



Gestão de Inventários de uma cadeia de retalho

JORGE RICARDO CORDEIRO DE OLIVEIRA MONTEIRO ALVES

outubro de 2023

Previsão da Procura em Pontos de Venda

Jorge Monteiro Alves

Estudante n.º <1210836>

2023

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Mecânica

Orientador da dissertação: Professor Manuel Pereira Lopes

isen

P.PORTO

PREVISÃO DA PROCURA EM PONTOS DE VENDA

Jorge Monteiro Alves

Estudante n.º <1210836>

Dissertação apresentada ao Instituto Superior de Engenharia do Porto para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia e Gestão Industrial, realizada sob a orientação do Doutor Manuel Pereira Lopes.

2023

Instituto Superior de Engenharia do Porto

Departamento de Engenharia Mecânica

isen

P.PORTO

AGRADECIMENTOS

Um obrigado à empresa pela disponibilização dos dados e pelo apoio na realização deste trabalho.

A todos os Professores que partilharam comigo o seu conhecimento ao longo destes mais de 19 anos estudo, com mais paciência do que a que mereci, o meu muito obrigado.

Um agradecimento especial ao Professor Manuel Pereira Lopes, por todo o apoio prestado neste projeto. Estou no mestrado devido à sua orientação.

Ao meu pai, obrigado por todos os conselhos. À minha mãe, obrigado por todo o apoio e carinho.

À Patrícia, uma página é pouco para agradecer. Sem ti não teria sido possível.

Por fim, o meu obrigado aos meus irmãos e aos meus amigos, que foram essenciais nos momentos de descontração.

página propositadamente em branco

RESUMO

O presente trabalho aborda a criação de um modelo de previsão de vendas semanais por artigo / localização com base em dados reais de uma empresa de retalho líder no setor de cosmética.

O modelo atualmente utilizado pela empresa apresenta indícios de não captar corretamente as características da procura dos seus clientes, o que gera ineficiências dentro da cadeia de abastecimento por excesso de alocação de stock a um determinado ponto, assim como falhas de stock e consequentes perdas de venda em alturas sazonais ou em que existem eventos especiais.

Foram aplicados métodos clássicos de previsão com base em séries temporais, como o método de Amortecimento Exponencial Simples, o Método de Holt, Holt-Winters, e o método ARIMA. Adicionalmente, foi também aplicado um algoritmo de multilayer perceptron (MLP) que utiliza variáveis de eventos culturais, nacionais, desportivos, e meteorológicos, além de informação sobre a família e subfamília do artigo. Para estimar os valores da sazonalidade, da tendência e os resíduos de cada série temporal, é utilizado o Amortecimento Exponencial Simples para prever os pontos do horizonte de previsão e aplicado o método STL com sazonalidade a um ano para extrair os valores estimados, que são posteriormente aplicados na previsão de vendas gerada pelo modelo MLP.

Os modelos foram testados utilizando um método de Janela Deslizante ao longo de um ano, em que a cada ponto se procurava prever os dois pontos futuros, utilizando o RMSE médio de cada modelo como avaliação. O modelo MLP conseguiu o melhor desempenho em 16 dos 19 artigos utilizados para teste, o que demonstra a potencialidade da utilização de algoritmos de Machine Learning para prever a procura de uma cadeia de retalho.

PALAVRAS-CHAVE

Previsão, Retalho, Métodos Estatísticos, Machine Learning

página propositadamente em branco

ABSTRACT

This study addresses the development of a sales forecasting model for weekly sales by item/location based on real data from a leading retail company in the cosmetics sector.

The model currently used by the company shows signs of not accurately capturing the demand characteristics within the supply chain, leading to inefficiencies due to overstocking at specific points, as well as stock shortages and subsequent sales losses during seasonal peaks or special events.

Classic forecasting methods based on time series were applied, such as the Simple Exponential Smoothing method, the Holt Method, Holt-Winters, and the ARIMA method. Additionally, a multilayer perceptron (MLP) algorithm was employed, incorporating variables from cultural, national, sports, and weather events, along with information about the item's family and subfamily. To estimate the seasonality, trend, and residuals of each time series, the Simple Exponential Smoothing method is used to predict the forecast horizon points, and the STL method with a one-year seasonality is applied to extract the estimated values, which are then utilized in the sales forecast generated by the MLP model.

The models were tested using a Rolling Window method over a year, where each point aimed to predict the next two future points, using the average RMSE of each model as an evaluation metric. The MLP model achieved the best performance on 16 out of the 19 items tested, demonstrating the potential of using Machine Learning algorithms to forecast the demand in a retail chain.

KEYWORDS

Forecasting, Retail, Statistical Methods, Machine Learning

página propositadamente em branco

ÍNDICE

Índice de Figuras	VII
Índice de Tabelas.....	IX
LISTAS DE SIGLAS E SÍMBOLOS.....	XI
1. Introdução.....	13
1.1. Descrição do problema	13
1.2. Questão da Investigação	13
1.3. Opções Metodológicas.....	13
1.4. Descrição do negócio	14
1.5. Estrutura do trabalho.....	15
1.6. Contribuições	15
2. Revisão Bibliográfica	17
2.1. Gestão de Inventários	17
2.2. Previsão	19
2.2.1. Métodos Estatísticos	19
2.2.2. Previsão com Machine Learning	21
3. Métodos e Aplicação.....	24
3.1. Estado Atual	24
3.2. Seleção dos dados.....	30
3.2.1. Horizonte temporal.....	30
3.2.2. Seleção da loja.....	30
3.2.3. Seleção dos artigos.....	31
3.2.4. Granularidade dos dados	33
3.3. Análise dos dados.....	34
3.3.1. Identificação dos componentes	35
3.3.2. Tratamento de Outliers.....	51
3.4. Seleção dos modelos de previsão	54
3.4.1. Métodos para Benchmark.....	54
3.4.2. Alisamento Exponencial	55
3.4.3. Método ARIMA.....	56
3.4.4. MultiLayer Perceptron	56
4. Avaliação de resultados	61
5. Conclusões.....	67
5.1. Trabalho Futuro.....	67
6. Bibliografia	69
APÊNDICE A.....	72

página propositadamente em branco

ÍNDICE DE FIGURAS

Figura 1 – Componentes de uma série temporal.....	20
Figura 2 – Análise ABC de 30 dias de vendas (em valor).....	25
Figura 3 – Análise ABC de 30 dias de vendas (em quantidade)	26
Figura 4 – Histórico de vendas e stock do artigo X1100 na loja 036.....	27
Figura 5 – Histórico de vendas e stock do artigo X1200 na loja 027.....	28
Figura 6 – Histórico de vendas e stock do artigo X1200 na loja 027.....	29
Figura 7 – Faturação normalizada em 2022 das lojas físicas VS loja 027.....	31
Figura 8 – Vendas semanais do artigo X0001	33
Figura 9 – Vendas diárias do artigo X0001.....	34
Figura 10 – Venda de 156 semanas do artigo X00001	35
Figura 11 – Autocorrelação X00001	35
Figura 12 – Anomalias X00001	36
Figura 13 – Venda de 156 semanas do artigo X00002	36
Figura 14 – Autocorrelação X00002	36
Figura 15 – Anomalias X00002	37
Figura 16 – Venda de 156 semanas do artigo X00003	37
Figura 17 – Autocorrelação X00003	37
Figura 18 – Anomalias X00003	37
Figura 19 – Venda de 156 semanas do artigo X00003	38
Figura 20 – Autocorrelação X00004	38
Figura 21 – Anomalias X00004.....	38
Figura 22 – Venda de 156 semanas do artigo X00005	39
Figura 23 – Autocorrelação X00005	39
Figura 24 – Anomalias X00005	39
Figura 25 – Venda de 156 semanas de artigos com características similares	40
Figura 26 – Autocorrelação de artigos com características similares.....	41
Figura 27 – Anomalias de artigos com características similares.....	42
Figura 28 – Venda de 156 semanas do artigo X00012	42
Figura 29 – Autocorrelação X00012	43
Figura 30 – Anomalias X00012	43
Figura 31 – Venda de 156 semanas do artigo X00013	43
Figura 32 – Autocorrelação X00013	44
Figura 33 – Anomalias X00013	44
Figura 34 – Venda de 156 semanas do artigo X00014	44
Figura 35 – Autocorrelação X00014	45
Figura 36 – Análise de anomalias X00014.....	45
Figura 37 – Venda de 156 semanas do artigo X00015	45
Figura 38 – Autocorrelação X00015	46
Figura 39 – Anomalias X00015	46
Figura 40 – Venda de 156 semanas do artigo X00016	46
Figura 41 – Autocorrelação X00016	47
Figura 42 - Anomalias X00016.....	47
Figura 43 – Venda de 156 semanas do artigo X00017	47
Figura 44 – Autocorrelação X00017	48
Figura 45 – Autocorrelação X00017	48

Figura 46 – Venda de 156 semanas do artigo X00018	48
Figura 47 – Autocorrelação X00018	49
Figura 48 – Autocorrelação X00018	49
Figura 49 – Venda de 156 semanas do artigo X00019	49
Figura 50 – Autocorrelação X00019	50
Figura 51 – Autocorrelação X00019	50
Figura 52 – Venda de 156 semanas do artigo X00020	50
Figura 53 – Autocorrelação X00020	51
Figura 54 – Autocorrelação X00020	51
Figura 55 – Efeito de campanha promocional na procura do artigo X00001	52
Figura 56 – rotura prolongada do artigo X00013	52
Figura 57 – Efeito da correção de roturas no artigo X00013	53
Figura 58 – Efeito da correção de roturas no artigo X00013	54
Figura 59 – Rede Neural de um algoritmo MultiLayer Perceptron	57
Figura 60 – Aplicação do método STL ao artigo X00003	59
Figura 61 – Evolução do erro de treino e de validação	61
Figura 62 – Representação de uma Janela Deslizante	62
Figura 63 – Aplicação do método Atual ao artigo X00012	64
Figura 64 – Real vs Previsto pelo modelo MLP para o artigo X00003	65
Figura 65 – Real vs Previsto pelo modelo MLP para o artigo X00018	65

ÍNDICE DE TABELAS

Tabela 1 – Resultados da análise ABC (em valor)	25
Tabela 2 – Resultados da análise ABC (em quantidade)	26
Tabela 3 – Nível de Serviço do artigo	26
Tabela 4 – Peso de venda por família das lojas físicas VS loja 027 no ano de 2022	31
Tabela 5 – Artigos Selecionados.....	32
Tabela 6 – Tempo de computação em segundos com diferente número de amostras	61
Tabela 7 – Resultados obtidos pelos modelos de previsão	63
Tabela 8 – Iterações do modelo	66

página propositadamente em branco

LISTAS DE SIGLAS E SÍMBOLOS

Lista de Siglas

ISEP	Instituto Superior de Engenharia do Porto
P.Porto	Instituto Politécnico do Porto
ReLU	Unidade Linear Retificada
FMCG	Fast Moving Consumer Goods
SKU	Stock Keeping Unit
KPI	Key Performance Indicator
STL	Seasonal and Trend decomposition using Loess
MLP	MultiLayer Perceptron
RMSE	Root Mean Square Error

página propositadamente em branco

1. Introdução

Neste capítulo introdutório, será apresentado o contexto deste projeto. É feita uma breve caracterização da empresa, seguida da descrição do problema de gestão de inventários que é abordado. No final do capítulo encontra-se a questão de investigação e os objetivos a que este projeto se propõe.

1.1. Descrição do problema

Determinar a quantidade de stock a enviar para um ponto de venda é uma decisão crucial para o sucesso de uma empresa de retalho. Enviar quantidade a menos significa perder oportunidade de venda e limitar o crescimento da empresa, enquanto o envio de quantidade excessiva pode representar custos de armazenagem, capital parado e até perda total de parte do produto. Para decidir a quantidade a enviar de um produto para um determinado ponto de venda, é necessário prever a procura desse produto num determinado horizonte temporal.

Este projeto irá debruçar-se sobre o problema de previsão da procura de um produto num ponto de venda de uma empresa de retalho de produtos cosméticos. Para isso, foram implementados diversos métodos de previsão e o seu desempenho foi comparado com o modelo em uso pela empresa.

1.2. Questão da Investigação

Com base no problema apresentado no ponto anterior, este projeto tem o objetivo de responder às seguintes perguntas:

- É possível desenvolver um novo algoritmo de previsão, com melhor desempenho, e com ganhos mensuráveis para a empresa?
- Os métodos de previsão com *Machine Learning* têm melhor desempenho que os métodos estatísticos?

Para responder a estas questões, os objetivos a cumprir são os seguintes:

- Identificar as limitações do algoritmo atual;
- Implementar algoritmos de previsão com métodos estatísticos;
- Implementar um algoritmo de *Machine Learning* para previsão;
- Comparar o desempenho dos diferentes algoritmos.

1.3. Opções Metodológicas

Na literatura, a metodologia é dividida pela maioria dos autores em duas perspetivas:

- A investigação qualitativa, em que os dados recolhidos “estão sob a forma de palavras, imagens ou som e nunca em forma de números” (Costa et al., 2021);

- A investigação quantitativa, que visa explicar e controlar certos fenómenos, “através da objetividade dos procedimentos e da quantificação das medidas” (Costa et al., 2021)

Sendo o objetivo deste projeto comparar os resultados de diferentes algoritmos e mensurar os ganhos da sua implementação, enquadra-se na metodologia quantitativa.

Em relação à abordagem de investigação, pode apresentar uma abordagem indutiva ou dedutiva. A abordagem indutiva é guiada pelos dados, ou seja, os temas identificados são associados aos próprios dados. Na abordagem dedutiva, ou teórica, “a análise parte de um conjunto de categorias ou temas predeterminados” (Costa et al., 2021).

Walliman (2011) apresenta a abordagem hipotética-dedutiva como a combinação das duas abordagens, em que selecionamos a abordagem através do pensamento indutivo, formulámos uma hipótese com base no pensamento dedutivo, e testamos empiricamente os resultados, sendo que, se não obtivermos os resultados esperados, voltamos a ajustar a teoria. Esta será a abordagem deste projeto (Walliman, 2011)..

A natureza da investigação pode tomar várias formas, sendo elas caracterizadas por Oliveira (2011) como descritiva, exploratória, analítica, sintética e atuante (Oliveira, 2011). Tendo este projeto o objetivo de construir algo adaptável que possa ser utilizado em contexto real, terá, necessariamente, que ter como base um entendimento profundo do problema, capaz não só de apresentar uma solução quantificável como de explicar o seu funcionamento. Podemos por isso concluir que este projeto tem uma base de investigação analítica.

1.4. Descrição do negócio

A empresa em estudo é uma empresa portuguesa de comércio ao retalho de produtos cosméticos. Tem mais de 100 lojas no continente, sendo a loja mais a Norte em Bragança e a mais a Sul no Algarve, além de presença na Madeira. O seu volume de negócios no ano transato foi de 52 milhões, e a meta de crescimento para este ano é de 25%, com planos de chegar às 114 lojas.

As vendas são divididas em dois segmentos, clientes “público” (qualquer cliente) e clientes “profissionais” – os clientes profissionais têm de apresentar prova de serem profissionais da cosmética (curso de cabeleireiro, unhas de gel, esteticista, etc.) para acederem a uma tarifa especial, com descontos que podem ser superiores a 50% em alguns produtos. De momento, as vendas dividem-se em 53% para profissionais e 47% para público, sendo que, devido aos descontos, a quantidade de produtos comprados por profissionais é bastante superior.

O reabastecimento das lojas é feito por um armazém na Maia gerido pela empresa, onde fica também a sua sede. A mercadoria para as lojas é separada de segunda a sexta e a lógica de separação é gerida através do cálculo de stock mínimo e máximo (a loja ao atingir o stock mínimo de um produto, o armazém separa para o stock máximo desse produto para essa loja). Os parâmetros de stock mínimo e máximo são recalculados diariamente com base nas vendas dos últimos 30 dias. O Lead time do armazém às lojas é de um dia útil, a não ser no caso das lojas do Algarve, onde o lead time é de dois dias.

Introdução

A lógica do negócio é uma mistura do mercado de fashion com o mercado FMCG. Existem componentes de sazonalidade em algumas gamas (solares, coffrets, etc.) e são constantemente feitas campanhas, sendo que a informação se a venda teve uma campanha é registada na base de dados no momento da transação.

1.5. Estrutura do trabalho

Este trabalho é dividido em cinco capítulos. O primeiro capítulo, introdução, apresenta a descrição do problema, da investigação e as opções metodológica, assim como uma breve descrição da empresa em estudo.

O segundo capítulo contém a revisão bibliográfica, onde é feito um levantamento de outros trabalhos académicos que trataram problemas relacionados com previsão e gestão de inventário.

O terceiro capítulo descreve a seleção e análise dos dados em estudo, seguido de uma descrição dos métodos utilizados para previsão.

No quarto capítulo são apresentados e discutidos os resultados obtidos, fazendo uma comparação quantitativa dos métodos utilizados.

O quinto capítulo apresenta as conclusões que foram obtidas com o trabalho desenvolvido, sendo também apresentados os trabalhos a serem desenvolvidos.

1.6. Contribuições

A parte prática deste trabalho resultou no desenvolvimento de uma aplicação em *Python*, cujo código pode ser consultado no apêndice. Para a criação da aplicação, foram essenciais o uso de diversas bibliotecas. A seguir, apresenta-se uma breve descrição das principais bibliotecas utilizadas.

A biblioteca *StatsForecast* proporciona várias ferramentas para análise estatística e previsão de séries temporais. A sua capacidade de processar grandes volumes de dados em tempo real facilitou o desenvolvimento do projeto.

A biblioteca *StatsModels* oferece uma ampla variedade de métodos estatísticos que foram cruciais para a criação do modelo e validação dos resultados obtidos.

A biblioteca *Keras* é uma das bibliotecas mais usadas em *Python* para a criação de algoritmos de *Deep Learning*, tendo sido usada para a implementação do modelo de *machine learning Multilayer Perceptron*. A utilização desta biblioteca permitiu a criação de um modelo mais complexo e otimizado.

No capítulo 3.1., é apresentado o estado atual utilizado atualmente pela empresa de retalho de produtos cosméticos, e os principais problemas do modelo são listados. No capítulo 3.2. é apresentada a forma de seleção e recolha dos dados de venda utilizados neste trabalho. O capítulo 3.4.4.1. expande este tema, apresentando a estrutura de dados selecionada para a aplicação do modelo de *machine learning* e como foi construída.

Para o desenvolvimento da aplicação em *Python*, foram aplicados diversos métodos de previsão de procura, além do modelo atualmente em uso pela empresa:

- Métodos Estatísticos:
 - Método Ingénuo;
 - Amortecimento Exponencial Simples;
 - Método de Holt;
 - Método de Holt-Winters;
 - ARIMA;
 - SARIMA;
- Método de *machine learning* MLP – *Multilayer Perceptron*.

Por fim, os resultados obtidos por estes modelos foram comparados utilizando o método de Janela Deslizante para obter o RMSE médio de cada modelo ao longo de um ano para cada um dos artigos selecionados, o que demonstrou que o modelo atualmente usado pela empresa não é adequado para a realidade da procura.

2. Revisão Bibliográfica

De forma a contextualizar o problema abordado nesta dissertação, este capítulo irá abordar a definição e os métodos de Gestão de Inventários, seguindo-se uma revisão sobre previsão estocástica. Inicialmente será feita uma apresentação sobre algoritmos estatísticos de previsão, passando para uma revisão dos trabalhos mais recentes, utilizando técnicas de machine learning.

2.1. Gestão de Inventários

A gestão de inventários é um fator crítico de sucesso para um retalhista. Tradicionalmente, o inventário representa uma parte muito significativa dos ativos de uma empresa de retalho, o que significa que o inventário é uma representação do capital monetário de qualquer empresa, pelo que pode ser usado como uma medida para confirmar a liquidez da empresa. Para além destes fatores, Agrawal et al. argumentam que o inventário é um fator crítico para o desempenho de um retalhista porque “um retalhista não pode vender o que não tem” (Agrawal & Smith, 2013). O inventário não é um recurso ilimitado, pelo que não basta definir o stock alvo de cada loja – é necessário criar estratégias de otimização para alocar o stock a cada ponto de venda (Agrawal & Smith, 2013). Além do custo do inventário, devem ser considerados os custos fixos e variáveis do reabastecimento de uma loja. Os custos fixos podem ser caracterizados pelo custo do espaço que o produto irá ocupar e custos de transporte e de mão de obra que sejam independentes da quantidade a alocar (Agrawal & Smith, 2013).

A implementação de uma estratégia de gestão de inventários é essencial na gestão de cadeia de abastecimento. Uma cadeia de abastecimento pode ser definida como uma rede de nós que têm como objetivo final satisfazer as necessidades dos clientes, em que os nós estão organizados em escalões. Estes nós interligam-se através de uma relação de procura e oferta. Um dos aspetos mais desafiantes nestes problemas é controlar o nível de inventário, através da determinação do tamanho da encomenda para cada escalão num determinado produto para otimizar uma função objetivo. (Zhou et al., 2013). Nirmala et al definem como principais objetivos da gestão de inventário o aumento do lucro da empresa através da melhoria da gestão de inventário, prever o impacto das estratégias da empresa nos níveis de inventários e minimizar o custo total das atividades logística sem comprometer o nível de serviço ao cliente final (Annie Rose Nirmala et al., 2022),.

O controlo do nível de inventário em cada ponto do nó tem repercussões positivas em todos os intervenientes da cadeia de abastecimento. A partilha de estimativas de procura entre cadeias de retalho, ou pontos de venda, e os distribuidores, resulta numa distribuição de stock mais eficaz em toda a cadeia de abastecimento. Um exemplo desta colaboração entre distribuidor e retalhista é a adoção de modelos VMI (Vendor Managed Inventory, ou Inventário Gerido pelo Fornecedor), em que o stock no ponto de venda é controlado pelo distribuidor. Este sistema permite ao retalhista ter uma oferta maior do fornecedor no espaço de venda, o que aumenta a rentabilidade da marca do distribuidor para ambas as partes (Achabal et al., 2000).

A maioria dos modelos de gestão de inventário das cadeias de retalho assume que todas as lojas na rede têm uma distribuição de probabilidade idêntica na procura, o que não corresponde

à realidade. Quando estas diferenças são exploradas, pode aumentar significativamente o lucro da operação (Agrawal & Smith, 2013). A rede de retalho Macy's reportou um aumento considerável na sua margem de lucro após uma iniciativa que visou personalizar a oferta de cada loja à procura do mercado em que a loja se enquadra (Wall Street Journal, 2010)

Um dos principais objetivos da gestão de inventários é evitar roturas de stock. A rotura gera uma perda de venda potencial e gera insatisfação nos clientes, que podem simplesmente trocar de loja. Para evitar roturas, os retalhistas aumentam o stock de segurança de um produto, o que reduz o lucro da operação (Huang et al., 2014). Para balancear o custo do stock de segurança e a perda gerada pela rotura de stock, é importante ter um algoritmo de previsão de vendas ao nível de um determinado artigo, numa determinada loja, num determinado momento. No entanto, a implementação destes métodos de previsão é difícil, devido à quantidade de fatores que influenciam a procura de um artigo, tal como a redução de preços e campanhas promocionais (Huang et al., 2014). Um problema recorrente nas cadeias de retalho é a criação de campanhas promocionais em determinados artigos ou grupos de artigos. Estes eventos não têm apenas impacto nos artigos em campanha no decorrer da mesma, como podem também afetar os períodos temporais adjacentes à campanha e outros artigos concorrentes vendidos pelo retalhista. Apesar de intuitivamente podermos definir que este impacto é negativo, observações empíricas são contraditórias neste ponto (Achabal et al., 1990).

Além do nível de roturas, existem vários KPI's (Key Performance Indicators) que podem ser utilizados para medir a gestão de inventários de uma cadeia de retalho, como as vendas perdidas e a rotação de Stock, que é habitualmente um dos KPI's mais utilizados pelos gestores para medir o desempenho da empresa. Contudo, Gaur et al. argumentam que a rotação de stock por si só é um mau indicador do desempenho do inventário, em grande parte pela variação que este indicador tem ao longo de um ano. No seu lugar, propõem um modelo mais completo, que tem em consideração a margem bruta, a intensidade do capital (ou seja, o rácio entre ativos imobilizados e ativos totais) e "vendas surpresa" (o rácio entre vendas reais contra vendas esperadas de um ano). Na sua análise, os autores apontam que a rotação de inventário anual tem uma correlação negativa com a margem bruta e uma correlação positiva com a intensidade do capital e "vendas surpresa" (Gaur et al., 2004). Os autores dão a seguinte hipótese para explicar a correlação negativa entre a rotação de inventário e a margem bruta:

- Nível de serviço: um aumento da margem bruta gera um aumento no nível de serviço, o que por sua vez gera um aumento no nível de stock do artigo e, consequentemente, na sua rotação;
- Preço: um aumento no preço do artigo gera um aumento na margem bruta do mesmo, contudo, a procura é geralmente negativamente correlacionada com o preço;
- Variedade de oferta: maior variedade de produtos leva a uma maior incerteza quanto ao preço praticado por parte do retalhista, o que leva a aumentar o preço médio (Gaur et al., 2004).

A gestão de inventários tem passado por várias inovações no passado recente. Graças à prevalência de scanners e de tecnologia EDI (Electronic Data Interchange, ou Troca de Dados Eletrónica) nos POS (Point Of Sale) atuais, a decisão de alocação de stock a uma loja é efetuada na informação de venda mais recente. Isto permite que a loja seja reabastecida diariamente com a informação de vendas do dia anterior, através do centro de distribuição mais próximo (Agrawal & Smith, 2013). A rapidez do tempo de resposta da cadeia de abastecimento, com os dados mais atuais e completos possíveis, é um fator crítico para o sucesso de qualquer cadeia de abastecimento de uma empresa de retalho.

2.2. Previsão

Tendo a procura de uma cadeia de retalho, por definição, um comportamento estocástico, a previsão assume um papel fundamental em todas as áreas da empresa. Na área de logística, é a base para a quantidade de stock a alocar nos centros de distribuição e nas lojas. As cadeias de retalho têm lojas de diferentes tipologias, que servem clientes de segmentos diferentes em locais diferentes. As vendas de uma determinada loja são impactadas pela localização, demografia dos consumidores, campanhas promocionais, meteorologia, altura do ano e eventos locais (Fildes et al., 2022).

A procura estimada de um produto numa loja continua a ser maioritariamente estimada com base no desempenho histórico numa série temporal. Contudo, alguns investigadores propõem que os modelos tradicionais são inadequados à realidade complexa das cadeias de retalho, identificando indícios de não linearidade e de volatilidade que modelos tradicionais baseados em séries temporais não conseguem tratar (Fildes et al., 2022). O uso de modelos com base em machine learning, que conseguem “aprender” padrões através da exploração de grandes fontes de dados, tem o potencial de revolucionar a previsão em retalho. Contudo, ainda não é claro que o desempenho destes modelos seja superior ao dos modelos tradicionais (Huber & Stuckenschmidt, 2020). Na quarta edição da M competition, a M4, vista por muitos académicos como a competição mais importante em previsão, a grande maioria dos modelos com melhor pontuação utilizaram uma combinação de métodos estatísticos e de Machine Learning, existindo apenas um modelo entre os 10 melhores que usava só métodos estatísticos. Entre os 10 piores, todos eram puramente estatísticos ou puramente Machine Learning (Makridakis et al., 2020).

2.2.1. Métodos Estatísticos

Existem várias abordagens clássicas que podem ser utilizadas na previsão (Rushton et al., 2014):

- Métodos de julgamento, em que a abordagem é subjetiva e baseada na opinião de peritos como fornecedores, marketeers, ou os próprios clientes;
- Métodos experimentais, que são utilizados quando não temos informação sobre o produto (como por exemplo num lançamento);
- Métodos causais, que são utilizados quando a procura de um produto é dependente de outros fatores, como promoções, preços, etc.;
- Métodos projetivos ou de séries temporais, em que usamos os dados passados para identificar quaisquer tendências na procura e projetá-la para o futuro

A aplicação de métodos de séries temporais implica ter disponíveis dados do passado recente, de onde se extrapola o futuro. Harvey et al. argumenta que a razão fundamental para construir um modelo de série temporal é determinar as propriedades de uma série temporal e retirar os seus componentes, como tendência, ciclicidade e sazonalidade, de onde podemos retirar uma interpretação direta (Harvey, 2006). Podemos observar as componentes mais comuns de uma série temporal na figura 1.

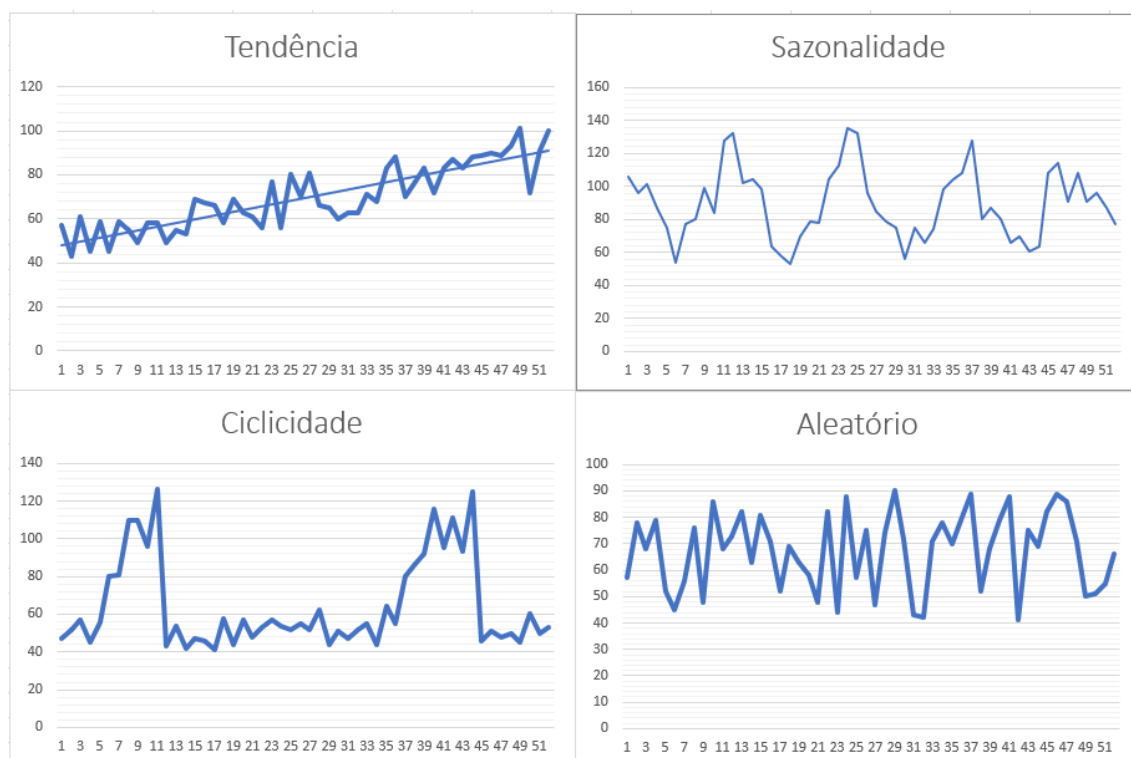


Figura 1 – Componentes de uma série temporal

Os métodos mais comuns de previsão através de séries temporais são a Média Móvel e o Alisamento Exponencial (Rushton et al., 2014). O uso da média móvel é de fácil aplicação – assume-se que o valor da previsão é igual à média do evento dentro de um determinado intervalo temporal anterior. O principal problema deste método é que ignora a idade da previsão, ou seja, atribui o mesmo peso a todos os pontos de dados dentro do intervalo da previsão, o que leva a que ignore as potenciais componentes da série temporal. O alisamento exponencial procura contrariar esse problema, ao introduzir a possibilidade de atribuir um peso aos diferentes períodos temporais (Rushton et al., 2014). Holt definiu que a aplicação deste modelo tem as seguintes propriedades desejáveis: deve ser atribuído menos peso aos dados mais antigos, é extremamente fácil de computar, e são necessários poucos dados para o aplicar (Holt, 2004).

Os métodos baseados na média móvel deparam-se com várias imprecisões devido aos diversos fatores que ditam a procura no mercado retalhista (Rushton et al., 2014). Devido a não ter em consideração as componentes das séries temporais, foram construídos modelos que procuram incorporar estas variáveis. O método de Holt, ou Alisamento Exponencial com Tendência, continua a ser um dos mais utilizados pela academia.

Holt expandiu o seu trabalho com o seu aluno Peter Winters de forma a expandir a usabilidade do novo método, adicionando uma componente para explorar a sazonalidade. Este novo método é denominado Modelo de Holt – Winters, com tendência Linear e Sazonalidade Multiplicativa ou Aditiva. As diferenças entre os métodos Multiplicativo e o Aditivo é a forma como são calculados o nível e a componente de sazonalidade, além de como a sazonalidade é considerada na previsão (Hyndman & Athanasopoulos, 2021).

O método ARIMA (Autoregressive Integrated Moving Average) é definido por Ho et al. como um método exploratório de dados que “ajusta” o modelo à própria estrutura de dados. Através de análises de correlação, a natureza estocástica dos dados pode ser aproximada, o que permite identificar as componentes da série temporal (Ho et al., 1998). Este modelo é uma derivação do modelo ARMA (Autoregressive-moving average), que utiliza uma auto-regressão e a média móvel para encontrar a previsão, desenvolvido por Box e Jenkins, que permite incorporar parâmetros para tratar a sazonalidade ou a tendência. Este modelo é composto por três fases: a de identificação, em que determinamos as características da série temporal e tentamos enquadrá-la no modelo, a de estimativa, em que estimamos os parâmetros usando os dados disponíveis, e diagnóstico, em que avaliamos o modelo produzido (Liu et al., 1992).

2.2.2. Previsão com Machine Learning

O termo *Machine Learning* é atribuído a Arthur Samuel, que desenvolveu em 1959 um algoritmo para ensinar uma máquina a jogar o jogo de damas. Programando apenas as regras do jogo, a direção do jogo, e parâmetros que o próprio autor define como redundantes e incompletos, a máquina ultrapassava o jogador humano com apenas 8 a 10 horas de treino. Segundo o investigador, “os princípios de *Machine Learning* verificados nestas experiências são, claramente, aplicados em muitas outras situações”(Samuel, 1967).

A grande quantidade de dados extraídos e armazenados pelas organizações tem crescido a um ritmo vertiginoso. Segundo Sagiroglu et al., até 2003 foram criados 5 exabytes (10^{18} bytes) de dados. Essa informação era armazenada em dois dias em 2013 (Sagiroglu & Sinanc, 2013). Estes dados são tão abrangentes que seria impossível para o ser humano aproveitar todo o seu potencial, pelo que *Machine Learning* é usado para extrair conhecimento valioso desses dados (Mahesh, 2018). *Machine Learning* tem vindo a ser aplicado em diferentes áreas devido à sua capacidade de caracterizar relações dentro de grandes matrizes de dados que resolvem problemas como análise em *big data* ou o reconhecimento de padrões (Awad & Khanna, 2015).

Em previsão, o objetivo dos métodos de *Machine Learning* é o mesmo dos métodos estatísticos, ou seja, melhorar a precisão da previsão através da minimização de uma função de perda, tal como a soma dos erros quadrados. A sua principal diferença é que os métodos de *Machine Learning* utilizam algoritmos não lineares para atingir este objetivo, enquanto os métodos estatísticos utilizam algoritmos lineares. (Makridakis et al., 2018).

Apesar das potencialidades dos algoritmos de *Machine Learning*, é extraordinariamente intrincado explicar o resultado do output, pois a complexidade do algoritmo gerado faz com que as computações sejam consideradas “caixas negras” de informação (Azodi et al., 2020). O facto destas decisões escaparem do controlo dos gestores é algo que é, usualmente, indesejado pelas organizações. A *Data Science* procura combinar os métodos estatísticos e de *Machine Learning* para produzir modelos fiáveis, compreensíveis, adaptáveis e que permitam utilizar em pleno as funcionalidades dos métodos estatísticos e de *Machine Learning* (Donoho, 2017).

Diferentes métodos podem ser aplicados para previsão. Modelos com base em Redes Neurais Artificiais têm sido estudados devido a não haver necessidade de fazer suposições sobre os dados na construção do modelo, visto que a rede é definida pelas características dos dados (Khashei & Bijari, 2010). Existem diferentes tipos de Redes Neurais, no entanto, o seu funcionamento tem a mesma base. É criada uma rede de “neurónios” interligados, que processam e transmitem informação. Cada neurónio recebe informação, processa-o, e

proporciona um output que se transforma no input da camada seguida. Atribuindo pesos a diferentes parâmetros, o output de cada neurónio é avaliado por uma função de transferência para verificar se ativa o neurónio seguinte (Oken, 2017).

Lauret et al. aplicaram um algoritmo de Redes Neurais Bayesianas num problema de Short term lead forecasting (STLF), ou Previsão de Carga a Curto Prazo, a um caso real, obtendo dados que demonstram a potencialidade desta técnica em evitar um dos problemas mais comuns dos algoritmos de previsão com base em Redes Neurais, o OverFitting (Lauret et al., 2008). Este problema é causado pela alta complexidade dos modelos gerados com estas técnicas, que levam a que o modelo tenha um ótimo desempenho nos dados de treino, mas quando apresentado com novos dados, revela-se incapaz de generalizar (Lauret et al., 2008)

Tendo como base dados reais de procura de um retalhista de moda na Alemanha, Therani et al. utilizaram um modelo de Kernel Machine Regression para prever as vendas de uma época. O modelo utiliza uma regressão logística ordinal para criar clusters de produtos com base no seu perfil de vendas, seguido da aplicação de um modelo de regressão para prever as vendas do produto (Lauret et al., 2008).

Existem outros tipos de técnicas que podem ser aplicadas à previsão. Kück et al. utilizaram dados de procura de uma indústria, utilizadas na competição M3, para desenvolver um modelo de previsão tendo como base o algoritmo de K-Nearest Neighbor. O modelo desenvolvido foi comparado com outros algoritmos tendo como base redes neuronais e o método ARIMA. Os resultados obtidos pelos autores mostram que o modelo com melhor performance foi o método ARIMA, no entanto, o erro do método KNN foi bastante próximo, tendo uma performance a nível de recursos computacionais superior aos métodos de redes neuronais e ao método ARIMA (Kück & Freitag, 2021).

A função de ativação é uma componente essencial em redes neuronais, sendo responsável por introduzir a não-linearidade no processo de treino, o que é crucial para permitir que o modelo simule as relações complexas entre os dados de entrada. A introdução de funções de ativação como a ReLU (Rectified Linear Unit) ou a Maxout tiveram um efeito profundo no desempenho dos modelos de *deep learning* (Gulcehre et al., 2016).

Muitas aplicações de *deep learning* utilizam uma arquitetura de redes neuronais em *feedforward*, em que a rede inicia numa camada inicial de tamanho fixo e termina numa camada final também de tamanho fixo. Para passar entre camadas, os dados são processados em neurónios que calculam a média ponderada dos *inputs* dados pela camada anterior e processam o resultado numa função não-linear. (Rynkiewicz, 2019) Uma das funções mais populares é a ReLU, ou Unidade Linear Retificada, devido à sua simplicidade computacional e a eficácia em mitigar o problema do desaparecimento do gradiente, que impede a atualização dos pesos utilizados nos neurónios.

A otimização dos hiperparâmetros do modelo é uma tarefa crucial no desempenho final do modelo, mas envolve várias decisões subjetivas. Um modelo proposto utiliza o modelo STL (Decomposição de Tendência -Sazonal usando LOESS) para decompor uma série temporal para alimentar um algoritmo de Rede Neuronal Convolucional. O modelo utilizado foi comparado para prever séries temporais contra outros algoritmos tradicionais de *machine learning* e demonstrou ter um desempenho superior nos exemplos aplicados (Vasco-Carofilis et al., 2020). O modelo STL é um procedimento de filtragem que decompõe uma série temporal nos componentes de tendência, sazonalidade e de resíduo, utilizando uma aplicação sequencial de Loess.

Revisão Bibliográfica

A utilização de Recurrent Neural Networks para previsão em séries temporais tem sido bastante utilizada pelos académicos. Slawek Smyl utilizou um modelo híbrido de amortecimento exponencial e de Recurrent Neural Networks na competição M4, que obteve o melhor resultado da competição. A técnica de amortecimento exponencial é utilizada para fazer o pré-processamento dos dados usados para treinar a rede neuronal (Smyl, 2020). O autor demonstra que a utilização das equações de amortecimento exponencial permite ao método capturar as componentes que compõe a série temporal, enquanto o algoritmo de rede neurais captura as tendências não lineares e potencia a aprendizagem cruzada do modelo (Smyl, 2020). O método tem três elementos principais (Smyl, 2020):

- Dessazonalização, através de utilização de fórmulas de amortecimento exponencial;
- Aplicação de uma rede neuronal que é treinada com dados dessazonalizados e normalizados;
- Utilização de métodos de *ensemble* para combinar as previsões do passo anterior.

O modelo de redes neurais Multilayer Perceptron foi utilizado na competição GEFCOM2014 (Global Energy Forecasting Competition 2014), que procura prever os preços de eletricidade do dia seguinte com base em vários preditores. Para ultrapassar o problema de alto tempo de treino necessário para a otimização destes modelos, o autor utilizou o algoritmo de levenberg-marquardt com regularização bayesiana para aproximar a velocidade de treino de uma rede neuronal e reduzir a complexidade do modelo, reduzindo o risco de overfitting (Dudek, 2016).

3. Métodos e Aplicação

Neste capítulo será apresentado o modelo atualmente utilizado pela empresa, assim como uma descrição das suas limitações. Posteriormente é descrito o processo de seleção dos dados utilizados no trabalho, seguido de uma análise dos componentes das séries temporais selecionadas. Por fim, são apresentados os métodos de previsão propostos como alternativa ao modelo atual.

3.1. Estado Atual

Atualmente, a decisão da quantidade de stock a alocar de um determinado artigo a uma loja é efetuada com base no pressuposto que as vendas seguem uma distribuição gaussiana. Os níveis de stock são calculados diariamente, tendo como base os dados de vendas dos últimos 30 dias agrupadas à loja e ao SKU. É calculado um stock de segurança e um stock máximo para cada combinação de loja / artigo, sendo o stock repostado para o stock máximo no dia de trabalho do armazém seguinte ao atingir o stock de segurança.

A fórmula que a empresa usa para calcular o stock mínimo (denominado SM) é apresentada abaixo:

$$SM = (\bar{x} + \sigma * z) * lt_{max}$$

\bar{x} = média de 30 dias de vendas

σ = desvio padrão de 30 dias de vendas

z = nível de serviço

lt_{max} = tempo máximo que um artigo demora a chegar a uma loja (4 dias)

A partir desta fórmula podemos depreender que o objetivo da empresa é que em situações normais de venda, uma loja deve ter em qualquer altura pelo menos 4 dias de stock para qualquer artigo.

O nível de serviço é retirado de uma tabela de distribuição normal e varia com base na classificação ABC do artigo, em valor e quantidade, no último mês.

Na figura 2 podemos verificar a distribuição das vendas, seguindo uma análise de Pareto.

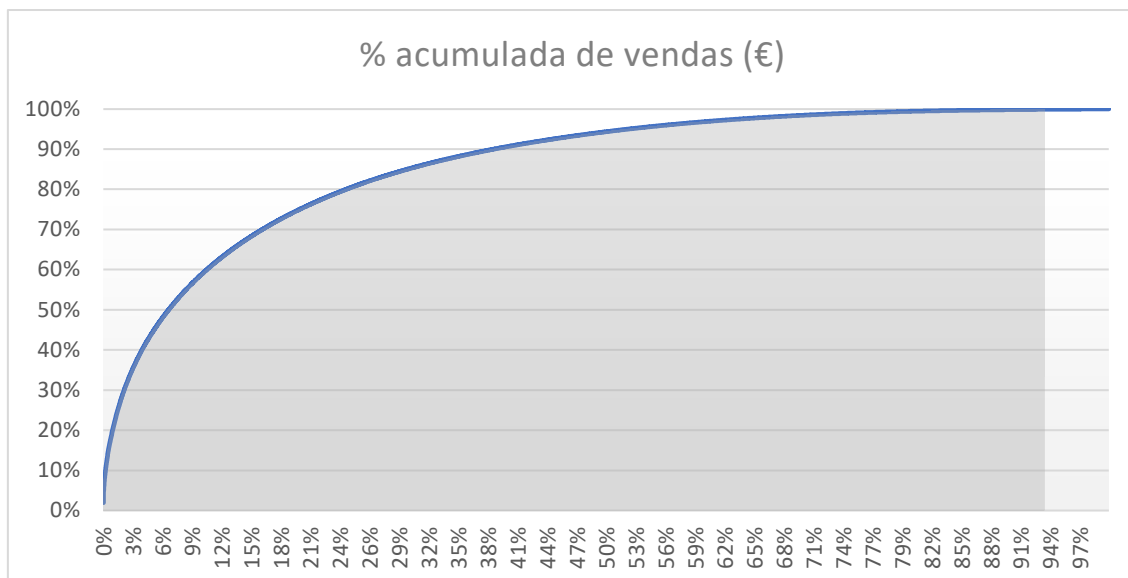


Figura 2 – Análise ABC de 30 dias de vendas (em valor)

Na tabela 1 podemos verificar que cerca de 20 % dos artigos da empresa correspondem a 80 % da faturação total:

Classificação	Total de Artigos	% de itens	Volume
A (0 - 20 %)	61	1%	1 086 711
B (20 - 80 %)	1 516	23%	3 265 058
C (80 - 100 %)	4 948	76%	1 088 276
Total	6 525	100%	5 440 045

Tabela 1 – Resultados da análise ABC (em valor)

Como um dos principais objetivos da empresa é também garantir que a loja tem artigos estratégicos para o cliente profissional – ou seja, artigos de baixo valor, mas que pela sua tipologia são essenciais para o trabalho do profissional, como por exemplo cera depilatória, oxidantes para tintas de cabelo, etc., considera também na atribuição do nível de serviço a análise ABC em quantidade. Podemos verificar a distribuição em quantidade na figura 3.

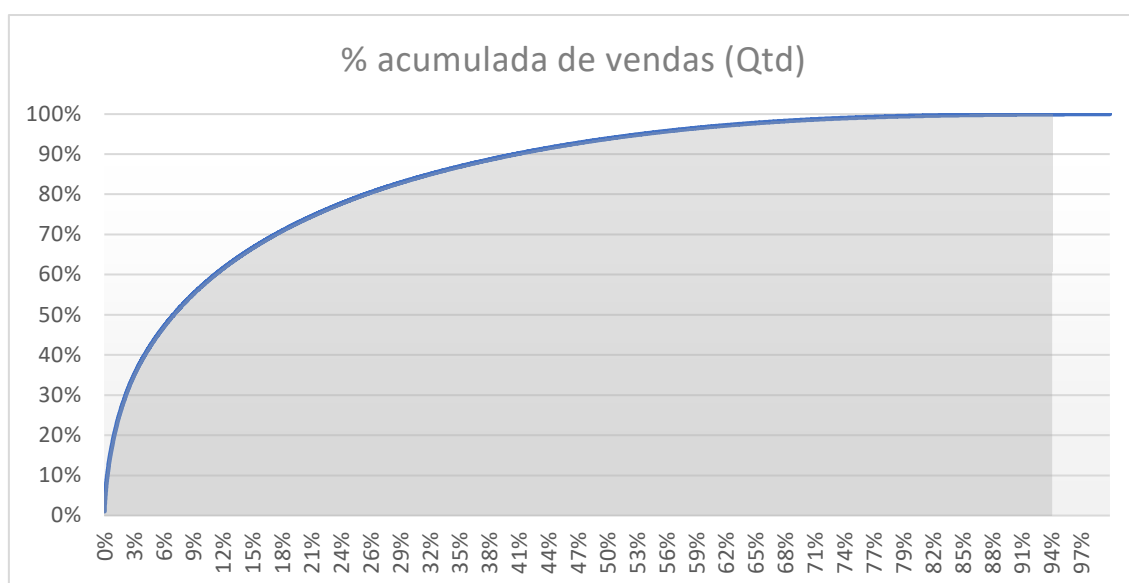


Figura 3 – Análise ABC de 30 dias de vendas (em quantidade)

Os dados extraídos da análise ABC são apresentados na tabela 2.

Classificação	Total de Artigos	% de itens	Volume
A (0 - 20 %)	64	1%	187 567
B (20 - 80 %)	1 631	25%	569 072
C (80 - 100 %)	4 830	74%	189 251
Total	6 525	100%	945 890

Tabela 2 – Resultados da análise ABC (em quantidade)

O nível de serviço do artigo é atribuído através do cruzamento das duas análises ABC. Os valores atualmente utilizados pela empresa são apresentados na tabela 3.

Class. mensal Valor	Class. mensal Qty	Nível de serviço	Cobertura pretendida	Nº artigos	% de itens	Volume faturação	Qtd Vendida
A	A	2,8	99,7%	23	0,4%	425 236 €	76 643
A	B	2,6	99,5%	37	0,6%	650 206 €	28 094
B	A	2,4	99,2%	35	0,5%	165 340 €	41 374
A	C	2,4	99,2%	1	0,0%	11 269 €	87
C	A	2,2	98,6%	6	0,1%	2 791 €	1 110
B	B	2,2	98,6%	1 123	17,2%	2 547 505 €	407 846
B	C	2,0	97,7%	358	5,5%	539 391 €	95 329
C	B	2,0	97,7%	471	7,2%	208 284 €	51 080
C	C	1,6	94,5%	4 471	68,5%	867 736 €	241 386
Total				6 525	100,0%	5 417 758 €	942 949

Tabela 3 – Nível de Serviço do artigo

Métodos e Aplicação

O stock máximo é calculado com base no tempo de stock que a empresa pretende que o artigo tenha na loja. O valor predefinido é de dois dias adicionais ao lead time, contudo, este valor pode ser alterado manualmente pelo gestor de stock. A decisão de aumentar o tempo máximo de stock de um artigo em loja pode ser tomada com base no valor do artigo, na sua dimensão, e na importância estratégica desse artigo para a loja.

A fórmula do stock máximo é apresentada abaixo:

$$S_{max} = (\bar{x} + \sigma * z) * (lt_{max} + t)$$

\bar{x} = média de 30 dias de vendas

σ = desvio padrão de 30 dias de vendas

z = nível de serviço

lt_{max} = tempo máximo que um artigo demora a chegar a uma loja (4 dias)

t = tempo adicional ao lead time que um artigo deve ter de stock

Nas figuras 4, 5 e 6 são apresentados três casos representativos da realidade da empresa. O 1º artigo, que denominaremos por X1100, é um artigo AA, ou seja, é um artigo de alta importância estratégica para a empresa. Sendo um artigo de venda regular, o algoritmo é capaz de responder às necessidades da empresa, como podemos observar na figura 4.

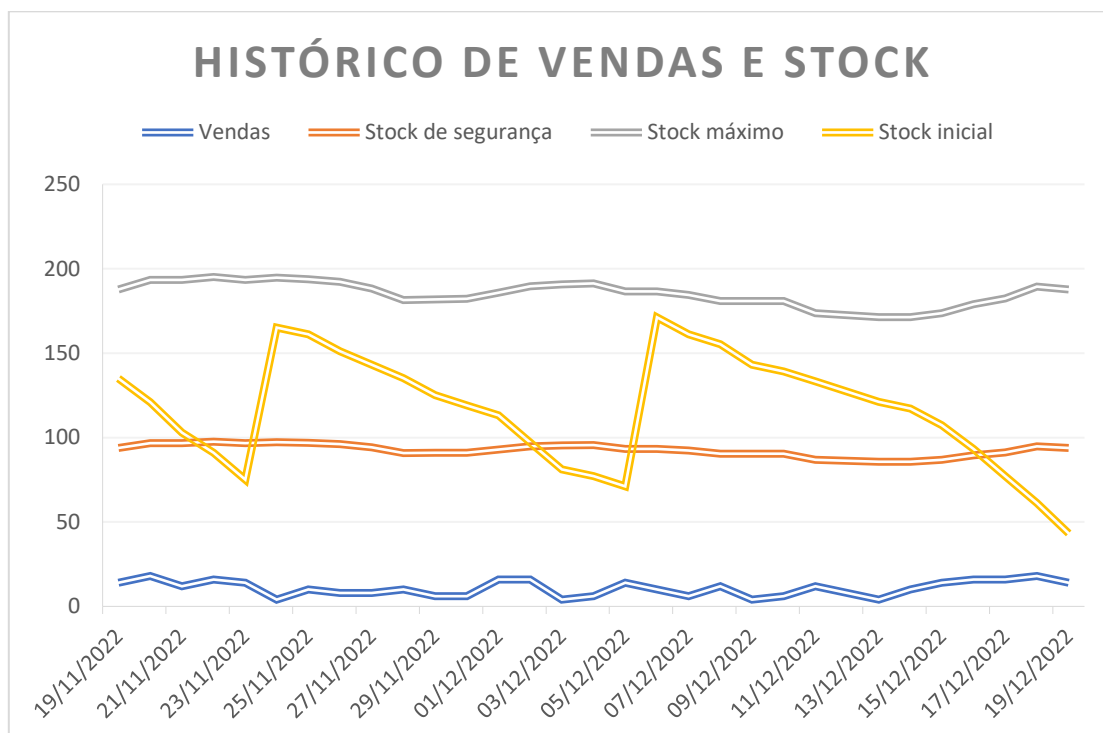


Figura 4 – Histórico de vendas e stock do artigo X1100 na loja 036

Podemos observar que após o artigo atingir o ponto de reabastecimento, o stock de reposição chega à loja após 2 dias. O mesmo não é observado no dia 19/12/2022, visto o artigo ter descido do ponto de reabastecimento a um sábado.

Devido a assumir-se a normalidade das vendas, a resposta do algoritmo aos picos de venda, como por exemplo, artigos com sazonalidade ou campanha promocional, é tardia. De seguida apresentamos um caso de artigo com sazonalidade na época balnear, que denominaremos como X1200. Podemos observar o seu comportamento na figura 5.

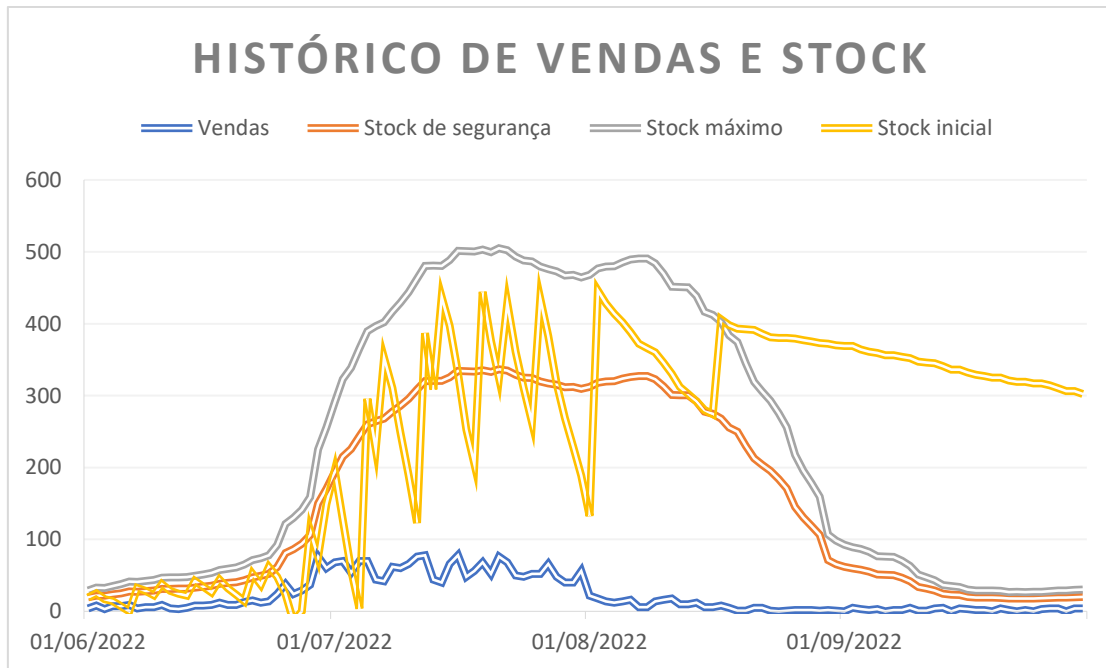


Figura 5 – Histórico de vendas e stock do artigo X1200 na loja 027

Podemos observar que a resposta inicial do algoritmo é tardia, o que resulta em perda de vendas para a empresa, e que o seu ajustamento final é demorado, o que resulta em custos elevados para a empresa por excesso de stock.

No 3º caso apresentado, verificamos o efeito de uma campanha promocional no artigo X2100. A empresa iniciou uma campanha promocional no dia 12/09/2022, sábado, o que levou a que o artigo entrasse em rotura nos primeiros 3 dias de campanha. O ajuste do algoritmo é feito de forma tardia e excessiva, originando novamente custos desnecessários para a empresa.

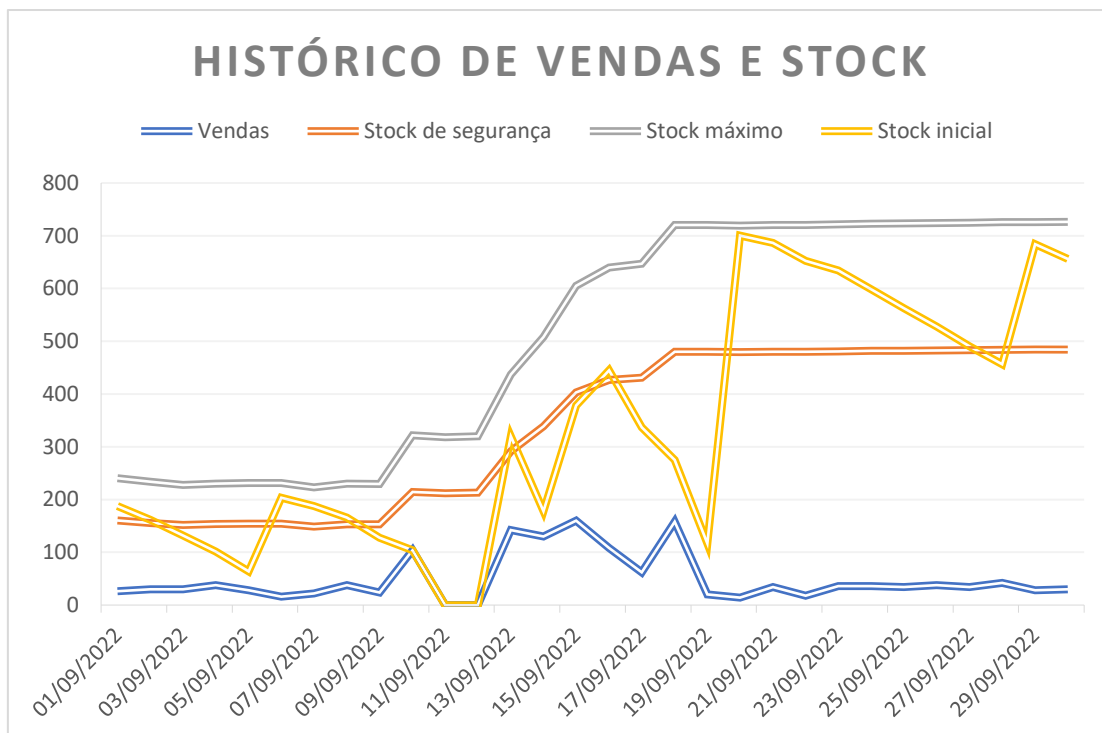


Figura 6 – Histórico de vendas e stock do artigo X1200 na loja 027

Nos casos dos artigos com sazonalidade e afetados por uma campanha promocional, podemos observar o efeito chicote decorrente do pico de vendas, ou seja, a procura do artigo pelo cliente na loja é desfasada do reabastecimento da loja pelo armazém, o que gera oportunidade de venda perdida e valores excessivos de stock. Como as vendas destes artigos apresentam distribuições não normais, os valores do desvio padrão fazem com que a quantidade enviada para as lojas seja bastante acima das necessidades reais.

Atualmente, cabe ao gestor de stock alocar inputs manuais ao algoritmo para prevenir situações de perda de venda ou de excesso de stock. O objetivo principal deste projeto será a criação de um algoritmo funcional, que considere as variações previstas nos níveis de venda, com base em eventos conhecidos como sazonalidade, campanhas promocionais, fim de ciclo do produto, de forma a maximizar vendas e reduzir custos operacionais e de inventário.

3.2. Seleção dos dados

Neste subcapítulo é descrito o processo de seleção, recolha e tratamento dos dados utilizados para previsão.

3.2.1. Horizonte temporal

Para criar o modelo inicial, foi selecionada uma amostra de dados que se considera representativa do universo de lojas físicas. Para que esta amostra capture a evolução da empresa e os diferentes componentes de uma série temporal, tais como a sazonalidade anual e a tendência, que têm um impacto significativo nas vendas da empresa, foram selecionados 3 anos completos de venda. Devido à pandemia de COVID-19, os anos de 2020 e 2021 representam uma realidade completamente distinta do funcionamento normal da empresa, devido aos períodos em que as lojas físicas estiveram fechadas ou com funcionamento condicionado. Assim, de forma a poder trabalhar com os dados mais recentes, foram considerados os dados de vendas de 2018, 2019 e 2022.

3.2.2. Seleção da loja

Foram selecionados dados de vendas de apenas uma loja. Foram considerados os seguintes critérios na escolha:

- Não ter alterações significativas no período selecionado (remodelações, realocações, etc.);
- Ter um volume de faturação significativo para a empresa;
- Loja com horário de funcionamento completo (sem fechar ao fim de semana e feriados);
- Não ter limitações de gamas ou de quantidades de stock devido ao espaço;
- Ter um padrão de venda semelhante ao da empresa.

Tendo estes fatores em consideração, a loja selecionada foi a 027, que fica localizada dentro de um centro comercial do município de Sintra. A loja não passou por nenhuma remodelação extensiva entre 2018 e 2022, trabalha com horário completo, é considerada uma loja de dimensão grande para a realidade da empresa, e foi a loja que mais faturou nos anos de 2018 e 2019, ficando em segundo lugar em 2022. Adicionalmente, tem um padrão de venda semelhante à realidade global da empresa, como se pode verificar na tabela 4:

Família	Vendas Lojas Físicas	Peso % Família	Vendas Loja 027	Peso % Família
Haircare	19 955 598 €	34,8%	270 019 €	32,6%
Technical	11 316 515 €	19,8%	155 565 €	18,8%
NailCare	7 223 730 €	12,6%	112 231 €	13,5%
Equipments	4 814 063 €	8,4%	79 196 €	9,6%
Acessories	4 575 711 €	8,0%	78 203 €	9,4%
Make Up	3 570 301 €	6,2%	40 607 €	4,9%
Aesthetics	4 542 915 €	7,9%	75 453 €	9,1%
Seasonal	469 881 €	0,8%	4 808 €	0,6%
Men	457 032 €	0,8%	7 980 €	1,0%
Other	352 860 €	0,6%	4 467 €	0,5%
Total Geral	57 278 606 €	100,0%	828 529 €	100,0%

Tabela 4 – Peso de venda por família das lojas físicas VS loja 027 no ano de 2022

Comparando a evolução da faturação em euros, também é possível concluir que a loja tem um comportamento semelhante ao longo do ano com as restantes lojas da empresa. A figura 7 compara o volume de faturação diário normalizado das lojas físicas da empresa com a loja 027 diariamente, durante o ano de 2022. A correlação entre as duas séries temporais é de 0,64, ou seja, a faturação da loja 027 apresenta uma correlação forte com a faturação da restante empresa.

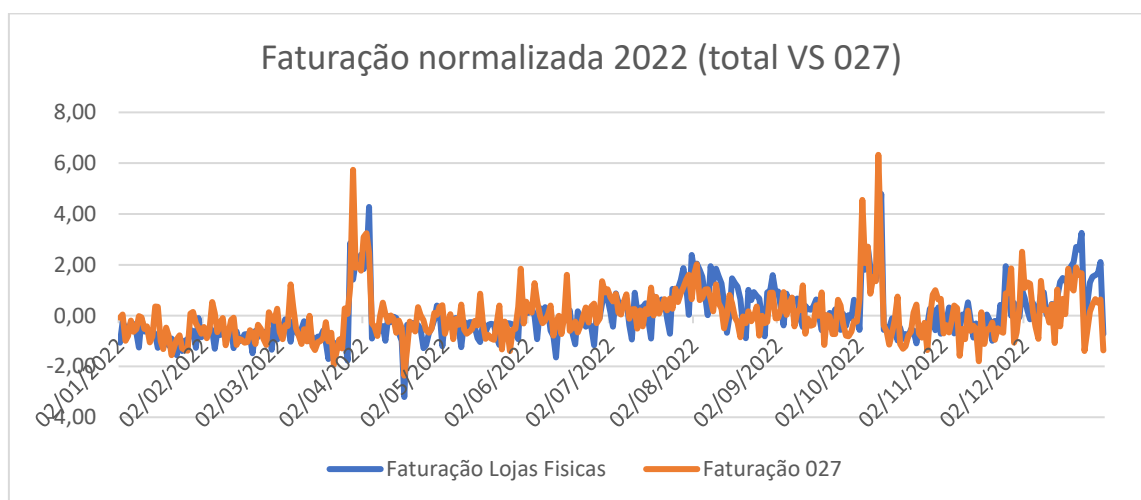


Figura 7 – Faturação normalizada em 2022 das lojas físicas VS loja 027

É necessário salientar, no entanto, que a loja foi afetada pela expansão da empresa em 2022, devido à abertura de outra loja no município de Sintra. O crescimento anual da loja nesse ano foi de apenas 2%, comparado com 7% em outras lojas de Centro Comercial *Like for Like* (ou seja, lojas abertas durante esse período sem remodelações).

3.2.3. Seleção dos artigos

Para testar o modelo, foi selecionado um conjunto de artigos que se consideram representativos da realidade da empresa. Os critérios para seleção dos artigos foram o seguinte:

- Ter um volume de vendas em euros significativo, ou ser um artigo de importância estratégica para a empresa;
- Não ter sido descontinuado durante o período em análise;
- Não ter tido alterações significativas na embalagem ou na dosagem no período em análise;
- A quantidade vendida deve ser expressiva para a análise.

O 2º artigo mais vendido pela loja durante o período foi um artigo da família *Haircare* que foi descontinuado em junho de 2022, pelo que a sua inclusão desvirtuaria a análise. O 7º artigo foi um artigo da família *Equipments*, que tem um custo unitário perto dos 200 € - no entanto, o artigo apenas vendeu 88 unidades no período em análise, pelo que não é relevante fazer previsão de vendas. A família mais prejudicada por estes critérios é a família dos cabelos, que representa o maior peso na venda da empresa, onde é habitual que os fornecedores apresentem novos produtos, com ligeiras alterações na fórmula ou na embalagem, o que obriga à criação de um novo SKU.

Adicionalmente, pretende-se que esta análise inicial comporte diferentes artigos, com diferentes comportamentos. Foram assim adicionados os artigos mais vendidos das famílias *Haircare*, *Technical* e *Nails*. Adicionalmente, foram adicionados alguns artigos da família “*Aesthetics*”, da Subfamília *Epilation*, que se sabe ter sazonalidade. A seleção inicial é resumida na tabela 5.

SKU	Familia	SubFamilia	Qty Vendida 3 anos	Valor Total	Ranking
X00001	Technical	Technical	10 240	22 108 €	3
X00012	Aesthetics	Epilation	8 317	6 352 €	39
X00013	Technical	Technical	7 316	4 536 €	87
X00007	Technical	Coloring	6 176	17 968 €	5
X00014	Nails	Nails	5 960	4 847 €	73
X00002	Nails	Gel Polish	5 721	27 368 €	1
X00008	Technical	Coloring	5 709	16 648 €	6
X00003	Technical	Technical	5 548	11 343 €	15
X00009	Technical	Coloring	5 236	15 149 €	8
X00005	Technical	Technical	4 505	6 006 €	43
X00015	Aesthetics	Epilation	4 434	4 994 €	67
X00010	Technical	Coloring	4 240	12 432 €	9
X00016	Aesthetics	Epilation	4 238	4 802 €	76
X00004	Nails	Hybrid	3 784	11 631 €	14
X00017	Aesthetics	Epilation	3 732	4 029 €	108
X00018	Aesthetics	Epilation	2 936	4 433 €	90
X00006	Technical	Technical	2 574	19 225 €	4
X00019	Haircare	Finisher	1 796	10 690 €	16
X00020	Haircare	Finisher	1 794	5 389 €	55

Tabela 5 – Artigos Selecionados

3.2.4. Granularidade dos dados

A questão da granularidade dos dados é essencial para a criação do modelo. Devido à reposição da loja a partir do centro de distribuição ser feita diariamente, com um tempo de entrega de 1 dia útil, não é necessária uma granularidade menor que a diária. Como um dos objetivos da empresa é a minimização da quantidade de stock em loja, de forma a minimizar o valor de stock e maximizar o espaço em loja, uma granularidade maior que a semanal é considerada excessiva, pois obrigaria a ter stock no ponto disponível para cobrir um horizonte temporal maior.

Considerando que a empresa não comercializa produtos rapidamente perecíveis e que a empresa também tem como objetivo minimizar o número de vezes que um produto é separado em armazém, de forma a reduzir os custos de transporte e de operação, o agrupamento dos dados à semana pode ser adequado. Adicionalmente, concentrar os dados semanais permite uma melhor análise dos efeitos das campanhas promocionais, que são maioritariamente realizadas semanalmente.

Outro fator tido em consideração é o da grandeza dos números – como o objetivo é prever a quantidade vendida por um determinado produto numa determinada loja, o volume dos dados não é considerável, pelo que ao agrupamento dos dados à semana permite reduzir o ruído, como pode ser observado ao analisar as figuras 8 e 9. Os picos em 2022 são derivados de uma campanha comercial com um desconto especialmente agressivo, sendo que a quantidade vendida nos dois anos é bastante semelhante – no entanto, a análise diária do mesmo período é bastante mais dispersa.

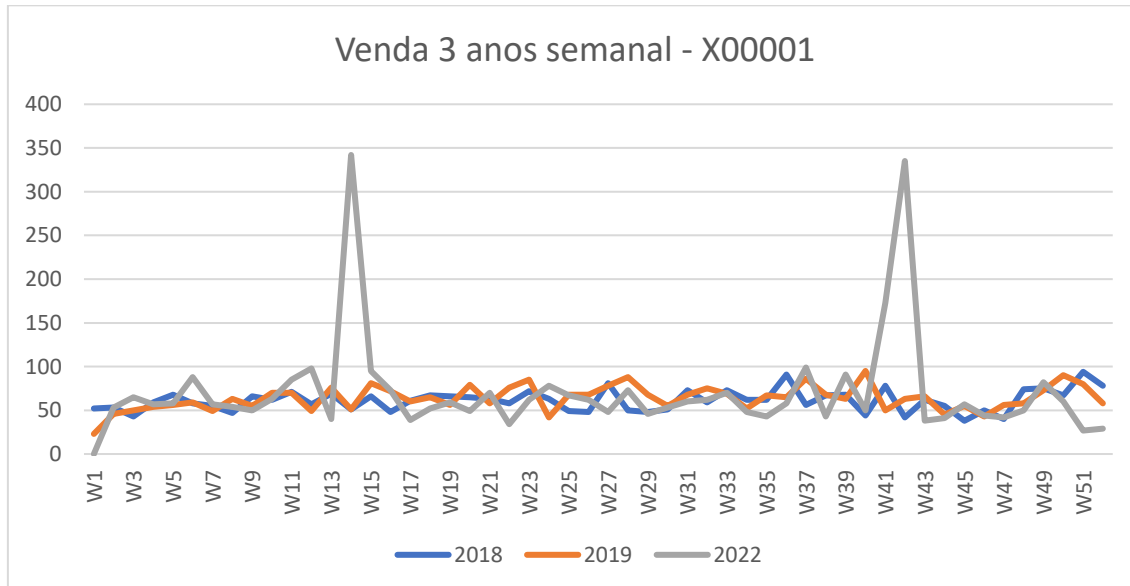


Figura 8 – Vendas semanais do artigo X0001

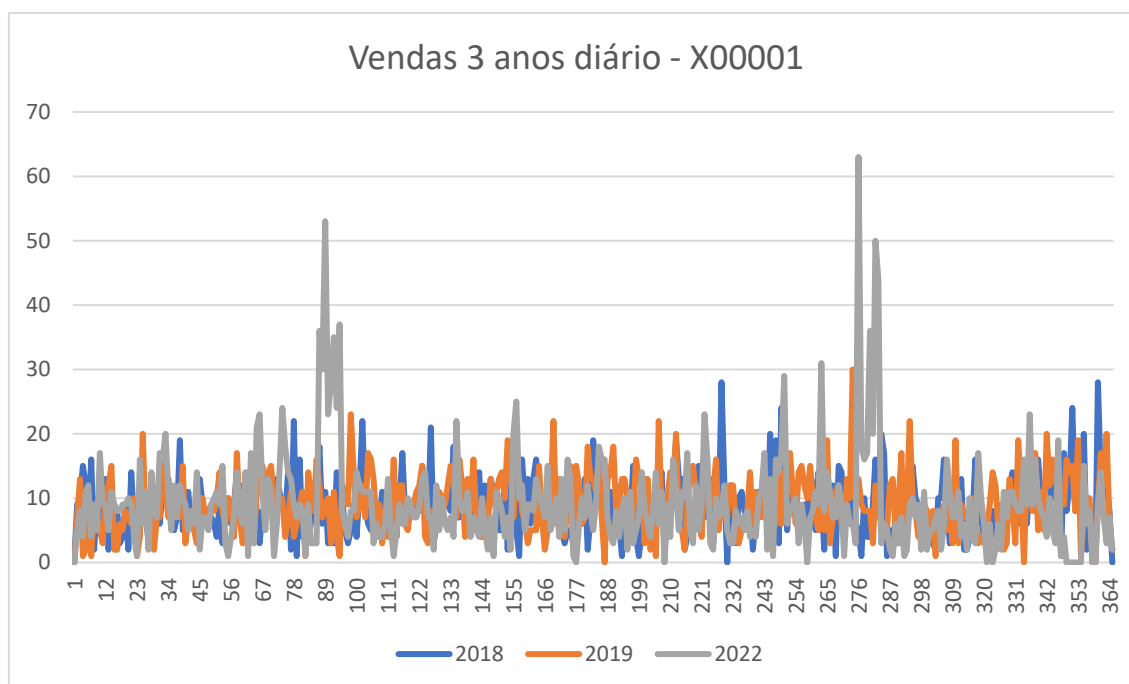


Figura 9 – Vendas diárias do artigo X0001

3.3. Análise dos dados

A identificação dos componentes presentes numa série temporal permite selecionar e adequar o modelo de previsão com base no comportamento passado e no que sabemos que vai acontecer no futuro, como campanhas promocionais ou de marketing, épocas festivas, etc. A *representação* da série temporal num gráfico de linhas permite ao analista identificar a presença de outliers, dados em falta, ou estabelecer a hipótese da presença de componentes como a sazonalidade ou a tendência.

Estas hipóteses podem ser reforçadas com uma análise de autocorrelação, que mede a relação entre valores desfasados da mesma série temporal. Quando uma série temporal tem tendência, as autocorrelações para desfasamentos curtos tendem a ser altas e positivas, que caem lentamente à medida que o desfasamento aumenta. Quando existe sazonalidade, as autocorrelações tendem a ser maiores nos intervalos com sazonalidade do que nos outros. Em séries temporais com sazonalidade e tendência, vemos uma combinação dos dois efeitos, com uma autocorrelação alta nos desfasamentos curtos diminui à medida que os desfasamentos aumentam, mas com picos nos intervalos sazonais (Hyndman & Athanasopoulos, 2021). Se a série não tiver autocorrelação, ou seja, se for composta por ruído branco, então podemos esperar que 95% dos valores de autocorrelação estejam dentro de $\pm 2/\sqrt{T}$, em que T é o tamanho da série temporal (Hyndman & Athanasopoulos, 2021).

Para identificar anomalias na série temporal, é possível gerar um modelo de previsão probabilístico, com um nível de confiança alto, e comparar a previsão obtida para um dos períodos utilizados nos dados de treino com a previsão do modelo para esse mesmo período. Se a previsão estiver fora dos valores mínimos e máximos para um intervalo de confiança de 99%, podemos inferir que esses dados são outliers. Bandara et al. criaram um modelo que permite a decomposição de séries temporais nos seus componentes. Este modelo, chamado *Multiple*

Métodos e Aplicação

Seasonal-Trend decomposition using Loess (MSTL, ou Decomposição múltipla de Sazonalidade-Tendência usando Loess), é uma extensão do modelo *STL* que permite o tratamento de séries temporais com sazonalidades em diferentes defasamentos temporais.

3.3.1. Identificação dos componentes

O artigo X00001 é um artigo da família Technical, que apresenta um comportamento estável ao longo do ano. É o artigo mais vendido pela empresa em quantidade e é considerado um *commodity* no ramo cosmético, sendo um artigo não diferenciado. Podemos observar na figura 11 que o comportamento do artigo ao longo dos 3 anos de análise apresenta bastante estabilidade, com um pico de venda na semana 119 e outro na semana 146. Estes dois picos são frutos de uma forte campanha promocional que a empresa efetuou em duas semanas distintas no ano de 2022, que afeta tanto a divulgação do artigo como o preço do mesmo, e que tem um grande impacto na procura da empresa nessas semanas e nas semanas imediatamente antecedentes e posteriores a esse evento – podemos observar na figura 10 que existe uma quebra na procura na semana anterior, quando é habitualmente feita a divulgação da data da campanha, e nas semanas seguintes.



Figura 10 – Venda de 156 semanas do artigo X00001

Analisando o gráfico de autocorrelação na figura 11, podemos observar que apenas um valor saí fora do intervalo $\pm 2/\sqrt{T}$, representado pelo fundo azul, pelo que podemos concluir que a série é composta por ruído branco.

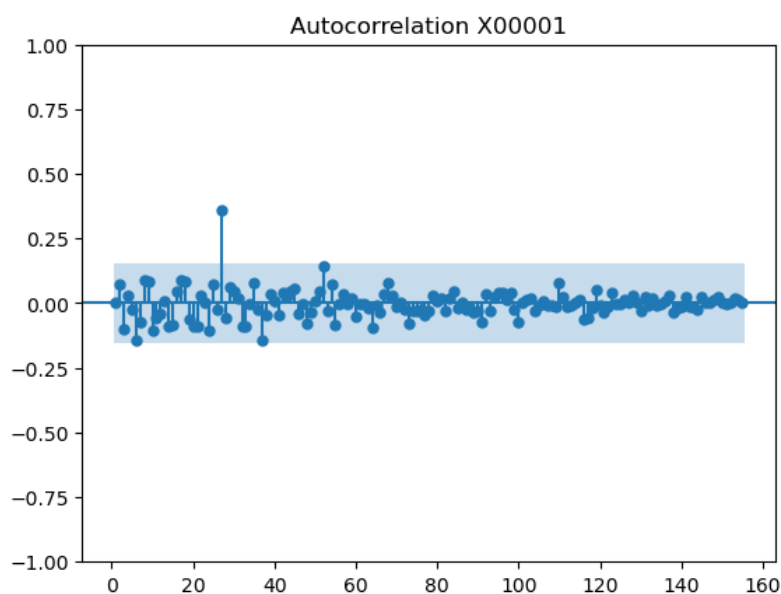


Figura 11 – Autocorrelação X00001

Usando o modelo *MSTL* para criar um modelo de previsão com a série temporal e utilizando esse modelo para prever o comportamento da série temporal, podemos observar na figura 12 que são identificados 5 outliers, nas semanas 67, 74, 80, 94 e 102.

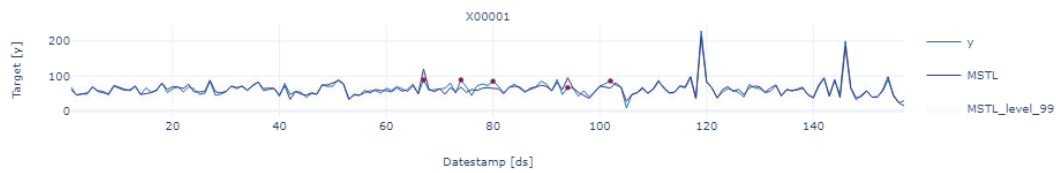


Figura 12 – Anomalias X00001

O artigo X00002 é um artigo da família *Nails*. Podemos observar na figura 13 que o produto tem uma procura estável. O pico na semana 119 é fruto da campanha promocional mencionada anteriormente, sendo que o artigo não esteve em campanha na semana 146.

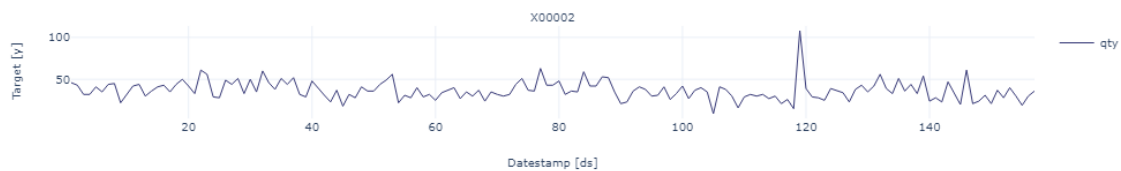


Figura 13 – Venda de 156 semanas do artigo X00002

Através da análise do gráfico de autocorrelação na figura 14, podemos concluir que a série temporal é composta por ruído branco.

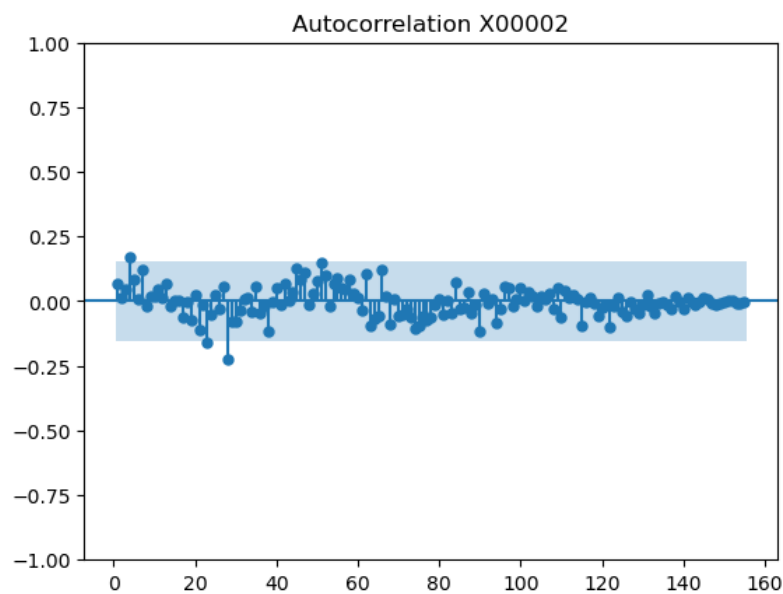


Figura 14 – Autocorrelação X00002

A análise de anomalias revela que existem 5 períodos com dados anómalos, como observado na figura 15.

Métodos e Aplicação

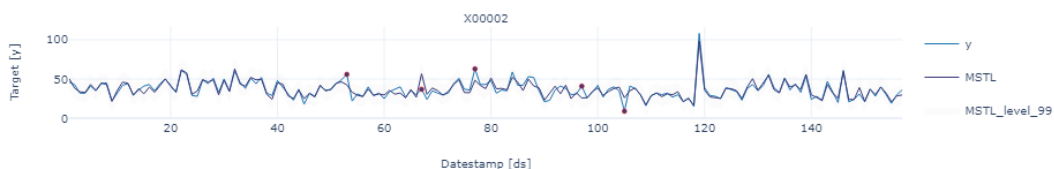


Figura 15 – Anomalias X00002

A análise gráfica das vendas do produto X00003 (figura 16) parece indicar uma sazonalidade anual, visto que apresenta um formato elíptico com picos a meio do ano e que decaí ao longo do ano. Essa sazonalidade aparenta não ser tão presente no terceiro ano.



Figura 16 – Venda de 156 semanas do artigo X00003

O gráfico de autocorrelação na figura 17 reforça a presença de sazonalidade, como podemos observar pela flutuação da autocorrelação com picos a cada 52 semanas. Como mais de 5% dos picos estão fora da zona azul, podemos afirmar que este efeito não é apenas ruído branco.

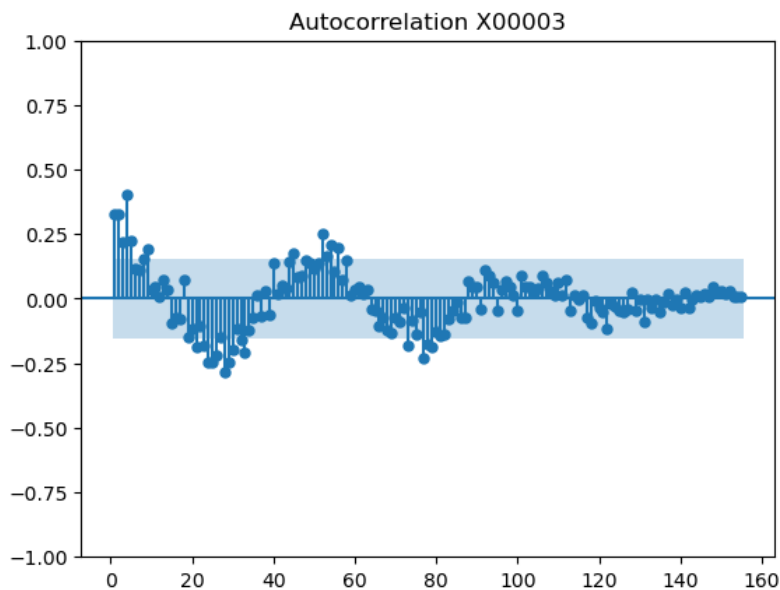


Figura 17 – Autocorrelação X00003

A análise de anomalias, na figura 18, revela 5 períodos com dados anómalos.

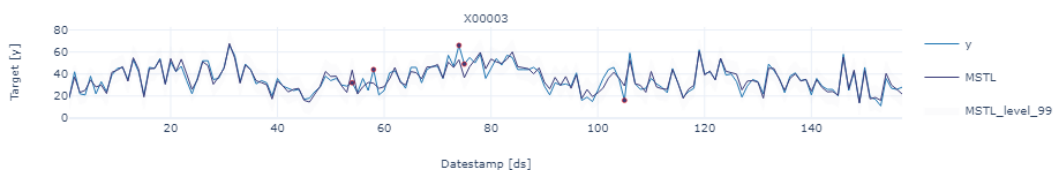


Figura 18 – Anomalias X00003

Na figura 19, podemos observar que o artigo X00004 apresenta indícios de sazonalidade anual. Existem picos nas semanas 119 e 146, frutos de uma campanha promocional.

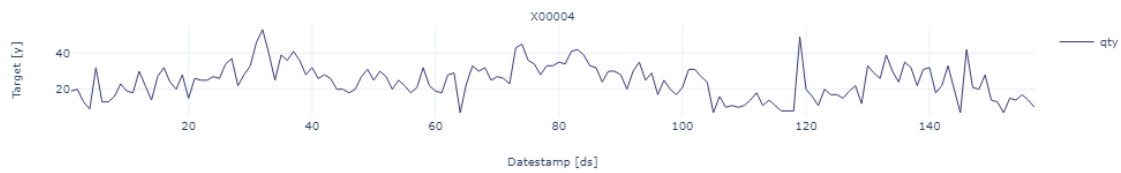


Figura 19 – Venda de 156 semanas do artigo X00003

O formato ondulado do gráfico de autocorrelação na figura 20 suporta a possibilidade de existir sazonalidade.

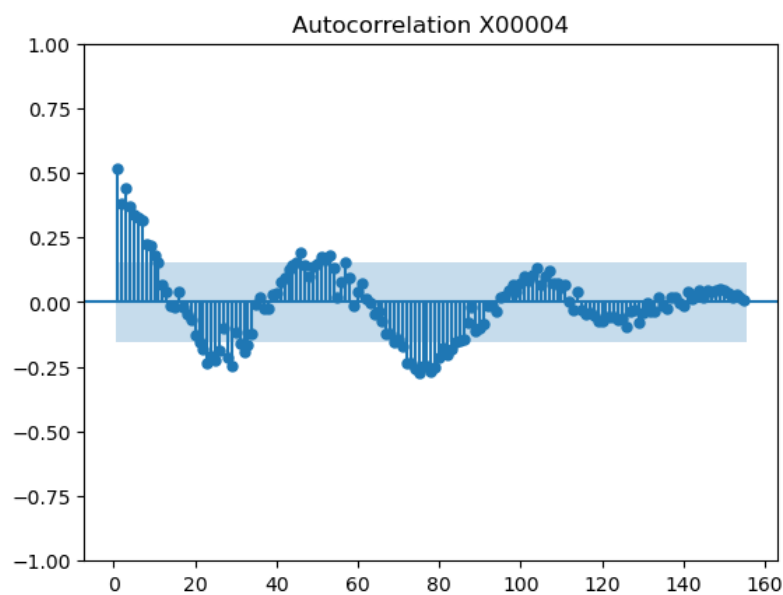


Figura 20 – Autocorrelação X00004

Na figura 21 podemos observar 4 anomalias, entre a semana 65 e a semana 90.

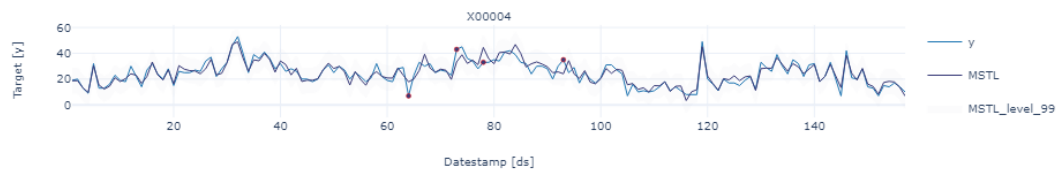


Figura 21 – Anomalias X00004

A análise gráfica da progressão de vendas do artigo X00005, que pode ser observada na figura 22, indica que o produto tem uma venda bastante regular. O facto de a semana 51 ter procura zero pode indicar um período de rotura.

Métodos e Aplicação



Figura 22 – Venda de 156 semanas do artigo X00005

Através da análise de autocorrelação na figura 23, podemos inferir que a série temporal é composta apenas por ruído branco, visto mais de 95% dos pontos estarem dentro do patamar $2/\sqrt{T}$.

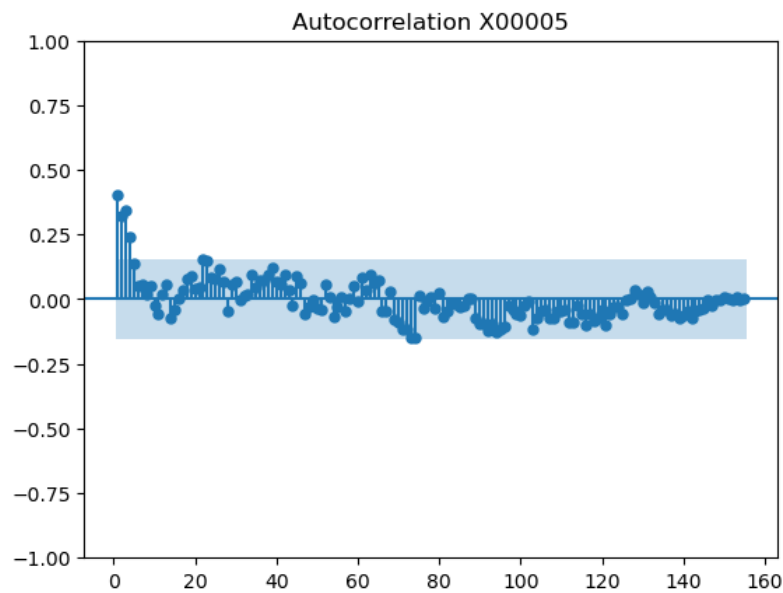


Figura 23 – Autocorrelação X00005

A análise de anomalias na figura 24 indica 3 períodos anómalos.

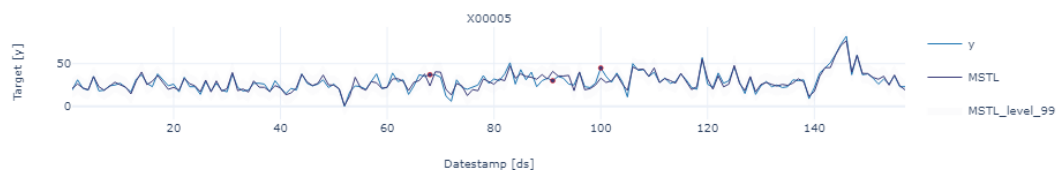


Figura 24 – Anomalias X00005

Os artigos X00006, X00007, X00008, X00009 e X00010 têm um padrão de procura semelhante ao longo dos 3 anos de análise – isto deve-se ao facto de serem artigos da família “technical”, que são de utilização regular pelos profissionais da área de cosmética, e que são impactados pelas mesmas campanhas. Podemos observar na figura 25 que existem picos nas semanas 48 e 92 que são comuns a todos os artigos, assim como uma quebra na procura na semana 106. Todos os artigos tiveram um grande pico de procura nas campanhas promocionais da semana 118 e 146, com o mesmo comportamento de uma quebra da procura nos períodos

imediatamente anteriores e subsequentes. As análises destes gráficos não demonstram tendência nem sazonalidade.

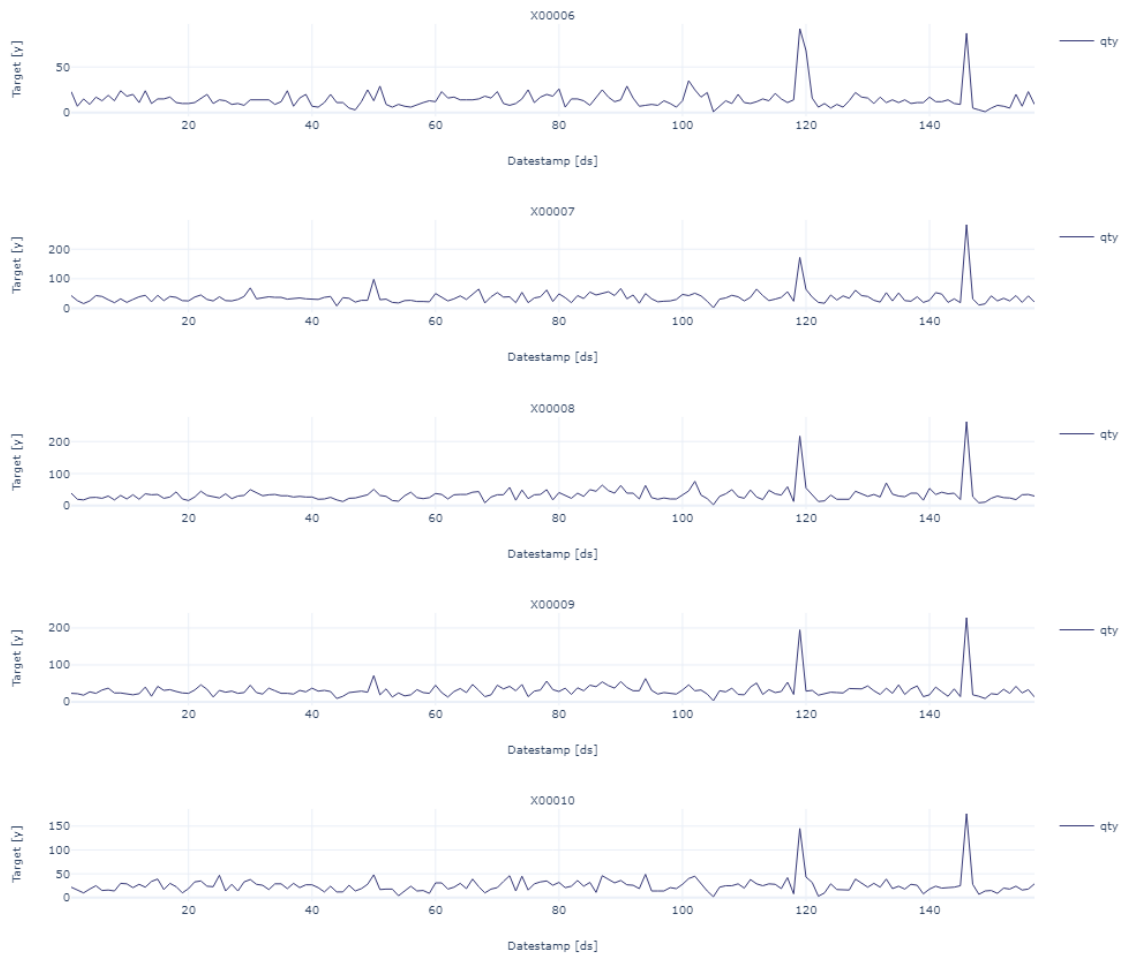


Figura 25 – Venda de 156 semanas de artigos com características similares

A análise dos gráficos de autocorrelação na figura 26 corrobora a hipótese de as séries temporais serem compostas por ruído branco, visto estarem praticamente todos os valores dentro do patamar $\pm 2/\sqrt{T}$.

Métodos e Aplicação

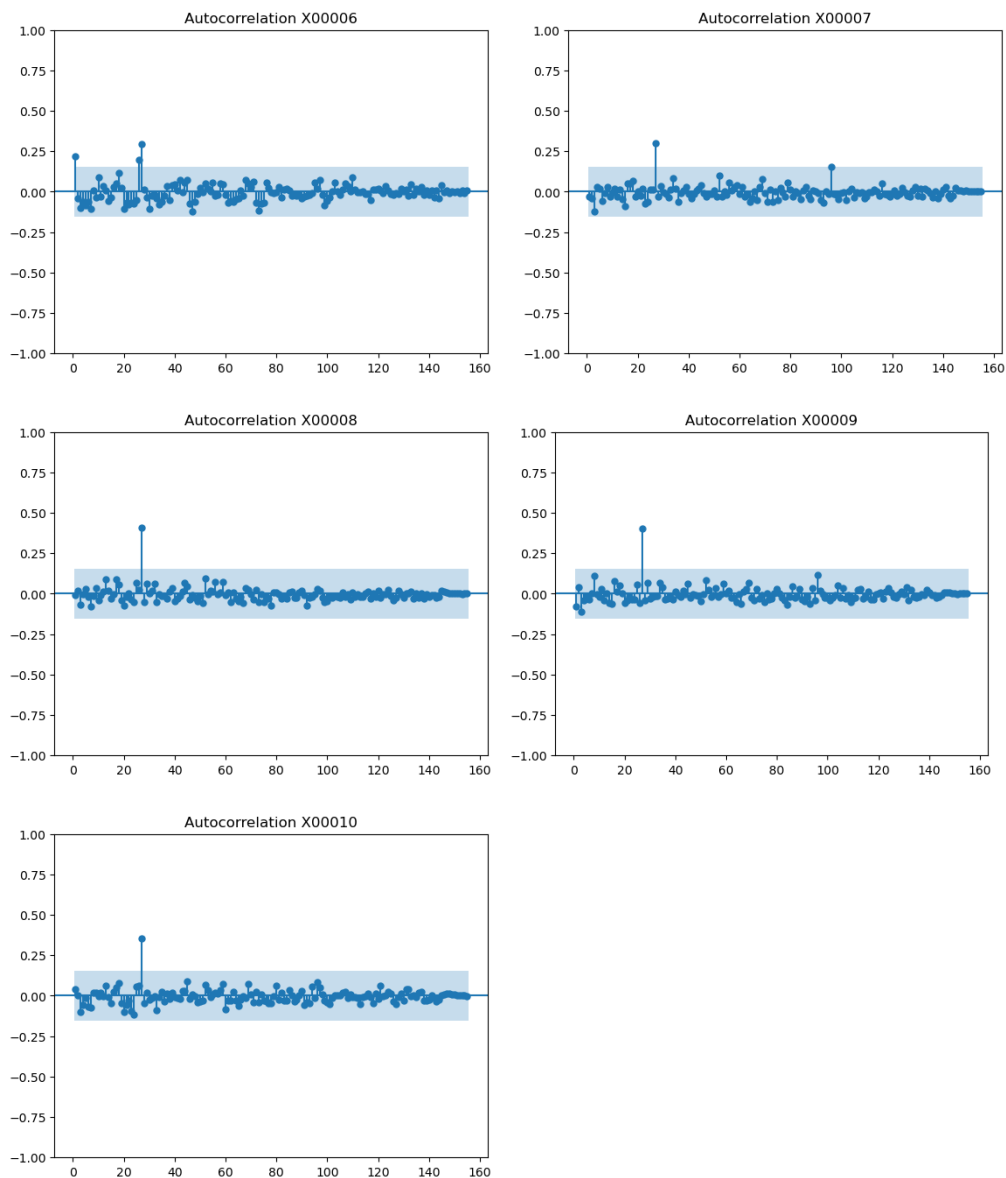


Figura 26 – Autocorrelação de artigos com características similares

Através das análises de anomalias, podemos observar na figura 27 que todos os artigos apresentam entre 4 e 5 períodos anómalos, sendo que os períodos de campanha são particularmente suscetíveis de serem considerados anormais.

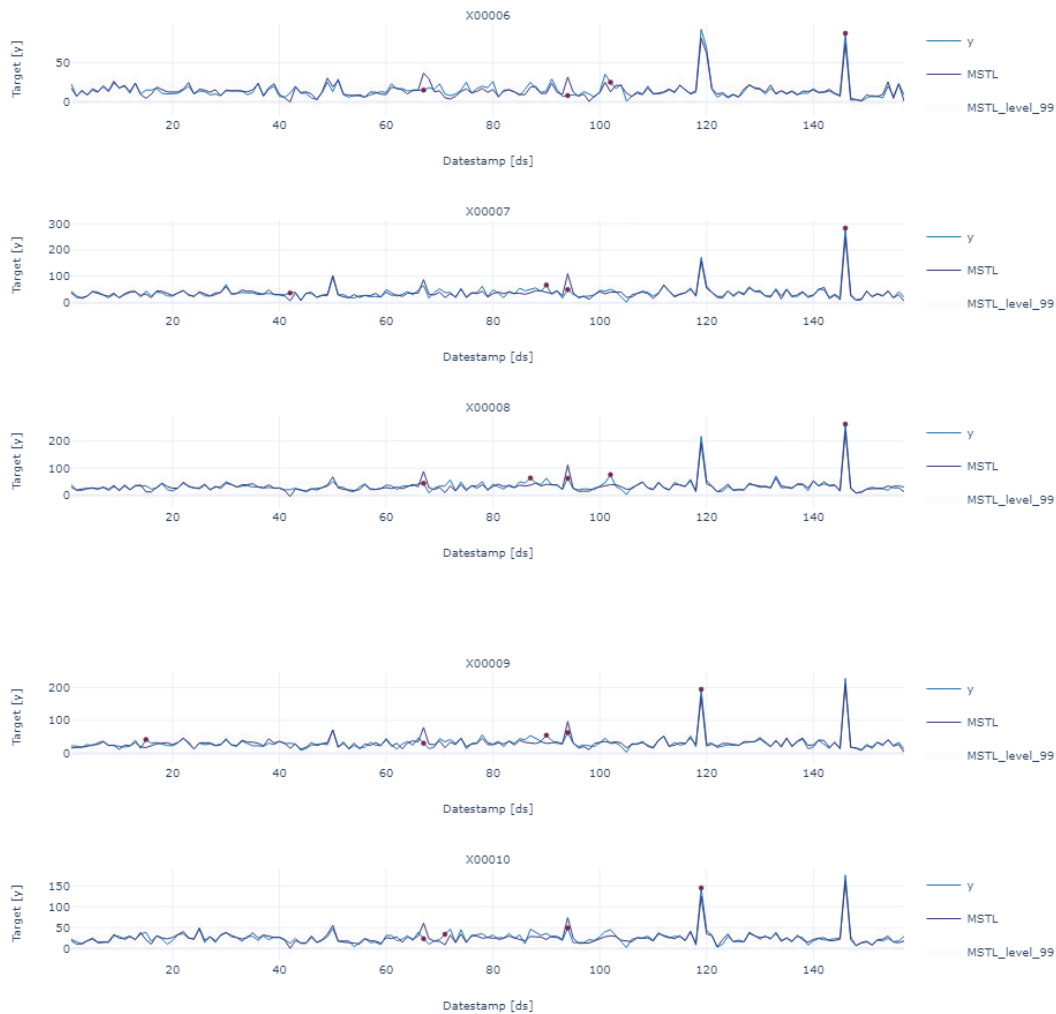


Figura 27 – Anomalias de artigos com características similares

A série temporal do artigo X00012 apresenta vários picos ao longo dos 3 anos, com uma sazonalidade aparente que culmina na altura do Verão. É de salientar que existem quebras na procura abruptas durante o Verão, como podemos observar nas semanas 72 e 133. Estas quebras podem ser originadas por roturas intermitentes ou por campanhas promocionais.



Figura 28 – Venda de 156 semanas do artigo X00012

O gráfico de autocorrelação da figura 29 apresenta o formato ondulado entre valores positivos e negativos, com picos a cada 52 semanas, que é habitualmente associado a sazonalidade. O formato elíptico da curva do gráfico pode também indicar que a série temporal tem tendência.

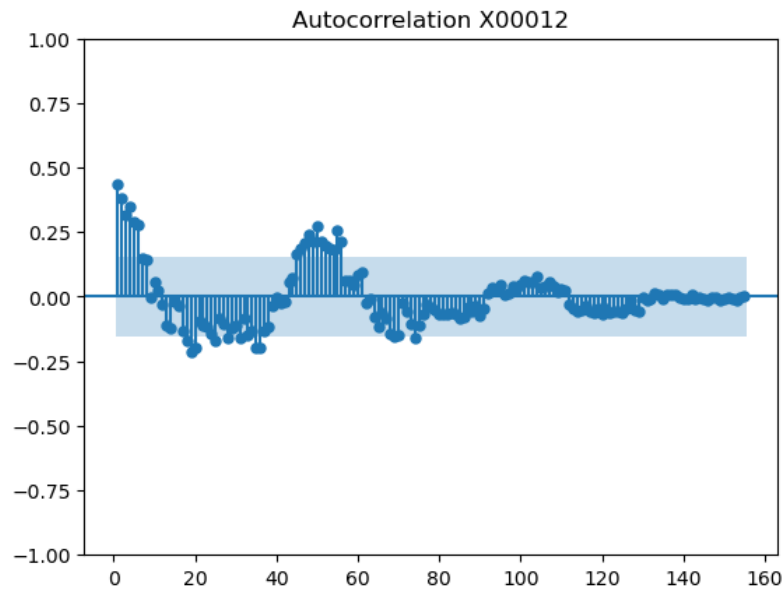


Figura 29 – Autocorrelação X00012

A análise de anomalias indica apenas dois períodos anómalos.

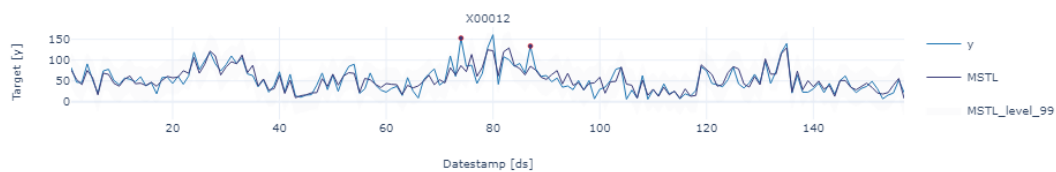


Figura 30 – Anomalias X00012

O produto X00013 apresenta uma procura estável. Na figura 31, podemos observar que nas semanas 107, 108 e que entre a semana 142 e a 151 existem vários períodos com venda zero, o que pode indicar períodos alargados de rotura. Existe um pico de venda no período 118, fruto da campanha promocional mencionada anteriormente.



Figura 31 – Venda de 156 semanas do artigo X00013

A análise do gráfico de autocorrelação na figura 32 sugere que o artigo tem sazonalidade e tendência, devido ao formato ondulado do gráfico e dos valores de autocorrelação irem caindo ao longo do tempo.

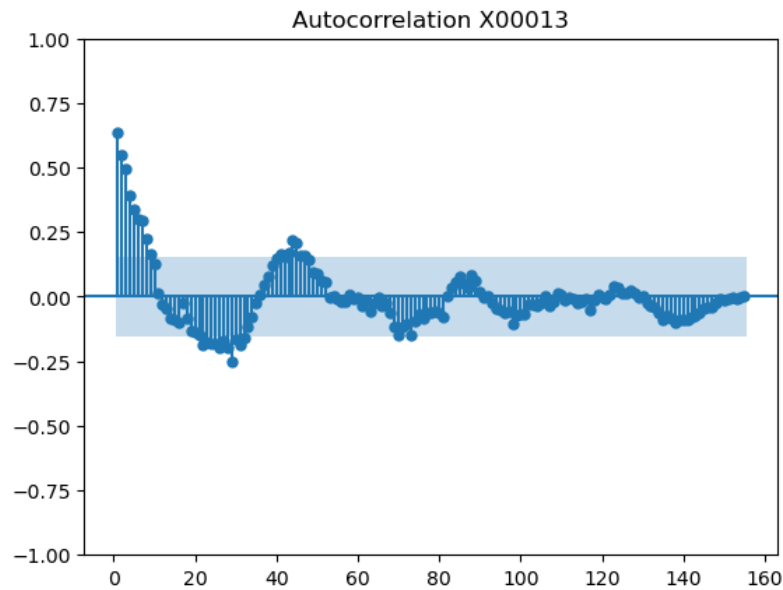


Figura 32 – Autocorrelação X00013

A análise de anomalias na figura 33 apresenta 3 períodos anómalos, entre os períodos 45 e 75.

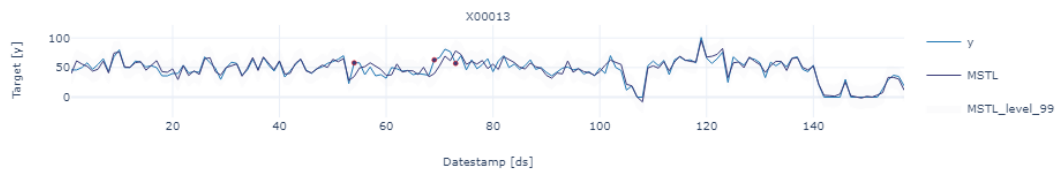


Figura 33 – Anomalias X00013

O artigo X00014 apresenta vários períodos com vendas zero entre o período 0 e 20, o que pode indicar uma rotura prolongada. Na figura 45, podemos observar outros períodos com vendas zero nas semanas 85,86, e 120. É também perceptível um aumento na quantidade vendida ao longo do tempo, o que indica que a série temporal tem uma tendência.



Figura 34 – Venda de 156 semanas do artigo X00014

A existência de tendência é suportada pela análise de autocorrelação – os valores são altos para os períodos com intervalos curtos, diminuindo à medida que o desfasamento entre os períodos vai aumentando.

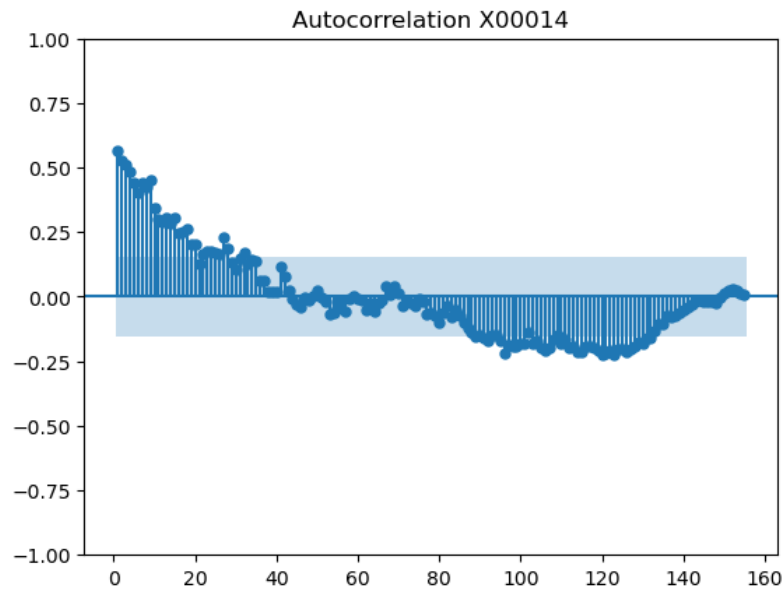


Figura 35 – Autocorrelação X00014

A análise de anomalias indica apenas duas anomalias, nas semanas 80 e 83.



Figura 36 – Análise de anomalias X00014

A análise do gráfico de linhas das vendas do artigo X00015 (figura 37) sugere que o artigo pode ter sazonalidade anual, devido ao padrão elíptico com valores mais altos nos meses de Verão. Podemos observar que este artigo é particularmente suscetível de auto-canibalização em períodos de campanha – o artigo tem um grande pico de vendas na semana 82, fruto de uma campanha promocional, seguido de uma forte quebra de vendas que apenas recupera no próximo período. O mesmo acontece na semana 118, em que as vendas do artigo caem durante as próximas 5 semanas. Suspeita-se que o facto de ser um artigo não perecível usado por profissionais leva a que os clientes aguardem por ações promocionais para fazer stock do artigo.



Figura 37 – Venda de 156 semanas do artigo X00015

O formato ondulado do gráfico de autocorrelação da figura 38, com picos a cada 52 semanas, suporta a presença de sazonalidade na série temporal. Os valores da autocorrelação vão caindo à medida que o desfasamento vai aumentando, o que sugere também a presença de tendência.

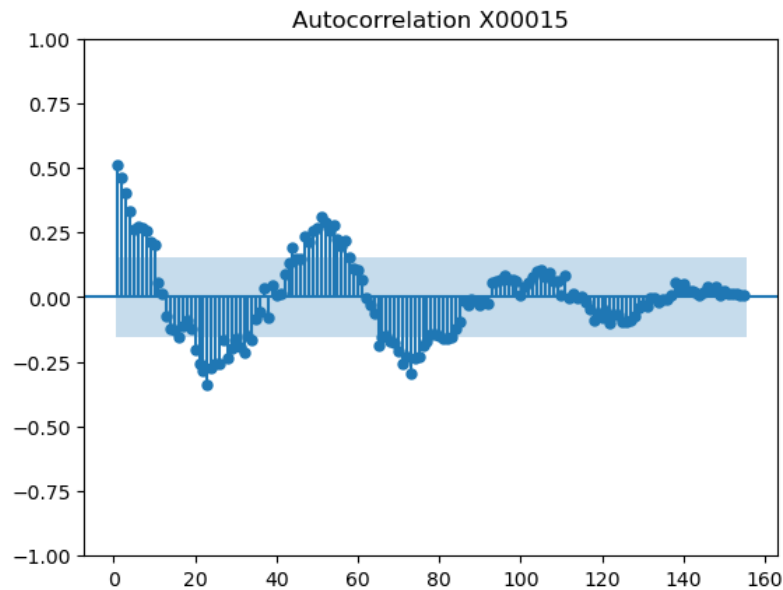


Figura 38 – Autocorrelação X00015

Na figura 39, podemos observar que a análise de anomalias usando o método MSTL para previsão apresenta apenas 2 períodos anómalos, entre a semana 82 e 87.

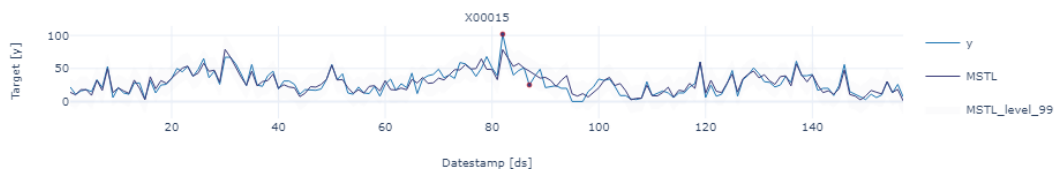


Figura 39 – Anomalias X00015

As vendas do artigo X00016 apresenta um formato semelhante ao artigo X00015, devido a ambos serem da família “Epilation”. Contudo, apresenta picos de venda menos acentuados, o que pode ser explicado por ser um produto com uma volumetria substancialmente maior, o que pode desincentivar os clientes a criarem stock próprio do artigo. Na figura 40, é igualmente perceptível uma sazonalidade anual.

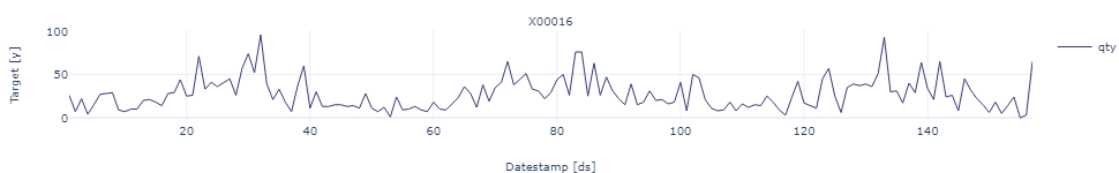


Figura 40 – Venda de 156 semanas do artigo X00016

Métodos e Aplicação

A análise do gráfico de autocorrelação na figura 41 suporta a presença de sazonalidade, e a redução dos valores de autocorrelação à medida que os desfasamentos se tornam maiores indica também a presença de tendência.

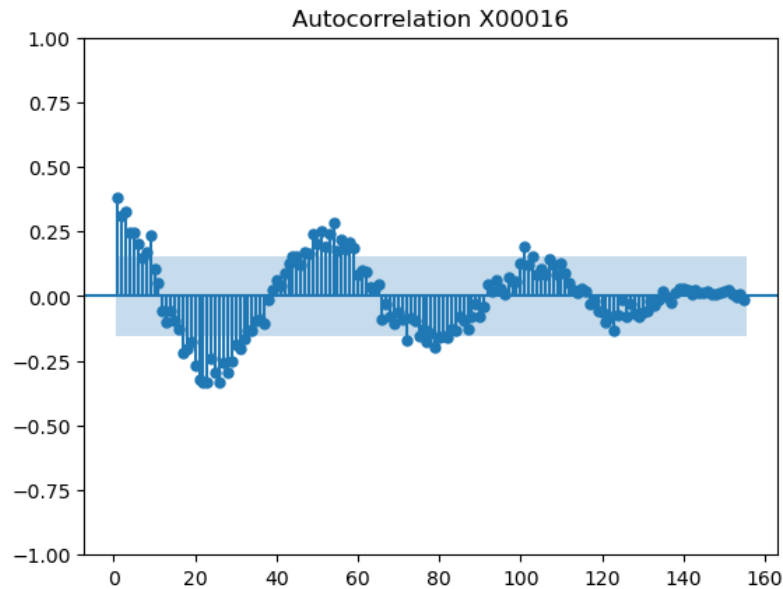


Figura 41 – Autocorrelação X00016

A análise de anomalias, que pode ser observada na figura 42, indica 3 períodos anómalos, nas semanas 87, 102 e 103.

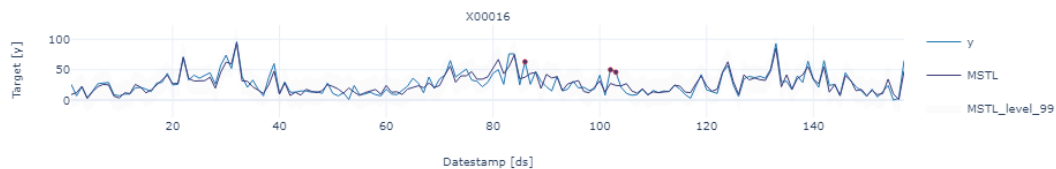


Figura 42 - Anomalias X00016

A análise do gráfico de vendas do artigo X00017 (figura 43) parece indicar sazonalidade e tendência nos primeiros 2 anos, devido ao formato curvo e ascendente da série temporal. No entanto, a partir da semana 115 as vendas do artigo têm vários picos de vendas seguidos de procura muito reduzida, o que pode indicar roturas intermitentes ou picos de venda em alturas de campanhas agressivas.

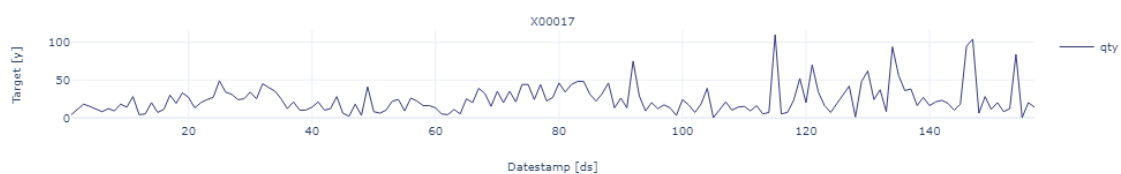


Figura 43 – Venda de 156 semanas do artigo X00017

A análise do gráfico de autocorrelação na figura 44 sugere que o artigo é composto apenas por ruído branco, visto mais de 95% dos valores estarem dentro do patamar $\pm 2/\sqrt{T}$.

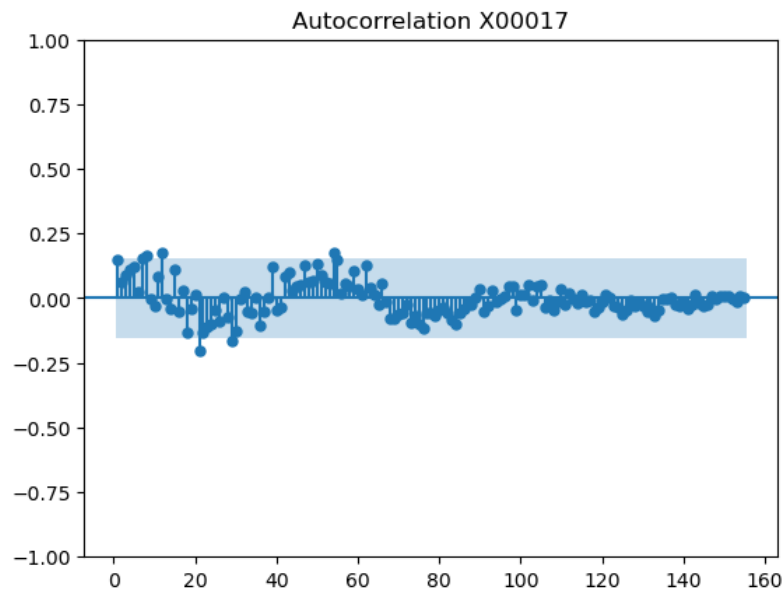


Figura 44 – Autocorrelação X00017

A análise de anomalias na figura 45 apresenta 4 períodos anómalos, nas semanas 62, 70, 90 e 94.



Figura 45 – Autocorrelação X00017

O artigo X00018 tem várias semanas com vendas zero, como podemos observar no gráfico da figura 46. Isto pode indicar períodos alargados de rotura, o que desvirtua a procura real do artigo – o último ano de vendas parece indicar uma sazonalidade anual com pico na altura de Verão.

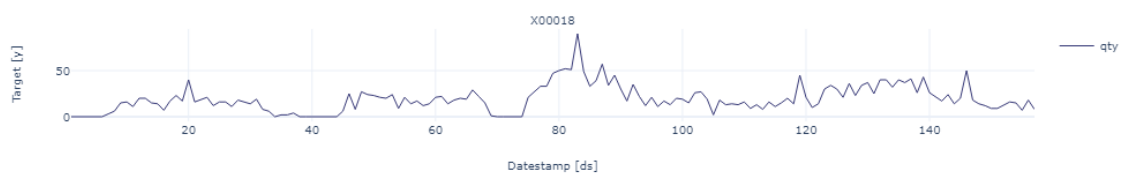


Figura 46 – Venda de 156 semanas do artigo X00018

Métodos e Aplicação

O gráfico na figura 47 sugere que a série temporal é composta por sazonalidade e tendência.

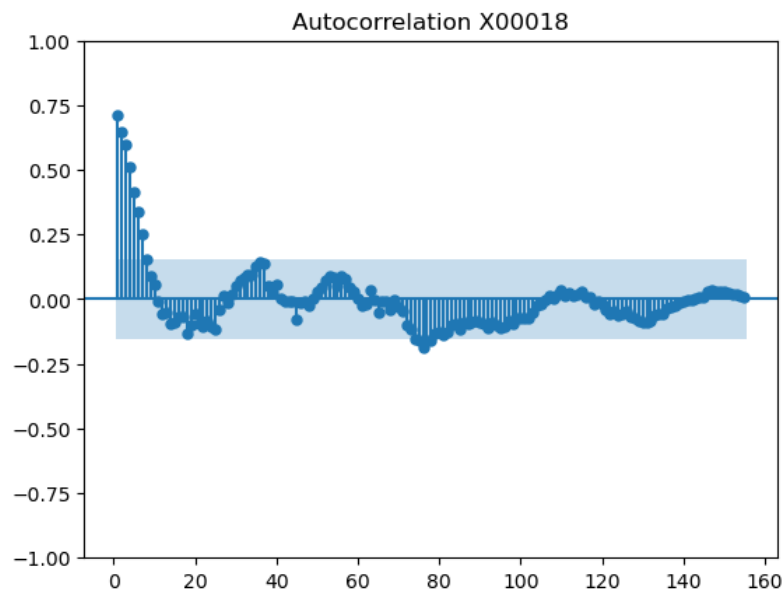


Figura 47 – Autocorrelação X00018

A análise de anomalias na figura 48 identifica 4 períodos anómalos, todos entre a semana 75 e 95, período em que as roturas regularizam.

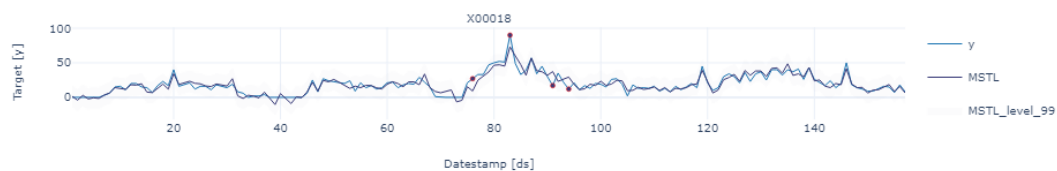


Figura 48 – Autocorrelação X00018

A análise gráfica do artigo X00019 mostra que o artigo tem uma procura regular, aparentes picos de venda a cada 20 semanas, o que pode indicar uma sazonalidade trimestral.

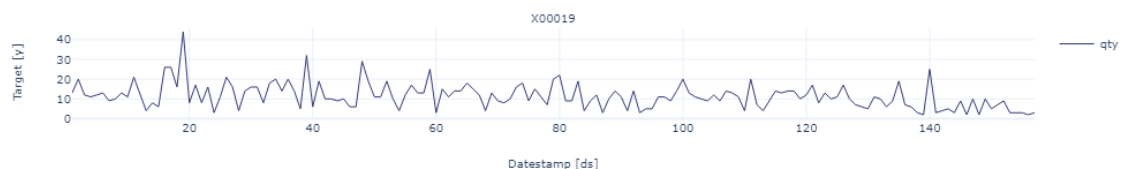


Figura 49 – Venda de 156 semanas do artigo X00019

No entanto, a análise do gráfico de autocorrelação na figura 50 sugere que a série temporal é composta por ruído branco.

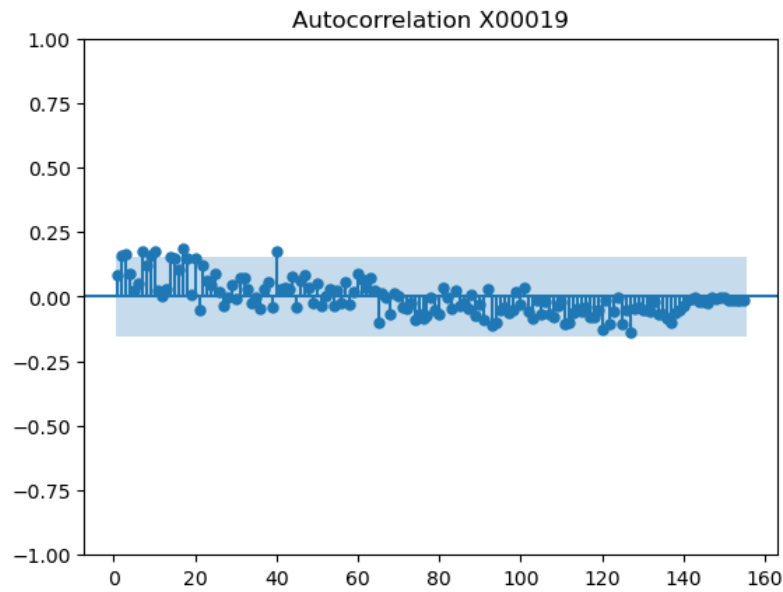


Figura 50 – Autocorrelação X00019

A análise de anomalias indica a presença de 3 períodos anómalos, como se pode observar na figura 51.

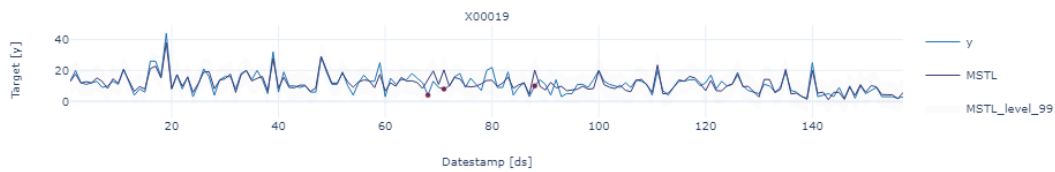


Figura 51 – Autocorrelação X00019

O gráfico de vendas do artigo X00020, na figura 52, mostra que o artigo tem uma procura constante ao longo do tempo, com picos de procura em intervalos irregulares. Não é perceptível sazonalidade ou tendência.

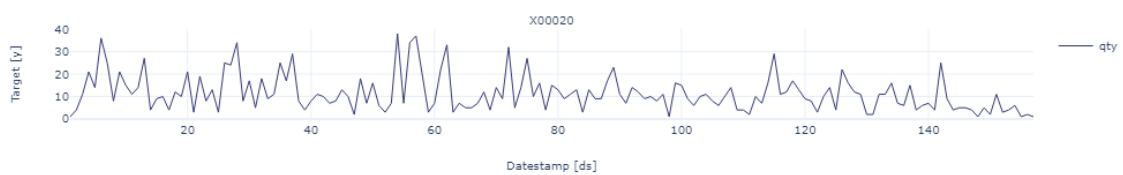


Figura 52 – Venda de 156 semanas do artigo X00020

O gráfico de autocorrelação na figura 53 suporta que a série temporal é composta por ruído branco, visto os valores estarem dentro do patamar $\pm 2/\sqrt{T}$.

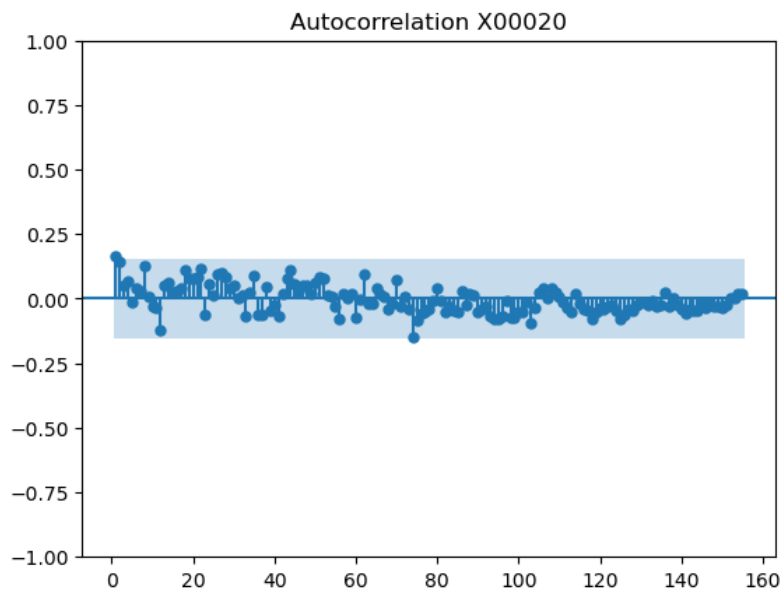


Figura 53 – Autocorrelação X00020

A análise de anomalias no gráfico da figura 54 apresenta 4 períodos anómalos.

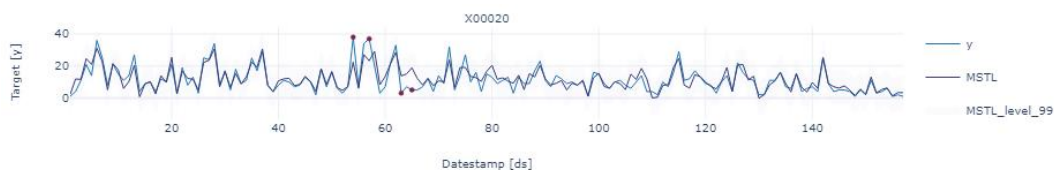


Figura 54 – Autocorrelação X00020

3.3.2. Tratamento de Outliers

Uma parte substancial dos *outliers* identificados podem ser explicados com campanhas promocionais organizadas pela empresa. Estas campanhas podem ser feitas de diferentes formas:

- Redução do preço do artigo;
- Promoção do artigo através de meios de marketing tradicionais ou digitais;
- Pagamento de comissões aos funcionários de loja na venda de determinado artigo ou marca.

Através dos dados disponíveis para o desenvolvimento deste trabalho, é apenas possível identificar momentos em que houve uma alteração no preço do artigo.

Com base na observação empírica de campanhas passadas, a empresa considera que uma ação promocional na semana n tem um efeito nas vendas desde a semana $n-1$ até à semana $n+2$. A hipótese estabelecida é que:

- Na semana $n-1$ a campanha é divulgada ao público, o que leva a que alguns clientes adiem a compra, o que reduz a procura;

- Na semana n existe um aumento substancial da procura, fruto da divulgação campanha e da relação conhecida entre o preço e a procura;
- Nas semanas $n+1$ e $n+2$ a procura volta a reduzir, devido aos clientes que anteciparam a compra para aproveitar a redução do preço e dos clientes que compraram uma quantidade superior ao habitual.

De forma a testar esta hipótese, procedeu-se à análise de períodos de venda em que o aumento da procura é coincidente com uma alteração do preço. Podemos observar o efeito da campanha promocional da semana 119, explicada no ponto 2., na procura do artigo X00001 na figura 55.



Figura 55 – Efeito de campanha promocional na procura do artigo X00001

Analisando o gráfico da figura 55, podemos observar que a quebra na procura na semana anterior à campanha é existente, mas que a normalização da procura é bem mais demorada do que na hipótese estabelecida pela empresa.

O artigo tem vendas estáveis com vendas entre 60 e 90 nas semanas anteriores à campanha, caindo para 36 na semana $n-1$. Na semana da campanha o artigo vende 229 unidades, tendo um efeito de contágio na seguinte, devido a reservas feitas na semana da feira que foram faturadas apenas na semana seguinte. Na semana $n+2$ o artigo vende apenas 38 unidades, registando uma subida para 54 unidades na semana $n+3$.

No sentido inverso, sabemos que uma rotura pode resultar em que a procura de um artigo não seja completamente satisfeita. Uma rotura pode ser do Centro de Distribuição, o que vai ter um efeito prolongado, ou apenas da loja, o que implica uma reposição rápida por parte do centro de distribuição. As roturas da loja podem ser causadas por uma venda acima do normal, por uma discrepância entre o stock físico e em sistema, ou por uma procura não prevista.

Na figura 56 podemos observar o efeito de uma rotura prolongada do artigo X00013 no centro de distribuição.

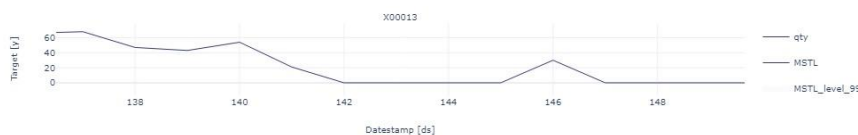


Figura 56 – rotura prolongada do artigo X00013

No caso dos algoritmos estatísticos, o aparecimento de um outlier nas vendas de uma semana pode ter um efeito significativo na previsão das semanas subsequentes, visto que a

Métodos e Aplicação

previsão de vendas de uma determinada semana tem em conta o comportamento histórico anterior. Portanto, pode ser desejável tratar estes outliers para que o seu efeito na previsão seja reduzido. No entanto, num contexto real, é desejável que um modelo de previsão seja capaz de responder a eventos previsíveis como uma campanha promocional, pelo que o tratamento dos *outliers* utilizando técnicas de alisamento podem desvirtuar o comportamento real da série temporal e melhorar o desempenho do modelo para lá do que seria esperado. Uma rotura prolongada, ou seja, uma rotura que não é causada por um aumento inesperado da procura do artigo no ponto de venda, mas sim por um problema no centro de distribuição, pode ter um impacto considerável no desempenho da previsão em modelos estatísticos. É, por isso, desejável tratar casos em que a procura de um artigo foi zero ou próxima de zero numa determinada semana por causa de uma rotura prolongada sejam alisados.

Para fazer esse alisamento, será aplicada a média dos dois períodos antecedentes e dos dois períodos subsequentes sem rotura. Esses dados corrigidos serão utilizados para as previsões dos modelos estatísticos e, posteriormente, comparados com os valores reais observados. O efeito desta correção no artigo X00013, o artigo mais afetado por roturas em 2022, pode ser observado na figura 57.

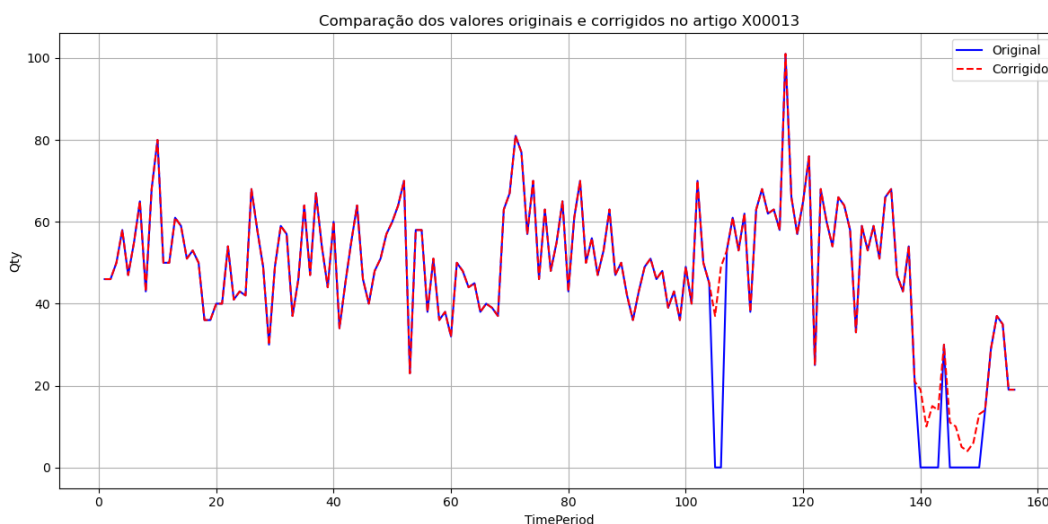


Figura 57 – Efeito da correção de roturas no artigo X00013

A figura 58 mostra a aplicação da correção no artigo X00016, que teve apenas um período de rotura.

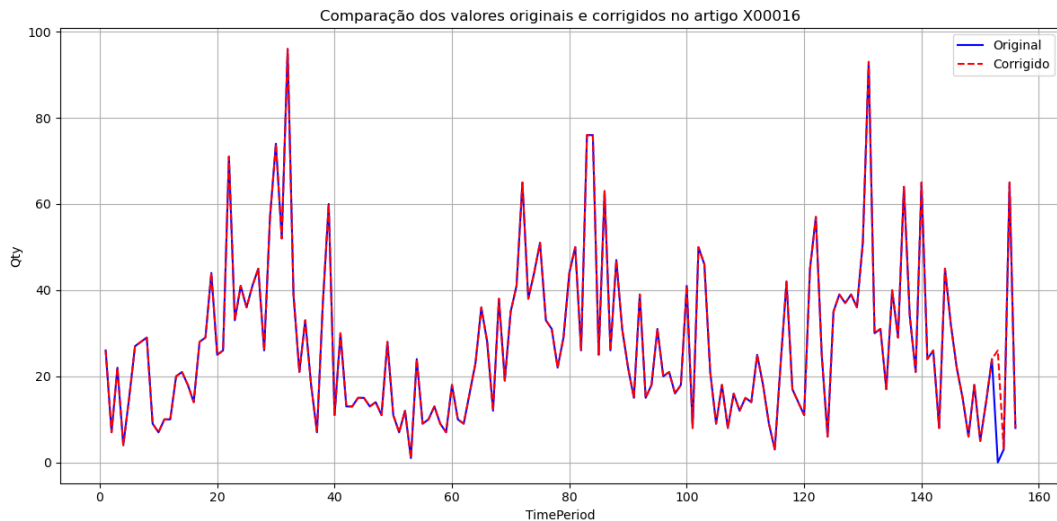


Figura 58 – Efeito da correção de roturas no artigo X00013

É possível fornecer parâmetros a um algoritmo de *machine learning* que indicam eventos que impactaram as vendas numa determinada semana. Como o preço de uma determinada semana é definido com antecedência, é possível usar esse valor tanto para o treino do algoritmo, como para os valores a prever. A rotura futura não é conhecida, no entanto, passando a informação de que um artigo teve rotura num determinado número de dias durante uma semana, o algoritmo pode reconhecer o padrão e ajustar as vendas em conformidade.

3.4. Seleção dos modelos de previsão

A seleção dos modelos de previsão a utilizar requer um conhecimento aprofundado dos dados a prever. A partir do trabalho desenvolvido no ponto 2.1., sabemos que os nossos dados contêm sazonalidade e tendência, ou seja, não contêm estacionariedade (Hyndman & Athanasopoulos, 2021).

A partir desse princípio, sabemos que métodos simples, que não têm em consideração os componentes da série temporal, não se adequam aos nossos dados. No entanto, estes métodos podem ser utilizados como Benchmark para comparar o seu desempenho com métodos mais avançados.

O código desenvolvido está disponível no Apêndice A.

3.4.1. Métodos para Benchmark

A aplicação de métodos amplamente estudados e validados pela academia, de aplicação simples, permite comparar o desempenho do nosso modelo contra modelos standards. O método ingénuo assume que o valor esperado do próximo ponto a prever é igual ao último valor observado. É o modelo mais simples de aplicar.

$$Y_t = F_{t-1}$$

O modelo Atual utiliza um nível de serviço variável para os artigos consoante a sua classificação ABC. Como vários dos artigos selecionados são artigos “A”, será utilizado o nível de

Métodos e Aplicação

serviço que a empresa aplica a estes artigos, que é de 2,8. O método para cálculo é apresentado abaixo:

$$F_{t+1} = \frac{\sum_{t=1}^T y_t}{T} + \sigma * z$$

F_{t+1} = Previsão para o momento $t + 1$

y_t = Quantidade vendida no momento t

T = Número de dias (30)

σ = Desvio Padrão de 30 dias

z = nível de serviço

3.4.2. Alisamento Exponencial

Os modelos de Alisamento Exponencial são frequentemente usados para a previsão de séries temporais. O modelo de Alisamento Exponencial Simples é o de aplicação mais simples, em que a previsão F_t é calculada com base numa média ponderada entre a última procura Y_{t-1} e a última previsão F_{t-1} , usando α , que varia entre 0 e 1, como fator de ponderação:

$$F_{t+h} = \alpha * Y_t + (1 - \alpha) * F_t$$

O valor de α deve ser definido empiricamente, no entanto, é possível utilizar um algoritmo para minimizar o RMSE (raiz do erro médio quadrático) nas previsões passadas.

O Método de Holt é uma variação deste método que inclui um parâmetro β , para representar a tendência da série temporal (Hyndman & Athanasopoulos, 2021):

$$F_{t+h} = L_t + b_t * h$$

$$L_t = \alpha * Y_t + (1 - \alpha) * (L_{t-1} + b_{t-1})$$

$$b_t = \beta * (L_t - L_{t-1}) + (1 - \beta) * b_{t-1}$$

Novamente, é possível calcular β utilizando valores passados conhecidos para fazer previsões e minimizar o RMSE.

O Método de Holt-Winters acrescenta uma nova componente para tratar a sazonalidade, γ , onde m especifica a frequência da sazonalidade. Existem duas versões principais deste método, o método aditivo, onde a sazonalidade é relativamente constante e a sua componente é adicionada à tendência:

$$F_{t+h} = L_t + b_t * h + s_{t+h-m(k+1)}$$

$$L_t = \alpha * (Y_t - s_{t-m}) + (1 - \alpha) * (L_{t-1} + b_{t-1})$$

$$b_t = \beta * (L_t - L_{t-1}) + (1 - \beta) * b_{t-1}$$

$$s_t = \gamma * (Y_t - L_{t-1} - b_{t-1}) + (1 - \gamma) * s_{t-m}$$

E a multiplicativa, em que a variação sazonal é proporcional à mudança de nível da série, pelo que a sua componente é multiplicada pela tendência:

$$F_{t+h} = (L_t + b_t * h) * s_{t+h-m(k+1)}$$

$$L_t = \alpha * \frac{Y_t}{s_{t-m}} + (1 - \alpha) * (L_{t-1} + b_{t-1})$$

$$b_t = \beta * (L_t - L_{t-1}) + (1 - \beta) * b_{t-1}$$

$$s_t = \gamma * \frac{Y_t}{L_{t-1} + b_{t-1}} + (1 - \gamma) * s_{t-m}$$

Tal como no método simples e aditivo, é possível otimizar os parâmetros γ , β e α minimizando o RMSE em valores passados.

3.4.3. Método ARIMA

Como descrito no ponto 2.2.1., o método ARIMA considera auto-regressão e a média móvel para encontrar a previsão. É necessário especificar os seguintes parâmetros:

- p – Ordem da parte Autorregressiva;
- d – Grau da primeira diferenciação envolvida;
- q – Ordem da parte da média móvel

É necessário selecionar os valores adequados para cada parâmetro para a série temporal que estamos a tentar prever, o que requer uma análise técnica. No entanto, é possível fazer uma abstração do desempenho do modelo minimizando o AIC (Akaike Information Criterion), variando os valores de p e q para encontrar o modelo mais adequado (Hyndman & Athanasopoulos, 2021).

O modelo SARIMA (Seasonal ARIMA) utiliza um padrão adicional para enquadrar a componente de sazonalidade no modelo. Os valores de P , D , e Q têm o mesmo propósito que no modelo anterior, mas aplicam-se apenas à componente sazonal, e m denota o *lag* sazonal. Como os nossos dados são divididos em semanas e verificou-se no ponto 2.1. que algumas das séries temporais apresentam indícios de sazonalidade anual, iremos utilizar o valor de m de 52.

3.4.4. MultiLayer Perceptron

MultiLayer Perceptron (MLP) é um algoritmo de Machine Learning que utiliza uma rede neural dividida em três camadas: uma primeira camada recebe os *inputs*, várias camadas escondidas processam os dados, e a última camada dá o *output*. A figura 59 apresenta uma visualização deste processo.

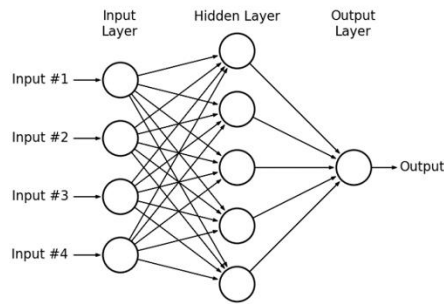


Figura 59 – Rede Neural de um algoritmo MultiLayer Perceptron

Os valores são processados em neurónios, em que o valor de output de cada camada é fornecido pela camada anterior pela seguinte equação:

$$\alpha = \varphi * \left(\sum_j w_j * x_j + b \right)$$

Onde x_j são os *inputs* do neurónio, w_j é o peso, b é o *bias* e φ é a função não linear de ativação, que fornece a não-linearidade necessária, o que permite que a rede “aprenda” relações complexas entre os dados. A função ReLU (Rectified Linear Unit), tem-se tornado popular em redes neurais profundas, pois apresenta propriedades que auxiliam na aceleração da convergência durante o treino.

MLP é uma rede neural *feed-forward*, em que os neurónios estão dispostos por camadas e em que cada camada tem o mesmo número de neurónios. A velocidade de treino e o risco de *overfitting* são os principais problemas destes modelos (Dudek, 2016). Na fase de *feed-forward*, é calculado o MSE do modelo, enquanto na fase de *backpropagation* os pesos são calculados com o valor do gradiente:

$$\Delta_w(t) = -b * \frac{dE}{dw_e} + \alpha \Delta_w(t-1)$$

Onde dE é o erro, dw é o peso e α é o grau de aprendizagem.

Para assegurar que o modelo está dentro de controlo e não entra em *overfitting*, é necessário ajustar os Hiperparâmetros. É possível controlar o número de amostras que o modelo processa antes de ajustar os pesos dos neurónios, e controlar o número de *epochs*, ou seja, passagens completas que o programa faz antes de finalizar o treino do modelo.

O uso deste modelo é facilitado pela biblioteca *Keras*, que permite a implementação de um modelo de redes neuronais *feed forward*. Esta implementação revelou ter um desempenho mais eficiente e com menor erro do que outros modelos de *machine learning* testados na fase de implementação.

3.4.4.1. Preparação dos dados

Os dados de vendas foram obtidos através de uma *query* a uma base de dados SQL Server. Para garantir que os valores nulos fossem representados como 0, a tabela principal foi a de um calendário, fazendo o *join* à tabela de vendas, que contém, entre outros, a informação da data de venda, a loja, o artigo, a quantidade e o valor. Foi também efetuado um *join* à tabela de artigos, para retirar a informação da família e subfamília do artigo. Os dados foram agrupados por ano, semana, utilizando a função *weekofyear()* nativa ao SQL Server, que determina a semana de acordo com a norma ISO 8601, e por artigo, efetuando a soma da quantidade vendida e a divisão do valor total de vendas pela quantidade, de forma a obter o *pricing*.

O resultado foi uma tabela com a seguinte informação:

- Ano;
- Semana;
- Família ;
- SubFamília;
- Loja;
- SKU;
- Qtd;
- Pricing;

Para obter a informação sobre se um determinado artigo esteve em rotura numa determinada semana, foi necessário fazer uma reconstrução do histórico de movimentos de stock de um determinado artigo na loja. Para isso, foi efetuada uma query a uma tabela que regista todos os movimentos de stock da organização, como transferências, vendas, devoluções ou acertos de stock. A soma destes movimentos foi agrupado por dia, por loja e por artigo. De seguida foi utilizado o stock atual como ponto de partida e feita uma recriação do stock teórico, sendo considerado como rotura os momentos em que um determinado artigo chega ao final do dia em rotura numa determinada loja.

Esta informação foi agrupada por ano, semana, loja e artigo, fornecendo assim o número de dias que um determinado artigo esteve em rotura numa determinada loja, numa determinada semana.

A tabela resultante foi a seguinte:

- Ano;
- Semana;
- Loja;
- SKU;
- DiasRotura;

Existem vários dados que podem ter impacto na previsão de vendas, como informação sobre a economia, preço dos combustíveis, previsão meteorológica ou datas especiais (Ma & Fildes, 2021). A competição M5, que utilizou dados de vendas de lojas da Walmart em 4 Estados, teve em consideração feriados nacionais, culturais, religiosos ou desportivos (Theodorou et al., 2022).

Métodos e Aplicação

Para incluir dados meteorológicos, foram exportados do site do SNIRH (Sistema Nacional de Informação de Recursos Hídricos) os dados históricos de precipitação da estação meteorológica “Linhó (21ª/09G)” e os dados de temperatura do ar média em graus Celsius da estação do Monte da Caparica. Para serem apresentados por semana, foi calculada a precipitação total por semana e a temperatura média, máxima e mínima da semana.

A informação de datas especiais, tal como feriados religiosos ou nacionais, datas culturais que influenciam o consumo como o Dia da Mãe, Dia do Pai ou o Dia dos Namorados, ou de eventos desportivos como as finais de grandes competições como o mundial, europeu, liga dos campeões ou a Taça de Portugal, são apresentados como o número de eventos numa determinada semana. Devido ao impacto do Natal, essa informação é prestada numa coluna à parte.

Para alimentar os modelos de *Machine Learning*, foi criada uma abstração de cada SKU, Família e SubFamília. Para esse fim, foi criada uma coluna binária com a informação de cada valor único dessas colunas.

Adicionalmente, pretende-se dar informação ao modelo sobre a sazonalidade e tendência da série temporal. O método STL (Decomposição de Tendência e Sazonalidade usando Loess) permite decompor séries temporais, enquanto Loess permite estimar relações não lineares (Hyndman & Athanasopoulos, 2021).

Apesar de ter várias vantagens sobre métodos de decomposição clássicos, como a possibilidade de implementar qualquer tipo de sazonalidade, ser possível alterar a sazonalidade ao longo do tempo, permitir ao analista controlar a suavidade do ciclo de tendência e ser robusto a outliers, também apresenta algumas desvantagens – não lida com variações do calendário de forma automática e apenas permite decomposições aditivas (Hyndman & Athanasopoulos, 2021).

O método STL permite recolher informações sobre a sazonalidade, tendência e o resíduo, ou seja, o ruído da série temporal após removermos esses componentes. Na figura 60, podemos observar o resultado da aplicação do método STL ao artigo X00003, indicando uma sazonalidade de 52 semanas.

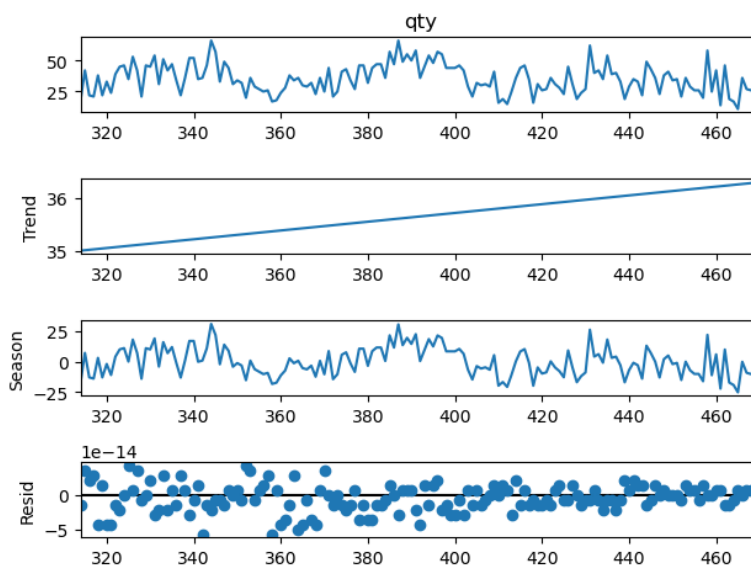


Figura 60 – Aplicação do método STL ao artigo X00003

Para alimentar o modelo com estes dados, é possível utilizar um método de previsão estatístico para prever os valores dentro do horizonte de previsão e utilizar o método STL para decompor a série temporal com os novos valores estimados. O processo seguirá os seguintes passos:

- Por artigo, utilizar Amortecimento Exponencial Simples para prever os valores do nosso horizonte de previsão;
- Aplicar o método STL – como sabemos que existe sazonalidade anual nalguns artigos, usa-se a sazonalidade de 51;
- Agregar os valores de Sazonalidade, Tendência e Resíduo aos nossos dados para treino;

Desta forma, utilizamos os valores previstos pelo melhor método estatístico, que será selecionado com base nos resultados dos testes a efetuar, para estimar como os componentes da série temporal a prever afetam os pontos que pretendemos prever.

Um passo crítico na criação de um modelo de Redes Neurais para previsão de séries temporais é o número de *lags*, ou seja, o valor observado da nossa variável a prever em períodos anteriores, a serem implementadas no algoritmo. Esse número pode ser estimado utilizando técnicas como funções de autocorrelação (Surakhi et al., 2021). A utilização de *lags* permite ao modelo interpretar os componentes da série temporal. No entanto, no nosso modelo pretendemos que essa informação seja obtida através dos valores gerados pelo método STL. Assim, foi aplicado um *lag* temporal de 4 semanas, de forma a fornecer ao modelo a informação aproximada da quantidade vendida nos períodos mais recentes, mas que a evolução da procura seja abstraída a partir dos valores de sazonalidade, tendência e de resíduo estimados.

Como para o trabalho presente apenas está a ser utilizada uma loja para análise, as colunas de *StoreId*, *StoreLocation* e *StoreType* são retiradas, visto serem redundantes. A coluna Total também é removida, visto proporcionar informação ao modelo sobre o futuro.

O resultado é uma tabela com 49 colunas e 2983 linhas.

3.4.4.2. Teste e validação do modelo

Para testar o ajuste do algoritmo aos nossos dados, foi criada uma versão de teste com a função não linear *ReLU*. Os dados foram normalizados para alimentar o algoritmo, em que o valor normalizado z da amostra x é calculado através de $z = (x - u)/s$, em que u é a média das amostras de treino e s o desvio padrão da amostra.

O número de *epochs* foi definido em 300, sendo criada uma função de paragem precoce caso o modelo não consiga melhorar o MSE após 25 iterações.

Foram feitos testes com amostras de 16, 32 e 124, sendo que a convergência foi significativamente mais rápida com 32.

	Nº de Amostras			
	16	32	64	128
Iteração 1	9,55	7,25	7,68	8,74
Iteração 2	9,46	6,93	6,84	8,41
Iteração 3	3,93	6,29	8,66	10,59
Iteração 4	12,49	8,71	11,27	9,00
Iteração 5	9,35	7,61	8,52	34,00
Média	8,96	7,36	8,59	11,71

Tabela 6 – Tempo de computação em segundos com diferente número de amostras

Avaliando a perda do modelo, verificámos que o erro do treino e de validação seguem a mesma linha. Se o modelo estivesse em *underfitting*, seria esperado que tanto o erro de validação como o de treino se mantivessem altos, enquanto se o modelo estivesse em *overfitting*, seria esperado que o erro do modelo de validação aumentasse no final da cauda, pelo que o modelo não apresenta indícios de estar em *overfitting* ou *underfitting*. A figura 61 ilustra a progressão do erro de validação e de treino.

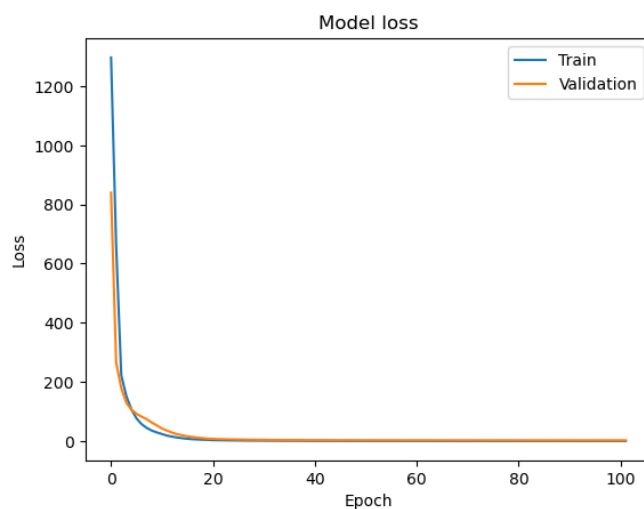


Figura 61 – Evolução do erro de treino e de validação

4. Avaliação de resultados

Ao testar um modelo de previsão de uma série temporal é necessário ser particularmente cuidadoso com a divisão dos dados, para garantir que:

- Não fornecemos informação ao modelo que não estaria disponível na altura da previsão;
- A amostra selecionada para análise é representativa da evolução da série temporal.

A informação do *pricing* praticado na próxima semana pode ser conhecida pela organização, mas não o número de recibos numa determinada loja, devido à natureza estocástica da procura.

Da mesma forma, ao selecionarmos uma amostra aleatória de uma série não estacionária, que tenha presente os componentes de sazonalidade e de tendência, podemos enviesar as nossas amostras de treino e de validação com dados que não sejam representativos da natureza da série temporal.

Uma abordagem que permite a divisão de uma série temporal, assegurando a manutenção dos seus componentes, é a *Rolling Window Time Split*, ou Janela Deslizante (Zivot & Wang, 2006). Para aplicar esta técnica, devemos selecionar o nosso horizonte de previsão, h , e o tamanho da nossa amostra, m . Assumindo que a nossa série temporal tem o tamanho T , ao fazermos a previsão para o nosso primeiro horizonte, avançamos a nossa amostra para o próximo período. Esta abordagem é representada na figura 62.

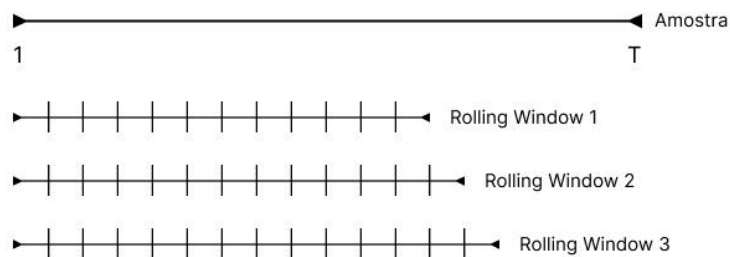


Figura 62 – Representação de uma Janela Deslizante

A seleção do horizonte temporal tem um impacto significativo no desempenho do modelo. Modelos mais complexos podem ter um bom desempenho com horizontes curtos, mas ao abstrair os dados para um horizonte mais alargado, distorcer os dados de forma significativa.

Estando a trabalhar um caso real que tem como objetivo melhorar a alocação de stock a um ponto de venda em que a reposição é feita num máximo de 4 dias, não é necessário que o nosso horizonte de previsão seja superior a 2 semanas.

Como na secção 2.1. foram identificados vários artigos com sazonalidade anual, pretende-se verificar como o modelo responde às flutuações anuais. Para esse efeito, iremos utilizar o último ano para previsão, tendo assim 52 pontos de previsão, em que em cada ponto se prevê as próximas 2 semanas com base em todo o histórico anterior.

Para comparar o erro na previsão, será usado o método *RMSE*, que mede a raiz do erro médio quadrático. A expressão para calcular o *RMSE* é a seguinte:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - p_i)^2}$$

Em que n é o número de previsões, y_i é o valor observado e p_i o valor previsto.

Avaliação de resultados

O método *RMSE* apresenta várias vantagens sobre outros métodos. Ao elevar o erro ao quadrado, todos os erros assumem um valor positivo, além de penalizarmos os erros mais elevados.

A tabela 7 apresenta o *RMSE* médio de cada modelo, para cada artigo, na previsão de vendas das 52 semanas de 2022.

SKU	Valor Médio	Valor Max	Valor Min	Naive	Modelo Atual	SES	Holt	Holt Winters	Arima	SARIMA	MLP
X00001	64,2	229	24	33,8	70,0	22,8	24,8	24,0	23,3	23,7	20,9
X00002	34,4	108	15	12,7	31,7	11,0	10,6	11,5	11,5	12,3	9,8
X00003	32,6	62	11	12,2	27,5	9,5	9,6	11,3	10,0	9,8	8,6
X00004	19,5	49	7	8,6	20,5	7,5	7,5	7,8	8,1	7,3	6,5
X00005	34	82	9	11,7	22,6	12,2	12,6	14,4	11,3	11,3	11,1
X00006	15,7	92	1	14,6	32,6	11,4	12,4	12,6	9,9	15,7	10,3
X00007	41,8	283	11	38,2	79,9	22,8	24,3	26,1	22,7	46,0	19,9
X00008	39,7	262	9	35,5	84,4	23,9	25,4	26,9	24,5	24,5	21,3
X00009	35	228	9	30,8	70,7	20,2	21,8	22,9	19,6	39,1	18,2
X00010	28,1	176	3	23,6	55,6	15,5	17,8	18,8	15,4	30,8	13,6
X00012	41,4	140	6	26,4	60,5	23,5	23,4	30,4	24,0	23,2	21,2
X00013	40	101	0	12,5	30,2	12,9	12,9	13,6	13,0	13,0	14,9
X00014	58,4	134	2	29,7	53,0	23,0	23,3	25,3	20,2	20,2	19,3
X00015	22,5	61	3	14,7	31,9	13,2	13,3	12,9	13,2	12,4	12,1
X00016	26,2	93	0	17,2	38,7	15,2	15,3	15,4	16,1	16,3	14,4
X00017	28,8	110	0	29,1	65,7	22,5	22,5	22,6	24,3	23,0	19,8
X00018	22,1	50	7	9,4	20,2	8,6	8,7	9,7	8,3	8,3	14,7
X00019	11,6	44	3	8,4	15,7	5,5	5,5	6,7	5,5	5,7	10,4
X00020	5,1	12	1	4,3	8,7	4,7	4,3	6,7	4,2	4,2	4,8
RMSE Médio				19,7	43,2	15,0	15,6	16,8	15,0	18,3	14,3

Tabela 7 – Resultados obtidos pelos modelos de previsão

Verifica-se que o Modelo Atual é o que tem o RMSE mais alto em todos os artigos, o que revela que não se adapta corretamente à realidade da empresa. Uma justificação deste resultado é que o modelo é excessivamente agressivo na sua previsão, o que aumenta o nível de inventário dos pontos de venda, o que permite colmatar as falhas do modelo. É o único modelo com um RMSE superior ao método Ingénuo.

Na figura 63 podemos observar o comportamento do modelo na previsão do artigo X00012, que sabemos que contém sazonalidade e tendência. Podemos observar que o modelo não capta corretamente os componentes da série temporal, o que leva a uma previsão abaixo do real no início do período sazonal e uma resposta demorada ao seu fim.

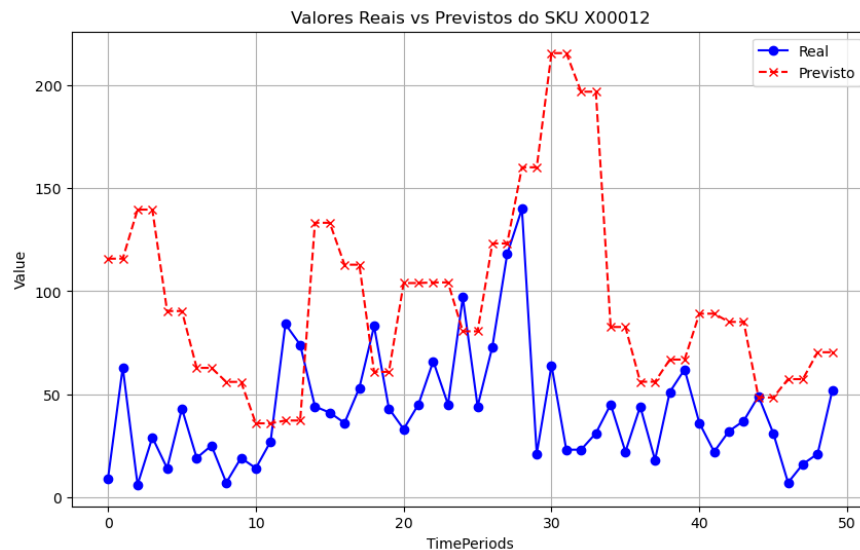


Figura 63 – Aplicação do método Atual ao artigo X00012

Dentro dos modelos de Amortecimento Exponencial, o Amortecimento Exponencial Simples teve o melhor desempenho em praticamente todos os artigos. Surpreendentemente, tem um desempenho superior ao Método de Holt-Winters nos artigos em que é identificada sazonalidade, o que sugere que a introdução da componente de sazonalidade acrescenta uma complexidade excessiva ao modelo, o que faz com que se não adapte aos outros componentes. O Método de Holt tem um desempenho semelhante ao do Amortecimento Exponencial Simples, tendo um erro comparável.

O método SARIMA apresenta um bom desempenho na generalidade dos artigos com sazonalidade, no entanto, a introdução de parâmetros extras nos artigos compostos por ruído branco desenquadra o modelo da evolução da série temporal, o que faz com que seja o modelo com pior desempenho. No entanto, o método ARIMA apresenta o melhor RMSE (excluindo o método MLP), apresentando resultados comparáveis ao método de Amortecimento Exponencial Simples em quase todos os artigos.

O modelo MLP apresenta o RMSE mais baixo, com um valor de 14,3 que compara com 15 do segundo melhor método. Isto demonstra que o modelo foi capaz de captar melhor a curva da procura real. A figura 64 demonstra os valores reais contra os previstos pelo modelo para o artigo X00003, onde foi o modelo com a melhor performance. Podemos observar que apesar de a evolução da curva estar próxima da procura real, o valor previsto fica muito aquém do valor real quando existem campanhas promocionais, ou seja, subestima o impacto de uma campanha promocional.

Avaliação de resultados

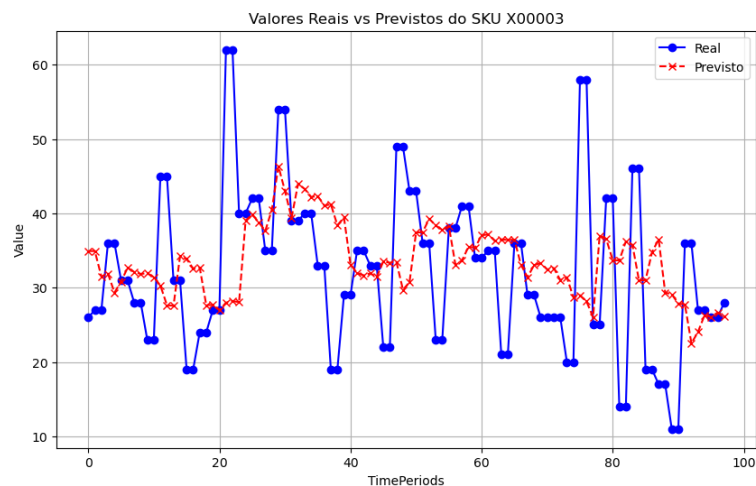


Figura 64 – Real vs Previsto pelo modelo MLP para o artigo X00003

O modelo, no entanto, teve o segundo pior desempenho nos artigos X00013, X00018 e X00019, ficando apenas à frente do Modelo Atual. A figura 65 mostra o desempenho do modelo a prever as vendas do artigo X00018, onde teve o pior desempenho. Podemos observar que para este artigo em específico, o modelo prevê sempre uma procura inferior ao real. Devido à natureza dos algoritmos de rede neuronal, não é possível verificar o motivo, no entanto, poderá estar relacionada com a evolução da procura de outros artigos da mesma família ou subfamília.

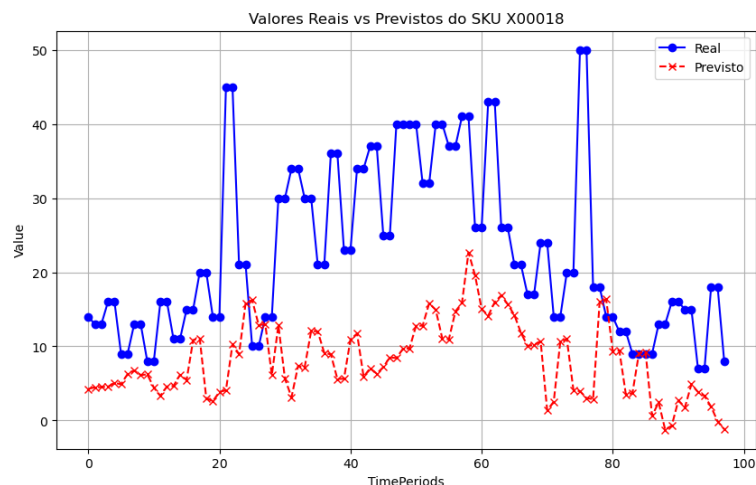


Figura 65 – Real vs Previsto pelo modelo MLP para o artigo X00018

Para a verificar a robustez e replicabilidade dos resultados, foram corridas várias iterações do algoritmo, de forma a verificar a consistência nos resultados. Como se pode observar na tabela X, os resultados obtidos foram comparáveis. Foi também calculado o tempo de treino médio do modelo para uma previsão, utilizando um Processador 11th Gen Intel Core(TM) i5-11400F @ 2.60GHz.

Componentes Identificados	SKU	Iteração1	Iteração2	Iteração3	Iteração4	Iteração5	Média	Desvio P.
Ruído Branco	X00001	20,6	20,9	20,7	20,8	20,8	20,8	0,1
Ruído Branco	X00002	9,7	9,8	9,7	9,7	9,8	9,7	0,1
Sazonalidade	X00003	8,6	8,6	8,6	8,7	8,6	8,6	0,1
Sazonalidade	X00004	6,5	6,5	6,6	6,6	6,7	6,6	0,1
Ruído Branco	X00005	11,0	11,1	11,0	10,9	11,0	11,0	0,1
Ruído Branco	X00006	10,5	10,3	10,5	10,6	10,5	10,5	0,1
Ruído Branco	X00007	20,0	19,9	20,1	19,9	20,1	20,0	0,1
Ruído Branco	X00008	21,2	21,3	21,1	21,2	21,0	21,2	0,1
Ruído Branco	X00009	18,4	18,2	18,2	18,1	18,2	18,2	0,1
Ruído Branco	X00010	13,5	13,6	13,5	13,6	13,7	13,6	0,1
Sazonalidade / Tendência	X00012	21,3	21,2	21,5	21,4	21,3	21,3	0,1
Sazonalidade / Ruído	X00013	14,9	14,9	14,9	15,0	14,9	14,9	0,0
Tendência	X00014	19,7	19,3	19,4	19,3	19,7	19,5	0,2
Sazonalidade / Tendência	X00015	12,0	12,1	12,0	12,0	12,3	12,1	0,1
Sazonalidade / Tendência	X00016	14,2	14,4	14,1	14,2	14,0	14,2	0,1
Ruído Branco	X00017	19,2	19,8	19,2	19,5	19,5	19,4	0,2
Ruído Branco	X00018	15,1	14,7	15,3	14,0	14,7	14,8	0,4
Ruído Branco	X00019	9,7	10,4	10,2	10,1	10,1	10,1	0,2
Ruído Branco	X00020	5,0	4,8	4,9	4,8	5,0	4,9	0,1
RMSE Médio		14,3	14,3	14,3	14,2	14,3	14,3	0,0
Tempo médio por previsão (s)		5,7	5,71	5,73	5,68	5,72	5,7	5,71

Tabela 8 – Iterações do modelo

5. Conclusões

A previsão da procura de um determinado artigo, num determinado local, tem um papel fulcral numa cadeia de retalho. Subestimar a procura leva a potenciais vendas perdidas, insatisfação no cliente, e lucro perdido. Se a procura for sobrestimada, pode causar excesso de stock num determinado ponto de venda, o que impacta o capital da empresa, gera custos de armazenagem ou de realocização do produto e, se o mesmo for perecível, perdas diretas.

A natureza estocástica da procura torna a previsão da quantidade vendida de um artigo num período específico um problema difícil de resolver. A utilização de técnicas estatísticas que procuram estimar o comportamento futuro da procura através do estudo dos componentes da série temporal não permite captar com precisão todas as variáveis que influenciam a procura, o que frequentemente leva a ajustes na procura estimada com base em eventos futuros já conhecidos.

A realização de uma campanha promocional, um evento desportivo ou meteorológico, ou até a rotura de outro artigo comparável no mesmo ponto de venda levam a flutuações na procura que não são captadas pela decomposição de uma série temporal, e são interpretados como “ruído branco”. No entanto, a aplicação de técnicas de *machine learning* permitem criar uma abstração desta realidade e captar os padrões escondidos entre várias variáveis.

O potencial do uso de algoritmos de *Machine Learning* para previsão é demonstrado neste trabalho através do uso do modelo MLP. O resultado obtido pelo modelo foi melhor do que o obtido pelos métodos clássicos de previsão, como o método de Holt-winters ou o método ARIMA, em 15 dos 19 artigos analisados, pelo que podemos concluir que um modelo de previsão com *machine learning* é capaz de ter um desempenho superior aos modelos estatísticos.

Foi também demonstrado que o método atualmente utilizado pela empresa não é adequado para a sua realidade, obtendo piores resultados do que a aplicação do método ingénuo. No entanto, o fraco desempenho do modelo atual na previsão de vendas é escondido pelo excesso de stock gerado pela previsão. A implementação de um novo modelo de previsão, enquadrado dentro de um modelo de gestão de stocks, permitiria à empresa uma redução do valor em stock em cada ponto de venda e melhorar a resposta a eventos como campanhas promocionais, roturas ou alturas sazonais, o que melhoraria o nível de serviço ao cliente.

Desta forma, podemos concluir que a implementação do modelo MLP pode ter um impacto significativo na gestão de stocks da empresa, permitindo simultaneamente aumentar o nível de serviço ao cliente e diminuir o valor de stock num ponto de venda.

5.1. Trabalho Futuro

A aplicabilidade do modelo para previsão das mais de 100 lojas e mais de 7 000 SKU's terá de ser estudada. Teoricamente, o modelo poderá obter melhores resultados sendo treinado com toda a informação da empresa, no entanto, é necessário verificar se é viável treinar o modelo em tempo útil para gerar previsões.

Simultaneamente, serão estudados outros métodos de *machine learning*, mais recentes, de forma a comparar resultados com o algoritmo utilizado. O método LightGBM (Light Gradient

Boosting Machine) tem sido aplicado com sucesso em vários modelos de previsão, sendo inclusivamente usado pelo modelo vencedor da competição *M5* (Theodorou et al., 2022).

Apesar de o método desenvolvido obter melhores resultados que os métodos tradicionais testados, é de realçar que os valores de RMSE encontram-se perto da média de quantidades vendidas por semana, o que indica que exista uma margem larga para melhoria. Através da comparação entre os valores previstos e os reais, podemos verificar que o modelo não está a adaptar corretamente as campanhas promocionais, o que indica que o pricing não é uma variável suficiente para o ajustamento do modelo. Outras variáveis devem ser estudadas, como a quantidade de menções que um determinado produto tem nas redes sociais, a exposição do artigo num local de destaque ou se está a ser dado algum incentivo à equipa de loja para vender o artigo.

Colocando em produção uma versão do modelo, será também aplicado um modelo de alocação de stock a ponto de venda com base no algoritmo de previsão de vendas, tendo em conta, além da procura prevista, o balanceamento da carga de trabalho do armazém e a redução do valor de stock em loja.

6. Bibliografia

- Achabal, D. D., McIntyre, S. H., Smith, S. A., & Kalyanam, K. (2000). A Decision Support System for Vendor Managed Inventory. *Journal of Retailing*, 76(4), 430–454.
- Achabal, D. D., McIntyre, S., & Smith, S. A. (1990). Maximizing profits from periodic department store promotions. In *From: Journal of Retailing* (Vol. 66, Issue 4).
- Agrawal, N., & Smith, S. A. (2013). Optimal inventory management for a retail chain with diverse store demands. *European Journal of Operational Research*, 225(3), 393–403. <https://doi.org/10.1016/j.ejor.2012.10.006>
- Annie Rose Nirmala, D., Kannan, V., Thanalakshmi, M., Joe Patrick Gnanaraj, S., & Appadurai, M. (2022). Inventory management and control system using ABC and VED analysis. *Materials Today: Proceedings*, 60, 922–925. <https://doi.org/10.1016/j.matpr.2021.10.315>
- Awad, M., & Khanna, R. (2015). *Efficient Machine Learning*.
- Azodi, C. B., Tang, J., & Shiu, S. H. (2020). Opening the Black Box: Interpretable Machine Learning for Geneticists. In *Trends in Genetics* (Vol. 36, Issue 6, pp. 442–455). Elsevier Ltd. <https://doi.org/10.1016/j.tig.2020.03.005>
- Costa, A., Moreira, A., & Sá, P. (2021). *Reflexões em torno de Metodologias de Investigação*. <https://doi.org/10.34624/dws9-6j98>
- Dagher, V. (2010, August 12). Macy's Tailored Merchandise Pays Off. *The Wall Street Journal*.
- Donoho, D. (2017). 50 Years of Data Science. In *Journal of Computational and Graphical Statistics* (Vol. 26, Issue 4, pp. 745–766). American Statistical Association. <https://doi.org/10.1080/10618600.2017.1384734>
- Dudek, G. (2016). Multilayer perceptron for GEFCom2014 probabilistic electricity price forecasting. *International Journal of Forecasting*, 32(3), 1057–1060. <https://doi.org/10.1016/j.ijforecast.2015.11.009>
- Fildes, R., Ma, S., & Kolassa, S. (2022). Retail forecasting: Research and practice. *International Journal of Forecasting*, 38(4), 1283–1318.
- Gaur, V., Fisher, M. L., Raman, A., & Stern, L. N. (2004). *An Econometric Analysis of Inventory Turnover Performance in Retail Services*.
- Gulcehre, C., Moczulski, M., Com, M. D., Bengio, Y., & Ca, B. U. (2016). *Noisy Activation Functions Misha Denil † †*.
- Harvey, A. (2006). Chapter 7 Forecasting with Unobserved Components Time Series Models. *Handbook of Economic Forecasting*, 1, 327–412. [https://doi.org/10.1016/S1574-0706\(05\)01007-4](https://doi.org/10.1016/S1574-0706(05)01007-4)

- Ho, S. L., & Xie, M. (1998). THE USE OF ARIMA MODELS FOR RELIABILITY FORECASTING AND ANALYSIS. In *Computers ind. Engng* (Vol. 35, Issue 2).
- Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1), 5–10.
<https://doi.org/10.1016/j.ijforecast.2003.09.015>
- Huang, T., Fildes, R., & Soopramanien, D. (2014). The value of competitive information in forecasting FMCG retail product sales and the variable selection problem. *European Journal of Operational Research*, 237(2), 738–748.
<https://doi.org/10.1016/j.ejor.2014.02.022>
- Huber, J., & Stuckenschmidt, H. (2020). Daily retail demand forecasting using machine learning with emphasis on calendric special days. *International Journal of Forecasting*, 36(4), 1420–1438. <https://doi.org/10.1016/j.ijforecast.2020.02.005>
- Hyndman, R., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3rd ed.). OTexts.
- Khashei, M., & Bijari, M. (2010). An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with Applications*, 37(1), 479–489.
<https://doi.org/10.1016/j.eswa.2009.05.044>
- Kück, M., & Freitag, M. (2021). Forecasting of customer demands for production planning by local k-nearest neighbor models. *International Journal of Production Economics*, 231.
<https://doi.org/10.1016/j.ijpe.2020.107837>
- Lauret, P., Fock, E., Randrianarivony, R. N., & Manicom-Ramsamy, J. F. (2008). Bayesian neural network approach to short time load forecasting. *Energy Conversion and Management*, 49(5), 1156–1166. <https://doi.org/10.1016/j.enconman.2007.09.009>
- Liu, L.-M., Hudak, G. B., Box, G. E. P., Muller, M. E., & Tiao, G. C. (1992). *FORECASTING AND TIME SERIES ANALYSIS USING THE SCA STATISTICAL SYSTEM VOLUME 1 Box-Jenkins ARIMA Modeling Intervention Analysis Transfer Function Modeling Outlier Detection and Adjustment Exponential Smoothing Related Univariate Methods in collaboration with.*
- Ma, S., & Fildes, R. (2021). Retail sales forecasting with meta-learning. *European Journal of Operational Research*, 288(1), 111–128. <https://doi.org/10.1016/j.ejor.2020.05.038>
- Mahesh, B. (2018). Machine Learning Algorithms-A Review. *International Journal of Science and Research*. <https://doi.org/10.21275/ART20203995>
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward. *PLoS ONE*, 13(3).
<https://doi.org/10.1371/journal.pone.0194889>
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2020). The M4 Competition: 100,000 time series and 61 forecasting methods. *International Journal of Forecasting*, 36(1), 54–74.
<https://doi.org/10.1016/j.ijforecast.2019.04.014>
- Oken, A. (2017). *An Introduction To and Applications of Neural Networks*.
- Oliveira, L. (2011). *Dissertação e Tese em Ciência e Tecnologia Segundo Bolonha*.

Bibliografia

- Rushton, Alan., Croucher, P., & Baker, P. (2014). *The handbook of logistics and distribution management : understanding the supply chain*.
- Rynkiewicz, J. (2019). Asymptotic statistics for multilayer perceptron with ReLU hidden units. *Neurocomputing*, 342, 16–23. <https://doi.org/10.1016/j.neucom.2018.11.097>
- Sagiroglu, S., & Sinanc, D. (2013). Big data: A review. *Proceedings of the 2013 International Conference on Collaboration Technologies and Systems, CTS 2013*, 42–47. <https://doi.org/10.1109/CTS.2013.6567202>
- Samuel, A. L. (1967). *Some Studies in Machine Learning Using the Game of Checkers*.
- Smyl, S. (2020). A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting*, 36(1), 75–85. <https://doi.org/10.1016/j.ijforecast.2019.03.017>
- Surakhi, O., Zaidan, M. A., Fung, P. L., Motlagh, N. H., Serhan, S., Alkhanafseh, M., Ghoniem, R. M., & Hussein, T. (2021). Time-lag selection for time-series forecasting using neural network and heuristic algorithm. *Electronics (Switzerland)*, 10(20). <https://doi.org/10.3390/electronics10202518>
- Theodorou, E., Wang, S., Kang, Y., Spiliotis, E., Makridakis, S., & Assimakopoulos, V. (2022). Exploring the representativeness of the M5 competition data. *International Journal of Forecasting*, 38(4), 1500–1506. <https://doi.org/10.1016/j.ijforecast.2021.07.006>
- Vasco-Carofilis, R. A., Gutiérrez-Naranjo, M. A., & Cárdenas-Montes, M. (2020). PBIL for Optimizing Hyperparameters of Convolutional Neural Networks and STL Decomposition. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 12344 LNAI, 147–159. https://doi.org/10.1007/978-3-030-61705-9_13
- Walliman, N. (2011). *Research Methods - The Basics*.
- Zhou, W. Q., Chen, L., & Ge, H. M. (2013). A multi-product multi-echelon inventory control model with joint replenishment strategy. *Applied Mathematical Modelling*, 37(4), 2039–2050. <https://doi.org/10.1016/j.apm.2012.04.054>
- Zivot, E., & Wang, J. (2006). *Modelling Financial Time Series with S-PLUS, Second Edition*.

APÊNDICE A

```

import numpy as np
import statsmodels.api as sm
#import statsmodels.formula.api as smf
import matplotlib.pyplot as plt
import pandas as pd
from statsforecast import StatsForecast #Imports the core StatsForecast class
from statsmodels.graphics import tsaplots
from statsmodels.regression.linear_model import OLS
from sklearn.model_selection import KFold
from statsforecast.models import MSTL
from sklearn.model_selection import train_test_split
from sklearn.model_selection import cross_val_score
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import mean_squared_error, mean_absolute_error
from statsmodels.tsa.holtwinters import SimpleExpSmoothing
from statsmodels.tsa.seasonal import STL
from sklearn.preprocessing import StandardScaler
from keras.models import Sequential
from keras.layers import Dense
from keras.callbacks import EarlyStopping
import time
import warnings
from statsmodels.tsa.arima.model import ARIMA
from statsmodels.tsa.statespace.sarimax import SARIMAX
import lightgbm as lgb
from itertools import product
from pmdarima import auto_arima
from sklearn.metrics import r2_score

#functions

# Create Lags
def create_lags(df):
    for i in range(1, 5):
        df[f'sales_lag_{i}'] = df['qty'].shift(i)
    return df

# Calculate RMSE
def compute_rmse(actual, forecast):
    return np.sqrt(np.mean((actual - forecast)**2))

# Return STL Components
def stl_decomposition(series, period=52):

    stl = STL(series, seasonal=period)
    result = stl.fit()
    return result.seasonal, result.trend, result.resid

#plot actual vs predicted
def plot_actual_vs_predicted(sku):
    actual = predicted_values[sku]['actual']
    predicted = predicted_values[sku]['predicted']

    plt.figure(figsize=(10, 6))
    plt.plot(actual, label='Real', color='blue', marker='o')
    plt.plot(predicted, label='Previsto', color='red', linestyle='--', marker='x')

```

APÊNDICE A

```
plt.title(f"Valores Reais vs Previstos do SKU {sku}")
plt.xlabel("TimePeriods")
plt.ylabel("Value")
plt.legend()
plt.grid(True)
plt.show()
```

Importing Data

```
path_name = 'C:/Users/ricar/Desktop/MEGI/Dissertação/Material Jorge/Vendas Sintra 2018 a
2022/'
file_name = 'dados V3.csv'
df_sales = pd.read_csv(path_name+file_name)#, index_col="Month")

# SKU to return
id_to_match = 'X00001'

# Created new DataFrame with a single SKU
new_df = df_sales.query("`SKU` == @id_to_match")

# Historical period length
periods = len(new_df)
new_df.insert(0,['unique_id'], [id_to_match for i in range(periods)],True)
new_df.insert(1,['ds'], [i for i in range(1,periods+1)],True)
selected_columns = ['unique_id', 'ds', 'qty']
Y_df = new_df.loc[:, selected_columns]
print(Y_df)

# Decompose the time series
df = Y_df

# Decompose the time series
stl = STL(df['qty'], seasonal=51, period=53) # setting seasonal period and period both to 52 weeks
result = stl.fit()
result.plot()
plt.show()

# Extract components
df['seasonality'] = result.seasonal
df['trend'] = result.trend
df['residual'] = result.resid

print(df[['unique_id', 'ds', 'seasonality', 'trend', 'residual']])

StatsForecast.plot(Y_df)

# Extract relevant columns for plotting
x = Y_df["ds"]
y = Y_df["qty"]
title = Y_df["unique_id"].iloc[0]

# Set the size of the plot (width, height)
fig, ax = plt.subplots(figsize=(8, 3))

# Plotting
ax.plot(x, y, '-o', label='Vendas')
ax.set_xlabel('Semana')
```

```
ax.set_ylabel('Vendas')
ax.set_title(title)
ax.legend()
```

```
plt.show()
```

```
x = new_df['qty'].values
print(x)
```

Time Series Components Analysis

```
# Autocorrelation Test
```

```
k=155
```

```
sm.tsa.stattools.acf(x) #, adjusted=False, nlags=k, qstat=False, fft=True, alpha=0.05,
bartlett_confint=True, missing='none')
```

```
#sm.tsa.stattools.acf(x, adjusted=False, nlags=None, qstat=False, fft=True, alpha=None,
bartlett_confint=True, missing='none')
```

```
#statsmodels.graphics.tsaplots.plot_acf(x, ax=None, lags=None, *, alpha=0.05, use_vlines=True,
adjusted=False, fft=False, missing='none', title='Autocorrelation', zero=True, auto_ylims=False,
bartlett_confint=True, vlines_kwargs=None, **kwargs)
```

```
ACF=sm.graphics.tsa.plot_acf(x,bartlett_confint=False,lags=k, zero=False, title='Autocorrelation ' +
id_to_match) #,lags=k, alpha=0.05,fft=True, title='Autocorrelation ' + tsname, zero=False,
auto_ylims=False, bartlett_confint=False)
```

```
#plt.show()
```

```
# Linear regression
```

```
X = new_df['ds']
```

```
Y = new_df['qty']
```

```
model = OLS(Y, sm.add_constant(X)).fit()
```

```
print(f'Regression coefficient (trend): {model.params['ds']}")
```

```
model = OLS(Y, sm.add_constant(X)).fit()
```

```
# Print the summary
```

```
print(model.summary())
```

Anomaly Detection

```
models = [MSTL(season_length = [4, 52])]
```

```
sf = StatsForecast(
    df = Y_df,
    models = models,
    freq = 'W',
    n_jobs = -1
)
```

```
horizon = 4
```

```
levels = [99]
```

```
fcst = sf.forecast(h = 4, level = levels, fitted = True)
```

```
fcst = fcst.reset_index()
```

```
fcst.head()
```

```
StatsForecast.plot(Y_df, fcst, plot_random = False)
```

APÊNDICE A

```

insample_forecasts = sf.forecast_fitted_values().reset_index()
insample_forecasts.head()

anomalies = insample_forecasts.loc[(insample_forecasts['y'] >= insample_forecasts['MSTL-hi-99']) |
(insample_forecasts['y'] <= insample_forecasts['MSTL-lo-99'])]
anomalies.head()

StatsForecast.plot(insample_forecasts, plot_random = False, plot_anomalies = True)

```

Prepare data for machine learning

```

####Create Lags###

data = df_sales

# Group by SKU
data = data.sort_values(by=['SKU', 'TimePeriod']) # Ensure data is sorted by SKU and then by time
data = data.groupby('SKU').apply(create_lags)
# Delete SKU's without lags
data.dropna(inplace=True)

forecast_df = data

forecast_df.drop(columns=['StoreId', 'StoreLocation', 'StoreType', 'Total'], inplace=True)

# Convert categorical variables using one-hot encoding
forecast_df = pd.get_dummies(data, columns=['Family', 'SubFamily', 'SKU'], drop_first=True)
forecast_df

```

Forecasting Methods

```

####Stock Out Correction###

#find last non zero value
def find_last_non_zero(idx, col):
    while idx >= 0:
        if df.at[idx, col] != 0:
            return df.at[idx, col]
        idx -= 1
    return None

# Unique SKUs in df_sales
unique_skus = df_sales['SKU'].unique()
new_df = df_sales[['TimePeriod', 'SKU', 'qty']]

all_stock_out_df = pd.DataFrame()

for sku in unique_skus:
    id_to_match = sku

    # Created new DataFrame with a single SKU
    df = new_df.query("`SKU` == @id_to_match").reset_index(drop=True)

    # Find zero values

```

```

zero_values = df[df['qty'] == 0].index

# For each zero value, replace with previous two and next two available values
for idx in zero_values:
    values = []

    # add next two values
    if idx - 1 >= 0:
        values.append(df.at[idx - 1, 'qty'])
    if idx - 2 >= 0:
        values.append(df.at[idx - 2, 'qty'])
    else:
        last_non_zero = find_last_non_zero(idx - 1, 'qty')
        if last_non_zero is not None:
            values.append(last_non_zero)

    # add last two values
    if idx + 1 < len(df):
        values.append(df.at[idx + 1, 'qty'])
    if idx + 2 < len(df):
        values.append(df.at[idx + 2, 'qty'])

    # replace values by average
    if values:
        df.at[idx, 'qty'] = round(sum(values) / len(values), 0)
    else:
        print(f"Não é possível substituir o valor zero no índice {idx} pois não existem valores disponíveis.")

all_stock_out_df = pd.concat([all_stock_out_df, df])

#Corrected vs Original

SKU_plot = 'X00013'

original_df = df_sales[['TimePeriod', 'SKU', 'qty']]
original_df = original_df.query("`SKU` == @SKU_plot")
corrected_df = all_stock_out_df.query("`SKU` == @SKU_plot")
original_df

# Plotting
plt.figure(figsize=(12, 6))
plt.plot(original_df['TimePeriod'], original_df['qty'], label='Original', color='blue')
plt.plot(corrected_df['TimePeriod'], corrected_df['qty'], label='Corrigido', color='red', linestyle='--')

plt.xlabel("TimePeriod")
plt.ylabel("Qty")
plt.title(f"Comparação dos valores originais e corrigidos no artigo {SKU_plot}")
plt.legend()
plt.grid(True)
plt.tight_layout()
plt.show()

###Naive Method###

start_period = 105

```

APÊNDICE A

```

# List to store overall RMSE values for each SKU
all_rmse_values = []

# Unique SKUs in df_sales
unique_skus = df_sales['SKU'].unique()

for sku in unique_skus:
    # Filter the dataframe for the current SKU
    df_forecast = all_stock_out_df[all_stock_out_df['SKU'] == sku]
    df_forecast = df_forecast[['TimePeriod', 'qty']]

    # List to store RMSE values for the current SKU
    rmse_values = []

    # Loop through the dataframe using a step size of 2 starting from start_period
    for i in range(start_period, len(df_forecast) - 1, 2):
        # Get the forecasted value from the previous period using all_anomalies_df
        naive_forecast = df_forecast.loc[df_forecast['TimePeriod'] == i - 1, 'qty'].values[0]

        # Extract the actual values for the next 2 periods from df_sales (original dataframe)
        actual_values = df_forecast.loc[(df_forecast['TimePeriod'] >= i) & (df_forecast['TimePeriod']
        < i + 2), 'qty'].values

        # Check if there are two periods to forecast, if not, break the loop
        if len(actual_values) < 2:
            break

        # Compute the RMSE for this window and append to the list
        rmse_values.append(compute_rmse(actual_values, [naive_forecast, naive_forecast]))

    # Compute overall RMSE for the current SKU and store it
    overall_rmse = np.mean(rmse_values)
    all_rmse_values.append((sku, overall_rmse))
    print(f"{sku} RMSE: {overall_rmse:.2f}")

####Current Method###

# Service Level
NS = 2.8
start_period = 105

# List to store overall RMSE values for each SKU
all_rmse_values = []

# Dictionary to store actual and predicted values for all SKUs
predicted_values = {}

# Unique SKUs in df_sales
unique_skus = df_sales['SKU'].unique()

for sku in unique_skus:
    # Filter the dataframe for the current SKU
    df_forecast = all_stock_out_df[all_stock_out_df['SKU'] == sku]
    df_forecast = df_forecast[['TimePeriod', 'qty']]

    # Lists to store actual and predicted values for the current SKU
    actual_sku_values = []
    predicted_sku_values = []

```

```

# List to store RMSE values for the current SKU
rmse_values = []

# Loop through the dataframe using a step size of 2 starting from start_period
for i in range(start_period, len(df_forecast) - 1, 2):
    # Extract the last 4 timeperiods before the current forecast period
    last_4_weeks = df_forecast.loc[(df_forecast['TimePeriod'] >= i - 4) & (df_forecast['TimePeriod']
< i), 'qty'].values

    # Compute the forecast value
    forecast_value = np.mean(last_4_weeks) + (NS * np.std(last_4_weeks))

    # Extract the values for the next 2 periods
    actual_values = df_forecast.loc[(df_forecast['TimePeriod'] >= i) & (df_forecast['TimePeriod'] < i +
2), 'qty'].values

    # Check if there are two periods to forecast, if not, break the loop
    if len(actual_values) < 2:
        break

    # Add the actual and forecasted values to the respective lists
    actual_sku_values.extend(actual_values)
    predicted_sku_values.extend([forecast_value, forecast_value])

    # Compute the RMSE for this window and append to the list
    rmse_values.append(compute_rmse(actual_values, [forecast_value, forecast_value]))

# Update the predicted_values dictionary for the current SKU
predicted_values[sku] = {'actual': actual_sku_values, 'predicted': predicted_sku_values}

# Compute overall RMSE for the current SKU and store it
overall_rmse = np.mean(rmse_values)
all_rmse_values.append((sku, overall_rmse))
print(f"SKU: {sku} -> Overall RMSE: {overall_rmse:.2f}")

###Simple Exponential Smoothing###

# Get unique SKUs
unique_SKUs = df_sales['SKU'].unique()

# Dictionary to store RMSE for each SKU
sku_rmse_dict = {}

# Loop over each SKU
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    for sku in unique_SKUs:

        df_forecast = all_stock_out_df[all_stock_out_df['SKU'] == sku]
        df_forecast = df_forecast[['TimePeriod', 'qty']]

        start_period = 105
        rmse_values = []

        # Model training and forecasting
        for i in range(start_period, len(df_forecast) - 1, 2):

```

APÊNDICE A

```

train = df_forecast.loc[df_forecast['TimePeriod'] < i, 'qty']
test = df_forecast.loc[df_forecast['TimePeriod'].between(i, i+1), 'qty']

model = sm.tsa.ExponentialSmoothing(train).fit(optimized=True)

# Forecast 2 steps ahead
forecast = model.forecast(steps=2)

# In case the test data is less than 2 periods (at the end of dataset)
steps = len(test)

rmse = compute_rmse(test.values, forecast[:steps])
rmse_values.append(rmse)

overall_rmse = np.mean(rmse_values)
sku_rmse_dict[sku] = overall_rmse
print(f'SKU: {sku}, Overall RMSE: {overall_rmse:.2f}')

###Holt Exponential Smoothing with Trend###

# Get unique SKUs
unique_SKUs = df_sales['SKU'].unique()

# Dictionary to store RMSE for each SKU
sku_rmse_dict = {}

# Loop over each SKU
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    for sku in unique_SKUs:

        df_forecast = all_stock_out_df[all_stock_out_df['SKU'] == sku]
        df_forecast = df_forecast[['TimePeriod', 'qty']]

        start_period = 105
        rmse_values = []

        # Model training and forecasting
        for i in range(start_period, len(df_forecast) - 1, 2):
            train = df_forecast.loc[df_forecast['TimePeriod'] < i, 'qty']
            test = df_forecast.loc[df_forecast['TimePeriod'].between(i, i+1), 'qty']

            model = sm.tsa.ExponentialSmoothing(train, trend='add').fit(optimized=True)

            # Forecast 2 steps ahead
            forecast = model.forecast(steps=2)

            # In case the test data is less than 2 periods (at the end of dataset)
            steps = len(test)

            rmse = compute_rmse(test.values, forecast[:steps])
            rmse_values.append(rmse)

        overall_rmse = np.mean(rmse_values)
        sku_rmse_dict[sku] = overall_rmse
        print(f'SKU: {sku}, Overall RMSE: {overall_rmse:.2f}')

###Holt-Winters Exponential Smoothing with Trend and Seasonality###

```

```

# Get unique SKUs
unique_SKUs = df_sales['SKU'].unique()

# Dictionary to store RMSE for each SKU
sku_rmse_dict = {}

# Loop over each SKU
with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    for sku in unique_SKUs:

        df_forecast = all_stock_out_df[all_stock_out_df['SKU'] == sku]
        df_forecast = df_forecast[['TimePeriod', 'qty']]

        start_period = 105
        rmse_values = []

        # Model training and forecasting
        for i in range(start_period, len(df_HW) - 1, 2):
            train = df_forecast.loc[df_forecast['TimePeriod'] < i, 'qty']
            test = df_forecast.loc[df_HW['TimePeriod'].between(i, i+1), 'qty']

            model = sm.tsa.ExponentialSmoothing(train, trend='add', seasonal='add',
            seasonal_periods=52).fit(optimized=True)

            # Forecast 2 steps ahead
            forecast = model.forecast(steps=2)

            # In case the test data is less than 2 periods (at the end of dataset)
            steps = len(test)

            rmse = compute_rmse(test.values, forecast[:steps])
            rmse_values.append(rmse)

        overall_rmse = np.mean(rmse_values)
        sku_rmse_dict[sku] = overall_rmse
        print(f"SKU: {sku}, Overall RMSE: {overall_rmse:.2f}")

### ARIMA ###

start_period = 105
all_rmse_values = []
unique_skus = df_sales['SKU'].unique()

with warnings.catch_warnings():
    warnings.simplefilter("ignore")
    for sku in unique_skus:

        df_forecast = all_stock_out_df[all_stock_out_df['SKU'] == sku]
        df_forecast = df_forecast.set_index('TimePeriod')['qty']

        rmse_values = []

        # auto-detect ARIMA order
        order = auto_arima(df_forecast, seasonal=False, trace=False, error_action="ignore",
        suppress_warnings=True, stepwise=True).order

        for i in range(start_period, len(df_forecast) - 1, 2):

```

APÊNDICE A

```

# Train up to the current point
train = df_forecast.iloc[:i]
test = df_forecast.iloc[i:i+2]

if len(test) < 2:
    break

# Initialize and train the ARIMA model
model = ARIMA(train, order=order)
model_fit = model.fit()

# Make forecasts
forecast = model_fit.forecast(steps=2)

# Compute RMSE for this forecast window
rmse_values.append(compute_rmse(test.values, forecast))

# Average RMSE for this SKU
overall_rmse = np.mean(rmse_values)
all_rmse_values.append((sku, overall_rmse))
print(f"SKU: {sku} -> Overall RMSE: {overall_rmse:.2f}")

### SARIMA ###

warnings.filterwarnings("ignore")

start_period = 105
all_rmse_values = []
unique_skus = df_sales['SKU'].unique()

with warnings.catch_warnings():
    warnings.simplefilter("ignore")

    for sku in unique_skus:

        df_forecast = all_stock_out_df[all_stock_out_df['SKU'] == sku]
        df_forecast = df_forecast.set_index('TimePeriod')['qty']

        rmse_values = []

        # Auto-detect SARIMA order
        model_order = auto_arima(df_sku, m=52, seasonal=True, trace=False, error_action="ignore",
            suppress_warnings=True, stepwise=True)

        order = model_order.order
        seasonal_order = model_order.seasonal_order

        for i in range(start_period, len(df_forecast) - 1, 2):
            # Train up to the current point
            train = df_forecast.iloc[:i]
            test = df_forecast.iloc[i:i+2]

            if len(test) < 2:
                break

            # Initialize and train the SARIMA model
            model = SARIMAX(train, order=order, seasonal_order=seasonal_order)
            model_fit = model.fit()

```

```

# Make forecasts
forecast = model_fit.forecast(steps=2)

# Compute RMSE for this forecast window
rmse_values.append(compute_rmse(test.values, forecast))

# Average RMSE for this SKU
overall_rmse = np.mean(rmse_values)
all_rmse_values.append((sku, overall_rmse))
print(f"SKU: {sku} -> Overall RMSE: {overall_rmse:.2f}")

###MultiLayer Perceptron

early_stop = EarlyStopping(monitor='val_loss', mode='min', verbose=1, patience=25)

data = forecast_df.copy()

start_period = 105
end_period = 154
rmse_values = {}
r2_values = {}
predicted_values = {}

unique_skus = data.index.get_level_values(0).unique() # Retrieve unique SKUs

for current_end in range(start_period, end_period):
    print(f"Training until TimePeriod: {current_end} and predicting TimePeriods: {current_end + 1}
    and {current_end + 2}")
    start_time = time.time() # Start the timer

    for sku in unique_skus:
        # 1. Filter the Data, Run Simple Exponential Smoothing, Combine with Original Data
        filtered_df = data.loc[(data.index.get_level_values(0) == sku) & (data["TimePeriod"] <=
        current_end)]

        # Check if enough data points exist for STL decomposition
        if len(filtered_df) > 53:
            model = SimpleExpSmoothing(filtered_df['qty'].values)
            fit_model = model.fit()
            forecast = fit_model.forecast(steps=2)

            # Creates index to join the decomposed values
            max_index_value = filtered_df.index.get_level_values(1).max() # get the max value of the
            second level of the index
            next_index_1 = max_index_value + 1
            next_index_2 = max_index_value + 2

            # Create a multi-index for forecasted values
            forecast_index = pd.MultiIndex.from_tuples([(sku, next_index_1), (sku, next_index_2)])
            forecast_series = pd.Series(forecast, index=forecast_index)

            # Combine the filtered_df and forecasted values
            union_series = pd.concat([filtered_df['qty'], forecast_series])

            # Apply STL on union_series
            try:
                stl = STL(union_series.values, seasonal=51, period=52).fit()

```

APÊNDICE A

```

idx_to_assign = union_series.index

seasonal_series = pd.Series(stl.seasonal, index=idx_to_assign)
trend_series = pd.Series(stl.trend, index=idx_to_assign)
residual_series = pd.Series(stl.resid, index=idx_to_assign)

data.loc[idx_to_assign, 'seasonality'] = seasonal_series
data.loc[idx_to_assign, 'trend'] = trend_series
data.loc[idx_to_assign, 'residual'] = residual_series
except Exception as e:
    print(f"Error processing SKU {sku} for STL: {e}")

# Model training and predictions
train_data = data[data['TimePeriod'] <= current_end]
X_train = train_data.drop('qty', axis=1)
y_train = train_data['qty']

scaler = StandardScaler().fit(X_train)
X_train = scaler.transform(X_train)

model = Sequential()
model.add(Dense(128, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(64, activation='relu'))
model.add(Dense(1, activation='linear'))
model.compile(loss='mean_squared_error', optimizer='adam')
model.fit(X_train, y_train, epochs=300, batch_size=32, validation_split=0.2,
callbacks=[early_stop], verbose=0)

for sku in unique_skus:
    for week in [current_end + 1, current_end + 2]:
        test_data = data.loc[(data.index.get_level_values(0) == sku) & (data['TimePeriod'] == week)]
        if test_data.empty:
            continue

        X_test = test_data.drop('qty', axis=1)
        y_test = test_data['qty']
        X_test = scaler.transform(X_test)
        predictions = model.predict(X_test, verbose=0).flatten()

        if sku not in predicted_values:
            predicted_values[sku] = {
                'actual': [],
                'predicted': []
            }

        predicted_values[sku]['actual'].extend(y_test.values)
        predicted_values[sku]['predicted'].extend(predictions)

        rmse = compute_rmse(predictions, y_test.values)
        if sku not in rmse_values:
            rmse_values[sku] = []
            rmse_values[sku].append(rmse)
        r2 = r2_score(y_test.values, predictions)
        if sku not in r2_values:
            r2_values[sku] = []
            r2_values[sku].append(r2)

elapsed_time = time.time() - start_time # Calculate elapsed time
print(f"Time taken for TimePeriod {current_end}: {elapsed_time:.2f} seconds")

```

```
for sku, rmse_list in rmse_values.items():
    avg_rmse = np.mean(rmse_list)
    print(f"Average RMSE for SKU {sku} over 52 weeks: {avg_rmse:.2f}")

#for sku, r2_list in r2_values.items():
#    avg_r2 = np.mean(r2_list)
#    print(f"Average R^2 for SKU {sku} over 52 weeks: {avg_r2:.2f}")

#Plot real vs prediction
plot_actual_vs_predicted('X00012')
```