UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE MATEMÁTICA

ISCTE
INSTITUTO UNIVERSITÁRIO DE LISBOA
DEPARTAMENTO DE FINANÇAS

Econometria Vs. Machine Learning: Big Data em Finanças

Amadú Baldé

Mestrado em Matemática Financeira

Dissertação orientada por:
Professora Doutora Diana Aldea Mendes

2020

Agradecimentos

Agradeço primeiro à Deus por esta bênção e todas as outras que tem proporcionado na minha vida, agradeço por toda a força e esperança que me tem dado ao longo desta jornada.

Ao meu pai, meu grande herói e alicerce não existem palavras e gestos suficientes para agradecê-lo por toda a dedicação e esforço que fez para proporcionar-me a educação e a vida de que hoje usufruo.

A minha mãe, minha rainha, palavras de agradecimento não são suficientes por todos os sacrifícios e conselhos, a si devo a minha vida.

Aos meus irmãos, o meu profundo e eterno agradecimento por sempre protegerem-me e aconselharem-me a seguir bons caminhos.

Aos meus familiares mais próximos um grande obrigado por se fazerem presentes.

Aos meus verdadeiros amigos um grande obrigado pela alegria que me proporcionam, pelo ombro amigo e principalmente por mostrarem-me que a diversidade, respeito e amor ao próximo é que fazem deste mundo um mundo melhor.

A minha orientadora, o meu profundo agradecimento pela dedicação, disponibilidade e todo o conhecimento que me proporcionou.

Resumo

A previsão dos preços dos índices bolsistas é uma das mais desafiadoras, complexas e fascinantes tarefas, uma vez que os conjuntos de dados onde estes se inserem, chamadas séries temporais, apresentam várias irregularidades (ruído, não-estacionariedade, não linearidades, entre outras). Vários têm sidos os estudos feitos ao longo dos anos com vista a encontrar técnicas mais eficazes, que sejam capazes de contornar essas irregularidades.

Com o crescimento exponencial dos dados e a não homogeneidade dos mesmos, torna-se cada vez mais difícil a verificação dos pressupostos nos modelos econométricos.

Tendo em conta os presentes desafios, a presente dissertação terá como principal objetivo comparar os métodos clássicos de econometria com os novos métodos de *machine learning*, para tal ir-se-á recorrer aos dados do índice bolsista *S&P 500*, no qual pretende-se prever no final os preços de fecho da série.

Numa primeira fase, com vista a uma melhor compreensão das temáticas que serão abordadas faz-se uma contextualização sustentada na literatura científica e num conjunto de conceitos considerados essenciais para a compreensão dos temas abordados.

Numa segunda fase, prossegue-se com o estudo empírico, onde ir-se-á analisar as estatísticas descritivas, os gráficos, os pressupostos dos modelos e depois escolhidos os potenciais modelos. Este capítulo será divido em dois subcapítulos. No primeiro subcapítulo o estudo será feito sob a alçada do programa estatístico *Eviews* onde serão abordadas as técnicas clássicas da econometria. No segundo subcapítulo o estudo será feito no *software Python*, considerado atualmente um dos *softwares* mais populares no mundo científico, académico e empresarial.

No *Eviews*, uma vez obtida a estacionariedade da série procede-se com a modelização através da metodologia de Box-Jenkins, mais especificamente o modelo Autorregressivo Integrado de Médias Móveis – *ARIMA*. Uma vez escolhido o modelo, procede-se com a previsão dos preços de fecho da série. Por outro lado, no *Python*, serão abordadas vertentes mais inovadoras, sendo uma delas a aplicação das *feature engineering* que resultarão em trinta e uma (31) novas variáveis. Ao contrário dos modelos clássicos, os modelos obtidos pelos algoritmos de *machine learning* não necessitam da verificação dos pressupostos habituais econométricos, uma vez que a máquina aprende de forma "autónoma" a contornar certas irregularidades. Os algoritmos utilizados serão o de Regressão Linear/*Linear Regression* (LR), *Suport Vector Regression* (SVR) e *Random Forest* (RF).

Por fim, é feita uma interpretação critica dos resultados obtidos ao longo de todo o estudo e comparam-se os resultados, atingindo assim o objetivo inicialmente delineado para a dissertação.

Palavras chaves: Big Data, Machine Learning**,** Série temporal; Índice Bolsista; Standard & Poor's 500, ARIMA/ARMA, Previsão.

Abstract

Forecasting the prices of stock market indexes is one of the most challenging, complex and fascinating tasks, since the data sets where they are inserted, called time series, exhibit various irregularities (noise, non-stationarity, non-linearity, among others). Several studies have been carried out over the years with a view to finding more effective techniques that are capable to work around these irregularities.

With the exponential growth of the data and the heterogeneity, it becomes more and more difficult to verify the assumptions in the econometric models.

Taking into account the present challenges, this dissertation will have as main objective to compare the classic econometrics methods with the new machine learning algorithms, and for this we will use the data of the *S&P 500* stock index, from which it is intended to predict at the end the closing prices of the series.

In a first phase, with a view to a better understanding of the themes that will be approached, a contextualization based on scientific literature and on a set of concepts considered essential for the comprehension of the topics covered is made.

In a second phase, we proceed with the empirical study, where we will analyze the descriptive statistics, the graphs, the assumptions of the models and then the potential models will be chosen. This chapter will be divided into two sub-chapters. In the first sub-chapter, the study will be carried out under the statistical program Eviews, where the classical econometrics techniques will be approached. In the second sub-chapter the study will be done in Python software, currently considered one of the most popular software in the scientific, academic and business world.

In Eviews, once the time series is stationary, it is proceeded with the modeling through the Box-Jenkins methodology, more specifically the Integrated Autoregressive Moving Average model - *ARIMA*. After establishing the final model, the closing prices for the S&P 500 series are forecasted. On the other hand, in Python, more innovative aspects will be addressed, one of which is the application of feature engineering that will result in thirty-one (31) new variables. Unlike the classic models, the algorithms obtained from machine learning do not need to check the usual econometric assumptions, since the machine learns "autonomously" to work around certain irregularities. The algorithms used in this dissertation are the following: *Linear Regression (LR)*, *Support Vector Regression (SVR)* and *Random Forest (RF)*.

Finally, a critical interpretation of the obtained results it is made and the results are compared, thus reaching the objective initially outlined for the dissertation.

Key words: *Big Data, Machine Learning, Time series; Stock Index; Standard & Poor's 500, ARIMA / ARMA, Forecast.*

Índice

Índice de Figuras

Índice de Tabelas

Abreviaturas/Acrónimos

ML → *Machine Learning*

AI → *Artificial Intelligence*

LR → *Linear Regressor*

SVR → *Support Vector Regressor*

RF → *Random Forest*

J-B → *Jarque-Bera*

B-G → Bresch-Godfrey Serial Correlation

KPSS → *Kwiatkowski-Phillips-Schmidt-Shin*

PP → *Phillips-Perron*

DF → *Dickey-Fuller*

ADF → *Augmented Dickey–Fuller*

ACF → *Autocorrelation Coefficient Function*

PACF → *Partial Autocorrelation Coefficient Function*

B-J → *Box-Jenkins*

JB → *Jarque-Bera*

AR → *Autoregressivo*

MA → *Média Móvel*

ARIMA → Autorregressivo Integrado e de Média Móvel

ARMA → Autorregressivo e de Média Móvel

AIC → *Akaike Information Criterion*

SBIC → *Schwarz Bayesiana Information Criterion*

HQIC → *Hannan-Quinn Information Criterion*

DW → *Durbin Watson*

ME → *Mean Error*

MAE → *Mean Absolut Error*

MAPE → *Mean Percentual Error*

MSE → *Mean Square Error*

RMSE → *Root Mean Square Error*

Introdução

Tradicionalmente, os investidores utilizavam diferentes modelos econométricos para a previsão a longo e curto prazo, sendo os modelos autorregressivos os mais usados - *AR, MA, ARIMA/ARMA*. Mas, tendo em conta algumas limitações destes modelos e a crescente quantidade de dados derivados do "*big data*", desenvolveram-se novas tecnologias que abriram portas para novos métodos e modelos através da introdução de algoritmos de Inteligência Artificial (*AI*), nomeadamente os algoritmos de *Machine Learning* (*ML*).

Partindo desta explanação, o presente trabalho levanta as seguintes questões: Conseguem os modelos clássico de econometria responder as exigências atuais e futuras dos mercados financeiros? Serão os métodos de previsão em *Machine Learning* mais eficazes do que dos métodos clássicos?

Com base nestas questões, este trabalho irá abordar de forma critica e científica os conceitos teóricos e práticos relativamente aos principais conceitos de econometria e de *Machine Learning* com uso direto na elaboração desta tese.

Portanto, esta dissertação terá como objetivo estudar e comparar, de forma cuidadosa, os modelos autorregressivos com alguns algoritmos de *Machine Learning* – nomeadamente: *Linear Regression*, *Support Vector Regression* e *Random Forest.*

Este estudo foi motivado pelos vários desafios que os avanços tecnológicos colocam e pela crescente necessidade de adaptação e inovação dos métodos clássicos as novas procuras e necessidades do mercado.

Segundo o relatório elaborado pela *Intelligence Unit Limited (2013)* as organizações têm vindo cada vez mais a integrar os dados no processo de tomada de decisões, estando estas fortemente focadas nas oportunidades e desafios apresentados pelo "*big data*", reconhecendo que o seu uso pode levá-las a obter vantagens e ganhos significativos relativamente a concorrência.

No que toca a estrutura desta dissertação, tem-se numa primeira parte a revisão da literatura, o enquadramento teórico de alguns conceitos econométricos e também dos conceitos de *big data*, *machine learning* e assim como as suas implementações.

Numa segunda fase, é feita a abordagem teórica dos principais conceitos de séries temporais. Conceitos esses como a estacionariedade, testes de raiz unitária, modelos de previsão lineares e critérios de informação.

Na terceira parte aborda-se os principais conceitos de *Machine Leaning*, onde é feita uma abordagem aos diferentes tipos de aprendizagem e algoritmos de previsão.

A quarta parte é dedicada a implementação dos conceitos teóricos, ou seja, o estudo empírico. A série temporal financeira abordada é proveniente do índice bolsista *S&P 500*, datada desde 03/01/2000 até 03/04/2019. A escolha da série *S&P 500* vai ao encontro da verificação da precisão dos resultados obtidos, pois é uma das séries mais estudadas ao nível internacional. Este capítulo foi dividido em duas partes. Sendo a primeira parte dedicada ao estudo dos modelos autorregressivos, considerados neste

estudo como métodos “clássicos”. Na segunda parte abordam-se os modelos *LR*, o *SVR* e o *RF* considerados como métodos “inovadores” de *Machine Learning*. De forma a medir a precisão das previsões dos diferentes modelos serão utilizadas medidas de perda de informação/erro de previsão - MSE (*Mean Square Error*), RMSE (*Root Mean Square Error*), MAPE (*Mean Absolute Prediction Error*).

Por fim, na quinta parte tem-se a conclusão, onde se explanam e comparam de forma critica as principais conclusões retiradas dos resultados obtidos através dos métodos abordados.

Espera-se que esta tese seja um pequeno contributo para a divulgação da importância de novos algoritmos e *softwares* poderosos para a análise e previsão de dados financeiros.

1. Revisão Da Literatura

Nos últimos anos o *Big Data* ou "grande volume" de dados, tem sido tema de discussão em diversas áreas, sendo este visto como uma "preocupação" ou "inovação" por parte de cientistas, governos, empresas e diversas outras entidades.

Embora não exista uma única definição (Torrecilla, et al., 2018), podemos considerar o *big data*, no contexto geral, como sendo um grande conjunto de dados que contém milhares ou milhões de observações geradas a alta frequência (Wells Fargo Securities Economics Group, 2017), constituindo assim um grande potencial de aquisição de conhecimento nos meios em que se inserem. Os métodos tradicionais revelam-se incapazes de lidar com esses dados, uma vez que estes são geralmente complexos, diversos e heterogéneos.

Na área das Finanças os dados são um ativo valioso, uma vez que neste campo são gerados e transacionados milhões de dados provenientes de diversas fontes (finanças corporativas, relatórios de análise, notícias, entre outras fontes). A pressão exercida sobre as entidades financeiras para que estas agissem de forma mais transparente (apresentação de relatórios, publicação de dados, limitação na atuação, etc.) e que utilizassem de forma mais eficiente os dados, fizeram com que: o volume de dados crescesse, a indústria se reinventasse através da inovação nos serviços e infraestruturas de forma a responder aos novos desafios.

Do ponto de vista da estatística clássica a grande quantidade de dados pode representar uma "bênção", uma vez que estar-se-ia perto da real população dos dados e da convergência assintótica dos modelos (Torrecilla, et al., 2018). Mas na verdade, o processo não é tão simples quanto parece. Segundo os mesmos autores, o grande desafio para os estatísticos é lidar com as diferentes formas de heterogeneidade inerentes ao *big data*, com a não homogeneidade das populações e a não normalidade das mesmas, o que faz com que as aproximações clássicas de estatística baseadas na convergência e no teorema do limite central sejam invalidas ou parcialmente invalidas. Os dados vindos de diferentes plataformas (base de dados, web, redes sociais, sensores, entre outros) podem ser completamente destruturados, levando ao aparecimento de diferentes problemas como a codificação própria das informações e a combinação de diferentes estruturas de dados (categóricos, contínuos, imagens, etc). O *big data* geralmente tende a incluir e também aumentar os problemas usuais de alta dimensionalidade.

Em 2015, Hassani & Silva analisaram o uso do *Big Data* para a previsão em diferentes áreas, identificando os problemas e desafios da sua implementação, as oportunidades do seu uso e as suas aplicações. No que toca aos problemas, estes identificaram que as ferramentas tradicionais de previsão não conseguiam lidar com o grande volume de dados, com a velocidade com que estes são gerados e com a complexidade dos mesmos. Em relação as oportunidades, estes afirmam que podem ser diversas, sendo uma delas nas previsões meteorológicas em que podem ser analisadas diversas variáveis de forma a reduzir o erro de previsão. As habilidades necessárias para lidar com os novos desafios impostos pelo *big data*, o sinal e o ruido dos dados, o *hardware* e o *software*, a arquitetura dos algoritmos, a significância estatística dos testes

e o próprio *big data* são apontados por estes como os principais desafios que impedem o processo de obtenção de boas previsões. Quanto à sua implementação e aplicabilidade, estes identificaram diversas oportunidades, nomeadamente nas políticas monetárias e também na área financeira.

No artigo escrito por Faraway & Augustin (2018), estes abordam a temática do *big data* de uma forma menos otimista, afirmando que por vezes é preferivel ter-se uma pequena amostra de dados do que uma grande amostra (*big data*). Justificando que a qualidade é sempre melhor do que a quantidade, os custos de armazenamento dos dados nem sempre justificam os beneficios e de que a inferência estatística produz melhores resultados em pequenas amostras.

Como consequência do *big data* e da interseção entre metodos estatistícos e computacionais surge o aprendizagem de máquina, mais conhecido como *machine learning* (*ML*). Este veio atona como um novo paradigma cientifico em diversas ciências, mas o seu uso na economia e em econometria encontra-se ainda bastante atrasado (Cerulli, 2020). Segundo o mesmo autor o objetivo principal do *ML* trata-se de transformar as informações disponiveis em um valioso conhecimento "deixando os dados falarem por si". Alguns economistas acreditam que o *ML* é bastante poderoso para previsões, mas pouco útil para a inferência (Athey, 2019).

Dingli & Founier (2017), abordaram várias técnicas recorrendo ao *machine learning* para prever o mercado de ações, destacando as vantagens e limitações das suas aplicações. Para o estudo empírico usaram dados históricos (ações de empresas tecnológicas e financeiras) tanto de longo prazo como de curto prazo, obtidos nomeadamente através da plataforma *Yahoo-finance*. Como forma de acompanhar a evolução das previsões eles organizaram os *outputs* da seguinte forma: Direção do próximo período, que constitui uma variável binária, que apenas indica "Up" ou "Down"; Mudança do preço no próximo período, que constitui uma variável numérica continua; Preço atual do próximo período, variável numérica. Todos esses *outputs* foram divididos em períodos variados (diários, semanais, mensais, trimestrais e anuais). No final das experimentações obtiveram 81% de precisão na previsão do futuro da indústria financeira, utilizando a regressão logística com periocidade anual, enquanto que na indústria tecnológica obtiveram 77% de precisão utilizando o método de *K-Neighbors Classifier* com dados anuais. No que diz respeito as técnicas de regressão, obtiveram 0.0117 RMSE para o preço do dia seguinte e 0.0613 RMSE para a mudança do preço no dia seguinte.

Em 2009, Lu *et al*., propuseram dois estágios para a elaboração de um modelo de previsão de séries temporais financeiras com vista a suprir as limitações dos métodos existentes e reduzir a influência do ruido. Os dois estágios consistiam na combinação do método da análise de componentes principais (ACP) com o algoritmo *Support Vector Regression* (*SVR*), este método foi intitulado por estes como o modelo *ICA-SVR* (*independent component analysis and support vector regression*). A análise de componentes principais foi usada como forma de detetar e remover o ruido das séries temporais financeiras, após esta ação as restantes componentes independentes foram usadas para reconstruir as variáveis para a previsão contendo menos ruido. Uma vez reconstruída as variáveis, essas são usadas pelo *support vector regression* para construir o modelo de previsão. Para a implementação das suas propostas, utilizaram o índice de

preços de abertura da Nikkei225[1] e os de fecho da TAIEX[2]. Na análise empírica, usaram para medir a performance dos algoritmos de previsão a raiz quadrada da média ao quadrado dos erros (*RMSE*), a média ao quadrado dos erros normalizada (*NMSE*), a média das diferenças absolutas (*MAD*), direção da simetria (*DS*), correta previsão de subida (*CP*) e correta previsão de descida (*CD*). Sendo as medidas *RMSE, NMSE, MAD* usadas para avaliar o erro de previsão e as restantes para medir a precisão. Os resultados do modelo proposto, *ICA-SVR*, foram comparados com o modelo *SVR* (onde este não usufruiu da utilização de variáveis "filtradas") e com o modelo de passeio aleatório (*Random Walks*) usando os preços anteriores para prever os preços atuais. No final chegaram a conclusão de que o modelo por eles proposto, apresentou um menor erro de previsão e uma alta precisão, em comparação com o método tradicional *SVR* e o modelo de passeio aleatório. Sendo assim, o modelo proposto capaz de detetar e remover o ruido de forma eficaz fazendo com que a performance do algoritmo *SVR* melhore.

No artigo escrito por Olhede & Wolfe, em 2018, encontram-se questões ligadas a regulamentação e a transparência dos algoritmos, uma vez que estes são temas de grande debate. Em termos de regulamentação, estes dão o exemplo da regulamentação europeia para a proteção de dados, em que os cidadãos têm o direito, se afetados por algum algoritmo especifico, em pedir explicações relativamente a decisão deste. Nos últimos anos, têm sido várias as exigências feitas por parte de organizações governamentais e não governamentais para que sejam traçadas leis que regulamentem o uso de informações pessoais e de outro lado também perceber as consequencias nas análises quando um individuo tem o direito de ser removido de uma base de dados. No que toca a transparência, os autores enfatisam que nem sempre é facil "explicar" a decisão dos algoritmos, uma vez que estes envolvem inumeras complexidades. As interações complexas que geraram a previsão são baseadas em (ou que podem ser) preditores interpretáveis, mas a interpretação de sua combinação não é clara e, se tivermos muitos preditores, seu uso combinado pode corresponder a uma aproximação de variáveis cujo uso possa ser razoavelmente percebido como discriminatório (por exemplo, variáveis de *proxy* para raça ou gênero na determinação das perspetivas de emprego de um indivíduo ou capacidade de obtenção de crédito). Segundo os mesmos autores, quanto mais complexo o algoritmo preditivo tende a ser, maior a dificuldade em buscar um entendimento claro de seus mecanismos.

> "Existem também problemas teóricos de decisão que se relacionam com a nossa compreensão da transparência. Pode ser que, em um determinado cenário, o erro preditivo aumente à medida que tornamos os modelos mais transparentes. O que é um trade-off razoável entre erro de previsão e transparência, e como podemos estudar e determinar formalmente esse trade-off? Usando métodos matemáticos adequados, com critérios de otimização explícitos e quantitativos, podemos imaginar fazer essa troca bem definida e, portanto, eventualmente igualmente bem compreendida." (Olhede, et al., 2018)

[1] Principal índice econômico da Bolsa de Valores de Tóquio

[2] Taiwan Capitalization Weighted Stock Index é um índice que abrange todas as ações listas das empresas negociadas na Bolsa de Taiwan, excluindo ações preferenciais, ações de entrega completa e ações listadas por menos de um mês civil.

1.1 Enquadramento Teórico

1.1.1 Mercados Financeiros

Em finanças, os mercados financeiros são todos os ambientes em que ocorrem operações de investimento financeiro, ou seja, todas as operações de compra e venda de ativos financeiros. Esses ativos podem por exemplo ser valores mobiliários (ações, obrigações, entre outros.), mercadorias (pedras valiosas, *commodities*, entre outras.) ou câmbio. Todas essas operações são feitas de forma aberta e regulada. Nestes mercados existem duas contrapartes, os compradores (*buyers*) e vendedores (*sellers*), num único local em que estes podem encontrar-se facilmente de modo a facilitar as negociações.

No estudo de mercados financeiros tem-se dois elementos importantes: ativos financeiros e investidores. Os ativos financeiros são transacionados em índices de bolsa específicos, estando agregados a um registo histórico. Por outro lado, os investidores são cidadãos ou pessoas coletivas que investem o seu dinheiro em determinado projeto, adquirindo assim títulos financeiros – ações, obrigações de empresas ou de tesouro, entre outros.

De uma forma mais prática, podemos ver os mercados financeiros como canais em que fluem fundos, em que se tem um investidor com excesso de ativos/fundos que empresta o que tem em excesso para outro investidor com défice dos mesmos.

Figura 1: Interação/Fluxo dos mercados financeiros

Estes tipos de mercado são sem dúvida um dos mais importantes e interessantes, uma vez que estes influenciam na atuação das empresas e no consumo de bens e serviços dos cidadãos.

Os mercados financeiros podem ser divididos em quatro mercados (Estrategia-Bolsa, 2020):

- Mercado monetário

Este é um mercado onde são transacionados ativos financeiros de curto prazo, estes instrumentos financeiros (notas comerciais, papel comercial, certificados de deposito, entre outos.) são facilmente convertidos em dinheiro a um baixo custo de transação. As operações financeiras neste tipo de mercado podem ter a sua duração estendida até um ano e são negociadas a curto prazo. Dada a dimensão das operações

realizadas, este tipo de mercados financeiros é essencialmente interbancário, onde as instituições financeiras com excedentes de tesouraria emprestam fundos a outras instituições com necessidade de liquidez. Este tipo de mercado representa uma fonte institucional de capital de trabalho para as empresas.

- Mercado Forex

Trata-se do mercado com maior liquidez, uma vez que as negociações são praticamente 24 horas por dia e quase 7 dias por semana. Segundo Silva (2015), o mercado cambial ou *forex* ou também representado pela sigla *fx* (*Foreign Exchange*) é um mercado global onde se transacionam todos os pares de divisas mundiais e cujo *flow* e liquidez são assegurados alternativamente pela Europa, Estados Unidos e Ásia. Tais transações podem ser efetuadas hoje (mercado *spot*) ou em uma data futura (mercado *forward*).

- Mercado de *commodities*

Trata-se de um mercado onde são comercializadas/negociadas matérias primas essenciais, onde a qualidade e as características são uniformes, sendo o preço uniformemente determinado pela oferta e procura internacional. Normalmente os produtos negociados podem ser mantidos em stock durante algum tempo sem perder a qualidade, como é o caso por exemplo do petróleo ou dos cereais.

- Mercados de capitais

O mercado de capitas é onde as empresas conseguem financiar-se/captar recursos através da negociação de títulos com pessoas que querem investir e multiplicar o seu dinheiro, ou seja, este mercado funciona como intermediário de negociações entre quem precisa captar recursos para financiar projetos e quem deseja investir. Este mercado pode ser subdivido em três partes:

-Mercado de derivados

Neste mercado negociam-se títulos cujos seus valores baseiam-se num ativo subjacente (ações, índice bolsista, matéria prima, entre outros ativos financeiros). Os derivados não têm existência física, mas emergem do contrato entre duas partes. Um dos derivados mais comercializados neste tipo de mercado são futuros, *forwards*, *swaps* e opções.

-Mercado de obrigações

No mercado de obrigações o investidor empresta uma certa quantia por um determinado período e a uma taxa fixa. Uma obrigação é um título de crédito que promete pagar o seu valor na maturidade. Periodicamente a obrigação paga de igual forma um juro

(cupão), fixado na data de emissão ou calculado periodicamente. Desta forma, existem obrigações de taxa fixa e obrigações de taxa variável.

- Mercado de ações

No mercado de ações as empresas emitem títulos com vista a obterem financiamento, sendo assim uma ação um título representativo de uma parcela do capital social de uma empresa. Os investidores ao comprarem uma ação automaticamente adquirem o direito de voto (existindo ações sem direito de voto, transacionadas a um preço mais baixo) e o direito ao recebimento de uma parte dos dividendos distribuídos, esse pagamento é calculado a partir dos lucros realizados pela empresa graças ao financiamento obtido e proporcional ao número de ações adquiridas.

1.1.2 Índices Bolsistas

Entende-se como índices bolsistas as séries temporais financeiras que contêm o valor de uma determinada carteira de ações, as quais englobam diversas indústrias, setores e economias consideradas representativas no movimento dos mercados. Um índice é um indicador estatístico que é medido em pontos que não representam valores de unidade monetárias, mas refletem o valor médio combinado de valores de ativos. A variação do índice geralmente reflete a tendência (alta ou baixa) da bolsa.

Os índices bolsistas podem ser vistos como um "medidor" do funcionamento do mercado, uma vez que as ações tendem a seguir um movimento em grupo, desvalorizando-se ou valorizando-se conjuntamente. Sendo assim a evolução do respetivo grupo de ações traduzido em um número, ou seja, na média das subidas e descidas das cotações das ações constituintes do grupo. Esse número serve de indicador para os investidores na tomada de decisões.

No que diz respeito ao cálculo de um índice bolsista tanto pode-se considerar todas as ações cotadas numa bolsa (índices gerais) ou pode-se restringir o índice as ações mais importantes/negociadas em bolsa (índices de seleção), sendo estas últimas ponderadas de forma a refletir a sua importância no mercado.

A composição dos índices é geralmente revista periodicamente, de modo a manter a sua representatividade em bolsa. Sendo a entrada de uma ação, a curto prazo, num índice de referência um atrativo para os investidores. A entrada e saída de empresas na constituição dos índices são geralmente anunciadas com antecedência pelas entidades responsáveis. Os investidores têm capacidade de fazer as suas próprias previsões, uma vez que as fórmulas de cálculo são conhecidas. Sendo assim frequente observar-se que quando uma empresa/ação se encontra prestes a integrar um índice a procura por este aumente e por consequente a sua cotação também aumenta. De forma oposta, quando uma ação se encontra prestes a sair do índice a sua cotação cai. Apesar destas pequenas oscilações, verifica-se que a longo prazo estas alterações tendem a ser irrelevantes no valor de um índice.

Tendo em conta que os índices bolsistas são cotados com base nos dados de um determinado grupo de títulos, ao compor tais índices define-se primeiramente o conteúdo do mesmo conteúdo – a sua composição, o tipo e número de ações cotadas, a ponderação atribuída as ações e a média estabelecida do índice. Sendo a forma como o índice vai seguir o mercado vai depender desse.

Em relação a metodologia de construção, os índices podem ser classificados por meio de dois critérios: primeiro pelo procedimento do cálculo da média, que pode ser uma média aritmética ponderada ou uma média geométrica; segundo pelo sistema de ponderação que pode ser ponderação pelo preço, valor de mercado ou ponderação igual.

1.1.3 Previsão dos Índices Bolsistas

A previsão do preço de índices bolsistas tem sido tema de interesse dos investidores ao longo dos anos, que recorrem cada vez mais a algoritmos e profissionais das áreas de computação ou com conhecimentos sólidos em *data science*, de forma a obterem previsões cada vez mais precisas.

Os investidores por norma querem comprar ações a preços baixos e vendê-las a preços altos. À primeira vista parece algo fácil, mas é uma tarefa que se revela bastante desafiadora. De acordo com a Teoria da Eficiência dos Mercados desenvolvida por Fama (1970), considera-se que um mercado é eficiente quando toda a informação disponível é refletida nos preços. De acordo com este princípio, não é possível prever o preço das ações, uma vez que as mesmas devem seguir um padrão de passeio aleatório, ou seja, imprevisível. De ponto de vista prático e empírico, está cada vez mais claro que, os mercados nem sempre são tão eficientes, fazendo com que os investidores procurem identificar essas ineficiências temporárias, obtendo lucro quando os mercados regressam à eficiência.

Existem dois diferentes grupos que estudam os fatores e condições que levam a mudança de preços e se dedicam a prever os preços futuros, nomeadamente:

- Análise fundamental: estuda os fatores subjacentes que influenciam o valor das empresas e negócios, incluindo fatores económicos e condições da indústria, condições financeiras das empresas, organização e outros fatores.

- Análise técnica: consiste na previsão do movimento dos preços através de estudos estatísticos que analisam as atividades passadas (movimento dos preços, volume, dados do mercado, entre outros).

Para o presente estudo será abordada a análise técnica. Nos últimos anos vários modelos e algoritmos forma desenvolvidos para prever o preço das ações. Um dos mais importantes são os modelos autorregressivos integrados e de médias móveis (*ARIMA*), conhecidos pela sua eficiência e robustez em previsões de séries temporais financeiras, especialmente quando se trata de previsões de curto prazo. Estes modelos foram

introduzidos por Box e Jenkins (1970), a fim de modelarem a função de autocorrelação de uma série temporal com o mínimo de parâmetros possíveis, utilizando uma combinação de termos de auto-regressão (*AR*), integração (*I*) e média móvel (*MA*). A metodologia de Box-Jenkins (B-J) é composta por um conjunto de atividades para identificar, estimar e diagnosticar modelos *ARIMA* com dados de séries temporais. O método de B-J capta as correlações históricas entre os dados e extrapola-as para períodos futuros. Sendo assim, nos modelos *ARIMA* o valor futuro de uma variável a combinação linear de valores e erros passados. Quanto mais fortes, homogéneas e estáveis forem as correlações históricas melhor será a exatidão e a previsão dos modelos em questão.

Segundo o método B-J, a serie temporal em estudo é ajustada a um modelo matemático que apresenta o menor erro em relação aos outros modelos aplicados. A aplicação deste método depende se a série temporal é ou não estacionária. Caso a serie seja estacionária aplica-se o modelo *ARMA*, caso contrário o modelo *ARIMA*. Sendo este último integrado até que se obtenha a estacionariedade, geralmente só sendo necessário integrar-se a série uma única vez.

A aplicação do método B-J exige a remoção de padrões não estacionários da série (tendência, sazonalidades, volatilidade, entre outros). A não estacionariedade é retirada através de transformações matemáticas, sendo as mais usadas a diferenciação e logaritmização dos dados da série.

Uma vez obtida a estacionariedade da série, identificam-se as ordens apropriadas dos parâmetros de *MA* – através dos coeficientes de autocorrelação (ACF) – e as ordens adequadas para os parâmetros de *AR* – através dos coeficientes de autocorrelação parcial (PACF).

Para a estimação dos parâmetros dos modelos em questão (*ARMA* ou *ARIMA*) recorre-se ao método dos mínimos quadrados que é um método de otimização não linear que minimiza a soma dos quadrados dos erros. Posto isto, verifica-se se os resíduos são ruido branco. Caso sejam ruido branco, significa que o modelo escolhido é adequado aos dados, caso contrário tem-se que voltar a escolher outro modelo e estimar novamente os parâmetros e analisar os resíduos.

A previsão de séries temporais pode ser feita *in-sample* - em que são estimados valores dentro da série com o objetivo de testar a precisão da mesma - ou *out-of-sample* – prevê valores futuros fora da janela temporal da amostra.

No presente estudo será feito primeiramente a previsão do índice *in-sample*, em que com o auxílio das medidas de erro seja escolhido o melhor modelo para a previsão do índice. Tendo o modelo sido escolhido, será feita a previsão *out-of-sample* para a obtenção dos valores futuros.

Na previsão de uma série, o intuito é escolher o modelo que melhor se adequa aos dados para prever de forma eficaz a variável em foco, e não escolher o modelo que melhor explica a mesma variável. Sendo assim, a seleção do(s) modelo(s) adequado(s) é feita com o auxílio das medidas do erro de previsão. As medidas mais utilizadas na literatura são: Erro absoluto médio (MAE – *Mean Absolut Error*), Erro percentual médio (MPE – *Mean Percentual Error*), Média dos erros percentuais absolutos (MAPE – *Mean Absolut*

Percentual Error), Erro quadrático médio (MSE – *Mean Square Error*) e Raiz do erro quadrático médio (RMSE – *Root Mean Square Error*).

1.2 Big Data

1.2.1 Os 5 V's no mercado Financeiro

O *big data* geralmente é caracterizado pelo **volume**, **variedade** e **velocidade**, conhecidos como os **3 Vs**. Tendo em conta a evolução e o contexto atual, foram sugeridos e adicionados mais dois **Vs**, sendo eles referentes a **veracidade** e ao **valor**.

1. **Volume**

Um dos grandes desafios na utilização do *big data* é o seu volume de dados, uma vez que são precisas ferramentas e técnicas para o seu tratamento e análise de forma a que os mesmos possam ser convertidos em informações úteis nos meios em que se inserem. Segundo o site TechJury (2019), hoje em dia cada utilizador na internet gera cerca de 2.5 quintiliões de *bytes*[3] de dados por dia, sendo previsto para 2020 um total de 40 *zettabytes*[4] de dados produzidos.

2. **Velocidade**

Em paralelo com o grande volume de dados, tem-se a velocidade com que os mesmos são gerados. Sendo hoje em dia cada vez mais crucial para as empresas e outras entidades obterem informações rapidamente ou até mesmo em tempo real, uma vez que é necessário utilizá-los antes que estes percam o seu valor. Tomemos como exemplo as empresas de transportes não convencionais (Uber, Bolt, Táxi, etc.), estes tipos de empresas precisam monitorar em tempo real a quantidade de pedidos que são feitos, de modo a responder a procura necessária de que cada zona.

No contexto financeiro, antigamente as entidades financeiras reportavam os preços e movimentos do mercado uma vez por dia. No cenário atual qualquer pessoa pode acompanhar em tempo real ou em intervalos de tempo pré-definidos a evolução dos mercados, de modo a que possam tomar decisões atempadamente. O uso de dispositivos moveis e da web contribuiu drasticamente para o aumento da velocidade e frequência das transações bancárias e pagamentos.

[3] É o tamanho ou a quantidade de memória de um certo dispositivo, sendo a sua codificação usual de 8 bits (um bit assume apenas dois valores: 0 ou 1).
[4] É uma unidade de informação ou memória, correspondente a 1180591620717411303424 (2^{70}).

3. Variedade

Os dados podem ser provenientes de diferentes fontes, tais como: redes sociais, GPS, aplicativos, emails, entre outras. Uma vez que não seguem um único padrão os dados podem ser classificados como estruturados, semiestruturados e não estruturados.

Os dados estruturados referem-se a dados organizados por tabelas, em que nas colunas encontra-se as variáveis e nas linhas as observações. Estes dados são os mais comuns no mercado financeiro, nomeadamente em forma de séries temporais.

Em relação aos dados não estruturados estes contêm informações desorganizadas, sem qualquer tipo de padrão pré-definido. Estes tipos de dados provêm normalmente de redes sociais, emails e notícias, sendo que com o avanço da tecnologia este podem conter áudio e vídeo.

Tal como o nome indica, os dados semiestruturados referem-se a mistura de dados estruturados com os não-estruturados.

4. Veracidade

Com o aumento drástico do volume e da velocidade astronómica com que os dados são gerados, muitas vezes estes contêm erros ou informações irrelevantes. Tornando-se assim essencial verificar a veracidade dos dados, ou seja, a qualidade. Questões como os dados encontram-se bem organizados ou atualizados, existem valores em falta, estarão os dados "limpos" ou irão estes acrescentar algum valor, são questões importantes para verificar a veracidade ou qualidade dos mesmos.

5. Valor

Com o crescente volume de dados é cada vez mais recorrente a presença de dados ou variáveis pouco relevantes no acréscimo de valor, por isso é cada vez mais importante desenvolver métodos capazes de transformar um "tsunami" de dados em algo que acrescente valor.

> *"Nós conseguimos localizar os nossos clientes mais importantes e conseguimos saber quais não tem qualquer valor, uma vez que estes estão constantemente a cancelar o contrato ou a pedir descontos ou por suspeita de fraude. Estes clientes têm um perfil idêntico, mas são todos diferentes. Durante anos não eramos capazes de distingui-los, hoje em dia conseguimos graças ao big data*"
>
> Testemunho da MetLife para a BBVA.

1.2.2 Data Science

Data Science ou Ciência de Dados, é uma disciplina antiga no mundo académico, uma vez que os cientistas sempre recorreram aos dados como forma de compreenderem comportamentos e fenómenos através das observações.

Nos últimos anos a expressão "*Data Science*" ganhou uma maior visibilidade devido ao *big data*, pois se antigamente os dados eram recolhidos de forma controlada, nos dias atuais os dados proveem de diversas plataformas/fontes, tais como: sensores, websites, redes sociais, entre outras. Além do mais apresentam diferentes formatos e são cada vez mais complexos. Atualmente não existe nenhuma área em que a Ciência dos Dados não esteja presente ou influencie.

Pode-se assim definir, *Data Science* como uma ciência interdisciplinar e colaborativa, que envolve várias disciplinas, nomeadamente: Estatística, Matemática, Computação, Engenharia, Economia, Finanças, entre outras. A ciência de dados compreende o significado dos dados e os utiliza como um instrumento para a resolução de problemas e para a tomada de decisões.

1.2.3 Econometria Vs. Machine Learning

A palavra **Econometria** provém das letras gregas *métron* e *oikonomía*, "medição em ou da economia". A Econometria baseia-se em modelos matemáticos e estatísticos para avaliar as teorias economicas e financeiras, fazer previsões de variáveis financeiras, examinar o efeito nos mercados financeiros de uma mudança na economia, entre outras aplicações.

Segundo Brooks (2008), para a formulação de um modelo econométrico é preciso seguir os seguintes passos:

- Ter uma noção geral do problema

Neste primeiro passo será feita a formulação do modelo teorico ou será utilizada a intuição da teoria financeira de que duas ou mais variaveis podem estar de certo modo relacionadas umas com as outras. Sendo o modelo incapaz de capturar todos os fenomenos do mundo real, mas deve ser capaz de apresentar boas aproximações ao que é pretendido.

- Aquisição de dados relevantes para o modelo
- Escolha de metodos de estimação relevantes para o objetivo proposto no primeiro passo
- Adequabilidade do modelo escolhido

Neste passo é verificado quais são os requisitos necessários para estimar de forma correta os parâmetros do modelo e se os mesmos são satisfeitos pelos dados.Por último se o modelo descrever de forma correta os dados prossegue-se para o passo seguinte, caso contrário será necessário voltar a reformular o problema ou alterar os dados.

- Aplicar o modelo

Uma vez obtido o modelo final, este pode ser utilizado para testar a teoria abordada na primeira fase ou pode ser utilizado para fazer previsões.

Com a crescente quantidade de dados e com os problemas que dele advém, como heterogeneidade e o viés , as ferramentas tradicionais utilizadas para a análise econometrica têm-se revelado incapazes de lidar com estas questões. Nos últimos anos imensos estudos têm sido publicados recorrendo as técnicas de ***Machine Learning*** como forma de contornar ou solucionar os problemas que as ferramentas tradicionais não conseguem lidar.

1.2.4 Aplicações do *big data*

As novas tecnologias inerentes ao *big data* estão a revolucionar a indústria financeira. Grandes empresas estão a implementar cada vez mais essas novas tecnologias (*AI, Machine Learning, cloud*, entre outras) como forma de responder a procura dos consumidores e da indústria, aumentar os lucros e implementar a transformação digital. O *big data* contém um vasto leque de aplicações em diversas áreas, sendo que na área das finanças este pode ser aplicado em áreas como:

- **Previsão de séries temporais financeiras**

Com os recentes desenvolvimentos das tecnologias inerentes ao *big data*, é possível obter-se previsões cada vez mais precisas através da introdução de diversas variáveis e dos inúmeros dados. Lu, *et al*. (2009) usaram técnicas de *ML*, nomeadamente *Support Vector Regression* (*SVR*) com auxílio da análise de componentes principais (*Independent Component Analysis - ICA*) para prever séries temporais. Qian (2019) no seu artigo comparou os métodos tradicionais (*ARIMA, GARCH*) com os novos métodos (*logistic regression, multiple-layer perceptron, support vector machine*) para a previsão de séries temporais, chegando a conclusão de que os novos métodos de *ML* fornecem melhores resultados.

- **Conceção de crédito e empréstimos**

Através por exemplo dos algoritmos do *machine learning* é possível aprovar empréstimos e créditos bancários em questão de segundos.

Informações provenientes de redes sociais, *e-commerce* e estatísticas micro geográficas são usadas para determinar matematicamente a capacidade de crédito de um determinado conjunto clientes (Yu, et al., 2016).

- **Deteção de Fraudes**

Os algoritmos de *machine learning* podem ser usados por bancos para monitorar milhares de transações, de modo a identificar operações fraudulentas. Esses algoritmos são geralmente muito eficazes e rápidos, sendo possível prevenir fraudes em tempo real.

- **Bancos comerciais**

A análise dos dados ajuda os bancos a economizar, a maximizar o valor dos dados e a entender melhor os seus clientes de forma a prever possíveis atritos, estar um pé a frente da concorrência, fazer ofertas personalizadas aos clientes e entre outras vantagens.

- **Trading**

Com a automatização é possível com que os computadores executem operações em velocidades e frequências astronómicas. As negociações utilizando algoritmos de *machine learning* são baseados em modelos matemáticos, sendo as operações efetuadas em momentos oportunos, reduzindo assim erros e aumentando os lucros. Tendo em conta a versatilidade dos algoritmos no que diz respeito ao tipo de dados (estruturados, semiestruturados e não-estruturados) que recebem como input, estes incorporam notícias em tempo real, dados de redes sociais e outros dados relevantes de forma a gerar melhores decisões de negociação.

- **Políticas monetárias**

As entidades financeiras (bancos centrais, governos, etc.) têm um grande interesse em entender/prever a reação das pessoas e do mercado no geral quando se procede com alguma alteração na política monetária (subida ou descida das taxas de juro, austeridade, etc.), sendo muitas vezes difícil fazer essa previsão. Com o uso do *big data* e das ferramentas que dele advém, estas entidades podem adquirir diversos conhecimentos/dados que poderão ajudar na estimação do efeito da subida ou descida das taxas em diversos sectores da economia. Segundo Wells Fargo Securities Economics Group (2017) a resposta dos consumidores a uma subida das taxas de juro pode ser estimada através da utilização de cartões de credito (refletindo os gastos regulares), gastos em coisas duráveis (pode demonstrar uma mudança no consumo a longo termo) e em hipotecas. Essas estimativas podem depois ser divididas de acordo com a demografia ou sectores, de modo a ampliar as possibilidades e a poder-se tomar decisões de acordo com cada caso/grupo.

Estas são algumas das principais aplicações do *big data* na indústria financeira. Existem outras diversas aplicações, mas a explanação de todas elas seria dispendiosa para o corrente trabalho

2. Séries Temporais

2.1 Noções gerais de Séries Temporais

Uma série temporal ou sucessão cronológica pode ser definida como um conjunto de dados que foram coletados durante um período de tempo determinado com intervalos iguais, sendo que estes podem conter uma ou mais variáveis. O período de tempo é a frequência com que os dados são registados, por exemplo dias, meses ou anos.

Uma série temporal pode ser caracterizada por:

$$\begin{cases} Y = F(t), & t = 1,2,\dots,n \quad \text{se discreta} \\ Y = F(t), & t \in I \subseteq \mathbb{R} \quad \text{se continua} \end{cases} \tag{2.1}$$

Podendo os movimentos característicos das séries temporais serem classificados em quatro tipos principais, denominados componentes de uma série temporal:

- Os **movimentos de tendência**, descrevem/referem-se a direção geral segundo a qual a série temporal tende a desenvolver-se num intervalo de tempo longo.

- Os **movimentos ou variações cíclicas**, referem-se às oscilações de longo prazo ou desvios em torno da reta ou da curva de tendência. Estes ciclos, como são frequentemente denominados, podem ou não seguir padrões exatamente semelhantes ao longo de intervalos de tempos iguais.

- Os **movimentos ou variações por estação (sazonalidade),** referem-se a padrões idênticos (ou quase), a que uma série temporal parece obedecer durante os mesmos meses ou períodos de anos sucessivos.

- Os **movimentos irregulares ou aleatórios**, referem-se aos deslocamentos esporádicos das sucessões cronológicas, provocados por acontecimentos casuais, como desastres naturais, eleições, entre outros acontecimentos. Embora, admita-se que esses acontecimentos produzem variações apenas em um curto período, é concebível que elas sejam tão intensas que acarretem novos movimentos cíclicos ou de outra natureza.

As séries temporais podem também ser caracterizadas pela sua dependência aos eventos. Tendo assim:

- **Independência**, quando uma série é puramente aleatória ou ruido branco, ou seja, conhecer y_t não ajuda em nada a prever y_{t+1};
- **Memória longa**, quando a dependência desaparece lentamente, ou seja, os valores de pontos no passado influenciam momentos muito avançados no tempo;
- **Memória curta**, quando a dependência desaparece rapidamente.

Antes de proceder-se com a análise econométrica de uma série temporal, faz-se primeiro a análise descritiva dos dados, de modo a entender a distribuição dos mesmos. Analisa-se assim:

Assimetria

A assimetria não é nada mais do que, o grau de afastamento de uma distribuição do seu eixo de simetria. Esse afastamento pode ocorrer do lado esquerdo (assimetria positiva) ou do lado direito (assimetria negativa) da distribuição:

- Distribuição simétrica
  $\bar{x} = Moda = Median$

- Distribuição assimétrica positiva ou à direita
  $\bar{x} \geq Mediana \geq Moda$

- Distribuição assimétrica negativa ou a à esquerda
  $\bar{x} \leq Mediana \leq Moda$

Figura 2: Distribuição simétrica, assimétrica positiva e assimétrica negativa

A assimetria pode também ser estudada, através do coeficiente do momento de assimetria:

$$s_k = \frac{\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^3}{\sqrt{(\frac{1}{n}\sum_{i=1}^{n}(x_i - \bar{x})^2)^3}} \qquad (2.2)$$

onde o *n* corresponde ao número de observações da série e x_i ao valor registado no instante *i*.

De acordo com o resultado obtido no cálculo do valor do coeficiente, pode-se ter os seguintes casos:

- Distribuição simétrica, se $s_k = 0$;
- Distribuição assimétrica positiva, se $s_k > 0$;
- Distribuição assimétrica negativa, se $s_k < 0$.

Curtose

A curtose é o grau de achatamento da distribuição, ou por outras palavras, a curtose medo o quanto uma curva de frequência será achatada em relação a uma curva normal de referência.

De forma a avaliar a curtose, recorre-se ao seu respetivo coeficiente do momento:

$$k = \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^4}{\sqrt{(\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2)^2}} \tag{2.3}$$

Consoante o valor observado do coeficiente do momento de curtose associado a série, assume-se que a mesma se rege de acordo com uma das seguintes distribuições:

- $k = 3$, distribuição mesocúrtica;
- $k > 3$, distribuição leptocúrtica;
- $k < 3$, distribuição platicúrtica.

De forma geral, os coeficientes de assimetria e de curtose, são utilizados para fazer inferência sobre a normalidade da série ou dos resíduos em estudo. Diz-se que uma série segue uma distribuição normal se a assimetria for próxima ou igual a zero e se o coeficiente de curtose for igual a três, ou seja, a distribuição é mesocúrtica.

De forma a obter resultados mais robustos acerca da normalidade, recorre-se também aos testes estatísticos, sendo o mais conhecido o teste de *Jarque-Bera*.

Teste Jarque-Bera (JB)

O teste de JB é um dos testes estatísticos mais usados para testar a normalidade (Brooks, 2008), sendo que este usa o terceiro e quarto momento da distribuição para medir a normalidade – respetivamente a assimetria e a curtose.

A estatística de teste de Jarque-Bera é dada por:

$$W = T\left[\frac{(b_1)^2}{6} + \frac{(b_2 - 3)^2}{24}\right] \sim \chi^2(2) \qquad (2.4)$$

onde T é o tamanho da amostra; $b_1 = \frac{E[u^3]}{(\sigma^2)^{3/2}}$ e $b_2 = \frac{E[u^4]}{(\sigma^2)^2}$, sendo u o erro e σ^2 a sua variância.

Sendo as hipóteses a testar as seguintes:

$$H_0: Distribuição\ Normal \qquad \text{Versus}$$

$$H_1: Distribuição\ não\ Normal$$

Se a hipótese nula for rejeitada, conclui-se que a série ou os resíduos não seguem uma distribuição normal.

2.2 Processos Estocásticos

Dado um espaço de probabilidade[5] $(\Omega, \mathcal{F}, \mathcal{P})$ e um espaço mensurável [6] $(\mathcal{S}, \Sigma)$, entende-se por um processo estocástico como um conjunto/família de variáveis aleatórias $\{Y_t: t \in T\}$, onde cada Y_t é uma variável aleatória de valor $\mathcal{S}$ em Ω.

Tendo em conta que se pretende analisar uma série temporal financeira recorrendo a métodos estatísticos, convém considerar as séries $(y, y_2, \dots, y_T)$ observadas como uma realização particular de um processo estocástico.

Tendo assim, um processo estocástico linear a seguinte forma:

$$Y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \ \dots + a_n y_{t-n} + \varepsilon_t \qquad (2.5)$$

Onde ε_t representa alguma perturbação ou inovação da equação com diferenças lineares de ordem *n* em *y* e onde a_i, *i = 1,…,n,* são coeficientes reais.

[5] $(\Omega, \mathcal{F}, \mathcal{P})$ é o espaço de probabilidade de uma experiência aleatória, onde Ω é o espaço de resultados, $\mathcal{F}$ o espaço de acontecimentos e $\mathcal{P}$ uma função de probabilidade.
[6] Em teoria da medida, um espaço mensurável é um conjunto $\mathcal{S}$ dotado de uma sigma-álgebra Σ.

Estatísticas

Seja Y_t um processo estocástico com $E[y(t)^2] < \infty$, define-se como suas estatísticas:

- **Valor médio**

$$\mu_t = E[y_t] \tag{2.6}$$

- **Variância**

$$\sigma_{y_t}^2 = Var[y_t] = E[(y_t - \mu_t)^2] \tag{2.7}$$

- **Covariância**

$$cov(X, Y) = E\{[X - \mu_X][Y - \mu_Y]\} = E[X\,Y] - \mu_X\,\mu_Y \tag{2.8}$$

- **Autocovariância**

A estabilidade da covariância é quantificada pela função de autocovariância (a autocovariância de *lag s* é a covariância entre y_t e y_{t-s}) definida por:

$$\begin{aligned}\gamma(t, s) = cov[y_t, y_{t-s}] &= E\{[y_t - \mu_t][y_{t-s} - \mu_{t-s}]\} \\ &= E[y_t\,y_{t-s}] - \mu_t\,\mu_{t-s}\end{aligned} \tag{2.9}$$

- **Coeficiente de correlação de *Pearson***

$$\rho_{X,Y} = \frac{\gamma_{X,Y}}{\sigma_X\,\sigma_Y} = \frac{cov[X, Y]}{\sqrt{Var(X)\,Var(Y)}} \tag{2.10}$$

Sendo os seus valores compreendidos no intervalo [-1,1].

- **Função de autocorrelação (*ACF*)**

A sequência de coeficientes de autocorrelação designa-se por função de autocorrelação

$$\rho_{(t,s)} = \frac{cov[y_t, y_{t-s}]}{\sqrt{Var[y_t]\,Var[y_{t-s}]}} = \frac{\gamma(t, s)}{\gamma(0)} \tag{2.11}$$

- **Correlograma**

Ao momento

$$E((y_t - E(y_t))\,(y_{t-s} - E(y_{t-s})) = \gamma_s\,, s = 0,1,2,\ldots \tag{2.12}$$

designa-se função de autocovariância. Quando $s = 0$, obtém-se a autocovariância no lag zero, ou seja, a autocovariância de y_t com y_t, mais concretamente a variância de *y*. A autocovariância revela-se uma medida pouco útil na medição da relação entre *y* e os seus restantes valores, pelo que se recorre a autocorrelção, que não é nada mais nada menos do que a autocovariância normalizada:

$$\tau_s = \frac{\gamma_s}{\gamma_0}\,, s = 0,1,2,\ldots \tag{2.13}$$

Sendo os valores da série τ_s contidos no intervalo entre -1 e +1. Pondo os valores de τ_s num gráfico teremos então a **função de autocorrelação** (***ACF***), também conhecida como **correlograma**.

- **Operador Lag**

O operador *lag* (desfasamento) L é definido como $Ly_t = y_{t-1}$ e as suas iteradas de ordem superior $L^h y_t = y_{t-h}$, onde *h* é o número de iterações.
Tipicamente opera-se sobre uma série não com o operador *lag* mas sim com o polinómio operador *lag*, isto é*:*

$$P(Y) = a_0 y_t + a_1 y_{t-1} + \ \ldots + a_n y_{t-n} = \sum_{k=1}^{n} a_k y_{t-k} \tag{2.14}$$

2.3 Estacionariedade

Uma das mais importantes propriedades das séries temporais é a estacionariedade, uma vez que a maioria das propriedades dos estimadores dos modelos econométricos só são válidos se as séries forem estacionárias.

Diz-se que um processo estocástico é estritamente estacionário se as propriedades da série temporal não são afetadas por uma mudança no tempo. Em outras palavras, a distribuição de probabilidade conjunta em qualquer intervalo de tempo $t_1, t_2, \ldots, t_m$ deve ser o mesmo que a distribuição de probabilidade conjunta nos intervalos $t_1 + k, t_2 + k, \ldots, t_m + k$, onde k é uma variável arbitrária no tempo.

Caso tenhamos $m = 1$, isso implica que a distribuição de probabilidade marginal não depende do tempo, o que implica que a média e a variância sejam constantes.

$$E(y_1) = E(y_2) = \ \ldots \ = E(y_T) = E(y_t) = \mu_t \tag{2.15}$$

$$\begin{gathered} Var(y_1) = Var(y_2) = \cdots = Var(y_T) = Var(y_t) \\ = E[(y_t - \mu_t)^2] = \sigma_{y_t}^2 \end{gathered} \tag{2.16}$$

Se $m = 2$, a estrita estacionariedade implica que as distribuições bivariadas não dependam do tempo (t), o que quer dizer que a covariância depende só lags.

$$cov(y_1, y_{1+k}) = cov(y_2, y_{2+k}) = \ \dots \ = cov(y_{T-k}, y_T) = \ cov(y_t, y_{t-k}) \tag{2.17}$$

É importante observar que, as propriedades apresentadas são aplicadas apenas para o primeiro e segundo momento do processo, mais conhecido por estatísticas de segunda ordem.

Tendo em conta o que foi referido no paragrafo anterior, as propriedades de estacionariedade estrita não se verificam na sua totalidade, pelo que daqui em diante quando se falar em estacionariedade ir-se-á referir a **estacionariedade fraca**.

Um processo/modelo estocástico diz-se fracamente estacionário (estacionariedade fraca ou em covariância) se, $\forall\ t, t-s \in I$, verificar-se as seguintes propriedades:

- **Média constante**

$$E(y_t) = E(y_{t-s}) = \mu \tag{2.18}$$

- **Variância constante**

$$Var(y_t) = E((y_t - \mu)^2) = E((y_{t-s} - \mu)^2) = \sigma_y^2 < \infty \tag{2.19}$$

- **Covariância constante**

$$E[(y_t - \mu)(y_{t-s} - \mu)] = E[(y_{t-s} - \mu)(y_{t-j-s} - \mu)] = \gamma(s) \tag{2.20}$$

É de realçar que a estacionariedade implica estacionariedade fraca, mas o contrário não é necessariamente verdade (só é verdade para a distribuição Gaussiana). Uma série temporal que falha alguma das propriedades acima referidas, diz-se não-estacionária.

Na maioria das vezes as séries não apresentam estacionariedade, pelo que é necessário proceder com algumas alterações nas séries com o intuito de obter a estacionariedade:

- **Diferenças sucessivas**

$$\Delta y_t = y_t - y_{t-1}$$
$$\Delta^2 y_t = y_t - 2y_{t-1} + y_{t-2} \tag{2.21}$$

- **Diferenças *logaritmizadas***

$$\Delta\log(y_{t)} = \log(y_t) - \log(y_{t-1}) \tag{2.22}$$

2.3.1 Ruído branco (*White Noise*)

Uma sequência $\{\varepsilon_t\}$ define ruído branco (*shock*) se tiver média zero, variância constante e não apresentar correlação serial, isto é:

- $E(\varepsilon_t) = 0$;
- $Var(\varepsilon_t) = \sigma^2 < \infty$;
- $E(\varepsilon_t, \varepsilon_{t-1}) = Cov(\varepsilon_t, \varepsilon_{t-s}) = 0, para\ s \neq 0$

Geralmente, o ruído branco denota-se da seguinte forma:

$$\varepsilon_t \sim WN(0, \sigma^2) \text{ ou } y_t \sim WN(0, \sigma^2)$$

2.4 Testes de raiz unitária

Uma série temporal diz-se ter uma raiz unitária quando esta é não-estacionária. Como forma de evitar os problemas que a não-estacionariedade pode induzir, foram criados os testes de raiz unitária, que tem as seguintes hipóteses:

$$H_0: \rho = 1 \quad Vs. \quad H_1: \rho < 1$$

Onde caso não se rejeite a hipótese nula H_0, estaremos perante uma série não estacionaria, ou seja, a série possui uma raiz unitária. Caso se rejeite H_0, a série é estacionaria.

Dentro dos vários testes propostos, destacam-se: o teste de *Dickey & Fuller* (*DF*) e *DF* Aumentado desenvolvidos por Dickey e Fuller (1979); o teste *KPSS* desenvolvido por Kwiatkowski *et al.* (1992) e o teste de *Phillips & Perron* desenvolvidos *por* Phillips e Perron (1988).

Teste de Dickey-Fuller (DF)

O teste de *Dickey-Fuller* (*DF*) foi introduzido em uma versão inicial dos trabalhos de Dickey (1976), Fuller (1976) e Dickey & Fuller (1979). Mais tarde o mesmo foi objeto de uma generalização no trabalho de Dickey-Fuller (1981), conhecida como o teste de *Dickey-Fuller Aumentado* (*ADF*).

No teste de *DF* considera-se o seguinte modelo de regressão linear

$$Y_t = \rho Y_{t-1} + \beta_0 + \beta_1 t + \varepsilon_t \tag{2.23}$$

Onde *t* representa a tendência determinística. Altera-se o modelo ao subtrair-se por Y_{t-1} em ambos os lados da equação. Obtendo-se assim:

$$Y_t - Y_{t-1} = (\rho - 1)\, Y_{t-1} + \beta_0 + \beta_1 t + \varepsilon_t \qquad (2.24)$$

onde, $\rho - 1 = \alpha$.

Sendo assim as hipóteses a serem testadas são as seguintes:

$$H_0: \alpha = \rho - 1 = 0 \quad Vs. \quad H_1: \alpha \neq 0$$

A não rejeição da hipótese nula, implica a não-estacionariedade da série em estudo, caso contrário conclui-se que a série é estacionária.

O teste *DF* é usado quando os erros não têm correlação serial, ou seja, quando são um processo de ruido branco. Mas, nem sempre essa condição é verificada, sendo assim necessário executar o teste com correção paramétrica, conhecido como teste de *Dickey-Fuller* Aumentado (*ADF*).

Teste de *Dickey-Fuller* Aumentado (*ADF*)

O teste de *ADF* é utilizado quando os erros (ε_t) não são ruído branco. Sendo a razão para esta distinção o aumento da regressão que este teste proporciona através da adição de um número de variáveis defasadas da primeira diferença de Y_t, até que os erros obtidos não sejam autocorrelacionados:

$$\Delta Y_t = (\rho - 1)\, Y_{t-1} + \sum \beta_i \Delta Y_{t-i} + \varepsilon_t \qquad (2.25)$$

Sendo as hipóteses a serem testadas as seguintes:

$$H_0: \rho = 1 \quad Vs. \quad H_0: \rho < 1$$

A não rejeição da hipótese nula, leva com que se conclua a não-estacionariedade da série, e caso se rejeite, conclui-se que não existem indícios que levem a rejeitar a estacionariedade da série.

Teste *Phillips-Perron* (*PP*)

Phillips (1987) e Phillips & Perron (1988) propuseram/desenvolveram um outro teste de raiz unitária que generaliza o teste de *ADF* para uma ampla classe de modelos em que os erros são correlacionados e heterogéneos. Este teste, conhecido como o teste de *Phillips & Perron* (*PP*), tem os mesmos procedimentos do teste de *ADF*.

Assim, para o modelo de regressão

$$Y_t = \rho\, Y_{t-1} + \beta_0 + \beta_1 t + \varepsilon_t \qquad (2.26)$$

Tem-se as seguintes hipóteses a serem testadas:

$$H_0: \rho = 1 \quad Vs. \quad H_1: \rho < 1$$

Sendo que, a não rejeição da hipótese nula leva com que se conclua a não estacionariedade da série.

Teste *Kwiatkowski-Phillips-Schmidt-Shin (KPSS)*

Ao contrário dos testes anteriores, o teste de *Kwiatkowski-Phillips-Schmidt-Shin* (*KPSS*) assume como hipótese nula, H_0, a estacionariedade da série, tendo assim as seguintes hipóteses:

$$H_0: \rho < 1 \quad Vs. \quad H_1: \rho = 1$$

Sendo que, ao rejeitar-se a hipótese nula assume-se que a série é não-estacionária. As hipóteses no teste de KPSS são ou não rejeitadas de acordo com os valores críticos associados aos níveis de significância usuais (1%, 5% e 10%).

Ou seja, a hipótese nula é rejeitada se

$$LM_{KPSS} > valores\ críticos$$

Sendo,

$$LM = \sum_{i=1}^{t} \frac{s_t}{n^2 \hat{o}^2} \tag{2.27}$$

onde $s_t = \sum_{i=1}^{t} \varepsilon_i$ e $\hat{o}^2$ um estimador para a variância dos erros.

Este teste é usado como complemento dos testes anteriormente descritos, uma vez que é usada como forma de desempate quando se obtêm diferentes resultados nos testes.

2.5 Modelos Econométricos Lineares

Como se sabe e foi mencionado na seção 1.2.3, a Econometria baseia-se em modelos matemáticos e estatísticos de modo a avaliar as teorias economicas e financeiras, ou seja, tenta entender a relação entre as variaveis de interesse. Sendo cada um desses modelos, de uma forma geral, nada mais do que um conjunto de distribuições conjuntas que satisfazem certos pressupostos.

2.5.1 Modelos Autoregressivos de ordem *p* - *AR*(*p*)

Nos modelos autorregressivos o valor da variável em estudo, *y*, depende apenas dos valores tomados por esta no passado e de um termo erro/perturbação aleatório. O modelo é expresso da seguinte forma:

$$y_t = a_0 + a_1 y_{t-1} + a_2 y_{t-2} + \ \dots + a_p y_{t-p} + \varepsilon_t \ , \varepsilon_t \sim WN(0, \sigma^2) \tag{2.28}$$

onde *p* designa a ordem de desfasamento (*lag order*), $a_i, i = 1, \dots, p$ são coeficientes reais (parâmetros a estimar) e ε_t a inovação (é um processo de ruído branco que assinala a nova informação recebida no passo *t*).

Ou de forma mais compacta,

$$y_t = a_0 + \sum_{i=1}^{p} a_i y_{t-i} + \varepsilon_t \tag{2.29}$$

Ou usando o operador *Lag*,

$$y_t = a_0 + \sum_{i=1}^{p} a_i L^i y_t + \varepsilon_t \tag{2.30}$$

Ou

$$\phi(L) y_t = a_0 + \varepsilon_t \text{ , onde } \phi(L) = 1 - \textstyle\sum_{i=1}^{p} a_i L^i \tag{2.31}$$

Um processo *AR*(*p*) diz-se estacionário (estável) se e só se todas as raízes $z = \frac{1}{\alpha_i}$ do polinómio

$$\phi(z) = 1 - \sum_{i=1}^{p} a_i z^i = \prod_{i=1}^{p} (1 - \alpha_i z) \tag{2.32}$$

estiverem situadas no exterior do círculo unitário, ou seja, $|\alpha_i| < 1$.

Função de autocorrelação

A função de autocorrelação parcial de um processo *AR*(p) é dada por

$$r(p) = \begin{cases} a_h \neq 0, & \text{para } h < p \ (existe\ PACF) \\ 0, & \text{para } h > p \ (não\ existe\ PACF) \end{cases}$$

o que denota a convergência brusca para zero ao fim de *p* passos. A função *ACF* tem uma convergência lenta para zero.

2.5.2 Modelos de Média Móvel de ordem *q* – *MA(q)*

Os modelos de Média Móvel de ordem *p*, *MA(p)*, representam-se por:

$$y_t = \varepsilon_t + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} = \varepsilon_t + \sum_{i=1}^{q} \theta_i L^i \varepsilon_t \tag{2.33}$$

Ou por

$$y_t = \Theta(L)\varepsilon_t \tag{2.34}$$

onde $\Theta(L) = 1 + \sum_{i=1}^{q} \theta_i L^i$ e $\varepsilon_t \sim WN(o, \sigma^2)$. Isto quer dizer que estamos perante um modelo onde as variáveis independentes são os desfasamentos do termo erro, até *lag q*, sendo $\theta_i, i = 1, \dots, q$ os parâmetros a estimar.

Tem-se assim:

- Média

$$E(y_t) = 0 \tag{2.35}$$

- Variância

$$Var(y_t) = 1 + \sigma^2 \sum_{i=1}^{q} \theta_i^2 \tag{2.36}$$

- Função de autocorrelação

$$\gamma(s) = \begin{cases} \sigma^2 \left(\theta_s + \sum_{i=1}^{q} \theta_{s+i}\theta_i \right) para\ s = 1,2,\dots \\ 0\ para\ s > q \end{cases} \tag{2.37}$$

A função de autocorrelação (*ACF*) de um processo *MA*(*q*) anula-se no *lag (q+1)* e a função de autocorrelação parcial converge suavemente para zero.

Um processo *MA*(*q*) é fracamente estacionário para todos os valores de $\{\theta_i\}$.

2.5.3 Modelos Autoregressivos de Média Móvel - *ARMA(p,q)*

Ao combinar os modelos *AR*(*p*) e *MA*(*q*) obtém-se um modelo *ARMA(p,q)*, que é dado por:

$$\phi(L)y_t = a_0 + \Theta(\mathrm{L})\varepsilon_t \tag{2.38}$$

onde $\phi(L) = 1 - \sum_{i=1}^{p} a_i L^i$ e $\Theta(\mathrm{L}) = 1 + \sum_{i=1}^{q} a_i L^i$

Ou

$$y_t = a_0 + \sum_{i=1}^{p} a_i y_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t \tag{2.39}$$

com $E(\varepsilon_t) = 0, E(\varepsilon_t^2) = \sigma^2, E(\varepsilon_t \varepsilon_s) = 0, t \neq s$.

O valor médio de um modelo *ARMA(p,q)* é dada por

$$E(y_t) = \frac{a_0}{1 - \sum_{i=1}^{p} a_i} \tag{2.40}$$

No que diz respeito a função de autocorrelação de um processo *ARMA* esta apresenta combinações de comportamentos derivados dos modelos *AR*(*p*) e *MA*(*q*). Quando os *lags* forem inferiores a *q*, a função *ACF* é idêntica a de um modelo *AR*(*p*).

2.5.4 Modelos Autoregressivos Integrados de Média Móvel – *ARIMA(p,d,q)*

Um modelo *ARMA(p,q)* só pode ser aplicado quando uma série é estacionária. Mas caso a série não seja estacionária, é necessário recorrer ao método de diferenciação sucessiva de modo a tornar-lhe estacionária.

Um modelo *ARIMA(p,d,q)*, corresponde a um modelo *ARMA(p,q)* aplicado a uma série não-estacionária que foi diferenciada *d* vezes até que a sua estacionariedade fosse obtida. Sendo que:

- *p* é a ordem da componente *AR*;

- d é o número de vezes que a série foi diferenciada até se conseguir a estacionariedade;
- q é a ordem da componente *MA.*

2.6 Critérios de comparação

A identificação dos modelos numa primeira fase é feita recorrendo a análise gráfica das funções *ACF* e *PACF* (Função de Auto Correlação Parcial), mas como por vezes os dados são bastante "confusos/desorganizados", esses gráficos não conseguem exibir padrões claros quanto as ordens a escolher. Torna-se difícil nestes casos de interpretar as funções *ACF* e *PACF*, dificultando assim a especificação do modelo.

De forma a eliminar a subjetividade envolvida na interpretação da *ACF* e *PACF*, recorre-se a uma outra técnica conhecida como **critérios de informação** (Brooks, 2008). Estes critérios envolvem dois fatores: um termo que é uma função da soma dos resíduos ao quadrado, e uma penalidade pela perda de graus de liberdade ao adicionar-se parâmetros extras. Entretanto, adicionar uma nova variável ou um *lag* adicional ao modelo irá ter dois efeitos concorrentes nos critérios de informação: a soma dos resíduos quadrados diminui, mas o valor do termo de penalidade aumenta. O objetivo é escolher o número de parâmetros que diminui o valor do critério de informação. Logo, adicionar um termo extra só irá reduzir o valor do critério de informação se a soma dos resíduos ao quadrado diminuir de forma significativa de modo a compensar o aumento no valor do termo da penalidade.

Segundo Brooks (2008), os três critérios de informação mais importantes são o **critério de informação de Akaik*e* – *AIC*** (1974), o **critério de informação de Schwarz Bayesiana – *SBIC*** (1978) e o **critério de informação de Hannan- Quinn (*HQIC*)**.

Algebricamente, os três critérios podem ser expressos da seguinte forma:

- $AIC = \ln(\hat{\sigma}^2) + \frac{2k}{T}$ (2.41)
- $SBIC = \ln(\hat{\sigma}^2) + \frac{2k}{T} \ln(T)$ (2.42)
- $HQIC = \ln(\hat{\sigma}^2) + \frac{2k}{T} \ln(\ln(T))$ (2.43)

onde, $\hat{\sigma}^2$ é a variância residual (equivalente a soma dos resíduos ao quadrado dividido pelo número de observações, T), $K = p + q + 1$ é o número total de parâmetros estimados e T é o tamanho da amostra.

É de realçar que o *SBIC* incorpora um termo de penalidade que é mais rígido/restrito do que o *AIC*, enquanto que o *HQIC* se encontra entre ambos.

2.7 Validação e escolha dos modelos

Depois de selecionado o modelo que melhor descreve os dados, procede-se com a validação do mesmo. A validação consiste na análise dos resíduos do modelo, sendo que estes devem verificar os seguintes pressupostos:

1. **Os erros devem ter média nula, $E(\varepsilon_t) = 0$**
2. **Os erros devem ter variância constante e finita, $Var(\varepsilon_t) = \sigma^2 < \infty$**

Assume-se que a variância dos erros é constante, ou seja, assume-se a homocedasticidade dos erros. Caso a variância não seja constante, estamos perante a heterocedasticidade dos erros.

De forma a verificar se a variância dos resíduos é constante, usam-se testes estatísticos para a heterocedasticidade. Segundo Brooks (2008) o teste mais popular é o **teste de White** (1980), sendo este bastante útil uma vez que faz poucas suposições relativamente a provável forma de heterocedasticidade.

3. **Os erros devem ser linearmente independentes, $Cov(\varepsilon_i, y_j) = 0$**

A covariância entre os termos do erro é zero, ou seja, assume-se que os erros não são correlacionados. Caso os erros sejam correlacionados, diz-se que estes são "autocorrelacionados" ou "correlacionados serialmente".

Numa primeira fase, recorre-se a análise gráfica dos resíduos, a fim de encontrar algum indício de autocorrelação. Mas, mais uma vez os métodos gráficos podem revelar-se difíceis de interpretar, por isso recorre-se a um teste estatístico. Segundo Brooks (2008) o teste mais simples pertence a Durbin e Watson (1951). Este teste apesar da sua simplicidade, é bastante fraco. Sendo os testes de Ljung-box e o de Breuch-Godfrey os mais eficazes para testar a independência.

O **teste de Durbin-Watson (DW)** é um teste de autocorrelação de primeira ordem, o que quer dizer, que testa apenas a relação entre o erro e o seu valor imediatamente a seguir.

Tendo as seguintes hipóteses:

$$H_0: \rho = 0 \quad Vs. \quad H_1: \rho \neq 0$$

Caso a hipótese nula seja rejeitada, conclui-se que existem evidencias de que existe relação/correlação entre sucessivos resíduos. Caso não se rejeite, os erros em $t-1$e t são independentes.

A estatística de Durbin-Watson é expressa da seguinte forma:

$$DW = \frac{\sum_{t=2}^{T}(\hat{\varepsilon}_t - \hat{\varepsilon}_{t-1})^2}{\sum_{t=2}^{T} \hat{\varepsilon}^2{}_t} \tag{2.44}$$

Podendo ainda ser expressa de forma compacta, por

$$DW \approx 2(1 - \hat{\rho}) \tag{2.45}$$

onde $\hat{\rho} = (\hat{\varepsilon}_t, \hat{\varepsilon}_{t-1})$ é a estimativa do coeficiente de correlação. Sendo $\hat{\rho}$ limitada por $-1 \leq \hat{\rho} \leq 1$, se substituirmos na equação acima iremos obter o limite para *DW*, $0 \leq \hat{\rho} \leq 4$. De acordo com os valores que *DW* pode obter, tem-se:

- $\hat{\rho} = 0, DW = 2$ neste caso não existe correlação entre os resíduos, ou seja, a hipótese nula não será rejeitada se a estatística de *DW* for próxima de 2;
- $\hat{\rho} = 1, DW = 0$ neste caso existe auto-correlação positiva nos resíduos;
- $\hat{\rho} = -1, DW = 4$ neste caso existe auto-correlação negativa entre os resíduos.

4. Os erros devem ser normalmente distribuídos, $\boldsymbol{\varepsilon_t \sim N(0, \sigma^2)}$

Assumir que os erros são normalmente distribuídos é necessária para a realização dos testes de hipótese e intervalos de confiança para os parâmetros do modelo. Para verificação deste pressuposto recorre-se ao **teste de normalidade de *Jarque-Bera***.

2.8 Previsão

A previsão ou "*forecasting*" nada mais é do que uma tentativa de determinar a probabilidade de valores futuros de uma série. No caso específico de séries temporais, fazer a previsão é tentar prever os valores futuros da série através dos valores originais da série e/ou através dos termos do erro.

Segundo Brooks (2008), fazer previsões é extremamente essencial, uma vez que estes são uteis principalmente para a tomada de decisões financeiras que geralmente envolvem muitos recursos a longo termo, que só trazem retorno dependendo do que acontecer no futuro.

Determinar a precisão da previsão de um determinado modelo é uma parte/teste essencial para que se possa perceber a adequabilidade do mesmo. Sendo que alguns economistas defendem que adequabilidade estatística de um modelo, quer em termo dos cinco pressupostos abordados na secção 2.6, quer em termos da significância dos seus parâmetros, é bastante irrelevante desde que o modelo produza previsões precisas (Brooks, 2008).

É importante também ter-se noção de que a previsão pode ser feita para um único valor ou para um intervalo de valores, sendo que neste último espera-se que os valores estejam de acordo com o intervalo de confiança dado. De acordo com o valor ou valores que se pretende(m) prever, estes podem ser feitos dentro da amostra (*in-sample*) ou fora dela (*out-of-sample*).

2.8.1 Erro de previsão

Quando se faz a previsão a principal questão que se coloca é se o resultado fornecido pela mesma é preciso ou não. Sendo assim que, numa primeira abordagem calcula-se o **erro de previsão**, que nada mais é do que, a diferença entre o valor real/atual (Y_i) e o valor predito ($\hat{Y}_i$) da observação *i*, sendo este expresso da seguinte forma:

$$e_i = Y_i - \hat{Y}_i \tag{2.46}$$

Neste caso então, o erro pode ser positivo (negativo) se o valor predito for inferior (superior) ao valor real.

De modo a avaliar de forma mais rigorosa a precisão da previsão de determinado modelo, torna-se fundamental introduzir um indicador que seja capaz de fornecer um valor fidedigno que classifique a previsão. Considerando *n* o número de observações, temos assim as seguintes medidas mais comuns:

- **Erro médio (*ME* – Mean error)**

$$ME = \frac{\sum_{i=1}^{n} e_i}{n} \tag{2.47}$$

- **Erro absoluto médio (*MAE* – Mean absolute error)**

$$MAE = \frac{\sum_{i=1}^{n} |e_i|}{n} \tag{2.48}$$

- **Erro quadrático médio (*MSE* – Mean squared error)**

$$MSE = \frac{\sum_{i=1}^{n} {e_i}^2}{n} \tag{2.49}$$

- **Raiz do erro quadrático médio (*RMSE* – Root mean squared error)**

$$RMSE = \sqrt{MSE} \tag{2.50}$$

Tendo as medidas acima calculadas, pode-se então fazer a comparação entre os modelos escolhidos, desde que os dados e a janela de previsão sejam os mesmos. Sendo o modelo com os valores dos erros mais baixos o que terá maior precisão.

2.8.2 Previsão com modelos *ARIMA/ ARMA*

O objetivo é prever um valor Y_{t+s}, sendo $s \geq 1$, supondo que todas as observações até ao momento são conhecidas. Denote-se por $\hat{Y}_t(s)$ a previsão feita usando um modelo *ARIMA(p,d,q)/ ARMA(p,q)*, no instante *t* para *s* passos após esse mesmo instante para uma determinada série *Y*. Sendo a equação da previsão dada por:

$$\hat{Y}_t(s) = \sum_{i=1}^{p+q} a_i \hat{Y}_t(s-1) \quad , \quad s > q \tag{2.51}$$

A solução geral terá a forma:

$$\hat{Y}_t(s) = \sum_{i=1}^{p+q} c_i^{(t)} f_i(s) \quad , \quad s > q - p - d \tag{2.52}$$

Onde $f_i(s)$ é função de *s*, *s=1, 2, ..., p+q*, e $c_i^{(t)}$ são coeficientes adaptados que dependem da origem da previsão *t* e são determinados por $\hat{Y}_t(s)$.

3. Machine Learning

Machine Learning (*ML*) ou aprendizagem de máquina é uma área científica que se concentra no estudo e na construção de modelos computacionais que podem "aprender" através de dados ou eventos, a realizar uma tarefa específica sem ser explicitamente programado. Um exemplo clássico é a deteção de *spam* em emails ou também conhecidos como filtros, em que o algoritmo[7] aprende a sinalizar/reconhecer emails como spam através de vários exemplos de emails com *spam* (muitas vezes sinalizados pelo utilizador) e também de exemplos de emails comuns que não são considerados *spam*.

Por outro lado, Burkov (2019) afirma no seu livro que as máquinas não aprendem, pois o que elas fazem é encontrar uma fórmula matemática que quando aplicada a uma coleção de *inputs* chamada "*training data*", produzem o *output* desejado. Essa fórmula matemática gera *outputs* corretos para muitos outros inputs (distintos do *training data*), sob a condição que esses inputs sejam provenientes da mesma ou similar distribuição estatística dos *inputs* do *training data*.

Para Tom Mitchell (1997), um programa de computador aprende com a experiência **E** em relação a alguma tarefa **T** e alguma medida de desempenho **P**, se o seu desempenho em **T**, medido por **P**, melhora com a experiência **E**. Considerando os filtros de spam dos emails teremos como tarefa **T** a deteção de spam em novos emails, a experiência **E** será o conjunto de emails fornecidos, e a medida **P** é arbitrária, por exemplo poderia ser o rácio de emails detetados corretamente como *spam.*

A aplicação de *machine learning* em grandes volumes de dados pode ajudar a descobrir padrões que muitas vezes não são imediatamente detetados, a isto é chamado *data mining* ou mineração de dados.

Data mining é a exploração e análise de grandes quantidades de dados, a fim de encontrar padrões nos mesmos. As técnicas proveem dos campos da estatística e computação, constituindo hoje em dia uma subárea da inteligência artificial (*Artificial Intelligence - AI*).

Os algoritmos de ML têm-se revelado cada vez mais importantes em diversas áreas. Com o crescente volume de dados, tarefas, negócios e entre outras variáveis que possam surgir é cada vez mais importante maximizar o tempo e a qualidade das decisões. No mundo dos mercados financeiros os investidores procuram cada vez mais melhorar as suas decisões, antecipar o mercado e rentabilizar os seus lucros. Com as técnicas antigas torna-se cada vez mais difícil acompanhar os mercados, pelo que os algoritmos de ML são extremamente importantes na área financeira.

Segundo Géron (2017), *ML* é excelente para: problemas que requerem imensa atenção humana ou muitas regras envolvidas, sendo que um algoritmo de ML consegue facilmente simplificar o código e melhorar a performance; problemas complexos em que não existem boas soluções recorrendo ao métodos tradicionais, os melhores algoritmos

[7] é uma sequência lógica, finita e definida de instruções que devem ser seguidas para resolver um problema ou executar uma tarefa. (TecMundo)

de *ML* conseguem encontrar uma solução; ambientes instáveis, os algoritmos de ML adaptam-se bem a novos dados; obter informação de problemas complexos e com um grande número de dados.

3.1 Tipos de *Machine Learning*/Aprendizagem

Existem diferentes tipos de (sistemas) *machine learning* (Liu, 2017), o que se torna útil classificá-los em diferentes categorias, conforme os algoritmos.

- Sejam ou não treinados (*training data*) com supervisão humana (*supervised learning*, *semi-supervised learning*, *unsupervised leaning* e *reinforcement learning*);
- Possam ou não aprender de forma incremental em tempo real (*online versus batch learning*);
- Comparem novos pontos de dados com pontos de dados conhecidos ou, em vez disso, detetam padrões nos dados de treino e construam um modelo preditivo (*instance-based versus model-based learning*).

Para o corrente estudo/tese irá se trabalhar com algoritmos de aprendizagem supervisionada, em que estes apenas irão aprender com os dados fornecidos (*batch learning*) e irão construir modelos preditivos de acordo com estes dados (*model-based*).

3.1.1 Aprendizagem supervisionada/*Supervised Learning*

Segundo Burkov (2019), na aprendizagem supervisionada, o conjunto de dados ou *training data*, é uma coleção de dados rotulados[8] $\{X_i, Y_i\}_{i=1}^{N}$, onde *N* é a dimensão da amostra. Cada elemento x_i é chamado vetor característico. Em cada um desses vetores, podem encontrar-se uma série de valores, designados por característica, mais conhecido na sua expressão em inglês por *feature* e é denotado por $x^{(j)}$, $j \in N$. Por exemplo se cada *x* representar uma pessoa, a primeira característica, $x^{(1)}$, pode conter o peso, a segunda característica, $x^{(2)}$, o género e assim por diante. É importante realçar que a posição da característica em *j* é a mesma em todo o vetor característico, ou seja, por exemplo se a característica do individuo *i* em *j* for o peso então o mesmo se aplica para o individuo *i+1*. O rótulo y_i pode ser um elemento pertencente a um conjunto finito de classes {1,2, ..., C} ou um número real, ou uma estrutura mais complexa, como um vetor, uma matriz, uma árvore ou um gráfico. Por exemplo se o conjunto de dados forem emails e o objetivo for a deteção de spam, então o y_i será {*spam, not-spam*}.

O principal objetivo da aprendizagem supervisionada é de usar um conjunto de dados para produzir um modelo, em que o input seja um vetor característico e o output seja alguma informação deduzida através do vetor característico. Por exemplo, se o conjunto de dados conter informação de saúde de algumas pessoas e o input for vetor

[8] São exemplos/dados em que já se conhece os valores do output.

característico com a descrição de cada pessoa, então o output pode ser a probabilidade da pessoa ter alguma doença (câncer, diabetes, entre outras doenças).

A aprendizagem supervisionada pode ser dividida em duas categorias distintas:

- Classificação/*Classification*

A classificação atribui automaticamente um rotulo/etiqueta aos exemplos não rotuladas (Burkov, 2019), sendo a deteção de spam em emails um bom exemplo. Nos problemas de classificação são usados algoritmos de classificação, em que os inputs são uma coleção de exemplos rotulados que produzem um modelo que pode ter como input dados não rotulados e ainda gerar diretamente um rótulo ou output de um número que possa ser usado pelos analistas de dados de forma a deduzir o rótulo facilmente.

O rótulo é um conjunto finito de classes, caso o tamanho do conjunto de classes for dois como por exemplo “doente” ou “saudável”, designa-se por classificação binária ou binomial dependendo do autor. Caso o número de classes seja maior que dois, estamos perante uma classificação multinominal.

- Previsão/*Prediction*

A previsão consiste em prever um rótulo de valor real através de um conjunto de exemplos não rotulados, a esse rotulo geralmente chama-se de *target*, que traduzido pode ser interpretado como a variável objetivo *y*. Um bom exemplo é a estimativa do preço das casas com base em características próprias da mesma (área, número de quartos, localização, entre outras).

Uma vez conhecida a divisão dos algoritmos supervisionados, apresenta-se de seguida alguns dos seus principais algoritmos:

- *K-Nearest Neighbors*
- *Linear Regression*
- *Logistic Regression*
- *Support Vector Machines (SVMs)*
- *Decision Trees e Random Forest*
- *Neural Networks*

3.2 Principais algoritmos

Neste subcapítulo serão abordados três algoritmos de *Machine Learning* que serão usados para o presente estudo. A escolha destes baseou-se não só na sua popularidade, mas também na sua eficácia e robustez na aprendizagem. A abordagem para a explicação dos algoritmos, será baseada no livro *The Hundred-Page Machine Learning* (Burkov, 2019) e no artigo *A tutorial on support vector regression* (Smola, et al., 2003).

3.2.1 Suport Vector Regression – SVR

O algoritmo *SV* é uma generalização não linear do algoritmo *Portrait Generalized*, tendo este sido desenvolvido na Rússia nos anos sessenta por Vapnik & Lerner (1963), Vapnik & Chervonenkis (1964).

O algoritmo de *SVM* tal como conhecemos hoje, foi desenvolvida por Vapnik no seu laboratorio (AT&T Bell) juntamente com os seus colaboradores (Boser, Guyon e Kapnik 1992, Guyon, Boser e Kapnik 1993, Cortes e Vapnik, 1995, Schëlkopf, Burges e Vapnik 1995, 1996, Vapnik, Golowich e Smola 1997). Devido ao contexto industrial a que estes estavam inseridos, a pesquisa em *SV* teve, até o momento, uma orientação sólida para aplicações do mundo real. Inicialmente estes foram usados para o reconhecimento óptico de caracteres (*OCR*). Tornando-se em pouco tempo comparáveis com os melhores sistemas disponíveis na altura para tarefas de reconhecimento de objetos e OCR. Em termos de regressão e previsão de series temporais, excelentes resultados foram também obtidos.

De acordo com Smola, *et al*. (2003), dado um conjunto de dados $\{x_i, y_i\}_{i=1}^{N}$, o objetivo na regressão $\varepsilon - SV$ (Vapnik, 1995) é de encontrar uma função $g(x)$ que tenha para todos os dados de treino, no máximo um desvio ε dos alvos realmente obtidos y_i e ao mesmo tempo seja o mais plano possível. Ou seja, a função não se importa com os erros desde que estes sejam menores que ε e nunca maiores. Este tipo de restrições pode ser importante, por exemplo para delimitar o limite de perdas nas negociações de ações.

Numa primeira fase, descreve-se o caso de uma função linear *g:*

$$g(x) = \langle w, x \rangle + b, \qquad w \in X \; e \; b \in \mathbb{R} \tag{3.1}$$

onde $\langle w, x \rangle$ é o produto escalar.

No caso da função linear, pretende-se minimizar a norma de w, $\|w\|^2 = \langle w, x \rangle$. Podendo este problema ser formulado como um problema de otimização convexa:

$$\min \frac{1}{2} \|w\|^2 \tag{3.2}$$

Sujeito a

$$\begin{matrix} y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{matrix} \tag{3.3}$$

As restrições acima (eq. 3.3) garantem que existe uma função g que aproxima todos os pares (x_i, y_i) com precisão ε, ou seja, que o problema de otimização convexa é viável. Por vezes, este pode não ser o caso ou pode-se querer permitir alguns erros. Analogamente à função de perda de "margem branda" (Bennett e Mangasarian 1992), usada em máquinas *SV* por Cortes e Vapnik (1995), pode-se introduzir variáveis de folga ζ_i e $\zeta^*{}_i$ para lidar com restrições inviáveis do problema de otimização (eq. 3.2). Chegando assim, à formulação declarada em Vapnik (1995):

$$\min \frac{1}{2}\|w\|^2 + C\sum(\zeta_i + \zeta^*{}_i) \tag{3.4}$$

Sujeito a

$$\begin{matrix} y_i - \langle w, x_i \rangle - b \leq \varepsilon + \zeta_i \\ \langle w, x_i \rangle + b - y_i \leq \varepsilon + \zeta^*{}_i \\ \zeta_i\,, \zeta^*{}_i \geq 0 \end{matrix} \tag{3.5}$$

Onde a constante $C > 0$ determina o *trade-off* entre a curva de g e a tolerância perante os valores maiores que ε. Corresponde assim, a lidar com uma função perda $|\zeta|_\varepsilon$ chamada $\varepsilon - insensitive$, sujeita a:

$$\begin{matrix} 0, \quad se\ |\zeta| \leq \varepsilon \\ |\zeta| - \varepsilon, \qquad c.c \end{matrix} \tag{3.6}$$

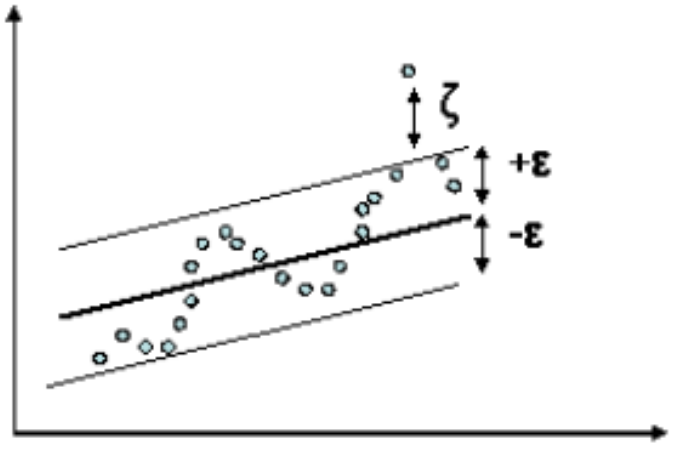


Figura 3: Exemplificação de um problema de SVR em que se ajusta um tubo com raio ε aos dados e variáveis de folga positivas ζ_i que medem os pontos localizados fora do tubo

Acontece que, na maioria dos casos, o problema de otimização (eq. 3.5) pode ser resolvido facilmente em sua formulação dupla. Além disso, a formulação dupla fornece a chave para estender o SVM para funções não lineares. Portanto, usaremos um método de dualização padrão utilizando multiplicadores Lagrange (Fletcher, 1989).

A ideia principal é construir uma função de Lagrange a partir da função objetivo, a qual se dá o nome de função objetivo primordial, e construir as restrições correspondentes, introduzindo um conjunto duplo de variáveis. Pode ser demonstrado que esta função possui um ponto de sela em relação às variáveis primárias e duplas na solução (Mangasarian (1969), McCormick (1983) e Vanderbei (1997)).

$$\begin{aligned} L \coloneqq \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\left(\zeta_i + \zeta^*{}_i\right) - \sum_{i=1}^{l}\left(\eta_i\zeta_i + \eta_i^*\zeta^*{}_i\right) \\ - \sum_{i=1}^{l}\alpha_i(\varepsilon + \zeta_i - y_i + \langle w, x_i\rangle + b) \\ - \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \zeta^* - y_i - \langle w, x_i\rangle - b) \end{aligned} \tag{3.7}$$

Sujeito a

$$\alpha_i, \alpha_i^*, \eta_i, \eta_i^* \geq 0 \tag{3.8}$$

onde L é a *Lagrangiana*, e $\eta_i, \eta_i^*, \alpha_i, \alpha_i^*$ são os multiplicadores de *Lagrange*.

Pela condição do ponto de sela, as derivadas parciais de L com relação às variáveis primárias $(w, b, \zeta_i, \zeta^*{}_i)$ precisam ser eliminadas, com vista a otimizar a função.

$$\partial_b L = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \tag{3.9}$$

$$\partial_w L = w - \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)x_i = 0 \tag{3.10}$$

$$\partial_{\zeta_i} L = C - \alpha_i - \eta_i = 0 \tag{3.11}$$

$$\partial_{\zeta^*} L = C - \alpha_i^* - \eta_i^* = 0 \tag{3.12}$$

Substituindo as expressões acima (Eq. 3.9, 3.10, 3.11 e 3.12) na equação 3.7 obtém-se o problema de otimização dupla.

$$\begin{aligned} \text{maximizar}\Bigg\{ -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)\langle x_i, x_j\rangle \\ - \varepsilon\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) + \sum_{i=1}^{l} y_i\,(\alpha_i - \alpha_i^*)\Bigg\} \end{aligned} \tag{3.13}$$

sujeito a $\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)$ e $\alpha_i, \alpha_i^* \in [0, C]$

Uma vez eliminadas as variáveis duplas ηi, η i na equação 3.13 através das condições na equação 3.11 e 3.12, estas podem ser reformuladas como $\eta_i = C - \alpha_i$ e $\eta_i^* = C - \alpha_i^*$.

A equação 3.10 também pode ser reescrita da seguinte forma:

$$w = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i \tag{3.14}$$

Podendo assim a função *g* inicial (3.1) ser reescrita:

$$g(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) \langle x_i, x \rangle + b \tag{3.15}$$

A isto é chamada a expansão do SVM, ou seja, *w* pode ser completamente descrito como uma combinação linear dos padrões de treino x_i.

De certo modo, a complexidade da representação de uma função por *SV's* é independente da dimensionalidade do espaço de entrada *X* e depende apenas do número de *SV's*. Além disso, observa-se que o algoritmo completo pode ser descrito em termos de produtos escalares entre os dados. Mesmo ao avaliar *g (x)*, não é necessário calcular *w* explicitamente. Essas observações serão úteis para a formulação de uma extensão não linear.

Através das condições de Karush-Kuhn-Tucker (*KKT*) (Karush 1939, Kuhn e Tucker 1951) pode-se encontrar a solução de *b* na equação 3.15. O que quer dizer que, no ponto da solução, o produto entre variáveis duplas e as suas restrições devem desaparecer.

$$\begin{aligned} \alpha_i(\varepsilon + \zeta_i - y_i + \langle w, x_i \rangle + b) = 0 \\ \alpha_i^*\left(\varepsilon + \zeta^*{}_i + y_i - \langle w, x_i \rangle - b\right) = 0 \end{aligned} \tag{3.16}$$

e

$$\begin{aligned} (C - \alpha_i)\zeta_i = 0 \\ (C - \alpha_i^*)\zeta^*{}_i = 0 \end{aligned} \tag{3.17}$$

Pode-se assim tirar as seguintes conclusões:

- Apenas as amostras (x_i, y_i) com os correspondentes $\alpha_i, \alpha_i^* = C$ ficam fora do tubo $\varepsilon - insensitive$;
- $\alpha_i, \alpha_i^* = 0$, ou seja, as variáveis duplas nunca podem ser simultaneamente diferentes de zero.

$$\begin{cases} \varepsilon - y_i + \langle w, x_i \rangle + b \geq 0 \quad e \quad \zeta_i = 0 \quad , se\ \alpha_i < C & (3.18) \\ \varepsilon - y_i + \langle w, x_i \rangle + b \leq 0 \qquad\qquad\qquad\quad se\ \alpha_i > C & (3.19) \end{cases}$$

Em conjunto com uma análise análoga em α_i^*, tem-se

$$max\{-\varepsilon + y_i - \langle w, x_i\rangle | \alpha_i < C \; ou \; \alpha_i^* > 0\} \leq b \leq min\{-\varepsilon + y_i - \langle w, x_i\rangle | \alpha_i > 0 \; ou \; \alpha_i^* < C\} \tag{3.20}$$

Caso α_i ou $\alpha_i^* \in (O, C)$ as desigualdades tornam-se igualdades.

Segue que apenas para $|g(x) - y_i| \geq \varepsilon$ os multiplicadores de Lagrange podem ser diferentes de zero, em outras palavras, para todas as amostras dentro do tubo ε (região sombreada na Figura 3) o α_i, α_i^* desaparecem: para $|g(x) - y_i| < \varepsilon$ o segundo fator em (3.14) é diferente de zero, por isso α_i, α_i^* deve ser zero, de modo que as condições de *KKT* sejam satisfeitas. Portanto, temos uma expansão dispersa de *w* em termos de x_i (ou seja, não precisamos de todos os x_i para descrever *w*).

Uma vez que nem sempre se consegue separar linearmente os dados, o próximo passo será tornar o algoritmo de SVM não linear.

A não linearidade pode ser conseguida simplesmente através do pré-processamento dos padrões de treinamento x_i por mapeamento Φ: $X \to \mathcal{F}$ em algum espaço $\mathcal{F}$, conforme descrito em Aizerman, Braverman and Rozonoér (1964) & Nilsson (1965) e, em seguida, aplicando o algoritmo de regressão de SV.

Como observado nos passos anteriores, o algoritmo de SV apenas depende dos produtos vetoriais entre os padrões de x_i. Portanto, basta conhecer $K(x, x') \coloneqq \langle \Phi(\mathrm{x}), \Phi(\mathrm{x}')\rangle$ em vez de explicitamente Φ, o que permite restabelecer o problema de otimização do SVM:

$$maximizar \left\{ \begin{array}{c} -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(x_i - x_j) \\ -\varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) \end{array} \right\} \tag{3.21}$$

sujeito a $\sum_{i=1}^{l}(\alpha_i - \alpha_i^*)$ e $\alpha_i, \alpha_i^* \in [0, C]$. A função K é chamada de *kernel*.

Da mesma forma, a expansão das equações 3.14 e 3.15 podem ser escritas como

$$w = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\Phi(x_i) \tag{3.22}$$

$$f(x) = \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)k(x_i, x) + b \tag{3.23}$$

Uma exemplificação de uma função Kernel correspondente ao produto escalar em algum espaço característico $\mathcal{F}$ é a função Kernel Hilbert Space (RKHS)[9]. As restantes funções podem ser consultadas no artigo de Smola, et al (2003) - A Tutorial on Support Vector Regression.

3.2.2 Regressão Linear

A Regressão Linear (*Linear Regression – LR*) é um dos mais populares algoritmos de *ML*, sendo a sua aprendizagem baseada em combinações lineares das variáveis de entrada (*input*).

No modelo *LR* tem-se uma coleção de dados rotulados $\{x_i, y_i\}_{i=1}^{N}$, onde N é o tamanho da amostra, x_i é um vetor característico D-dimensional dos exemplos *i=1,…, N* , $y_i \in \mathbb{R}$. Sendo cada característica $x_i{}^j, j = 1, \dots, D$ um número real.

O objetivo é criar um modelo $f(x)$ como combinação linear das características dos exemplos *x*:

$$f(x) = wx + b \tag{3.23}$$

onde w são os pesos, sendo este um vector D-dimensional dos parâmetros e b é um número real.

A *LR* é usada para prever um *y* desconhecido através de um *x* dado, $y \leftarrow f(x)$. Tendo em conta que dois modelos parametrizados por dois pares diferentes (*w*,*b*) produzem resultados diferentes quando aplicados aos mesmos exemplos, ou seja o mesmo conjunto de dados, será necessário encontrar valores ótimos (*w**,*b*) para que assim se possa encontrar as melhores previsões.

Para encontrar os valores ótimos para w*e b*, define-se uma função objetivo conhecida em *ML* como **função custo**/***cost function***, ou seja, minimizamos a seguinte expressão:

$$\frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 \tag{3.24}$$

onde $(f_{w,b}(x_i) - y_i)^2$ é chamada **função perda**/***loss function*** que mede a penalidade sobre as classificações incorretas da amostra *i*, sendo a expressão também conhecida como erro quadrático.

[9] Um Kernel do tipo $k(x, x') = k\langle w, x\rangle$ definido num espaço dimensional infinito de Hilbert, com $k(t) = \sum_{n=0}^{\infty} a_n t^n$ é admissível se e só se $a_n > 0$.

É de notar que na regressão linear, a *cost function* é o risco empírico, ou seja, a média de todas as penalidades obtidas ao aplicar o modelo aos dados do *training data.*

De forma a encontrar o mínimo de uma função recorre-se ao gradiente descendente/*gradient descent*[10]. Do modelo de regressão linear (eq. 3.23) não se sabe os valores ótimos de w e b, pelo que se minimiza o erro médio quadrático:

$$l = \frac{1}{N}\sum_{i=1}^{N}\big(y_i - (wx_i + b)\big)^2 \tag{3.25}$$

Calcula-se as derivadas parciais para todos os parâmetros:

$$\frac{\partial l}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} -2x_i\left(y_i - (wx_i + b)\right) \tag{3.26}$$

$$\frac{\partial l}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} -2\left(y_i - (wx_i + b)\right) \tag{3.27}$$

Igualando as derivadas parciais a zero, obtêm-se:

$$w_i \leftarrow \eta\frac{-2x_i(y_i - (w_{i-1}x_i + b_{i-1}))}{N} \tag{3.28}$$

$$b_i \leftarrow \eta\frac{-2(y_i - (w_{i-1}x_i + b_{i-1}))}{N} \tag{3.29}$$

onde η controla a velocidade dos passos da aprendizagem, w_i e b_i denotam os valores de w e b depois de usar (x_i, y_i) para a atualização.

É de notar-se que o algoritmo de gradiente descendente é sensível relativamente a escolha dos passos de η, e este revela-se também lento em grandes amostras. Mas tendo em conta ao desenvolvimento das tecnologias, várias melhorias foram propostas (Burkov, 2019). Uma destas propostas é o gradiente **descendente estocástico**/**stochastic gradient descendt (SGD)** que é uma versão que acelera o processo de computação aproximando o gradiente utilizando pequenos subconjuntos dos dados de treino.

[10] Gradiente descendente é um algoritmo interativo de otimização para encontrar o mínimo de uma função. Para encontrar um mínimo local de uma função usando o gradiente descendente, começa-se de um ponto aleatório w e segue-se proporcionalmente em direção ao gradiente negativo da função do ponto em questão, estes passos são controlados pelo *learning rate* η.

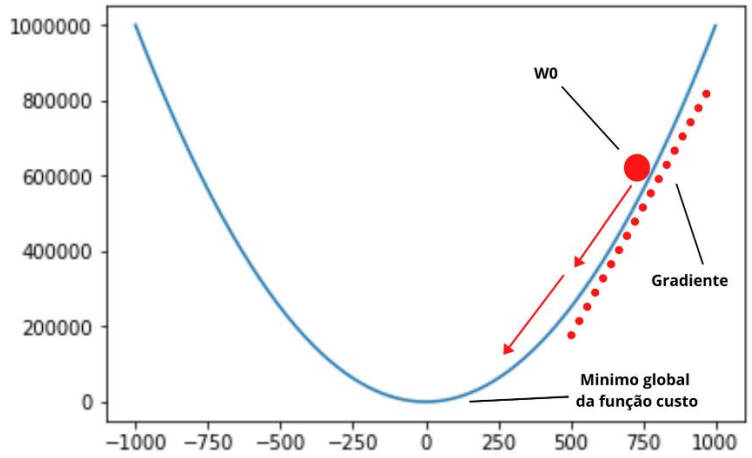


Figura 4:Simulação do funcionamento do gradiente descendente através de uma função convexa

Como forma de evitar o overfit dos dados e diminuir a complexidade do modelo, recorre-se ao método de regularização[11], que consiste em modificar a função objetivo/custo através de penalizações em que o valor das mesmas torna-se alto quando o modelo é demasiado complexo. Os dois tipos de regularização mais usados são **regularização L1** e **regularização L2**, sendo que para o caso da regressão linear será utilizada a regularização L2, uma vez que se pretende maximizar a performance do modelo e porque o L2 é diferenciável sendo assim possível a usar o gradiente descendente para otimizar a função custo.

$$\min_{w,b} \alpha\|w\|^2 + \frac{1}{N}\sum_{i=1}^{N}(f(x_i) - y_i)^2 \tag{3.30}$$

onde $\|w\|^2 \stackrel{\text{def}}{=} \sum_{j=1}^{D}(w^{(j)})^2$ e α é um híper-parâmetro que controla a importância da regularização.

3.2.3 Árvores e florestas de decisão

Uma árvore de decisão (*Random Tree and Random Forest – RF*) é um gráfico acíclico que pode ser usado para a tomada de decisões. Em cada nó da ramificação do gráfico, uma característica específica *j* do vetor característico é examinada. Se o valor da característica estiver abaixo de um limite específico, a ramificação esquerda será seguida, caso contrário segue-se a ramificação direita. A medida que se alcança o nó folha é tomada uma decisão sobre a classe a qual o exemplo pertence.

Numa árvore de decisão tem-se uma coleção de exemplos rotulados $\{x_i, y_i\}_{i=1}^{N}$, sendo esses rótulos pertencentes ao conjunto $\{0, 1\}$. Pretende-se construir uma árvore de decisão que possa prever a classe de um exemplo, dado um vetor característico.

11

Para a formulação da RF, considera-se neste caso o algoritmo **ID3**. Em que se constrói um modelo não paramétrico:

$$f_{ID3}(x) = P(y = 1|x) \tag{3.31}$$

Sendo o critério de otimização, neste caso a *average log-likelihood:*

$$\frac{1}{N}\sum_{i=1}^{N} y_i \, lnf_{ID3}(x_i) + (1 - y_i) \ln (1 - f_{ID3}(x_i)) \tag{3.32}$$

onde f_{ID3} é a árvore de decisão.

Veja-se agora o funcionamento do algoritmo de aprendizagem ID3. Inicialmente, a RF começa com um nó inicial que contém todos os exemplos, $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$.

Começa-se com um modelo constante

$$f_{ID3}^{\mathcal{S}} = \frac{1}{|\mathcal{S}|}\sum_{(X,Y)\in\mathcal{S}} y \tag{3.33}$$

De seguida, pesquisa-se todas características j=1, …, D e todos os limites t, divide-se ainda o conjunto $\mathcal{S}$ em dois subconjuntos: $\mathcal{S}_- = \{(x,y)|(x,y) \in \mathcal{S}, x^{(j)} < t\}$ e $\mathcal{S}_+ = \{(x,y)|(x,y) \in \mathcal{S}, x^{(j)} \geq t\}$.

Esses dois novos subconjuntos devem ir para dois novos nós folha, e avalia-se para todos os pares possíveis (*j*,*t*) a qualidade da divisão feita. Por ultimo, escolhe-se os melhores valores (*j*,*t*), divide-se $\mathcal{S}$ em $\mathcal{S}_+$ e $\mathcal{S}_-$, formando dois novos nós folha e continua-se recursivamente em $\mathcal{S}_+$ e $\mathcal{S}_-$ ou encerra-se se nenhuma divisão produzir um modelo melhor que o atual.

De forma a avaliar-se a qualidade da divisão, recorre-se ao critério chamado entropia/*entropy*. Este critério mede a incerteza sobre uma variável aleatória, atingindo o seu máximo quando todos os valores das variáveis forem equiprováveis e atingindo o seu mínimo quando a variável aleatória só pode ter um valor.

A *entropia* de um conjunto de exemplos $\mathcal{S}$ é dada por:

$$H(\mathcal{S}) = -f_{ID3}^{\mathcal{S}} \, lnf_{ID3}^{\mathcal{S}} - \left(1 - f_{ID3}^{\mathcal{S}}\right) \ln (1 - f_{ID3}^{\mathcal{S}}) \tag{3.34}$$

Sendo que quando dividimos um conjunto de exemplos por uma determinada característica *j* e um limite *t*, a *entropia* dessa divisão, H($\mathcal{S}_+$, $\mathcal{S}_-$), é a soma ponderada de duas *entropias:*

$$H(\mathcal{S}_+, \mathcal{S}_-) = \frac{|\mathcal{S}_-|}{|\mathcal{S}|} H(\mathcal{S}_-) + \frac{|\mathcal{S}_+|}{|\mathcal{S}|} H(\mathcal{S}_+) \tag{3.35}$$

Pela equação anterior, tem-se que no ID3, em cada etapa de cada nó folha encontra-se uma divisão que minimiza a *entropia*. Intuitivamente, a *entropia* atinge o seu mínimo ,0, quando todos os exemplos do conjunto $\mathcal{S}$ têm o mesmo rótulo. Do outro lado, tem-se que atinge o seu máximo ,1, quando metade dos exemplos em $\mathcal{S}$ estão rotulados com 1.

Tendo em conta que no ID3, a decisão de dividir o conjunto de dados em cada iteração é local, ou seja, não depende de divisões futuras, o algoritmo não consegue garantir uma ótima solução.

3.3 Subvalorização, sobrevalorização e viés

Quando se cria um modelo, o objetivo principal é que o mesmo explique de forma precisa a variabilidade dos dados, faça uma boa previsão das variáveis de interesse e que também se possa implementar o mesmo modelo em outros dados, mas nem sempre isso acontece. Caso o modelo cometa muitos erros no *training data*, diz-se que o modelo tem um viés muito elevado ou que o modelo está subvalorizado (*underfit*). O que quer dizer que o modelo é demasiado simples para aprender ou tracejar a estrutura dos dados. Podem existir diversas razões para subvalorização do modelo, mas as mais comuns são:

- O modelo é demasiado simples para os dados. Por exemplo no caso de um modelo de regressão linear em que na sua representação gráfica em vez dos valores preditos tracejarem uma linha reta traçam uma linha curva;
- Os parâmetros não fornecem informação suficiente. Por exemplo, pretende-se prever se um paciente tem ou não cancro de próstata, e as variáveis disponíveis são altura, pressão arterial e cor do cabelo. Claramente estas variáveis não serão bons preditores, pelo que o modelo não será capaz de aprender as relações existentes entre estas variáveis e o rotulo.

Uma solução para este tipo de problema é tentar criar um modelo mais complexo ou criar *engineer features* que tenham um grande poder preditivo.

Por outro lado, podemos também ter a sobrevalorização (*overfitting*) do modelo, em que neste caso o modelo é demasiado complexo relativamente a quantidade e ruido dos dados. Este comporta-se bem na amostra de dados (*training data*), mas não representa/generaliza a realidade. A grande quantidade de variáveis e um número pequeno de *training exemples* também causam o *overfitting* do modelo.

O viés ou erro sistemático em estatística, mede o quão bem o modelo consegue ajustar-se aos dados. Um modelo diz-se que tem um baixo viés se o mesmo prever de forma correta os rótulos dos dados de treinamento (*training data*), caso contrário diz-se que tem um alto viés.

4. Estudo Empírico

Os índices bolsistas, são um dos principais indicadores dos movimentos do mercado bolsista, uma vez que estes espelham na maioria das vezes a tendência do mercado. Tendo em conta a representatividade e importância dos índices, neste capítulo será feita a análise e previsão de um dos mais importantes índices bolsistas do mercado americano, o *S&P 500*.

Desde 1957 que o índice *S&P 500* foi introduzido pela Standard & Poor's com o objetivo de seguir o valor de mercado das quinhentas maiores empresas listadas nos índices *New York Stock Exchange* e *NASDAQ Composite.*

Os dados relativamente a série em questão são diários (incluindo os feriados), mas não englobam os fins de semana. A amostra contém cinco mil e vinte três (5023) observações, sendo esta composta por quatro variáveis continuas (preço de abertura e fecho, preço mais alto e baixo). Estes dados serão analisados numa janela temporal de 3 de Janeiro de 2000 até 03 de Abril de 2019.

Para os testes estatísticos e intervalos de confiança será fixado ao longo de todo o estudo um nível de significância, $\alpha = 5\%$.

Tendo em conta que se pretende comparar os métodos econométricos tradicionais com os novos métodos propostos, este capítulo será dividido em duas partes, na primeira será feita a análise e previsão no programa *Eviews* e na segunda parte no programa *Python*.

4.1 Eviews

O programa *Eviews* foi lançado em 1994, sendo este um programa estatístico usado principalmente para análises econométricas.

Tendo em conta que, o principal objetivo deste estudo é prever os preços de fecho da série em questão, vai-se trabalhar apenas com a variável preço de fecho, intitulada "close". Tendo em conta que os feriados e os acontecimentos inesperados (como por exemplo o fecho da bolsa de valores) não apresentam valores, sendo estes considerados pelo programa como *Missing Values* (NA), recorreu-se a interpolação linear [12]como forma de preencher as perdas de informação.

[12] É uma aproximação linear que preenche os Nas através de valores não ausentes, ou seja, baseia-se no valor anterior e no valor seguinte. O valor interpolado é calculado da seguinte forma:
$IV_{Lin} = (1 - \lambda)P_{i-1} + \lambda P_{i+1}$, onde P_{i-1} é o valor anterior ao NA, P_{i+1} é o próximo valor ao NA e λ é a posição relativa do NA dividida pelo número total de Nas existentes na variável.

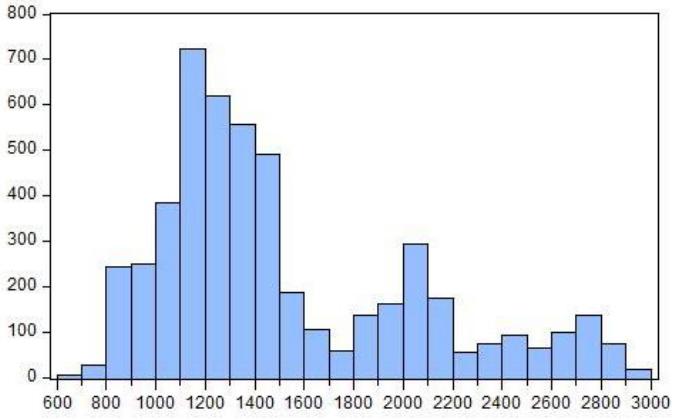
4.1.1 Análise gráfica e estatísticas descritivas

Pela Figura 5, pode-se constatar que a série é provavelmente não estacionária, uma vez que não é possível estabelecer uma tendência geral da série – a movimentação do índice é bastante irregular e não apresenta sinais de sazonalidade. Os movimentos inconstantes da série, resultados de uma grande variabilidade, indiciam uma média e variância inconstantes, o que contribui ainda mais para a suspeita de não estacionariedade da série em questão. Verifica-se ainda entre 2007 e 2008 uma queda abrupta dos preços de fecho, o que se deveu possivelmente a crise financeira desencadeada na mesma altura nos Estados Unidos da América (EUA). Esta crise desencadeou-se devido a falência do tradicional banco de investimentos *Lehman Brothers*, que em efeito dominó, outras grandes instituições também faliram, desencadeando uma grande crise internacional/mundial.

Figura 5: Gráfico de linhas dos preços de fecho do S&P500

Após a análise gráfica, procede-se com a análise das estatísticas descritivas. Verifica-se pela Tabela 1 (ver página seguinte), que a série apresenta uma distribuição assimétrica positiva e platicúrtica, uma vez que o coeficiente de assimetria é maior que zero e o coeficiente do momento de curtose é inferior a 3. Pelo teste de *Jarque-Bera*, rejeita-se a hipótese nula de que a série segue uma distribuição normal, observando-se ainda um padrão trimodal no histograma da variável em estudo.

Tabela 1: Estatísticas descritivas S&P500

Estatísticas	**S&P500**
Média	1520.453
Mediana	1342.54
Mínimo	676.530
Máximo	2930.750
Desvio Padrão	526.053
Assimetria	0.983
Curtose	2.994
Jarque-Bera	809.104
p-value	0.000
Histograma	

4.1.2 Estacionariedade do índice S&P 500

Para o estudo da estacionariedade serão considerados três testes: os testes de raiz unitária *Argumented Dickey-Fuller* (*ADF*) e *Phillips-Perron* (*PP*) e o teste de estacionariedade *Kwiatkowski-Phillips-Achidt-Shin* (*KPSS*). A aplicação dos três testes permite que, no caso de dois serem contraditórios, recorra-se à conclusão do terceiro para desempatar a decisão.

Os testes serão apresentados na tabela abaixo, sendo os valores tabelados correspondentes ao valor da estatística de teste. Note-se que a hipótese nula só será rejeitada se o valor da estatística de teste for superior em módulo aos valores críticos.

Nos testes de ADF e PP (Tabela 2) não se rejeita a hipótese nula de que a série é não estacionária, pois em módulo as estatísticas de teste são menores que os valores críticos. No que diz respeito ao teste de KPSS, rejeita-se a hipótese de estacionariedade, ou seja, chega-se a mesma conclusão de que a série é não estacionária. Comprova-se assim que, a série é não estacionária como já se suspeitava na análise gráfica.

Tabela 2: Testes de raiz unitária ADF, PP e KPSS

		Testes		
		ADF	**PP**	**KPSS**
		0.756	**0.929**	**6.237**
Valores críticos	1%	-3.431	-3.431	0.739
	5%	-2.861	-2.861	0.463
	10%	-2.567	-2.567	0.347

Uma vez que a série é não estacionaria e o objetivo principal é de prever os valores futuros dos preços de fecho, que só é possível caso a série seja estacionaria, procede-se com a diferenciação dos *logaritmos* da série (obtenção dos *log-retornos* da série dos preços) como forma de obter a estacionariedade da mesma. A seguir aplicam-se os testes de raiz unitária e de estacionariedade para os *log-retornos*.

4.1.2.1 Análise gráfica e estatísticas descritivas – Diferença dos logaritmos

Observa-se pela Figura 6, que houve uma estabilização do valor médio mas em termos de variabilidade dos dados observamos clusters de volatilidade, sendo isto último, um dos factos estilizados das séries financeiras.

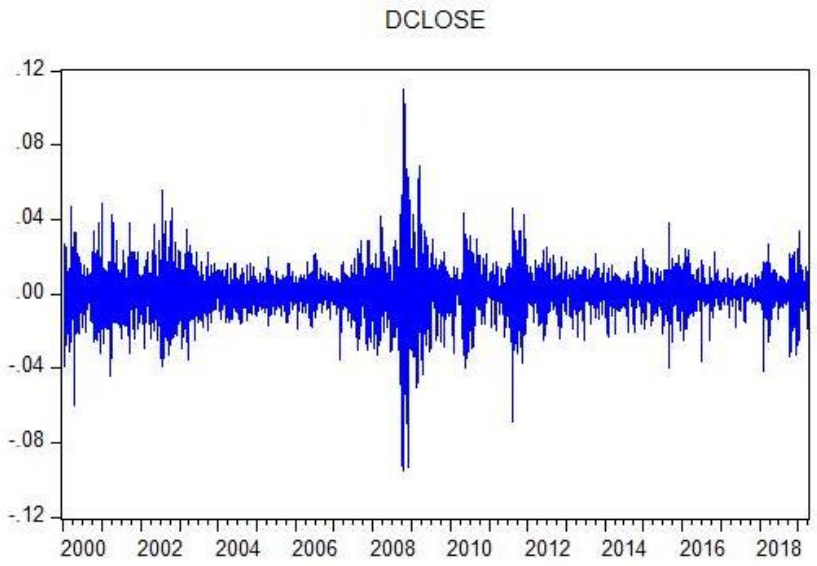
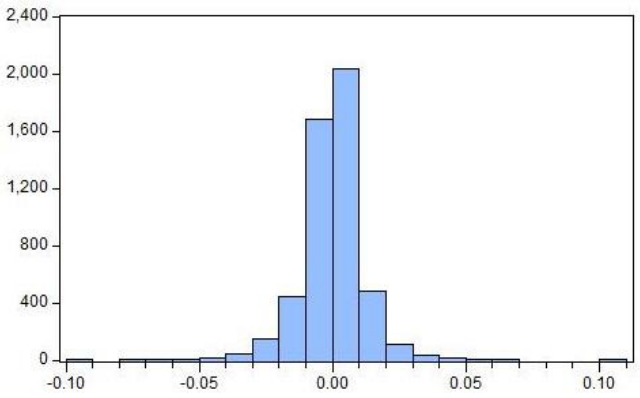


Figura 6: Gráfico de linhas do índice S&P500 logaritmizado

Pela Tabela 3, verifica-se que a série dos *log-retornos* apresenta uma distribuição assimétrica negativa e leptocúrtica. A semelhança da série dos preços de fecho, continua-se a rejeitar a hipótese de normalidade dos dados pelo teste de *Jarque-Bera*.

Tabela 3: Estatísticas descritivas da série diferenciada e logaritmizada

Estatísticas	**S&P500**
Média	0.000135
Mediana	0.000470
Mínimo	-0.094695
Máximo	0.109572
Desvio Padrão	0.011678
Assimetria	-0.206704
Curtose	12.16236
Jarque-Bera	
p-value	0.000
Histograma	

Pelos resultados dos testes feitos e apresentados na Tabela 4, chega-se à conclusão, quer pelo teste de ADF, PP ou KPSS de que não existem evidencias estatísticas que levem a rejeição da estacionariedade da série. Sendo assim obtida a estacionariedade dos *log-retornos* do índice *S&P 500*. Conclui-se então que estamos perante uma séries temporal integrada de ordem um.

Tabela 4: Testes de raiz unitária

		Testes		
		ADF	**PP**	**KPSS**
		-54.854	**-76.109**	**0.3363**
Valores criticos	1%	-3.4314	-3.4314	0.7390
	5%	-2.8619	-2.8619	0.4630
	10%	-2.5670	-2.5670	0.3470

4.1.3 Síntese

Numa primeira fase, ao analisar-se a série original verificou-se que esta apresentava movimentos inconstantes e sem sinais de sazonalidade, tendo a série uma distribuição assimétrica positiva e platicúrtica. Aplicando o teste de Jarque-Bera rejeitou-se a hipótese de normalidade e quando aplicado os testes de raiz unitária chegou-se a conclusão de que a série é não estacionária.

Numa segunda fase, aplicou-se o método de diferenciação de primeira ordem e de logaritmização da série, de modo a obter a estabilização da mesma. Pelas estatísticas descritivas verificou-se que ao contrário da série original, a nova série apresentava uma distribuição assimétrica negativa e leptocúrtica, mais uma vez rejeitou-se a hipótese de normalidade da série. Mas ao contrário da série não diferenciada, pelos testes de raiz unitária concluiu-se que a série dos log-retornos é estacionária.

4.1.4 Identificação e estimativa dos modelos econométricos

Uma vez obtida a estacionariedade da série, segue-se com a escolha do modelo que melhor se adequa aos dados. Sendo que a construção e a utilização destes modelos não são baseadas do ponto de vista de qualquer modelo teórico subjacente ao comportamento da variável levando a que, em alguns casos, os modelos que aparentam melhor se ajustarem aos dados, não sejam aqueles que levam a melhores previsões, atribuindo assim um caracter subjetivo à estimação dos mesmos.

Para a escolha do modelo que melhor se adequa aos dados e a previsão, iremos seguir os seguintes passos segundo a metodologia de Box & Jenkins (1976):

1. **Identificação e escolha do modelo**

Nesta primeira fase determina-se a ordem dos parâmetros *p* e *q* dos possíveis modelos ARMA/ARIMA através de métodos gráficos (ACF, PACF). Após a identificação segue-se com a escolha do modelo através dos critérios de informação já abordados na secção 2.6, sendo o modelo escolhido o que apresentar os valores mais baixos.

2. **Estimativa do modelo**

Estima-se os parâmetros dos modelos identificados na primeira etapa. A estimação pode ser feita através do método dos mínimos quadrados ou através do método da máxima verosimilhança (Cambridge University Press, 2008). Para este estudo, recorre-se ao método dos mínimos quadrados, também designado por OLS.

3. **Validação do modelo**

Nesta etapa, faz-se a averiguação se o modelo especificado e estimado é ou não adequado aos dados. A averiguação é feita através da análise de resíduos, em que o principal objetivo é verificar se existem evidencias estatísticas de heterocedasticidade e de dependência linear entre os resíduos.

Tendo em conta o que foi descrito anteriormente, temos que os coeficientes de autocorrelação serão considerados significativos se os seus valores se encontrarem fora do intervalo de confiança, Brooks (2014). O intervalo de confiança pode ser calculado da através da fórmula seguinte:

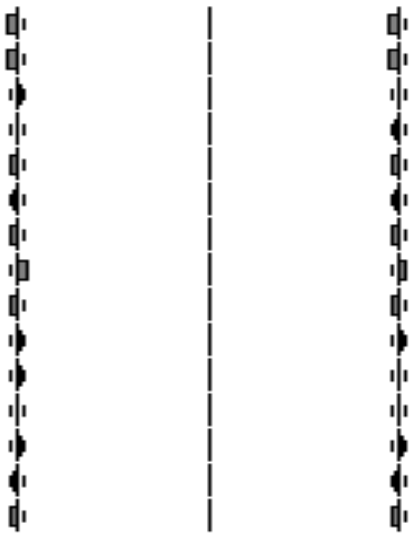
$$\pm 1.96 \times \frac{1}{\sqrt{T}},$$

onde T é o número de observações.

Neste caso tem-se, $T = 5022$, tendo o intervalo de confiança os seguintes valores $[-0.0277;\ 0.0277]$.

Pela Tabela 5, observa-se que os dois primeiros *lags* são significativos, assim como o *lag* 5 e os *lags* 7 e 9. Tendo em conta este resultado, o modelo mais adequado será provavelmente *ARIMA(1,1,1)/ARMA(1,1)* - uma vez que pelo correlograma, apresentado na tabela acima mencionada, não se consegue definir uma tendência. De forma a chegar-se uma conclusão assertiva serão testadas todas as hipóteses – neste caso *AR*(1), *AR*(2), *MA*(1), *MA*(2), *ARMA*(1,1), *ARMA*(1,2), *ARMA*(2,1) e *ARMA*(2,2).

Tabela 5: Correlograma da série dos log-retornos

Autocorrelação	**Correlação Parcial**	***Lags***	**AC**	**PAC**
		1	**-0.059**	**-0.059**
		2	**-0.058**	**-0.061**
		3	0.015	0.007
		4	-0.006	-0.009
		5	**-0.029**	**-0.029**
		6	-0.011	-0.015
		7	**-0.032**	**-0.037**
		8	**0.042**	**0.036**
		9	**-0.028**	**-0.028**
		10	0.017	0.018
		11	0.009	0.006
		12	-0.004	-0.002
		13	0.021	0.023
		14	-0.011	-0.010
		15	-0.042	-0.037

Pela análise dos outputs[13] obtidos na estimação dos modelos, verifica-se que os modelos *ARMA*(1,2), *ARMA*(2,1) e *ARMA*(2,2) não são adequados, uma vez que pelo menos um dos seus coeficientes são não significativos fazendo com que não se possa validar o modelo.

[13] Ver Anexo 1

Prossegue-se assim com os modelos *AR*(1), *AR*(2), *MA*(1), *MA*(2) e *ARMA*(1,1), sendo utilizados os critérios de informação como forma de desempate. Pela Tabela 6, verifica-se que os valores dos critérios de informação dos diferentes modelos são muito próximos, o que dificulta a escolha do melhor modelo.

Tabela 6: Critérios de informação do modelo ARMA(p,q)

		ARMA(p,q)/ARIMA(p,d,q)			
		p/q	0	1	2
Critério de Informação	Akaike information criterion	0	---	**-6.0655**	-6.0686
		1	-6.0672	-6.0692	---
		2	**-6.0670**	---	---
	Schwarz Bayesian information criterion	0	---	**-6.0631**	-6.0647
		1	**-6.0647**	-6.0651	---
		2	-6.0666	---	---
	Hannan-Quinn criterion	0	---	**-6.0648**	-6.0671
		1	**-6.0663**	-6.0678	---
		2	-6.0691	---	---
Durbin-Watson statistic		0	---	1.9895	1.9983
		1	**2.0074**	**1.9921**	---
		2	1.9991	---	---

Uma vez que estes valores são muito próximos uns dos outros, procede-se com a análise dos resíduos dos mesmos e depois faz-se a previsão para que assim se possa fazer uma escolha mais acertada sobre o modelo a escolher.

4.1.5 Análise dos resíduos *ARIMA(p,d,q)*

Na análise de resíduos pretende-se fazer inferência sobre os resíduos de forma a aferir se os mesmos cumprem com os pressupostos do ruido branco, isto é, média zero, independência (falta de autocorrelação) e variância constante. O pressuposto de normalidade também vai ser verificado, apesar de não ser requerido para a validação do modelo.

Antes de prosseguir-se com os testes, faz-se uma pequena análise gráfica dos resíduos. O gráfico dos resíduos[14] dos modelos em estudos são bastante semelhantes, sendo o gráfico do modelo *ARMA*(1,1) na Figura 7 semelhante aos restantes modelos. Verifica-se assim que os resíduos apresentam uma variação bastante pequena e observa-se ainda

14

alguns valores extremos principalmente no ano de 2008, o que evidencia a possível não heterocedasticidade dos resíduos.

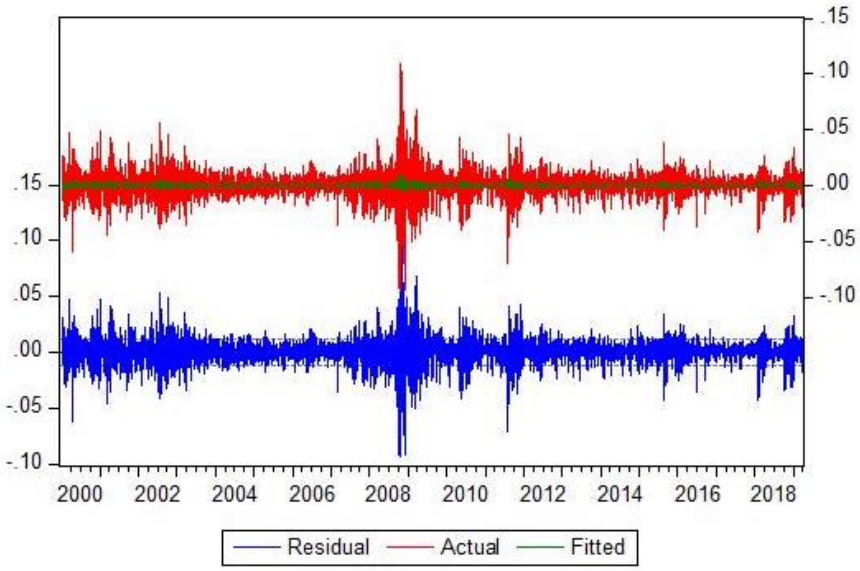


Figura 7: Gráfico dos resíduos, ARMA(1,1)

Pelo diagnostico dos resíduos (Tabela 7), verifica-se que em todos os modelos, os resíduos tem média nula, mas a curtose afasta-se bastante do valor desejado 3 que garantiria a normalidade dos resíduos, sendo os resultados do teste de *J-B* a confirmação da não normalidade das mesmas.

Tabela 7:Estatísticas descritivas dos modelos de Previsão

	AR(1)	*AR*(2)	*MA*(1)	*MA*(2)	*ARMA*(1,1)
Média	$3.74e^{-19}$	$-3.41e^{-19}$	$-5.63e^{-7}$	$-1.01e^{-6}$	$6.52e^{-6}$
Mediana	0.000432	0.000535	0.000466	0.000537	0.000571
Máximo	0.1087717	0.103798	0.108258	0.103401	0.103146
Mínimo	-0.0.095164	-0.093099	-0.094708	-0.092751	-0.093069
Desvio Padrão	0.011645	0.011624	0.011654	0.011635	0.011632
Skewness	-0.271425	-0.335612	-0.289972	-0.350391	-0.379126
Curtose	12.12326	11.64369	12.04734	11.60781	11.72779
Jarque-Bera	17474.87	15721.78	17198.42	15607.02	16056.60
Probabilidade	0.000	0.000	0.000	0.000	0.000

Na Tabela 8 tem-se os resultados do teste de Breusch-Godfreyp ara a correlação serial e do teste ARCH de heterocedasticidade para os resíduos dos diferentes modelos ajustados, onde no teste de *B-G* usou-se para todos os modelos 2 *lags* e para o teste ARCH 1 *lag.*

Pelos resultados apresentados verifica-se que os modelos *AR*(2), MA(2) e *ARMA*(1,1) não têm correlação serial, mas quando se testa para *lags* maiores rejeita-se essa hipótese, o que faz com que se suspeite que ainda existe correlação serial nesses modelos. No que diz respeito aos efeitos ARCH, verifica-se que para todos os modelos existem efeitos ARCH nos resíduos, ou seja, estes apresentam variância não constante (heterocedásticos). Os valores que se encontram na Tabela 8 representam os p-values dos respetivos testes.

Tabela 8: Teste de correlação serial de Bresch-Godfrey e teste ARCH

Modelo		*AR*(1)	*AR*(2)	*MA*(1)	*MA*(2)	*ARMA*(1,1)
Teste	Breusch-Godfrey Serial Correlation	0.0001	0.7513	0.000	0.6960	0.0935
	ARCH	0.0000	0.0000	0.0000	0.0000	0.0000

Uma vez que os modelos *AR*(2), MA(2) e *ARMA*(1,1) não apresentam correlação serial, iremos prosseguir com estes para a previsão dos preços de fecho.

4.1.6 Síntese

Como forma de identificação dos possíveis modelos que melhor se adequam aos dados, recorreu-se primeiramente a representação gráfica dos coeficientes de autocorrelação (*AC*) e de autocorrelação parcial (*PAC*), onde pouco se concluiu, uma vez que os gráficos não apresentavam padrões definidos. Tendo em conta as suspeitas levantadas pela análise gráfica, fez-se a estimação dos modelos, chegando a conclusão que apenas os modelos *AR*(1), *AR(*2), *MA*(1), *MA*(2) e A*RMA*(1,1) são adequados. Como forma de desempate, recorre-se aos critérios de informação, que por sua vez pouco se consegue concluir, pois os valores dos diferentes modelos são bastante próximos.

Tendo em conta a dificuldade em escolher-se um modelo, procede-se com a análise dos resíduos para os diferentes modelos em estudo. Sendo que para todos eles, rejeita-se a hipótese de normalidade pelo teste de *J-B*. Em termos de heterocedasticidade dos resíduos, todos os modelos o apresentam. No que toca a correlação serial apenas os modelos *AR*(2), *MA*(2) e *ARMA*(1,1) não o apresentam.

4.1.7 Previsão

Neste subcapítulo o objetivo é prever o preço de fecho do índice *S&P 500*. Sendo a janela temporal decomposta da seguinte forma: numa primeira fase a previsão será feita *in-sample,* ou seja, dentro da janela temporal correspondente. Depois numa segunda fase será feita a previsão *out-of-simple,* que consiste na previsão de valores fora da janela temporal, neste caso regista-se o último valor do preço de fecho do dia 03/04/2019 e logo de seguida retira-se esse valor.

Na previsão o objetivo é escolher os modelos que melhor se adequam para prever a variável em questão, assim sendo a comparação entre dois ou mais modelos feita através do erro de previsão associado ao respetivo modelo.

4.1.7.1 Previsão *in-sample*

A previsão *in-sample* será feita sobre toda a amostra, totalizando 5023 observações contempladas entre 03/01/200 e 03/04/2019.

Analisando a Tabela 9, verifica-se que em termos do erro todos os modelos apresentam valores bastante similares de *RMSE* e *MAPE*. No que toca ao *Bias Proportion* e a *Covariance Proportion*, verifica-se que apesar dos valores serem bastante próximos, os modelos *AR*(2) e *MA*(2) apresentam melhores resultados. Pelo que não se consegue chegar a uma conclusão de qual modelo será o melhor em prever os valores de fecho dentro da amostra.

Tabela 9: RMSE, MAE e MAPE, previsão in-sample

	AR(2)	*MA(2)*	*ARMA(1,1)*
RMSE	15.53906	15.55637	15.54974
MAE	10.69282	10.70159	10.69818
MAPE	0.771258	0.771929	0.771857
Theil Inequality Coefficient	0.004830	0.004836	0.004834
Bias Proportion	0.000037	0.000037	0.000051
Variance Proportion	0.000449	0.000489	0.000526
Covariance Proportion	1.001509	1.000211	0.997905

4.1.7.2 Previsão *out-of-simple*

Como foi mencionado na introdução deste subcapítulo a previsão *out-of-sample* prevê valores fora da janela temporal da amostra. Sendo o objetivo numa primeira fase prever o último valor da amostra (03/04/2019), procedendo primeiramente com o registo desse valor, e posteriormente com a retirada do mesmo, fazendo logo de seguida a previsão. Uma vez obtida a previsão procede-se com a comparação do valor previsto com o valor real.

Tem-se assim 2873.400 como valor real do preço de fecho do dia 03/04/2019.

Ao compararmos o valor real com o valor predito do dia 03/04/2019, verifica-se que os valores do erro de previsão nos diferentes modelos são bastante próximos, sendo que o modelo ARMA(1,1) o que apresenta um menor valor.

Tabela 10:Previsão out-of-sample, valor registrado e erro de previsão

	AR(2)	*MA(2)*	*ARMA(1,1)*
Valor registado	2865.671	2865.450	2865.712
Erro de previsão	7.729	7.950	7.688

Como conclusão final, retiramos o facto que o modelo ARMA(1,1) teve o melhor desempenho na previsão fora-da-amostra da série temporal financeira em estudo, com um erro de previsão de 0.2%.

4.2 Python

Nos últimos anos uma grande variedade de métodos recorrendo ao *machine learning* têm sido desenvolvidos para a previsão dos preços dos índices. Neste subcapítulo serão abordados alguns dos mais populares algoritmos de regressão linear, sendo estes a ***regressão linear***, ***random forest*** e ***support vector machine***.

Para a implementação desses algoritmos recorre-se a linguagem de programação *Python*, que foi lançado em 1991 e atualmente é considerada uma das linguagens mais importantes, não só por ser gratuita, mas também pela sua fácil manipulação.

O objetivo mantém-se o mesmo, prever os preços de fecho da série, sendo que os dados também são os mesmos da seção anterior.

Para previsão dos preços de fecho não bastam apenas as variáveis que se têm disponíveis na amostra, uma vez que estas podem levar a subvalorização dos resultados, ou seja, fornece resultados que não espelham a realidade. Por isso recorre-se as *feature engineerings*[15], de modo a obter-se melhores resultados.

Segundo (Yu, et al., 2016) *feature engineering* é o processo de criar variáveis concretas a partir de variáveis já existentes. Essa construção requer um conhecimento prévio sobre o tema e a sua implementação revela-se muitas vezes difícil e de grande consumo de tempo.

Tendo em conta os fatores que os investidores consideram importantes para a análise e tomada de decisões, criou-se trinta e uma variáveis baseadas nas variáveis já existentes, sendo estas as seguintes:

- Preço médio de fecho dos últimos 5 dias, $AvgPrice_5$
- Preço médio de fecho do mês anterior, $AvgPrice_{30}$
- Preço médio de fecho do ano anterior, $AvgPrice_{365}$
- Rácio entre o preço médio dos últimos 5 dias e do mês anterior, $\frac{AvgPrice_5}{AvgPrice_{30}}$
- Rácio entre o preço médio dos últimos 5 dias e do ano anterior, $\frac{AvgPrice_5}{AvgPrice_{365}}$
- Rácio entre o preço médio do último mês e do ano anterior, $\frac{AvgPrice_{30}}{AvgPrice_{365}}$
- Volume médio dos últimos 5 dias, $AvgVolume_5$
- Volume médio do mês passado, $AvgVolume_{30}$
- Volume médio do ano passado, $AvgVolume_{365}$
- Rácio entre o volume médio dos últimos 5 dias e do mês anterior, $\frac{AvgVolume_5}{AvgVolume_{30}}$
- Rácio entre o volume médio dos últimos 5 dias e do ano anterior, $\frac{AvgVolume_5}{AvgVolume_{365}}$
- Rácio entre o volume médio do mês passado e do ano anterior, $\frac{AvgVolume_{30}}{AvgVolume_{365}}$
- Desvio padrão do preço de fecho dos últimos 5 dias, $StdPrice_5$
- Desvio padrão do preço de fecho do mês anterior, $StdPrice_{30}$
- Desvio padrão do preço de fecho do ano anterior, $StdPrice_{365}$

[15] É o processo de criação de variáveis especificas baseadas nas variáveis já existentes.

- Rácio entre o desvio padrão do preço de fecho dos últimos 5 dias e do mês anterior, $\frac{StdPrice_5}{StdPrice_{30}}$
- Rácio entre o desvio padrão do preço de fecho dos últimos 5 dias e do ano anterior, $\frac{StdPrice_5}{StdPrice_{365}}$
- Rácio entre o desvio padrão do preço de fecho do último mês e do ano anterior, $\frac{StdPricee_{30}}{StdPrice_{365}}$
- Desvio padrão do volume dos últimos 5 dias, $StdVolume_5$
- Desvio padrão do volume do mês anterior, $StdVolume_{30}$
- Desvio padrão do volume do ano anterior, $StdVolume_{365}$
- Rácio entre o desvio padrão do volume dos últimos 5 dias e do mês anterior, $\frac{StdVolume_5}{StdVolume_{30}}$
- Rácio entre o desvio padrão do volume dos últimos 5 dias e do ano anterior, $\frac{StdVolume_5}{StdVolume_{365}}$
- Rácio entre o desvio padrão do volume do último mês e do ano anterior, $\frac{StdVolume_{30}}{StdVolume_{365}}$
- Retorno diário do dia anterior, $return_{i:i-1}$
- Retorno semanal da semana anterior, $return_{i:i-5}$
- Retorno mensal do mês anterior, $return_{i:i-30}$
- Retorno anual do ano anterior, $return_{i:i-365}$
- Média móvel dos retornos diários dos últimos 5 dias, $MovingAvg_{i\_5}$
- Média móvel dos retornos diários do mês anterior, $MovingAvg_{i\_30}$
- Média móvel dos retornos diários do ano anterior, $MovingAvg_{i\_365}$

Essas novas variáveis foram criadas, baseadas em seis variáveis originais:

- Preço de abertura, $OpenPrice_i$
- Preço de abertura do dia anterior, $OpenPrice_{i-1}$
- Preço de fecho do dia anterior, $ClosePrice_{i-1}$
- Preço mais alto do dia anterior, $HighPrice_{i-1}$
- Preço mais baixo do dia anterior, $LowPrice_{i-1}$
- Volume do dia anterior, $VolumePrice_{i-1}$

Estas variáveis revelam-se importantes porque os investidores normalmente analisam o histórico dos preços, o histórico do volume de transações, a volatilidade do mercado, o retorno dos investimentos e não apenas o preço do dia anterior (Liu, 2017). Sendo assim importante fornecer ao algoritmo essas informações. Estas variáveis revelam-se também importantes para a generalização dos dados de treino em diferentes circunstâncias.

4.2.1 Previsão

Sabe-se que uma das principais utilidades do *Machine Learning* é a construção de algoritmos que podem aprender a fazer previsões sobre os dados, sendo isto feito através de modelos matemáticos a partir dos dados de entrada, também designados por *input data*. O modelo final é resultado de diferentes conjuntos de dados.

O modelo é inicialmente testado no ***training data***, tendo este uma quantidade definida dos dados da amostra em que é feita a estimação dos parâmetros do modelo. No final para a validação do modelo, tem-se o ***test data***, que é formado por um conjunto de dados que fornecem uma avaliação rigorosa do ajuste final do modelo no conjunto de dados de treinamento (*training data*).

Tem-se inicialmente uma amostra constituída pela janela temporal de 03/01/2000 até 03/04/2019, sendo que quando aplicado o método das *feature engineerings*, a janela temporal encurta-se, passando assim a ser o período em analise de 04/01/2001 até 03/04/2019. Isto deve-se ao facto de as *feature engineerings* considerarem os períodos anteriores, uma vez que não se tem informação sobre o período anterior a 2000 considera-se o período de 2001 ao qual recorre-se ao ano anterior como base e assim por diante.

Para os dados de *training* definiu-se uma percentagem de aproximadamente oitenta porcento (04/01/2001 até 07/08/2015), correspondendo a três mil seiscentos e setenta observações (3670). Os restantes vinte porcentos são pertencentes ao *test data* (08/08/2015 até 03/04/2019), correspondente a novecentos e dezanove observações (919).

Em termos de variáveis de controlo para os diferentes algoritmos[16], tem-se:

- LR: $\boldsymbol{\alpha}$ controla o quanto se quer regularizar o modelo; e o $\boldsymbol{\eta}$ é a medida de aprendizagem que controla o tamanho dos *updates*.
- Random forest: ***Max_depht*** *(Maximum depth of the tree)* controla a profundidade máxima que um nó pode atingir. É de notar que um nó só para de crescer quando a sua profundidade não é menor que a profundidade máxima da árvore; ***Max_features*** nº máximo de *features*; ***Min_samples_leaf*** nº mínimo de folhas na amostra; ***Min_samples_split*** nº mínimo de partições na amostra.
- SVR: $\boldsymbol{C}$ é um híper-parâmetro que regula o *tradeoff* entre o aumento da linha de decisão/*decision boundary* e a garantia que cada x_i esteja no lado correto da linha de decisão; $\boldsymbol{\varepsilon}$ é a distância entre as linhas de decisão; ***k*** *é a função kernel.*

Pela tabela Tabela 11, Tabela 12 e Tabela 13 (ver pág. seguinte) verifica-se que em todos os algoritmos o MSE e RMSE são muito elevados, principalmente no algoritmo *Random Forest.* Como tal, recorre-se ao MAPE como forma de comparar percentualmente o erro, onde se observa que os algoritmos *Linear Regression* e *Support Vector Regressor* apresentam erros percentuais muito baixos, enquanto que o algoritmo

[16] As suas implementações em Python podem ser encontradas em Anexo 2

RF apresenta um valor bastante elevado. Em relação ao coeficiente de determinação verifica-se no algoritmo *RF* um valor negativo[17], o que significa que este não se ajusta corretamente aos dados, ou seja, o modelo é "pobre", o que seria de se esperar uma vez que o modelo *RF* é mais usado para a classificação de dados discretos. Ao contrário do *RF*, os algoritmos *LR* e *SVR* apresentam um coeficiente de determinação bastante próximo de 1, o que significa que estes explicam quase na sua totalidade a variabilidade dos dados, ou seja, ajustam-se corretamente aos dados.

Tabela 11: Estatísticas da previsão in-sample, Regressão Linear

MSE	371.042
RMSE	19.262
MAE	12.849
MAPE	0.547
R^2	0.996
α	0.0001
η	0.03
Tempo de execução[18]	61.429

Tabela 12: Estatísticas da previsão in-sample, Random Forest

MSE	166180.904
RMSE	407.653
MAE	305.867
MAPE	11.588
R^2	-0.811
Max_depht	80
Max_features	auto
Min_samples_leaf	5
Min_samples_split	5
Tempo de execução	291.85

Tabela 13: Estatísticas da previsão in-sample, SVR

MSE	353.355
RMSE	18.798
MAE	12.463
MAPE	0.526
R^2	0.996
C	500
ε	0.0001
K	Linear
Tempo de execução	305.706

[17] O coeficiente de determinação compara o ajuste do modelo com uma linha horizontal. Se o modelo escolhido não se ajustar bem em comparação com a linha horizontal, o valor do coeficiente será negativo. O coeficiente de Determinação nem sempre é positivo, sendo que um valor negativo não significa nenhuma violação das regras matemáticas.

[18] Em segundos

Pela Figura 8 (página seguinte) pode-se observar o ajuste dos modelos aos dados no conjunto de teste, confirmando mais uma vez a inadequabilidade do modelo *Random Forest* neste caso. O desempenho do modelo de regressão linear e do SVM é bastante semelhante, sendo as curvas ajustadas praticamente sobrepostas sobre a curva da série temporal dos preços de fecho de S&P 500.

Como futuros projetos de trabalho, podemos ainda reduzir o conjunto de teste, para vermos se o RF acompanha a qualidade na previsão da série, visto que, após cerca 300 passos preditos, perdeu a tendência e a dinâmica da série. Também podem ser analisados os mesmos algoritmos apenas para as variáveis originais (sem utilização de *feature engineering*) e posteriormente comparar com os resultados aqui obtidos.

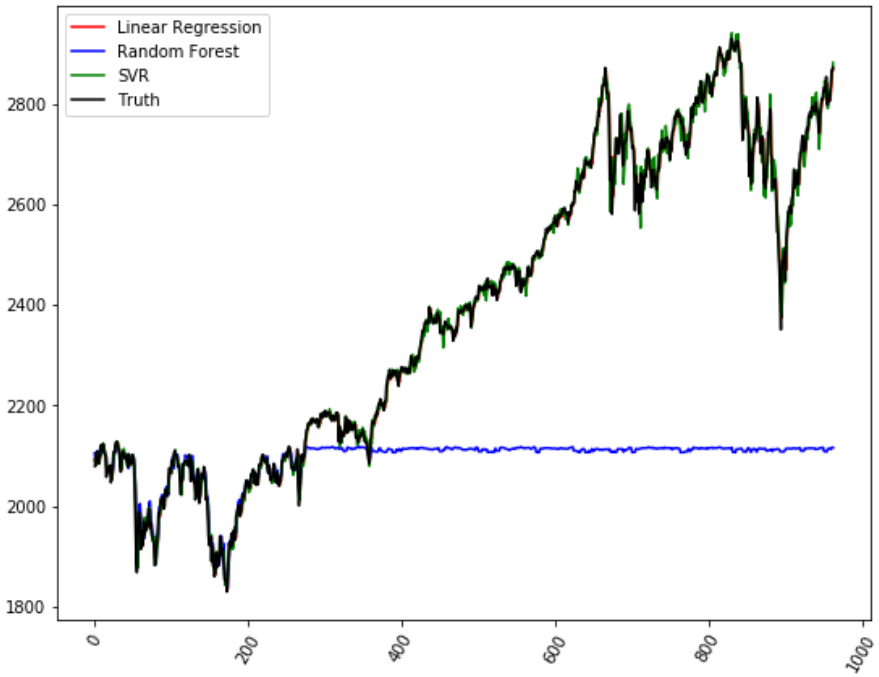


Figura 8: Gráfico de linhas da previsão in-sample dos modelos LR, RF, SVR Vs. Valor Real

Sendo assim, o modelo SVM apresenta o menor erro percentual, isto é, 0.526% para a previsão dos 919 elementos do conjunto de teste, com base nas 31 variáveis construídas a partir da série original. Comparando com o erro de 0.2% para a previsão de um único valor pelo modelo ARMA(1,1), prova-se claramente a utilidade da *feature engineering* e o uso de *big data* e algoritmos de *Machine learning* na previsão de séries financeiras. Rematamos ainda que os erros de previsão obtidos estão em linha e até competem com os resultados de outros autores.

5. Conclusão

Em virtude do propósito desta tese, que é comprar os resultados que se obtiveram através dos diferentes modelos clássicos da família ARMA e de *Machine Learning* aplicados, usou-se os dados de um dos mais importantes índices americanos, o *S&P 500*.

Ainda tendo em conta os propósitos da mesma, os dados foram analisados em dois *softwares* diferentes, *Eviews* e *Python*. No primeiro software, foram explorados os métodos clássicos econométricos, enquanto que no segundo os métodos de *Machine Leaning*.

No programa *Eviews*, após uma rigorosa análise descritiva e inferência sobre os dados, verificou-se que a série dos log-retornos é estacionária, sendo a série em níveis integrada de ordem 1. Prosseguiu-se com a identificação dos modelos que melhor se adequam aos dados, assim como a sua validação chegando a conclusão que os modelos AR(2), MA(2) e ARMA(1,1) são os que melhor adequam e cumprem com os requisitos. Uma vez obtidos os modelos, prosseguiu-se com a previsão dos mesmos onde se concluiu que o modelo ARMA(1,1) é o que apresenta maior precisão entre os demais modelos, sendo assim este o melhor modelo para a previsão dos preços de fecho.

No programa *Python*, recorreu-se ao auxílio das *features engineering* como forma de obter melhores resultados nas previsões. Tais metodologias têm como pressuposto basearem-se em resultados anteriores, o que implicou a redução da janela temporal. Passando esta a começar em 2001 em vez do ano 2000, uma vez que este serve como base para o começo dos cálculos.

Uma vez obtida as *FE*, aplicaram-se os algoritmos. Tendo estes revelado uma enorme eficácia e robustez, uma vez que estes não necessitam propriamente de verificar os pressupostos, mas precisam antes ser controlados através de parâmetros de controlo. No que diz respeito a precisão, o modelo *Random Forest* foi o que apresentou piores resultados, revelando-se assim pouco estável para previsão dos preços de fecho. Por outro lado, os modelos *Linear Regression* e *Support Vector Regressor* apresentaram excelentes resultados. Sendo SVR o melhor modelo, uma vez que este apresenta um *MAPE* inferior ao *LR*.

Quando se comparam os resultados obtidos pelos diferentes métodos aplicados nos dois *sofwtares* considerados, chega-se a conclusão, através da medida do erro MAPE, de que os algoritmos de ML (*SVR* e *LR*) aplicados no *Python* apresentam melhores resultados do que os modelos clássicos econométricos aplicados no *Eviews*. Contudo, há que reparar que os algoritmos de *ML* têm a vantagem de os dados serem baseados em valores anteriores, pelo que não há propriamente perda de informação. Uma outra questão é que nos algoritmos de *ML* não se tem muito controlo sobre os mesmos, enquanto que no *Eviews* é feita uma exaustiva análise e verificação dos pressupostos do modelo.

No presente estudo os modelos clássicos mostraram serem capazes de lidar com um grande número de dados, porém possuem de pouca versatilidade em relação aos métodos de *ML*. É necessário que as técnicas usadas para os modelos clássicos sejam reformuladas e adaptadas as exigências futuras do mercado.

Bibliografia

BBVA The five V's of big data [Online] // BBVA. - 17 de Novembro de 2019. - https://www.bbva.com/en/five-vs-big-data/.

Brooks Chris Introductory Econometrics for Finance [Livro]. - Nova Iorque : Cambridge University Press, 2008.

Burkov Andriy The Hundred - Page Machine Learning Book [Livro]. - 2019.

Cambridge University Press Introductory Econometrics for Finance [Livro]. - New York : Cambridge University Press, 2008.

Ceri Stefano On the role of statistics in the era of big data: A computer [Diário] // Elsevier. - 2018. - p. 5.

Cerulli Giovanni A Super-Learning Machine for Predicting Economic Outcomes [Diário] // Munich Personal RePEc Archive. - 2020. - p. 9.

Dingli Alexiei e Founier Karl Sant Financial Time Series Forecasting - A Machine Learning Approach [Diário] // Machine Learning and Applications: An International Journal. - 2017.

Dingli Alexiei e Founier Karl Sant FINANCIAL TIME SERIES FORECASTING - A MACHINE LEARNING APPROACH [Diário] // Machine Leaning and Applications: An International Journal (MLAIJ). - 2017.

Economist Intelligence Unit The evolving role of data in decision making [Relatório]. - [s.l.] : The Economist Intelligence Unit Limited, 2013.

EDUCBA Introduction to Financial Markets [Online] // EDUCBA. - 11 de Dezembro de 2019. - https://www.educba.com/financial-markets/.

Estrategia-Bolsa Como funciona o mercado de ações? [Online] // Estrategia-Bolsa. - Janeiro de 2020. - https://www.estrategia-bolsa.pt/como-funciona-mercado-acoes.html.

Faraway Julian J. e Augustin Nicole H. When small data beats big data [Diário] // ELSEVIER. - 2018. - p. 4.

Géron Aurélien Hands-On Machine Learning with Scikit-Learn and TensorFlow [Livro]. - United States of America : O'Reilly Media, Inc., 2017.

Hassani Hossein e Silva Emmanuel Sirimal Forecasting with Big Data: A Review [Diário] // Annals of Data Science. - 2015.

Judith Hurwitz Alan Nugent, Fern Halper, Marcia Kaufman Data Mining for Big Data [Online] // Dummies. - https://www.dummies.com/programming/big-data/engineering/data-mining-for-big-data/.

Liu Yuxi Python Machine Leaning By Example [Livro]. - Birmingham : Packt Publishing Ltd., 2017.

Lu Chi_Jie, Lee Tian-Shyug e Chiu Chih-Chou Financial time series forecasting using independent component analysis and support vector regression [Diário] // ELSEVIER. - 2009. - p. 11.

MIT OpenCourseWare MIT OpenCourseWare [Online] // MIT OpenCourseWare. - Maio de 2020. - https://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/.

Olhede Sofia C. e Wolfe Patrick J. The future of statistics and data science [Diário] // Statistics and Probability Letters. - 2018. - p. 5.

Qian Xinyao Financial Series Prediction: Comparison Between Precision of Time [Artigo] // Financial Series Prediction: Comparison Between Precision of Time. - 25 de Dezembro de 2019. - p. 9.

Silva Miguel Gomes da Bolsa - Investir nos mercados financeiros [Livro]. - [s.l.] : Bookout, 2015.

Smola Alex J. e Scholkopf Bernhard A tutorial on support vector regression [Diário] // Statistics and Computing. - 2003. - p. 24.

TechJury Big Data Statistics 2019 [Online] // TechJury. - 17 de Novembro de 2019. - https://techjury.net/stats-about/big-data-statistics/#gref.

Toro CTVM Ltda Mercado de capitais: o que é e como funciona [Online] // Toro investimentos. - Janeiro de 2020. - https://blog.toroinvestimentos.com.br/mercado-de-capitais-o-que-e.

Torrecilla José L. e Romo Juan Data learning from big data [Diário] // Elsevier. - 2018. - p. 5.

Wells Fargo Securities Economics Group Big Data Applications in the Economics/Financial World Part I: Opportunities and Challengs [Artigo] // Big Data Applications in the Economics/Financial World Part I: Opportunities and Challengs. - 06 de Abril de 2017. - p. 8.

Wikipedia Mercado de capitais [Online] // Wikipedia. - janeiro de 2020. - https://pt.wikipedia.org/wiki/Mercado_de_capitais.

Yu Shui e Guo Song Big Data Concepts, Theories, and Aplications [Livro]. - Nova Iorque : Springer, 2016.

Anexo 1

Null Hypothesis: CLOSE_INTER has a unit root
Exogenous: Constant
Lag Length: 2 (Automatic - based on SIC, maxlag=31)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		0.756824	0.9933
Test critical values:	1% level	-3.431471	
	5% level	-2.861920	
	10% level	-2.567015	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(CLOSE_INTER)
Method: Least Squares
Date: 06/10/20 Time: 12:30
Sample (adjusted): 1/06/2000 4/03/2019
Included observations: 5020 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
CLOSE_INTER(-1)	0.000316	0.000418	0.756824	0.4492
D(CLOSE_INTER(-1))	-0.037469	0.014108	-2.655967	0.0079
D(CLOSE_INTER(-2))	-0.055634	0.014089	-3.948804	0.0001
C	-0.160805	0.671425	-0.239497	0.8107

R-squared	0.004402	Mean dependent var	0.293086
Adjusted R-squared	0.003807	S.D. dependent var	15.56686
S.E. of regression	15.53720	Akaike info criterion	8.325148
Sum squared resid	1210885.	Schwarz criterion	8.330344
Log likelihood	-20892.12	Hannan-Quinn criter.	8.326968
F-statistic	7.392997	Durbin-Watson stat	1.998533
Prob(F-statistic)	0.000061		

Figura 9: Teste de ADF, variável "close"

Null Hypothesis: CLOSE_INTER has a unit root
Exogenous: Constant
Bandwidth: 30 (Newey-West automatic) using Bartlett kernel

		Adj. t-Stat	Prob.*
Phillips-Perron test statistic		0.929341	0.9959
Test critical values:	1% level	-3.431470	
	5% level	-2.861920	
	10% level	-2.567015	

*MacKinnon (1996) one-sided p-values.

Residual variance (no correction)	242.7966
HAC corrected variance (Bartlett kernel)	179.1336

Phillips-Perron Test Equation
Dependent Variable: D(CLOSE_INTER)
Method: Least Squares
Date: 06/10/20 Time: 12:31
Sample (adjusted): 1/04/2000 4/03/2019
Included observations: 5022 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
CLOSE_INTER(-1)	0.000219	0.000418	0.522877	0.6011
C	-0.050130	0.672902	-0.074498	0.9406

R-squared	0.000054	Mean dependent var	0.282393
Adjusted R-squared	-0.000145	S.D. dependent var	15.58391
S.E. of regression	15.58503	Akaike info criterion	8.330897
Sum squared resid	1219324.	Schwarz criterion	8.333495
Log likelihood	-20916.88	Hannan-Quinn criter.	8.331807
F-statistic	0.273401	Durbin-Watson stat	2.068230
Prob(F-statistic)	0.601083		

Figura 10: Teste de PP, variável "close"

Null Hypothesis: CLOSE_INTER is stationary
Exogenous: Constant
Bandwidth: 55 (Newey-West automatic) using Bartlett kernel

		LM-Stat.
Kwiatkowski-Phillips-Schmidt-Shin test statistic		6.237448
Asymptotic critical values*:	1% level	0.739000
	5% level	0.463000
	10% level	0.347000

*Kwiatkowski-Phillips-Schmidt-Shin (1992, Table 1)

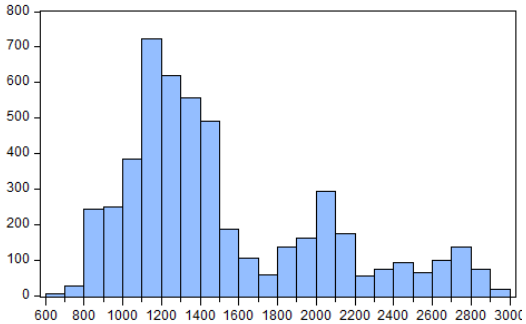
Residual variance (no correction)	276676.6
HAC corrected variance (Bartlett kernel)	15223287

KPSS Test Equation
Dependent Variable: CLOSE_INTER
Method: Least Squares
Date: 06/10/20 Time: 12:32
Sample: 1/03/2000 4/03/2019
Included observations: 5023

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1520.453	7.422460	204.8449	0.0000

R-squared	0.000000	Mean dependent var	1520.453
Adjusted R-squared	0.000000	S.D. dependent var	526.0529
S.E. of regression	526.0529	Akaike info criterion	15.36888
Sum squared resid	1.39E+09	Schwarz criterion	15.37018
Log likelihood	-38597.94	Hannan-Quinn criter.	15.36933
Durbin-Watson stat	0.000878		

Figura 11: Teste de KPSS, variável "close"



Series: CLOSE_INTER	
Sample 1/03/2000 4/03/2019	
Observations 5023	
Mean	1520.453
Median	1342.540
Maximum	2930.750
Minimum	676.5300
Std. Dev.	526.0529
Skewness	0.983093
Kurtosis	2.994288
Jarque-Bera	809.1041
Probability	0.000000

Figura 12: Histograma, estatísticas descritivas e teste de normalidade de Jarque-Bera, variável "close"

Null Hypothesis: DCLOSE has a unit root
Exogenous: Constant
Lag Length: 1 (Automatic - based on SIC, maxlag=31)

		t-Statistic	Prob.*
Augmented Dickey-Fuller test statistic		-54.85427	0.0001
Test critical values:	1% level	-3.431471	
	5% level	-2.861920	
	10% level	-2.567015	

*MacKinnon (1996) one-sided p-values.

Augmented Dickey-Fuller Test Equation
Dependent Variable: D(DCLOSE)
Method: Least Squares
Date: 06/10/20 Time: 12:35
Sample (adjusted): 1/06/2000 4/03/2019
Included observations: 5020 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DCLOSE(-1)	-1.124608	0.020502	-54.85427	0.0000
D(DCLOSE(-1))	0.061418	0.014076	4.363400	0.0000
C	0.000160	0.000164	0.976460	0.3289

R-squared	0.531541	Mean dependent var	4.50E-08
Adjusted R-squared	0.531355	S.D. dependent var	0.016984
S.E. of regression	0.011627	Akaike info criterion	-6.070450
Sum squared resid	0.678184	Schwarz criterion	-6.066553
Log likelihood	15239.83	Hannan-Quinn criter.	-6.069084
F-statistic	2846.294	Durbin-Watson stat	1.999079
Prob(F-statistic)	0.000000		

Figura 13: Teste de ADF, variável "dclose"

Null Hypothesis: DCLOSE has a unit root
Exogenous: Constant
Bandwidth: 23 (Newey-West automatic) using Bartlett kernel

		Adj. t-Stat	Prob.*
Phillips-Perron test statistic		-76.10933	0.0001
Test critical values:	1% level	-3.431471	
	5% level	-2.861920	
	10% level	-2.567015	

*MacKinnon (1996) one-sided p-values.

Residual variance (no correction)	0.000136
HAC corrected variance (Bartlett kernel)	0.000111

Phillips-Perron Test Equation
Dependent Variable: D(DCLOSE)
Method: Least Squares
Date: 06/10/20 Time: 12:36
Sample (adjusted): 1/05/2000 4/03/2019
Included observations: 5021 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
DCLOSE(-1)	-1.059496	0.014074	-75.27783	0.0000
C	0.000151	0.000164	0.920620	0.3573

R-squared	0.530309	Mean dependent var	8.21E-06
Adjusted R-squared	0.530216	S.D. dependent var	0.016992
S.E. of regression	0.011646	Akaike info criterion	-6.067260
Sum squared resid	0.680758	Schwarz criterion	-6.064662
Log likelihood	15233.85	Hannan-Quinn criter.	-6.066349
F-statistic	5666.751	Durbin-Watson stat	2.007382
Prob(F-statistic)	0.000000		

Figura 14: Teste de PP, variável "dclose"

Null Hypothesis: DCLOSE is stationary
Exogenous: Constant
Bandwidth: 26 (Newey-West automatic) using Bartlett kernel

		LM-Stat.
Kwiatkowski-Phillips-Schmidt-Shin test statistic		0.336391
Asymptotic critical values*:	1% level	0.739000
	5% level	0.463000
	10% level	0.347000

*Kwiatkowski-Phillips-Schmidt-Shin (1992, Table 1)

Residual variance (no correction)	0.000136
HAC corrected variance (Bartlett kernel)	9.86E-05

KPSS Test Equation
Dependent Variable: DCLOSE
Method: Least Squares
Date: 06/10/20 Time: 12:36
Sample (adjusted): 1/04/2000 4/03/2019
Included observations: 5022 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000135	0.000165	0.822101	0.4111

R-squared	0.000000	Mean dependent var	0.000135
Adjusted R-squared	0.000000	S.D. dependent var	0.011678
S.E. of regression	0.011678	Akaike info criterion	-6.062052
Sum squared resid	0.684721	Schwarz criterion	-6.060753
Log likelihood	15222.81	Hannan-Quinn criter.	-6.061597
Durbin-Watson stat	2.116737		

Figura 15: Teste de KPSS, variável "dclose"

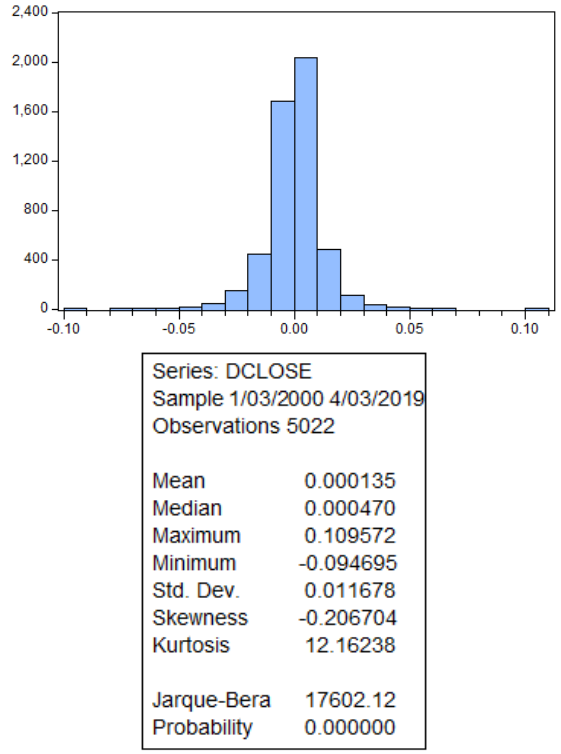


Figura 16: Histograma, estatísticas descritivas e teste de normalidade de Jarque-Bera, variável "dclose"

Dependent Variable: DLOG(CLOSE_INTER)
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 06/10/20 Time: 12:39
Sample (adjusted): 1/05/2000 4/03/2019
Included observations: 5021 after adjustments
Convergence achieved after 2 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000143	0.000155	0.920681	0.3573
AR(1)	-0.059496	0.014074	-4.227205	0.0000

R-squared	0.003548	Mean dependent var	0.000143
Adjusted R-squared	0.003349	S.D. dependent var	0.011666
S.E. of regression	0.011646	Akaike info criterion	-6.067260
Sum squared resid	0.680758	Schwarz criterion	-6.064662
Log likelihood	15233.85	Hannan-Quinn criter.	-6.066349
F-statistic	17.86927	Durbin-Watson stat	2.007382
Prob(F-statistic)	0.000024		

Inverted AR Roots	-.06

Figura 17: Estimação do modelo AR(1)

Dependent Variable: DLOG(CLOSE_INTER)
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 05/21/20 Time: 04:53
Sample (adjusted): 1/06/2000 4/03/2019
Included observations: 5020 after adjustments
Convergence achieved after 2 iterations
Coefficient covariance computed using outer product of gradients

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000143	0.000146	0.976607	0.3288
AR(1)	-0.063190	0.014091	-4.484269	0.0000
AR(2)	-0.061418	0.014076	-4.363400	0.0000

R-squared	0.007311	Mean dependent var	0.000143
Adjusted R-squared	0.006915	S.D. dependent var	0.011667
S.E. of regression	0.011627	Akaike info criterion	-6.070450
Sum squared resid	0.678184	Schwarz criterion	-6.066553
Log likelihood	15239.83	Hannan-Quinn criter.	-6.069084
F-statistic	18.47407	Durbin-Watson stat	1.999079
Prob(F-statistic)	0.000000		

Inverted AR Roots	-.03+.25i	-.03-.25i

Figura 18: Estimação do modelo AR(2)

Dependent Variable: DLOG(CLOSE_INTER)
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 06/10/20 Time: 12:42
Sample (adjusted): 1/04/2000 4/03/2019
Included observations: 5022 after adjustments
Failure to improve likelihood (non-zero gradients) after 5 iterations
Coefficient covariance computed using outer product of gradients
MA Backcast: 1/03/2000

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000136	0.000153	0.886935	0.3752
MA(1)	-0.067682	0.014081	-4.806494	0.0000

R-squared	0.004031	Mean dependent var	0.000135
Adjusted R-squared	0.003833	S.D. dependent var	0.011678
S.E. of regression	0.011655	Akaike info criterion	-6.065693
Sum squared resid	0.681961	Schwarz criterion	-6.063096
Log likelihood	15232.95	Hannan-Quinn criter.	-6.064783
F-statistic	20.31806	Durbin-Watson stat	1.989573
Prob(F-statistic)	0.000007		

Inverted MA Roots	.07

Figura 19: Estimação do modelo MA(1)

Dependent Variable: DLOG(CLOSE_INTER)
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 06/10/20 Time: 12:42
Sample (adjusted): 1/04/2000 4/03/2019
Included observations: 5022 after adjustments
Failure to improve likelihood (non-zero gradients) after 4 iterations
Coefficient covariance computed using outer product of gradients
MA Backcast: 12/31/1999 1/03/2000

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000136	0.000144	0.944286	0.3451
MA(1)	-0.062553	0.014093	-4.438540	0.0000
MA(2)	-0.058137	0.014091	-4.125683	0.0000

R-squared	0.007273	Mean dependent var	0.000135
Adjusted R-squared	0.006877	S.D. dependent var	0.011678
S.E. of regression	0.011638	Akaike info criterion	-6.068555
Sum squared resid	0.679741	Schwarz criterion	-6.064659
Log likelihood	15241.14	Hannan-Quinn criter.	-6.067190
F-statistic	18.38550	Durbin-Watson stat	1.998362
Prob(F-statistic)	0.000000		

Inverted MA Roots	.27	-.21

Figura 20: Estimação do modelo MA(2)

Dependent Variable: DLOG(CLOSE_INTER)
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 06/10/20 Time: 12:41
Sample (adjusted): 1/05/2000 4/03/2019
Included observations: 5021 after adjustments
Failure to improve likelihood (non-zero gradients) after 17 iterations
Coefficient covariance computed using outer product of gradients
MA Backcast: 1/04/2000

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000142	0.000139	1.024450	0.3057
AR(1)	0.569717	0.107518	5.298806	0.0000
MA(1)	-0.636951	0.100742	-6.322616	0.0000

R-squared	0.005862	Mean dependent var	0.000143
Adjusted R-squared	0.005466	S.D. dependent var	0.011666
S.E. of regression	0.011634	Akaike info criterion	-6.069186
Sum squared resid	0.679177	Schwarz criterion	-6.065290
Log likelihood	15239.69	Hannan-Quinn criter.	-6.067821
F-statistic	14.79399	Durbin-Watson stat	1.992102
Prob(F-statistic)	0.000000		

Inverted AR Roots	.57	
Inverted MA Roots	.64	

Figura 21: Estimação do modelo ARMA(1,1)

Dependent Variable: DLOG(CLOSE_INTER)
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 06/10/20 Time: 13:11
Sample (adjusted): 1/05/2000 4/03/2019
Included observations: 5021 after adjustments
Failure to improve likelihood (non-zero gradients) after 5 iterations
Coefficient covariance computed using outer product of gradients
MA Backcast: 1/03/2000 1/04/2000

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	-0.161752	0.183991	-0.879127	0.3794
MA(1)	0.099383	0.183794	0.540729	0.5887
MA(2)	-0.068077	0.017434	-3.904855	0.0001

R-squared	0.007181	Mean dependent var	0.000143
Adjusted R-squared	0.006786	S.D. dependent var	0.011666
S.E. of regression	0.011626	Akaike info criterion	-6.070515
Sum squared resid	0.678275	Schwarz criterion	-6.066618
Log likelihood	15243.03	Hannan-Quinn criter.	-6.069149
Durbin-Watson stat	2.000309		

Inverted AR Roots	-.16	
Inverted MA Roots	.22	-.32

Figura 22: Estimação do modelo ARMA(1,2)

Dependent Variable: DLOG(CLOSE_INTER)
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 06/10/20 Time: 13:17
Sample (adjusted): 1/06/2000 4/03/2019
Included observations: 5020 after adjustments
Failure to improve likelihood (non-zero gradients) after 7 iterations
Coefficient covariance computed using outer product of gradients
MA Backcast: 1/05/2000

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	-0.196851	0.223547	-0.880580	0.3786
AR(2)	-0.069176	0.018101	-3.821558	0.0001
MA(1)	0.134391	0.224070	0.599772	0.5487

R-squared	0.007188	Mean dependent var	0.000143
Adjusted R-squared	0.006793	S.D. dependent var	0.011667
S.E. of regression	0.011627	Akaike info criterion	-6.070327
Sum squared resid	0.678267	Schwarz criterion	-6.066430
Log likelihood	15239.52	Hannan-Quinn criter.	-6.068961
Durbin-Watson stat	2.000080		

Inverted AR Roots	-.10+.24i	-.10-.24i
Inverted MA Roots	-.13	

Figura 23: Estimação do modelo ARMA(2,1)

Dependent Variable: DLOG(CLOSE_INTER)
Method: ARMA Conditional Least Squares (Gauss-Newton / Marquardt steps)
Date: 06/10/20 Time: 13:17
Sample (adjusted): 1/06/2000 4/03/2019
Included observations: 5020 after adjustments
Failure to improve likelihood (non-zero gradients) after 20 iterations
Coefficient covariance computed using outer product of gradients
MA Backcast: 1/04/2000 1/05/2000

Variable	Coefficient	Std. Error	t-Statistic	Prob.
AR(1)	-0.199466	0.248695	-0.802051	0.4226
AR(2)	-0.026786	0.157546	-0.170020	0.8650
MA(1)	0.136938	0.248592	0.550853	0.5818
MA(2)	-0.043117	0.164233	-0.262534	0.7929

R-squared	0.007202	Mean dependent var	0.000143
Adjusted R-squared	0.006608	S.D. dependent var	0.011667
S.E. of regression	0.011628	Akaike info criterion	-6.069942
Sum squared resid	0.678258	Schwarz criterion	-6.064745
Log likelihood	15239.55	Hannan-Quinn criter.	-6.068121
Durbin-Watson stat	1.999982		

Inverted AR Roots	-.10+.13i	-.10-.13i
Inverted MA Roots	.15	-.29

Figura 24: Estimação do modelo ARMA(2,2)

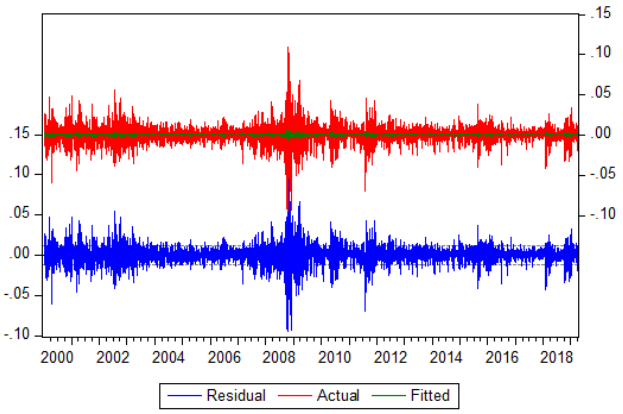


Figura 25: Gráfico de linhas AR(1)

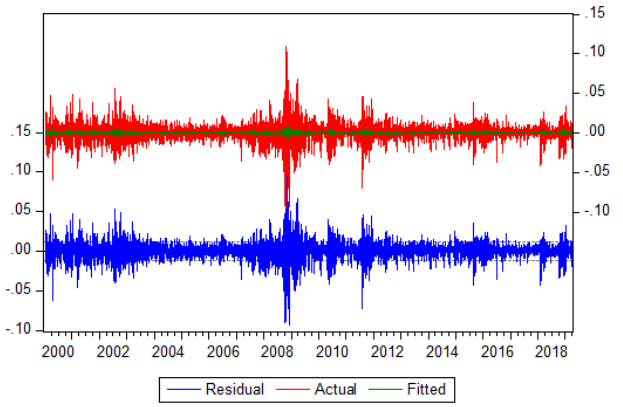


Figura 26: Gráfico de linhas AR(2)

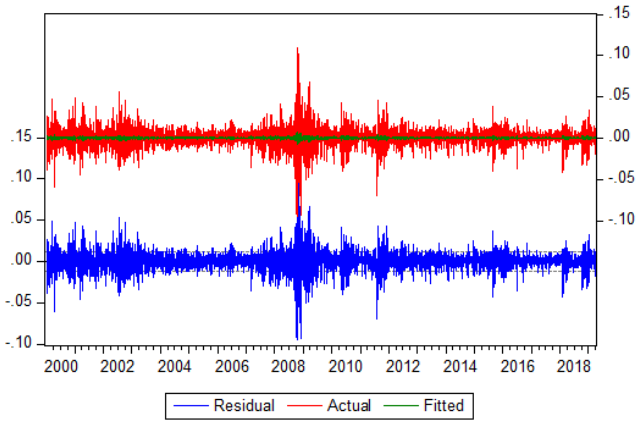


Figura 27: Gráfico de linhas MA(1)

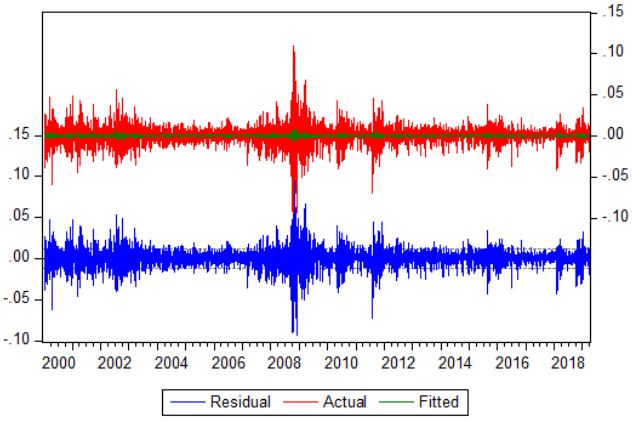


Figura 28: Gráfico de linhas MA(2)

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	9.361745	Prob. F(2,5017)	0.0001
Obs*R-squared	18.66875	Prob. Chi-Square(2)	0.0001

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 06/10/20 Time: 12:44
Sample: 1/05/2000 4/03/2019
Included observations: 5021
Coefficient covariance computed using outer product of gradients
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	1.96E-07	0.000155	0.001268	0.9990
AR(1)	0.032553	0.294977	0.110357	0.9121
RESID(-1)	-0.036465	0.295248	-0.123506	0.9017
RESID(-2)	-0.058908	0.022508	-2.617270	0.0089

R-squared	0.003718	Mean dependent var	3.74E-19
Adjusted R-squared	0.003122	S.D. dependent var	0.011645
S.E. of regression	0.011627	Akaike info criterion	-6.070188
Sum squared resid	0.678227	Schwarz criterion	-6.064993
Log likelihood	15243.21	Hannan-Quinn criter.	-6.068367
F-statistic	6.241163	Durbin-Watson stat	1.998537
Prob(F-statistic)	0.000317		

Figura 29: Teste de correlação serial Breusch-Godfrey do modelo AR(1)

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	0.285981	Prob. F(2,5015)	0.7513
Obs*R-squared	0.572466	Prob. Chi-Square(2)	0.7511

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 05/21/20 Time: 04:47
Sample: 1/06/2000 4/03/2019
Included observations: 5020
Coefficient covariance computed using outer product of gradients
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	6.81E-07	0.000146	0.004665	0.9963
AR(1)	-0.216934	0.290309	-0.747253	0.4549
AR(2)	0.086263	0.171821	0.502048	0.6157
RESID(-1)	0.217433	0.290704	0.747956	0.4545
RESID(-2)	-0.100125	0.183049	-0.546987	0.5844

R-squared	0.000114	Mean dependent var	-3.58E-19
Adjusted R-squared	-0.000683	S.D. dependent var	0.011624
S.E. of regression	0.011628	Akaike info criterion	-6.069767
Sum squared resid	0.678106	Schwarz criterion	-6.063272
Log likelihood	15240.12	Hannan-Quinn criter.	-6.067491
F-statistic	0.142990	Durbin-Watson stat	1.999993
Prob(F-statistic)	0.966119		

Figura 30: Teste de correlação serial Breusch-Godfrey do modelo AR(2)

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	10.82990	Prob. F(2,5018)	0.0000
Obs*R-squared	21.58390	Prob. Chi-Square(2)	0.0000

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 05/21/20 Time: 04:45
Sample: 1/04/2000 4/03/2019
Included observations: 5022
Coefficient covariance computed using outer product of gradients
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	3.95E-06	0.000153	0.025782	0.9794
MA(1)	-4.149043	1.787539	-2.321092	0.0203
RESID(-1)	4.152217	1.787099	2.323440	0.0202
RESID(-2)	0.224143	0.121803	1.840209	0.0658

R-squared	0.004298	Mean dependent var	-5.63E-07
Adjusted R-squared	0.003703	S.D. dependent var	0.011654
S.E. of regression	0.011633	Akaike info criterion	-6.069204
Sum squared resid	0.679030	Schwarz criterion	-6.064009
Log likelihood	15243.77	Hannan-Quinn criter.	-6.067383
F-statistic	7.219928	Durbin-Watson stat	1.996577
Prob(F-statistic)	0.000078		

Figura 31: Teste de correlação serial Breusch-Godfrey do modelo MA(1)

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	0.362468	Prob. F(2,5017)	0.6960
Obs*R-squared	0.725553	Prob. Chi-Square(2)	0.6957

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 05/21/20 Time: 04:39
Sample: 1/04/2000 4/03/2019
Included observations: 5022
Coefficient covariance computed using outer product of gradients
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	-1.83E-07	0.000144	-0.001269	0.9990
MA(1)	0.300293	0.366388	0.819604	0.4125
MA(2)	-0.205056	0.260129	-0.788284	0.4306
RESID(-1)	-0.300476	0.366533	-0.819779	0.4124
RESID(-2)	0.186562	0.242819	0.768317	0.4423

R-squared	0.000144	Mean dependent var	-1.01E-06
Adjusted R-squared	-0.000653	S.D. dependent var	0.011635
S.E. of regression	0.011639	Akaike info criterion	-6.067903
Sum squared resid	0.679643	Schwarz criterion	-6.061410
Log likelihood	15241.50	Hannan-Quinn criter.	-6.065628
F-statistic	0.181225	Durbin-Watson stat	1.998416
Prob(F-statistic)	0.948212		

Figura 32: Teste de correlação serial Breusch-Godfrey do modelo MA(2)

Breusch-Godfrey Serial Correlation LM Test:

F-statistic	2.370380	Prob. F(2,5016)	0.0935
Obs*R-squared	4.741005	Prob. Chi-Square(2)	0.0934

Test Equation:
Dependent Variable: RESID
Method: Least Squares
Date: 05/21/20 Time: 04:44
Sample: 1/05/2000 4/03/2019
Included observations: 5021
Coefficient covariance computed using outer product of gradients
Presample missing value lagged residuals set to zero.

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	2.58E-06	0.000139	0.018649	0.9851
AR(1)	0.370716	0.297219	1.247283	0.2124
MA(1)	-0.246564	0.223127	-1.105043	0.2692
RESID(-1)	-0.119826	0.081172	-1.476199	0.1400
RESID(-2)	-0.073146	0.035656	-2.051434	0.0403

R-squared	0.000944	Mean dependent var	6.52E-06
Adjusted R-squared	0.000147	S.D. dependent var	0.011632
S.E. of regression	0.011631	Akaike info criterion	-6.069334
Sum squared resid	0.678535	Schwarz criterion	-6.062840
Log likelihood	15242.06	Hannan-Quinn criter.	-6.067059
F-statistic	1.184796	Durbin-Watson stat	1.998487
Prob(F-statistic)	0.315264		

Figura 33: Teste de correlação serial Breusch-Godfrey do modelo ARMA(1,1)

Heteroskedasticity Test: ARCH

F-statistic	214.4831	Prob. F(1,5018)	0.0000
Obs*R-squared	205.7732	Prob. Chi-Square(1)	0.0000

Test Equation:
Dependent Variable: RESID^2
Method: Least Squares
Date: 06/10/20 Time: 12:45
Sample (adjusted): 1/06/2000 4/03/2019
Included observations: 5020 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000108	6.53E-06	16.57066	0.0000
RESID^2(-1)	0.202461	0.013824	14.64524	0.0000

R-squared	0.040991	Mean dependent var	0.000136
Adjusted R-squared	0.040800	S.D. dependent var	0.000452
S.E. of regression	0.000443	Akaike info criterion	-12.60583
Sum squared resid	0.000985	Schwarz criterion	-12.60323
Log likelihood	31642.63	Hannan-Quinn criter.	-12.60492
F-statistic	214.4831	Durbin-Watson stat	2.146392
Prob(F-statistic)	0.000000		

Figura 34: Teste ARCH, modelo AR(1)

Heteroskedasticity Test: ARCH

F-statistic	230.4984	Prob. F(1,5017)	0.0000
Obs*R-squared	220.4615	Prob. Chi-Square(1)	0.0000

Test Equation:
Dependent Variable: RESID^2
Method: Least Squares
Date: 06/10/20 Time: 12:46
Sample (adjusted): 1/07/2000 4/03/2019
Included observations: 5019 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000107	6.36E-06	16.78146	0.0000
RESID^2(-1)	0.209584	0.013805	15.18217	0.0000

R-squared	0.043925	Mean dependent var	0.000135
Adjusted R-squared	0.043735	S.D. dependent var	0.000441
S.E. of regression	0.000431	Akaike info criterion	-12.66014
Sum squared resid	0.000932	Schwarz criterion	-12.65754
Log likelihood	31772.63	Hannan-Quinn criter.	-12.65923
F-statistic	230.4984	Durbin-Watson stat	2.152070
Prob(F-statistic)	0.000000		

Figura 35: Teste ARCH, modelo AR(2)

Heteroskedasticity Test: ARCH

F-statistic	211.4361	Prob. F(1,5019)	0.0000
Obs*R-squared	202.9699	Prob. Chi-Square(1)	0.0000

Test Equation:
Dependent Variable: RESID^2
Method: Least Squares
Date: 06/10/20 Time: 12:47
Sample (adjusted): 1/05/2000 4/03/2019
Included observations: 5021 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000108	6.51E-06	16.63328	0.0000
RESID^2(-1)	0.200872	0.013814	14.54084	0.0000

R-squared	0.040424	Mean dependent var	0.000136
Adjusted R-squared	0.040233	S.D. dependent var	0.000451
S.E. of regression	0.000442	Akaike info criterion	-12.61200
Sum squared resid	0.000979	Schwarz criterion	-12.60940
Log likelihood	31664.43	Hannan-Quinn criter.	-12.61109
F-statistic	211.4361	Durbin-Watson stat	2.144304
Prob(F-statistic)	0.000000		

Figura 36: Teste ARCH, modelo MA(1)

Heteroskedasticity Test: ARCH

F-statistic	233.2628	Prob. F(1,5019)	0.0000
Obs*R-squared	222.9920	Prob. Chi-Square(1)	0.0000

Test Equation:
Dependent Variable: RESID^2
Method: Least Squares
Date: 06/10/20 Time: 12:48
Sample (adjusted): 1/05/2000 4/03/2019
Included observations: 5021 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000107	6.35E-06	16.77781	0.0000
RESID^2(-1)	0.210539	0.013785	15.27294	0.0000

R-squared	0.044412	Mean dependent var	0.000135
Adjusted R-squared	0.044221	S.D. dependent var	0.000440
S.E. of regression	0.000430	Akaike info criterion	-12.66365
Sum squared resid	0.000929	Schwarz criterion	-12.66105
Log likelihood	31794.08	Hannan-Quinn criter.	-12.66274
F-statistic	233.2628	Durbin-Watson stat	2.152076
Prob(F-statistic)	0.000000		

Figura 37: Teste ARCH, modelo MA(2)

Heteroskedasticity Test: ARCH

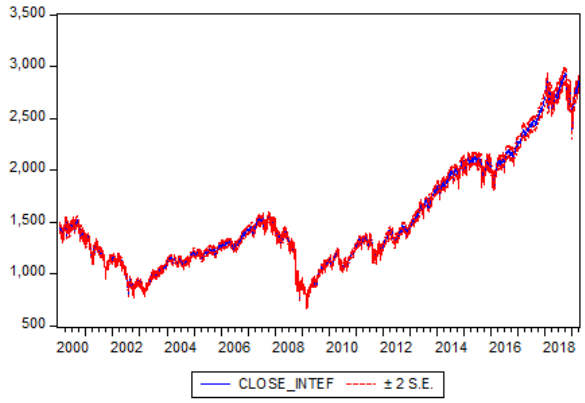
F-statistic	238.5440	Prob. F(1,5018)	0.0000
Obs*R-squared	227.8095	Prob. Chi-Square(1)	0.0000

Test Equation:
Dependent Variable: RESID^2
Method: Least Squares
Date: 06/10/20 Time: 12:47
Sample (adjusted): 1/06/2000 4/03/2019
Included observations: 5020 after adjustments

Variable	Coefficient	Std. Error	t-Statistic	Prob.
C	0.000106	6.39E-06	16.65913	0.0000
RESID^2(-1)	0.213028	0.013793	15.44487	0.0000

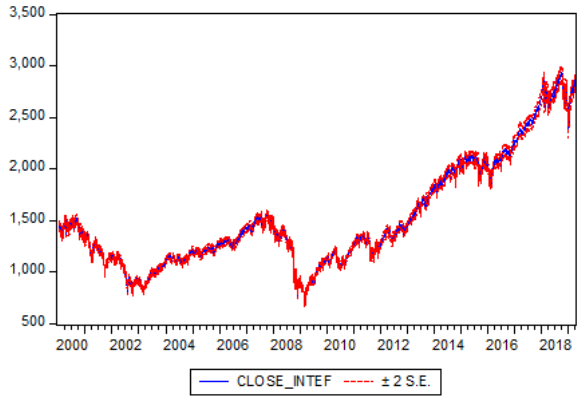
R-squared	0.045380	Mean dependent var	0.000135
Adjusted R-squared	0.045190	S.D. dependent var	0.000443
S.E. of regression	0.000433	Akaike info criterion	-12.65133
Sum squared resid	0.000941	Schwarz criterion	-12.64873
Log likelihood	31756.83	Hannan-Quinn criter.	-12.65042
F-statistic	238.5440	Durbin-Watson stat	2.156518
Prob(F-statistic)	0.000000		

Figura 38: Teste ARCH, modelo ARMA(1,1)



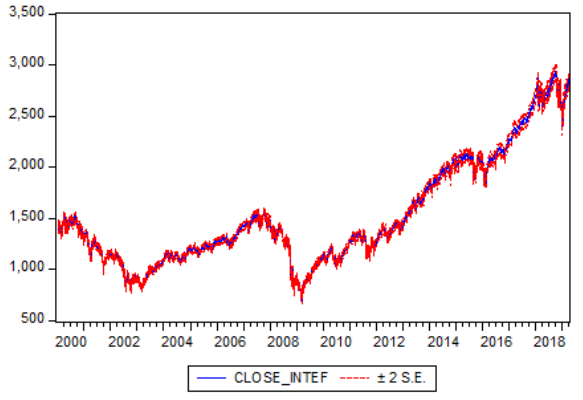
Forecast: CLOSE_INTEF	
Actual: CLOSE_INTER	
Forecast sample: 1/03/2000 4/03/2019	
Adjusted sample: 1/04/2000 4/03/2019	
Included observations: 5022	
Root Mean Squared Error	15.55914
Mean Absolute Error	10.71921
Mean Abs. Percent Error	0.772881
Theil Inequality Coefficient	0.004836
Bias Proportion	0.000453
Variance Proportion	0.000752
Covariance Proportion	0.997182
Theil U2 Coefficient	0.996095
Symmetric MAPE	0.772344

Figura 39: Previsão in-sample do modelo AR(2)



Forecast: CLOSE_INTEF	
Actual: CLOSE_INTER	
Forecast sample: 1/03/2000 4/03/2019	
Adjusted sample: 1/04/2000 4/03/2019	
Included observations: 5022	
Root Mean Squared Error	15.55914
Mean Absolute Error	10.71921
Mean Abs. Percent Error	0.772881
Theil Inequality Coefficient	0.004836
Bias Proportion	0.000453
Variance Proportion	0.000752
Covariance Proportion	0.998550
Theil U2 Coefficient	0.996095
Symmetric MAPE	0.772344

Figura 40: Previsão in-sample do modelo MA(2)



Forecast: CLOSE_INTEF	
Actual: CLOSE_INTER	
Forecast sample: 1/03/2000 4/03/2019	
Adjusted sample: 1/05/2000 4/03/2019	
Included observations: 5021	
Root Mean Squared Error	15.55326
Mean Absolute Error	10.71825
Mean Abs. Percent Error	0.772933
Theil Inequality Coefficient	0.004834
Bias Proportion	0.000557
Variance Proportion	0.000819
Covariance Proportion	0.998280
Theil U2 Coefficient	0.996653
Symmetric MAPE	0.772388

Figura 41: Previsão in-sample do modelo ARMA(1,1)

Anexo 2

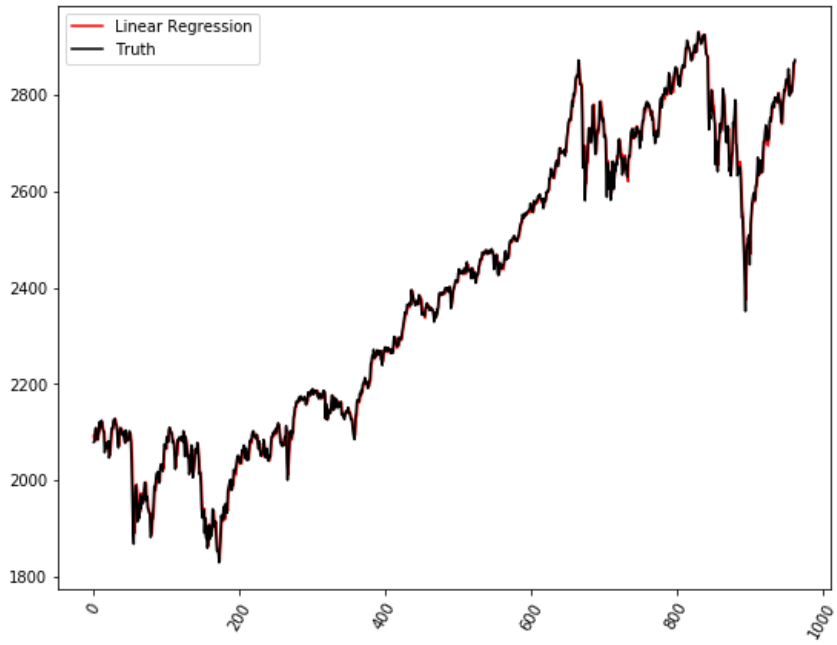


Figura 42: Gráfico de linhas da previsão in sample, LR Vs. Valor Real

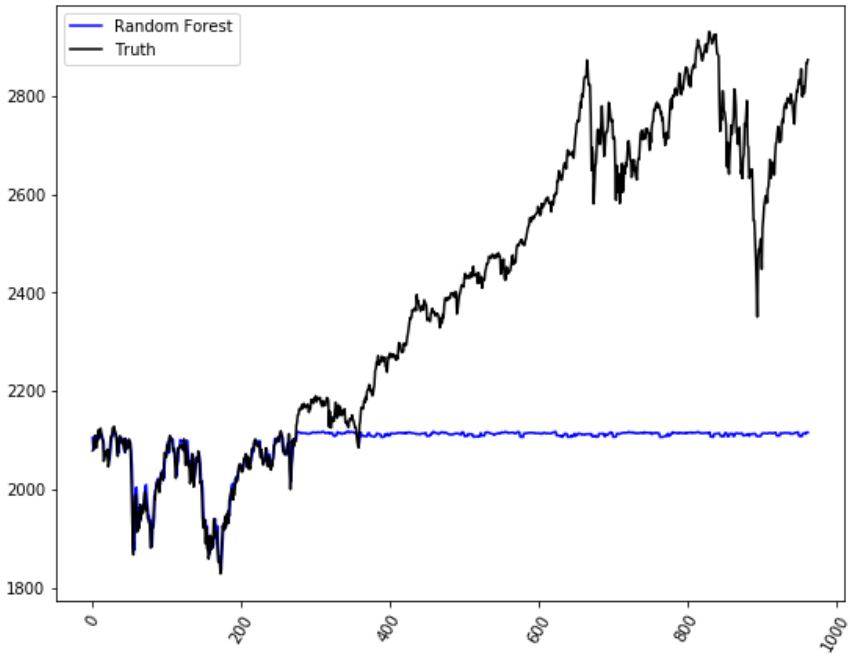


Figura 43: Gráfico de linhas da previsão in sample, RF Vs. Valor Real

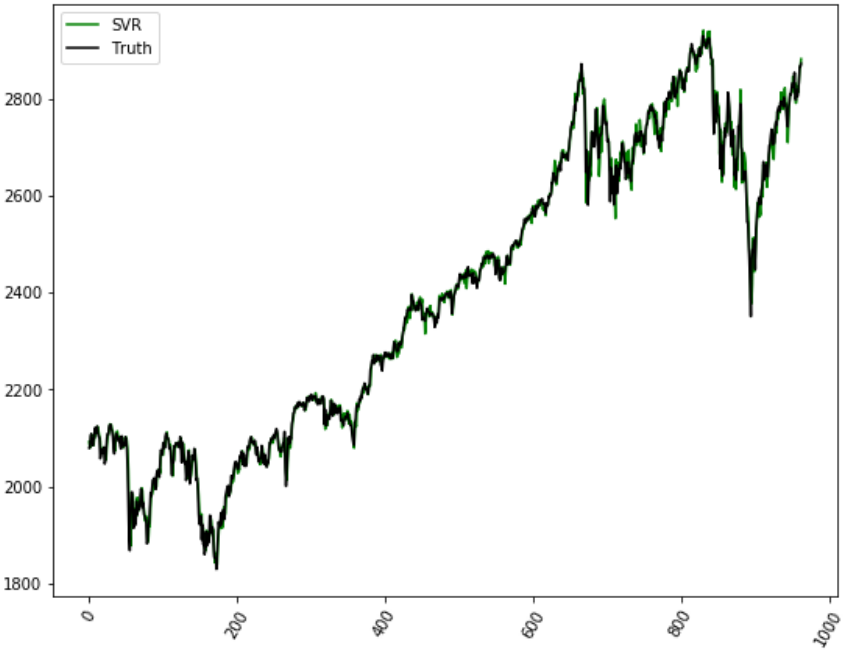


Figura 44: Gráfico de linhas da previsão in sample, SVR Vs. Valor Real

Linhas de código python

'''

Source codes for Python Machine Learning By Example 2nd Edition (Packt Publishing)

Chapter 9: Stock Price Prediction with Regression Algorithms

Author: Yuxi (Hayden) Liu

modified by: Amadú Baldé

'''

import pandas as pd

import numpy as np

from sklearn.model_selection import GridSearchCV

from sklearn.metrics import mean_squared_error, mean_absolute_error, r2_score

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import time

```
def generate_features(df):
    """
    Generate features for a stock/index based on historical price and performance
    @param df: dataframe with columns "Open", "Close", "High", "Low", "Volume", "Adjusted Close"
    @return: dataframe, data set with new features
    """
    df_new = pd.DataFrame()
    # 6 original features
    df_new['open'] = df['Open']
    df_new['open_1'] = df['Open'].shift(1)
    df_new['close_1'] = df['Close'].shift(1)
    df_new['high_1'] = df['High'].shift(1)
    df_new['low_1'] = df['Low'].shift(1)
    df_new['volume_1'] = df['Volume'].shift(1)
    # 31 generated features
    # average price
    df_new['avg_price_5'] = df['Close'].rolling(5).mean().shift(1)
    df_new['avg_price_30'] = df['Close'].rolling(21).mean().shift(1)
    df_new['avg_price_365'] = df['Close'].rolling(252).mean().shift(1)
    df_new['ratio_avg_price_5_30'] = df_new['avg_price_5'] / df_new['avg_price_30']
    df_new['ratio_avg_price_5_365'] = df_new['avg_price_5'] / df_new['avg_price_365']
    df_new['ratio_avg_price_30_365'] = df_new['avg_price_30'] / df_new['avg_price_365']
    # average volume
    df_new['avg_volume_5'] = df['Volume'].rolling(5).mean().shift(1)
    df_new['avg_volume_30'] = df['Volume'].rolling(21).mean().shift(1)
    df_new['avg_volume_365'] = df['Volume'].rolling(252).mean().shift(1)
    df_new['ratio_avg_volume_5_30'] = df_new['avg_volume_5'] / df_new['avg_volume_30']
    df_new['ratio_avg_volume_5_365'] = df_new['avg_volume_5'] / df_new['avg_volume_365']
    df_new['ratio_avg_volume_30_365'] = df_new['avg_volume_30'] / df_new['avg_volume_365']
    # standard deviation of prices
    df_new['std_price_5'] = df['Close'].rolling(5).std().shift(1)
    df_new['std_price_30'] = df['Close'].rolling(21).std().shift(1)
    df_new['std_price_365'] = df['Close'].rolling(252).std().shift(1)
    df_new['ratio_std_price_5_30'] = df_new['std_price_5'] / df_new['std_price_30']
```

```
    df_new['ratio_std_price_5_365'] = df_new['std_price_5'] / df_new['std_price_365']
    df_new['ratio_std_price_30_365'] = df_new['std_price_30'] / df_new['std_price_365']
    # standard deviation of volumes
    df_new['std_volume_5'] = df['Volume'].rolling(5).std().shift(1)
    df_new['std_volume_30'] = df['Volume'].rolling(21).std().shift(1)
    df_new['std_volume_365'] = df['Volume'].rolling(252).std().shift(1)
    df_new['ratio_std_volume_5_30'] = df_new['std_volume_5'] / df_new['std_volume_30']
    df_new['ratio_std_volume_5_365'] = df_new['std_volume_5'] / df_new['std_volume_365']
    df_new['ratio_std_volume_30_365'] = df_new['std_volume_30'] / df_new['std_volume_365']
    # # return
    df_new['return_1'] = ((df['Close'] - df['Close'].shift(1)) / df['Close'].shift(1)).shift(1)
    df_new['return_5'] = ((df['Close'] - df['Close'].shift(5)) / df['Close'].shift(5)).shift(1)
    df_new['return_30'] = ((df['Close'] - df['Close'].shift(21)) / df['Close'].shift(21)).shift(1)
    df_new['return_365'] = ((df['Close'] - df['Close'].shift(252)) / df['Close'].shift(252)).shift(1)
    df_new['moving_avg_5'] = df_new['return_1'].rolling(5).mean().shift(1)
    df_new['moving_avg_30'] = df_new['return_1'].rolling(21).mean().shift(1)
    df_new['moving_avg_365'] = df_new['return_1'].rolling(252).mean().shift(1)
    # the target
    df_new['close'] = df['Close']
    df_new = df_new.dropna(axis=0)
    return df_new

data_raw = pd.read_csv('SP2000_2019Day.csv')
print("checking if any null values are present\n", data_raw.isna().sum())
data_raw.head(5)
data_raw['Date'] = pd.to_datetime(data_raw.Date,format='%Y-%m-%d')
data_raw.index = data_raw['Date']

data = generate_features(data_raw)
print("checking if any null values are present\n", data.isna().sum())
print(data.head(5))
print(data.tail(5))
```

```
#plot
plt.figure(figsize=(10,5))
plt.plot(data['close'], label='close Price history')

start_train = '2001-01-03'
end_train = '2015-06-07'
start_test = '2015-06-08'
end_test = '2019-04-03'

data_train = data.loc[start_train:end_train]
X_train = data_train.drop('close', axis=1).values
y_train = data_train['close'].values
print(X_train.shape)
print(y_train.shape)

data_test = data.loc[start_test:end_test]
X_test = data_test.drop('close', axis=1).values
y_test = data_test['close'].values
print(X_test.shape)

# MAPE
def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

# First experiment with linear regression
scaler = StandardScaler()
X_scaled_train = scaler.fit_transform(X_train)
X_scaled_test = scaler.transform(X_test)
start_lr = time.time()
param_grid = {
        "alpha": [1e-5, 3e-5, 1e-4],
        "eta0": [0.01, 0.03, 0.1],
        }
```

```
# Regression
from sklearn.linear_model import SGDRegressor
lr = SGDRegressor(penalty='l2', n_iter=2000)
grid_search = GridSearchCV(lr, param_grid, cv=5, scoring='r2')
grid_search.fit(X_scaled_train, y_train)
print(grid_search.best_params_)
lr_best = grid_search.best_estimator_
predictions_lr = lr_best.predict(X_scaled_test)
print('Linear Regression')
print('MSE: {0:.3f}'.format(mean_squared_error(y_test, predictions_lr)))
print('RMSE: {0:.3f}'.format((mean_squared_error(y_test, predictions_lr))**0.5))
print('MAE: {0:.3f}'.format(mean_absolute_error(y_test, predictions_lr)))
print('MAPE: {0:.3f}'.format(mean_absolute_percentage_error(y_test, predictions_lr)))
print('R^2: {0:.3f}'.format(r2_score(y_test, predictions_lr)))

end_lr = time.time()
print('Tempo de execução LR : %f' % (end_lr - start_lr))

# Experiment with random forest
start_rf=time.time()

param_grid = {
    'max_depth': [50, 70, 80],
    'min_samples_split': [5, 10],
    'max_features': ['auto', 'sqrt'],
    'min_samples_leaf': [3, 5]
}

from sklearn.ensemble import RandomForestRegressor
rf = RandomForestRegressor(n_estimators=200, n_jobs=-1)
grid_search = GridSearchCV(rf, param_grid, cv=5, scoring='r2', n_jobs=-1)
grid_search.fit(X_train, y_train)
print(grid_search.best_params_)
rf_best = grid_search.best_estimator_
```

```
predictions_rf = rf_best.predict(X_test)
print('Random Forest Regressor')
print('MSE: {0:.3f}'.format(mean_squared_error(y_test, predictions_rf)))
print('RMSE: {0:.3f}'.format((mean_squared_error(y_test, predictions_rf))**0.5))
print('MAE: {0:.3f}'.format(mean_absolute_error(y_test, predictions_rf)))
print('MAPE: {0:.3f}'.format(mean_absolute_percentage_error(y_test, predictions_rf)))
print('R^2: {0:.3f}'.format(r2_score(y_test, predictions_rf)))

end_rf=time.time()
print('Tempo de execução RF : %f' % (end_rf - start_rf))
```

 # Experiment with SVR

```
start_svr=time.time()

param_grid = [
    {'kernel': ['linear'], 'C': [100, 300, 500], 'epsilon': [0.00003, 0.0001]},
    {'kernel': ['rbf'], 'gamma': [1e-3, 1e-4], 'C': [10, 100, 1000], 'epsilon': [0.00003, 0.0001]}
]

from sklearn.svm import SVR
svr = SVR()
grid_search = GridSearchCV(svr, param_grid, cv=2, scoring='r2')
grid_search.fit(X_scaled_train, y_train)
print(grid_search.best_params_)
svr_best = grid_search.best_estimator_
predictions_svr = svr_best.predict(X_scaled_test)
print('SVR')
print('MSE: {0:.3f}'.format(mean_squared_error(y_test, predictions_svr)))
print('RMSE: {0:.3f}'.format((mean_squared_error(y_test, predictions_svr))**0.5))
print('MAE: {0:.3f}'.format(mean_absolute_error(y_test, predictions_svr)))
print('MAPE: {0:.3f}'.format(mean_absolute_percentage_error(y_test, predictions_svr)))
print('R^2: {0:.3f}'.format(r2_score(y_test, predictions_svr)))
```

```
end_svr = time.time()
print('Tempo de execução SVR : %f' % (end_svr - start_svr))

#Plot
plt.figure(figsize=(9,7))

#Linear Reegression
plt.plot(predictions_lr, color='red', label='Linear Regression')
plt.plot(y_test, color = 'black', label = 'Truth')
# Add a legend in the top left corner of the plot
plt.legend(loc='upper left')

# Specify the orientation of the xticks
plt.xticks(rotation=60)

# Display the plot
plt.show()

#Random Forest
plt.figure(figsize=(9,7))
plt.plot(predictions_rf, color='blue', label='Random Forest')
plt.plot(y_test, color = 'black', label = 'Truth')
plt.legend(loc='upper left')
plt.xticks(rotation=60)
plt.show()

#SVR
plt.figure(figsize=(9,7))
plt.plot(predictions_svr, color='green', label='SVR')
plt.plot(y_test, color = 'black', label = 'Truth')
plt.legend(loc='upper left')
plt.xticks(rotation=60)
plt.show()
```

```
#compare
plt.figure(figsize=(9,7))

plt.plot(predictions_lr, color='red', label='Linear Regression')
plt.plot(predictions_rf, color='blue', label='Random Forest')
plt.plot(predictions_svr, color='green', label='SVR')
plt.plot(y_test, color = 'black', label = 'Truth')
plt.legend(loc='upper left')
plt.xticks(rotation=60)
plt.show()
```