

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
BACHARELADO EM MATEMÁTICA APLICADA E COMPUTACIONAL

**Predição de séries temporais utilizando
técnicas clássicas e de aprendizado de
máquina**

*Uma aplicação a índices setoriais
econômicos*

Richard Sousa Antunes

MONOGRAFIA

MAP 2040 — TRABALHO DE
FORMATURA

Orientador: Prof. Dr. Flavio Almeida de Magalhães Cipparrone

São Paulo
2023

*O conteúdo deste trabalho é publicado sob a licença CC BY 4.0
(Creative Commons Attribution 4.0 International License)*

Ficha catalográfica elaborada com dados inseridos pelo(a) autor(a)
Biblioteca Carlos Benjamin de Lyra
Instituto de Matemática e Estatística
Universidade de São Paulo

Antunes, Richard Sousa

Predição de séries temporais utilizando técnicas clássicas
e de aprendizado de máquina: Uma aplicação a índices setoriais
econômicos / Richard Sousa Antunes; orientador, Flavio Almeida
de Magalhães Cipparrone. – São Paulo, 2023.

77 p.: il.

Trabalho de Conclusão de Curso (Graduação) – Matemática
Aplicada / Instituto de Matemática e Estatística /
Universidade de São Paulo.

Bibliografia

1. MODELOS EM SÉRIES TEMPORAIS. 2. MODELOS ECONÔMETRICOS.
3. APRENDIZADO COMPUTACIONAL. 4. REDES NEURAIS. I. Almeida
de Magalhães Cipparrone, Flavio. II. Título.

Bibliotecárias do Serviço de Informação e Biblioteca
Carlos Benjamin de Lyra do IME-USP, responsáveis pela
estrutura de catalogação da publicação de acordo com a AACR2:
Maria Lúcia Ribeiro CRB-8/2766; Stela do Nascimento Madruga CRB 8/7534.

Agradecimentos

Investir em conhecimento rende sempre os melhores juros

— Benjamin Franklin

Em primeiro lugar, sou grato a Deus por me guiar ao longo de minha vida e pelas inúmeras oportunidades que me foram dadas. Aos meus pais, minha gratidão pelo apoio e companhia na minha jornada.

Ao Professor Flavio, meu orientador, expresso minha gratidão pela paciência e pela disponibilidade em me orientar, assim como pela organização dos seminários de Matemática na Indústria, que ele promoveu juntamente com outros colaboradores. Em especial, agradeço ao Marcelo, que contribuiu com sugestões valiosas para o meu trabalho.

A minhas professoras de Matemática do ensino fundamental e médio, Otilia e Vania, respectivamente, que nutriram minha paixão pela matemática, incentivando-me a buscar o conhecimento com dedicação e participar de olimpíadas científicas, o que culminou na minha escolha pela graduação em Matemática Aplicada e Computacional.

Aos amigos de longa data, David, Henrique, Sarah e Thamires, que me ofereceram suporte nos momentos desafiadores, e aos que conheci na Universidade de São Paulo, no Instituto de Matemática e Estatística, companheiros valiosos na jornada acadêmica que guardarei para a vida, incluindo Guilherme, Gabriel, Milton, Felipe, Pedro, Lucka, Kauê, Peterson, Ian, Luis, Gabriela e tantos outros.

Ao grupo de extensão BeeData, pelo papel fundamental em introduzir-me ao mundo dos dados e por despertar em mim o desejo de aprender cada vez mais sobre este campo. Um agradecimento especial a Ana, Giovanni e Renato pelos diversos projetos e eventos que realizamos juntos ao longo do ano.

Por fim, sou grato aos amigos das repúblicas Amora e Galvão, com quem tive o prazer de compartilhar momentos de alegria e descontração em meio ao caos diário da vida; em especial, ao Marcos, pela companhia e apoio nas madrugadas de estudos.

Resumo

Richard Sousa Antunes. **Predição de séries temporais utilizando técnicas clássicas e de aprendizado de máquina : Uma aplicação a índices setoriais econômicos.**

Monografia (Bacharelado). Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2023.

Este trabalho explora a aplicação e comparação de técnicas clássicas e modernas de análise de séries temporais na previsão do comportamento de setores econômicos dos Estados Unidos, utilizando dados das ações listadas no S&P 500 agrupados em índices setoriais. O foco é em dois métodos tradicionais: Vetores Auto-Regressivos (VAR) e Vetores Auto-Regressivos com Médias Móveis (VARMA), bem como em uma abordagem moderna de aprendizado de máquina, que utiliza a Rede Neural Multicamadas (MLP). Através da comparação dessas metodologias, buscamos oferecer uma visão abrangente das capacidades preditivas dessas técnicas, abordando desde a coleta e tratamento dos dados até a implementação e avaliação dos modelos. Este trabalho não só introduz os modelos básicos amplamente utilizados na econometria e no aprendizado de máquina, mas também investiga suas aplicações práticas na modelagem de setores econômicos. Os resultados obtidos demonstram a complexidade e dificuldade de prever com sucesso o futuro do mercado financeiro e econômico com abordagens tradicionais e modernas na análise de séries temporais, oferecendo insights valiosos para futuras pesquisas e aplicações práticas no campo da previsão econômica.

Palavras-chave: Séries Temporais. Setores Econômicos. Modelos Auto-Regressivos. Aprendizado de Máquina. Redes Neurais.

Abstract

Richard Sousa Antunes. **Time series prediction using classical and machine learning techniques: *An application to economic sector indices***. Undergraduate Thesis (Bachelor). Institute of Mathematics and Statistics, University of São Paulo, São Paulo, 2023.

This paper explores the application and comparison of classic and modern techniques for time series analysis in forecasting the behavior of economic sectors in the United States, using data from stocks listed on the S&P 500 grouped into sectoral indices. The focus is on two traditional methods: Vector Autoregression (VAR) and Vector Autoregression with Moving Averages (VARMA), as well as on a modern machine learning approach, which uses the Multilayer Perceptron (MLP). Through the comparison of these methodologies, we aim to provide a comprehensive view of the predictive capabilities of these techniques, addressing everything from data collection and processing to implementation and evaluation of the models. This work not only introduces the basic models widely used in econometrics and machine learning but also investigates their practical applications in modeling economic sectors. The results obtained demonstrate the complexity and difficulty of successfully predicting the future of the financial and economic market with traditional and modern approaches in time series analysis, offering valuable insights for future research and practical applications in the field of economic forecasting.

Keywords: Time Series. Economic Sectors. autoregressive Models. Machine Learning. Neural Networks.

Lista de abreviaturas

S&P 500	Standard and Poor's 500
GICS	Global Industry Classification Standard
MSCI	Morgan Stanley Capital International
AR	Autoregressive model
MA	Moving-Average model
VAR	Vector Autoregression model
VMA	Vector Moving Average model
VARMA	Vector Auto-regressive Moving Average model
MLP	Multi-layer Perceptron
Var	Variância
COV	Covariância
RB	Ruído Branco
RBG	Ruído Branco Gaussiano
WN	White Noise
GWN	Gaussian White Noise
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
HQ	Hannan-Quinn Information Criterion
SC	Schwarz Criterion
MQO	Mínimos Quadrados Ordinários
MQG	Mínimos Quadrados Generalizados
MV	Máxima Verossimilhança
IA	Inteligência Artificial
ADF	Augmented Dickey–Fuller Test
MSE	Mean Square Error
RMSE	Root Mean Square Error
ReLU	Rectified Linear Unit

Lista de símbolos

\mathcal{Z}	Processo estocástico
Z_t	Série temporal no instante t
Ω	Espaço amostral
\mathcal{A}	σ -álgebra
\mathcal{P}	Medida de probabilidade
\mathbb{P}	Função de probabilidade
\mathbb{E}	Esperança matemática
$\ln \mathcal{L}$	log-verossimilhança
μ	Média da série temporal
ρ	Coefficiente de correlação
\mathcal{D}	Matriz de autocorrelação
γ	Função de autocovariância
Γ	Matriz de autocovariância
ε_t	Ruído branco
L	Operador de defasagem
Δ_p	Operador de p-ésima diferença
η	Intercepto
δ	Vetor de intercepto
u_t	Ruído branco
U_t	Vetor de ruído branco
α_i	Coefficiente dos componentes auto-regressivos
β_i	Coefficiente dos componentes de médias móveis
Φ_i	Matriz de coeficientes dos componentes vetoriais auto-regressivos
Θ_i	Matriz de coeficientes dos componentes vetoriais de médias móveis
Ξ	Matriz de coeficientes dos componentes determinísticos
Λ	Matriz de coeficientes dos componentes Exógenos
D	Vetor de componentes determinísticos
G	Vetor de componentes exógenos
$A(L), B(L)$	Polinômios de operador Lag
Σ_u	Matriz de covariâncias
\otimes	Produto de Kronecker
\mathcal{R}	Retorno médio

Lista de figuras

2.1	Índice S&P500 de Jan/22 a Set/23	6
2.2	evolução do preço de fechamento das ações GOOG, NVDA, TSLA e MDLZ (listadas na S&P) de Jan/22 a Set/23	7
4.1	Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo. Fonte: CHOLLET, 2018	18
4.2	Exemplo de rede neural com 4 componentes na entrada X , 1 na saída $f(X) = Y$ e 5 ativações (não observáveis) na camada oculta única. Fonte: JAMES <i>et al.</i> , 2023	22
4.3	Funções de ativação Sigmoide e ReLU. ReLU com escala reduzida em um fator de cinco para facilitar a comparação de suas formas. Fonte: JAMES <i>et al.</i> , 2023	23
4.4	Exemplo de uma rede neural multi camadas com múltiplas entradas, 2 camadas ocultas e 10 saídas. Fonte: JAMES <i>et al.</i> , 2023	24
5.1	Retorno médio do Setor de Tecnologia	29
5.2	Retorno médio do Setor de Serviços de Comunicação	29
5.3	Retorno médio do Setor de Consumo Cíclico	29
5.4	Retorno médio do Setor de Saúde	30
5.5	Retorno médio do Setor de Serviços Financeiros	30
5.6	Retorno médio do Setor de Bens de Consumo Discricionário	30
5.7	Retorno médio do Setor de Energia	31
5.8	Retorno médio do Setor de Materiais Básicos	31
5.9	Retorno médio do Setor Industrial	31
5.10	Retorno médio do Setor de Serviços Públicos	32
5.11	Retorno médio do Setor Imobiliário	32
6.1	Divisão do conjunto de dados em treinamento, validação e teste. Fonte: GREG BLAND (2020)	39

6.2	Processo de treinamento com conjuntos de treino e validação, escolha do modelo e uso do teste para checar a performance final. Fonte: AUFFARTH (2021)	39
6.3	Exemplo de Cross-Validation k-folds com 5 folds. Fonte: PEDREGOSA et al. (2011)	40
6.4	Técnicas de Janelas Expansivas e Janelas Móveis para separação entre treino e teste dos dados. Fonte: CHAN e MÁTYÁS (2022)	42

Lista de tabelas

5.1	Top 10 empresas em valor de mercado dos setores de Tecnologia, Serviços de comunicação e Consumo Cíclico	27
5.2	Top 10 empresas em valor de mercado dos setores de Serviços Financeiros, Saúde e Energia	28
5.3	Top 10 empresas em valor de mercado dos setores de Consumo Defensivo, Materiais Básicos e Industriais	28
5.4	Top 10 empresas em valor de mercado dos setores de Utilidades Públicas e Imobiliário	28
5.5	Exemplo de dados de aprendizado de máquina	33
5.6	Exemplo de dados de série temporal	33
5.7	Exemplo de uma série temporal bivariada transformada em um problema de aprendizado supervisionado para predição de 1 passo à frente utilizando 1 passo anterior	33
5.8	Remodelagem dos dados de série temporal para aprendizado supervisionado para predição de 2 passos À frente utilizando 3 passos anteriores	34
5.9	Remodelagem simplificada dos dados de série temporal para aprendizado supervisionado para predição de 2 passos à frente utilizando 3 passos anteriores	34
5.10	Série temporal generalizada em formato de tabela	35
5.11	Generalização da remodelagem dos dados de série temporal para aprendizado supervisionado utilizando p passos anteriores para prever até h passos à frente	35

5.12	Dados de série temporal achatados para aprendizado supervisionado . . .	35
5.13	Exemplo simples de uma série temporal multivariada. Fonte: BROWNLEE (2018, p. 81)	36
5.14	Exemplo simples da remodelagem de uma série temporal multivariada para o formato de aprendizado supervisionado. Fonte: BROWNLEE (2018, p. 83)	36
5.15	Exemplo remodelado para ser possível aplicar MLP	37
7.1	Critérios de informação no modelo VAR nas 3 janelas de treinamentos para cada valor da ordem p	44
7.2	Resultados de MSE e RMSE do VAR(1) para diferentes períodos de treinamento	46
7.3	Resultados dos critérios de informação AIC, BIC e HQ do modelo VARMA para diferentes valores de p e q	47
7.4	Resultados do modelo para diferentes datas de treinamento.	48
7.5	Comparação da RMSE para cada um dos modelos para diferentes setores.	50
7.6	Média do RMSE por modelo	50
7.7	Retorno acumulado por setor.	50

Lista de programas

A.1	Extração dos 500 símbolos da S&P	53
A.2	Extração do valor de mercado das 500 ações da S&P	53
A.3	Extraí top 10 maiores ações em valor de mercado de cada setor	54
A.4	Download dos dados para dar início ao trabalho	54
B.1	Divisão das janelas de treino e teste	56
B.2	Verificação de critérios de informação para escolha da ordem de VAR(p) .	56
B.3	Treino e predição do modelo VAR(1)	57
B.4	Verificação de critérios de informação para escolha da ordem de VARMA(p,q)	57
B.5	Treino e predição do modelo VARMA(1,1)	58
B.6	Função para transformar no tipo X e Y	59
B.7	Implementação do MLP com uso de 1 dia anterior e com 5 dias anteriores, modelo com 3 camadas ocultas, com 30, 100 e 30 ativações cada	59

Sumário

Introdução	1
1 Contexto Econômico	3
1.1 Setores Econômicos	3
1.2 Descrição dos setores da S&P	4
2 Conceitos de Séries temporais	6
2.1 Séries Temporais	6
2.2 Estacionariedade	7
2.3 Ruído Branco	8
2.4 Operador Lag (defasagem)	9
3 Modelos Tradicionais de Séries Temporais	10
3.1 Modelos univariados	10
3.1.1 Modelo Auto-Regressivo (AR)	10
3.1.2 Modelo de Média Móvel (MA)	11
3.1.3 Modelo Auto-Regressivo com Média Móvel (ARMA)	12
3.2 Modelos Multivariados	12
3.2.1 Modelo Vetorial Auto-Regressivo (VAR)	12
3.2.2 Modelo Vetorial Auto-Regressivo de Médias Móveis (VARMA)	16
4 Modelos com Aprendizado de Máquina	18
4.1 Inteligência Artificial, Aprendizado de Máquina e Aprendizado profundo	18
4.1.1 Inteligência Artificial	19
4.1.2 Aprendizado de Máquina	19
4.1.3 Aprendizado Profundo	21
4.2 Redes Neurais	21
4.2.1 Redes Neurais de camada única	21
4.2.2 Redes Neurais de Múltiplas Camadas (MLP)	23

5	Extração e tratamento de dados	25
5.1	Extração dos dados	25
5.2	Dados nulos	26
5.3	Transformação dos dados	26
5.3.1	Top 10 ações escolhidas de cada setor	27
5.3.2	Retornos médios dos setores	29
5.4	Remodelagem de séries temporais para um problema de Aprendizado Supervisionado	32
5.4.1	Formato dos dados para MLP	35
5.4.2	Exemplo prático	36
6	Técnicas de treinamento e avaliação dos modelos	38
6.1	Divisão do conjunto inicial de dados	38
6.2	Cross-Validation k-folds em aprendizado de máquina com dados não sequenciais	40
6.2.1	O problema de aplicar Cross-Validation k-folds em séries temporais	41
6.3	Cross-Validation de Séries Temporais	42
6.3.1	Janelas Expansivas dos dados que serão utilizadas	43
7	Implementação dos modelos	44
7.1	VAR(p)	44
7.2	VARMA(p, q)	46
7.3	MLP	48
7.4	Erros finais	49
8	Conclusão	51

Apêndices

A	Código de extração dos dados	53
B	Código de implementação dos modelos	56
B.1	VAR	56
B.2	VARMA	57
B.3	MLP	59

Anexos

Referências

61

Introdução

Desde o início, a análise de séries temporais tem sido extensivamente utilizada em finanças e economia, com modelos econométricos desempenhando um papel crucial na modelagem e previsão de eventos econômicos. Essas técnicas permitem aos analistas e pesquisadores capturar tendências, ciclos e padrões em dados históricos para fazer projeções futuras. Além de sua importância em economia e finanças, a análise de séries temporais é fundamental em áreas como meteorologia, engenharia e saúde pública.

Modelos clássicos, como os modelos auto-regressivos (AR) e de médias móveis (MA), têm sido amplamente adotados devido à sua eficácia em identificar padrões temporais e facilidade de interpretação. Com o avanço tecnológico na computação, modelos de aprendizado de máquina, especialmente redes neurais, destacaram-se pela capacidade de processar grandes volumes de dados e identificar complexidades que modelos tradicionais podem não capturar.

A combinação dessas abordagens tradicionais com técnicas modernas de aprendizado de máquina oferece novas perspectivas para a análise de séries temporais, possibilitando uma compreensão mais aprofundada e previsões mais acuradas. Modelos de Perceptrons Multicamadas (MLP), especificamente, fornecem uma flexibilidade significativa para modelar relações não-lineares sem a exigência de especificar a forma do modelo antecipadamente, tornando-os adequados para adaptar-se a padrões complexos nos dados.

Este estudo visa aplicar e comparar brevemente duas técnicas clássicas de análise de séries temporais multivariadas — Vetores Auto-Regressivos (VAR) e Vetores Auto-Regressivos com Médias Móveis (VARMA) — e uma técnica avançada de aprendizado de máquina, a Rede Neural Multicamadas (MLP), para prever o comportamento de setores econômicos dos Estados Unidos utilizando dados de ações listadas no S&P 500.

O objetivo deste trabalho é oferecer uma introdução aos modelos básicos amplamente empregados na econometria e aos modelos de aprendizado de máquina, cobrindo todo o processo de predição, desde a coleta e transformação dos dados até a predição e a avaliação dos resultados. Além disso, o estudo visa explorar possibilidades adicionais que possam ser aplicadas no contexto de análise de séries temporais em geral, culminando na aplicação específica aos dados de setores econômicos americanos, conforme proposto. A estrutura do texto é organizada da seguinte forma:

Contexto econômico: introduz os setores econômicos do mercado americano e a S&P, o qual será a fonte dos dados que serão usados para modelar os setores.

Conceitos de Séries Temporais: aborda conceitos estatísticos importantes de séries

temporais que são requisitos fundamentais para aplicação dos modelos de predição.

Modelos Tradicionais: introduz brevemente a teoria por trás dos modelos clássicos VAR e VARMA, desde a formulação matemática univariada e multivariada, até a técnica utilizada para modelá-los aos dados.

Modelos com Aprendizado de Máquina: fornece uma introdução ao contexto de aprendizado de máquina e uma breve apresentação do modelo de Perceptron Multicamadas.

Extração e tratamento de dados: detalha o processo de extração e tratamento dos dados utilizado, modelando as ações aos setores econômicos, e realizando transformações nos dados para uma forma adequada para os modelos tradicionais e de aprendizado de máquina.

Técnicas de treinamento e avaliação dos modelos: exhibe as técnicas de divisão dos dados a fim de ajustar os modelos aos dados e formas de avaliar a predição calculada, utilizando algumas métricas adequadas.

Implementação dos modelos: implementa os modelos com as técnicas e procedimentos fornecidos ao longo do texto e traz os resultados obtidos.

Conclusão: sumariza as principais discussões realizadas neste trabalho e sugere possíveis direções para futuras pesquisas.

Apêndice: contém códigos fontes em Python, utilizados para implementar os modelos e obter os resultados.

Capítulo 1

Contexto Econômico

Historicamente, a economia de um país tem sido um reflexo da dinâmica econômica global, evoluindo desde simples negociações de ações em praças de comércio até o intrincado sistema de mercados interconectados de hoje. Esta evolução ilustra não apenas a crescente complexidade e globalização financeira, mas também a maneira como eventos em um país ou setor podem ter repercussões mundiais.

Com essa interconexão global, os mercados financeiros desempenham um papel vital na economia mundial. Cada setor econômico, abrangendo áreas como tecnologia, saúde e energia, contribui de maneira significativa para o crescimento econômico e a estabilidade. O mercado de ações, em particular, é frequentemente visto como um termômetro da saúde econômica de um país, refletindo as expectativas de investidores sobre o futuro e reagindo a mudanças econômicas globais.

Neste cenário, índices econômicos, como o [S&P 500 \(2023\)](#), desenvolvido pela Standard & Poor's, assumem uma importância fundamental. Eles não apenas ajudam investidores e analistas a avaliar o desempenho do mercado, mas também fornecem insights sobre as tendências econômicas por meio da classificação dos setores. O índice S&P 500, que abrange as 500 maiores empresas listadas nas bolsas dos Estados Unidos e representa aproximadamente 80% da capitalização de mercado disponível, é um dos principais indicadores do mercado acionário americano e um barômetro da economia dos EUA.

1.1 Setores Econômicos

Para a classificação desses setores, utiliza-se o Padrão Global de Classificação da Indústria (GICS, na sigla em inglês), uma estrutura de análise setorial criada pela [MSCI \(2023\)](#) e pela [S&P DOW JONES INDICES \(2023\)](#). O GICS proporciona um sistema de classificação hierárquica da indústria em quatro níveis: 11 Setores, 25 Grupos Industriais, 74 Indústrias e 163 Subindústrias. As 500 empresas do índice S&P 500 são agrupadas nesses setores definidos pelo GICS, refletindo uma ampla variedade de atividades econômicas. Esse agrupamento não só facilita a análise do mercado acionário, mas também ajuda a entender as dinâmicas econômicas subjacentes. Os 11 setores principais, conforme definidos pelo GICS, são os seguintes ([S&P GLOBAL, 2023](#)):

- Energia
- Materiais
- Industrial
- Bens de Consumo Discricionário
- Produtos Básicos de Consumo
- Assistência Médica
- Serviços Financeiros
- Tecnologia da Informação
- Serviços de Comunicações
- Serviços de Utilidade Pública
- Imóveis

1.2 Descrição dos setores da S&P

A seguir, uma breve descrição de cada setor, conforme apresentado em [S&P GLOBAL \(2023\)](#).

O setor de **Energia** compreende empresas envolvidas na exploração, produção, refino, comercialização, armazenamento e transporte de petróleo, gás, carvão e combustíveis consumíveis. Abrange também companhias que fornecem equipamentos e serviços para a indústria de petróleo e gás.

O setor de **Materiais** inclui empresas que fabricam produtos químicos, materiais de construção, produtos florestais, vidro, papel e produtos de embalagem relacionados, bem como empresas de metais, minerais e mineração, incluindo produtores de aço.

O setor **Industrial** abrange fabricantes e distribuidores de bens de capital, como a indústria aeroespacial e de defesa, produtos de construção, equipamentos elétricos e máquinas. Inclui também empresas que oferecem serviços de construção e engenharia. Abrange fornecedores de serviços comerciais e profissionais, incluindo impressão, serviços ambientais e de instalações, serviços e materiais de escritório, serviços de segurança e alarmes, serviços de recursos humanos e de emprego, além de serviços de pesquisa e consultoria. Engloba ainda empresas que prestam serviços de transporte.

Empresas do setor de **Bens de Consumo Discricionário** abrangem aquelas que tendem a ser mais sensíveis aos ciclos econômicos. Seu segmento de fabricantes inclui automóveis e componentes, bens domésticos duráveis, produtos de lazer, têxteis e vestuário. O segmento de serviços inclui hotéis, restaurantes e outros estabelecimentos de lazer, além de distribuidores e varejistas de produtos de consumo discricionário.

O setor de **Produtos Básicos de Consumo** engloba empresas cujos negócios são menos sensíveis aos ciclos econômicos. Inclui fabricantes e distribuidores de alimentos, bebidas e tabaco, e produtores de bens domésticos não duráveis e produtos pessoais.

Abrange também distribuidores e varejistas de produtos básicos de consumo, incluindo empresas de varejo de alimentos e medicamentos.

O setor de **Assistência Médica** inclui provedores e serviços de saúde, empresas que fabricam e distribuem equipamentos e suprimentos de assistência médica, bem como empresas de tecnologia de assistência médica. Engloba empresas envolvidas na pesquisa, desenvolvimento, produção e comercialização de produtos farmacêuticos e biotecnológicos.

Empresas no setor de **Serviços Financeiros** contêm empresas envolvidas em atividades bancárias, serviços financeiros, financiamento ao consumidor, mercado de capitais e seguros. Abrange bolsas financeiras, REITs de armazenamento de dados e hipotecários.

O setor de **Tecnologia da Informação** abrange empresas que oferecem serviços de software e tecnologia da informação, fabricantes e distribuidores de hardware e equipamentos de tecnologia, como equipamentos de comunicação, telefones celulares, computadores e periféricos, equipamentos eletrônicos e instrumentos relacionados, além de semicondutores e equipamentos e materiais relacionados.

O setor de **Serviços de Comunicações** inclui empresas que facilitam a comunicação e oferecem conteúdos e informações relacionadas através de vários meios. Engloba empresas de telecomunicações e mídia e entretenimento, incluindo produtores de produtos de jogos interativos e empresas envolvidas na criação ou distribuição de conteúdo e informações através de plataformas próprias.

Empresas no setor de **Serviços de Utilidade Pública** incluem fornecedores de serviços públicos, como eletricidade, gás e água. Abrange produtores independentes de eletricidade, comerciantes de energia e empresas que geram e distribuem eletricidade usando fontes renováveis.

Por fim, o setor de **Imóveis** inclui empresas envolvidas em atividades de desenvolvimento e operação

Cada um desses setores tem características distintas e desempenha um papel único na economia, que podem ter informações relevantes ao tentar entender e prever o que vem pela frente.

Capítulo 2

Conceitos de Séries temporais

2.1 Séries Temporais

Uma série temporal é qualquer conjunto de observações ordenadas no tempo. Os modelos usados para séries temporais são processos estocásticos. (MORETTIN e TOLOI, 2018)

Um processo estocástico, por definição, é uma família de variáveis aleatórias $\mathcal{Z} = \{\mathcal{Z}(t), t \in \mathcal{T}\}$ definidas em um mesmo espaço de probabilidade $(\Omega, \mathcal{A}, \mathcal{P})$. Como $\mathcal{Z}(t)$ é uma variável aleatória definida sobre Ω , $\mathcal{Z}(t)$ pode ser melhor escrita como uma função $\mathcal{Z}(t, \omega), t \in \mathcal{T}, \omega \in \Omega$.

Com $\omega \in \Omega$ fixado, obtém-se uma *série temporal* Z_t , também chamado de *realização* ou *trajetória*. (MORETTIN e TOLOI, 2018).

Na figura 2.1 temos como exemplo de série temporal, a evolução do índice S&P.

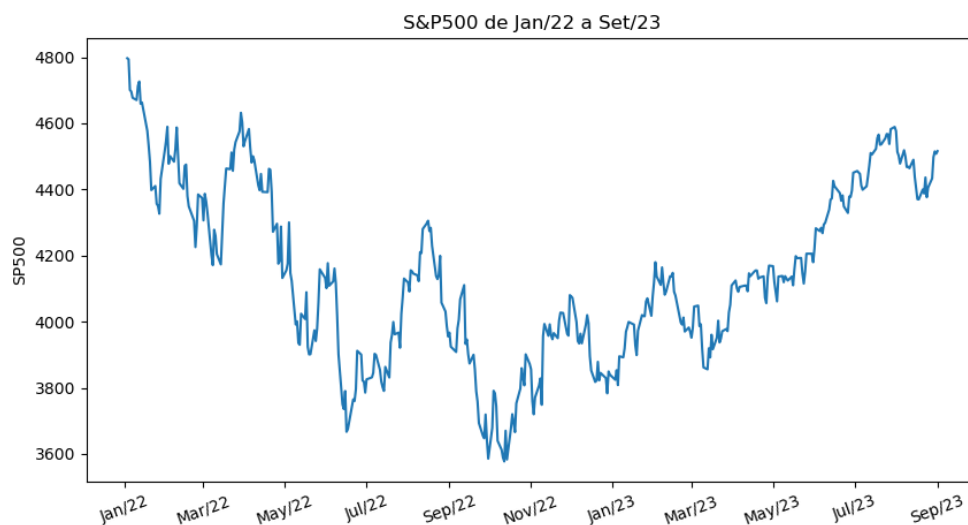


Figura 2.1: Índice S&P500 de Jan/22 a Set/23 .

Um conjunto de séries temporais é exemplificada da figura 2.2, com o histórico dos preços de fechamento das ações da Google, NVidia, Tesla e Mondelez listadas na S&P, de janeiro de 2022 a setembro de 2023.

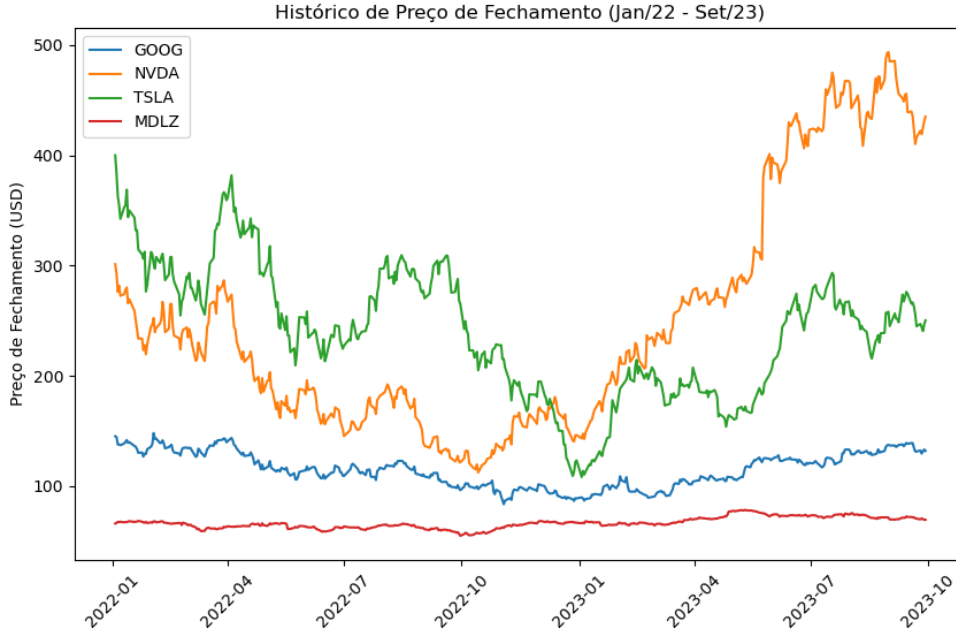


Figura 2.2: evolução do preço de fechamento das ações GOOG, NVDA, TSLA e MDLZ (listadas na S&P) de Jan/22 a Set/23 .

Múltiplas séries temporais, como as do exemplo, podem ser analisadas de forma conjunta; isto é, n séries temporais podem ser consideradas como uma única série temporal multivariada de n componentes. Essa abordagem também é chamada de série temporal vetorial, denotada por:

$$Z_t = (Z_{1t}, Z_{2t}, \dots, Z_{nt}), \quad t \in \mathbb{Z} \quad (2.1)$$

Desta forma, é possível, além de capturar relações da série com si mesma, como ocorre nas séries univariadas, capturar também relações entre diferentes séries. (MORETTIN, 2017, p. 217)

2.2 Estacionariedade

Define-se 2 tipos de processos estacionários: Processo estritamente estacionário (ou fortemente estacionário) e processo fracamente estacionário (MORETTIN, 2017):

Seja o processo estocástico $\{Z(t), t \in \mathcal{T}\}$ e t_1, \dots, t_n quaisquer elementos de \mathcal{T} , as distribuições finito-dimensionais é definido como:

$$F(z_1, \dots, z_n; t_1, \dots, t_n) = \mathbb{P}(Z(t_1) \leq z_1, \dots, Z(t_n) \leq z_n) \quad (2.2)$$

e a função de autocovariância dada por:

$$\gamma(t_1, t_2) = E[Z(t_1)Z(t_2)] - E[Z(t_1)]E[Z(t_2)] \quad (2.3)$$

Processo estritamente estacionário

Um processo estocástico é estritamente estacionário se, sob translação temporal, as distribuições finito dimensionais (2.2) são as mesmas para quaisquer t_1, \dots, t_n e $t \in \mathcal{T}$, isto é:

$$F(z_1, \dots, z_n; t_1, \dots, t_n) = F(z_1, \dots, z_n; t_1 + \tau, \dots, t_n + \tau) \quad (2.4)$$

Isto mostra que as distribuições finito-dimensionais são invariantes sob translações do tempo,

Processo fracamente estacionário

Um processo estocástico é fracamente estacionário (também chamado de processo estacionário de segunda ordem) se, e somente se:

$$E[Z(t)] = \mu(t) = \mu, \forall t \quad (2.5)$$

$$E[(Z(t))^2] < \infty \quad (2.6)$$

$$\gamma(t_1, t_2) = COV(Z(t_1), Z(t_2)) \quad \text{é uma função apenas de } |t_1 - t_2| \quad (2.7)$$

Isto é, a média é constante para todo t , a variância é finita e a covariância depende apenas de t

Esse processo será chamado, por simplicidade, apenas de processo estacionário.

2.3 Ruído Branco

Seja o processo $\varepsilon_t, t \in \mathcal{T}$, este será um ruído branco se o processo for estacionário, isto é, $E[\varepsilon_t] = \mu$, $Var[\varepsilon_t] = \sigma^2$ e, além disso, $COV[\varepsilon_t, \varepsilon_s] = 0, s \neq t$. Particularmente, será preferido usar por convenção $\mu = 0$ (MORETTIN, 2017).

O Ruído Branco é denotado pela sigla RB (ou ainda WN, do inglês White Noise) :

$$\varepsilon_t \sim RB(0, \sigma^2) \quad (2.8)$$

Em particular, se o processo ε_t é i.i.d. (independente e identicamente distribuído) e segue uma distribuição normal, este denomina-se como Ruído Branco Gaussiano, RBG (ou GWN, do inglês Gaussian White Noise), isto é (ZIVOT e WANG, 2006) :

$$\varepsilon_t \sim iid \quad N(0, \sigma^2) \quad \equiv GWN(0, \sigma^2) \quad (2.9)$$

2.4 Operador Lag (defasagem)

Introduzimos o operador Lag L para se obter uma melhor relação entre as diferenças e médias móveis em uma série temporal (KIRCHGÄSSNER *et al.*, 2012). Quando aplicado o operador Lag, todos os valores de uma série temporal y são defasados em 1 período, ou seja:

$$Lz_t = z_{t-1} \quad (2.10)$$

Aplicando L em z_{t-1} obtém-se:

$$Lz_{t-1} = L(Lz_t) = L^2 z_t = z_{t-2} \quad (2.11)$$

E por iteração:

$$L^k z_t = z_{t-k} \quad (2.12)$$

Com $k = 0$, tem-se a identidade $L^0 z_t = y_t$, com $k > 0$ a série é deslocada k períodos para trás e com $k < 0$ a série é deslocada $|k|$ períodos para frente.

De forma mais geral:

$$L^m z_{t-n} = L^m (L^n z_t) = L^{(m+n)} z_t = z_{t-(m+n)} \quad (2.13)$$

Com isso é possível também definir a primeira diferença e por iteração a p -ésima diferença:

$$\Delta z_t = z_t - z_{t-1} = (1 - L)z_t \quad (2.14)$$

$$\Delta_p z_t = z_t - z_{t-p} = (1 - L^p)z_t \quad (2.15)$$

Usando o operador Lag é possível também representar um polinômio de defasagens de grau p :

$$\alpha(L)z_t = (1 - \alpha_1 L - \alpha_2 L^2 - \dots - \alpha_p L^p)z_t = y_t - \alpha_1 z_{t-1} - \alpha_2 z_{t-2} - \dots - \alpha_p z_{t-p} \quad (2.16)$$

Capítulo 3

Modelos Tradicionais de Séries Temporais

3.1 Modelos univariados

3.1.1 Modelo Auto-Regressivo (AR)

AR de ordem 1

Um processo auto-regressivo de ordem 1, $AR(1)$ pode ser escrita como uma equação de diferença estocástica não homogênea: (KIRCHGÄSSNER *et al.*, 2012)

$$z_t = \eta + \alpha z_{t-1} + u_t \quad (3.1)$$

Onde u_t é ruído branco, η o intercepto e α coeficiente do termo auto-regressivo.

Usando o método de substituição, é possível reescrever a equação (3.1) em termos de $t = t_0$:

$$z_t = \alpha^{t-t_0} z_{t_0} + \frac{1 - \alpha^{t-t_0}}{1 - \alpha} \eta + \sum_{j=0}^{t-t_0-1} \alpha^j u_{t-j}, \quad t = t_0 + h \quad (3.2)$$

Tomando $t_0 = 0$ como instante inicial:

$$z_t = \alpha^t z_0 + \frac{1 - \alpha^t}{1 - \alpha} \eta + \sum_{j=0}^{t-1} \alpha^j u_{t-j} \quad (3.3)$$

a equação (3.3) obtida possui um termo constante com η e um processo aleatório puro u_{t-j} dependentes do tempo

Em termos do operador Lag definido em 2.4:

$$(1 - \alpha L)z_t = \eta + u_t \quad (3.4)$$

Com algumas manipulações algébricas e com $|\alpha| < 1$, fazemos o limite com $t \rightarrow -\infty$, a expressão converge para: (KIRCHGÄSSNER *et al.*, 2012):

$$z_t = \frac{\eta}{1 - \alpha} + \sum_{j=0}^{\infty} \alpha^j u_{t-j} \quad (3.5)$$

AR de ordem p

Por vezes é preferível usar AR de ordem p a fim de levar em consideração mais de um instante anterior

$$z_t = \eta + \alpha_1 z_{t-1} + \alpha_2 z_{t-2} + \dots + \alpha_p z_{t-p} + u_t \quad (3.6)$$

$$z_t = \eta + \sum_{i=1}^p \alpha_i z_{t-i} + u_t \quad (3.7)$$

Em termos do operador Lag:

$$(1 - \alpha_1 L - \alpha_2 L^2 - \dots - \alpha_p L^p)z_t = \eta + u_t \quad (3.8)$$

3.1.2 Modelo de Média Móvel (MA)

MA de ordem 1

Um processo de Média Móvel de ordem 1 é dado por: (KIRCHGÄSSNER *et al.*, 2012)

$$z_t = \mu + u_t + \beta u_{t-1} \quad (3.9)$$

ou ainda:

$$(1 - \beta L)u_t = z_t - \mu \quad (3.10)$$

MA de ordem q

Um processo de Média Móvel de ordem q é dado por:

$$z_t = \mu - \beta_1 u_{t-1} - \beta_2 u_{t-2} - \dots - \beta_q u_{t-q} + u_t \quad (3.11)$$

$$z_t = \mu - \sum_{j=1}^q \beta_j u_{t-j} + u_t \quad (3.12)$$

Em termos do operador Lag:

$$(1 - \beta_1 L - \beta_2 L^2 - \dots - \beta_q L^q)u_t = z_t - u_t \quad (3.13)$$

3.1.3 Modelo Auto-Regressivo com Média Móvel (ARMA)

o modelo de ordem 1 ARMA(1,1) é dado por:

$$z_t = \eta + \alpha z_{t-1} + u_t - \beta u_{t-1} \quad (3.14)$$

Escrevendo em termos de L:

$$(1 - \alpha L)z_t = \eta + (1 - \beta L)u_t \quad (3.15)$$

Modelo geral de ordem p e q ARMA(p, q), dado por:

$$z_t = \eta + \alpha_1 z_{t-1} + \alpha_2 z_{t-2} + \dots + \alpha_p z_{t-p} - \beta_1 u_{t-1} - \beta_2 u_{t-2} - \dots - \beta_q u_{t-q} + u_t \quad (3.16)$$

$$z_t = \eta + \sum_{i=1}^p \alpha_i z_{t-i} - \sum_{j=1}^q \beta_j u_{t-j} + u_t \quad (3.17)$$

Reescrevendo com L:

$$(1 - \alpha_1 L - \alpha_2 L^2 - \dots - \alpha_p L^p)z_t = \eta + (1 - \beta_1 L - \beta_2 L^2 - \dots - \beta_q L^q)u_t \quad (3.18)$$

3.2 Modelos Multivariados

3.2.1 Modelo Vetorial Auto-Regressivo (VAR)

Seja a série temporal multivariada $Z_t = (z_{1t}, z_{2t}, \dots, z_{nt})'$, um vetor $(n \times 1)$. o modelo básico VAR(p), que assume um processo Estacionário, é descrito como: (MORETTIN, 2017, p. 224)

$$Z_t = \delta + \Phi_1 Z_{t-1} + \Phi_2 Z_{t-2} + \dots + \Phi_p Z_{t-p} + U_t = \delta + \sum_{i=1}^p \Phi_i Z_{t-i} + U_t \quad (3.19)$$

Com Φ_i uma matriz de coeficientes ($n \times n$), U_t um ruído branco e δ um vetor de intercepto.

$$\begin{pmatrix} z_{1t} \\ z_{2t} \\ \vdots \\ z_{nt} \end{pmatrix} = \begin{pmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \end{pmatrix} + \begin{pmatrix} \phi_{11}^1 & \phi_{12}^1 & \cdots & \phi_{1n}^1 \\ \phi_{21}^1 & \phi_{22}^1 & \cdots & \phi_{2n}^1 \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1}^1 & \phi_{n2}^1 & \cdots & \phi_{nn}^1 \end{pmatrix} \begin{pmatrix} z_{1t-1} \\ z_{2t-1} \\ \vdots \\ z_{nt-1} \end{pmatrix} + \dots + \begin{pmatrix} \phi_{11}^n & \phi_{12}^n & \cdots & \phi_{1n}^n \\ \phi_{21}^n & \phi_{22}^n & \cdots & \phi_{2n}^n \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{n1}^n & \phi_{n2}^n & \cdots & \phi_{nn}^n \end{pmatrix} \begin{pmatrix} z_{1t-p} \\ z_{2t-p} \\ \vdots \\ z_{nt-p} \end{pmatrix} + \begin{pmatrix} u_{1t} \\ u_{2t} \\ \vdots \\ u_{nt} \end{pmatrix} \quad (3.20)$$

Por [ZIVOT e WANG, 2006](#), um modelo VAR(p) mais geral poderia ser descrito por:

$$Z_t = \delta + \Phi_1 Z_{t-1} + \Phi_2 Z_{t-2} + \dots + \Phi_p Z_{t-p} + \Xi D_t + \Lambda G_t + U_t \quad (3.21)$$

Sendo D_t um vetor ($n \times 1$) de componentes determinísticos, Λ_t um vetor ($n \times 1$) de variáveis exógenas e Φ e G matrizes ($n \times n$) de coeficientes. Com uso matricial do operador Lag, podemos simplificar (3.20) como:

$$A(L) = I_n - \Phi_1 L - \Phi_2 L^2 - \dots - \Phi_p L^p \quad (3.22)$$

$$A(L)Z_t = \delta + U_t \quad (3.23)$$

com as propriedades:

$$E[U_t] = 0 \quad (3.24)$$

$$E[U_t U_t'] = \Sigma_u \quad (3.25)$$

$$E[U_t U_s'] = 0, \quad t \neq s \quad (3.26)$$

Os resíduos U podem estar correlacionados contemporaneamente o que indica relações instantâneas entre as variáveis endógenas da expressão (3.20). Este sistema é estável se e somente se todas as raízes da equação característica do polinômio defasado estiverem fora do círculo unitário, ou seja, ([KIRCHGÄSSNER et al., 2012](#))

$$\det(A(L)) \neq 0 \quad \text{para} \quad |z| < 1 \quad (3.27)$$

De (3.23), chega-se após algumas manipulações em sua representação VMA:

$$Z_t = \mu + B(L)U_t, \quad (3.28)$$

Com

$$B(L) = A^{-1}(L)\delta, \quad \mu = A^{-1}(1)\delta = B(1)\delta \quad (3.29)$$

A matriz de autocovariância é definida como:

$$\Gamma_Y(\tau) = E[(Z_t - \mu)(Z_{t-\tau} - \mu)'] \quad (3.30)$$

Sem perda de generalidade, tomando $\delta = 0$ e, conseqüentemente, $\mu = 0$, tem-se devido a (3.20):

$$E[Z_t Z_{t-\tau}'] = \Phi_1 E[Z_{t-1} Z_{t-1-\tau}'] + \Phi_2 E[Z_{t-2} Z_{t-2-\tau}'] + \dots + \Phi_p E[Z_{t-p} Z_{t-p-\tau}'] + E[U_t Z_{t-\tau}'] \quad (3.31)$$

O que leva a determinar a matriz de covariância como:

$$\Gamma_Y(\tau) = \Phi_1 \Gamma_Y(\tau - 1) + \Phi_2 \Gamma_Y(\tau - 2) + \dots + \Phi_p \Gamma_Y(\tau - p) + \Sigma_U \quad (3.32)$$

$$\Gamma_Y(0) = \Phi_1 \Gamma_Y(-1) + \Phi_2 \Gamma_Y(-2) + \dots + \Phi_p \Gamma_Y(-p) + \Sigma_U \quad (3.33)$$

e, tendo que $\gamma_{ij}(\tau) = \gamma_{ji}(-\tau)$, com $\gamma_{ij}(\tau)$ sendo os elementos de $\Gamma_Y(\tau)$, tem-se que $\Gamma_Y(\tau) = \Gamma_Y(-\tau)'$. (KIRCHGÄSSNER *et al.*, 2012) Com isso tem-se também:

$$\Gamma_Y(0) = \Phi_1 \Gamma_Y(1)' + \Phi_2 \Gamma_Y(2)' + \dots + \Phi_p \Gamma_Y(p)' + \Sigma_U \quad (3.34)$$

Daqui, segue que o coeficiente de correlação individual é expresso por:

$$\rho_{ij}(\tau) = \frac{\gamma_{ij}(\tau)}{\sqrt{\gamma_{ii}(0) \cdot \gamma_{jj}(0)}} \quad (3.35)$$

Finalmente, tem-se a matriz de autocorrelação:

$$r_Z(\tau) = D^{-1} \Gamma_Y(\tau) D^{-1} \quad (3.36)$$

Com:

$$D^{-1} = \begin{pmatrix} \frac{1}{\sqrt{\gamma_{11}(0)}} & 0 & \dots & 0 \\ 0 & \frac{1}{\sqrt{\gamma_{22}(0)}} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \frac{1}{\sqrt{\gamma_{kk}(0)}} \end{pmatrix} \quad (3.37)$$

Estimação de VAR(p)

Define-se a partir da (3.19), considerando uma sequência de tamanho T :

$$\begin{aligned} \Psi &:= (Z_1, Z_2, \dots, Z_T), \quad (n \times T) \\ \mathfrak{B} &:= (\delta, \Phi_1, \Phi_2, \dots, \Phi_p), \quad (n \times (np + 1)) \end{aligned}$$

$$\mathfrak{X}_t := \begin{bmatrix} 1 \\ Z_t \\ \vdots \\ Z_{t-p+1} \end{bmatrix}, \quad ((np + 1) \times 1) \quad (3.38)$$

$$\begin{aligned} \mathfrak{X} &:= (\mathfrak{X}_0, \dots, \mathfrak{X}_{T-1}), \quad ((np + 1) \times T) \\ \mathcal{U} &:= (U_1, \dots, U_T) \quad (n \times T) \\ \psi &:= \text{vec}(\Psi) \quad (nT \times 1) \\ \mathcal{B} &:= \text{vec}(\mathfrak{B}) \quad ((n^2 p + n) \times 1) \\ \mathcal{B}^* &:= \text{vec}(\mathfrak{B}') \quad ((n^2 p + n) \times 1) \\ \nu &:= \text{vec}(\mathcal{U}), \quad (nT \times 1) \end{aligned}$$

Com essas transformações, o modelo $VAR(p)$ de (3.19) se reduz a um problema de regressão múltipla (HELMUT, 2005, p. 70), isto é:

$$\Psi = \mathfrak{B}\mathfrak{X} + \mathcal{U} \quad (3.39)$$

o qual pode ser escrita com as matrizes transformadas anteriormente, utilizando vetorização e produto de Kronecker :

$$vec(\Psi) = vec(\mathfrak{B}\mathfrak{X}) + vec(\mathcal{U}) = (\mathfrak{X}' \otimes I_n)vec(\mathfrak{B}) + vec(\mathcal{U}) \quad (3.40)$$

$$\psi = (\mathfrak{X}' \otimes I_n)\mathcal{B} + v \quad (3.41)$$

A matriz de covariância é dada por:

$$\Sigma_{\mathcal{U}} = I_n \otimes \Sigma_u \quad (3.42)$$

Com isso, é possível estimar \mathcal{B} usando estimação por Mínimos Quadrados Ordinários (MQO) ou ainda Mínimos Quadrados Generalizados (MQG). Com o estimador por MQO, chega-se a (HELMUT, 2005, p. 72):

$$\hat{\mathfrak{B}} = \Psi\mathfrak{X}'(\mathfrak{X}\mathfrak{X}')^{-1} \quad (3.43)$$

Seleção da ordem p do modelo $VAR(p)$

Existem várias abordagens para determinar a ordem apropriada para um modelo autorregressivo. Uma maneira é realizar testes de hipóteses, como o teste t, para verificar a significância estatística de cada coeficiente autorregressivo Φ_i , testando a hipótese nula de que $\Phi_i = 0$ para as defasagens i . Outra abordagem comum é utilizar critérios de informação que buscam equilibrar a qualidade do ajuste do modelo com a penalidade por adicionar mais parâmetros, visando minimizar a perda de informação ou o erro de previsão fora da amostra.

Para este trabalho será considerado os critérios de informação para a seleção da ordem do modelo, serão usados 3 destes critérios, como usados por HELMUT, 2005: Critério de Informação de Akaike (AIC), Critério de Informação de Schwarz (SC) - também conhecido como Critério de Informação Bayesiano (BIC) - e Critério de informação Hannan-Quinn (HQ).

Seja $\tilde{\Sigma}_u$ a matriz de covariância dos resíduos quando se obtém o estimador por verossimilhança para o modelo $VAR(p)$ (HELMUT, 2005, p. 140):

$$\tilde{\Sigma}_u = \frac{1}{T}(\Psi - \tilde{\mathcal{B}}\mathfrak{X})(\Psi - \tilde{\mathcal{B}}\mathfrak{X})' \quad (3.44)$$

Onde $\tilde{\mathfrak{B}}$ é a matriz de coeficientes correspondente a $\tilde{\mathcal{B}}$ que por sua vez é o estimador por MQO ou MV (Máxima Verossimilhança), isto é (HELMUT, 2005, p. 139):

$$\tilde{\mathcal{B}} = vec(\tilde{\mathfrak{B}}) \quad (3.45)$$

$$\tilde{\mathcal{B}} = ((\mathfrak{X}\mathfrak{X}^T)^{-1}\mathfrak{X} \otimes I_n)\psi \quad (3.46)$$

Os critérios de informação citados são definidos a seguir::

$$\begin{aligned} AIC(m) &= \ln|\tilde{\Sigma}_u(m)| + \frac{2mn}{T} \\ BIC(m) &= \ln|\tilde{\Sigma}_u(m)| + \frac{\ln T}{T} mn^2 \\ HQ(m) &= \ln|\tilde{\Sigma}_u(m)| + \frac{2\ln\ln T}{T} mn^2 \end{aligned} \quad (3.47)$$

onde $|\tilde{\Sigma}_u(m)|$ é o determinante de $\tilde{\Sigma}_u(m)$. A estimativa da ordem p , $\hat{p}(\cdot)$, é feita minimizando o valor do critério (HELMUT, 2005, p. 150). Os critérios HQ e BIC determinam de forma consistente a ordem (finita) do Lag máximo, enquanto AIC tende a superestimá-lo (KIRCHGÄSSNER *et al.*, 2012, p. 135). Além disso, HELMUT (2005, p. 151) mostra empiricamente a seguinte relação para amostras pequenas :

$$\hat{p}(BIC) \leq \hat{p}(HQ) \leq \hat{p}(AIC), \quad T \geq 16 \quad (3.48)$$

Entretanto, por vezes o processo que está sendo analisado necessitaria de um modelo VAR de ordem p infinita, e está sendo usado uma amostra de tamanho finito e, conseqüentemente, um modelo de ordem finita para estimá-lo, nesses casos, critérios de informação como o AIC podem dar resultados melhores em termos de precisão da previsão quando se tem tamanho moderado da amostra.

Com isso, é uma boa estratégia comparar as estimativas dadas por esses diferentes critérios e possivelmente realizar análises com diferentes ordens de VAR (HELMUT, 2005, p. 157).

3.2.2 Modelo Vetorial Auto-Regressivo de Médias Móveis (VARMA)

Seja a série temporal multivariada $Z_t = (z_{1t}, z_{2t}, \dots, z_{nt})'$, um vetor $(n \times 1)$. o modelo básico VARMA(p,q), é descrito como (HELMUT, 2005, p. 423):

$$\begin{aligned} Z_t &= \delta + \Phi_1 Z_{t-1} + \Phi_2 Z_{t-2} + \dots + \Phi_p Z_{t-p} + U_t - B_1 U_{t-1} - B_2 U_{t-2} - \dots - B_q U_{t-q} \\ Z_t &= \delta + U_t + \sum_{i=1}^p \Phi_i Z_{t-i} - \sum_{i=1}^q M_i U_{t-i} \end{aligned} \quad (3.49)$$

Novamente, $U_t \sim RB(0, \Sigma_u)$ um vetor $(n \times 1)$ com matriz de covariância Σ_u não-singular e com a série assumida estacionária como no modelo VAR citado anteriormente.

Com uso da notação do operador Lag:

$$A(L)Z_t = \delta + M(L)U_t \quad (3.50)$$

$$\begin{aligned} A(L) &:= I_n - \Phi_1 L - \Phi_2 L^2 - \dots - \Phi_p L^p \\ M(L) &:= I_n + M_1 L + M_2 L^2 + \dots + M_q L^q \end{aligned} \quad (3.51)$$

Note que se $q = 0$, obtém-se um modelo VAR puro e com $p = 0$ um modelo VMA puro

(MORETTIN, 2017, p. 236), pois $VAR(p)$ (3.19):

$$Z_t = \delta + \Phi_1 Z_{t-1} + \Phi_2 Z_{t-2} + \dots + \Phi_p Z_{t-p} + U_t = \delta + \sum_{i=1}^p \Phi_i Z_{t-i} + U_t$$

e $VMA(q)$:

$$Z_t = \delta + U_t - M_1 U_{t-1} - \dots - M_q U_{t-q} = \delta + U_t - \sum_{i=1}^q M_i U_{t-i} \quad (3.52)$$

com $U_t \sim RB(0, \Sigma_u)$

O modelo VARMA(p,q) com $p \neq 0$ e $q \neq 0$, com algumas manipulações pode ser representado pelos modelos VAR e VMA de ordem infinita (HELMUT, 2005, p. 423), o que não será abordado nesse trabalho.

Estimação de VARMA(p,q)

Os modelos VARMA tem uma dificuldade computacional maior no processo de estimação e problema de identificabilidade, pois sem condições mais restritivas, a matriz de covariâncias não determina univocamente Φ_i , B_j e Σ_u . Com esta não-identificabilidade a superfície não linear de verossimilhança não terá um único máximo (MORETTIN, 2017, p. 240),

Embora o VARMA seja mais complicado de estimar e tenha um custo computacional maior em comparação ao VAR de ordem finita, existem razões que justificam o seu uso. Primeiramente, o VARMA pode ser mais parcimonioso em termos do número de parâmetros; ou seja, geralmente, é necessário estimar menos parâmetros do que no VAR. Além disso, há casos em que a teoria subjacente pressupõe uma parte significativa de médias móveis nas séries, o que ajuda a derivar resultados mais precisos. Adicionalmente, em algumas situações, ele pode resultar em previsões mais eficientes, compensando assim o aumento da complexidade e do custo computacional. (HELMUT, 2005, p. 512).

Uma das formas de estimação é pelo método de Máxima Verossimilhança, que consiste em maximizar a função de verossimilhança, ou equivalentemente, a log-verossimilhança.

A função de log-verossimilhança aproximada para o modelo VARMA(p,q) é dado a seguir:

$$\ln \mathcal{L}(\mu, \gamma, \Sigma_u) = -\frac{T}{2} |\Sigma_u| - \frac{1}{2} \sum_{t=0}^T u_t(\mu, \gamma)' \Sigma_u^{-1} u_t(\mu, \gamma) \quad (3.53)$$

$$u_t(\mu, \gamma) = (z_t - \mu) - \sum_{i=1}^{t-1} \Pi_i(\gamma)(z_{t-i} - \mu) \quad (3.54)$$

Onde γ é o vetor de parâmetros associados e Π a matriz de coeficientes deste vetor

Para a escolha da ordem do modelo VARMA(p,q), segue-se a mesma abordagem sugerida no modelo VAR(p) isto é, executa para vários valores de p e q e então escolhe o que minimizar os critérios de informação AIC, BIC ou HQ.

Capítulo 4

Modelos com Aprendizado de Máquina

4.1 Inteligência Artificial, Aprendizado de Máquina e Aprendizado profundo

Nos últimos anos, a inteligência artificial (IA) tem sido objeto de intenso exagero da mídia. Aprendizado de máquina, aprendizado profundo e IA são temas recorrentes em inúmeros artigos, geram tanto entusiasmo quanto apreensão. Artigos são repletos de visões de futuros tecnológicos, onde chatbots inteligentes, carros que dirigem sozinhos e assistentes digitais não são apenas possíveis, mas inevitáveis.

As projeções alternam entre um otimismo tecnológico, onde a IA resolve nossos maiores desafios, e uma distopia onde a automação suplanta o trabalho humano. Contudo é necessário definir de forma clara o que são cada coisa. A figura 4.1 ilustra como tais conceitos se conectam.

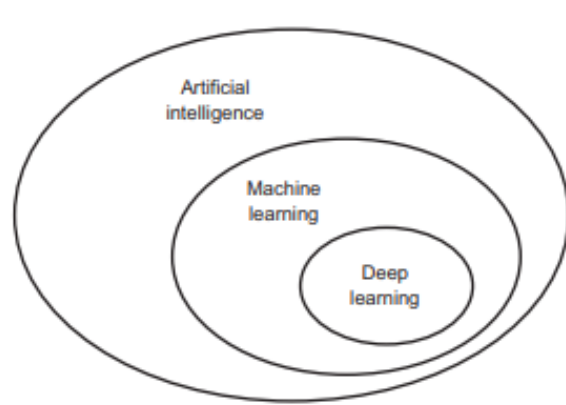


Figura 4.1: Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo. Fonte: *CHOLLET, 2018*

4.1.1 Inteligência Artificial

Uma definição concisa de inteligência artificial poderia ser o esforço para automatizar atividades intelectuais normalmente desempenhadas por humanos (CHOLLET, 2018, p. 4).

Inteligência Artificial (IA) é um domínio abrangente que incorpora o Aprendizado de Máquina (Machine Learning - ML) e o Aprendizado Profundo (Deep Learning), além de englobar uma variedade de outras abordagens que não se baseiam necessariamente em aprendizado. As primeiras implementações de IA eram caracterizadas por conjuntos de regras lógicas predefinidas, formuladas por programadores, que guiavam a tomada de decisão ou a execução de tarefas sem a capacidade de aprendizado. Estas regras eram projetadas para abarcar todas as eventualidades que o sistema pudesse encontrar.

Tomemos, por exemplo, um sistema especialista projetado para diagnóstico médico. Tal sistema poderia estar programado com regras do tipo: "se a temperatura do paciente exceder 38°C e ele apresentar dor de garganta, recomende um teste para infecção bacteriana". De maneira similar, um chatbot de serviço ao cliente pode ser programado para responder a "esqueci minha senha" com uma instrução pré-definida: "Clique no link 'Esqueceu a senha' e siga as instruções para redefini-la", operando sem a necessidade de aprender ou se adaptar a partir de interações anteriores.

Por um período, predominou a crença de que a inteligência artificial ao nível humano poderia ser alcançada através da compilação de um extenso e abrangente conjunto de regras lógicas explícitas.

Todavia, essa abordagem, conhecida como IA simbólica, embora eficaz para resolver problemas de lógica bem delimitados, como o jogo de xadrez, mostrou-se limitada diante de desafios mais nebulosos e complexos, como a classificação de imagens, o reconhecimento de fala e a tradução de línguas. Em resposta a essas limitações, emergiu uma nova metodologia: o Aprendizado de Máquina.

4.1.2 Aprendizado de Máquina

O aprendizado de máquina subverte o paradigma da inteligência artificial simbólica, a qual se restringe às instruções explicitamente definidas por programadores. Em contraste, o aprendizado de máquina capacita os computadores a identificar padrões e tomar decisões a partir dos próprios dados, sem depender de regras pré-determinadas.

Na IA clássica, os programadores precisam entender minuciosamente o problema para inserir manualmente uma combinação de regras e dados que orientam o sistema a gerar respostas apropriadas. Por exemplo, um sistema de diagnóstico médico baseado em regras pode ser programado para recomendar um teste de diabetes se os níveis de glicose do paciente estiverem elevados.

Em contrapartida, no aprendizado de máquina, o sistema é alimentado com um conjunto de dados que inclui tanto as características de interesse (como níveis de glicose) quanto os resultados (diagnóstico de diabetes ou não). O algoritmo de aprendizado de máquina analisa esses dados para descobrir padrões que associam certas características aos resultados. Uma vez que um modelo é treinado com esses dados, ele pode avaliar novas informações e

prever resultados com base nos padrões que aprendeu, mesmo que seja a primeira vez que encontra uma combinação particular de características.

O aprendizado de máquina se utiliza da estatística para identificar tendências nos dados e da otimização para ajustar modelos preditivos aos padrões encontrados. Isso é feito através de um processo chamado treinamento, durante o qual o modelo ajusta iterativamente seus parâmetros internos para minimizar a diferença entre as previsões e os resultados reais conhecidos. Por exemplo, um sistema de reconhecimento de voz é exposto a milhares de horas de fala humana e seus textos transcritos correspondentes, permitindo que ele aprenda a traduzir ondas sonoras em palavras escritas com grande precisão.

A capacidade de aprender com os dados confere ao aprendizado de máquina uma flexibilidade imensa. Ele é aplicado desde a filtragem de e-mails, distinguindo spam de mensagens legítimas, até a direção autônoma de veículos, onde o sistema deve interpretar continuamente os dados sensoriais para tomar decisões de navegação em tempo real.

Aprendizado Supervisionado

No âmbito do aprendizado supervisionado, o algoritmo é meticulosamente treinado utilizando um conjunto de dados rotulados. Isto é, tanto as entradas quanto as saídas desejadas são fornecidas ao sistema. Um exemplo emblemático dessa abordagem é o reconhecimento de dígitos escritos à mão, onde o sistema é treinado usando imagens de dígitos como entradas e os rótulos correspondentes que indicam o número representado por cada imagem como saídas. Tal treinamento capacita o sistema a identificar dígitos em imagens inéditas, nunca antes vistas.

O aprendizado supervisionado é descrito como um sistema em que são fornecidas variáveis de entrada, X (também conhecidas como preditores ou vetores de características), e a saída, Y (variável resposta) (JAMES *et al.*, 2023, p. 17):

$$Y = f(X) + \epsilon \quad (4.1)$$

O objetivo é estimar a função $f(\cdot)$, resultando em:

$$\hat{Y} = \hat{f}(X) \quad (4.2)$$

Onde \hat{f} representa a estimativa de f , e \hat{Y} é o valor predito de Y .

No entanto, é importante destacar que os modelos de aprendizado supervisionado tradicionalmente não são projetados para processar dados sequenciais, como é o caso das séries temporais. Para abordar essa limitação, uma reestruturação dos dados será realizada na seção subsequente, permitindo a aplicação eficaz desses modelos a problemas de séries temporais.

Aprendizado Não Supervisionado

O aprendizado não supervisionado representa uma categoria distinta de sistemas de aprendizado de máquina, caracterizada pela ausência de rótulos nos dados de entrada.

Neste cenário, os dados fornecidos não vêm pré-classificados, cabendo ao algoritmo de aprendizado não supervisionado a tarefa de organizar os dados e elucidar sua estrutura subjacente.

Diferentes métodos de aprendizado não supervisionado, tais como análise de clusters, detecção de anomalias e análise de componentes principais, são empregados para explorar e descobrir padrões nos dados sem orientação explícita. Essas técnicas são descritas com mais detalhes por LAZZERI (2020, p. 15), evidenciando a versatilidade e o poder do aprendizado não supervisionado para revelar insights ocultos em conjuntos de dados complexos.

4.1.3 Aprendizado Profundo

O aprendizado profundo é uma especialização dentro do campo do aprendizado de máquina que enfatiza a criação de representações de dados em camadas sucessivas de abstração e complexidade crescentes.

A terminologia "profundo" no contexto do aprendizado profundo não alude a uma compreensão profunda no sentido filosófico, mas sim à arquitetura das redes neurais que compõem esses modelos, que são construídas em múltiplas camadas de processamento (CHOLLET, 2018, p. 8). A quantidade de camadas empilhadas em um modelo, e, por consequência, a sua "profundidade", é um fator determinante da capacidade do modelo em representar e processar dados complexos.

4.2 Redes Neurais

As Redes Neurais Artificiais, ou simplesmente Redes Neurais, constituem a espinha dorsal tanto do aprendizado de máquina quanto do aprendizado profundo, variando conforme a profundidade de suas camadas. Estes modelos são distintos de outras metodologias de aprendizado de máquina devido à sua estrutura única, que busca emular o modo como os neurônios biológicos do cérebro humano interagem entre si.

Apesar de alguns dos princípios fundamentais do aprendizado profundo serem inspirados na nossa compreensão do funcionamento cerebral, é crucial reconhecer que os modelos de aprendizado profundo não pretendem ser réplicas do cérebro humano. De fato, não existe evidência de que o cérebro humano opere baseado nos mesmos mecanismos presentes nos modelos contemporâneos de aprendizado profundo (CHOLLET, 2018, p. 8).

4.2.1 Redes Neurais de camada única

Redes neurais simples usam uma rede neural feed-forward para modelar uma resposta quantitativa utilizando p preditores, isto é, tomam um vetor de entrada de p componentes $X = (X_1, \dots, X_p)$ e constroem uma função não linear $f(X)$ para prever a resposta Y , como exemplificado na figura 4.2

Onde X_1, X_2, X_3 e X_4 representam a camada de entradas, A_1, A_2, A_3, A_4 e A_5 a camada oculta única (não observável) de ativações não-lineares e $f(X) = Y$ representa a camada

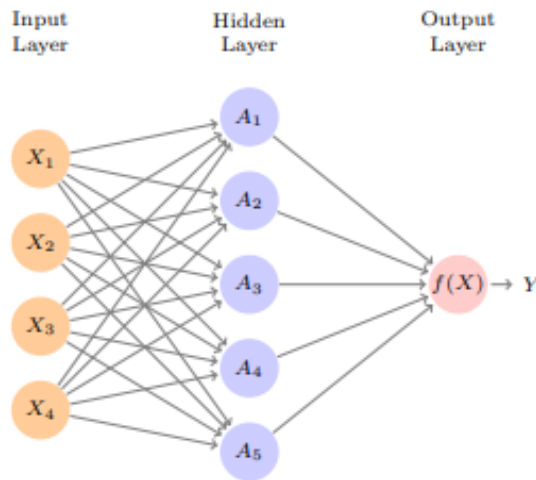


Figura 4.2: Exemplo de rede neural com 4 componentes na entrada X , 1 na saída $f(X) = Y$ e 5 ativações (não observáveis) na camada oculta única. Fonte: JAMES *et al.*, 2023

de saída. As setas indicam que cada uma das entradas se conecta a cada uma das ativações ocultas.

Uma rede neural tem a forma (JAMES *et al.*, 2023, p. 400)

$$f(X) = \beta_0 + \sum_{k=1}^K \beta_k h_k(X) = \beta_0 + \sum_{k=1}^K \beta_k g \left(u_{k0} + \sum_{j=1}^p u_{kj} X_j \right). \quad (4.3)$$

Com K sendo o número de ativações na camada oculta, β_k os pesos associados às ativações da camada oculta, β_0 o termo de viés da camada de saída para dar maior flexibilidade à saída, u_{kj} os pesos associados às entradas da rede neural para a k -ésima unidade da camada oculta e u_{k0} é o termo de viés para a k -ésima unidade da camada oculta. Esses últimos 4 parâmetros, β_k , β_0 , u_{kj} e u_{k0} serão aprendidos no processo de treinamento.

Além disso, a construção é feita em duas etapas. Primeiro, as K ativações A_k , $k = 1, \dots, K$, na camada oculta são computadas como funções das características de entrada X_1, \dots, X_p ,

$$A_k = h_k(X) = g \left(u_{k0} + \sum_{j=1}^p u_{kj} X_j \right). \quad (4.4)$$

Onde $g(\cdot)$ é uma função de ativação não linear que é especificada antecipadamente. Estas K ativações da camada oculta alimentam a camada de saída, resultando em:

$$f(X) = \beta_0 + \sum_{k=1}^K \beta_k A_k \quad (4.5)$$

Uma das funções de ativação mais utilizada no começo do desenvolvimento de redes neurais é a sigmoide, que é a mesma função usada em regressão logística para converter uma entrada linear em uma probabilidade entre zero e um, dada por:

$$g(z) = \frac{1}{1 + e^{-z}} \quad (4.6)$$

Por outro lado, no uso moderno de redes neurais, a função que se tornou a preferida é a ReLU (Rectified Linear Unit), uma função de ativação que pode ser computada e armazenada de maneira mais eficiente do que uma ativação sigmoide (JAMES *et al.*, 2023, p. 401):

$$g(z) = \max(0, z) \quad (4.7)$$

A forma de ambas as funções de ativação são exibidas na figura 4.3

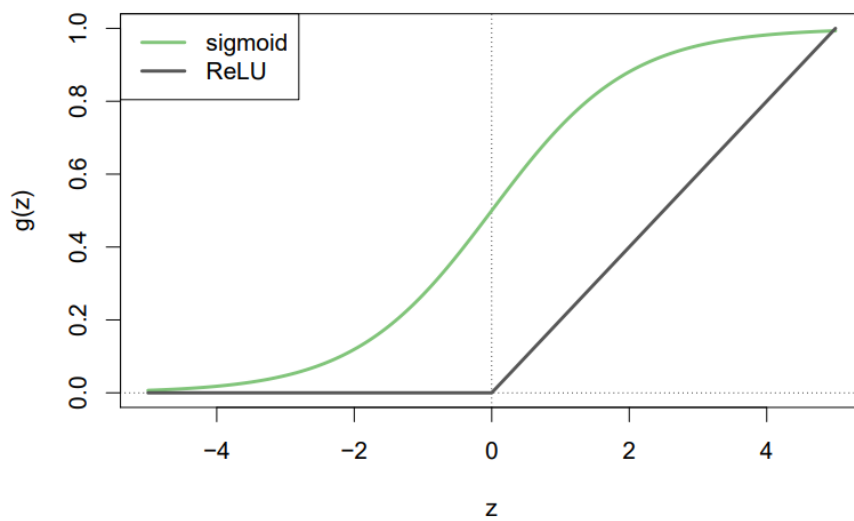


Figura 4.3: Funções de ativação Sigmoide e ReLU. ReLU com escala reduzida em um fator de cinco para facilitar a comparação de suas formas. Fonte: JAMES *et al.*, 2023

Treinar uma rede neural requer estimar os parâmetros desconhecidos da equação 4.3. Normalmente é usada a perda de erro quadrático, de modo que os parâmetros são escolhidos para minimizar:

$$\sum_{i=1}^n (y_i - f(x_i))^2 \quad (4.8)$$

4.2.2 Redes Neurais de Múltiplas Camadas (MLP)

As arquiteturas modernas de redes neurais frequentemente incorporam múltiplas camadas ocultas, cada uma podendo conter um número substancial de unidades (neurônios). Embora, teoricamente, uma única camada oculta com uma quantidade adequada de unidades possa aproximar uma vasta gama de funções complexas, descobriu-se que estruturas com múltiplas camadas, mesmo que de tamanho mais modesto, facilitam significativamente o processo de aprendizado. A presença de camadas adicionais permite que a rede desenvolva uma compreensão hierárquica dos dados de entrada, onde conceitos

simples nas camadas inferiores se combinam para formar representações mais complexas em camadas superiores (JAMES *et al.*, 2023, p. 402). Esta abordagem hierárquica é visualizada na figura 4.4.

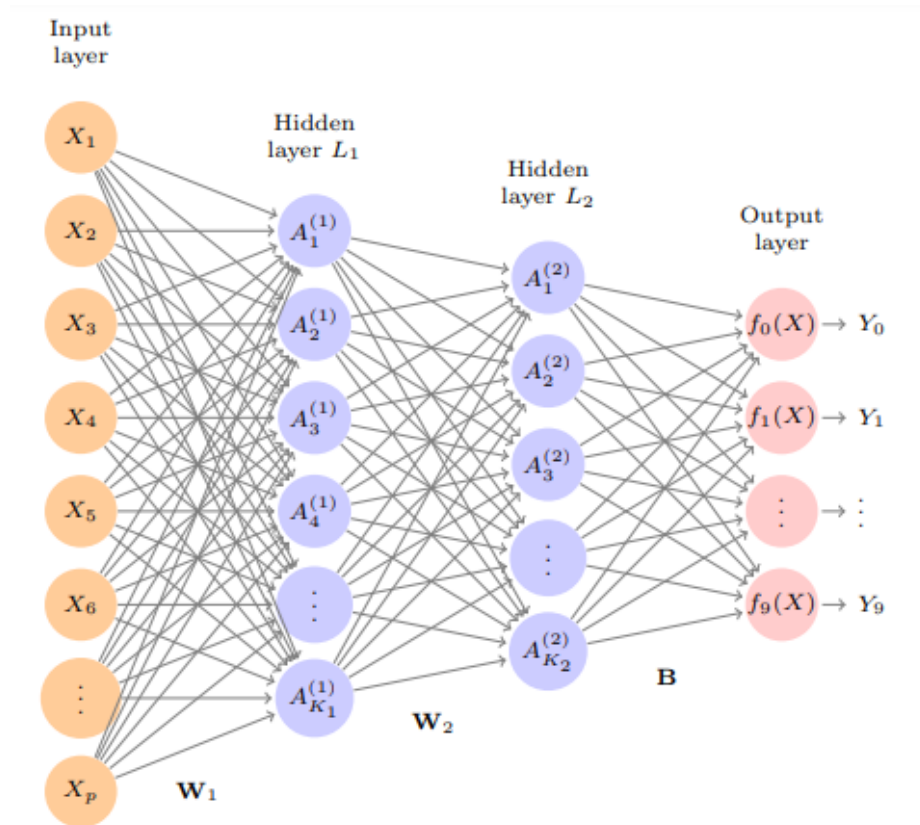


Figura 4.4: Exemplo de uma rede neural multi camadas com múltiplas entradas, 2 camadas ocultas e 10 saídas. Fonte: JAMES *et al.*, 2023

O processo de treinamento em redes de múltiplas camadas é conceitualmente similar ao de redes com uma única camada oculta, embora a complexidade aumente devido ao maior número de parâmetros que necessitam ser otimizados. Este aumento no número de parâmetros melhora a capacidade da rede de aprender padrões complexos nos dados, mas também exige técnicas mais sofisticadas de otimização e regularização para evitar o sobreajuste.

Capítulo 5

Extração e tratamento de dados

No processo de modelagem estatística e de aprendizado de máquina, diversas etapas críticas devem ser realizadas antes da aplicação efetiva dos modelos. A primeira etapa envolve a extração e a preparação do conjunto de dados. Esta fase é crucial para assegurar que os dados estejam alinhados com os requisitos específicos de cada tipo de modelo.

Modelos estatísticos como o VAR e VARMA, citados nas seções 3.2.1 e 3.2.2 pressupõem que as séries temporais sejam estacionárias. Já no contexto de aprendizado de máquina, citado no capítulo 4, frequentemente é necessário que os dados estejam estruturados em um formato de aprendizado supervisionado, o que implica na definição clara de variáveis preditoras e de resposta.

O tratamento dos dados constitui um processo essencial, que inclui a normalização ou padronização das variáveis, bem como a análise e o manejo de valores ausentes. Investigar a razão da ausência desses dados é importante para aplicar as técnicas de tratamento mais adequadas, que podem variar desde a interpolação em séries temporais com espaçamento regular até a exclusão de períodos específicos, quando justificável (LAZZERI, 2020, p. 12 e 84). Esses procedimentos são cruciais para assegurar a integridade e a qualidade dos dados, fundamentais para a utilização correta dos algoritmos de previsão.

5.1 Extração dos dados

Para coleta das informações de ações da S&P foi usado como fonte o website Yahoo Finance (YAHOO FINANCE, 2023), extraído através das bibliotecas em Python para automação *yfinance* (AROUSSI, 2023) e *yahoo_fin* (TREADWAY, 2021), melhor detalhado no Apendice A.

O período total dos dados escolhido foi de 01 de Janeiro de 2022 a 30 de Setembro de 2023.

5.2 Dados nulos

Como a S&P opera apenas nos dias úteis, serão considerados para as séries somente esses dias, e os dados serão tratados como igualmente espaçados (diários). Assim, o dia imediatamente posterior a uma sexta-feira será uma segunda-feira, sem influência dos fins de semana. Isto é, não se assumirá que a dinâmica de uma sexta-feira para uma segunda-feira seja diferente da de uma quarta-feira para uma quinta-feira, por exemplo.

Poderia ser possível considerar um efeito significativo dos fins de semana na bolsa de valores e nas ações das empresas; no entanto, essa abordagem não será adotada no presente trabalho.

Além disso, considerando a natureza dos dados, ou seja, que o mercado opera continuamente nos dias úteis para todas as suas componentes (a bolsa abre ou fecha de modo equivalente para cada empresa, com todas as empresas operando nos mesmos dias e horários), essa será a abordagem adotada. Devido aos pontos levantados, resultará em um conjunto de dados contínuo e sem a presença de dados faltantes ou nulos.

5.3 Transformação dos dados

Como ilustrado na Figura 2.2, observa-se que diferentes empresas podem operar em escalas distintas. Por exemplo, uma empresa pode apresentar variações médias entre 20 e 40, enquanto outra varia de 100 a 400. Essa discrepância nas escalas pode representar um desafio ao analisar as relações dinâmicas entre as séries temporais multivariadas

Para eliminar problemas com escala de valores, será usado o retorno líquido simples em vez do preço de fechamento das ações, também chamado de taxa de retorno, os quais tem propriedades estatísticas importantes, como estacionariedade e ergodicidade, que são necessárias em alguns modelos de séries temporais (MORETTIN, 2017, p. 8). É definido a seguir:

$$R_t = \frac{P_t}{P_{t-1}} - 1 \quad (5.1)$$

Como na S&P tem-se 500 símbolos de empresas listadas e no presente trabalho o objetivo é analisar os 11 setores econômicos da seção 1.1, a estratégia adotada é agrupar ações por setor a fim de diminuir a quantidade de dados históricos.

Para o agrupamento, é definido o retorno medio \mathcal{R}_{it} por setor como sendo a média dos retornos das 10 ações com maior valor de mercado:

$$\mathcal{R}_{it} = \frac{1}{10} \sum_{k=1}^{10} \left(\frac{P_{kit}}{P_{kit-1}} - 1 \right) \quad (5.2)$$

Onde P_{kit} é o preço de fechamento da ação k , no setor i e período t .

Note que tal medida, é o retorno médio de um período para o próximo, no caso do presente trabalho, de um dia para outro. MORETTIN (2017, p. 12) apresenta tal medida, quanto se passa um período k , agregando o retorno de forma temporal, dada pela equação:

$$\mathcal{R}_{i,j}[k] = \prod_{i=0}^{k-1} (1 + \mathcal{R}_{i,j-i}) - 1, \quad (5.3)$$

Esta medida emprega uma média aritmética dos retornos diários das 10 principais ações de cada setor, sem ponderação pelo valor de mercado ou quantidade de ações. Diferente das abordagens tradicionais em carteiras de investimento que utilizam ponderação de mercado, pois trata cada ação igualmente, destacando sua contribuição uniforme ao desempenho do setor.

Esta abordagem pode ser particularmente útil para capturar os padrões gerais dos setores utilizados. Contudo, é importante frisar que tal metodologia pode não ser ideal para a maximização de retornos em carteiras de investimentos, onde a ponderação de mercado é mais representativa.

5.3.1 Top 10 ações escolhidas de cada setor

As 10 ações escolhidas com maior valor de mercado de cada setor são dadas a seguir:

Rank	Technology		Communication Services ¹		Consumer Cyclical	
1	MSFT	2.89T	GOOG	1.79T	AMZN	1.60T
2	AAPL	2.88T	META	962.39B	TSLA	695.83B
3	NVDA	1.35T	NFLX	215.41B	HD	354.02B
4	AVGO	518.55B	TMUS	187.97B	MCD	212.87B
5	ORCL	293.04B	CMCSA	173.04B	NKE	159.18B
6	ADBE	272.10B	DIS	165.37B	LOW	125.90B
7	CRM	263.23B	VZ	162.11B	BKNG	122.20B
8	AMD	236.77B	T	117.83B	TJX	108.30B
9	ACN	223.42B	CHTR	53.98B	SBUX	104.55B
10	CSCO	204.56B	EA	37.10B	ABNB	87.90B

Tabela 5.1: Top 10 empresas em valor de mercado dos setores de Tecnologia, Serviços de comunicação e Consumo Cíclico

¹ GOOGL não foi incluída no ranking devido à sua associação com a mesma entidade empresarial da GOOG e à semelhança dos valores de mercado entre as duas, evitando assim a multicolinearidade e a duplicação de informações na análise.

Rank	Healthcare		Financial Services		Consumer Defensive	
1	LLY	610.33B	V	542.92B	WMT	434.31B
2	UNH	482.36B	JPM	492.34B	PG	354.95B
3	JNJ	390.92B	MA	402.40B	COST	303.17B
4	MRK	300.61B	BAC	259.12B	KO	261.09B
5	ABBV	286.72B	WFC	170.59B	PEP	229.97B
6	TMO	210.31B	MS	148.55B	PM	147.89B
7	ABT	197.77B	BX	144.34B	MDLZ	99.51B
8	DHR	166.44B	SPGI	138.44B	MO	73.11B
9	AMGN	164.04B	AXP	132.66B	CL	66.62B
10	PFE	162.05B	GS	123.19B	TGT	65.05B

Tabela 5.2: Top 10 empresas em valor de mercado dos setores de Serviços Financeiros, Saúde e Energia

Rank	Energy		Basic Materials		Industrials	
1	XOM	399.48B	LIN	198.28B	CAT	147.84B
2	CVX	278.01B	SHW	77.43B	UNP	144.91B
3	COP	132.63B	FCX	58.71B	GE	141.31B
4	SLB	71.35B	APD	58.70B	UPS	134.87B
5	EOG	67.67B	ECL	56.72B	HON	132.57B
6	MPC	59.79B	NEM	43.38B	BA	131.70B
7	PSX	58.46B	NUE	41.52B	RTX	124.13B
8	PXD	52.26B	DOW	37.43B	LMT	114.92B
9	OXY	50.95B	PPG	34.32B	DE	108.22B
10	VLO	44.77B	CTVA	32.49B	ADP	97.05B

Tabela 5.3: Top 10 empresas em valor de mercado dos setores de Consumo Defensivo, Materiais Básicos e Industriais

Rank	Utilities		Real Estate	
1	NEE	125.24B	PLD	121.55B
2	SO	77.89B	AMT	97.63B
3	DUK	75.88B	EQIX	76.52B
4	SRE	47.46B	SPG	55.01B
5	PCG	44.69B	PSA	52.02B
6	AEP	42.85B	WELL	50.88B
7	D	39.61B	CCI	48.88B
8	CEG	36.06B	O	42.64B
9	EXC	35.91B	DLR	42.40B
10	XEL	33.83B	CSGP	34.06B

Tabela 5.4: Top 10 empresas em valor de mercado dos setores de Utilidades Públicas e Imobiliário

5.3.2 Retornos médios dos setores

Agora, utiliza-se essas ações escolhidas para extrair suas séries temporais e então aplica o retorno médio de 5.2 em cada setor. a representação gráfica deste são mostrados a seguir:

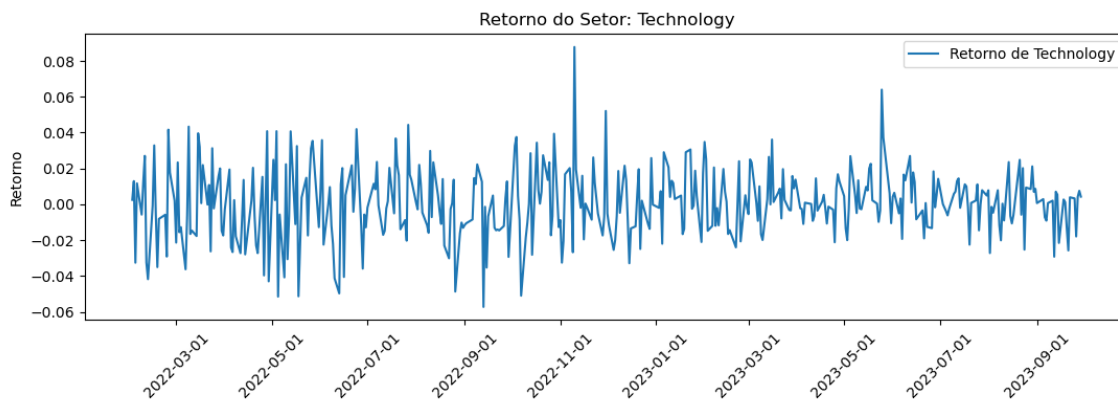


Figura 5.1: Retorno médio do Setor de Tecnologia

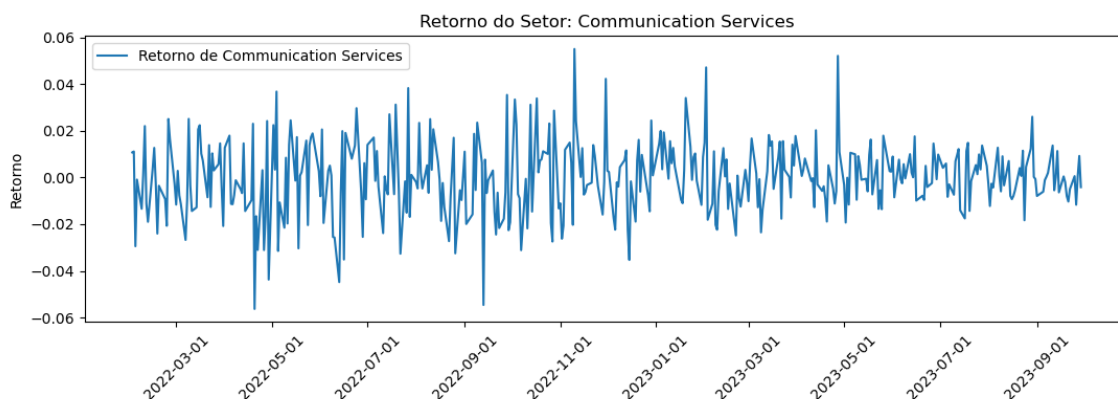


Figura 5.2: Retorno médio do Setor de Serviços de Comunicação

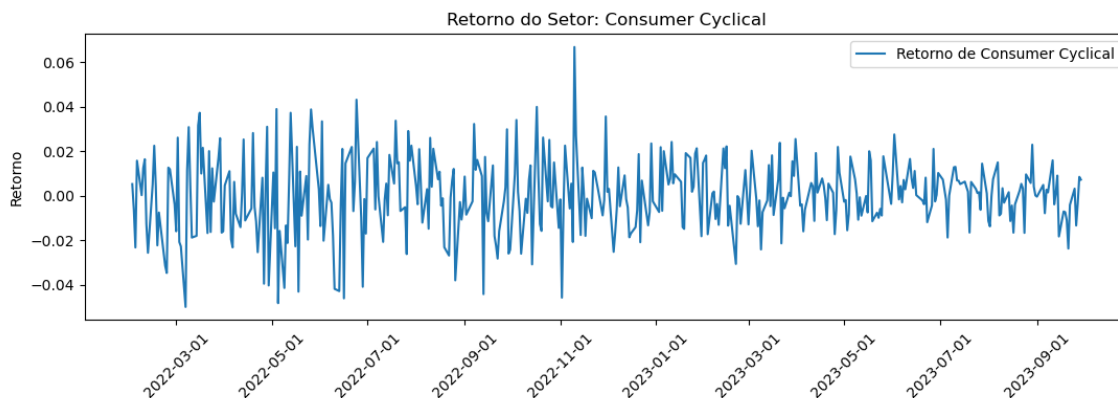


Figura 5.3: Retorno médio do Setor de Consumo Cíclico

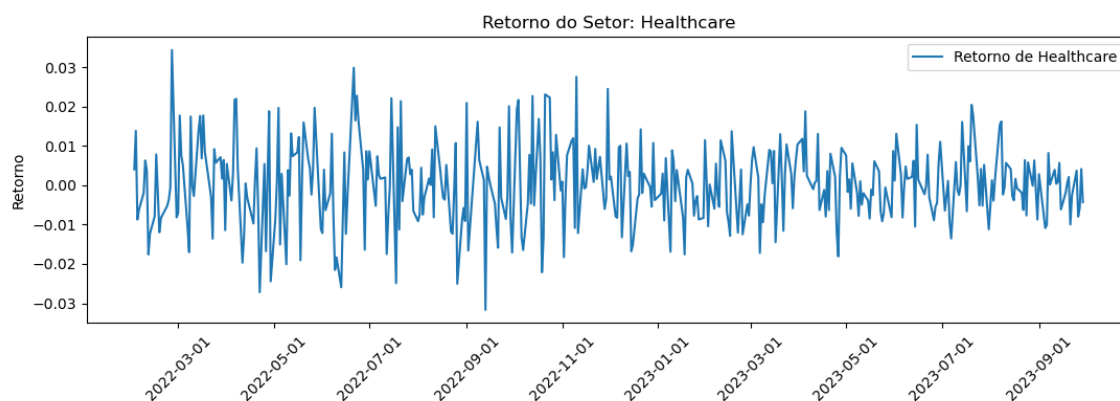


Figura 5.4: Retorno médio do Setor de Saúde

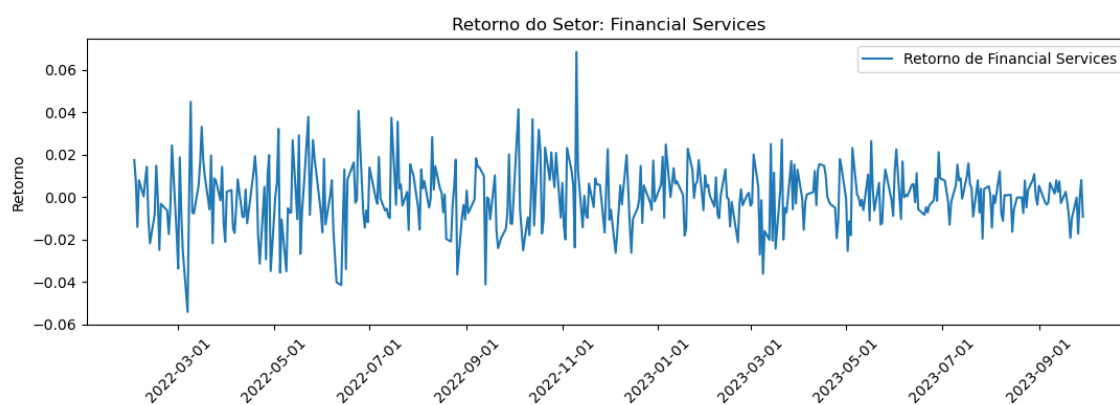


Figura 5.5: Retorno médio do Setor de Serviços Financeiros

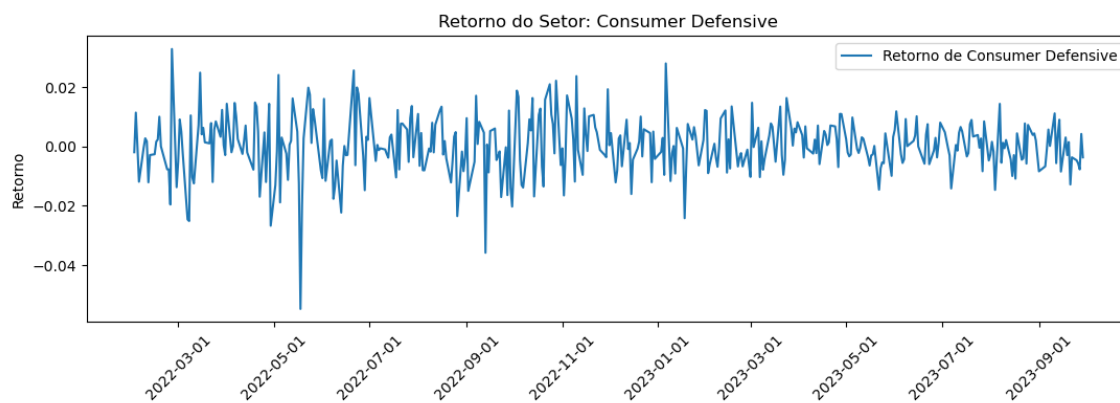


Figura 5.6: Retorno médio do Setor de Bens de Consumo Discricionário

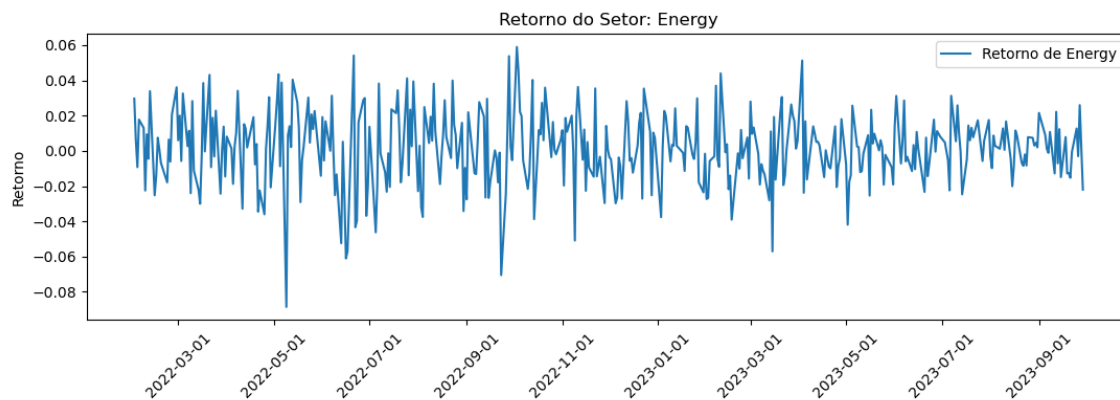


Figura 5.7: Retorno médio do Setor de Energia

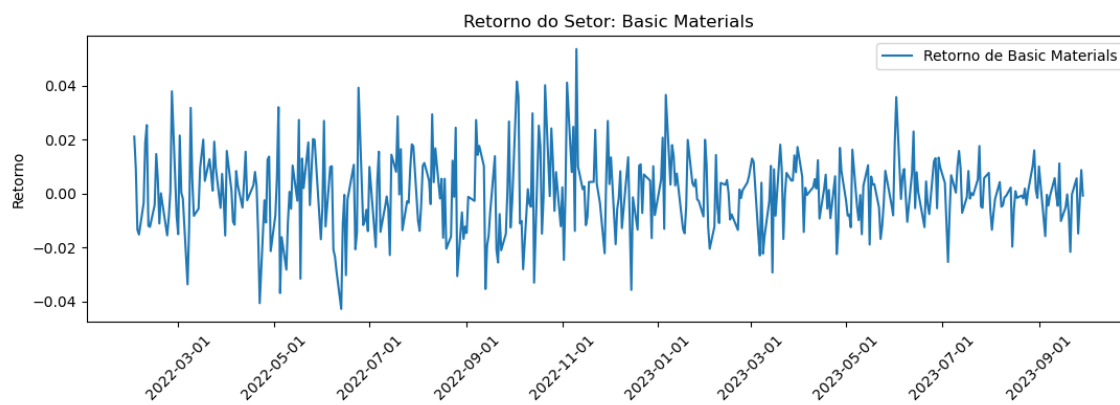


Figura 5.8: Retorno médio do Setor de Materiais Básicos

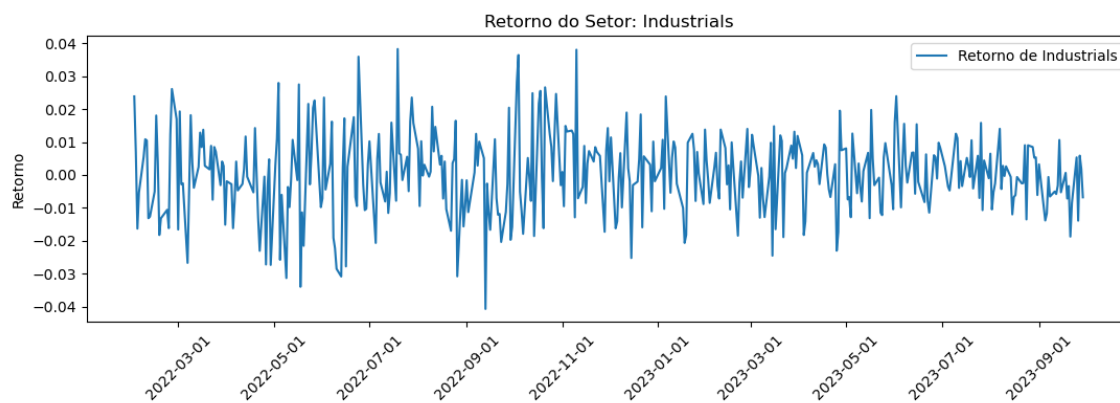


Figura 5.9: Retorno médio do Setor Industrial

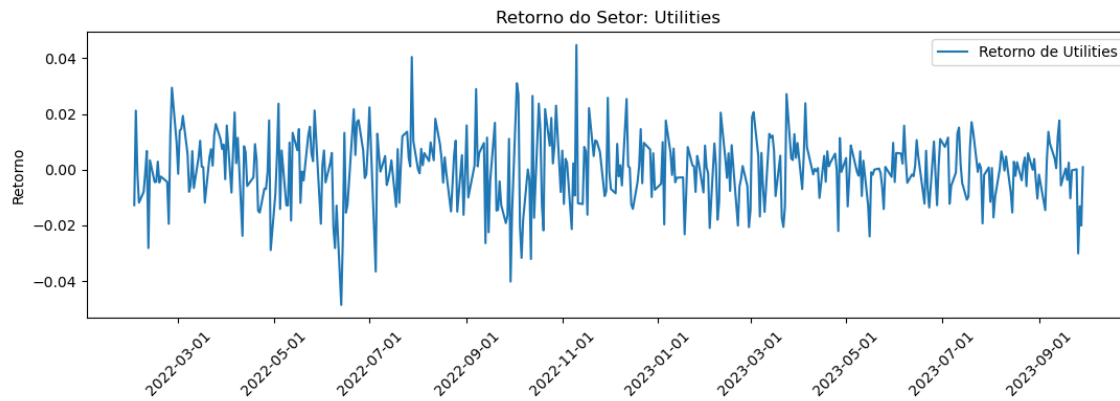


Figura 5.10: Retorno médio do Setor de Serviços Públicos

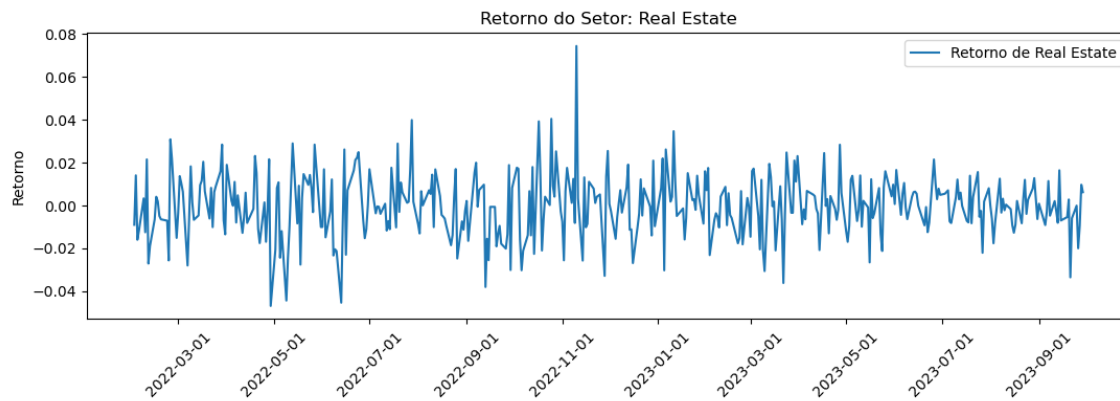


Figura 5.11: Retorno médio do Setor Imobiliário

Note que as séries aparentam serem estacionárias, como afirmado por MORETTIN (2017, p. 8) ao se utilizar retornos financeiros simples. Uma das formas de verificar a estacionariedade é com o Teste de Dickey-Fuller Aumentado (Teste ADF), que verifica, como hipótese nula, se modelos autoregressivos possui raiz unitária, característica esta que torna as séries não estacionárias. Além deste, possui também o teste de cointegração de johansen que verifica se é necessário aplicar diferenças a séries para que se torne estacionária, isto abre margem para outras famílias de modelos econométricos como VECM e VARIMA.

5.4 Remodelagem de séries temporais para um problema de Aprendizado Supervisionado

Para modelos de Aprendizado de máquina, é necessário remodelar os dados para a forma da equação 4.2. As tabelas 5.5 e 5.6 exemplificam essa diferença da estrutura clássica de séries temporais e o formato de Aprendizado de máquina supervisionado (LAZZERI, 2020, p. 4):

ID	valor
1	25
1	27
2	31
2	40
3	18
3	31

ID	variavel de tempo	valor
1	01/01/2023	25
1	07/01/2023	27
2	01/01/2023	31
2	07/01/2023	40
3	01/01/2023	18
3	07/01/2023	31

Tabela 5.5: Exemplo de dados de aprendizado de máquina

Tabela 5.6: Exemplo de dados de série temporal

Note que no caso de séries temporais, como no exemplo da Tabela 5.6 os dados são sequenciais, com um índice temporal, enquanto em aprendizado de máquina supervisionado de modo geral isto não é necessário, como em 5.5.

Entretanto, é possível remodelar um problema de série temporal em um problema de aprendizado supervisionado, por exemplo, utilizando passos anteriores como parte da entrada X e passos posteriores como parte da saída Y (LAZZERI, 2020, p. 18).

tempo	$z_1 t$	$z_2 t$
01/02/2023	12	89
02/02/2023	0	7
03/02/2023	34	42
04/02/2023	21	22
05/02/2023	12	1

↓

x_1	x_2	y_1	y_2
-	-	12	89
12	89	0	7
0	7	34	42
34	42	21	22
21	22	12	1
12	1	y_1 a ser predito	y_2 a ser predito

Tabela 5.7: Exemplo de uma série temporal bivariada transformada em um problema de aprendizado supervisionado para predição de 1 passo à frente utilizando 1 passo anterior

Seja uma série temporal multivariada (Z_1, Z_2, \dots, Z_T) , com $Z_t = (z_{1t}, z_{2t}, \dots, z_{nt})'$, a forma descrita de remodelagem para predição de 1 passos à frente utilizando 1 passo anterior é exemplificada na tabela 5.7 (BROWNLEE, 2018, p. 27).

O caso geral dessa remodelagem para um exemplo de predição de 2 passos à frente utilizando 3 passos anteriores é dado pela tabela 5.8

X	Y
$\begin{bmatrix} z_{11} & z_{21} & \cdots & z_{n1} \\ z_{12} & z_{22} & \cdots & z_{n2} \\ z_{13} & z_{23} & \cdots & z_{n3} \\ z_{12} & z_{22} & \cdots & z_{n2} \\ z_{13} & z_{23} & \cdots & z_{n3} \\ z_{14} & z_{24} & \cdots & z_{n4} \\ \vdots & & & \end{bmatrix}$	$\begin{bmatrix} z_{14} & z_{24} & \cdots & z_{n4} \\ z_{15} & z_{25} & \cdots & z_{n5} \\ \\ z_{15} & z_{25} & \cdots & z_{n5} \\ z_{16} & z_{26} & \cdots & z_{n6} \\ \vdots & & & \end{bmatrix}$
$\begin{bmatrix} z_{1,T-4} & z_{2,T-4} & \cdots & z_{n,T-4} \\ z_{1,T-3} & z_{2,T-3} & \cdots & z_{n,T-3} \\ z_{1,T-2} & z_{2,T-2} & \cdots & z_{n,T-2} \\ z_{1,T-3} & z_{2,T-3} & \cdots & z_{n,T-3} \\ z_{1,T-2} & z_{2,T-2} & \cdots & z_{n,T-2} \\ z_{1,T-1} & z_{2,T-1} & \cdots & z_{n,T-1} \\ z_{1,T-2} & z_{2,T-2} & \cdots & z_{n,T-2} \\ z_{1,T-1} & z_{2,T-1} & \cdots & z_{n,T-1} \\ z_{1,T} & z_{2,T} & \cdots & z_{n,T} \end{bmatrix}$	$\begin{bmatrix} z_{1,T-1} & z_{2,T-1} & \cdots & z_{n,T-1} \\ z_{1,T} & z_{2,T} & \cdots & z_{n,T} \\ \\ z_{1,T} & z_{2,T} & \cdots & z_{n,T} \\ z_{1,T+1} & z_{2,T+1} & \cdots & z_{n,T+1} \\ \\ z_{1,T+1} & z_{2,T+1} & \cdots & z_{n,T+1} \\ z_{1,T+2} & z_{2,T+2} & \cdots & z_{n,T+2} \end{bmatrix}$

Tabela 5.8: Remodelagem dos dados de série temporal para aprendizado supervisionado para predição de 2 passos À frente utilizando 3 passos anteriores

Ou ainda, de forma mais enxuta, representada pela tabela 5.9. Note que Z_{T+1} e Z_{T+2} , são os valores que irão ser preditos, ou seja, são desconhecidos.

X	Y
$[Z_1 \ Z_2 \ Z_3]'$	$[Z_4 \ Z_5]'$
$[Z_2 \ Z_3 \ Z_4]'$	$[Z_5 \ Z_6]'$
\vdots	\vdots
$[Z_{T-4} \ Z_{T-3} \ Z_{T-2}]'$	$[Z_{T-1} \ Z_T]'$
$[Z_{T-3} \ Z_{T-2} \ Z_{T-1}]'$	$[Z_T \ Z_{T+1}]'$
$[Z_{T-2} \ Z_{T-1} \ Z_T]'$	$[Z_{T+1} \ Z_{T+2}]'$

Tabela 5.9: Remodelagem simplificada dos dados de série temporal para aprendizado supervisionado para predição de 2 passos à frente utilizando 3 passos anteriores

O padrão se segue para dimensões maiores, como mostra BROWNLEE (2018, p. 81). Na tabela 5.10 e 5.11 são mostrados a forma geral dessa remodelagem, com uma série de n componentes, p passos anteriores usados para predição e h passos que se deseja prever:

t	Z			
1	$Z_{1,1}$	$Z_{2,1}$	\cdots	$Z_{n,1}$
2	$Z_{1,2}$	$Z_{2,2}$	\cdots	$Z_{n,2}$
\vdots		\vdots		
T	$Z_{1,T}$	$Z_{2,T}$	\cdots	$Z_{n,T}$
T + 1	$Z_{1,T+1}$	$Z_{2,T+1}$	\cdots	$Z_{n,T+1}$
T + 2	$Z_{1,T+2}$	$Z_{2,T+2}$	\cdots	$Z_{n,T+2}$
\vdots		\vdots		
T + h	$Z_{1,T+h}$	$Z_{2,T+h}$	\cdots	$Z_{n,T+h}$

Tabela 5.10: Série temporal generalizada em formato de tabela

X	Y
$[Z_1 \ \cdots \ Z_p]'$	$[Z_{p+1} \ \cdots \ Z_{p+h}]'$
$[Z_2 \ \cdots \ Z_{p+1}]'$	$[Z_{p+2} \ \cdots \ Z_{p+h+1}]'$
\vdots	\vdots
$[Z_{T-p} \ \cdots \ Z_{T-1}]'$	$[Z_T \ \cdots \ Z_{T+h-1}]'$
$[Z_{T-p+1} \ \cdots \ Z_T]'$	$[Z_{T+1} \ \cdots \ Z_{T+h}]'$

Tabela 5.11: Generalização da remodelagem dos dados de série temporal para aprendizado supervisionado utilizando p passos anteriores para prever até h passos à frente

Note que $Z_{T+1}, Z_{T+2}, \dots, Z_{T+h}$ são os valores que serão preditos, isto é, são desconhecidos.

5.4.1 Formato dos dados para MLP

Observe que, nos exemplos anteriores, X e Y são estruturados como matrizes de matrizes, ou seja, são tensores, o que traz abordagens matemáticas mais complexas para operá-los.

Esse formato é compatível com vários modelos de aprendizado de máquina especializados no tratamento de séries temporais. Contudo, no caso específico de MLP, uma transformação adicional é necessária para adaptar esses tensores a uma forma bidimensional. Tal transformação é mostrada na tabela 5.12

X	Y
$z_{1,1}, \dots, z_{1,n}, \dots, z_{p,1}, \dots, z_{p,n}$	$z_{p+1,1}, \dots, z_{p+1,n}, \dots, z_{p+h,1}, \dots, z_{p+h,n}$
\vdots	\vdots
$z_{T-p,1}, \dots, z_{T-p,n}, \dots, z_{T,1}, \dots, z_{T,n}$	$z_{T-p+h,1}, \dots, z_{T-p+h,n}, \dots, z_{T+h,1}, \dots, z_{T+h,n}$

Tabela 5.12: Dados de série temporal achatados para aprendizado supervisionado

O algoritmo computacional para realizar essas transformações podem ser consultadas nos Apêndices B.3 e B.3.

5.4.2 Exemplo prático

Exemplo extraído de [BROWNLEE \(2018, p. 83\)](#) onde são usados 3 dias anteriores para prever 2 dias à frente:

Série temporal original:

t	Z		
1	10	15	25
2	20	25	45
3	30	35	65
4	40	45	85
5	50	55	105
6	60	65	125
7	70	75	145
8	80	85	165
9	90	95	185

Tabela 5.13: Exemplo simples de uma série temporal multivariada. Fonte: [BROWNLEE \(2018, p. 81\)](#)

Transformação para aprendizado supervisionado:

X	Y
$\begin{bmatrix} 10 & 15 & 25 \\ 20 & 25 & 45 \\ 30 & 35 & 65 \end{bmatrix}$	$\begin{bmatrix} 40 & 45 & 85 \\ 50 & 55 & 105 \end{bmatrix}$
$\begin{bmatrix} 20 & 25 & 45 \\ 30 & 35 & 65 \\ 40 & 45 & 85 \end{bmatrix}$	$\begin{bmatrix} 50 & 55 & 105 \\ 60 & 65 & 125 \end{bmatrix}$
$\begin{bmatrix} 30 & 35 & 65 \\ 40 & 45 & 85 \\ 50 & 55 & 105 \end{bmatrix}$	$\begin{bmatrix} 60 & 65 & 125 \\ 70 & 75 & 145 \end{bmatrix}$
$\begin{bmatrix} 40 & 45 & 85 \\ 50 & 55 & 105 \\ 60 & 65 & 125 \end{bmatrix}$	$\begin{bmatrix} 70 & 75 & 145 \\ 80 & 85 & 165 \end{bmatrix}$
$\begin{bmatrix} 50 & 55 & 105 \\ 60 & 65 & 125 \\ 70 & 75 & 145 \end{bmatrix}$	$\begin{bmatrix} 80 & 85 & 165 \\ 90 & 95 & 185 \end{bmatrix}$

Tabela 5.14: Exemplo simples da remodelagem de uma série temporal multivariada para o formato de aprendizado supervisionado. Fonte: [BROWNLEE \(2018, p. 83\)](#)

Fazendo a transformação final para ser possível usá-lo em MLP:

X (após achatamento)	Y (após achatamento)
10, 15, 25, 20, 25, 45, 30, 35, 65	40, 45, 85, 50, 55, 105
20, 25, 45, 30, 35, 65, 40, 45, 85	50, 55, 105, 60, 65, 125
30, 35, 65, 40, 45, 85, 50, 55, 105	60, 65, 125, 70, 75, 145
40, 45, 85, 50, 55, 105, 60, 65, 125	70, 75, 145, 80, 85, 165
50, 55, 105, 60, 65, 125, 70, 75, 145	80, 85, 165, 90, 95, 185

Tabela 5.15: *Exemplo remodelado para ser possível aplicar MLP*

Capítulo 6

Técnicas de treinamento e avaliação dos modelos

O conjunto de dados utilizado para ajustar os parâmetros dos modelos é comumente chamado de conjunto de treinamento. Nele, fórmulas analíticas ou algoritmos de otimização são empregados para encontrar parâmetros e estimadores que sejam suficientemente acurados ou, no melhor dos casos, ótimos, com base em métricas específicas como a minimização do erro de predição. O ajuste dos modelos para efetuar previsões acerca de dados futuros é um dos principais objetivos deste estudo.

6.1 Divisão do conjunto inicial de dados

É crucial reconhecer que treinar um modelo com um conjunto de dados e avaliar sua acurácia no mesmo conjunto pode introduzir viés e criar uma expectativa excessivamente otimista de sua capacidade preditiva. Tal situação é ilustrada ao examinarmos um caso extremo, descrito por [BROWNLEE \(2017, p. 145\)](#), no qual um modelo memoriza os valores das séries temporais em cada instante. Nesse cenário, ao fazer previsões para um momento já conhecido durante o treinamento, o modelo apresentará uma correspondência exata, não por ter apreendido o padrão subjacente dos dados, mas por ter previamente acessado e memorizado esses valores.

Dessa forma, é essencial dividir o conjunto de dados inicial em, pelo menos, dois subconjuntos: geralmente o de treinamento e o de teste, como visto na figura [6.1](#). O conjunto de treinamento é usado para o ajuste do modelo, enquanto o de teste serve para avaliar o desempenho do modelo com dados novos ([JAMES *et al.*, 2023, p. 203](#)). Em alguns casos, adota-se uma divisão em três conjuntos — treinamento, validação e teste — permitindo testar o desempenho de diversos modelos e parâmetros distintos com os conjuntos de treinamento e validação.

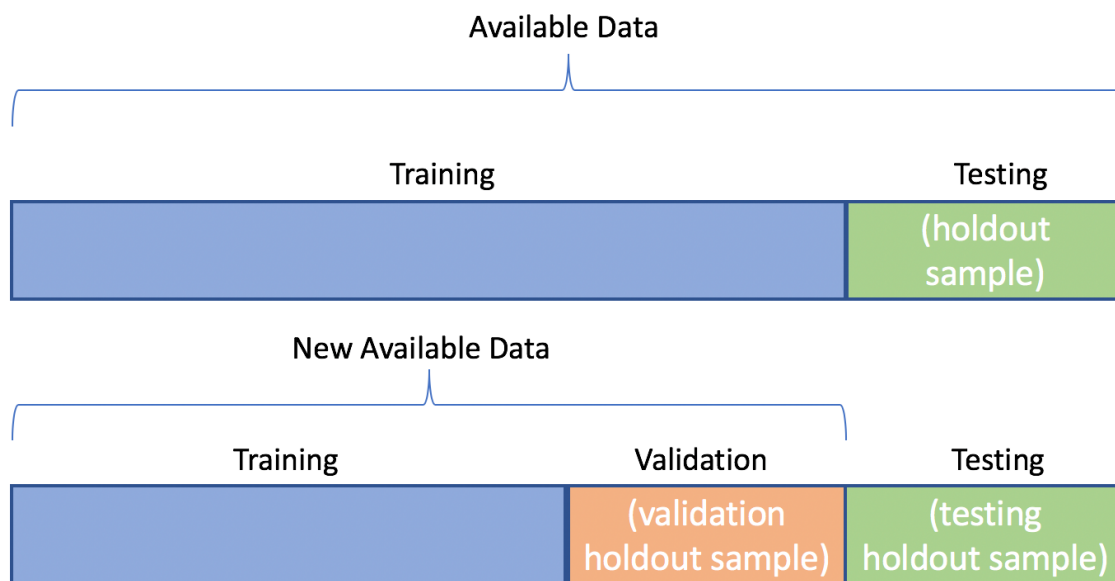


Figura 6.1: Divisão do conjunto de dados em treinamento, validação e teste. Fonte: GREG BLAND (2020)

Após executado os diferentes modelos com parâmetros e escolhido o mais eficaz segundo alguma métrica, este é finalmente testado no conjunto de teste para verificar o desempenho final em dados inéditos (AUFFARTH, 2021, p. 105), processo este exibido na figura 6.2.

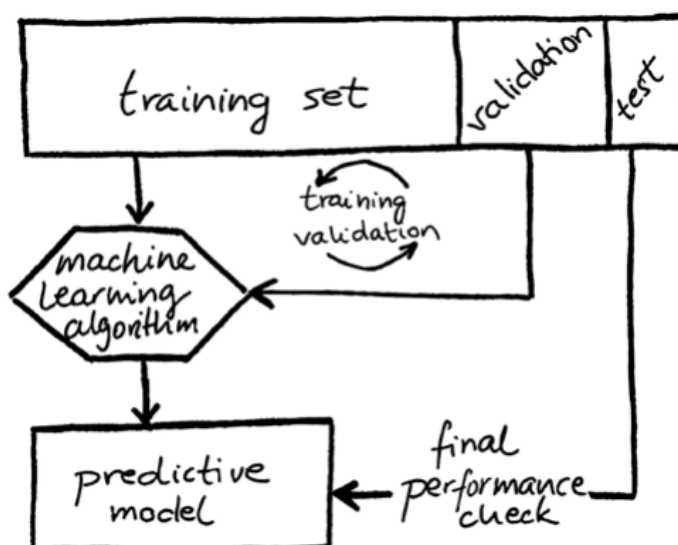


Figura 6.2: Processo de treinamento com conjuntos de treino e validação, escolha do modelo e uso do teste para checar a performance final. Fonte: AUFFARTH (2021)

6.2 Cross-Validation k-folds em aprendizado de máquina com dados não sequenciais

No contexto de aprendizado de máquina supervisionado, a validação cruzada k-folds, ou "cross-validation k-folds" em inglês, é uma estratégia amplamente adotada para treinar e avaliar modelos e seus diversos parâmetros. Essa técnica consiste em dividir aleatoriamente o conjunto de treinamento em k subconjuntos de tamanhos aproximadamente iguais, conhecidos como 'folds' (JAMES *et al.*, 2023, p. 206).

Na prática, o primeiro 'fold' é empregado como o conjunto de validação, enquanto os outros $k - 1$ 'folds' são usados para o treinamento do modelo. Após o modelo ser ajustado, uma métrica de avaliação de erro — como o Erro Quadrático Médio (MSE) ou a Raiz do Erro Quadrático Médio (RMSE) — é calculada utilizando os dados do conjunto de validação.

Este procedimento é repetido de forma que cada 'fold' seja utilizado uma vez como conjunto de validação, enquanto os subconjuntos restantes compõem o conjunto de treinamento. Um exemplo é mostrado na figura 6.3.

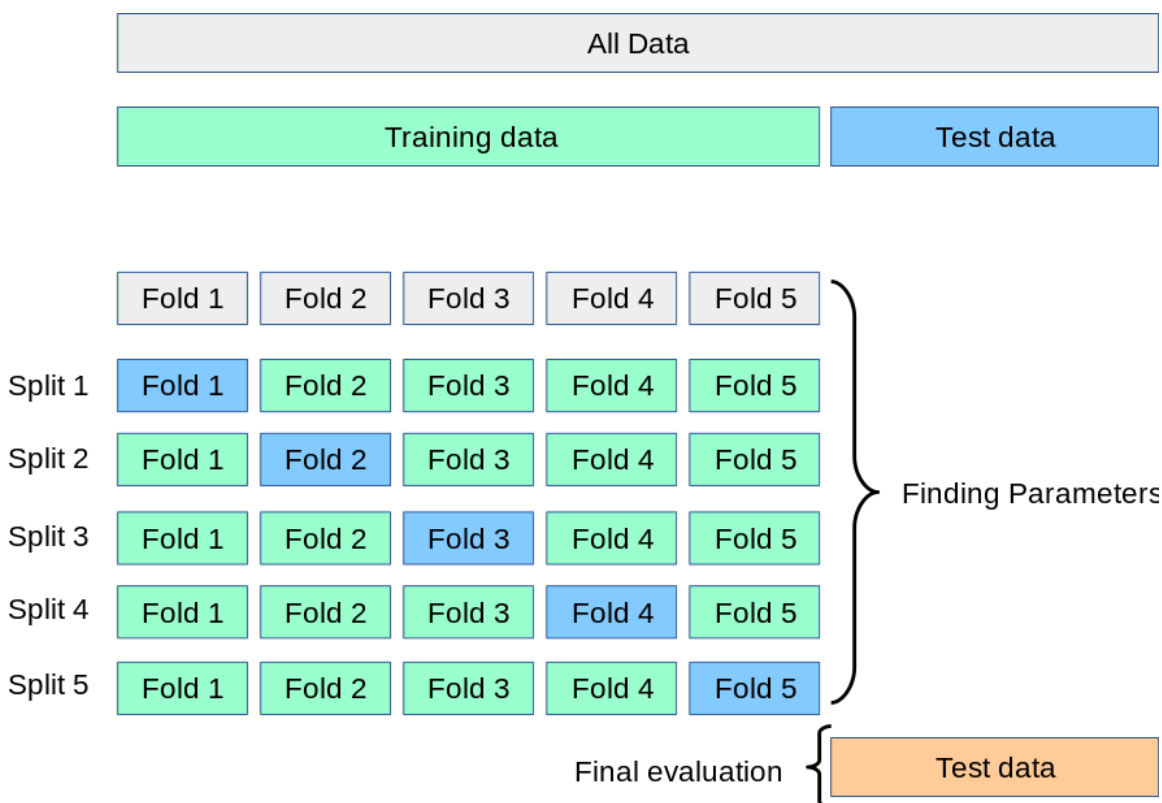


Figura 6.3: Exemplo de Cross-Validation k-folds com 5 folds. Fonte: PEDREGOSA *et al.* (2011)

Após a realização de todas as k iterações da validação cruzada k-folds, em que cada 'fold' i atua como conjunto de validação exatamente uma vez, a medida de erro para cada iteração $Erro_i$ é registrada. A métrica final de erro, $CV_{(k)}$, é então calculada como a média das medidas de erro de cada iteração, conforme a seguinte equação:

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^k Erro_i \quad (6.1)$$

Isso resulta em uma avaliação de desempenho mais robusta do modelo, pois cada parte do conjunto de treinamento contribui tanto para o treinamento quanto para a validação. Dessa forma, minimiza-se o risco de que uma boa performance de predição seja resultado de uma divisão fortuita dos dados entre os conjuntos de treino e validação.

6.2.1 O problema de aplicar Cross-Validation k-folds em séries temporais

O método de validação cruzada k-folds, amplamente utilizado no aprendizado de máquina, pode apresentar problemas de viés ao ser aplicado em séries temporais, especialmente nas financeiras.

Na sua forma original, o método divide o conjunto de dados em subconjuntos de maneira aleatória, sem levar em conta a sequência temporal. Para contornar isso, a divisão deve ser feita de forma que respeite a ordem cronológica dos dados.

Como apontado por [DE PRADO \(2018, p. 104\)](#), existe uma forte tendência de que valores consecutivos em séries temporais sejam correlacionados, ou seja, Z_t tende a ser próximo de Z_{t+1} . Isso significa que se os dados forem divididos sem levar em conta essa correlação, o modelo pode ter acesso, durante o treinamento, a informações similares às aquelas presentes no conjunto de teste, levando a um possível vazamento de informações, de forma análoga ao exemplo de memorização de dados citado na seção [6.1](#).

O problema é ainda maior em séries que possuem padrões autoregressivos que dependem de dados passados para previsões futuras. Nessas séries, como costuma ser utilizado os modelos VAR e VARMA, a utilização de dados históricos no treinamento pode capturar padrões das defasagens que não serão replicáveis em dados futuros desconhecidos. Essa sobreposição de informações entre conjuntos de treino e teste resulta em estimativas de desempenho infladas e não representativas da capacidade preditiva do modelo em dados verdadeiramente novos.

O desafio de vazamento de informações em séries temporais durante a validação cruzada k-folds é amplamente discutido por [BERGMEIR *et al.* \(2018\)](#). Estratégias para mitigar esse problema são propostas por [DE PRADO \(2018, p. 105\)](#), que sugere a "purificação" dos dados como uma solução viável, isto é, se propõe a excluir dados na fronteira entre treino e teste a fim de contornar o problema de correlação autoregressiva dos dados, além de cuidados adicionais.

No entanto, devido à complexidade de determinar a forma certa de eliminar o vazamento de dados e viés por correlação serial e o foco na simplicidade e praticidade neste trabalho, o método de validação cruzada k-folds não será adotado.

6.3 Cross-Validation de Séries Temporais

No estudo de séries temporais, duas abordagens amplamente reconhecidas e aceitas para dividir os dados em conjuntos de treinamento e teste são as técnicas de **Janelas Expansivas** (*Expanding Windows*) e de **Janelas Móveis** (*Rolling Windows*) (CHAN e MÁTYÁS, 2022, p. 116).

A validação com janelas móveis funciona dividindo o conjunto de dados de séries temporais em um número de "janelas" ou "folds" que rolam ao longo do tempo. Em cada iteração, um conjunto contíguo de dados é usado para treinar o modelo, e os dados imediatamente subsequentes são usados para testar o modelo. Essa abordagem garante que o treinamento é sempre realizado em dados do passado para prever dados futuros, respeitando a ordem cronológica dos eventos.

Uma das principais vantagens da validação com janelas móveis é a sua capacidade de simular um cenário mais realista de como o modelo será utilizado na prática. Ao treinar o modelo em diferentes subconjuntos de dados ao longo do tempo e testá-lo em amostras subsequentes, é possível obter uma compreensão mais confiável de como o modelo se comportará diante de novos dados.

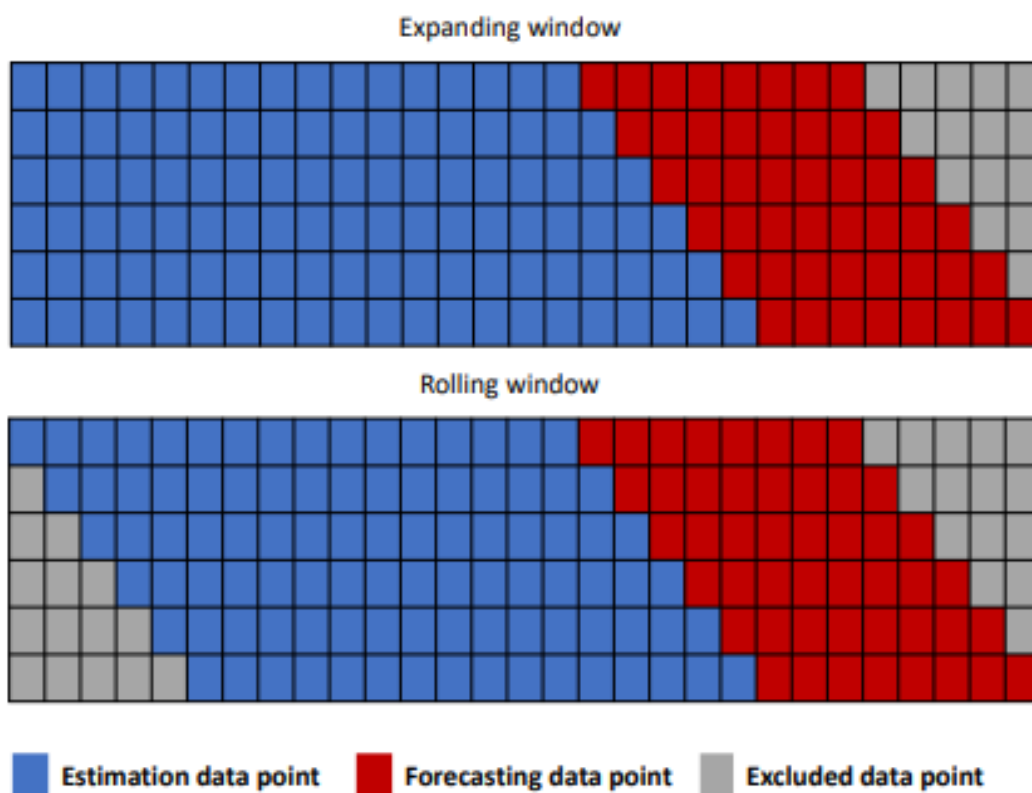


Figura 6.4: Técnicas de Janelas Expansivas e Janelas Móveis para separação entre treino e teste dos dados. Fonte: CHAN e MÁTYÁS (2022)

6.3.1 Janelas Expansivas dos dados que serão utilizadas

Para este trabalho é escolhido a técnica de separação dos dados com Janelas Expansivas dos dados, separando os subconjuntos de forma manual, como é feito por vezes na literatura (BROWNLEE, 2017, p. 148).

O período inicial extraído, como citado no capítulo 5, é de 01 de Janeiro de 2022 a 30 de Setembro de 2023 e a divisão será feita em 3 janelas de treino e teste:

1. treino: 03/01/2020 a 11/07/2022, teste: 12/07/2022 a 18/07/2022
2. treino: 03/01/2020 a 13/01/2023, teste: 17/01/2023 a 18/07/2023
3. treino: 03/01/2020 a 22/09/2023, teste: 25/09/2023 a 29/09/2023

Note que a primeira janela de treino possui 130 dias úteis, segunda janela 260 e na terceira, 433, enquanto as três janelas de teste possuem 5 dias úteis cada.

Capítulo 7

Implementação dos modelos

7.1 VAR(p)

Para a implementação do modelo VAR(p) descrito em 3.19, utilizou a biblioteca do Python *Statsmodels* (SEABOLD e PERKTOLD, 2010),

Para a seleção p do modelo foi executado o modelo com várias ordens diferentes e para cada uma foi avaliado seus critérios de informações AIC, BIC e HQ de 3.47, com os dados obtidos na seção 5, de 4 de Janeiro de 2022 até 22 de agosto de 2023, com a implementação detalhada no Apêndice B:

	Treino1			Treino2			Treino3		
	AIC	BIC	HQ	AIC	BIC	HQ	AIC	BIC	HQ
1	-100.40	-97.72	-99.31	-101.51	-99.84	-100.84	-103.28	-102.14	-102.83
2	-99.76	-94.37	-97.57	-101.10	-97.77	-99.76	-103.06	-100.77	-102.16
3	-99.11	-90.98	-95.81	-100.82	-95.81	-98.81	-102.83	-99.40	-101.48
4	-98.92	-88.02	-94.49	-100.50	-93.80	-97.81	-102.64	-98.06	-100.83
5	-98.46	-84.77	-92.90	-100.05	-91.65	-96.67	-102.36	-96.62	-100.09
6	-98.69	-82.18	-91.98	-99.69	-89.58	-95.62	-102.16	-95.26	-99.44
7	-99.58	-80.22	-91.72	-99.43	-87.60	-94.67	-101.96	-93.90	-98.78
8	-101.13	-78.88	-92.09	-99.08	-85.52	-93.62	-101.77	-92.54	-98.12
9	-105.95	-80.79	-95.73	-98.95	-83.66	-92.80	-101.71	-91.31	-97.60
10	-149.00	-120.90	-137.59	-98.89	-81.84	-92.03	-101.55	-89.97	-96.97

Tabela 7.1: Critérios de informação no modelo VAR nas 3 janelas de treinamentos para cada valor da ordem p

Conforme discutido na Seção 6.3.1 e na Subseção 3.2.1, ao analisar os critérios de informação AIC, BIC e HQ para a seleção da ordem do modelo VAR, observa-se um padrão distinto entre os diferentes conjuntos de treinamento.

Nos conjuntos de Treino 2 e Treino 3, o valor mínimo para todos os três critérios é encontrado na ordem 1. Isso sugere que uma ordem mais baixa é mais adequada para

esses conjuntos, indicando uma estrutura de dependência temporal mais simples nos dados.

Por outro lado, no conjunto de Treino 1, um comportamento diferente é observado, os critérios de informação inicialmente diminuem e, após atingirem uma certa ordem, começam a aumentar conforme a ordem do modelo é estendida até a ordem máxima testada, que é 10. Este aumento nos valores dos critérios de informação em ordens mais altas pode ser interpretado como um indicativo de sobreajuste.

Neste caso, o modelo parece estar se ajustando excessivamente aos detalhes específicos dos dados de treinamento, potencialmente "decorando" a dinâmica dos dados, em vez de capturar as verdadeiras relações subjacentes. Esta tendência de sobreajuste é uma preocupação importante, pois pode levar a previsões menos precisas em novos dados.

Portanto, com base nessa análise, a ordem 1 é escolhida para os modelos VAR nos conjuntos de Treino 2 e 3, devido à sua capacidade de minimizar os critérios de informação. Para o Treino 1, a cautela é recomendada devido ao potencial sobreajuste em ordens mais altas.

Observa-se que, analisando os 3 critérios nos diferentes subconjuntos utilizado, o mínimo se encontra na ordem 1, isto é, o modelo final utiliza apenas o dia anterior para prever os próximos dias.

O modelo VAR(1) para os retornos obtidos, $\mathcal{R}_t = (\mathcal{R}_{1,t}, \dots, \mathcal{R}_{11,t})$, fica expresso por:

$$\mathcal{R}_t = A\mathcal{R}_{t-1} + U_t \quad (7.1)$$

Com \mathcal{R}_t , \mathcal{R}_{t-1} e U_t sendo vetores de dimensão (11×1) e A uma matriz de dimensão (11×11)

Com isso, é implementado novamente o modelo VAR, com a ordem 1 escolhida, melhor detalhado no Apêndice B e então é feita a predição dos 5 dias seguintes, avaliando o erro quadrático médio (EQM) e a raiz do erro quadrático médio (REQM):

(a) Treino1		
Data	MSE	RMSE
2022-07-12	0.000101	0.010051
2022-07-13	0.000046	0.006782
2022-07-14	0.000121	0.010999
2022-07-15	0.000428	0.020677
2022-07-18	0.000153	0.012361

(b) Treino2		
Data	MSE	RMSE
2023-01-17	0.000040	0.006352
2023-01-18	0.000314	0.017734
2023-01-19	0.000121	0.011010
2023-01-20	0.000337	0.018367
2023-01-23	0.000158	0.012569

(c) Treino3		
Data	MSE	RMSE
2023-09-25	0.000022	0.004708
2023-09-26	0.000253	0.015906
2023-09-27	0.000100	0.010000
2023-09-28	0.000081	0.008994
2023-09-29	0.000069	0.008317

Tabela 7.2: Resultados de MSE e RMSE do VAR(1) para diferentes períodos de treinamento

7.2 VARMA(p, q)

O modelo VARMA(p,q) foi implementado com o mesmo período que o modelo VAR(p) citado anteriormente, também foi utilizado a biblioteca do Python *Statsmodel* (SEABOLD e PERKTOLD, 2010).

Para a seleção dos parâmetros p e q , utilizou-se os mesmos critérios que o modelo anterior, isto é, AIC, BIC e HQ, de forma a escolher os parâmetros que minimizem o critério. De forma análoga ao VAR, foi aplicado nos 3 conjuntos de treino, com p e q sendo 1 ou 2, a fim de utilizar um modelo mais parcimonioso (de ordem mais baixa), como sugerido na seção 3.2.2. O resultado dos critérios de informação é exibido na tabela 7.3.

No caso do conjunto de Treino 2, o programa utilizado não conseguiu extrair os 3 critérios para $p = 2$ devido a limitações intrínsecas ao modelo e também do método de otimização utilizado. A questão da convergência e do sucesso dos algoritmos em modelos VARMA e as dificuldades associadas à estimação desses modelos podem ser afetados quando os dados não possui um padrão adequado à aplicação de um modelo VARMA, quando os valores de p e q são grandes, quando há muitas variáveis a serem preditas ou

Treino 1				
p	q	AIC	BIC	HQ
1	1	-8626.66	-7743.46	-8267.79
1	2	-8524.52	-7294.34	-8024.66
2	1	-8526.27	-7296.10	-8026.41
2	2	-8300.37	-6723.23	-7659.53

Treino 2				
p	q	AIC	BIC	HQ
1	1	-17891.11	-16794.42	-17450.23
1	2	-17774.95	-16247.42	-17160.87

Treino 3				
p	q	AIC	BIC	HQ
1	1	-30821.04	-29567.25	-30326.10
1	2	-30705.37	-28959.03	-30015.99
2	1	-30703.69	-28957.34	-30014.30
2	2	-30467.40	-28228.50	-29583.57

Tabela 7.3: Resultados dos critérios de informação AIC, BIC e HQ do modelo VARMA para diferentes valores de p e q

quando a amostra não possui um tamanho suficiente para o modelo conseguir se ajustar bem.

Novamente, os critérios quando minimizados apontam para a escolha dos menores parâmetros, $p = 1$ e $q = 1$, este será a ordem utilizada.

O modelo VARMA(1,1) para os retornos obtidos fica expresso por:

$$\mathcal{R}_t = A\mathcal{R}_{t-1} + U_t - BU_{t-1} \quad (7.2)$$

Com $\mathcal{R}_t, \mathcal{R}_{t-1}, U_t$ e U_{t-1} sendo vetores de dimensão (11×1) e A e B uma matriz de dimensão (11×11)

Os erros da predição é dado a seguir na tabela .

(a) <i>Treino1</i>		
Date	MSE	RMSE
2022-07-12	0.000095	0.009748
2022-07-13	0.000045	0.006700
2022-07-14	0.000121	0.011001
2022-07-15	0.000429	0.020720
2022-07-18	0.000153	0.012355

(b) <i>Treino2</i>		
Date	MSE	RMSE
2023-01-17	0.000038	0.006177
2023-01-18	0.000315	0.017744
2023-01-19	0.000122	0.011030
2023-01-20	0.000337	0.018363
2023-01-23	0.000158	0.012569

(c) <i>Treino3</i>		
Date	MSE	RMSE
2023-09-25	0.000021	0.004573
2023-09-26	0.000253	0.015913
2023-09-27	0.000100	0.009994
2023-09-28	0.000081	0.008994
2023-09-29	0.000069	0.008317

Tabela 7.4: Resultados do modelo para diferentes datas de treinamento.

7.3 MLP

O modelo Multi-Layer Perceptron foi implementado usando a biblioteca *Keras* (CHOLLET *et al.*, 2015).

Como modelos de redes neurais não possuem um algoritmo definido para escolha dos parâmetros, como é o caso dos critérios de informação dos modelos VAR e VARMA, a abordagem utilizada é testar alguns parâmetros arbitrários e comparar os resultados da métrica de erro escolhido, neste caso, Erro Quadrático Médio (MSE) e Raiz do Erro Quadrático Médio (RMSE).

Foi utilizado uma rede neural do tipo Multi-Layer Perceptron com 3 camadas ocultas, com 30, 100 e 30 ativações em cada, respectivamente, utilizando a função ReLU.

(a) Treino 1 usando 1 dia

Date	MSE	RMSE
2022-07-12	0.000209	0.014462
2022-07-13	0.000077	0.008799
2022-07-14	0.000214	0.014628
2022-07-15	0.000080	0.008937
2022-07-18	0.000140	0.011814

(d) Treino 1 usando 5 dias

Date	MSE	RMSE
2022-07-12	0.001213	0.034827
2022-07-13	0.000313	0.017687
2022-07-14	0.000524	0.022884
2022-07-15	0.000637	0.025246
2022-07-18	0.000268	0.016373

(b) Treino 2 usando 1 dia

Date	MSE	RMSE
2023-01-17	0.000150	0.012230
2023-01-18	0.000154	0.012410
2023-01-19	0.000075	0.008672
2023-01-20	0.000117	0.010804
2023-01-23	0.000298	0.017271

(e) Treino 2 usando 5 dias

Date	MSE	RMSE
2023-01-17	0.000092	0.009589
2023-01-18	0.000345	0.018581
2023-01-19	0.000088	0.009371
2023-01-20	0.000308	0.017558
2023-01-23	0.000039	0.006263

(c) Treino 3 usando 1 dia

Date	MSE	RMSE
2023-09-25	0.000027	0.005221
2023-09-26	0.000266	0.016313
2023-09-27	0.000119	0.010895
2023-09-28	0.000069	0.008335
2023-09-29	0.000098	0.009924

(f) Treino 3 usando 5 dias

Date	MSE	RMSE
2023-09-25	0.000081	0.009019
2023-09-26	0.000053	0.007284
2023-09-27	0.000077	0.008776
2023-09-28	0.000183	0.013541
2023-09-29	0.000072	0.008483

7.4 Erros finais

Além dos erros médios por dia de previsão, é possível fazer por setor e médio total. Para isso, resgatemos a equação 5.3, isto é, para cada setor, soma 1 a todos os dias e multiplica todos, subtraindo 1 depois.

Considera-se a última janela de treinamento utilizada para cada modelo. Além disso, faz-se também a média desses erros, a fim de ter um escalar como métrica para cada modelo.

Sector	VAR(1)	VARMA(1,1)	MLP 1 dia	MLP 5 dias
Technology	0.000274	0.000502	0.002231	0.021428
Communication Services	0.007297	0.007528	0.002364	0.012190
Consumer Cyclical	0.000272	0.000174	0.003857	0.025443
Financial Services	0.020383	0.020695	0.016922	0.000198
Healthcare	0.011043	0.010949	0.009599	0.003771
Energy	0.014529	0.013961	0.007648	0.025666
Consumer Defensive	0.017096	0.017106	0.017373	0.015085
Basic Materials	0.005065	0.005356	0.003539	0.016515
Industrials	0.008938	0.009174	0.008189	0.008369
Utilities	0.061542	0.061500	0.061500	0.052960
Real Estate	0.015513	0.015713	0.011480	0.002176

Tabela 7.5: Comparação da RMSE para cada um dos modelos para diferentes setores.

Sector	VAR	VARMA	MLP 1 dia	MLP 5 dias
Average Value	0.014723	0.014787	0.013155	0.016709

Tabela 7.6: Média do RMSE por modelo

Note que aparenta ser um número pequeno, portanto uma boa aproximação, contudo a afirmação não é ao todo verdadeira, pois pela natureza dos dados, um retorno médio simples do setor, é esperado que seja pequeno.

Abaixo o retorno acumulado, pela fórmula da agregação temporal mostrada anteriormente, do conjunto de teste nesses 5 dias.

Sector	Retorno Acumulado
Technology	0.001354
Communication Services	-0.004807
Consumer Cyclical	0.001877
Financial Services	-0.018550
Healthcare	-0.010171
Energy	0.014850
Consumer Defensive	-0.017566
Basic Materials	-0.004277
Industrials	-0.007858
Utilities	-0.060948
Real Estate	-0.013772

Tabela 7.7: Retorno acumulado por setor.

Agora é possível ver que, comparando os números da tabela 7.7 com o erro médio de 7.6, a predição não está bom o suficiente.

Capítulo 8

Conclusão

A conclusão deste estudo revela aspectos fundamentais da análise de séries temporais aplicada à previsão de comportamentos de setores econômicos, utilizando tanto métodos clássicos quanto técnicas modernas de aprendizado de máquina. Através da aplicação dos modelos Vetores Auto-Regressivos (VAR), Vetores Auto-Regressivos com Médias Móveis (VARMA) e Redes Neurais Multicamadas (MLP), este trabalho buscou não apenas explorar a eficácia dessas abordagens na modelagem de dados econômicos, mas também oferecer uma comparação criteriosa entre elas.

Os resultados obtidos indicam que, apesar do potencial teórico de cada metodologia, a previsão precisa do comportamento de mercados financeiros e econômicos continua sendo uma tarefa desafiadora. A minimização dos critérios apenas na ordem 1 para os modelos VAR e VARMA, considerada a mais básica, juntamente com resultados semelhantes obtidos através de redes neurais, sugere limitações significativas dessas técnicas quando aplicadas isoladamente ou sem ajustes refinados às peculiaridades dos dados econômicos.

Este cenário reforça a complexidade inerente aos mercados financeiros, caracterizados por sua volatilidade, dependência de fatores externos incontáveis e a natureza frequentemente não linear de suas dinâmicas. A similaridade nos resultados entre métodos clássicos e modernos de aprendizado de máquina ressalta ainda mais a necessidade de abordagens híbridas ou o desenvolvimento de novas técnicas que possam melhor capturar e prever as complexidades dos dados econômicos.

Além disso, a análise destacou a importância de um tratamento cuidadoso dos dados, desde a coleta até a transformação, para adequar-se aos modelos tradicionais e de aprendizado de máquina. A experiência reitera que a qualidade da previsão depende não apenas da escolha do modelo, mas também do pré-processamento dos dados, da capacidade de incorporar conhecimento de domínio relevante. Note que, por exemplo, VAR e VARMA necessitam de que as séries sejam estacionárias, enquanto redes neurais pede uma remodelagem específica dos dados.

Este trabalho, portanto, não conclui com uma solução definitiva para a previsão de séries temporais econômicas, mas abre caminho para investigações futuras.

Encoraja-se a exploração de modelos mais complexos, como redes neurais profundas

e algoritmos de aprendizado de máquina que podem lidar com grande dimensionalidade e dependências de longo prazo, modelos que modelam a volatilidade dos preços, como a família de modelos GARCH, além de famílias de modelos baseados em árvores ou teoria do controle, modelos de redes neurais mais sofisticados como redes neurais recorrentes (RNN), redes neurais LSTM, transformers, entre outros. Também é válido agregar dados de diversas fontes além do mercado de ações, com diferentes períodos, desde dados mensais a dados atualizados a cada minuto.

A integração de conhecimento de domínio específico, através da colaboração interdisciplinar, pode fornecer novas direções para aprimorar a modelagem de séries temporais econômicas.

Em suma, este estudo ressalta a jornada contínua em busca de modelos mais eficazes e robustos para a previsão econômica. Através da experimentação e comparação detalhada de diferentes técnicas, este trabalho contribui para o corpo de conhecimento existente, pavimentando o caminho para futuras pesquisas que irão, eventualmente, superar as barreiras atuais na análise de séries temporais econômicas.

Apêndice A

Código de extração dos dados

Código para extrair a lista de símbolos das empresas listadas na S&P

Programa A.1 Extração dos 500 símbolos da S&P

```

1  #Python 3.11.3
2  import pandas as pd # versão 1.5.3
3  def obter_constituientes_sp500():
4      #obtem lista de símbolos do Wikipedia
5      url = 'https://en.wikipedia.org/wiki/List_of_S%26P_500_companies'
6      tabela = pd.read_html(url, header=0)[0]
7      nomes = tabela['Symbol'].tolist()
8      return nomes

```

Com essa lista obtém-se os setores e o valor de mercado com auxílio da biblioteca *yfinance* (AROUSSI, 2023) e *yahoo_fin* (TREADWAY, 2021):

Programa A.2 Extração do valor de mercado das 500 ações da S&P

```

1  from yahoo_fin import stock_info as si #versão 0.8.9.1
2  import yfinance as yf #versão 0.2.32
3  def get_info_all_tickers():
4      #Obtem lista dos símbolos da S&P
5      sp500_symbols = obter_constituientes_sp500()
6      data_dict = {}
7      # Itere através dos símbolos do S&P 500 e obtenha o valor de mercado e
        setor
8      for symbol in sp500_symbols:
9          try:
10             quote_table = si.get_quote_table(symbol)
11             market_cap = quote_table['Market Cap']
12             sector = yf.Ticker(symbol).info['sector']
13             data_dict[symbol] = {'Market Cap': market_cap, 'Sector': sector}
14         except:
15             pass
16     return data_dict

```

cont →

→ *cont*

```
17 data_dict = get_info_all_tickers()
```

Após isso, extraí as 10 maiores empresas, do ponto de vista de valor de mercado, por cada setor

Programa A.3 Extraí top 10 maiores ações em valor de mercado de cada setor

```
1  from collections import defaultdict
2  import itertools
3  MULT = {'K':1e3, 'M':1e6, 'B':1e9, 'T':1e12}
4
5  #data_dict éo dicionário retornado de get_info_all_tickers
6  def get_topN_each(data_dict,N=10):
7      sector_dict = defaultdict(list)
8      top_N_values, top_N_names = {}, {}
9      # Remove os multiplicadores K, M, B e T e ordena do maior para o menor
10     sorted_data = sorted(data_dict.items(), key=lambda x: float(x[1]['Market
        Cap'][:-1])*MULT[x[1]['Market Cap'][:-1]] if x[1]['Market Cap'][:-1] in
        ['K', 'M', 'B', 'T'] else float(x[1]['Market Cap']), reverse=True)
11
12     # No conjunto ordenado separa por setor
13     for symbol, company_data in sorted_data:
14         sector = company_data['Sector']
15         market_cap_aux = company_data['Market Cap']
16         market_cap = float(market_cap_aux[:-1]) * MULT[market_cap_aux[:-1]] if
            market_cap_aux[:-1] in MULT else float(market_cap_aux)
17         sector_dict[sector].append((symbol, market_cap))
18
19     # Extraí top N de cada setor
20     for sector, companies in sector_dict.items():
21         info = sorted(companies, key=lambda x: x[1], reverse=True)[:N]
22         names = [k[0] for k in info]
23         top_N_values[sector] = info
24         top_N_names[sector] = names
25
26     lista_N_names = list(itertools.chain.from_iterable(top_N_names.values()))
27     dict_nome_por_setor = {setor: [ticker for ticker, _ in empresas] for setor
        , empresas in top_N_values.items()}
28     return top_N_values, lista_N_names, dict_nome_por_setor
29
30 top_N_values, lista_N_names, dict_nome_por_setor = get_topN_each(data_dict)
```

Programa A.4 Download dos dados para dar início ao trabalho

```
1  date_init = '2021-12-31'
2  date_end = '2023-09-30'
3  precos_all = {}
4  #extraí preços de cada ação agrupado por setor
5  for setor in dict_nome_por_setor.keys():
```

cont →

```
→ cont
6     precos_all[setor] = yf.download(dict_nome_por_setor[setor],date_init,
7                                     date_end, auto_adjust=True)['Close']
8
9     df_teste = pd.DataFrame()
10    # extrai o índice temporal de uma das séries para auxílio
11    df_teste.index = precos_all['Technology'].index
12    for setor in dict_nome_por_setor.keys():
13        #calcula o retorno líquido simples e obtém a média por setor
14        df_teste[setor] = precos_all[setor].pct_change().mean(axis=1)
15    retorno_setores = df_teste.dropna()
```

Apêndice B

Código de implementação dos modelos

Programa B.1 Divisão das janelas de treino e teste

```

1  treino1, teste1 = retorno_setores[:130], retorno_setores[130:135]
2  treino2, teste2 = retorno_setores[:260], retorno_setores[260:265]
3  treino3, teste3 = retorno_setores[:5], retorno_setores[5:]

```

B.1 VAR

Abaixo o código para a escolha da ordem p do modelo VAR(p), o qual executa em um laço, variando p de 1 a 10 e extrai os critérios de informação AIC, BIC e HQ:

Programa B.2 Verificação de critérios de informação para escolha da ordem de VAR(p)

```

1  import pandas as pd
2  import numpy as np
3  from statsmodels.tsa.api import VAR
4  def criterios_var(treino, p_max=10):
5      dict_criterio_var = {}
6      #loop para analisar os criterios de informação e escolher a ordem do
        modelo
7      for ordem in range(1, p_max+1):
8          model_var = VAR(treino.values)
9          fitted_model = model_var.fit(ordem, trend='n')
10         dict_criterio_var[ordem] = {'AIC':fitted_model.aic, 'BIC':
            fitted_model.bic, 'HQ':fitted_model.hqic}
11     return pd.DataFrame(dict_criterio_var).T.round(2)
12 #criterios para as 3 janelas de dados
13 print(criterios_var(treino1))
14 print(criterios_var(treino2))
15 print(criterios_var(treino3))

```

No caso da implementação deste trabalho que obteve ordem 1 após rodar e escolher o mínimo do código acima, executou-se o modelo para ordem $p = 1$, obtendo os coeficientes do modelo:

Programa B.3 Treino e predição do modelo VAR(1)

```

1  def var_predicao(treino, ordem, indices_datas, n_predicao=5):
2      model = VAR(treino.values)
3      fitted_model = model.fit(ordem, trend='n')
4      predicao = fitted_model.forecast(y=treino.values[-ordem:], steps=
        n_predicao)
5      predicao_df = pd.DataFrame(predicao, index=indices_datas, columns=treino.
        columns)
6      return predicao_df
7
8  def var_EQM(predicao, teste):
9      EQM = ((teste - predicao)**2).mean(axis=1)
10     REQM = np.sqrt(((teste - predicao)**2).mean(axis=1))
11     return pd.DataFrame([EQM,REQM], index=['MSE', "RMSE"]).T
12
13     teste_datas1 = teste1.index
14     var_predicao_df1 = var_predicao(treino1, 1, teste_datas1)
15     print(var_EQM(var_predicao_df1, teste1))
16
17     teste_datas2 = teste2.index
18     var_predicao_df2 = var_predicao(treino2, 1, teste_datas2)
19     print(var_EQM(var_predicao_df2, teste2))
20
21     teste_datas3 = teste3.index
22     var_predicao_df3 = var_predicao(treino3, 1, teste_datas3)
23     print(var_EQM(var_predicao_df3, teste3))

```

B.2 VARMA

Abaixo o código utilizado para a escolha dos parametros p e q

Programa B.4 Verificação de critérios de informação para escolha da ordem de VARMA(p,q)

```

1  import statsmodels.api as sm
2
3  def criterios_varma(treino, p_max=2, q_max=2):
4      dict_criterio_varma = {}
5      #loop para analisar os criterios de informação e escolher a ordem do
        modelo
6      lista_criterios = []
7      for ordem_p in range(1, p_max+1):
8          for ordem_q in range(1, q_max+1):
9              #VARMA é um caso especial do VARMAX com nenhuma variável X (exógena)

```

cont →

```

→ cont
10         model_varma = sm.tsa.VARMAX(treino.values, order=(ordem_p, ordem_q),
11                                     trend='n')
12         try:
13             fitted_model = model_varma.fit()
14             lista_criterios.append({'p':ordem_p,'q':ordem_q,'AIC':
15                                   fitted_model.aic, 'BIC': fitted_model.bic, 'HQ':fitted_model.
16                                   hqic})
17         except:pass
18     return pd.DataFrame(lista_criterios).round(2)
19
20 #Critério para cada janela de dados
21 criterios_varma1 = criterios_varma(treino1)
22 criterios_varma2 = criterios_varma(treino2)
23 criterios_varma3 = criterios_varma(treino3)
24 print(criterios_varma1)
25 print(criterios_varma2)
26 print(criterios_varma3)

```

Com a execução deste, obteve $p = 1$ e $q = 1$. Executa-se então o modelo dessa ordem nas 3 janelas de dados.

Programa B.5 Treino e predição do modelo VARMA(1,1)

```

1  def varma_predicao_func(treino, p,q, indices_dadas, n_predicao=5):
2      model = sm.tsa.VARMAX(treino.values, order=(p,q), trend='n')
3      fitted_model = model.fit(method_kwargs={'maxiter':200})
4      predicao = fitted_model.forecast(steps=n_predicao)
5      predicao_df = pd.DataFrame(predicao, index=indices_dadas, columns=treino.
6                                columns)
7      return predicao_df
8
9  def varma_EQM(predicao, teste):
10     EQM = ((teste - predicao)**2).mean(axis=1)
11     REQM = np.sqrt(EQM)
12     return pd.DataFrame([EQM,REQM], index=['MSE', "RMSE"]).T
13
14 #VARMA(1,1) para a primeira janela de dados de treinamento
15 varma_predicao1 = varma_predicao_func(treino1, 1,1, teste1.index)
16 print(varma_EQM(varma_predicao1, teste1))
17
18 #VARMA(1,1) para a segunda janela de dados de treinamento
19 varma_predicao2 = varma_predicao_func(treino2, 1,1, teste2.index)
20 print(varma_EQM(varma_predicao2, teste2))
21
22 #VARMA(1,1) para a terceira janela de dados de treinamento
23 varma_predicao3 = varma_predicao_func(treino3, 1,1, teste3.index)
24 print(varma_EQM(varma_predicao3, teste3))

```

B.3 MLP

os códigos abaixo foram adaptados de [BROWNLEE \(2018, p. 84\)](#)

Programa B.6 Função para transformar no tipo X e Y

```

1  from numpy import array
2
3  def split_sequences(sequences, n_steps_in, n_steps_out):
4      X, y = list(), list()
5      for i in range(len(sequences)):
6          # find the end of this pattern
7          end_ix = i + n_steps_in
8          out_end_ix = end_ix + n_steps_out
9          # check if we are beyond the dataset
10         if out_end_ix > len(sequences):
11             break
12         # gather input and output parts of the pattern
13         seq_x, seq_y = sequences[i:end_ix, :], sequences[end_ix:out_end_ix, :]
14         X.append(seq_x)
15         y.append(seq_y)
16     return array(X), array(y)

```

Implementa o modelo MLP:

Programa B.7 Implementação do MLP com uso de 1 dia anterior e com 5 dias anteriores, modelo com 3 camadas ocultas, com 30, 100 e 30 ativações cada

```

1  from keras.models import Sequential
2  from keras.layers import Dense
3  # multivariate multi-step data preparation
4  def MLP_treina_predicao(treino, teste, n_lags, n_predicao):
5      # K.clear_session()
6      # converte em X e Y
7      X_treino, y_treino = split_sequences(treino.values.astype('float32'),
8                                          n_lags, n_predicao)
9      # converte para estrutura [linhas, colunas] para o conjunto de treino
10     n_input = X_treino.shape[1] * X_treino.shape[2]
11     X_treino = X_treino.reshape((X_treino.shape[0], n_input))
12     n_output = y_treino.shape[1] * y_treino.shape[2]
13     y_treino = y_treino.reshape((y_treino.shape[0], n_output))
14
15     # Construir e treinar o modelo
16     ## camada com 30, 100 e 30 ativações respectivamente
17     model = Sequential()
18     model.add(Dense(30, activation='relu', input_dim=n_input))
19     model.add(Dense(100, activation='relu'))
20     model.add(Dense(30, activation='relu'))
21     model.add(Dense(n_output))
22     model.compile(optimizer='adam', loss='mse')
23     model.fit(X_treino, y_treino, epochs=300, verbose=0)

```

cont →

→ *cont*

```

24     #predição
25     teste_modificado = treino[-n_lags:].values.reshape((1, n_input))
26     yhat = model.predict(teste_modificado, verbose=0)
27     return pd.DataFrame(yhat.reshape((5,11)), index=teste.index, columns=
        teste.columns)
28
29     def MLP_EQM(predicao, teste):
30         EQM = ((teste - predicao)**2).mean(axis=1)
31         REQM = np.sqrt(EQM)
32         return pd.DataFrame([EQM,REQM], index=['MSE', "RMSE"]).T
33
34     np.random.seed(71)
35     #conjunto de treino 1, usa 1 dia anterior para prever 5
36     predicao_MLP1_1 = MLP_treina_predicao(treino1, teste1, 1,5)
37     print(MLP_EQM(predicao_MLP1_1, teste1))
38
39     #conjunto de treino 2, usa 1 dia anterior para prever 5
40     predicao_MLP2_1 = MLP_treina_predicao(treino2, teste2, 1,5)
41     print(MLP_EQM(predicao_MLP2_1, teste2))
42
43     #conjunto de treino 3, usa 1 dia anterior para prever 5
44     predicao_MLP3_1 = MLP_treina_predicao(treino3, teste3, 1,5)
45     print(MLP_EQM(predicao_MLP3_1, teste3))
46
47     #conjunto de treino 1, usa 5 dias anterior para prever 5
48     predicao_MLP1_5 = MLP_treina_predicao(treino1, teste1, 5,5)
49     print(MLP_EQM(predicao_MLP1_5, teste1))
50
51     #conjunto de treino 2, usa 5 dias anterior para prever 5
52     predicao_MLP2_5 = MLP_treina_predicao(treino2, teste2, 5,5)
53     print(MLP_EQM(predicao_MLP2_5, teste2))
54
55     #conjunto de treino 3, usa 5 dias anterior para prever 5
56     predicao_MLP3_5 = MLP_treina_predicao(treino3, teste3, 5,5)
57     print(MLP_EQM(predicao_MLP3_5, teste3))

```

Referências

- [AROUSSI 2023] Ran AROUSSI. *yfinance*. Último acesso em Dezembro de 2023. Licença Apache 2.0. 2023. URL: <https://github.com/ranaroussi/yfinance> (citado nas pgs. 25, 53).
- [AUFFARTH 2021] Ben AUFFARTH. *Machine Learning for Time-Series with Python: Forecast, predict, and detect anomalies with state-of-the-art machine learning methods*. Packt Publishing Ltd, 2021 (citado nas pgs. viii, 39).
- [BERGMEIR *et al.* 2018] Christoph BERGMEIR, Rob J HYNDMAN e Bonsoo KOO. “A note on the validity of cross-validation for evaluating autoregressive time series prediction”. *Computational Statistics & Data Analysis* 120 (2018), pp. 70–83 (citado na pg. 41).
- [BROWNLIE 2017] Jason BROWNLIE. *Introduction to time series forecasting with python: how to prepare data and develop models to predict the future*. Machine Learning Mastery, 2017 (citado nas pgs. 38, 43).
- [BROWNLIE 2018] Jason BROWNLIE. *Deep learning for time series forecasting: predict the future with MLPs, CNNs and LSTMs in Python*. Machine Learning Mastery, 2018 (citado nas pgs. ix, 34, 36, 59).
- [CHAN e MÁTYÁS 2022] Felix CHAN e László MÁTYÁS. *Econometrics with machine learning*. Springer, 2022 (citado nas pgs. viii, 42).
- [CHOLLET *et al.* 2015] François CHOLLET *et al.* *Keras*. 2015. URL: <https://keras.io> (citado na pg. 48).
- [CHOLLET 2018] François CHOLLET. “Deep learning with python” (2018) (citado nas pgs. 18, 19, 21).
- [DE PRADO 2018] Marcos Lopez DE PRADO. *Advances in financial machine learning*. John Wiley & Sons, 2018 (citado na pg. 41).
- [S&P GLOBAL 2023] Standard & Poor’s GLOBAL. *GICS: Global Industry Classification Standard*. 2023. URL: <https://www.spglobal.com/spdji/en/landing/topic/gics/> (citado nas pgs. 3, 4).

- [S&P 500 2023] Standard & Poor's GLOBAL. *S&P Global*. 2023. URL: <https://www.spglobal.com/spdji/en/indices/equity/sp-500/#overview> (citado na pg. 3).
- [GREG BLAND 2020] GREG BLAND. *Training, Validation, and Test Data Set Illustration*. Último acesso em Dezembro de 2023. 2020. URL: <https://algotrading101.com/learn/wp-content/uploads/2020/06/training-validation-test-data-set.png> (citado nas pgs. vii, 39).
- [HELMUT 2005] Lütkepohl HELMUT. *New introduction to multiple time series analysis*. Springer, 2005 (citado nas pgs. 15–17).
- [MSCI 2023] MSCI INC. *The Global Industry Classification Standard (GICS®)*. 2023. URL: <https://www.msci.com/our-solutions/indexes/gics> (citado na pg. 3).
- [S&P DOW JONES INDICES 2023] Standard & Poor's Dow Jones INDICES. *About us | S&P Dow Jones Indices*. 2023. URL: <https://www.spglobal.com/spdji/pt/about-us/> (citado na pg. 3).
- [JAMES *et al.* 2023] Gareth JAMES, Daniela WITTEN, Trevor HASTIE, Robert TIBSHIRANI e Jonathan TAYLOR. *An introduction to statistical learning: With applications in python*. Springer Nature, 2023 (citado nas pgs. 20, 22–24, 38, 40).
- [KIRCHGÄSSNER *et al.* 2012] Gebhard KIRCHGÄSSNER, Jürgen WOLTERS e Uwe HASSLER. *Introduction to modern time series analysis*. Springer Science & Business Media, 2012 (citado nas pgs. 9–11, 13, 14, 16).
- [LAZZERI 2020] Francesca LAZZERI. *Machine learning for time series forecasting with Python*. John Wiley & Sons, 2020 (citado nas pgs. 21, 25, 32, 33).
- [MORETTIN 2017] Pedro A MORETTIN. *Econometria financeira: um curso em séries temporais financeiras*. Editora Blucher, 2017 (citado nas pgs. 7, 8, 12, 17, 26, 32).
- [MORETTIN e TOLOI 2018] Pedro A MORETTIN e Clélia MC TOLOI. *Análise de séries temporais: modelos lineares univariados*. Editora Blucher, 2018 (citado na pg. 6).
- [PEDREGOSA *et al.* 2011] PEDREGOSA, F. AND VAROQUAUX, G. AND GRAMFORT, A. AND MICHEL, V. AND THIRION, B. AND GRISEL, O. AND BLONDEL, M. AND PRETTENHOFER, P. AND WEISS, R. AND DUBOURG, V. AND VANDERPLAS, J. AND PASSOS, A. AND COURNAPEAU, D. AND BRUCHER, M. AND PERROT, M. AND DUCHESNAY, E. “Scikit-learn: machine learning in Python”. *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. URL: https://scikit-learn.org/stable/modules/cross_validation.html (citado nas pgs. viii, 40).
- [SEABOLD e PERKTOLD 2010] Skipper SEABOLD e Josef PERKTOLD. “Statsmodels: econometric and statistical modeling with python”. In: *9th Python in Science Conference*. 2010 (citado nas pgs. 44, 46).

REFERÊNCIAS

- [TREADWAY 2021] Andrew TREADWAY. *yahoo fin*. Último acesso em Dezembro de 2023. Licença MIT. 2021. URL: https://theautomatic.net/yahoo_fin-documentation/ (citado nas pgs. 25, 53).
- [YAHOO FINANCE 2023] YAHOO FINANCE. *Yahoo Finance*. Copyright © Yahoo Inc. 2021, All Rights Reserved. 2023. URL: <https://finance.yahoo.com/> (citado na pg. 25).
- [ZIVOT e WANG 2006] Eric ZIVOT e Jiahui WANG. *Modeling financial time series with S-PLUS*. Vol. 2. Springer, 2006 (citado nas pgs. 8, 13).