Previsão de Vendas para o Setor do Retalho Aplicando Máquinas de Vetores de Suporte

Fernando António Castro da Silva

Dissertação de Mestrado

Orientador na FEUP: Prof. Américo Azevedo
Orientador na Empresa: Prof. Patrícia Ramos

Mestrado Integrado em Engenharia e Gestão Industrial

2020-06-29

Resumo

No atual mundo empresarial, a tomada de decisão a nível estratégico e a nível operacional deverá, inevitavelmente, assentar num manancial de informação atualizada e fiável. O recurso a uma metodologia de previsão de vendas eficaz será fundamental para a recolha dessa informação.

Especificamente no setor do retalho, a competitividade exige que as empresas utilizem ferramentas fiáveis e rápidas de previsão de vendas, no sentido de otimizarem os seus custos operacionais e garantirem a satisfação do cliente.

Hoje em dia, as empresas da área do retalho possuem bases de dados que armazenam grandes quantidades de informação acerca dos diferentes SKUs (*Stock Keeping Units*), tornando o processo de previsão bastante complexo, visto que não existe nenhum modelo que apresente uma boa performance para todos os SKUs.

Assim, o presente trabalho surgiu numa tentativa de dar resposta a este problema. Este trabalho propõe uma metodologia inovadora, baseada em máquinas de vetores de suporte (SVMs - *Support Vector Machines*), que permite selecionar, de forma expedita e eficaz, o melhor modelo de previsão para cada SKU, com base nas características da respetiva série de vendas.

Para atingir o objetivo proposto, procedeu-se ao cálculo das previsões de um conjunto de modelos diversificados de forma a obter as respetivas medidas de erro e, com base nestas medidas, determinou-se o modelo com melhor desempenho para cada série temporal. Posteriormente, extraiu-se um conjunto de características correspondente a cada série e reduziu-se a dimensão do mesmo, recorrendo à análise das componentes principais. Por fim, o classificador SVM foi treinado com as componentes principais obtidas e com os identificadores dos melhores modelos para cada série de vendas.

No que concerne à implementação da metodologia, bem como à sua avaliação, foi utilizado o *software* R.

Os resultados obtidos revelaram que a *framework* proposta apresenta uma performance superior à dos modelos de previsão mais comuns, garantindo previsões fiáveis e um tempo computacional reduzido.

Palavras-chave: Previsão de vendas; Retalho; Seleção de modelos; Máquinas de vetores de suporte; Séries temporais

Abstract

In today's business world, the decision making at the strategic and operational level must, inevitably, be based on a set of updated and reliable information. The use of an effective sales forecasting methodology will be essential for the collection of this information.

Particularly, in the retail sector, its competitiveness requires companies to use reliable and fast sales forecasting tools, in order to optimize their operating costs and ensure customer satisfaction.

Nowadays, the retailers have databases that store a large amounts of information about the different SKUs (Stock Keeping Units), making the forecasting process quite complex, since there is no model that exhibit a good performance for all SKUs.

Thus, the present work arose in an attempt to answer this problem. The work proposes an innovative methodology, based on support vector machines (SVMs), which allows, in a fast and effective way, the selection of the best forecasting model for each SKU, based on the features of the respective sales series.

In order to reach the goal of this thesis, we calculate the predictions for a set of diversified models to obtain the respective error measures and, based on these measures, the model with the best performance for each time series was determined. Subsequently, a set of features corresponding to each time series was extracted and then, it was applied the principal components analysis for the dimensionality reduction. Finally, the SVM classifier was trained with the principal components obtained and the labels of the best models for each sales series

Regarding the implementation of the methodology, as well as its evaluation, it was used the software R.

The results revealed that the proposed framework outperforms the most common forecasting models, ensuring reliable forecasts and a reduced computational time.

Keywords: Sales forecasting; Retail; Model selection; Support vector machines; Time series

Agradecimentos

Em primeiro lugar, quero agradecer aos meus orientadores pelo apoio prestado ao longo do desenvolvimento deste projeto. Quero prestar um especial agradecimento à Professora Patrícia Ramos por todo conhecimento transmitido, o qual foi essencial para a realização desta dissertação. Também estou muito grato ao Professor Américo Azevedo que contribuiu para minha formação académica, quer com o seu saber, quer com a sua generosa disponibilidade.

Deixo o meu agradecimento ao INESC TEC que me facultou todos os recursos necessários para a realização deste trabalho.

Quero deixar um especial obrigado a toda a minha família, particularmente aos meus pais, não só pelo apoio prestado ao longo de toda a minha vida académica, mas também pelos seus ensinamentos que me ajudaram a enfrentar as adversidades com que me fui deparando.

Por fim, quero agradecer a todos os meus amigos que, ao longo deste trajeto, me deram bons conselhos e sempre me motivaram a alcançar a meta traçada.

Fernando António Castro da Silva

"If there's one thing that's certain in business, it's uncertainty."

Stephen R. Covey

Conteúdo

Acronyms and Symbols

ACF	Autocorrelation Function
AIC	Akaike Information Criteria
AICc	Corrected Akaike Information Criteria
ANN	Artificial Neural Networks
ARIMA	Autoregressive Integrated Moving Average
ARFIMA	Autoregressive Fractionally Integrated Moving Average
ASP	Algorithm Selection Problem
BIC	Bayesian Information Criteria
ENN	Evolutionary Neural Networks
ETS	ExponenTial Smoothing
GMRelMAE	Geometric Mean Relative Mean Absolute Error
IDE	Integrated Development Environment
KKT	Karush-Kuhn-Tucker
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MASE	Mean Absolute Scaled Error
OAA	One-Against-All
OAO	One-Against-One
PACF	Partial Autocorrelation Function
PCA	Principal Components Analysis
RNA	Redes Neuronais Artificiais
RelMAE	Relative Mean Absolute Error
RMSE	Root Mean Square Error
RSS	Residual Sum of Squares
SES	Simple Exponential Smoothing
SKU	Stock Keeping Unit
STL	Seasonal and Trend decomposition using Loess
SVM	Support Vector Machine
sMAPE	symetric Mean Absolute Percentage Error
TBATS	Trigonometric Box-Cox transform, ARMA errors, Trend, and Seasonal components

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Enquadramento e motivação

Atualmente, a previsão de vendas é uma atividade de extrema importância para as empresas, uma vez que fornece informação essencial para a tomada de decisão, tanto a nível estratégico como a nível operacional. A aplicação de modelos de previsão da procura não só permite diminuir o risco associado à variabilidade da procura, mas também possibilita um planeamento mais eficaz e eficiente das operações de toda a cadeia de abastecimento.

Especialmente na área do retalho, a eficácia da previsão da procura será determinante para a competitividade das empresas. Uma deficiente previsão de vendas poderá causar perdas significativas ao retalhista, quer por rutura, quer por excesso de *stock*. A falta de um determinado produto em *stock* poderá implicar custos para o retalhista, decorrentes da quantidade que não foi vendida e que o consumidor estaria disposto a comprar. Por outro lado, o excesso de inventário acarretará custos, nomeadamente, de armazenamento e de deterioração. Particularmente no setor do retalho alimentar, existe uma especial preocupação com o controlo de custos derivados da deterioração dos produtos, visto que muitos dos produtos são perecíveis, apresentando um tempo de vida útil bastante curto.

Por conseguinte, é imperativo que as empresas de retalho se apoiem numa eficaz ferramenta de previsão da procura, de forma a minimizarem os seus custos logísticos, sem comprometerem a satisfação do cliente.

Hoje em dia, devido aos avanços tecnológicos, é possível armazenar grandes quantidades de informação relativa aos diferentes SKUs, o que tornou o processo de previsão bastante exigente mesmo para as tecnologias mais sofisticadas.

De acordo com a literatura, nenhum dos modelos de previsão existentes apresenta um bom desempenho para todo tipo de séries temporais, dificultando o processo de previsão para os casos em que o número de SKUs é elevado (Kang et al., 2017; Timmermann, 2006). De facto, a performance dos modelos de previsão depende das séries temporais às quais são aplicados, uma vez que os modelos exploram diferentes características do comportamento das séries, adaptando-se melhor ou pior em função da existência dessas mesmas características (Petropoulos et al., 2014).

Efetivamente, uma das alternativas para contornar este problema seria a seleção do modelo mais adequado para cada série temporal. Contudo, esta opção não será viável para um grande conjunto séries, dado que implica um elevado tempo de processamento.

Especificamente para o setor do retalho, em virtude do número elevado de SKUs, é necessário um mecanismo que possibilite a seleção automática do melhor modelo para cada SKU.

Assim, o presente trabalho é motivado por esta necessidade dos retalhistas, mais concretamente do Grupo Jerónimo Martins, procurando desenvolver uma *framework* que permita selecionar, de uma forma rápida, o melhor modelo para cada SKU e que assegure a exatidão das previsões. Neste sentido, numa parceria com o INESC TEC, o Grupo Jerónimo Martins disponibilizou um conjunto de dados acerca das vendas dos SKUs das lojas Pingo Doce que permite avaliar o desempenho da metodologia proposta.

1.2 Objetivos do trabalho

O principal objetivo desta dissertação é desenvolver uma *framework* de previsão de vendas, baseada em máquinas de vetores de suporte para o setor do retalho.

As SVMs são algoritmos supervisionados bastante populares, que, graças à sua capacidade de generalização, permitem evitar problemas de *overfitting*. Desta forma, a metodologia proposta aplica este algoritmo para selecionar o melhor modelo para cada série, tendo em conta as suas características.

A *framework* está dividida em duas fases: a fase *online* e a fase *offline*.

Na fase *offline*, o algoritmo de classificação é treinado com as características de cada SKU e com o respetivo indentificador do melhor modelo. A identificação do melhor modelo para cada SKU é baseada nas medidas de erro associadas à utilização dos modelos candidatos.

Posteriormente, na fase *online*, será inserida uma série temporal correspondente a um novo SKU, para o qual são calculadas as respetivas características e se determina o melhor modelo a ajustar a esse SKU, através do algoritmo de classificação obtido na fase anterior.

O conjunto de modelos candidatos, utilizado na fase de treino do algoritmo SVM, inclui vários modelos na vanguarda do estado da arte, de forma a tornar o algoritmo mais robusto. Por conseguinte, esta fase requer um elevado esforço computacional, visto que todos os modelos são ajustados a cada série. Contudo, não será um entrave para a aplicação da *framework*, já que esta fase é concebida de uma forma *offline*, com o intuito de não afetar o tempo de processamento da metodologia.

Tanto para o desenvolvimento da *framework* sugerida, como para a sua avaliação serão utilizadas as funcionalidades do *software* R.

Este trabalho apresenta uma abordagem inovadora, visto que ainda não foi aplicada no contexto da previsão de vendas do setor do retalho. Também é digno de realce a complexidade envolvida no projeto, derivada não só pelo algoritmo SVM incorporado na *framework*, mas também pela utilização de um conjunto de modelos de previsão sofisticados. O desempenho da *framework*

será avaliado com base na comparação dos resultados obtidos para a mesma com os resultados da aplicação generalizada dos modelos candidatos.

O objetivo do trabalho será cumprido, caso a metodologia desenvolvida produza, de forma expedita, previsões com uma elevada exatidão, permitindo a sua implementação nas lojas Pingo Doce do Grupo Jerónimo Martins.

1.3 Estrutura da dissertação

A presente dissertação está dividida em quatro capítulos.

No primeiro capítulo, é introduzida a ideia subjacente ao projeto desenvolvido, bem como uma contextualização que sustenta a abordagem adotada.

No capítulo 2, são descritos os conceitos sobre os quais a metodologia é desenvolvida e é feita uma análise crítica dos diferentes contributos encontrados na literatura.

O capítulo 3 expõe o caso de estudo, apresentando a arquitetura da metodologia elaborada, o procedimento experimental e os resultados obtidos.

Por fim, no capítulo 4, constam as principais conclusões do trabalho realizado, assim como sugestões para um trabalho futuro.

Capítulo 2

Estado da Arte

2.1 Séries Temporais

Uma série temporal pode ser definida como uma sequência de observações relativamente a uma certa variável de interesse (Montgomery et al., 2008). As séries temporais podem ser discretas ou contínuas. No caso de serem discretas, as observações são feitas em instantes específicos e, geralmente, separadas por intervalos de tempo fixos. Por outro lado, nas séries contínuas, as observações são medidas continuamente durante um certo período de tempo (Brockwell and Davis, 2002).

A análise de séries temporais permite não só perceber as razões de acontecimentos passados, mas também prever eventos futuros. A previsão de eventos futuros é bastante importante para os processos de planeamento e de tomada de decisão presentes em diversas áreas: gestão de operações, gestão financeira, marketing, entre outras.

De um modo geral, uma das características das séries temporais é a dependência entre observações vizinhas. A análise das séries temporais pretende então estudar esta dependência, desenvolvendo modelos estocásticos. Os modelos estocásticos são processos probabilísticos que permitem calcular a probabilidade de um valor futuro entre dois limites especificados (Box et al., 2016).

A partir da análise das séries temporais é possível detetar determinados padrões, tais como: tendência, sazonalidade, observações aleatórias ou combinação de padrões (Montgomery et al., 2008).

2.2 *Meta-Learning*

Os sistemas de *meta-learning* são mecanismos que procuram entender como é que os sistemas de aprendizagem podem aumentar a sua eficiência, através do conhecimento adquirido da experiência anterior. O objetivo destes sistemas é estudar de que forma a aprendizagem pode ser flexível, de acordo com o âmbito do estudo (Vilalta and Drissi, 2002).

O conceito *meta-learning* foi introduzido em problemas de seleção de algoritmo por Rice (1976), tendo desenvolvido uma *framework* denominada por problema de seleção de algoritmo (ASP - *Algorithm Selection Problem*). Esta framework inclui quatro componentes:

- Espaço do problema (P) - conjuntos de dados do problema.
- Espaço das características (F) - medidas que caracterizam P.
- Espaço do algoritmo (A) - conjunto de algoritmos que podem solucionar os problemas de P.
- Medida de desempenho (Y) - medidas de desempenho do algoritmo: exatidão, rapidez ou outras, dependendo do caso.

Smith-Miles (2008) sugeriu uma definição formal para o problema de seleção de algoritmo:

Problema de Seleção de Algoritmo - Dada uma instância $x \in P$ com as características $f(x) \in F$, construir um mapeamento de seleção $S(f(x))$ no espaço A, de modo que o algoritmo selecionado $\alpha \in A$ maximize o desempenho do mapeamento $y(\alpha(x)) \in Y$.

O grande desafio do ASP é identificar o mapeamento de seleção S, dado que a *framework* apresentada por Rice (1976) não especifica como se obtém S.

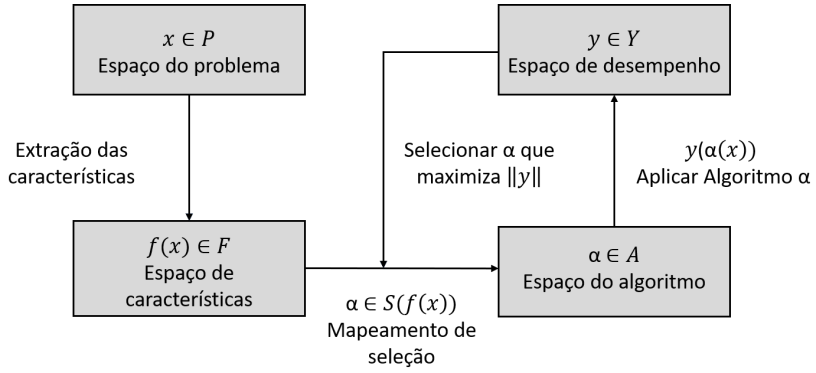

Figura 2.1: *Framework* proposto por Rice para o problema de seleção de algoritmo.

Aplicando a metodologia *ASP* no contexto da seleção de modelos de previsão, o problema foi definido por Talagala et al. (2018):

Problema de Seleção do Modelo de Previsão - Dada uma série temporal $x \in P$ com as características $f(x) \in F$, construir um mapeamento de seleção $S(f(x))$ no espaço A, de modo a que o algoritmo selecionado $\alpha \in A$ minimize medida de erro de previsão $y(\alpha(x)) \in Y$ que foi calculada para o conjunto de teste da série temporal.

2.3 Máquinas de Vetores de Suporte

As máquinas de vetores de suporte são algoritmos supervisionados de *machine learning* aplicados em problemas de classificação e análise de regressão (Hejazi et al., 2015).

Estes métodos introduzidos por Cortes and Vapnik (1995) têm sido aplicados com sucesso em diversas áreas. A sua popularidade está relacionada com a simplicidade do modelo e a obtenção do ótimo global em vez de um ótimo local (Arana-Daniel et al., 2018). A complexidade reduzida dos modelos obtidos conduz a uma maior eficiência na fase de teste, uma vez que permite contornar problemas de *overfitting* que estão associados à aproximação excessiva do modelo ao conjunto treino na determinação dos seus parâmetros (Gholami and Fakhari, 2017).

No contexto de problemas de classificação, as SVMs têm como finalidade obter uma função de separação, também conhecida por hiperplano, dos dados em duas ou mais classes. O hiperplano encontrado deverá não só explicar bem o conjunto de treino como permitir uma boa capacidade de generalização do modelo (Grossmann and Rinderle-Ma, 2015).

Na secção que se segue será inicialmente feita uma análise às SVMs lineares e não lineares para casos de classificação binária. Posteriormente, iremos abordar os casos de classificação com mais de duas classes.

2.3.1 SVMs Lineares

As SVMs lineares são algoritmos que permitem definir fronteiras lineares de separação de um conjunto de dados em duas classes.

De seguida, veremos como estes métodos de classificação são formulados para casos de conjunto de dados linearmente separáveis e de que forma podem ser estendidos para casos de dados não linearmente separáveis.

2.3.1.1 SVMs Lineares de Margens Rígidas

Consideremos um conjunto de N dados de treino x_i $(i = 1, 2, ..., N)$ onde $x_i \in \mathbb{R}^n$, e os respetivos rótulos $y_i \in Y$ sendo $Y = \{-1, +1\}$.

O objetivo das SVMs lineares é encontrar um hiperplano linear de separação para o conjunto de dados de treino x_i que permite maximizar a margem entre os dados pertencentes a cada uma das classes y_i, a fim de obter um modelo com uma boa capacidade de generalização (Gholami and Fakhari, 2017). O hiperplano será definido pela seguinte equação (ver Figura 2.2):

$$w^T x + b = 0, \tag{2.1}$$

onde w é um vetor de dimensão N normal ao hiperplano e b um escalar. O hiperplano está relacionado com duas margens que controlam a separabilidade dos dados, sendo definidas por:

$$\begin{cases} w^T x + b \geq +1 & se \quad y_i = +1 \\ w^T x + b \leq -1 & se \quad y_i = -1 \end{cases} \tag{2.2}$$

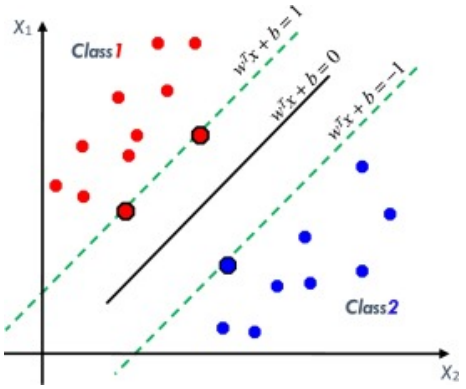

Figura 2.2: Representação de uma classificação binária usando SVMs lineares com margens rígidas - fonte: Gholami and Fakhari (2017).

O hiperplano ótimo será então definido com base na maximização da distância, d, entre as margens recorrendo à seguinte equação:

$$d\ (w,b;x) = \frac{|(w^T x+b-1)-(w^T x+b+1)|}{\|w\|} = \frac{2}{\|w\|}. \tag{2.3}$$

A partir da equação anterior, verifica-se que a maximização da distância d pode ser feita através da minimização de $\|w\|$. O presente problema de minimização é designado por problema primal e será formulado da seguinte forma:

$$\begin{aligned} &\min_{w,b} = \frac{1}{2}\ \|w\|^2 \\ &s.a \quad y_i\ (w^T x+b) \geq 1. \end{aligned} \tag{2.4}$$

No caso das SVMs de margens rígidas, é necessário garantir que não haja dados de treino entre as margens de separação das classes (linearmente separáveis). Esta limitação é assegurada pela restrição $y_i\ (w^T x+b) \geq 1$ representada no problema de otimização (2.4).

O presente problema pode ser resolvido recorrendo à minimização da seguinte função Lagrangiana que inclui a restrição anterior na função objetivo:

$$L\ (w,b,\alpha) = \frac{1}{2}\ \|w\|^2 - \sum_{i=1}^{N} \alpha_i\ (\ y_i\ (w^T x_i+b)-1). \tag{2.5}$$

A minimização da função Lagrangiana implicará a maximização de α_i e a minimização de w e b. Por conseguinte, o ponto obtido terá que obedecer às seguintes condições Karush–Kuhn–Tucker (KKT):

$$\frac{\partial L}{\partial w} = 0 \quad \Rightarrow \quad w = \sum_{i=1}^{N} \alpha_i x_i y_i \tag{2.6}$$

$$\frac{\partial L}{\partial b} = 0 \quad \Rightarrow \quad \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{2.7}$$

Substituindo as equações (2.6) e (2.7) na equação (2.5) obtém-se uma nova formulação denominada por forma dual:

$$\begin{aligned} \max \quad & L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ s.a \quad & \begin{cases} \alpha_i \geq 0 \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.8}$$

O problema também deverá respeitar uma outra condição de KKT que é a seguinte:

$$\alpha_i \; (\; y_i \; (w^T x_i + b) - 1 \;) = 0. \tag{2.9}$$

Observando a equação (2.9), é possível constatar que α_i é maior que 0, apenas para os dados que estão sobre as margens de decisão. Estes pontos designam-se por vetores de suporte, sendo os únicos que interferem na determinação da equação do hiperplano ótimo, uma vez que para os restantes pontos α_i é igual a 0 (Burges, 1998).

A partir da resolução do problema (2.8) é possível determinar os vetores de suporte e os respetivos dados de entrada. O valor de w é obtido recorrendo à equação (2.6), enquanto que o parâmetro b é calculado a partir da média dos valores de b para todos os vetores de suporte. A equação seguinte apresenta o cálculo de b, onde N_{sv} é o número de vetores de suporte e SV é o conjunto de vetores de suporte:

$$b^* = \frac{1}{N_{sv}} \sum_{x_j \in SV} (y_j - w^T x_j). \tag{2.10}$$

Na maior parte dos casos os dados não são linearmente separáveis. Contudo, em alguns casos é possível aplicar SVMs lineares de forma eficaz.

2.3.1.2 SVMs Lineares de Margens Suaves

As SVMs de margens suaves tratam-se de uma extensão das SVMs com margens rígidas, onde é introduzida uma variável de folga ξ_i que mede a distância entre o ponto mal classificado e a margem da respetiva classe atribuída, sendo $\xi_i \geq 0$ (ver Figura 2.3). De forma a minimizar os erros ξ_i para o conjunto de dados de treino, o problema (2.4) é reformulado da seguinte forma:

$$\begin{aligned} \min_{w,b,\xi} & = \frac{1}{2} \; \|w\|^2 + C \sum_{i=1}^{N} \xi_i \\ s.a \quad & y_i \; (w^T x + b) \geq 1 - \xi_i \end{aligned} \tag{2.11}$$

O parâmetro C é uma constante positiva que especifica o *trade-off* entre a maximização da margem e a minimização do número de erros de classificação. Quanto maior for o valor escolhido para C, maior será a tolerância às violações das margens e, por conseguinte, a margem entre as classes será maior (Casella et al., 2006). A restrição apresentada acima permite que os dados possam surgir entre margens, ao contrário do que acontecia no caso anterior.

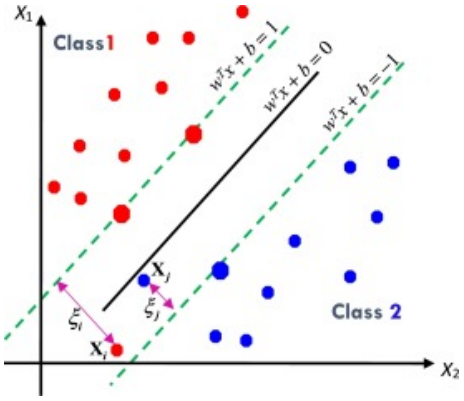

Figura 2.3: Representação de uma classificação binária usando SVMs lineares com margens suaves - fonte: Gholami and Fakhari (2017).

A resolução deste problema é semelhante à anterior, aplicando-se a função Lagrangiana e igualando as suas derivadas a zero. A formulação do novo problema para encontrar o hiperplano ótimo será a seguinte:

$$\begin{aligned} \max \quad & L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j x_i^T x_j \\ s.a \quad & \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.12}$$

Tal como se pode verificar, a formulação do problema para este caso é muito idêntica à do caso das SVMs de margens rígidas, sendo que apenas difere na restrição para α_i que é forçado a ser inferior ou igual a C.

2.3.2 SVMs Não Lineares

As SVMs lineares são eficazes em casos de conjuntos de dados linearmente separáveis ou com uma distribuição aproximadamente linear. Caso contrário, a sua aplicação não é adequada, uma vez que o modelo perderá a sua capacidade de generalização que permite obter bons resultados no período de teste (Gholami and Fakhari, 2017).

De forma a conseguir resolver problemas não lineares, as SVMs mapeiam o conjunto de treino do espaço original (espaço das entradas) para um espaço de maior dimensão denominado por espaço de características (*Hilbert space*). O mapeamento de x_i para o espaço de características é feito por meio de uma função transformação $\Phi(x)$, permitindo que a classificação do conjunto de treino possa ser feita por uma função de decisão linear. Para encontrar o hiperplano ótimo aplica-se uma SVM linear de margens suaves, o que resulta num problema de maximização idêntico a (2.12):

$$\begin{aligned} \max \quad & L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j (\Phi(x_i) \cdot \Phi(x_j)) \\ s.a \quad & \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^{N} \alpha_i y_i = 0 \end{cases} \end{aligned} \tag{2.13}$$

Tal como podemos verificar, a única informação necessária do mapeamento é o cálculo dos produtos internos $\Phi(x_i) \cdot \Phi(x_j)$. Porém, para um espaço de características com uma dimensão muito grande, o cálculo dos produtos internos será, a nível computacional, excessivamente dispendioso (Casella et al., 2006). No sentido de tornar o processo mais rápido, substitui-se o produto interno $\Phi(x_i) \cdot \Phi(x_j)$ por uma função designada por Kernel. Contudo, isto só é válido se a função vetorial não linear $g(x)$ obedecer à seguinte condição de Mercer:

$$\int K(x_i, x_j) g(x_i) g(x_j) dx_i dx_j \geq 0 \tag{2.14}$$

Existem várias funções de Kernel, sendo que as mais utilizadas são apresentadas na Tabela 2.1. O parâmetro γ, presente nas funções de Kernel exceto na função linear, trata-se de uma constante que determina o grau de similaridade entre dois pontos. Quando o valor de γ é baixo significa que dois pontos serão considerados similares mesmo que estejam distantes. Por outro lado, nos casos em que o valor de γ é elevado, dois pontos serão considerados idênticos apenas se estiverem próximos um do outro. A determinação do valor de γ é fundamental para uma boa performance da SVM, visto que, se o parâmetro γ for muito baixo, o classificador não irá captar a complexidade dos dados de treino, enquanto que, se o valor de γ for muito alto, o modelo será excessivamente complexo, podendo perder a sua capacidade de generalização e, por sua vez, não terá um bom desempenho na fase de teste.

Tabela 2.1: Funções de Kernel mais comuns.

Tipo de classificador	Função de Kernel
Linear	$x_i \cdot x_j$
Polinomial de grau d	$(\gamma x_i \cdot x_j + \text{coef0})^d$
Radial	$\exp(-\gamma \lVert x_i - x_j \rVert^2), \gamma > 0$
Sigmoide	$\tanh(\gamma(x_i \cdot x_j) + \text{coef0})$

Sabendo que o hiperplano é dado por:

$$f(x) = w^T \Phi(x) + b, \tag{2.15}$$

é necessário determinar os parâmetros w e b. No entanto, a utilização das funções de Kernel permite que não seja necessário o cálculo direto de w. Considerando que w é definido no espaço de características pela seguinte equação:

$$w = \sum_{i=1}^{N} y_i \alpha_i \Phi(x_i), \tag{2.16}$$

substituindo a equação referente a w na Equação (2.10), o parâmetro b pode ser escrito da seguinte forma:

$$b = y_i - \sum_{i,j=1}^{N_{sv}} \alpha_i y_i \, K(x_i, x_j). \tag{2.17}$$

Por fim, substituindo na Equação (2.15), o parâmetro w pela expressão dada na Equação (2.16), tem-se:

$$f(x) = \sum_{i=1}^{N} y_i \alpha_i \, K(x, x_i) + b \tag{2.18}$$

2.3.3 Classificação Multiclasse

Originalmente, as SVMs foram criadas para problemas de classificação binária, porém também podem ser aplicadas a problemas com mais de duas classes. Este tipo de problemas são decompostos em vários problemas binários, através da utilização de um dos seguintes métodos: *One-Against-All* (OAA) ou *One-Against-One* (OAO) (Hejazi et al., 2015).

A abordagem OAA decompõe o problema multiclasse em K problemas binários, onde cada problema compara uma das K classes com as restantes $K-1$ classes. A classe selecionada será codificada por $+1$ e as restantes por -1. Utilizando este método para a classificação de uma observação de teste x^*, diferentes SVMs poderão retornar o valor $+1$, indicando que a mesma observação pode pertencer a várias classes. Para solucionar este problema de ambiguidade, a classe atribuída a x^* será a classe k na qual a SVM produz o maior valor da função de decisão:

$$y^* = arg\max_k \; w_k^T x^* + b_k. \tag{2.19}$$

O método OAO constrói $\binom{K}{2}$ SVMs para comparar cada par de classes (k e k') em cada SVM, sendo que k toma o valor $+1$ e k' o valor -1. A classe atribuída a x^* será a classe que foi selecionada mais vezes no conjunto de $\binom{K}{2}$ SVMs (Casella et al., 2006).

2.4 Modelos de Previsão

Os modelos de previsão são ferramentas bastante úteis para empresas, dado que permitem um melhor planeamento das suas atividades. Os métodos de previsão procuram prever eventos futuros com base nos padrões e nas relações existentes no histórico de dados.

Existem vários modelos de previsão, sendo que a seleção do modelo a aplicar dependerá de múltiplos fatores como: o volume de dados disponível, os padrões presentes nos dados e o horizonte temporal das previsões (Hyndman and Athanasopoulos, 2018).

Note-se que a notação $\hat{y}_{t+h|t}$ será utilizada nas próximas secções e refere-se à previsão da variável y para o instante $t+h$, considerando a informação disponível até ao instante t, onde h é o número de passos à frente associado à previsão.

2.4.1 Modelos ETS

A presente secção está dividida em duas partes. Na primeira parte serão abordados vários métodos de alisamento exponencial e sua aplicação na previsão de séries temporais com diferentes características. Posteriormente, serão apresentados modelos de espaço de estados para os métodos de alisamento exponencial.

Os métodos de alisamento exponencial foram introduzidos no final dos anos 1950 por Brown (1959), Holt (1957) e Winters (1960). Estes métodos prevêem valores futuros através de combinações ponderadas de valores passados, onde os pesos atribuídos decrescem exponencialmente com a antiguidade das observações, o que explica a designação alisamento exponencial (Hyndman and Athanasopoulos, 2018).

A seleção do método mais adequado é feita com base nos principais padrões existentes nas séries temporais (tendência e sazonalidade) e na forma como estes se expressam no modelo de alisamento exponencial (aditiva ou multiplicativa).

2.4.1.1 Alisamento Exponencial Simples

O método de alisamento exponencial simples (SES - *Simple Exponential Smoothing*) é o método mais simples dentro do conjunto de métodos de alisamento exponencial. Esta abordagem é apropriada para a previsão de séries que não apresentem nem tendência, nem sazonalidade.

A metodologia SES atribui um peso maior às observações mais recentes, podendo ser escrita da seguinte forma:

$$\hat{y}_{t+1|t} = \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-2} + \cdots, \tag{2.20}$$

onde $0 \leq \alpha \leq 1$ é o parâmetro de alisamento que controla o decréscimo exponencial dos pesos associados às observações. Quanto menor for α, maiores serão os pesos das observações mais antigas. Por outro lado, se o valor de α for elevado, será atribuído um maior peso às observações mais recentes. Quando $\alpha = 1$ a previsão é igual à previsão do método naïve, $\hat{y}_{T+1|T} = y_T$.

É possível representar o método SES de duas formas que resultam na Equação (2.20).

Forma de média ponderada: A previsão para $t+1$ é a média ponderada entre a observação mais recente y_t e a previsão anterior $\hat{y}_{t|t-1}$:

$$\hat{y}_{t+1|t} = \alpha y_t + (1-\alpha)\hat{y}_{t|t-1}, \;\; t = 1,2,...,T \tag{2.21}$$

No processo de cálculo das previsões admite-se que a previsão para o primeiro instante $\hat{y}_{1|0}$ é igual ao valor inicial denominado por ℓ_0. A previsão para o instante $T+1$ pode então ser escrita do seguinte modo:

$$\hat{y}_{T+1|T} = \sum_{i=0}^{T-1} \alpha(1-\alpha)^i y_{T-i} + (1-\alpha)^T \ell_0 \tag{2.22}$$

Forma de componente: A representação sob forma de componente inclui uma equação da previsão e uma equação de alisamento para cada componente existente no método, sendo que existem três tipos de componentes: nível da série ℓ_t, tendência b_t e sazonalidade s_t. O método SES é representado em (2.24), considerando, apenas, a componente de nível ℓ_t.

$$\text{Equação da previsão} \quad \hat{y}_{t+h|t} = \ell_t \tag{2.23}$$

$$\text{Equação do nível} \quad \ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}, \tag{2.24}$$

para $t = 1,2,...,T$, onde ℓ_t é o nível da série no instante t.

Sendo a previsão para o instante $T+1$ igual ao último nível estimado ℓ_T, a previsão para um número de passos h é dada por:

$$\hat{y}_{T+h|T} = \hat{y}_{T+1|T} = \ell_T, \quad h = 2,3,... \tag{2.25}$$

Para a implementação do SES, é necessário estimar α e ℓ_0. Os valores de α e ℓ_0 podem ser obtidos a partir da minimização da soma dos erros quadráticos das previsões com $h = 1$. A função de minimização será a seguinte:

$$min\ SEQ = \sum_{t=1}^{T}(y_t - \hat{y}_{t|t-1}) = \sum_{t=1}^{T} e_t^2 \tag{2.26}$$

Na seleção do valor de ℓ_0 também é comum definir que $\ell_0 = y_1$.

2.4.1.2 Método de Tendência Linear de Holt

O método de tendência linear de Holt trata-se de uma extensão do método SES que permite obter previsões de séries que apresentam tendência. O cálculo das previsões por este método é baseado numa equação de previsão e em duas equações de alisamento (uma para o nível e outra para a tendência):

$$\text{Equação da previsão} \quad \hat{y}_{t+h|t} = \ell_t + hb_t \tag{2.27}$$

$$\text{Equação do nível} \quad \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{2.28}$$

$$\text{Equação da tendência} \quad b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}, \tag{2.29}$$

onde l_t é o nível estimado da série para instante t, b_t é a tendência estimada da série para instante t, α é o parâmetro de alisamento para o nível ($0 \leq \alpha \leq 1$), β é o parâmetro de alisamento para a tendência ($0 \leq \beta^* \leq 1$) e h é o passo da previsão.

Ao contrário do que se verifica no método SES, as previsões do presente método são uma função linear de h, uma vez que a previsão de $h-$passos à frente $\hat{y}_{T+h|T}$ é igual à soma do último valor estimado para l_t com o produto entre h e a última estimativa de b_t.

2.4.1.3 Método de Tendência Amortecida

As previsões geradas pelo método anterior apresentam uma tendência constante, indefinidamente crescente ou decrescente. Este tipo de métodos tende a ter problemas relacionados com *overprediction*, nomeadamente, para horizontes temporais longos, o que levou Gardner (1985) a propor uma alteração ao método linear de Holt, que permite "amortecer" a tendência (Hyndman and Athanasopoulos, 2018). O novo método pode ser escrito sob a seguinte forma:

$$\text{Equação da previsão} \quad \hat{y}_{t+h|t} = \ell_t + \phi_h b_t \tag{2.30}$$

$$\text{Equação do nível} \quad \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1}) \tag{2.31}$$

$$\text{Equação da tendência} \quad b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}, \tag{2.32}$$

para $t = 1,2,...,T$, onde $0 < \phi < 1$ é o parâmetro de amortecimento da tendência e $\phi_h = \phi + \phi^2 + \cdots + \phi^h$. É importante destacar que as previsões de longo prazo são constantes, uma vez que convergem para $\ell_T + \phi b_t/(1-\phi)$ quando $h \to \infty$, para $0 < \phi < 1$.

2.4.1.4 Método Sazonal de Holt-Winters

O método sazonal de Holt-Winters trata-se de uma extensão do método linear de Holt que permite captar a componente de sazonalidade das séries. Este método estabelece uma equação de previsão e três equações de alisamento: uma para nível ℓ_t, uma para tendência b_t e uma para a sazonalidade s_t.

O método de Holt-Winters apresenta duas variantes: método aditivo e método multiplicativo. O método aditivo é aplicado nos casos em que a sazonalidade varia de um modo, aproximadamente, constante ao longo do tempo. Neste método, a sazonalidade é expressa de forma absoluta nas unidades da série e a equação de nível é sazonalmente ajustada através da subtração da componente sazonal. A forma de componente para o método de Holt-Winters aditivo pode, então, ser escrita do seguinte modo:

$$\text{Equação da previsão} \quad \hat{y}_{t+h|t} = \ell_t + hb_t + s_{t+h-m(k+1)} \tag{2.33}$$

$$\text{Equação do nível} \quad \ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{2.34}$$

$$\text{Equação da tendência} \quad b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{2.35}$$

$$\text{Equação da sazonalidade} \quad s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}, \tag{2.36}$$

onde γ é o parâmetro de alisamento para a componente de sazonalidade ($0 \leq \gamma \leq 1$), m é o período de sazonalidade e k é o valor inteiro não superior a (h - 1)/m que garante que as estimativas dos índices sazonais usadas para a previsão provêm do último ano da série temporal.

Por outro lado, o método de Holt-Winters multiplicativo é adequado quando as flutuações da componente sazonal variam proporcionalmente com o nível da série. No método multiplicativo, a componente sazonal é expressa em termos relativos e a série é sazonalmente ajustada, dividindo-a

pela componente de sazonalidade (Hyndman and Athanasopoulos, 2018). O método de Holt-Winters multiplicativo pode ser formulado da seguinte maneira:

$$\text{Equação da previsão} \quad \hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t+h-m(k+1)} \tag{2.37}$$

$$\text{Equação do nível} \quad \ell_t = \alpha(\frac{y_t}{s_{t-m}}) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \tag{2.38}$$

$$\text{Equação da tendência} \quad b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1} \tag{2.39}$$

$$\text{Equação da sazonalidade} \quad s_t = \gamma(\frac{y_t}{\ell_{t-1} + b_{t-1}}) + (1-\gamma)s_{t-m}, \tag{2.40}$$

2.4.1.5 Taxonomia dos Métodos de Alisamento Exponencial

Os métodos de alisamento exponencial não estão circunscritos apenas aos métodos mencionados anteriormente. Considerando todas as combinações possíveis da tendência e da sazonalidade, existem 15 métodos diferentes, tal como podemos verificar na Tabela 2.2. Cada método de alisamento exponencial é identificado por um par de letras (T,S) que especificam, respetivamente, o tipo de tendência e de sazonalidade. As componentes de tendência e a sazonalidade terão as seguintes possibilidades: T = {N, A, A_d, M, M_d} e S = {N, A, M}.

Tabela 2.2: Classificação dos métodos de alisamento exponencial.

	Sazonalidade		
	N	A	M
Tendência	(Nenhuma)	(Aditiva)	(Multiplicativa)
N (Nenhuma)	(N,N)	(N,A)	(N,M)
A (Aditiva)	(A,N)	(A,A)	(A,M)
A_d (Amortecida aditiva)	(A_d,N)	(A_d,A)	(A_d,M)
M (Multiplicativa)	(M,N)	(M,A)	(M,M)
M_d (Amortecida multiplicativa)	(M_d,N)	(M_d,A)	(M_d,M)

No presente caso de estudo, apenas iremos considerar 9 métodos de alisamento exponencial, pois serão excluídos os métodos com tendência multiplicativa, uma vez que, geralmente, produzem más previsões (Hyndman and Athanasopoulos, 2018).

A Tabela 2.3 apresenta as fórmulas recursivas para cada método, indicando a equação de previsão e as equações de alisamento.

Tabela 2.3: Formulas recursivas para os métodos de alisamento exponencial - adaptado de Hyndman and Athanasopoulos (2018).

		Sazonalidade	
Tendência	N	A	M
N	$\hat{y}_{t+h\|t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m} + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m} + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1} + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$
A_d	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t+h-m(k+1)}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1}) + (1-\gamma)s_{t-m}$

2.4.1.6 Modelos de Espaço de Estados

Os métodos apresentados anteriormente limitam-se a produzir previsões pontuais. Nesta secção, apresentamos modelos de espaço de estados, baseados nos métodos de alisamento exponencial, que permitem não só gerar previsões pontuais, mas também construir intervalos de confiança e utilizar critérios objetivos de seleção de modelos (Hyndman and Athanasopoulos, 2018).

Cada modelo de espaço de estados compreende uma equação de medida, que descreve os dados observados e uma ou mais equações de estado que explicam o comportamento das componentes ou estados (nível, tendência, sazonalidade) ao longo do tempo.

Para cada método de alisamento exponencial existem dois modelos diferentes: um modelo com erros aditivos e outro com erros multiplicativos. De forma a ser possível distinguir estes dois tipos de modelos, cada modelo será identificado por 3 letras (E,T,S) que especificam o tipo de cada componente (erro, tendência e sazonalidade).

Considerando todas as combinações possíveis, existem 18 modelos ETS distintos, sendo que cada componente tem as seguintes possibilidades: E = $\{A, M\}$, T = $\{N, A, A_d\}$ e S = $\{N, A, M\}$.

ETS(A,A,N): Método de Tendência Linear de Holt com Erros Aditivos

Assumindo que se conhece os parâmetros do método de tendência linear de Holt, a previsão a um passo de y_t pode ser definida por $\hat{y}_{t|t-1} = \ell_{t-1} + b_{t-1}$. Sabendo que o erro ε_t, associado à previsão a 1-passo, é igual à diferença entre y_t e $\hat{y}_{t|t-1}$, então a equação de medida pode ser escrita do seguinte modo (Hyndman et al., 2008):

$$y_t = \ell_{t-1} - b_{t-1} + \varepsilon_t. \tag{2.41}$$

Substituindo a Equação (2.41) nas Equações (2.28) e (2.28) obtém-se as seguintes equações de estado:

$$\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t, \tag{2.42}$$

$$b_t = b_{t-1} - \beta\varepsilon_t, \tag{2.43}$$

onde $\beta = \alpha\beta^*$.

O termo de erro ε_t é independente e identicamente distribuído, seguindo uma distribuição gaussiana de média 0 e variância σ^2: $\varepsilon_t \sim \text{NID}(0, \sigma^2)$, onde NID significa "Normal e independentemente distribuído".

As equações, apresentadas anteriormente, juntamente com a distribuição de ε_t constituem um modelo de espaço de estados inovativos para o método linear de Holt. A designação "inovativos" advém do facto de todas as equações do modelo utilizarem o mesmo erro aleatório ε_t (Hyndman and Athanasopoulos, 2018).

ETS(M,A,N): Método Linear de Holt com Erros Multiplicativos

No caso do modelo ETS(M,A,N), os erros ε são expressos de forma relativa, sendo dados pela seguinte expressão:

$$\varepsilon_t = \frac{y_t - (\ell_{t-1} + b_{t-1})}{\ell_{t-1} + b_{t-1}}. \tag{2.44}$$

Seguindo um raciocínio similar ao que foi apresentado no modelo anterior, o modelo de espaço de estados inovativos subjacente ao método linear de Holt com erros multiplicativos é dado por:

$$y_t = (\ell_{t-1} - b_{t-1}) + (1 + \varepsilon_t), \tag{2.45}$$

$$\ell_t = (\ell_{t-1} + b_{t-1}) + (1 + \alpha\varepsilon_t), \tag{2.46}$$

$$b_t = b_{t-1} - \beta(\ell_{t-1} + b_{t-1})\varepsilon_t, \tag{2.47}$$

onde $\beta = \alpha\beta^*$ e $\varepsilon_t \sim \text{NID}(0, \sigma^2)$.

Os modelos ETS para os restantes métodos são obtidos a partir de uma abordagem semelhante à que foi utilizada para o método linear de Holt. As Tabelas A.1 e A.2 apresentam as equações para todos modelos ETS.

2.4.1.7 Estimação dos Parâmetros e dos Estados Iniciais

Os parâmetros (α, β, γ e ϕ) e os estados iniciais (ℓ_0, b_0, s_0, s_{-1},..., s_{-m+1}) do modelo podem ser obtidos a partir da maximização da verosimilhança do modelo.

A função de verosimilhança é baseada na densidade do vetor da série y e depende do vetor θ que contém os parâmetros do modelo, do vetor x_0 que inclui os estados iniciais e da variância residual σ^2. Sabendo que ε_t segue uma distribuição gaussiana, a verosimilhança do modelo pode ser apresentada sob a seguinte forma (Hyndman et al., 2008):

$$\mathscr{L}(\theta, x_0, \sigma^2 | y) = (2\pi\sigma^2)^{-n/2} \left| \prod_{t=1}^{n} r(x_{t-1}) \right|^{-1} \exp\left(-\frac{1}{2} \sum_{t=1}^{n} \varepsilon_t^2 / \sigma^2 \right), \tag{2.48}$$

por sua vez, a respetiva função logarítmica é dada por:

$$\log \mathscr{L} = -\frac{n}{2}\log(2\pi\sigma^2) - \sum_{t=1}^{n}\log|r(x_{t-1})| - \frac{1}{2}\sum_{t=1}^{n}\varepsilon_t^2/\sigma^2. \tag{2.49}$$

O parâmetro σ^2 pode ser estimado pela máxima verosimilhança, igualando a derivada parcial da Equação (2.49) a 0, resultando na seguinte expressão:

$$\hat{\sigma}^2 = n^{-1}\sum_{t=1}^{n}\varepsilon_t^2. \tag{2.50}$$

O cálculo de $\hat{\sigma}^2$ permite eliminar σ^2 da Equação (2.48), obtendo-se:

$$\mathscr{L}(\theta, x_0|y) = (2\pi e\hat{\sigma}^2)^{-n/2}\left|\prod_{t=1}^{n} r(x_{t-1})\right|^{-1}, \tag{2.51}$$

por conseguinte,

$$-2\log\mathscr{L}(\theta, x_0|y) = c_n + n\log\left(\sum_{t=1}^{n}\varepsilon_t^2\right) + 2\log|r(x_{t-1})|, \tag{2.52}$$

onde c_n é uma constante dependente de n definida por: $c_n = n\log(2\pi e) - n\log(n)$.

Com base na equação anterior, as estimativas de máxima verosimilhança dos parâmetros θ e x_0 podem ser obtidas a partir da minimização da equação:

$$\mathscr{L}^*(\theta, x_0) = n\log\left(\sum_{t=1}^{n}\varepsilon_t^2\right) + 2\log|r(x_{t-1})|. \tag{2.53}$$

2.4.1.8 Critérios de Informação para Seleção de Modelos

Como foi referido anteriormente, os modelos ETS permitem usar critérios objetivos de seleção de modelos. A escolha do melhor modelo ETS para cada série, poderá ser baseada na minimização dos seguintes critérios: Critério de Informação de Akaike, Critério de Informação de Akaike corrigido e Critério de Informação Bayesiano.

O Critério de Informação de Akaike (AIC - *Akaike Information Criteria*) baseia-se na verosimilhança do modelo e pode ser escrito da seguinte forma:

$$\text{AIC} = -2\log\mathscr{L}(\theta, x_0|y) + 2k, \tag{2.54}$$

onde k é o número de parâmetros e estados estimados, incluindo a variância residual. Note-se que a adição do termo $2k$ tem como objetivo evitar problemas de overfitting que estão associados à seleção de modelos com um número excessivo de parâmetros (Cryer and Chan, 2009).

Quando as amostras são pequenas, o AIC tende a selecionar modelos com um número excessivo de parâmetros. Hurvich and Tsai (1989) propôs um critério de seleção adequado para amostras pequenas, denominado por Critério de Informação de Akaike corrigido (AICc), que é dado por:

$$\text{AICc} = \text{AIC} + \frac{2k(k+1)}{n-k-1}, \tag{2.55}$$

sendo n o tamanho da amostra.

Por último, o Critério de Informação Bayesiano (BIC - *Bayesian Information Criteria*), proposto por Schwarz (1978), pode ser expressado da seguinte modo:

$$\text{BIC} = \text{AIC} + k[\log(n) - 2]. \tag{2.56}$$

Comparativamente com o AIC, o BIC é um critério mais consistente, no entanto, não é tão eficiente assintoticamente (Hyndman et al., 2008).

2.4.2 Modelos ARIMA

Os modelos ARIMA (*Autoregressive Integrated Moving Average*) tratam-se de uma abordagem estatística bastante aplicada na previsão de séries temporais e têm como propósito descrever as autocorrelações dos dados (Hyndman and Athanasopoulos, 2018).

Esta classe de modelos tem a particularidade de aplicar transformações nas séries temporais até que se verifique que estas são estacionárias. As séries estacionárias são conjuntos de dados que têm média e variância constantes e uma covariância independente do tempo, dependendo somente do desfasamento temporal. Do ponto vista prático, este tipo de séries não exibem tendência nem sazonalidade, apresentando uma média e uma variância constantes ao longo do tempo.

No caso de as séries temporais originais não serem estacionárias, o modelo só começa a trabalhar com os dados, depois de convertê-los em estacionários. A conversão das séries é realizada através de transformações que permitem estabilizar a média e a variância. No caso de ser necessário estabilizar a média, utilizam-se a diferenciação, enquanto que a para a estabilização da variância, recorre-se à família de transformações de Box-Cox ou a outro tipo de transformações mais simples. É importante salientar que a diferenciação deve ser feita só após a estabilização da variância.

Existem dois tipos de diferenciação: a diferenciação simples e a diferenciação sazonal. A diferenciação simples consiste no cálculo de todas as diferenças entre observações consecutivas, traduzindo-se na seguinte equação:

$$y'_t = y_t - y_{t-1}, \tag{2.57}$$

onde y_{t-1} é o valor da observação com um desfasamento (*lag*) igual a 1 em relação y_t.

Por vezes, as séries continuam a não ser estáticas após a diferenciação, aplicando-se uma nova diferenciação (diferenciação de 2ª ordem) que pode ser escrita da seguinte forma:

$$y''_t = y'_t - y'_{t-1} = (y_t - y_{t-1}) - (y_{t-1} - y_{t-2}) \tag{2.58}$$

No caso de a diferenciação não ter o efeito pretendido, faz-se uma diferenciação novamente, até que os dados sejam estáticos.

Nas situações em que as séries apresentam uma componente de sazonalidade, deve aplicar-se a diferenciação sazonal que se trata da diferença entre uma observação e a sua homóloga em relação ao período sazonal anterior. A diferença sazonal é definida pela seguinte equação:

$$y'_t = y_t - y_{t-s}, \tag{2.59}$$

onde s representa o número de observações abrangidos pelo período sazonal.

Com intuito de simplificar a equação que explicita a diferenciação, introduziu-se um operador de atraso B que, por cada vez que é aplicado, produz um atraso de um instante nos dados. A Equação (2.61) pode ser reformulada da seguinte forma:

$$y'_t = y_t - By_t = (1-B)y_t \tag{2.60}$$

De uma forma geral, uma diferenciação de ordem d pode então ser definida pela seguinte expressão:

$$(1-B)^d y_t \tag{2.61}$$

Os modelos ARIMA(p, d, q) resultam da combinação de três tipos de processos: processos auto-regressivos AR(p), processos de médias móveis MA(q) e processos de diferenciação I(d).

Os modelos auto-regressivos prevêem uma variável de resposta y_t com base na combinação linear dos valores passados da mesma variável, ao passo que, nos modelos de médias móveis, a previsão da variável y_t é feita através da combinação linear dos erros passados (Cryer and Chan, 2009).

A junção dos três processos referidos anteriormente resulta na Equação (2.62) que representa um modelo ARIMA(p, d, q) não sazonal.

$$y'_t = c + \phi_1 y'_{t-1} + ... + \phi_p y'_{t-p} + \varepsilon_t + \theta_1 \varepsilon_{t-1} + ... + \theta_q \varepsilon_{t-q}, \tag{2.62}$$

onde p, d e q são valores inteiros não negativos, sendo d o número de diferenciações simples necessárias, $c = \mu(1 - \phi_1 - ... - \phi_p)$ e μ é a média da série diferenciada y'_t. Os termos $\phi_1 ... \phi_p$ referem-se aos parâmetros do modelo auto-regressivo de ordem p, $\theta_1 ... \theta_q$ são os parâmetros do modelo de médias móveis de ordem q e ε_t é um ruído branco gaussiano de média 0 e variância σ^2.

Nos casos em que se verifica a componente de sazonalidade, aplicam-se os modelos ARIMA(p, d, q)(P, D, Q)$_s$ sazonais que são compostos não só pelos termos incluídos no modelo não sazonal, mas também pelos termos sazonais (P,D,Q)$_s$. O modelo ARIMA(p, d, q)(P, D, Q)$_s$ pode ser escrito da seguinte forma:

$$\phi_p(B)\Phi_p(B^s)(1-B)^d(1-B^s)^D y_t = c + \theta_q(B)\Theta_Q(B^s)\varepsilon_t, \tag{2.63}$$

onde

$$\phi_p(B) = 1 - \phi_1 B - ... - \phi_p B^p,$$

$$\theta_q(B) = 1 + \theta_1 B + ... + \theta_q B^q,$$

$$\Phi_P(B^s) = 1 - \Phi_1 B^s - ... - \Phi_P B^{Ps},$$

$$\Theta_Q(B^s) = 1 + \Theta_1 B^s + ... + \Theta_Q B^{Qs},$$

sendo s o período de sazonalidade, D o grau de diferenciação sazonal, B o operador de atraso, $c = \mu(1-\phi_1 - ... - \phi_p)(1-\Phi_1 - ... - \Phi_P)$, μ é a média de $(1-B)^d(1-B^s)^D y_t$. $\phi_p(B)$ e $\theta_q(B)$ são os polinomiais regulares auto-regressivos e de médias móveis e $\Phi_P(B^s)$ e $\Theta_Q(B^s)$ os polinomiais sazonais auto-regressivos e de médias móveis (Wei, 2005).

Para aplicar os modelos ARIMA é necessário definir os valores dos parâmetros p, d, q, P, D, Q. Os valores de d e D podem ser obtidos através da análise da Função de Autocorrelação (ACF - *Autocorrelation Function*) e da Função de Autocorrelação Parcial (PACF - *Partial Autocorrelation Function*) da série temporal. O decréscimo lento da ACF e a queda brusca para 0 da PACF a partir da *lag* 1 sugere que é necessário mais diferenciações. A determinação adequada de d e D é bastante importante para evitar aplicação de um número excessivo de diferenciações que pode levar à introdução de dependência quando ela não existe (Shumway and Stoffer, 2011). A partir da análise do comportamento da ACF e PACF para série estacionária, determina-se os parâmetros p, q, P, Q, identificando-se os modelos candidatos.

Posteriormente, é necessário selecionar o melhor modelo entre os candidatos. A seleção será feita com base na minimização dos critérios de informação: AIC, AICc e BIC.

O AIC pode ser escrito da seguinte forma:

$$\text{AIC} = -2\log \mathscr{L}(\theta, \sigma^2|y) + 2(p+q+P+Q+k+1), \tag{2.64}$$

onde $\log \mathscr{L}(\theta, \sigma^2|y)$ é o logaritmo da verosimilhança dos dados, $k = 1$ se $c \neq 0$ e $k = 0$ se $c = 0$. De acordo com Box and Jenkins (1994), o logaritmo da verosimilhança dos dados $\log \mathscr{L}(\theta, \sigma^2|y)$ é dado por:

$$\log \mathscr{L}(\theta, \sigma^2|y) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma^2) - \sum_{t=1}^{n} \frac{\varepsilon_t^2}{2\sigma^2} \tag{2.65}$$

O critério AICc pode ser definido por :

$$\text{AICc} = \text{AIC} + \frac{2(p+q+P+Q+k+1)(p+q+P+Q+k+2)}{n-p-q-P-Q-2} \tag{2.66}$$

e o BIC é dado pela seguinte expressão:

$$\text{BIC} = \text{AIC} + [\log(n) - 2](p+q+P+Q+k+1). \tag{2.67}$$

A utilização dos critérios de informação mencionados será útil, apenas, na determinação dos melhores valores p, q, P, Q, uma vez que a diferenciação altera os dados e, por conseguinte, os valores dos critérios obtidos para os modelos com diferentes ordens diferenciação não serão comparáveis.

2.4.3 Modelo TBATS

Como vimos na secção 2.4.1, os modelos ETS consideram vários tipos de tendência e de sazonalidade. No entanto, estes modelos não são capazes de tratar séries com vários padrões de sazonalidade.

De Livera et al. (2011) desenvolveram um modelo mais flexível, denominado por TBATS (*Trigonometric Box-Cox transform, ARMA errors, Trend, and Seasonal components*), capaz de lidar com séries que apresentem múltiplas sazonalidades. Cada componente de sazonalidade é modelada por uma representação trigonométrica baseada em séries de Fourier. O modelo TBATS é identificado por TBATS$(\omega,\phi,p,q,\{m_1,k_1\},\{m_2,k_2\},...,\{m_T,k_T\})$ e tem a seguinte formulação:

$$y_t^{(\omega)} = \begin{cases} \frac{y_t^{(\omega)}-1}{\omega}, & \omega \neq 0 \\ \log y_t, & \omega = 0 \end{cases} \tag{2.68}$$

$$y_t^{(\omega)} = l_{t-1} + \phi b_{t-1} + \sum_{j=1}^{k_i} s_t^{(i)} + d_t \tag{2.69}$$

$$l_t = l_{t-1} + \phi b_{t-1} + \alpha d_t \tag{2.70}$$

$$b_t = (1-\phi)b + \phi b_{t-1} + \beta d_t \tag{2.71}$$

$$s_t^{(i)} = s_{t-m_i}^{(i)} + \gamma_i d_t \tag{2.72}$$

$$d_t = \sum_{i=1}^{p} \varphi_i d_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t, \tag{2.73}$$

$$s_t^{(i)} = \sum_{j=1}^{k_j} s_{j,t}^{(i)} \tag{2.74}$$

$$s_{j,t}^{(i)} = s_{j,t-1}^{(i)} \cos \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \sin \lambda_j^{(i)} + \gamma_1^{(i)} d_t \tag{2.75}$$

$$s_{j,t}^{*(i)} = -s_{j,t-1} \sin \lambda_j^{(i)} + s_{j,t-1}^{*(i)} \cos \lambda_j^{(i)} + \gamma_2^{(i)} d_t \tag{2.76}$$

onde $y_t^{(\omega)}$ representa a série y_t transformada por Box-Cox com parâmetro ω; $m_1, m_2, ..., m_T$ correspondem aos períodos sazonais da série; l_t é o nível no instante t; b_t é a tendência de curto-prazo

no instante t; b corresponde à tendência de longo-prazo; $s_t^{(i)}$ corresponde à componente sazonal de ordem i no instante t; d_t representa o processo ARMA(p,q); ε_t é um processo de ruído branco Gaussiano com média 0 e variância σ^2. Os parâmetros de alisamento são dados por α, β e γ_i e o parâmetro de amortecimento da tendência por ϕ para $i = 1,2,...,T$. A notação $s_{j,t}^{(i)}$ corresponde ao nível estocástico da componente sazonal de ordem i. O nível estocástico de crescimento da componente sazonal i é dado por $s_{j,t}^{*(i)}$ e descreve as oscilações de sazonalidade ao longo do tempo. O termo k_j é o número de harmónicos para a componente sazonal i e $\lambda_j^{(i)} = 2\pi j/m_i$. Na abordagem do problema, atribui-se valores pares a m_i quando $k_i = m_i/2$ e valores ímpares quando $k_i = (m_i - 1)/2$.

2.4.4 Lasso

Consideremos o modelo de regressão linear com p-regressores representado na Equação (2.77), onde $\beta = (\beta_0, \beta_1, \beta_2, ..., \beta_p)^T$ correspondem aos parâmetros do modelo e ε_i são os erros, os quais são mutuamente independentes e seguem uma distribuição normal de valor esperado nulo e variância σ^2:

$$y_i = \beta_0 + \sum_{j=1}^{p} \beta_j x_{ij} + \varepsilon_i \tag{2.77}$$

Um dos métodos utilizados para a determinação dos parâmetros é o método dos mínimos quadrados. Este método estima os valores dos parâmetros com base na minimização da soma dos erros quadráticos (RSS - Residual Sum of Squares):

$$RSS = \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 . \tag{2.78}$$

Geralmente, todos coeficientes estimados, a partir deste método, são diferentes de zero, o que torna difícil a interpretação do modelo quando o número de regressores é elevado. Outro problema associado ao método dos mínimos quadrados é o facto de não se obter uma solução única nos casos em que o número de regressores é maior que o tamanho do conjunto de dados, havendo múltiplas soluções que igualam a função objetivo a zero, pelo que muitas delas levam a problemas de *overfitting* (Hastie et al., 2015).

O método Lasso (*Least Absolute Shrinkage and Selection Operator*) tem um funcionamento semelhante ao do método dos mínimos quadrados. No entanto, este método adiciona um termo de penalização $\lambda \sum_{j=1}^{p} |\beta_j|$ que permite a redução da magnitude dos coeficientes do modelo, podendo alguns deles serem forçados a ser nulos para valores de λ suficientemente elevados. Os coeficientes obtidos pelo método Lasso minimizam

$$\sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^{p} |\beta_j| = RSS + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.79}$$

onde $\lambda \geq 0$ é um parâmetro de regularização que controla o impacto relativo de cada um dos termos (Casella et al., 2006). Quando $\lambda = 0$, o efeito do termo de penalização será nulo e os coeficiente estimados serão iguais aos que são produzidos pelo método dos mínimos quadrados. Por outro lado, o crescimento de λ provoca um aumento do impacto da penalização, podendo levar à exclusão de determinados regressores.

Este método apresenta outra formulação que é equivalente à anterior, a qual pode ser escrita do seguinte modo:

$$\begin{aligned} \min_{\beta} &= \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{p} \beta_j x_{ij} \right)^2 \\ s.a \quad & \sum_{j=1}^{p} |\beta_j| \leq s, \end{aligned} \tag{2.80}$$

onde $s \geq 0$ é um parâmetro que limita $\sum_{j=1}^{p} |\beta_j|$. Nos casos em que o valor s é baixo, o somatório ficará mais restringido e, consequentemente, os coeficientes do modelo serão mais baixos.

A seleção dos parâmetros λ e s é crucial para a determinação do melhor modelo e pode ser realizada recorrendo à técnica de validação cruzada. De uma forma sucinta, o procedimento consiste na escolha de um conjunto de valores para λ ou s e, posteriormente, são ajustados os respetivos modelos e calculadas as medidas de erro correspondentes. Seguidamente, seleciona-se o valor de λ ou s com a menor medida de erro. Por fim, o modelo é reajustado com todas as observações e com o valor de λ ou s escolhido (Casella et al., 2006).

A utilização do método LASSO pode ser benéfica por diversos motivos. Em primeiro lugar, este método tende apresentar um erro de previsão baixo. Outra vantagem, associada à sua aplicação, é a redução da complexidade do modelo através da diminuição do número de variáveis, facilitando a sua interpretação.

De facto, o método Lasso será especialmente adequado para a previsão de séries temporais com um elevado número de regressores em relação ao tamanho da amostra. Porém, é importante salientar que o seu desempenho tende a diminuir, quando o número de regressores diminui relativamente ao número de observações (Chan-Lau, 2017).

2.4.5 Análise de Componentes Principais

A análise de componentes principais (PCA - *Principal Components Analysis*) é bastante útil em problemas com um grande conjunto de variáveis independentes correlacionadas. Esta técnica permite resumir o conjunto original de variáveis com um menor número de variáveis, denominadas por componentes principais, mantendo a informação mais relevante dos dados (Abdi and Williams, 2010).

As componentes principais tratam-se de combinações lineares das variáveis originais. A primeira componente principal é a combinação linear normalizada das variáveis originais que explica a maior fração da variância dos dados (Casella et al., 2006). Considerando o conjunto de variáveis $\{X_1, X_2, ..., X_p\}$, a primeira componente principal pode ser definida por:

$$Z_1 = \phi_{11}X_1 + \phi_{21}X_2 + \cdots + \phi_{p1}X_p, \tag{2.81}$$

onde $\phi_{11}, ..., \phi_{p1}$ representam as correlações entre as variáveis originais e a componente principal e $\sum_{i=1}^{p} \phi_{i1}^2 = 1$.

A segunda componente principal Z_2 consiste na combinação linear de $X_1, X_2, ..., X_p$, que apresenta a maior variância amostral e que é ortogonal a Z_1, uma vez que não pode ser correlacionada com Z_1. A componente Z_2 pode ser escrita da seguinte forma:

$$Z_2 = \phi_{12}X_1 + \phi_{22}X_2 + \cdots + \phi_{p2}X_p. \tag{2.82}$$

É importante referir que a soma dos quadrados das correlações entre cada variável e todas as componentes principais é igual a 1, dado que cada ϕ_{ij}^2 indica a proporção da variância da variável i explicada pela componente j.

Após serem determinadas as componentes principais, é possível obter os valores das mesmas para as observações originais. Do ponto de vista geométrico, estes valores tratam-se das projeções das observações originais para as componentes principais obtidas.

2.4.6 Erros de Previsão

Na presente secção serão apresentas várias medidas de erro utilizadas para a avaliação da exatidão dos modelos de previsão.

Atualmente, existem modelos bastante complexos que permitem ajustar o modelo ao conjunto de dados de forma exaustiva, o que conduz a problemas de *overfitting* e, por sua vez, a um fraco desempenho das previsões futuras. Por esta razão, a precisão das previsões deve ser baseada no desempenho do modelo num novo conjunto dados, ao qual o modelo não foi ajustado. Para tal, o conjunto de dados é divido em duas porções: o conjunto de treino e o conjunto de teste. O conjunto de treino é utilizado na determinação dos parâmetros do método de previsão, enquanto que o conjunto de teste é usado para medir a exatidão do método.

Tipicamente, o conjunto de teste corresponde a 20% da amostra. No entanto, é recomendável que o conjunto de teste não seja superior ao horizonte temporal máximo associado à previsão (Hyndman and Athanasopoulos, 2018).

2.4.6.1 Erro de Previsão

O erro de previsão consiste na diferença entre o valor observado e a sua previsão. Considerando que os conjuntos de treino e de teste são, respetivamente, constituídos por T e M observações, o erro de previsão pode ser definido por:

$$e_{T+h} = y_{T+h} - \hat{y}_{T+h|T} \quad \text{para} \quad h = 1, 2, ..., M. \tag{2.83}$$

2.4.6.2 Medidas de Erro Dependentes de Escala

Estas medidas de precisão apresentam uma escala que depende da escala dos dados. Por essa razão, as medidas dependentes de escala serão úteis na comparação da performance dos diferentes modelos, apenas, quando estes são aplicados ao mesmo conjunto de dados (Hyndman and Koehler, 2006). Existem várias medidas dependentes de escala, sendo que as mais comuns são baseadas nos erros quadráticos e absolutos:

Erro Quadrático Médio (MSE)

$$MSE = \frac{1}{M}\sum_{t=1}^{M} e_t^2, \tag{2.84}$$

Raiz Quadrada do Erro Quadrático Médio (RMSE)

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{M}\sum_{t=1}^{M} e_t^2}, \tag{2.85}$$

Erro Absoluto Médio (MAE)

$$MAE = \frac{1}{M}\sum_{t=1}^{M} |e_t|. \tag{2.86}$$

O MAE é bastante popular, dado que é fácil de calcular e de compreender. Por outro lado, o RMSE também é muito utilizado, porém, este é mais sensível a *outliers* do que o MAE.

2.4.6.3 Medidas Baseadas em Erros Percentuais

Os erros percentuais ($p_t = e_t/y_t \times 100$) têm a vantagem de serem independentes da escala dos dados, permitindo a comparação do desempenho de modelos aplicados a conjuntos de dados de dimensões diferentes. As medidas mais usadas são:

Erro Percentual Absoluto Médio (MAPE)

$$MAPE = \frac{1}{M}\sum_{t=1}^{M} |p_t|, \tag{2.87}$$

Erro Percentual Absoluto Médio simétrico (MAPE)

$$sMAPE = \frac{1}{M}\sum_{t=1}^{M} \left(\frac{|y_t - \hat{y}_t|}{y_t + \hat{y}_t} \times 200 \right). \tag{2.88}$$

O MAPE tem a desvantagem de ser indefinido quando y_t toma um valor nulo. Além disso, esta medida apresenta valores muito elevados, quando y_t é próximo de 0. Outra desvantagem associada ao MAPE é o facto de penalizar mais os erros negativos, comparativamente com os

erros positivos. A medida de desempenho sMAPE surgiu no sentido de resolver este problema, contudo, à semelhança do MAPE, esta medida pode envolver denominadores próximos de 0, visto que, quando o valor de y_t é próximo de 0, é provável que $\hat{y}_t$ também seja.

2.4.6.4 Medidas Baseadas em Erros Relativos

Os erros relativos recorrem a um termo de comparação comum para atenuar as diferenças de escala. Estes erros consistem na divisão de cada erro de previsão pelo erro associado ao método de referência (o método de naïve é o mais comum). Considerando que o erro relativo é dado por $r_t = e_t/e_t^b$, onde e_t^b é o erro obtido pelo método de referência, define-se como:

Erro Absoluto Relativo Médio (MRAE)

$$MRAE = \frac{1}{M}\sum_{t=1}^{M}|r_t|, \tag{2.89}$$

Média Geométrica do Erro Absoluto Relativo (GMRAE)

$$GMRAE = \sqrt[M]{\prod_{t=1}^{M}|r_t|}. \tag{2.90}$$

É importante sublinhar que o cálculo de r_t impõe que e_t^b não pode ser nulo e $e_t \neq e_t^b$. Tal como as medidas baseadas em erros percentuais, estas medidas apresentam problemas quando o e_t^b toma um valor próximo de 0, tornando o valor de r_t muito elevado (Hyndman and Koehler, 2006).

2.4.6.5 Medidas Relativas

De forma a contornar os problemas evidenciados pelos erros relativos, podem ser utilizadas medidas relativas que estabelecem uma comparação entre o método aplicado e o método de referência, recorrendo a medidas de erro como: MAE e RMSE. Seguidamente, são apresentados alguns exemplos de medidas relativas, sendo N o número de séries temporais.

Erro Absoluto Médio Relativo (RelMAE)

$$RelMAE = \frac{MAE}{MAE^b} \tag{2.91}$$

$$MAE = \tfrac{1}{M}\textstyle\sum_{t=1}^{M}|e_t| \quad MAE^b = \tfrac{1}{M}\textstyle\sum_{t=1}^{M}|e_t^b|$$

Raiz Quadrada do Erro Quadrático Médio Relativo (RelRMSE)

$$RelRMSE = \frac{RMSE}{RMSE^b} \tag{2.92}$$

$$RMSE = \sqrt{\tfrac{1}{M}\textstyle\sum_{t=1}^{M}(e_t^2)} \quad RMSE^b = \sqrt{\tfrac{1}{M}\textstyle\sum_{t=1}^{M}(e_t^2)^b}$$

Média Geométrica do Erro Absoluto Médio Relativo (GMRelMAE)

$$GMRelMAE = \sqrt[N]{\prod_{i=1}^{N} \frac{MAE_i}{MAE_i^b}} = exp\left[\frac{1}{N}\sum_{i=1}^{N} log\left(\frac{MAE_i}{MAE_i^b}\right)\right] \tag{2.93}$$

De acordo com Fleming and Wallace (1986), a média geométrica é o tipo de média mais indicado para a análise dos resultados relativos aos rácios de referência, uma vez que atribui o mesmo peso a diferenças relativas recíprocas. Por exemplo, o peso dado, no caso de $MAE_i = MAE_i^b/2$, será igual ao que é atribuído no caso contrário. A média geométrica do RelMAE para N séries (podem ser de diferentes escalas) é dado pela Equação (2.93). Quando $\sum_{i=1}^{N} \log(MAE_i/MAE_i^b) < 0$ significa que o método em análise resulta, em média, em previsões melhores do que as do método de referência, quando $\sum_{i=1}^{N} \log(MAE_i/MAE_i^b) > 0$, quer dizer o oposto.

As medidas relativas revelam vantagens na sua interpretação, porém, este tipo de medidas requer a aplicação de vários modelos à mesma série temporal, podendo tornar-se um inconveniente nas situações em que o tempo é limitado.

2.4.6.6 Erros Escalados

Como vimos anteriormente, as medidas relativas e as medidas baseadas em erros relativos permitem mitigar as diferenças de escala dos dados. No entanto, ambas apresentam lacunas. Os erros relativos possuem uma distribuição probabilística com média indefinida e variância infinita. Relativamente às medidas relativas, estas só podem ser obtidas quando existem várias previsões para a mesma série temporal.

Hyndman and Koehler (2006) propuseram uma alternativa, para escalar o erro, baseada no MAE obtido pelo método naïve para o conjunto de treino. No caso de séries não sazonais, o erro escalado é definido por:

$$q_t = \frac{e_t}{\frac{1}{T-1}\sum_{i=2}^{T} |y_i - y_{i-1}|} \quad \text{para} \quad t = 1,2,3,...,M, \tag{2.94}$$

enquanto que para séries sazonais, o erro escalado pode ser escrito da seguinte maneira:

$$q_t = \frac{e_t}{\frac{1}{T-s}\sum_{i=s+1}^{T} |y_i - y_{i-s}|} \quad \text{para} \quad t = 1,2,3,...,M, \tag{2.95}$$

onde s é o número de períodos da sazonalidade e q_t é independente da escala, pois tanto o numerador como o denominador estão na mesma escala dos dados originais.

Deste modo, o erro escalado absoluto médio MASE pode ser definido por:

$$MASE = \frac{1}{M}\sum_{t=1}^{M} |q_t|. \tag{2.96}$$

Quando o valor do MASE obtido é inferior a 1, significa que o método utilizado produz previsões mais precisas do que as previsões geradas pelo método de naïve.

O MASE é considerado a medida mais robusta e abrangente para a avaliação da exatidão dos modelos (Hyndman and Koehler, 2006). Esta medida destaca-se pela sua resistência às diferenças de escala dos dados e por serem infinitas ou indefinidas, apenas, nas situações em que todas as observações são iguais.

2.5 Análise Crítica ao Estado da Arte

Em estudos anteriores, vários modelos estatísticos foram aplicados na previsão de vendas do setor do retalho, entre os quais os modelos de regressão (Van Donselaar et al., 2016), modelos autorregressivos de médias móveis (Feng Yun and Ma Jun-hai, 2008; Ramos et al., 2015; Pereira da Veiga et al., 2014) e modelos baseados em alisamento exponencial (Gao et al., 2011; Ramos et al., 2015; Pereira da Veiga et al., 2014).

Van Donselaar et al. (2016) analisou o impacto das promoções nas vendas de produtos com um reduzido ciclo de vida na indústria do retalho. Para a previsão da procura durante as promoções, foram testados vários modelos de regressão (linear e quadrática) e modelos de médias móveis. A partir da análise das medidas de erro RMSE e MAPE, verificou-se uma maior precisão dos modelos de regressão em relação aos modelos de médias móveis.

Relativamente aos modelos auto-regressivos de médias móveis, Feng Yun and Ma Jun-hai (2008) aplicaram um modelo ARMA (1,1) para realizar previsões da procura para o retalho, estabelecendo equações para determinar o efeito “bullwhip” que é muito comum neste setor de atividade. Os resultados dessa implementação foram satisfatórios, uma vez que as simulações provam que os modelos ARMA podem reduzir significativamente o efeito "bullwhip".

Gao et al. (2011) criaram um algoritmo para previsão da procura com base no método de alisamento exponencial Holt-Winters. Também foi proposto um procedimento para o calculo dos parâmetros do método Holt-Winters. O algoritmo de previsão é limitado em certo grau, uma vez que o método Holt-Winters depende da fiabilidade dos dados da procura. No entanto, foram utilizados métodos de suavização para minimizar o impacto dos dados com pouca fiabilidade. Para medir o desempenho do algoritmo, foi calculado o MSE, tendo sido obtido um valor aceitável, o que indica a aplicabilidade deste algoritmo ao setor de retalho.

Ramos et al. (2015) e Pereira da Veiga et al. (2014) compararam os modelos ARIMA com os modelos de alisamento exponencial, verificando-se que não existe uma diferença significativa em termos de desempenho. Pereira da Veiga et al. (2014) aplicou o método Holt-Winters multiplicativo e o modelo ARIMA na previsão de vendas no setor de retalho alimentar e, posteriormente, comparou os dois modelos. Esta comparação é motivada pelo facto do modelo ARIMA ser equivalente a uma grande parte dos modelos de alisamento exponencial, exceto o modelo Holt-Winters multiplicativo. Com base nos indicadores de desempenho MAPE e estatística U (Theil), concluiu-se que o método Holt-Winters obteve melhores resultados, captando melhor o comportamento linear das séries. Contudo, a diferença em relação ao modelo ARIMA não foi significativa. O estudo Ramos et al. (2015) incide na empresa retalhista de calçado Foreva e estabelece uma comparação entre os modelos de espaço de estados ETS e modelos ARIMA para cinco artigos da

empresa. Recorreu-se ao AIC para a escolha do melhor modelo ETS e ARIMA, sendo escolhido, para ambos os casos, o modelo com menor AIC. Realizaram-se previsões para diferentes passos. Em seguida, ambas as metodologias foram comparadas com base no RMSE, MAE e MAPE. Os resultados obtidos demonstraram que os modelos ETS e ARIMA apresentam um desempenho idêntico, quer para previsões com um passo à frente, quer para as previsões com múltiplos passos.

Os modelos de alisamento exponencial e ARIMA não são capazes de captar padrões não-lineares apresentados, frequentemente, nas séries de vendas do retalho. De forma a incorporar a não linearidade da procura, foram propostos vários modelos de previsão mais avançados (Au et al., 2008; Aye et al., 2015; Choi et al., 2014; Sun et al., 2008; Wanchoo, 2019; Xia et al., 2012).

Uma das abordagens apresentadas foram as *Artificial Neural Networks* (ANN) que são modelos que imitam o cérebro humano. A arquitetura mais comum da ANN é a *Multilayer Perceptron* (MLP), onde os neurónios são agrupados em camadas, existindo, apenas, conexões diretas (Rocha et al., 2007). A ideia do algoritmo é treinar a rede neuronal e, posteriormente, utilizar a mesma para prever valores futuros. As ANN permitem não só estabelecer a relação entrada-saída para sistemas não lineares, mas também a tolerância ao ruído. A tarefa mais complexa neste algoritmo é o desenho da rede neuronal, tendo sido propostos vários métodos baseados em procedimentos *hill-climbing*. No entanto, estes métodos têm dificuldade em passar o mínimo local.

A computação evolucionária é uma alternativa para estrutura ótima da rede neuronal, uma vez que faz uma pesquisa global de vários pontos, localizando, de forma rápida, áreas com maior qualidade, mesmo que o espaço seja grande e complexo. Au et al. (2008) propuseram um modelo híbrido, denominado por *Evolutionary Neural Networks* (ENN), para o setor do retalho de moda, que combina a computação evolucionária com as ANN. O modelo foi comparado com o modelo ARIMA sazonal, com base no erro quadrático médio, e verificou-se que as redes neuronais evolucionárias têm um melhor desempenho do que os modelos ARIMA sazonais, em casos onde a procura não apresenta grandes flutuações ao longo da semana e também quando a tendência da sazonalidade é reduzida. Além disso, as ENN são processos altamente automatizados, ao contrário dos modelos ARIMA que exigem mais conhecimento humano, pois é necessário selecionar os parâmetros do modelo.

Uma das desvantagens dos modelos de previsão baseados em ANN e ENN é o elevado tempo computacional (Ren et al., 2017). No sentido de solucionar este problema, vários estudos, realizados para o setor do retalho de moda, sugeriram modelos baseados em *Extreme Learning Machines* (ELM) que exigem menores tempos computacionais. (Choi et al., 2014; Sun et al., 2008; Xia et al., 2012).

Na realidade, é extremamente difícil escolher um único modelo que funcione em todos os casos, devido à complexidade das séries de vendas e ao número elevado de produtos diferentes, vendidos pelas empresas (Fildes and Petropoulos, 2015). Recentemente, foram apresentadas técnicas de combinação de modelos (Aye et al., 2015; Montero-Manso et al., 2020) e de seleção automática dos mesmos (Talagala et al., 2018; Villegas et al., 2018) que possibilitam processos de previsão mais fiáveis.

Aye et al. (2015) usaram 26 modelos de previsão para o setor do retalho com o objetivo de

prever o volume de vendas, considerando diferentes horizontes temporais. O conjunto de modelos utilizado contempla modelos lineares (ARIMA, Holt-Winters, entre outros), modelos não lineares, incluindo os modelos baseados em ANN, e modelos combinados. Os modelos combinados propostos aplicam métodos de combinação simples (média das previsões dos diferentes modelos) e de combinação ponderada, onde os pesos são atribuídos de acordo com o desempenho recente de cada modelo individual. A partir da comparação dos modelos, constatou-se que o desempenho dos modelos individuais variava bastante com o horizonte temporal, ao contrário dos modelos combinados que atingiram melhores resultados que os restantes modelos para todos os horizontes temporais estudados, nomeadamente o modelo de combinação ponderada.

Montero-Manso et al. (2020) sugeriram uma nova metodologia, baseada em *meta-learning*, que permite obter os pesos para as combinações ponderadas das previsões, com base nas características das séries. A metodologia é dividida em duas fases. A primeira fase consiste na utilização de um conjunto de séries temporais para treinar o algoritmo, minimizando a média da medida de erro utilizada, que consiste na combinação do erro médio absoluto de escala com o erro percentual absoluto médio simétrico. Na segunda fase, são obtidas as previsões para as novas séries temporais, com base nos pesos atribuídos pelo modelo treinado anteriormente. Os resultados obtidos revelaram que o algoritmo sugerido supera os modelos de previsão individuais e os modelos baseados em combinação simples.

Talagala et al. (2018) propuseram um sistema baseado em *meta-learning* que possibilita a seleção automática do melhor modelo de previsão para cada série temporal, de forma rápida. A metodologia apresentada utiliza o algoritmo de classificação *random forest* para determinar o melhor modelo de previsão, tendo em conta as características das séries temporais. Tal como em Montero-Manso et al. (2020), o mecanismo de previsão pode ser divido em duas partes: treino do classificador *random forest* e previsão das novas séries temporais, recorrendo ao modelo selecionado pelo algoritmo *random forest*.

Villegas et al. (2018) sugeriu um mecanismo de previsão da procura semelhante ao de Talagala et al. (2018). Porém, este procedimento utiliza o algoritmo de classificação SVM, em vez do *random forest*, para a seleção do melhor modelo.

As SVMs têm tido uma grande popularidade pela sua resistência a problemas de *overfitting* e pela capacidade de obter soluções ótimas globais. Estes algoritmos têm sido aplicados em problemas de classificação de diversas áreas: diagnósticos de doenças cancerígenas (Huang et al., 2008), reconhecimento facial (Sun et al., 2019; Shen et al., 2016) e avaliação de solvabilidade e de risco de empresas (Chen, 2011) e entidades bancárias (Feki et al., 2012; Gogas et al., 2018).

Talagala et al. (2018) e Villegas et al. (2018) compararam as suas metodologias com os modelos individuais, geralmente utilizados (ARIMA, ETS, entre outros), e, com base nas medidas de erro, constataram que as metodologias propostas apresentam uma exatidão superior à dos modelos individuais. Contudo, o conjunto de modelos candidatos, utilizado no algoritmo de seleção proposto por Villegas et al. (2018), não considera fatores que devem ser tidos em conta nas previsões para o setor do retalho: a sazonalidade das séries e as informações promocionais.

Capítulo 3

Caso de Estudo

3.1 Conjunto de Dados

O Grupo Jerónimo Martins é um grupo internacional fundado em 1792, que opera nos setores da distribuição alimentar e do retalho especializado. Atualmente, conta com um vasto número de lojas, empregando mais de 115000 colaboradores distribuídos por três países: Portugal, Polónia e Colômbia. A sua principal atividade é a distribuição alimentar, que representa 95% das vendas consolidadas do grupo e na qual é líder em Portugal, através da cadeia de lojas Pingo Doce e Recheio.

O trabalho desenvolvido baseia-se num conjunto de dados disponibilizado pelo Grupo Jerónimo Martins ao INESC TEC, no âmbito de um protocolo de colaboração entre as duas organizações. Os dados são referentes ao volume de vendas de 12 supermercados Pingo Doce durante o período compreendido entre 3 de Janeiro de 2012 e 27 de Abril de 2015, perfazendo um total de 173 semanas. É importante salientar, que foi realizada uma análise semanal das vendas, visto que é a periodicidade com mais interesse para retalhista.

Neste projeto foram apenas selecionados os SKUs que apresentam vendas em todas as semanas, uma vez que são os produtos com maior impacto nas receitas e que exigem operações logísticas mais eficientes. Estes SKUs são habitualmente denominados por *fast moving goods* (Huang et al., 2014; Trapero et al., 2013, 2014).

Os produtos considerados na presente dissertação, pertencem às seis categorias principais: mercearia, perecíveis especializados, perecíveis não especializados, bebidas, detergentes e produtos de limpeza.

Para o desenvolvimento da *framework* proposta, recorreu-se ao histórico de vendas das lojas de maior e menor dimensão, que são referenciadas ao longo da dissertação, respetivamente, por loja 412 e loja 843. No caso da loja 412, foram selecionados 988 *fast moving goods*, enquanto que para a loja 843 foram considerados 717 *fast moving goods*, sendo que estas lojas têm 558 SKUs comuns, o que representa, aproximadamente, 78% dos SKUs da loja 843.

A partir da análise da evolução das vendas dos SKUs comuns às duas lojas (ver Figura 3.1),

constatou-se as séries temporais correspondentes à loja 412 apresentam padrões (tendência e sazonalidade) idênticos aos que são observados nas séries da loja 843, apesar do volume de vendas semanais serem distintos.

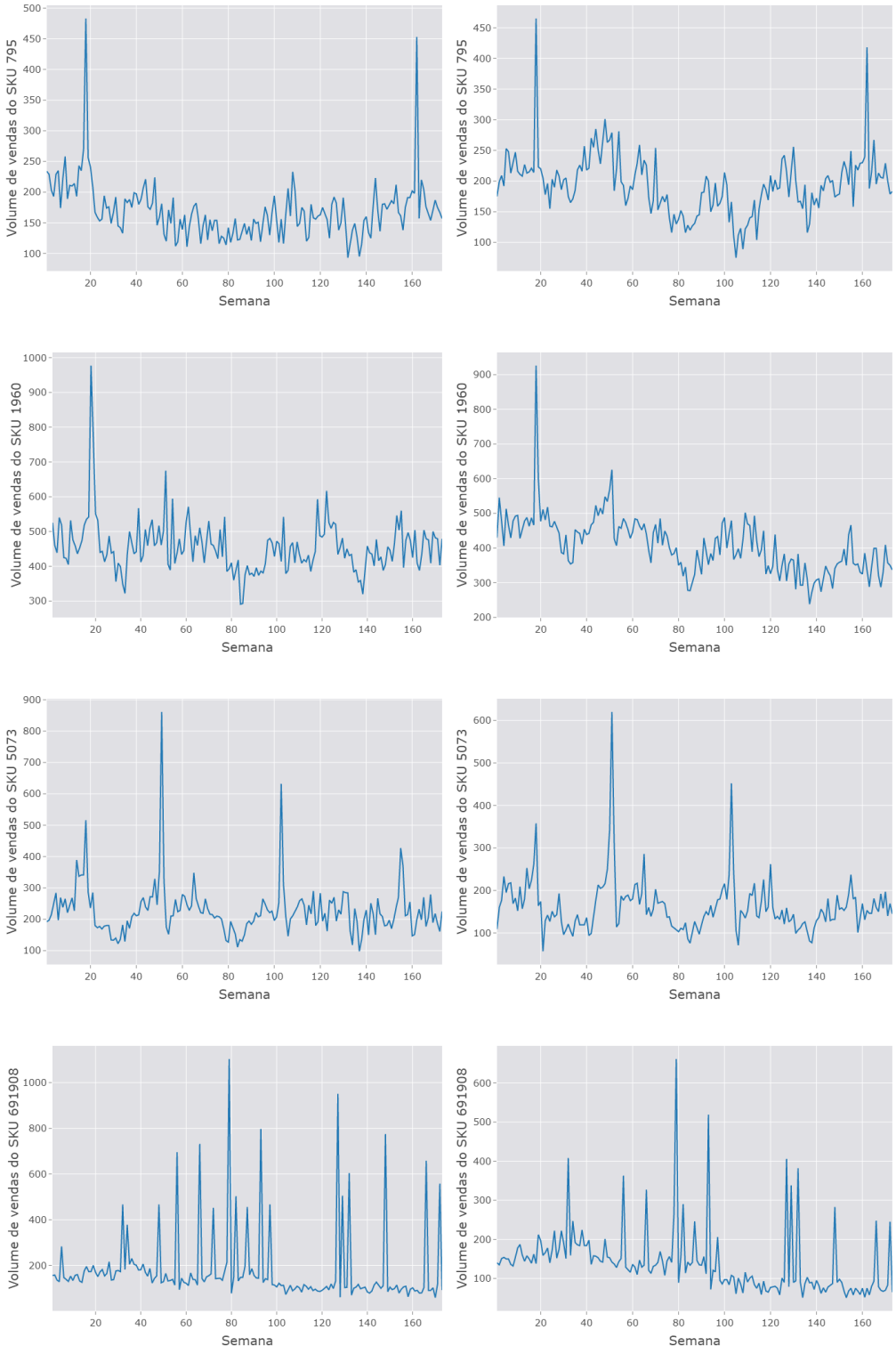

Figura 3.1: Evolução das vendas dos SKUs comuns às lojas 412 (à esquerda) e 843 (à direita).

3.2 Características das Séries Temporais

De acordo com Rice (1976), a exatidão associada aos métodos de previsão varia com a natureza dos dados. Desta forma, o estudo da natureza dos dados será importante para a determinação do melhor modelo para cada série temporal.

Na presente dissertação, é realizado um estudo do caráter dos dados baseado nas suas características, com o intuito de selecionar o modelo mais apropriado para cada série. Pode-se definir como característica, qualquer medida calculada, a partir dos valores de uma série temporal. A utilização de um conjunto de características, ao invés do uso direto das observações, permite atenuar a influência da escassez de dados e dos *outliers* (Wang et al., 2006).

As características usadas foram selecionadas com base nos estudos realizados por Talagala et al. (2018) e Montero-Manso et al. (2020), dado que têm um objetivo comum. É importante salientar que as características escolhidas são independentes de escala e assintoticamente independentes da duração das série temporais. Além disso, estas características não exigem um elevado tempo computacional, visto que este projeto procura obter uma metodologia que possibilite a previsão rápida de séries temporais.

Seguidamente, é feita uma descrição das diferentes características calculadas para o caso de estudo, as quais são apresentadas na Tabela B.1.

Características baseadas numa decomposição STL (*Seasonal and Trend decomposition using Loess*): Com base na decomposição da série temporal y_t, podem ser calculadas várias características como: a força da tendência, a força da sazonalidade, a linearidade, a curvatura, a *spikiness* e o primeiro coeficiente de autocorrelação da série residual. Numa primeira instância, a série y_t é submetida um processo de transformação Box-Cox (Guerrero, 1993), de forma a estabilizar a variância e tornar a sazonalidade aditiva. Posteriormente, a série transformada y_t^* é decomposta com recurso ao método de decomposição STL (Cleveland et al., 1990), resultando na equação $y_t^* = T_t + S_t + R_t$, onde T_t representa a tendência, S_t representa a componente sazonal e R_t representa a componente residual. A série desprovida de tendência é dada por $y_t^* - T_t = S_t + R_t$, ao passo que, a série desprovida de sazonalidade pode ser expressa por $y_t^* - S_t = T_t + R_t$.

A força da tendência é calculada a partir da comparação da variância da série desprovida de sazonalidade e da série residual (Wang et al., 2009):

$$\text{Tendência} = \max[0, 1 - var(R_t)/var(T_t + R_t)], \tag{3.1}$$

enquanto que a força da sazonalidade é definida por:

$$\text{Sazonalidade} = \max[0, 1 - var(R_t)/var(S_t + R_t)]. \tag{3.2}$$

As características linearidade e curvatura são baseadas nos coeficientes de uma regressão quadrática ortogonal que pode ser escrita da seguinte forma:

$$T_t = \beta_0 + \beta_1\phi_1(t) + \beta_2\phi_2(t) + \varepsilon_t \quad \text{para} \quad t = 1,2,...,T, \tag{3.3}$$

onde ϕ_1 e ϕ_2 são polinomiais ortogonais de 1ª e 2ª ordem. Os valores estimados para parâmetros β_1 e β_2 são, respetivamente, as medidas de linearidade e de curvatura. É importante realçar, que estas duas características são dependentes da escala da série temporal e por essa razão, as séries temporais são escaladas antes do cálculo destas características, de modo a que a média seja 0 e a variância 1.

Hyndman et al. (2015) sugeriram um índice para medir a *spikeness*, sendo útil nos casos em que as séries apresentam *outliers* ocasionais. O índice proposto é igual à variância das variâncias *leave-one-out* da série residual r_t.

Estabilidade e granulosidade: A estabilidade e a granulosidade de uma série temporal são obtidas com base na divisão da série em intervalos não coincidentes. A estabilidade consiste na variância das médias dos intervalos, ao passo que, a granulosidade diz respeito à variância das variâncias dos intervalos (Hyndman et al., 2015).

Entropia espectral de uma série temporal: A entropia espectral é uma característica que mede a previsibilidade de uma série temporal. As séries de fácil previsão estão associadas a uma entropia espectral reduzida, em oposição, as séries bastante ruidosas apresentam valores elevados desta característica. Para a sua determinação, foi utilizada uma medida que estima a entropia de *Shannon* da densidade espetral normalizada (Goerg, 2013):

$$H_s(y_t) = -\int_{-\pi}^{\pi} \hat{f}_y(\lambda)\log\hat{f}_y(\lambda)\,d\lambda, \tag{3.4}$$

onde $\hat{f}_y$ representa a estimativa da densidade espectral introduzida por Nuttall and Carter (1982).

Expoente de Hurst: O expoente de Hurst permite avaliar a memória de longo prazo de uma série temporal. Esta medida é dada por $H = d + 0.5$, onde d é o grau de diferenciação fracional estimado para um modelo $ARFIMA(0,d,0)$. O valor de H foi calculado recorrendo ao método de máxima verosimilhança.

Não Linearidade: O grau de não linearidade das séries temporais foi obtido a partir do valor da estatística calculada no teste de rede neuronal para não linearidade de Teräsvirta ((Teräsvirta et al., 1993)). Quando esta característica toma valores elevados significa que a série é não linear. No caso de assumir valores próximos de 0, quer dizer que a série é linear.

Testes de raiz unitária: De forma avaliar a estacionariedade das séries temporais, recorreu-se aos testes de Phillips-Perron e KPSS. O teste de Phillips-Perron é baseia-se na regressão:

$$y_t = c + \alpha y_{t-1} + \varepsilon_t \tag{3.5}$$

e trata-se do valor da estatística de teste Z-alpha com o parâmetro janela de Bartlett igual à parte inteira da expressão $4(T/100)^{0.25}$.

Relativamente ao teste KPSS, este é baseado na regressão:

$$y_t = c + \delta_t + \alpha y_{t-1} + \varepsilon_t \tag{3.6}$$

e diz respeito ao valor da estatística de teste KPSS com o parâmetro janela de Bartlett igual ao que foi utilizado no teste de Phillips-Perron.

Características baseadas em coeficientes de autocorrelação: Foram calculadas várias características baseadas nos valores das funções de autocorrelação e de autocorrelação parcial das séries originais e diferenciadas, as quais estão explicitadas na Tabela B.1.

3.3 Metodologia

O presente projeto foi desenvolvido no sentido de elaborar um mecanismo de previsão que possibilite não só selecionar o melhor modelo a ajustar a cada SKU, mas também obter as previsões de vendas de forma expedita (ver Figura 3.2).

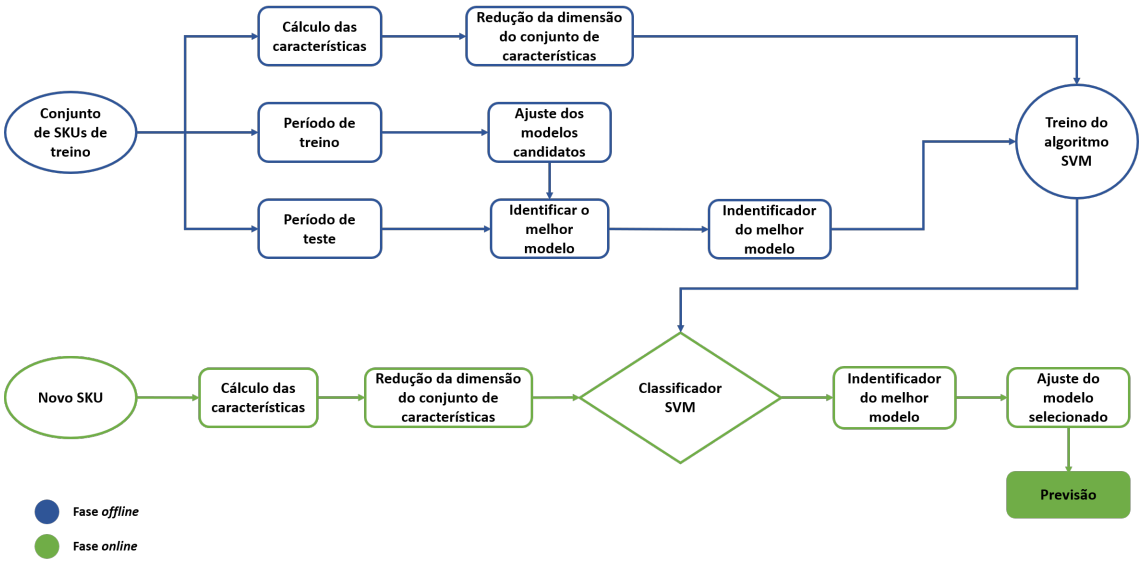

Figura 3.2: Representação esquemática da metodologia proposta.

A metodologia proposta é baseada no algoritmo de classificação SVM e é composta por duas fases: a fase *offline* e a fase *online*. A fase *offline* consiste no treino do algoritmo SVM que, posteriormente, é aplicado na fase *online* para determinar o melhor modelo para uma nova série temporal. O Algoritmo 1 descreve detalhadamente o funcionamento da metodologia apresentada.

Durante a fase *offline*, para cada série, são calculadas as respetivas características e, de seguida, reduz-se a dimensão do conjunto obtido, recorrendo ao método PCA. Uma vez que a quantidade de observações necessárias para gerar previsões fiáveis cresce exponencialmente com o aumento do

número de características (Hira and Gillies, 2015), é fundamental reduzir a dimensão do conjunto de características. Desta forma, aplicou-se o método PCA para reduzir o número de características, dado que o número de observações não é suficientemente elevado para o conjunto de características utilizado, permitindo que algoritmo SVM tenha uma maior capacidade de generalização e, por conseguinte, melhores resultados na fase *online*.

Seguidamente, cada série temporal é dividida em período de treino e de teste, utilizando um procedimento de validação cruzada. Neste procedimento, são definidos vários conjuntos de teste formados por uma única observação, sendo que o respetivo conjunto de treino é constituído pelas observações anteriores à observação de teste. Para cada período de treino, são ajustados todos modelos candidatos e calcula-se a previsão a 1 passo para o instante correspondente à observação de teste. Posteriormente, determina-se o erro de previsão para esse instante. A partir dos erros de previsão calculados, são obtidas as medidas de desempenho e, com base nas mesmas, seleciona-se o melhor modelo para cada série.

Após a determinação do melhor modelo para cada série temporal, envia-se para o algoritmo SVM o conjunto de dados, denominado por *meta-data*, que é composto pelas componentes principais de cada série e pelo indentificador do modelo atribuído à mesma. Por fim, são calculados os parâmetros ótimos do classificador SVM.

Na fase *online*, será necessário determinar o conjunto características para uma nova série temporal, reduzir a dimensão do mesmo, recorrendo ao modelo PCA obtido na fase *offline*, e enviar as componentes principais obtidas para o algoritmo SVM. De seguida, o classificador SVM indica o modelo que deve ser ajustado à série inserida e é calculada a previsão.

A fase *offline* é bastante exigente, em termos computacionais e de consumo de tempo, visto que nesta fase é necessário ajustar todos modelos candidatos e calcular os seus respetivos erros para treinar o algoritmo SVM. Quanto maior for número de modelos candidatos, maior será o esforço computacional e o tempo de processamento associado à fase *offline*. No entanto, justifica-se a utilização de um vasto conjunto de modelos, desde que a seleção seja feita de forma adequada, pois poderá levar a aumentos significativos em termos de exatidão da previsão. O elevado tempo computacional, requerido na fase *offline*, não afetará a velocidade de previsão da metodologia apresentada, dado que esta fase não é executada continuamente.

A fase *online* é rápida, já que o cálculo do conjunto de características e a redução da sua dimensão envolvem um tempo computacional bastante baixo.

Como foi referido na secção 2.3.2, o desempenho do algoritmo SVM depende da função de Kernel utilizada no mapeamento dos dados para o espaço de características. Assim, no presente caso de estudo, foram testadas as funções de Kernel mais comuns.

Admitindo que as séries temporais com características próximas têm um comportamento semelhante, é fundamental que as características das séries correspondentes aos SKUs usados na fase *offline* mapeiem o espaço de características das séries para as quais são realizadas as previsões na fase *online*. Por conseguinte, é extremamente importante que o conjunto de SKUs, usados na fase *offline*, seja suficientemente grande para captar o maior número de combinações possíveis de características, de forma a obter uma elevada exatidão das previsões na fase *online*.

Algoritmo 1: Metodologia

Fase *offline*: Treino do classificador SVM

Dado:

$O = \{x_1, x_2, x_3, ..., x_n\}$: Conjunto de n séries temporais.

L : Conjunto de m modelos candidatos (ARIMA, ETS, TBATS, LASSO, ...).

F : Função de cálculo das características.

T : Número de observações de cada série.

k : Número de observações do conjunto treino inicial.

Saída:

Classificador de atribuição de modelos.

Para $i = 1$ até n:

Aplicar F para calcular as características para série x_i.

Redução da dimensão do conjunto de características, recorrendo ao método PCA.

Para $j = 1$ até m:

Para $t = 1$ até $T - k$:

Selecionar a observação do instante $k + t$ para período de teste.

Ajustar o modelo l_j ao período de treino constituído pelas observações dos instantes $1, 2, ..., k + t - 1$.

Calcular o erro previsão para o instante $k + t$.

Determinar as medidas de desempenho para o modelo l_j, com base nos erros obtidos.

Atribuir a x_i o modelo com a melhor performance.

Meta-data: Componentes principais e o indentificador do modelo atribuído a cada série

Treino do algoritmo SVM com base no *Meta-data*.

Determinar os parâmetros ótimos do classificador SVM.

Obtenção do classificador SVM.

Fase *online*: Previsão

Dado:

x: Nova série temporal.

Saída:

Identificador do modelo apropriado para série temporal e a sua previsão.

Calcular as características de x.

Redução da dimensão do conjunto de características, aplicando o modelo PCA obtido na fase *offline*.

Ajustar o modelo atribuído pelo classificador SVM.

Realizar a previsão.

3.4 Procedimento Experimental

Para a avaliação da metodologia sugerida, considerou-se que a fase de treino da *framework* é realizada com os SKUs da loja 412 (988 SKUs), enquanto que para a fase de teste são utilizados os SKUs da loja 843 (717 SKUs). Esta escolha está relacionada com o facto de a loja 412 ser a de maior dimensão, apresentando um maior número de SKUs diferentes. Por conseguinte, o conjunto de comportamentos das séries abrangidas será mais diversificado, permitindo que o espaço de características dos SKUs da loja 412 compreenda as características relativas aos SKUs da loja 843. A Figura 3.3 representa a distribuição dos SKUs da loja 412 e da loja 843 no espaço de características para as duas primeiras componentes principais, sendo possível verificar que o espaço de características dos SKUs da loja 412 cobre as características dos SKUs da loja 843, tal como foi referido anteriormente.

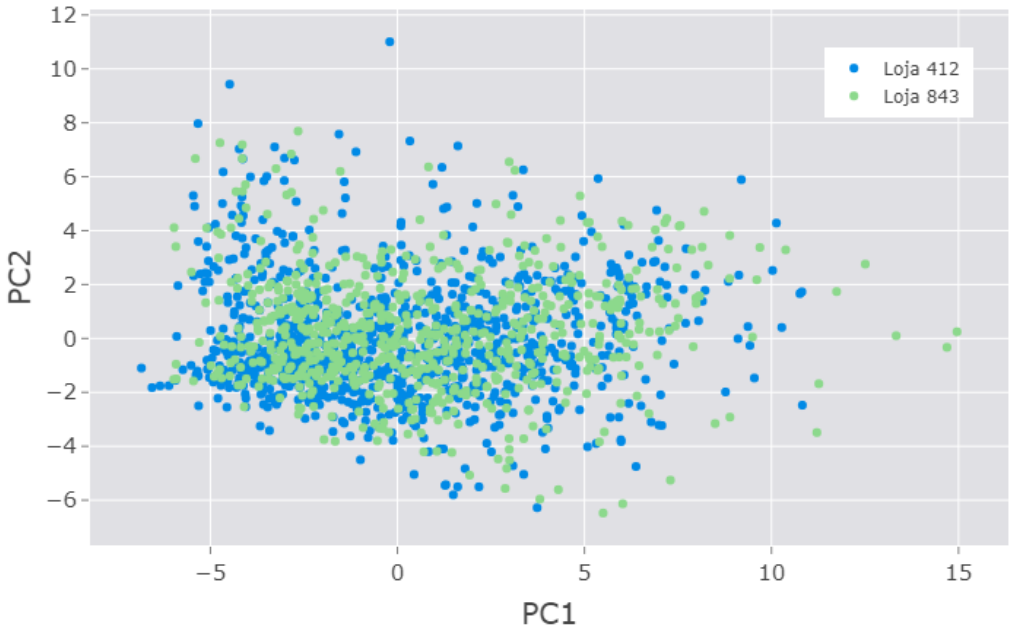

Figura 3.3: Distribuição dos SKUs da loja 412 e da loja 843 no espaço das características.

Na fase *offline*, calculou-se o conjunto de características da Tabela B.1 (um total de 36) para a série de vendas correspondente a cada SKU de treino e de seguida, reduziu-se a dimensão do conjunto obtido utilizando o método PCA. Antes de ter sido aplicado o método PCA, as características foram padronizadas para média 0 e variância 1, devido às diferenças de escala entre elas. A padronização tem como objetivo evitar que as características apresentem um maior peso nas componentes principais à custa da sua maior magnitude. Além disso, as características das séries de vendas relativas aos SKUs da loja 412 diferem, em termos de ordem de grandeza, das características referentes às séries da loja 843, o que implica a padronização das mesmas (ver Tabelas C.1 e C.2).

Após serem determinadas as componentes principais para o conjunto de características, resultantes da aplicação do método PCA, procedeu-se à seleção do número ótimo de componentes,

recorrendo ao método de validação cruzada 10-*fold*.

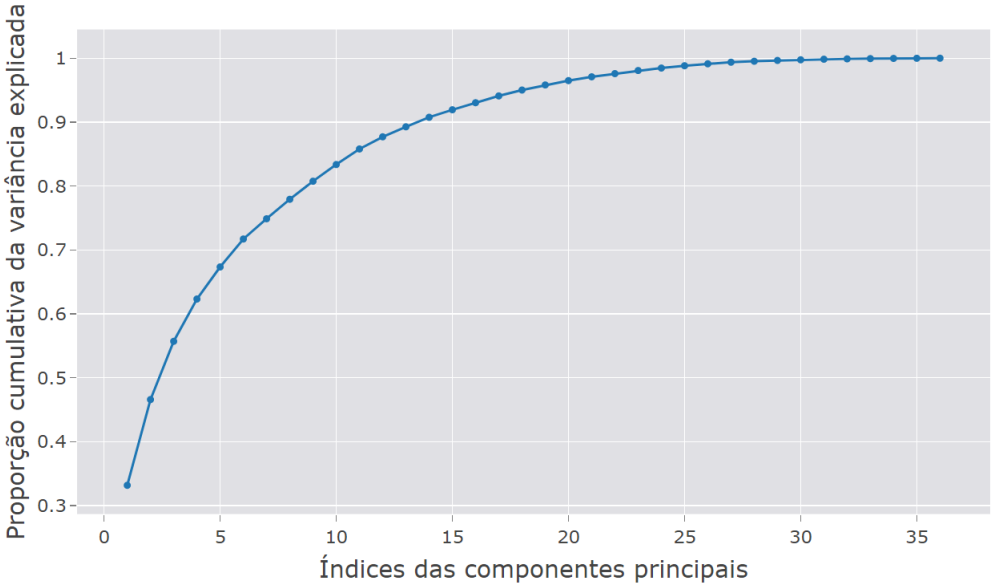

Figura 3.4: Proporção cumulativa da variância explicada pelas componentes principais utilizadas.

Ainda na fase *offline*, cada uma das séries de vendas foi dividida em período de treino e de teste, aplicando o processo de validação cruzada descrito na secção 3.3. Neste processo, foi definido um período de treino inicial que inclui as primeiras 156 observações de cada série de vendas. O conjunto de treino inicial é composto por um número elevado de observações (aproximadamente 90% das observações de cada série), visto que este conjunto deve caracterizar a série de vendas. Desta forma, o processo é composto por 17 passos, sendo que, em cada um deles, é incrementada uma observação de treino até ser atingido um total de 172 observações de treino. Para cada período de treino, foram ajustados todos modelos candidatos (num total de 18), calculando-se a previsão para o instante correspondente à observação de teste e o respetivo erro. Recorrendo aos erros de previsão obtidos, determinou-se o MAE e o MASE associados à utilização de cada modelo, e com base nestas medidas de desempenho, atribuiu-se o melhor modelo para cada série de vendas. É importante mencionar que, para o cálculo do MASE, os erros foram escalados pelo MAE resultante da aplicação do método Naïve sazonal com um período de sazonalidade igual a 52 semanas.

Posteriormente, treinou-se o classificador SVM com as componentes principais obtidas para cada série e com o identificador do modelo selecionado para a mesma. Visto que as funções de Kernel utilizadas na SVM influenciam a performance do classificador, no presente caso estudo, foram testadas as quatro funções mais comuns mencionadas na secção 2.3.2: linear, polinomial, sigmoide e radial. Para cada função de Kernel, determinaram-se os parâmetros C, γ, d e $coef0$ ótimos utilizando o método de validação cruzada 10-*fold*.

Tal como já foi mencionado, tanto o número ótimo de componentes principais, como os parâmetros do classificador SVM foram obtidos a partir da aplicação do procedimento de validação cruzada 10-*fold*. Para este processo, foram considerados diferentes valores para os parâmetros do SVM e para o número de componentes principais, tendo sido definidas as seguintes gamas de

valores: $NCPs = [7;18]$, $C = 10^{-3:6}$, $\gamma = 10^{-7:1}$, $d = \{2;3;4;5\}$ e $coef0 = \{0,01;0,1;1;2;3;4\}$. O intervalo de valores testado para o número de componentes principais foi estipulado de forma a que os valores considerados apresentem uma percentagem de variância explicada significativa. Nesse sentido, testou-se um número de componentes que permita explicar, aproximadamente, entre 75% e 95% da variância (ver Figura 3.4). Visto que não existe nenhuma informação prévia que nos permita estabelecer um critério objetivo, para a determinação dos parâmetros do classificador, fixou-se para cada parâmetro um conjunto de valores que compreendesse um intervalo com uma amplitude elevada, de maneira a que os valores ótimos se encontrem entre os limites dos intervalos definidos.

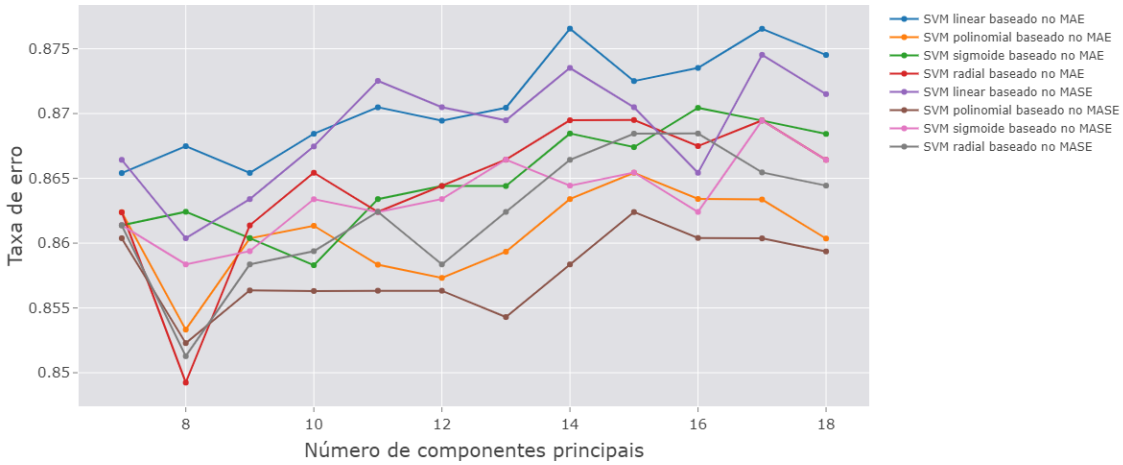

Figura 3.5: Evolução da taxa de erro das SVMs em função do número de componentes principais, considerando os parâmetros ótimos para cada Kernel.

O processo de validação cruzada aplicado consistiu em dividir, aleatoriamente, a amostra composta pelos valores das componentes principais para todos SKUs (um total de 988) e os pelos respetivos identificadores dos modelos atribuídos em 10 sub-amostras de igual dimensão. De seguida, definiu-se uma sub-amostra como período de teste e treinou-se o classificador SVM com as restantes 9 sub-amostras. Recorrendo ao modelo de classificação produzido, determinaram-se os identificadores dos melhores modelos para os SKUs pertencentes ao período de teste e com base nos identificadores obtidos, calculou-se a taxa de erro do algoritmo SVM. O processo descrito foi repetido 10 vezes para cada combinação de parâmetros, sendo que cada sub-amostra foi fixada, apenas uma vez, como período de teste. No final do processo, obtiveram-se os parâmetros ótimos apresentados na Tabela 3.1, com base na taxa de erro média.

A partir da observação da Figura 3.5, é possível constatar que a taxa de erro da SVM e o número de componentes principais escolhido não estabelecem uma relação linear. A Figura 3.5 também revela que o número de componentes principais ótimo varia com a função de Kernel utilizada e com o critério usado para a seleção de modelos. Deste modo, estes resultados indicam que, para este caso, o procedimento de validação adotado é mais adequado que a fixação de um valor para o número de componentes principais, com base, exclusivamente, na proporção de variância explicada.

Tabela 3.1: Parâmetros ótimos de cada tipo de classificador SVM para o caso do critério de seleção de modelos ser o MAE ou MASE.

MAE	*NCPs*	*C*	γ	*d*	*coef*0	**MASE**	*NCPs*	*C*	γ	*d*	*coef*0
SVM com Kernel linear	7	0.01	-	-	-	SVM com Kernel linear	8	10	-	-	-
SVM com Kernel polinomial	8	0.01	0.1	3	4	SVM com Kernel polinomial	8	1000	0.0001	4	2
SVM com Kernel radial	8	1000000	0.00001	-	-	SVM com Kernel radial	8	100	0.001	-	-
SVM com Kernel sigmoide	10	0.1	0.1	-	1	SVM com Kernel sigmoide	8	100000	0.0001	-	0.01

O conjunto de modelos candidatos foi constituído com base nos modelos propostos por Oliveira et al. (2020), os quais foram implementados no *software* estatístico R, recorrendo às seguintes funcionalidades:

1. ARIMA - função *auto.arima()* do package *forecast*;
2. ARIMA Fourier - função *auto.arima()* do package *forecast*;
3. ES - função *es()* do package *smooth*;
4. TBATS - função *tbats()* do package *forecast*;
5. LASSO-PACF - função *glmnet()* do package *glmnet*;
6. LASSO-PACF Fourier - função *glmnet()* do package *glmnet*;
7. Naïve sazonal - função *snaive()* do package *forecast*;
8. ARIMA + LASSOX - funções *auto.arima()* do package *forecast* e *glmnet()* do package *glmnet*;
9. ARIMA Fourier + LASSOX - funções *auto.arima()* do package *forecast* e *glmnet()* do package *glmnet*;
10. ES + LASSOX - funções *es()* do package *smooth* e *glmnet()* do package *glmnet*;
11. TBATS + LASSOX - funções *tbats()* do package *forecast* e *glmnet()* do package *glmnet*;
12. LASSO-PACF + LASSOX - função *glmnet()* do package *glmnet*;
13. LASSO-PACF Fourier + LASSOX - função *glmnet()* do package *glmnet*;
14. ARIMAX - funções *auto.arima()* do package *forecast* e *prcomp()* do package *stats*;
15. ARIMAX Fourier - funções *auto.arima()* do package *forecast* e *prcomp()* do package *stats*;
16. ESX - funções *es()* do package *smooth* e *prcomp()* do package *stats*;
17. LASSOX-PACF - função *glmnet()* do package *glmnet*;
18. LASSOX-PACF Fourier - função *glmnet()* do package *glmnet*;

Este conjunto de modelos inclui modelos univariados (1-7) e multivariados (8-18). Os modelos univariados apenas consideram o histórico de vendas, enquanto que os modelos multivariados incluem, para além dos dados referentes às vendas, regressores externos que integram informações relativamente ao SKU: preço, número de dias em promoção na semana, semana de final de mês, feriados e eventos de calendário (Natal, Páscoa, Carnaval, entre outros), considerando em certos casos *leads* e *lags*. Todos os modelos candidatos apresentados constituem o estado da arte no que diz respeito à previsão de vendas nos sectores da distribuição alimentar e do retalho especializado. Mais detalhes acerca da *pool* de modelos candidatos aplicada podem ser encontrados em Oliveira et al. (2020).

É importante salientar que, apesar dos modelos multivariados serem mais sofisticados devido à inclusão de regressores externos, a utilização de modelos univariados continua a ser adequada, dado que determinados SKUs não apresentam informação promocional. Analisando o gráfico da Figura 3.6, verifica-se uma variabilidade significativa no processo de determinação do melhor modelo para cada SKU de treino, o que demonstra que a *pool* de modelos, aplicada ao caso de estudo, é apropriada. Além disso, a Figura 3.6 mostra que a *pool* de modelos candidatos é equilibrada, permitindo que o conjunto de dados usado no treino do classificador SVM seja balanceado.

Relativamente à implementação do algoritmo, o projeto foi desenvolvido no software estatístico R com recurso ao IDE RStudio. O cálculo do conjunto de características foi realizado aplicando a função *features()* do package *feasts*, ao passo que o classificador SVM foi produzido a partir da utilização da função *svm()* do package *e1071*. Também importante referir que uma parte do processamento da fase *offline* foi efetuado com o auxílio do serviço GridFEUP disponibilizado pela Faculdade de Engenharia da Universidade do Porto.

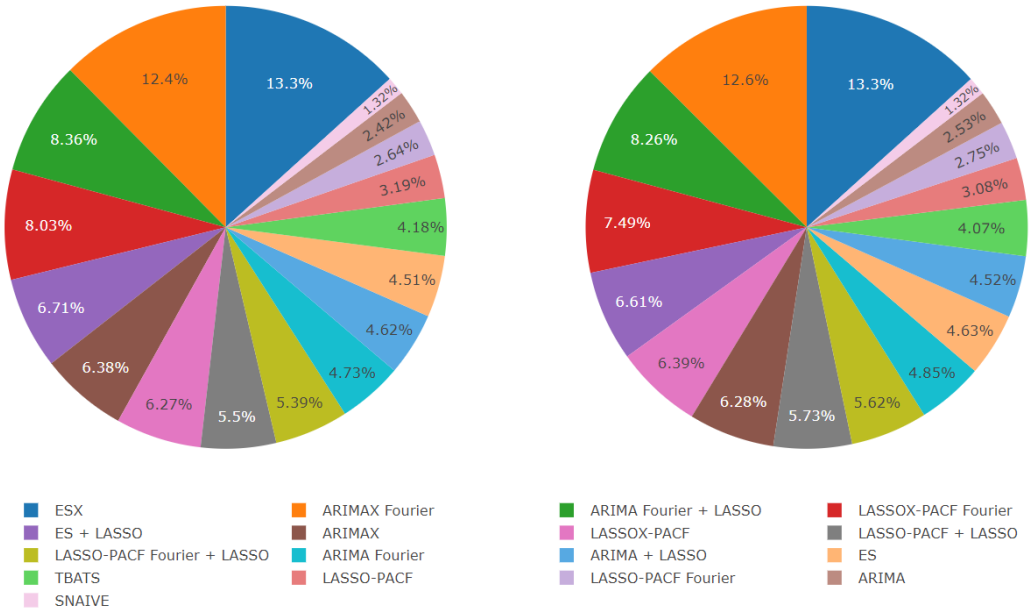

Figura 3.6: Frequência (%) com que os modelos candidatos são selecionados para os SKUs de treino, no caso de o critério de seleção ser o MAE (à esquerda) ou MASE (à direita).

3.5 Resultados

O desempenho da metodologia apresentada foi avaliado a partir da simulação da fase *online* para os SKUs de teste (ver Figura 3.7).

No procedimento de validação da *framework*, cada uma das séries de vendas foi dividida em período de treino e de teste, utilizando um processo de validação cruzada. Tal como na fase *offline*, o processo de estratificação definiu 17 períodos de treino diferentes, iniciando com um período de treino, composto pelas primeiras 156 observações de cada série de vendas. Para cada período de treino, foram calculadas as respetivas características. De seguida, estas foram padronizadas (média 0 e variância 1) e convertidas para as coordenadas das componentes principais determinadas na fase *offline*. Posteriormente, os valores das componentes principais foram encaminhados para o classificador SVM que indicou o identificador do modelo atribuído a cada série de vendas. Seguidamente, ajustou-se o modelo atribuído pela SVM, tendo sido calculada a previsão a 1 passo para o instante correspondente à observação de teste e o respetivo erro de previsão.

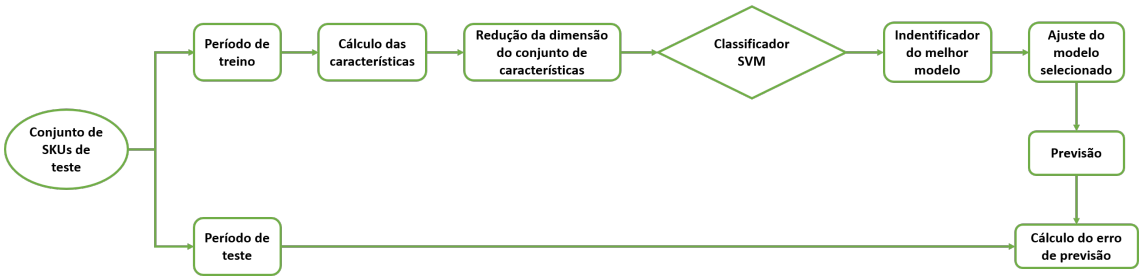

Figura 3.7: Representação esquemática do procedimento de validação da metodologia sugerida.

O procedimento descrito, anteriormente, foi realizado para o caso de, na fase *offline*, a seleção do melhor modelo para cada série de vendas ser baseada nas medidas de erro MAE ou MASE. Para cada uma das medidas de erro, foi avaliado o desempenho da *framework*, considerando as seguintes funções de Kernel: linear, polinomial, radial e sigmoide.

Finalmente, com base nos 17 erros de previsão obtidos para cada série de vendas, determinaram-se as medidas de erro MAE e MASE e, posteriormente, a GMRelMAE e o valor médio do MASE. Os erros usados no cálculo da GMRelMAE e do MASE foram escalados pelo MAE resultante da utilização do método Naïve sazonal, com um período de sazonalidade igual a 52 semanas.

No sentido de avaliar o desempenho da metodologia, para os dois cenários apresentados na Tabela 3.2, comparou-se a exatidão das previsões resultantes da utilização dos quatro SVMs com a exatidão de previsão associada à aplicação generalizada dos 18 modelos candidatos. Desta forma, obtiveram-se as medidas de desempenho, relativas a cada cenário, para todos os modelos candidatos, recorrendo ao procedimento de validação levado a cabo para a *framework* desenvolvida.

De uma forma sucinta, foram definidos 17 períodos de treino diferentes, sendo o período inicial composto pelas 156 observações de cada série de vendas. Para cada período de treino, ajustaram-se todos os modelos candidatos. Seguidamente, foram calculadas as previsões para o instante de

teste correspondente e o respetivo erro de previsão. Com base nos 17 erros obtidos para cada série de vendas, determinaram-se as medidas de desempenho GMRelMAE e MASE para cada modelo.

É importante sublinhar que, para a validação da metodologia, foram utilizadas somente medidas independentes de escala, uma vez que as séries de vendas analisadas apresentam escalas diferentes.

Tabela 3.2: Cenários considerados na fase de validação da metodologia proposta.

	Cenário 1	Cenário 2
Critério de seleção de modelos	MAE	MASE
Medida de avaliação dos modelos	GMRelMAE	MASE

Na Tabela 3.3, são apresentados os resultados obtidos para os dois cenários. O *rank* indica a classificação média dos modelos, tendo em conta o seu desempenho para os dois cenários analisados.

Os resultados evidenciam que, para os dois cenários estudados, a metodologia proposta apresenta uma performance superior à dos modelos individuais. A Tabela 3.3 revela ainda que a *framework* obteve, para todos os tipos de SVM, o melhor desempenho, demonstrando que o algoritmo desenvolvido é, manifestamente, uma melhor alternativa do que qualquer um dos modelos presentes na *pool* utilizada.

A partir da análise dos resultados, também é possível verificar que de uma forma geral, a função de Kernel mais adequada para este caso estudo é a radial, ao passo que a pior é o Kernel linear. Os resultados obtidos estão de acordo com o esperado, uma vez que o Kernel radial é uma função mais sofisticada que a função linear, permitindo agrupar os dados em pequenas "ilhas"devido à inclusão do parâmetro γ.

É importante destacar que a *framework* tem a melhor performance, independentemente do critério de seleção de modelos e da medida de avaliação usada, o que revela a robustez do algoritmo concebido.

De facto, na fase de treino, a taxa de erro associada ao classificador SVM é elevada (ver Figuras D.1,D.2,D.3,D.4). Esta fragilidade está relacionada com o elevado número de modelos candidatos e com o facto dos modelos apresentarem níveis de exatidão muito próximos. Contudo, a performance da metodologia desenvolvida não é afetada, significativamente, visto que o modelo selecionado para cada série de vendas, mesmo nos casos em que não é escolhido o melhor, revela um desempenho semelhante ao do melhor modelo.

Com o objetivo de evidenciar a importância da aplicação do método de análise de componentes principais para a eficácia da metodologia proposta, confrontou-se a exatidão da metodologia onde é aplicada a PCA com a exatidão da metodologia em que não é usada a PCA. De facto, a *framework* apresenta uma maior exatidão no caso em que é utilizado a PCA (ver Tabelas 3.3 e 3.4). Desta forma, fica patente o caráter decisivo da seleção de características para o bom desempenho do algoritmo proposto.

Tabela 3.3: Resultados referentes à aplicação da metodologia proposta e de todos os modelos candidatos para o cenário 1 (à esquerda) e 2 (à direita), no caso de ser aplicado o método PCA.

Modelo	GMRelMAE	Modelo	MASE	**Rank**
SVM com Kernel radial	0,5614	SVM com Kernel radial	0,6474	1,5
SVM com Kernel polinomial	0,5620	SVM com Kernel polinomial	0,6473	2
SVM com Kernel sigmoide	0,5619	SVM com Kernel sigmoide	0,6483	2,75
SVM com Kernel linear	0,5625	SVM com Kernel linear	0,6483	3,75
ESX	0,5667	ESX	0,6535	5
ARIMAX Fourier	0,5692	ARIMAX Fourier	0,6541	6
ES + LASSOX	0,5696	ES + LASSOX	0,659	7
ARIMA Fourier + LASSOX	0,5705	ARIMA Fourier + LASSOX	0,6577	8
TBATS + LASSOX	0,5744	TBATS + LASSOX	0,6607	9
LASSOX-PACF	0,5804	LASSOX-PACF	0,6683	10
ARIMAX	0,5811	ARIMAX	0,6716	11
ARIMA + LASSOX	0,5811	ARIMA + LASSOX	0,6731	12
LASSOX-PACF Fourier	0,5977	LASSOX-PACF Fourier	0,6897	13
LASSO-PACF + LASSOX	0,6010	LASSO-PACF + LASSOX	0,6933	14
LASSO-PACF Fourier + LASSOX	0,6042	LASSO-PACF Fourier + LASSOX	0,6958	15
ARIMA Fourier	0,6493	ARIMA Fourier	0,7724	16
ES	0,6499	ES	0,7774	17
TBATS	0,6598	TBATS	0,7827	18
ARIMA	0,6585	ARIMA	0,7852	19
LASSO-PACF	0,6777	LASSO-PACF	0,7997	20
LASSO-PACF Fourier	0,6867	LASSO-PACF Fourier	0,8077	21
Naïve sazonal	1,0000	Naïve sazonal	1,1352	22

Tabela 3.4: Resultados referentes à aplicação da metodologia proposta e de todos os modelos candidatos para o cenário 1 (à esquerda) e 2 (à direita), no caso de não ser aplicado o método PCA.

Modelo	GMRelMAE	Modelo	MASE	**Rank**
SVM com Kernel polinomial	0,5639	SVM com Kernel polinomial	0,6481	1,5
SVM com Kernel sigmoide	0,5630	SVM com Kernel sigmoide	0,6496	2,25
SVM com Kernel radial	0,5645	SVM com Kernel radial	0,6494	3
SVM com Kernel linear	0,5643	SVM com Kernel linear	0,6496	3,25
ESX	0,5667	ESX	0,6535	5
ARIMAX Fourier	0,5692	ARIMAX Fourier	0,6541	6
ES + LASSOX	0,5696	ES + LASSOX	0,659	7
ARIMA Fourier + LASSOX	0,5705	ARIMA Fourier + LASSOX	0,6577	8
TBATS + LASSOX	0,5744	TBATS + LASSOX	0,6607	9
LASSOX-PACF	0,5804	LASSOX-PACF	0,6683	10
ARIMAX	0,5811	ARIMAX	0,6716	11
ARIMA + LASSOX	0,5811	ARIMA + LASSOX	0,6731	12
LASSOX-PACF Fourier	0,5977	LASSOX-PACF Fourier	0,6897	13
LASSO-PACF + LASSOX	0,6010	LASSO-PACF + LASSOX	0,6933	14
LASSO-PACF Fourier + LASSOX	0,6042	LASSO-PACF Fourier + LASSOX	0,6958	15
ARIMA Fourier	0,6493	ARIMA Fourier	0,7724	16
ES	0,6499	ES	0,7774	17
TBATS	0,6598	TBATS	0,7827	18
ARIMA	0,6585	ARIMA	0,7852	19
LASSO-PACF	0,6777	LASSO-PACF	0,7997	20
LASSO-PACF Fourier	0,6867	LASSO-PACF Fourier	0,8077	21
Naïve sazonal	1,0000	Naïve sazonal	1,1352	22

Capítulo 4

Conclusões e Trabalho Futuro

4.1 Conclusões

Efetivamente, o setor do retalho é caracterizado pela sua elevada competitividade, exigindo às empresas um rigoroso planeamento estratégico e operacional. Por conseguinte, é fundamental a utilização de uma metodologia de previsão eficaz que possibilite, por um lado, um controlo minucioso dos custos operacionais e, por outro lado, um serviço de qualidade ao cliente.

Devido ao número elevado de SKUs, as empresas necessitam de um mecanismo que assegure a seleção automática do melhor modelo para cada SKU. Assim, o presente trabalho surge como uma potencial resposta a esta necessidade.

O estudo dos modelos mais comuns de previsão e do algoritmo supervisionado SVM foi determinante para a realização desta dissertação e sustentou a elaboração do algoritmo proposto.

A *framework* sugerida seleciona, de uma forma rápida, o modelo de previsão mais apropriado para cada SKU, com base na performance que cada modelo apresenta, quando aplicado às séries de vendas.

De modo a avaliar o desempenho da *framework*, comparou-se a sua exatidão de previsão com a dos modelos de previsão disponíveis na literatura. Nesse sentido, utilizaram-se medidas de erro adequadas para a realização desta comparação, que permitem mitigar as diferenças de escala evidenciadas pelo conjunto de séries considerado.

Todas as previsões geradas possuem uma periodicidade semanal, não só por estarem de acordo com as necessidades do setor do retalho, mas também por se ajustarem às necessidades do Grupo Jerónimo Martins.

O classificador SVM foi treinado com informação relativa a um conjunto diverso de séries temporais, conferindo-lhe uma boa capacidade de generalização. Devido à possibilidade de generalização do algoritmo concebido, recorreu-se, para a sua implementação, aos dados referentes às vendas dos SKUs das lojas 412 e 843.

Ao contrário das técnicas não supervisionadas, como o *Clustering*, o classificador SVM permitiu que o treino do algoritmo fosse realizado com dados relativos não só às características, mas

também aos identificadores dos modelos atribuídos. Assim, foi considerado um conjunto de dados mais completo, proporcionando um algoritmo de previsão mais eficaz.

A metodologia proposta revela um desempenho superior ao dos modelos de previsão desenvolvidos pela literatura, assegurando uma maior exatidão de previsão. Desta forma, é possível constatar que, de facto, o estudo do caráter dos dados, baseado nas características das séries temporais, permite obter informação útil para a escolha dos modelos de previsão mais adequados.

O trabalho realizado demonstra ainda que a aplicação do método PCA, para a seleção das características, possibilita previsões mais fiáveis, evidenciando a importância da utilização do PCA para a obtenção de uma boa performance do algoritmo SVM.

Conclui-se, então, que a *framework* sugerida oferece um mecanismo que permite gerar, de forma eficaz e expedita, previsões para os diversos SKUs da lojas Pingo Doce, alcançando, assim, o objetivo desta dissertação.

4.2 Trabalho Futuro

Embora a metodologia proposta tenha atingido os resultados pretendidos, no futuro, seria pertinente estudar técnicas alternativas às que foram aplicadas na implementação da metodologia sugerida.

Apesar dos resultados terem demonstrado que as características das séries temporais fornecem informação útil para a seleção dos modelos mais apropriados, também seria importante testar o algoritmo para o caso da seleção do modelo, para cada série, ser baseada nas características dos modelos, nomeadamente o AIC, BIC, testes de Portmanteau para autocorrelação e os parâmetros dos modelos.

Em futuras investigações, seria interessante comparar o PCA com outros métodos de seleção de características utilizados na literatura, como o KPCA e o ICA, analisando a performance da *framework* para os três casos.

Uma outra sugestão para trabalho futuro seria incluir, no conjunto de modelos candidatos, modelos combinados, os quais poderão trazer melhorias ao nível do desempenho da *framework*. Contudo, dado que combina as previsões de vários métodos, esta classe de modelos pode exigir um tempo de processamento elevado, pondo em causa a sua aplicabilidade, visto que se pretende uma ferramenta de previsão rápida.

Todas as sugestões apresentadas são vantajosas, não só por serem admissíveis de produzir boas soluções, mas também por possibilitarem um termo de comparação.

Bibliografia

Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459.

Arana-Daniel, N., Alanis, A. Y., and López-Franco, C. (2018). Data Classification Using Support Vector Machines Trained with Evolutionary Algorithms Employing Kernel Adatron. In *Bio-inspired Algorithms for Engineering*, chapter 2, pages 15–32.

Au, K. F., Choi, T. M., and Yu, Y. (2008). Fashion retail forecasting by evolutionary neural networks. *International Journal of Production Economics*, 114(2):615–630.

Aye, G. C., Balcilar, M., Gupta, R., and Majumdar, A. (2015). Forecasting aggregate retail sales: The case of South Africa. *International Journal of Production Economics*, 160:66–79.

Box, G. E. P. and Jenkins, G. M. (1994). *Time Series Analysis: Forecasting and Control*. Prentice Hall PTR, USA, 3rd edition.

Box, G. E. P., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2016). *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, Hoboken, New Jersey, 5 edition.

Brockwell, P. J. and Davis, R. A. (2002). *Introduction to Time Series and Forecasting*, volume 92. Springer-Verlag New York, 2 edition.

Brown, R. G. (1959). *Statistical forecasting for inventory control*. McGraw-Hill, New York, NY.

Burges, C. J. (1998). *A Tutorial on Support Vector Machines for Pattern Recognition*, volume 2.

Casella, G., Fienberg, S., and Olkin, I. (2006). *Springer Texts in Statistics*, volume 102.

Chan-Lau, J. (2017). Lasso regressions and forecasting models in applied stress testing. *IMF Working Papers*, 17:1.

Chen, M. Y. (2011). Bankruptcy prediction in firms with statistical and intelligent techniques and a comparison of evolutionary computation approaches. *Computers and Mathematics with Applications*, 62(12):4514–4524.

Choi, T. M., Hui, C. L., Liu, N., Ng, S. F., and Yu, Y. (2014). Fast fashion sales forecasting with limited data and time. *Decision Support Systems*, 59(1):84–92.

Cleveland, R., Cleveland, W. S., McRae, J. E., and Terpenning, I. J. (1990). Stl: A seasonal-trend decomposition procedure based on loess.

Cortes, C. and Vapnik, V. (1995). Support-Vector Networks. *Mach Learn*, 20:273–297.

Cryer, J. D. and Chan, K.-S. (2009). *Time series analysis with applications in R*. Springer.

De Livera, A. M., Hyndman, R. J., and Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496):1513–1527.

Feki, A., Ishak, A. B., and Feki, S. (2012). Feature selection using Bayesian and multiclass Support Vector Machines approaches: Application to bank risk prediction. *Expert Systems with Applications*, 39(3):3087–3099.

Feng Yun and Ma Jun-hai (2008). Demand forecasting in supply chains based on ARMA (1,1) demand process. *Industrial Engineering Journal*, 11(5).

Fildes, R. and Petropoulos, F. (2015). Simple versus complex selection rules for forecasting many time series. *Journal of Business Research*, 68(8):1692–1701.

Fleming, P. J. and Wallace, J. J. (1986). How not to lie with statistics: The correct way to summarize benchmark results. *Commun. ACM*, 29(3):218–221.

Gao, Y., Liang, Y., Zhan, S., Ren, X., and Ou, Z. (2011). Realization of a demand forecasting algorithm for retail industry. In *Chinese Control and Decision Conference (CCDC)*, pages 4227–4230. IEEE.

Gardner, E. S. (1985). Exponential smoothing: The state of the art. *Journal of Forecasting*, 4(1):1–28.

Gholami, R. and Fakhari, N. (2017). Support Vector Machine: Principles, Parameters, and Applications. In *Handbook of Neural Computation*, chapter 27, pages 515–535. Elsevier Inc., 1 edition.

Goerg, G. M. (2013). Forecastable component analysis. In *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, ICML'13, page II–64–II–72. JMLR.org.

Gogas, P., Papadimitriou, T., and Agrapetidou, A. (2018). Forecasting bank failures and stress testing: A machine learning approach. *International Journal of Forecasting*, 34(3):440–455.

Grossmann, W. and Rinderle-Ma, S. (2015). *Fundamentals of Business Intelligence*.

Guerrero, V. M. (1993). Time-series analysis supported by power transformations. *Journal of Forecasting*, 12(1):37–48.

Hastie, T., Tibshirani, R., and Wainwright, M. (2015). *Statistical Learning with Sparsity: The Lasso and Generalizations*. Chapman Hall/CRC.

Hejazi, M., Al-Haddad, S. A., Singh, Y. P., Hashim, S. J., and Aziz, A. F. A. (2015). Multiclass Support Vector Machines for Classification of ECG Data with Missing Values. *Applied Artificial Intelligence*, 29(7):660–674.

Hira, Z. and Gillies, D. (2015). A review of feature selection and feature extraction methods applied on microarray data. *Advances in Bioinformatics*, 2015:1–13.

Holt, C. (1957). *Forecasting seasonals and trends by exponentially weighted moving averages*. Pittsburgh, o.n.r. mem edition.

Huang, C.-l., Liao, H.-c., and Chen, M.-c. (2008). Prediction model building and feature selection with support vector machines in breast cancer diagnosis. *Expert Systems with Applications*, 34:578–587.

Huang, T., Fildes, R., and 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.

Hurvich, C. M. and Tsai, C.-l. (1989). Regression and time series model selection in small samples. *Biometrika*, 76(2):297–307.

Hyndman, R. J. and Athanasopoulos, G. (2018). *Forecasting: principles and practice*. Online Open-access Textbooks. https://OTexts.com/fpp2/.

Hyndman, R. J. and Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(4):679–688.

Hyndman, R. J., Koehler, A. B., Ord, J. K., and Snyder, R. D. (2008). *Forecasting with exponential smoothing: the state space approach*. Berlin: Springer-Verlag. doi:10.1007/978-3-540-71918-2.

Hyndman, R. J., Wang, E., and Laptev, N. (2015). Large-scale unusual time series detection. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 1616–1619.

Kang, Y., Hyndman, R. J., and Smith-Miles, K. (2017). Visualising forecasting algorithm performance using time series instance spaces. *International Journal of Forecasting*, 33(2):345 – 358.

Montero-Manso, P., Athanasopoulos, G., Hyndman, R. J., and Talagala, T. S. (2020). FFORMA: Feature-based forecast model averaging. *International Journal of Forecasting*, 36(1):86–92.

Montgomery, D., Jennings, C., and Kulahci, M. (2008). *Introduction to Time Series Analysis and Forecasting*.

Nuttall, A. H. and Carter, G. C. (1982). Spectral estimation using combined time and lag weighting. *Proceedings of the IEEE*, 70(9):1115–1125.

Oliveira, J., Ramos, P., Kourentzes, N., and Fildes, R. (2020). On the use of penalized dynamic regression and time series methods for modeling and foreasting retail product sales. *Production and Operations Management*.

Pereira da Veiga, C., Pereira da Veiga, C., Catapan, A., Tortato, U., and Vieira da Silva, W. (2014//). Demand forecasting in food retail: a comparison between the holtwinters and arima models. *WSEAS Transactions on Business and Economics*, 11:608 – 14.

Petropoulos, F., Makridakis, S., Assimakopoulos, V., and Nikolopoulos, K. (2014). 'horses for courses' in demand forecasting. *European Journal of Operational Research*, 237(1):152 – 163.

Ramos, P., Santos, N., and Rebelo, R. (2015). Performance of state space and ARIMA models for consumer retail sales forecasting. *Robotics and Computer-Integrated Manufacturing*, 34:151–163.

Ren, S., Chan, H. L., and Ram, P. (2017). A Comparative Study on Fashion Demand Forecasting Models with Multiple Sources of Uncertainty. *Annals of Operations Research*, 257(1-2):335–355.

Rice, J. R. (1976). The algorithm selection problem. *Advances in Computers*, 15:65–118.

Rocha, M., Cortez, P., and Neves, J. (2007). Evolution of neural networks for classification and regression. *Neurocomputing*, 70(16-18):2809–2816.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, 6(2):461–464.

Shen, L., Wang, H., Xu, L. D., Ma, X., Chaudhry, S., and He, W. (2016). Identity management based on PCA and SVM. *Information Systems Frontiers*, 18(4):711–716.

Shumway, R. H. and Stoffer, D. S. (2011). *Time series analysis and its applications: with R examples*. New York: Springer, 3rd edition.

Smith-Miles, K. A. (2008). Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1):1–25.

Sun, Y., Li, L., Zheng, L., Hu, J., Li, W., Jiang, Y., and Yan, C. (2019). Image classification base on pca of multi-view deep representation. *Journal of Visual Communication and Image Representation*, 62:253 – 258.

Sun, Z. L., Choi, T. M., Au, K. F., and Yu, Y. (2008). Sales forecasting using extreme learning machine with applications in fashion retailing. *Decision Support Systems*, 46(1):411–419.

Talagala, T. S., Hyndman, R. J., and Athanasopoulos, G. (2018). Meta-learning how to forecast time series. *Monash Econometrics and Business Statistics Working Papers*, (May):1–29.

Teräsvirta, T., Lin, C., and Granger, C. (1993). Power of the neural network linearity test. *Journal of Time Series Analysis*, 14(2):209–220.

Timmermann, A. (2006). Forecast combinations. volume 1, chapter 04, pages 135–196. Elsevier, 1 edition.

Trapero, J., Kourentzes, N., and Fildes, R. (2014). On the identification of sales forecasting models in the presence of promotions. *Journal of the Operational Research Society*, 66:299–307.

Trapero, J., Pedregal, D., Fildes, R., and Kourentzes, N. (2013). Analysis of judgmental adjustments in the presence of promotions. *International Journal of Forecasting*, 29:234–243.

Van Donselaar, K. H., Peters, J., De Jong, A., and Broekmeulen, R. A. (2016). Analysis and forecasting of demand during promotions for perishable items. *International Journal of Production Economics*, 172:65–75.

Vilalta, R. and Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18:77–95.

Villegas, M. A., Pedregal, D. J., and Trapero, J. R. (2018). A support vector machine for model selection in demand forecasting applications. *Computers and Industrial Engineering*, 121(April):1–7.

Wanchoo, K. (2019). Retail demand forecasting: a comparison between deep neural network and gradient boosting method for univariate time series. In *2019 IEEE 5th International Conference for Convergence in Technology (I2CT)*, pages 1–5.

Wang, X., Smith-Miles, K., and Hyndman, R. (2006). Characteristic-based clustering for time series data. *Data Min. Knowl. Discov.*, 13:335–364.

Wang, X., Smith-Miles, K., and Hyndman, R. (2009). Rule induction for forecasting method selection: Meta-learning the characteristics of univariate time series. *Neurocomputing*, 72(10-12):2581–2594.

Wei, W. S. (2005). *Time series analysis: univariate and multivariate methods*. Addison Wesley, 2nd edition.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Manage. Sci.*, 6(3):324–342.

Xia, M., Zhang, Y., Weng, L., and Ye, X. (2012). Fashion retailing forecasting based on extreme learning machine with adaptive metrics of inputs. *Knowledge-Based Systems*, 36:253–259.

Anexo A

Modelos ETS

Tabela A.1: Modelos ETS com erro aditivo - adaptado de Hyndman and Athanasopoulos (2018).

		Sazonalidade	
Tendência	N	A	M
N	$y_t = \ell_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$	$y_t = \ell_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/\ell_{t-1}$
A	$y_t = \ell_{t-1} + b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t$ $b_t = b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + b_{t-1})$
A_d	$y_t = \ell_{t-1} + \phi b_{t-1} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$	$y_t = \ell_{t-1} + \phi b_{t-1} + s_{t-m} + \varepsilon_t$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t$ $b_t = \phi b_{t-1} + \beta\varepsilon_t$ $s_t = s_{t-m} + \gamma\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha\varepsilon_t/s_{t-m}$ $b_t = \phi b_{t-1} + \beta\varepsilon_t/s_{t-m}$ $s_t = s_{t-m} + \gamma\varepsilon_t/(\ell_{t-1} + \phi b_{t-1})$

Tabela A.2: Modelos ETS com erro multiplicativo - adaptado de Hyndman and Athanasopoulos (2018).

		Sazonalidade	
Tendência	N	A	M
N	$y_t = \ell_{t-1}(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}(1+\alpha\varepsilon_t)$	$y_t = (\ell_{t-1} + s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1} + \alpha(\ell_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + s_{t-m})\varepsilon_t$	$y_t = \ell_{t-1}s_{t-m}(1+\varepsilon_t)$ $\ell_t = \ell_{t-1}(1+\alpha\varepsilon_t)$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
A	$y_t = (\ell_{t-1} + b_{t-1})(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1} + s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1} + b_{t-1} + \alpha(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + b_{t-1})s_{t-m}(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1} + b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$
A_d	$y_t = (\ell_{t-1} + \phi b_{t-1})(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1} + s_{t-m})(1+\varepsilon_t)$ $\ell_t = \ell_{t-1} + \phi b_{t-1} + \alpha(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$ $s_t = s_{t-m} + \gamma(\ell_{t-1} + \phi b_{t-1} + s_{t-m})\varepsilon_t$	$y_t = (\ell_{t-1} + \phi b_{t-1})s_{t-m}(1+\varepsilon_t)$ $\ell_t = (\ell_{t-1} + \phi b_{t-1})(1+\alpha\varepsilon_t)$ $b_t = \phi b_{t-1} + \beta(\ell_{t-1} + \phi b_{t-1})\varepsilon_t$ $s_t = s_{t-m}(1+\gamma\varepsilon_t)$

Anexo B

Características das Séries Temporais

Tabela B.1: Características de séries temporais usadas na metodologia proposta.

Característica		Descrição
1	trend	Numa decomposição STL da série temporal com r_t como série residual e z_t como a série desprovida de sazonalidade: $max[0, 1 - Var(r_t)/Var(z_t)]$.
2	seasonality	Numa decomposição STL da série temporal com r_t como série residual e x_t como a série desprovida de tendência: $max[0, 1 - Var(r_t)/Var(x_t)]$.
3	linearity	Numa decomposição STL da série temporal sendo T_t a componente tendência, ajustar o modelo quadrático: $T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \varepsilon_t$. A linearidade é o coeficiente β_1.
4	curvature	Numa decomposição STL da série temporal sendo T_t a componente tendência, ajustar o modelo quadrático: $T_t = \beta_0 + \beta_1 t + \beta_2 t^2 + \varepsilon_t$. A curvatura é a coeficiente β_2.
5	spikiness	Numa decomposição STL da série temporal com r_t como série residual, a variância das variâncias *leave one out* de r_t.
6	e_acf1	O primeiro valor da ACF da série residual de uma decomposição STL da série temporal.
7	e_acf10	A soma dos quadrados dos 10 primeiros valores da ACF da série residual de uma decomposição STL da série temporal.
8	stability	A variância das médias baseada numa divisão da série em intervalos não sobrepostos. O tamanho dos intervalos é igual à frequência da série ou 10 no caso da frequência ser 1.
9	entropy	A entropia espetral da série temporal. $H_s(x_t) = -\int_{-\pi}^{\pi} f_x(\lambda) \log f_x(\lambda) d\lambda$ onde a densidade é normalizada logo $\int_{-\pi}^{\pi} f_x(\lambda) d\lambda = 1$.
10	lumpiness	A variância das variâncias baseada numa divisão da série em intervalos não sobrepostos. O tamanho dos intervalos é igual à frequência da série ou 10 no caso da frequência ser 1.
11	hurst	O coeficiente de hurst, que indica o nível de diferenciação fracional de uma série temporal.
12	nonlinearity	Uma estatística de não linearidade baseada no teste Tearsvirta de não linearidade de uma série temporal.
13	alpha	O parâmetro de alisamento do nível num modelo ETS(A,A,N) ajustado à série temporal, α.
14	beta	O parâmetro de alisamento da tendência num modelo ETS(A,A,N) ajustado à série temporal, β.
15	ur_pp	O valor da estatística da versão "Z-alpha"do teste de raízes unitárias de Phillips & Perron com tendência constante e *lag* 1.
16	ur_kpss	O valor da estatística do teste de raízes unitárias de Kwiatkowski et al. com tendência linear e *lag* 1.
17	y_acf1	O primeiro valor da ACF da série temporal.

Tabela B.1: Características de séries temporais usadas na metodologia proposta.

Característica		Descrição
18	diff1y_acf1	O primeiro valor da ACF da série diferenciada.
19	diff2y_acf1	O primeiro valor da ACF da série diferenciada duas vezes.
20	y_acf10	A soma dos quadrados dos 10 primeiros valores da ACF da série temporal.
21	diff1y_acf10	A soma dos quadrados dos 10 primeiros valores da ACF da série diferenciada.
22	diff2y_acf10	A soma dos quadrados dos 10 primeiros valores da ACF da série diferenciada duas vezes.
23	seas_acf1	O coeficiente de autocorrelação do primeiro *lag* sazonal. Se a série é não sazonal então esta característica é 0.
24	y_pacf5	A soma dos quadrados dos 5 primeiros valores da PACF da série temporal.
25	diff1y_pacf5	A soma dos quadrados dos 5 primeiros valores da PACF da série diferenciada.
26	diff2y_pacf5	A soma dos quadrados dos 5 primeiros valores da PACF da série diferenciada duas vezes.
27	seas_pacf	O coeficiente de autocorrelação parcial do primeiro *lag* sazonal. Se a série é não sazonal então esta característica é 0.
28	crossing_point	O número de vezes que a série temporal cruza a mediana.
29	flat_spots	O número de *flat spots* da série temporal, calculado discretizando a série em 10 intervalos de igual comprimento e contando o maior *run lenght* em todos os intervalos.
30	peak	A localização do valor máximo da componente sazonal de uma decomposição STL da série temporal.
31	trough	A localização do "vale" (valor mínimo) da componente sazonal de uma decomposição STL da série temporal.
32	ARCH.LM	O valor da estatística baseada no teste de Multiplicador de Lagrange de Engle para heterocedasticidade condicional autorregressiva.
33	arch_acf	Depois da série ser *pre-whitened* usando um modelo AR e elevada ao quadrado. A soma dos quadrados dos 12 primeiros valores da ACF.
34	garch_acf	Depois da série ser *pre-whitened* usando um modelo AR, um modelo GARCH(1,1) é ajustado e os resíduos são calculados. A soma dos quadrados dos 12 primeiros valores da ACF dos quadrados dos resíduos.
35	arch_r2	Depois da série ser *pre-whitened* usando um modelo AR e elevada ao quadrado, o valor de R^2 de um modelo AR ajustado à mesma.
36	garch_r2	Depois da série ser *pre-whitened* usando um modelo AR, um modelo GARCH(1,1) é ajustado e os resíduos são calculados, o valor de R^2 de um modelo AR ajustado aos mesmos.

Anexo C

Estatística Descritiva das Características para as Duas Lojas

Tabela C.1: Estatística descritiva das características das séries de vendas da loja 412.

Característica		Mínimo	Q1	Mediana	Média	Q3	Máximo
1	trend	0.00000	0.07321	0.18836	0.25333	0.38784	0.90287
2	seasonality	0.1993	0.3718	0.4341	0.4600	0.5126	0.9087
3	linearity	-10.380	-4.717	-1.752	-2.026	0.590	9.960
4	curvature	-7.9538	-1.0284	0.3056	0.2505	1.4806	5.4667
5	spikiness	7.984e-07	1.766e-05	4.652e-05	1.189e-04	1.533e-04	1.190e-03
6	e_acf1	-0.41463	-0.02736	0.08588	0.10101	0.21523	0.76637
7	e_acf10	0.003822	0.066616	0.108285	0.160884	0.184196	2.758814
8	stability	0.000188	0.033620	0.108518	0.190888	0.277115	1.069793
9	entropy	0.5749	0.8947	0.9508	0.9256	0.9793	0.9996
10	lumpiness	0.0004943	0.0832366	0.2212171	0.4228534	0.5599705	2.8117457
11	hurst	0.5000	0.6190	0.7557	0.7448	0.8730	0.9978
12	nonlinearity	0.00032	0.13369	0.33184	0.52787	0.74675	3.39546
13	alpha	0.00010	0.03783	0.17188	0.18767	0.29322	0.98413
14	beta	0.000100	0.000100	0.000100	0.000747	0.000100	0.107266
15	ur_pp	-214.047	-147.088	-117.811	-111.378	-78.126	-7.268
16	ur_kpss	0.03133	0.22440	0.54159	0.79023	1.21299	2.87453
17	y_acf1	-0.2153	0.1183	0.3174	0.3191	0.5134	0.9126
18	diff1y_acf1	-0.70967	-0.50280	-0.46235	-0.44817	-0.40701	0.01797
19	diff2y_acf1	-0.8296	-0.6690	-0.6427	-0.6363	-0.6078	-0.3317
20	y_acf10	0.002084	0.108123	0.397614	0.835716	1.185060	6.566201
21	diff1y_acf10	0.04775	0.22785	0.26973	0.29038	0.33620	1.31469
22	diff2y_acf10	0.2040	0.4400	0.5013	0.5266	0.5891	1.7919
23	seas_acf1	-0.486607	-0.006632	0.052982	0.077744	0.139407	0.575433
24	y_pacf5	0.0007382	0.0620870	0.1851944	0.2457416	0.3788272	0.9601689
25	diff1y_pacf5	0.03729	0.31998	0.40256	0.40278	0.48403	0.89685
26	diff2y_pacf5	0.3873	0.8603	0.9536	0.9439	1.0398	1.4411
27	seas_pacf	-0.19827	-0.02460	0.01654	0.02956	0.07390	0.50072
28	crossing_point	8.00	41.00	52.00	51.01	61.00	88.00
29	flat_spots	2.00	5.00	7.00	11.81	12.00	108.00
30	peak	1.00	18.00	24.00	27.97	42.00	52.00
31	trough	1.00	25.00	33.00	31.84	45.00	52.00
32	ARCH.LM	0.000239	0.016968	0.067924	0.111271	0.157520	0.796386
33	arch_acf	0.0002016	0.0123364	0.0409885	0.0656795	0.0895755	0.9516111
34	garch_acf	0.0000014	0.0094119	0.0297973	0.0479609	0.0624214	0.4865402
35	arch_r2	0.000239	0.014025	0.046863	0.071847	0.096315	0.838853
36	garch_r2	0.0002863	0.0117704	0.0356617	0.0575378	0.0720339	0.8389969

Tabela C.2: Estatística descritiva das características das séries de vendas da loja 843.

Característica		Mínimo	Q1	Mediana	Média	Q3	Máximo
1	trend	0.002803	0.101309	0.228334	0.298865	0.456274	0.895842
2	seasonality	0.2428	0.3760	0.4394	0.4610	0.5171	0.9104
3	linearity	-11.0126	-5.5998	-2.3915	-2.4757	0.3494	10.1194
4	curvature	-7.7784	-0.9666	0.1683	0.2404	1.4915	6.3831
5	spikiness	6.637e-7	1.415e-5	3.480e-5	8.743e-5	9.077e-5	1.145e-3
6	e_acf1	-0.37399	-0.02840	0.09355	0.10668	0.22218	0.85913
7	e_acf10	0.01009	0.07248	0.11354	0.17178	0.18846	2.81259
8	stability	0.0003953	0.0547865	0.1591239	0.2451535	0.3628168	1.0825242
9	entropy	0.5881	0.8748	0.9380	0.9123	0.9727	0.9992
10	lumpiness	0.0000762	0.061131	0.0140606	0.3377542	0.459097	2.612448
11	hurst	0.5000	0.6558	0.7902	0.7721	0.8932	0.9990
12	nonlinearity	0.000575	0.149523	0.373522	0.573374	0.831974	3.201985
13	alpha	0.00010	0.07809	0.18384	0.20296	0.29705	0.99990
14	beta	0.000100	0.000100	0.000100	0.0001003	0.000100	0.087527
15	ur_pp	-215.621	-143.707	-107.402	-105.250	-70.551	-5.912
16	ur_kpss	0.03972	0.28809	0.75276	0.96018	1.55562	3.00312
17	y_acf1	-0.2590	0.1556	0.3634	0.3599	0.5451	0.9499
18	diff1y_acf1	-0.72551	-0.49856	-0.45397	-0.44302	-0.39849	0.05675
19	diff2y_acf1	-0.8385	-0.6701	-0.6413	-0.6343	-0.6060	-0.3004
20	y_acf10	0.00307	0.18010	0.55364	1.05014	1.49588	6.44286
21	diff1y_acf10	0.03826	0.22639	0.27701	0.28782	0.33397	1.15326
22	diff2y_acf10	0.2105	0.4451	0.5054	0.5265	0.5893	1.5937
23	seas_acf1	-0.337712	-0.005488	0.059005	0.077462	0.136815	0.538635
24	y_pacf5	0.001012	0.096667	0.231308	0.288777	0.440297	1.009330
25	diff1y_pacf5	0.02981	0.31753	0.39680	0.39881	0.48150	0.93801
26	diff2y_pacf5	0.4074	0.8569	0.9523	0.9413	1.0338	1.4282
27	seas_pacf	-0.16673	-0.02227	0.02470	0.03059	0.07272	0.42898
28	crossing_point	7.00	38.00	49.00	48.35	58.00	87.00
29	flat_spots	3.00	5.00	6.00	10.17	10.00	117.00
30	peak	1.00	18.00	27.00	28.83	44.00	52.00
31	trough	1.00	24.00	33.00	31.74	45.00	52.00
32	ARCH.LM	0.0004881	0.0280477	0.0794947	0.1250048	0.1792217	0.7637426
33	arch_acf	0.0003378	0.0198539	0.0521847	0.0733563	0.0962240	0.6714918
34	garch_acf	0.0001082	0.0131410	0.0361233	0.0495753	0.0676114	0.3583273
35	arch_r2	0.0004881	0.0217880	0.0589084	0.0759717	0.1061966	0.4295418
36	garch_r2	0.000497	0.015827	0.042970	0.058745	0.078850	0.951333

Anexo D

Heatmaps Obtidos na Parametrização dos SVMs

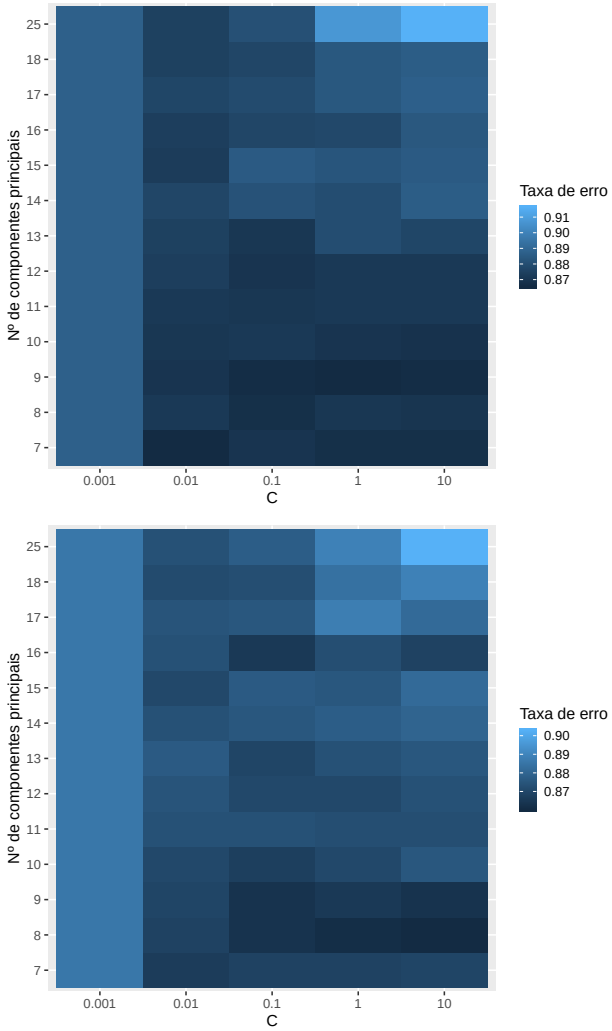

Figura D.1: *Heatmaps* relativos à parametrização do SVM com Kernel linear para o caso do critério de seleção ser o MAE (à esquerda) e o MASE (à direita).

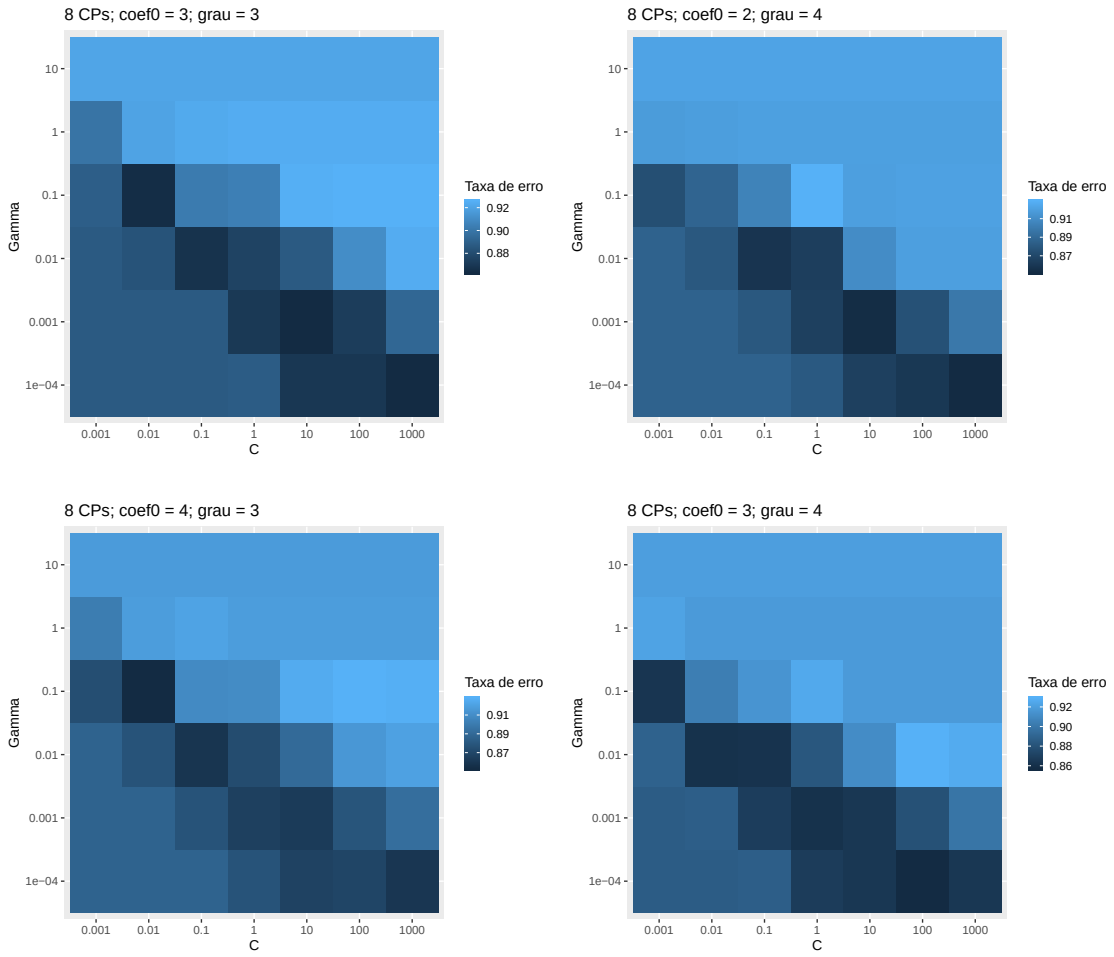

Figura D.2: Exemplos de *Heatmaps* relativos à parametrização do SVM com Kernel polinomial para o caso do critério de seleção ser o MAE (à esquerda) e o MASE (à direita).

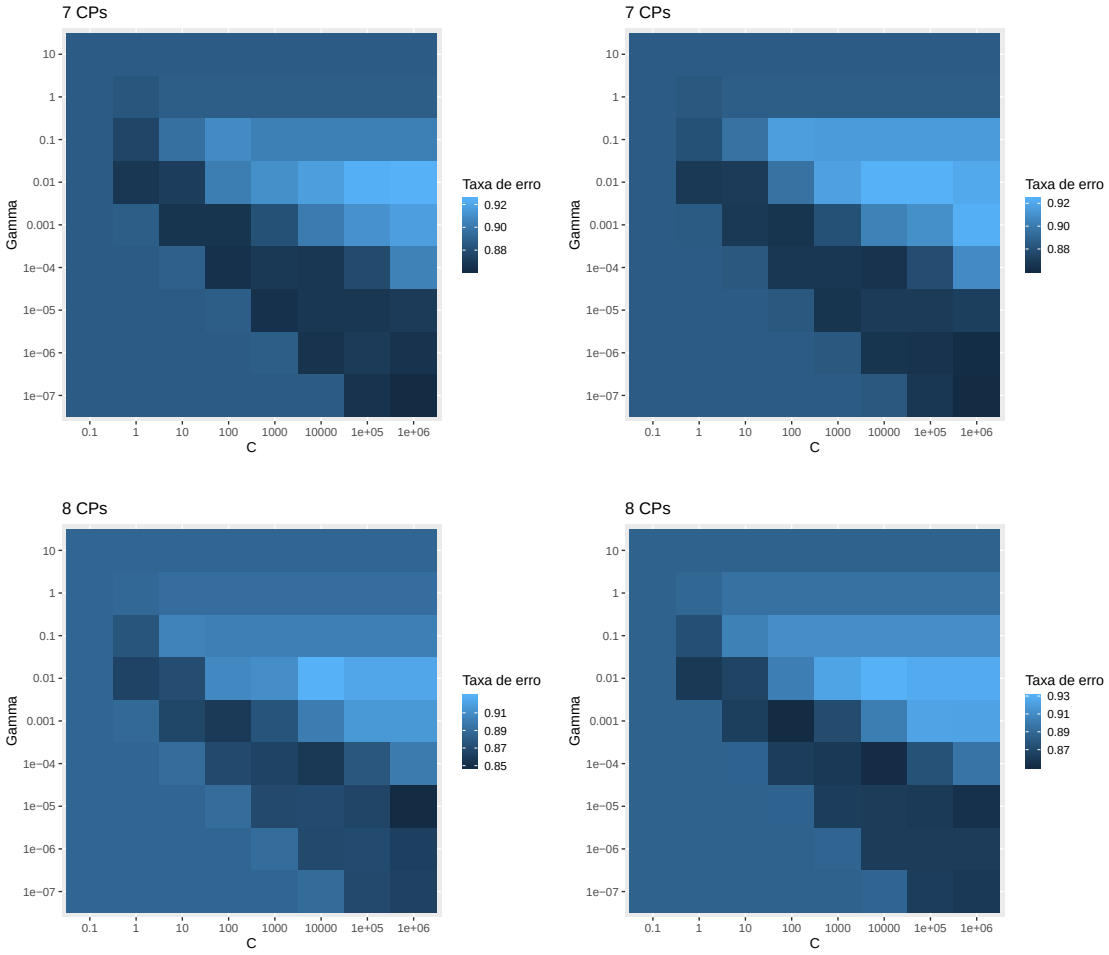

Figura D.3: Exemplos de *Heatmaps* relativos à parametrização do SVM com Kernel radial para o caso do critério de seleção ser o MAE (à esquerda) e o MASE (à direita).

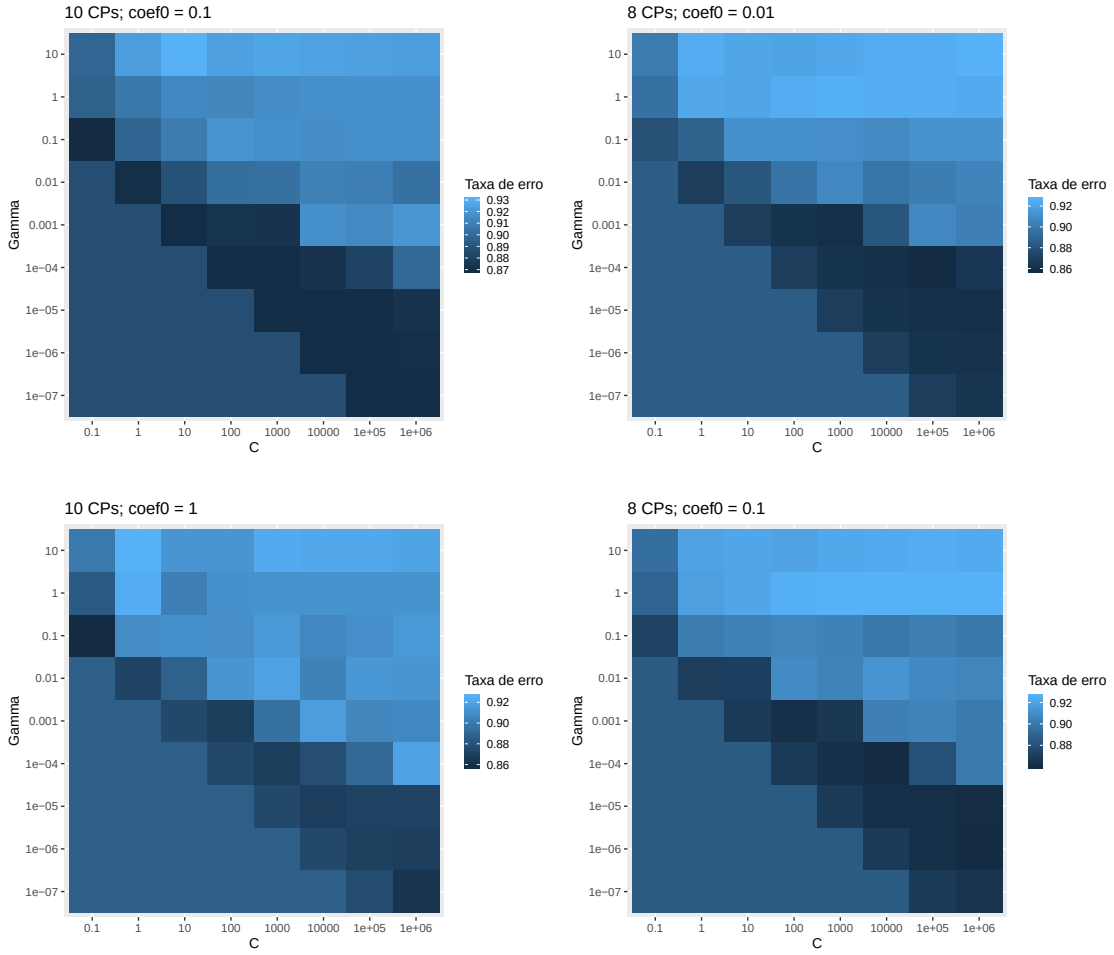

Figura D.4: Exemplos de *Heatmaps* relativos à parametrização do SVM com Kernel sigmoide para o caso do critério de seleção ser o MAE (à esquerda) e o MASE (à direita).