalization effects, as well as the effect of substitution -compared the forecasting results with the actual sales	Considering promotions as an external factor has increased the accuracy of prediction in almost all models. LSTM performed better for products that have more stable historical data (e.g., produce more reliable results for products with seasonally repeating behavior (e.g., cucumber).

Figure A.2: Additional studies comparing forecasting approaches

A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting	-	daily sales data of banana measured in kilograms from a major discount supermarket chain in Bavaria (Germany)	day of the week and day of the month, holiday, promotion, discount and weather	MAPE, RMSE	SARIMAX using multiple linear regression (SARIMA-MLR), hybrid SARIMA and Quantile Regression (SARIMA-QR)	-	Both of these models yield better predictions for out-sample data, compared to seasonal naive forecasting, SARIMA, and MLPNN models.
Demand forecasting in food retail: a comparison between the Holt-Winters and ARIMA models	performances between ARIMA and Holt-Winters (HW) models for the application of a time series formed by a group of perishable dairy products	dairy products with short life cycle, in the period from 2005 to 2013, composed of 50 SKUS	-	(MAPE) and the Theil inequality index (U-Theil)	ARIMA and Holt-Winters (HW) models	-	HW model obtained better results comparing to ARIMA model, according to MAPE and U-Theil
The impact of linear optimization on promotion planning	study the post-promotion dip effect by analyzing supermarket data	117 weeks of sales data from a supermarket for several brands of ground coffee. We divide our data in a training set of 82 weeks and a testing set of 35 weeks	-	MAPE, R^2, Revenue bias	log–log demand model	-	-
SKU demand forecasting in the presence of promotions	identify SKU sales forecasting methods in the presence of promotions	Subset of a single category that consists of four subcategories in four stores, amounting to 168 store-SKU combinations. 11936 store-SKU-week observations in total, spans a period of 76 weeks. (where store 1 carries 41 SKUs, store 2 carries 38 SKUs, store 3 carries 43 SKUs and store 4 carries 46 SKUs)	weekly sales with promotion and with no promotion, number of promotions, regular price, price cut (absolute and relative)	MAE	exponential smoothing, SVR method with three different kernel functions, regression tree, Stepwise linear regression	articular category data from four stores of a European grocery retailer	-
The derived demand for a advertising expenses and implications on sustainability, a comparative study using deep learning and traditional machine learning methods	use of machine learning techniques - including support vector regression, random forest regression, decision tree regressor, artificial neural network, and long short-term memory, for forecasting demand in marketing based on advertising expenses.	daily information from a television - broadcasting company, 17 individual programs with features, such as demand, sales, advertising expenses, and gross rating points, and has 2614 daily observations from 2010 to 2017	-	RMSE and MAE	Five machine learning (ML) models, including support vector regression (SVR), decision tree regressor (DTR), random forest regressor (RFR), artificial neural network (ANN), and long short-term memory (LSTM)	Limitations of the study include the use of a dataset from a single TV station and a focus on ML techniques for demand forecasting, and the lack of comparison with traditional statistical methods.	the study demonstrated the effectiveness of using neural network models, particularly LSTM, for demand forecasting in the context of advertising expenses
A big data driven framework for demand-driven forecasting with effects of marketing-mix variables	improve the accuracy of demand forecasts by considering various factors such as historical demand, advertising effectiveness, and marketing events.	daily TV sales and demand over a period of 372 weeks from 2010 to 2017. It includes information on sales, prices, supply chain, and marketing effectiveness	17 variables	MSE, MAPE and MAD	four-step process for predicting future demand and sales using a fuzzy artificial neural network (FANN) machine learning classifier, ARIMA, MLP, Random Forest, SVM, and ANN	-	The proposed model is able to efficiently analyze large data sets and predict future demand based on advertising expenditure, communication, promotion, and sales data.

Figure A.3: Additional studies comparing forecasting approaches

References

[1] Robert Fildes, Shaohui Ma, and Stephan Kolassa. Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4):1283–1318, 2022.

[2] Antonio Rafael Sabino Parmezan, Vinicius MA Souza, and Gustavo EAPA Batista. Evaluation of statistical and machine learning models for time series prediction: Identifying the state-of-the-art and the best conditions for the use of each model. *Information sciences*, 484:302–337, 2019.

[3] Ardalan Sameti and Hamidreza Khalili. Influence of in-store and out-of-store creative advertising strategies on consumer attitude and purchase intention. *Intangible Capital*, 13(3):523–547, 2017.

[4] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.

[5] Kusum Ailawadi, Bari Harlam, Jacques César, and David Trounce. Promotion profitability for a retailer: The role of promotion, brand, category, and store characteristics. *Journal of Marketing Research - J MARKET RES-CHICAGO*, 43:518–535, 11 2006. doi:10.1509/jmkr.43.4.518.

[6] Mahdi Abolghasemi, Eric Beh, Garth Tarr, and Richard Gerlach. Demand forecasting in supply chain: The impact of demand volatility in the presence of promotion. *Computers & industrial engineering*, 142:106380, 2020.

[7] Douglas C Montgomery, Cheryl L Jennings, and Murat Kulahci. *Introduction to time series analysis and forecasting*. John Wiley & Sons, 2015.

[8] Leonie Monigatti. Time series problems, simply explained as fast food combo meals, 2022.

[9] Suad A Alasadi and Wesam S Bhaya. Review of data preprocessing techniques in data mining. *Journal of Engineering and Applied Sciences*, 12(16):4102–4107, 2017.

[10] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[11] Patrícia Ramos, Nicolau Santos, and Rui Rebelo. Performance of state space and arima models for consumer retail sales forecasting. *Robotics and computer-integrated manufacturing*, 34:151–163, 2015.

[12] Claudimar Pereira Da Veiga, Cássia Rita Pereira Da Veiga, Anderson Catapan, Ubiratã Tortato, and Wesley Vieira Da Silva. Demand forecasting in food retail: A comparison between the holt-winters and arima models. *WSEAS transactions on business and economics*, 11(1):608–614, 2014.

[13] Ammar Aamer, LuhPutu Eka Yani, and IMade Alan Priyatna. Data analytics in the supply chain management: Review of machine learning applications in demand forecasting. *Operations and Supply Chain Management: An International Journal*, 14(1):1–13, 2020.

[14] Paulo Angelo Alves Resende and André Costa Drummond. A survey of random forest based methods for intrusion detection systems. *ACM Computing Surveys (CSUR)*, 51(3):1–36, 2018.

[15] Jehad Ali, Rehanullah Khan, Nasir Ahmad, and Imran Maqsood. Random forests and decision trees. *International Journal of Computer Science Issues(IJCSI)*, 9, 09 2012.

[16] Mohamad Khalil, A Stephen McGough, Zoya Pourmirza, Mehdi Pazhoohesh, and Sara Walker. Machine learning, deep learning and statistical analysis for forecasting building energy consumption—a systematic review. *Engineering Applications of Artificial Intelligence*, 115:105287, 2022.

[17] Philipp Probst, Marvin N Wright, and Anne-Laure Boulesteix. Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: data mining and knowledge discovery*, 9(3):e1301, 2019.

[18] Iliana Paliari, Aikaterini Karanikola, and Sotiris Kotsiantis. A comparison of the optimized lstm, xgboost and arima in time series forecasting. In *2021 12th International Conference on Information, Intelligence, Systems & Applications (IISA)*, pages 1–7. IEEE, 2021.

[19] Junling Luo, Zhongliang Zhang, Yao Fu, and Feng Rao. Time series prediction of covid-19 transmission in america using lstm and xgboost algorithms. *Results in Physics*, 27:104462, 2021.

[20] Ervin Ceperic, Vladimir Ceperic, and Adrijan Baric. A strategy for short-term load forecasting by support vector regression machines. *IEEE Transactions on Power Systems*, 28(4):4356–4364, 2013.

[21] Ajay Shrestha and Ausif Mahmood. Review of deep learning algorithms and architectures. *IEEE access*, 7:53040–53065, 2019.

[22] Suresh Dara and Priyanka Tumma. Feature extraction by using deep learning: A survey. In *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, pages 1795–1801. IEEE, 2018.

[23] S Velliangiri, S Alagumuthukrishnan, et al. A review of dimensionality reduction techniques for efficient computation. *Procedia Computer Science*, 165:104–111, 2019.

[24] David Salinas, Valentin Flunkert, Jan Gasthaus, and Tim Januschowski. Deepar: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, 36(3):1181–1191, 2020.

[25] Alex Sherstinsky. Fundamentals of recurrent neural network (rnn) and long short-term memory (lstm) network. *Physica D: Nonlinear Phenomena*, 404:132306, 2020.

[26] Rob J Hyndman and Yeasmin Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of statistical software*, 27:1–22, 2008.

[27] Ralph D Snyder, J Keith Ord, and Adrian Beaumont. Forecasting the intermittent demand for slow-moving inventories: A modelling approach. *International Journal of Forecasting*, 28(2):485–496, 2012.

[28] Lars-Erik Öller, Pär Stockhammar, et al. Rob j. hyndman, anne b. koehler, j. keith ord and ralph snyder, forecasting with exponential smoothing: The state space approach, springer (2008) 359 pp, isbn 978-3-540-71916-0 (paperback),[euro] 36.95, e-isbn 978-3-540-71918-2 (online),[euro] 25/chapter. *International Journal of Forecasting*, 26(1):204–205, 2010.

[29] Bryan Lim, Sercan Ö Arık, Nicolas Loeff, and Tomas Pfister. Temporal fusion transformers for interpretable multi-horizon time series forecasting. *International Journal of Forecasting*, 37(4):1748–1764, 2021.

[30] Shiyang Li, Xiaoyong Jin, Yao Xuan, Xiyou Zhou, Wenhu Chen, Yu-Xiang Wang, and Xifeng Yan. Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. *Advances in neural information processing systems*, 32, 2019.

[31] Boris N Oreshkin, Dmitri Carpov, Nicolas Chapados, and Yoshua Bengio. N-beats: Neural basis expansion analysis for interpretable time series forecasting. *arXiv preprint arXiv:1905.10437*, 2019.

[32] Thomas Rojat, Raphaël Puget, David Filliat, Javier Del Ser, Rodolphe Gelin, and Natalia Díaz-Rodríguez. Explainable artificial intelligence (xai) on timeseries data: A survey. *arXiv preprint arXiv:2104.00950*, 2021.

[33] Lloyd S Shapley et al. A value for n-person games. 1953.

[34] Andreas Messalas, Yiannis Kanellopoulos, and Christos Makris. Model-agnostic interpretability with shapley values. In *2019 10th International Conference on Information, Intelligence, Systems and Applications (IISA)*, pages 1–7. IEEE, 2019.

[35] Nari Sivanandam Arunraj and Diane Ahrens. A hybrid seasonal autoregressive integrated moving average and quantile regression for daily food sales forecasting. *International Journal of Production Economics*, 170:321–335, 2015.

[36] Nari Sivanandam Arunraj, Diane Ahrens, and Michael Fernandes. Application of sarimax model to forecast daily sales in food retail industry. *International Journal of Operations Research and Information Systems (IJORIS)*, 7(2):1–21, 2016.

[37] Maxim Vladimirovich Shcherbakov, Adriaan Brebels, Nataliya Lvovna Shcherbakova, Anton Pavlovich Tyukov, Timur Alexandrovich Janovsky, Valeriy Anatol'evich Kamaev, et al. A survey of forecast error measures. *World applied sciences journal*, 24(24):171–176, 2013.

[38] Rob J Hyndman and Anne B Koehler. Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4):679–688, 2006.

[39] Mahdi Abolghasemi, Jason Hurley, Ali Eshragh, and Behnam Fahimnia. Demand forecasting in the presence of systematic events: Cases in capturing sales promotions. *International Journal of Production Economics*, 230:107892, 2020.

[40] Juan R Trapero, Diego J Pedregal, Robert Fildes, and Nikolaos Kourentzes. Analysis of judgmental adjustments in the presence of promotions. *International Journal of Forecasting*, 29(2):234–243, 2013.

[41] Maxime C Cohen, Ngai-Hang Zachary Leung, Kiran Panchamgam, Georgia Perakis, and Anthony Smith. The impact of linear optimization on promotion planning. *Operations Research*, 65(2):446–468, 2017.

[42] Karel H Van Donselaar, Jordi Peters, Ad de Jong, and Rob ACM Broekmeulen. Analysis and forecasting of demand during promotions for perishable items. *International Journal of Production Economics*, 172:65–75, 2016.

[43] Sunil Gupta and Lee G Cooper. The discounting of discounts and promotion thresholds. *Journal of consumer research*, 19(3):401–411, 1992.

[44] Taha Falatouri, Farzaneh Darbanian, Patrick Brandtner, and Chibuzor Udokwu. Predictive analytics for demand forecasting–a comparison of sarima and lstm in retail scm. *Procedia Computer Science*, 200:993–1003, 2022.

[45] Özden Gür Ali, Serpil Sayın, Tom Van Woensel, and Jan Fransoo. Sku demand forecasting in the presence of promotions. *Expert Systems with Applications*, 36(10):12340–12348, 2009.

[46] Sule Birim, Ipek Kazancoglu, Sachin Kumar Mangla, Aysun Kahraman, and Yigit Kazancoglu. The derived demand for advertising expenses and implications on sustainability: a comparative study using deep learning and traditional machine learning methods. *Annals of Operations Research*, pages 1–31, 2022.

[47] Gianni Di Pillo, Vittorio Latorre, Stefano Lucidi, Enrico Procacci, et al. An application of learning machines to sales forecasting under promotions. *Control and Management Engineering*, 2013.

[48] Ajay Kumar, Ravi Shankar, and Naif Radi Aljohani. A big data driven framework for demand-driven forecasting with effects of marketing-mix variables. *Industrial marketing management*, 90:493–507, 2020.