

UNIVERSIDADE FEDERAL FLUMINENSE
INSTITUTO DE CIÊNCIA E TECNOLOGIA
MESTRADO PROFISSIONAL EM ENGENHARIA DE PRODUÇÃO E SISTEMAS
COMPUTACIONAIS

VINÍCIUS BARBOSA PAIVA

**ANÁLISE DO DESEMPENHO DE REGRA DE
NEGOCIAÇÃO VIA REDES NEURAIS ARTIFICIAIS EM
OPERAÇÕES *DAY TRADE***

Rio das Ostras, RJ

2021

VINÍCIUS BARBOSA PAIVA

**ANÁLISE DO DESEMPENHO DE REGRA DE
NEGOCIAÇÃO VIA REDES NEURAIS ARTIFICIAIS EM
OPERAÇÕES *DAY TRADE***

Dissertação apresentada ao Programa de Pós-graduação em Engenharia de Produção e Sistemas Computacionais, como requisito parcial à obtenção do título de Mestre em Engenharia de Produção e Sistemas Computacionais.

Linha de Pesquisa: Pesquisa Operacional e Simulação

Orientador: Profa. Dra. ANA PAULA BARBOSA SOBRAL

Rio das Ostras, RJ
2021

Ficha catalográfica automática - SDC/BRO
Gerada com informações fornecidas pelo autor

P142a Paiva, Vinícius Barbosa
Análise do desempenho de regra de negociação via redes neurais artificiais em operações day trade / Vinícius Barbosa Paiva ; Ana Paula Barbosa Sobral, orientadora. Niterói, 2021.
198 f. : il.

Dissertação (mestrado profissional)-Universidade Federal Fluminense, Rio das Ostras, 2021.

DOI: <http://dx.doi.org/10.22409/PPG-MESC.2021.m.11433932792>

1. Redes neurais artificiais. 2. Análise técnica. 3. Regra de negociação. 4. Produção intelectual. I. Sobral, Ana Paula Barbosa, orientadora. II. Universidade Federal Fluminense. Instituto de Ciência e Tecnologia. III. Título.

CDD -

Dedico este trabalho, à minha filha e esposa, Cecília de Souza Paiva e Letícia Paula de Souza. Vocês são minha motivação para nunca parar.

AGRADECIMENTOS

Agradeço, primeiramente, a Deus por permitir chegar até aqui.

Agradeço e dedico este trabalho para a minha esposa. Lembro-me, quando iniciei o curso, há época grávida e sozinha em casa, aguardava pacientemente minha chegada. Mesmo cansada, por muitas vezes, dispondo de um mal-estar decorrente da gravidez, sempre estava disposta a me apoiar. Só consegui finalizar este projeto graças a ti minha vida.

Agradeço aos meus pais, que apesar de todos as dificuldades, me ajudaram na realização do meu sonho.

Quero agradecer a minha professora orientadora Ana Paula Sobral, pelo empenho dedicado ao meu projeto de pesquisa.

Agradeço a todas as pessoas que direta ou indiretamente contribuíram para a realização da minha pesquisa.

Por fim, agradeço a CAPES pela contribuição a todas as pesquisas acadêmicas.

RESUMO

Determinar quando comprar ou vender determinado ativo financeiro é um tópico complexo para a maioria dos *daytraders*. No entanto, visando minimizar esta complexidade, muitos destes têm desenvolvido regras e procedimentos lógicos conhecidos como regras de negociação. As regras de negociação adotam como premissa que os padrões passados nos preços de mercado se repetirão no futuro. Contudo, com auxílio do rápido desenvolvimento da ciência, cada vez mais ferramentas estão sendo desenvolvidas visando aferir a acurácia de uma regra de negociação. Como exemplo, é possível citar estudos nas áreas de aprendizado de máquina, modelos de redes neurais e assim por diante. Este trabalho propõe medir a precisão da regra de negociação conhecida como cruzamento da média móvel. Para esta tarefa, foi proposto a aplicação do modelo de Rede Neural Profunda (DNN) com base na classificação binária. O ativo financeiro escolhido para a pesquisa foi o Mini Contrato Futuro do Dólar, a partir do segundo semestre de 2019. Para o modelo DNN, foram definidas 22 variáveis. Dentre estas: o preço de fechamento, preço de abertura, máxima do preço e a mínima do preço. Coletados a partir de um robô de negociações, tais variáveis serviram de parâmetros de entrada para a arquitetura DNN proposta. Os resultados finais mostraram que a acurácia encontrada para a regra de negociação de cruzamento de médias móveis, a partir do modelo DNN, apresentou uma baixa porcentagem de predição.

Palavras-chave: Regra de Negociação. Redes Neurais Artificiais. Análise Técnica. Mini Contrato Futuro do Dólar.

ABSTRACT

Predict whether a buy and sell operation will be profitable, is a very hot topic for the daytraders. So, many of them have been developing rules and logical procedures known as trading rules. The principle of trading rules has the premise that the patterns in market prices to recur in the future. Thanks to the rapid development of science, predict whether the operation will be profitable is advancing. Several studies have been developing by using different ways now. As an example, we can including machine learning, deep learning and so on. This paper proposes to measure the accuracy of the trading rule known as moving average crossing. For this task, we proposed a Deep Neural Network (DNN) model based on the binary classification. The financial asset chosen for research was Mini U.S. Dollar, from the second semester of 2019. For the DNN model, we set 22 features,i.e., close price, opening price, high price, low price. Deriving from the expert advisor, the features will serve as the input of the DNN architecture and then run and test the program. The final results showed that the accuracy found for the trading rule of crossing of moving averages, based on the DNN model, presented a low percentage of prediction.

Palavras-chave: Trading Rule. Deep Neural Network. Technical Analysis. Mini U.S. Dollar.

LISTA DE ILUSTRAÇÕES

Figura 1 – Dinâmica entre o número de busca pelo termo “ <i>day trade</i> ” em relação ao aumento da quantidade de <i>day traders</i>	15
Figura 2 – Classificação da Metodologia de Pesquisa.	18
Figura 3 – Diagrama estratégico do Mapeamento Bibliométrico.	25
Figura 4 – Diagrama estratégico do período de 1990 a 2000.	26
Figura 5 – Diagrama estratégico do período de 2001 a 2010.	27
Figura 6 – Diagrama estratégico do período de 2011 a 2019.	28
Figura 7 – Diagrama estratégico do período de 2006 a 2010.	30
Figura 8 – Diagrama estratégico do período de 2011 a 2015.	31
Figura 9 – Diagrama estratégico do período de 2016 a 2019.	33
Figura 10 – Roteiro da Fundamentação Teórica.	37
Figura 11 – Evolução no número de investidores pessoas físicas na B3.	39
Figura 12 – Representação de uma operação lucrativa.	41
Figura 13 – Exemplo de um gráfico padrão <i>Candlestick</i>	43
Figura 14 – Interpretando o padrão <i>Candlestick</i>	44
Figura 15 – Média móvel simples de 20 períodos na ação PETR4 (PN).	46
Figura 16 – Regra de negociação baseada em cruzamento de médias móveis simples.	47
Figura 17 – Média móvel simples e exponencial de 20 períodos na ação PETR4 (PN)	47
Figura 18 – Regra de negociação 9,1 para compra, criada pelo <i>trader</i> Larry Williams.	48
Figura 19 – Neurônio artificial.	50
Figura 20 – Arquitetura ANN.	53
Figura 21 – Diagrama de rede para a rede neural de duas camadas correspondente a equação.	54
Figura 22 – Gráfico função linear.	56
Figura 23 – Gráfico função <i>Relu</i>	57
Figura 24 – Gráfico função <i>sigmoid</i>	57
Figura 25 – Gráfico função <i>tanh</i>	58
Figura 26 – Gráfico função <i>LRelu</i>	58
Figura 27 – Entrada das variáveis independentes no neurônio artificial.	59
Figura 28 – Processo de saída do valor previsto no neurônio artificial.	60
Figura 29 – Exemplificação do procedimento backpropagation.	60
Figura 30 – Neurônio artificial treinado.	61
Figura 31 – Gráfico <i>Cross-entropy Loss</i>	63
Figura 32 – Validação Cruzada com 10 Agrupamentos.	64
Figura 33 – Método Proposto para a Condução do <i>Design Science Research</i>	69
Figura 34 – Produtos da <i>Design Science Research</i>	70
Figura 35 – Produtos da <i>Design Science Research</i> (artefatos).	71

Figura 36 – Estrutura da dissertação conforme metodologia <i>Design Science Research</i> .	73
Figura 37 – Regra de negociação para a compra.	78
Figura 38 – Regra de negociação para a venda.	79
Figura 39 – Gráfico das operações por dias da semana.	82
Figura 40 – Gráfico das operações por meses.	83
Figura 41 – Gráfico de operações por horário.	84
Figura 42 – Gráfico de barras e histograma dos pontos avançados.	85
Figura 43 – Gráfico das médias por grupo dos pontos avançados.	86
Figura 44 – Gráfico demonstrando a relação das instituições financeiras com as entradas no sentido da compra em relação à regra de negociação.	87
Figura 45 – Gráfico demonstrando a relação das instituições financeiras com as entradas no sentido da venda em relação à regra de negociação.	88
Figura 46 – Gráfico com a correlação das variáveis independentes com a variável dependente.	89
Figura 47 – Etapas do treinamento do modelo de redes neurais artificiais profunda. .	90
Figura 48 – Arquitetura do modelo de rede neural profundo do estudo.	94
Figura 49 – Acurácia do banco de dados de treino e teste.	97
Figura 50 – Perda do modelo para que os pesos possam ser atualizados.	98

LISTA DE TABELAS

Tabela 1 – Desempenho dos temas nos períodos de 1990 a 2000.	26
Tabela 2 – Desempenho dos temas nos períodos de 2001 a 2010.	27
Tabela 3 – Desempenho dos temas nos períodos de 2010 a 2019.	29
Tabela 4 – Desempenho dos temas nos períodos de 2006 a 2010.	30
Tabela 5 – Desempenho dos temas nos períodos de 2011 a 2015.	32
Tabela 6 – Desempenho dos temas nos períodos de 2016 a 2019.	33
Tabela 7 – Características técnicas do mini contrato do dólar futuro negociado na B3.	40
Tabela 8 – Artefatos e Conceitos Chaves.	71
Tabela 9 – Tabela com as variáveis coletadas a partir do <i>backtesting</i> da regra de negociação.	75
Tabela 10 – Resumo da estratégia.	76
Tabela 11 – Premissas para a execução da regra de negociação.	77
Tabela 12 – Tabela da matriz confusão (<i>confusion matrix</i>).	79
Tabela 13 – Parâmetros adotados para o treinamento do modelo.	95
Tabela 14 – Cinco maiores acurácias segundo os valores adotados para as camadas de abandono.	96
Tabela 15 – Valores da acurácia utilizando a validação cruzada.	99
Tabela 16 – Modelo de protocolo de pesquisa para a proposição de artefatos para resolver o problema específico.	100
Tabela 17 – Cumprimento dos Objetivos Específicos.	105

LISTA DE ABREVIATURAS E SIGLAS

ANN	Artificial Neural Network
B3	Bolsa de Valores de São Paulo
CV	Validação cruzada (cross validation)
CVM	Comissão de Valores Mobiliários
DNN	Deep neural network
IFR	Índice de Força Relativa
ISF	International Symposium on Forecasting
KOSPI	Korea Composite Stock Price Index
MME	Média Móvel Exponencial
MMS	Media Móvel Simples
MSE	Mean squared Error
NASDAQ	National Association of Securities Dealers Automated Quotations
NYSE	New York Stock Exchange
PETR4	Código da ação da Petrobras
PN	Preferencial
RNN	Recurrent neural network
SciMAT	Science Mapping Analysis software Tool
S&P 500	Standard & Poor's 500
SSE	Shanghai Stock Exchange
TWSE	Taiwan Capitalization Weighted Stock Index
WDO	Mini Contrato Futuro do Dólar

SUMÁRIO

1	INTRODUÇÃO	13
1.1	CONTEXTUALIZAÇÃO	13
1.2	PROBLEMA	14
1.3	OBJETIVOS	16
1.4	JUSTIFICATIVA	16
1.5	DELIMITAÇÃO DO TRABALHO	17
1.6	METODOLOGIA DE PESQUISA	17
1.7	ESTRUTURA DO TRABALHO	19
2	MAPEAMENTO BIBLIOMÉTRICO	20
2.1	PROCEDIMENTO ADOTADO	20
2.2	ANÁLISE DO MAPEAMENTO BIBLIOMÉTRICO	25
2.2.1	Mapeamento Bibliométrico para Redes Neurais Artificiais	25
2.2.2	Mapeamento Bibliométrico da Análise Técnica	29
2.3	Ponderação Sobre a Relevância dos Temas para a Pesquisa Científica	34
2.3.1	Relevância do Tema Redes Neurais Artificiais	34
2.3.2	Relevância do Tema Análise Técnica	35
3	FUNDAMENTAÇÃO TEÓRICA	37
3.1	MINI CONTRATO FUTURO DO DÓLAR	38
3.2	ANÁLISE TÉCNICA	41
3.2.1	Padrão de <i>Candlestick</i>	42
3.2.2	Média Móvel	44
3.2.2.1	Média Móvel Simples	45
3.2.2.2	Média Móvel Exponencial	47
3.3	REDES NEURAIS ARTIFICIAIS	48
3.3.1	Neurônio Artificial	49
3.3.2	Topologia da Rede Neural Artificial	51
3.3.2.1	Redes Neurais Artificiais Profundas	53
3.3.2.1.1	<i>Camada de Abandono (Dropout)</i>	55
3.3.3	Função de Ativação das Camadas Ocultas	55
3.3.4	Processo de Aprendizagem da Rede Neural Artificial	59
3.3.4.1	Estimativa de Momento Adaptativo (<i>Adam</i>)	61
3.3.4.2	Função de Custo (<i>Cost Function</i>)	62
3.3.4.2.1	<i>Validação Cruzada (K-fold Cross Validation)</i>	64
3.3.4.3	Algoritmo <i>Backpropagation</i>	65

4	METODOLOGIA DO TRABALHO	68
4.1	ANÁLISE DA REGRA DE NEGOCIAÇÃO	74
4.1.1	Levantamento dos Dados	74
4.1.2	Execução da Regra de Negociação	77
4.1.3	Matriz de Confusão (<i>Confusion Matrix</i>)	79
4.2	ANÁLISE DESCRITIVA DA REGRA DE NEGOCIAÇÃO	81
4.3	TREINAMENTO DO MODELO DE REDES NEURAIS ARTIFICIAIS PRO-	
	FUNDA	90
4.3.1	Pré-processamento dos Dados	90
4.3.2	Arquitetura da Rede Neural Profunda do Modelo	92
4.3.3	Processo de Aprendizagem	94
4.3.4	Validação do Modelo de Redes Neurais Profunda	98
5	CONCLUSÃO	100
5.1	ANÁLISE DOS ARTEFATOS EMPREGADOS	100
5.2	CONSIDERAÇÕES FINAIS	103
	REFERÊNCIAS	108
	APÊNDICES	113
	APÊNDICE A – CÓDIGO FONTES DAS APLICAÇÕES UTILIZADAS .	114

1 INTRODUÇÃO

1.1 CONTEXTUALIZAÇÃO

Regras de negociação são amplamente utilizadas por investidores individuais, investidores institucionais ou por sistemas automatizados de negociação¹. Tratando-se de um conjunto de regras e procedimentos lógicos perfeitamente definidos, as regras de negociação permitem aos investidores mecanismos que auxiliam a tomada de decisão no momento da compra ou venda de um determinado ativo financeiro. Isto acontece porque uma vez padronizadas as operações de compra e venda, possibilita ao investidor mensurar a assertividade do padrão lógico adotado, oferecendo assim, subterfúgios estatísticos que acabam corroborando a efetividade da regra de negociação.

A partir da absorção e processamento das informações globais e específicas de cada ativo financeiro durante a formação dos preços ao longo do dia, dados são gerados e fornecidos ao investidor através dos *home broker's*². Informações como: volume financeiro negociado, preço de abertura e fechamento, ou até mesmo, quais bancos e corretoras estão negociando um determinado ativo financeiro, são processados e armazenados para elaboração de gráficos de preços e indicadores técnicos que, *a priori*, são as ferramentas bases para a realização da análise técnica.

A utilização de gráficos de preços, em conjunto com indicadores técnicos, é de fato a base das maiorias regras de negociação. Através desta união, analistas e investidores investigam a trajetória passada dos preços visando encontrar algum padrão, para que futuramente, este padrão seja capaz de ser replicado em novas operações.

Contudo, a aplicação das regras de negociação exige um intenso nível de disciplina por parte do investidor. Esta disciplina se torna mais fácil de ser administrada quando esta regra é aplicada num ativo financeiro no longo prazo, onde a precificação se aproxima da realidade, e por consequência, os resultados gerados pelos indicadores técnicos estão mais filtrados em relação aos ruídos produzidos pelo mercado (notícias, calendários econômicos, catástrofes, etc.). Porém, tais regras de negociações também são empregadas em operações financeiras num curto período de tempo. Comumente chamado de *day trading*, esta modalidade de negociação permite ao investidor a possibilidade de ganhos rápidos a partir da volatilização dos preços através de operações de compra e venda obrigatoriamente encerradas no mesmo dia. O lucro de uma operação *day trading* é calculado entra a diferença do preço médio de venda em relação ao preço médio de compra. Sendo assim, em minutos, o *trader*³ procura captar o maior número de oportunidades a partir da especulação do ativo

¹ Também conhecidos como robôs de negociação.

² *Home Broker* é um sistema que possibilita a negociação de ações e outros ativos financeiros diretamente através da internet. Foi implantado em 1999 e seu principal objetivo é facilitar o acesso ao mercado acionário e permitir que cada vez mais pessoas façam parte dele, de forma simples e eficiente (Investpedia).

³ *Trader* é um investidor do mercado financeiro que busca ganhar dinheiro com operações de curto prazo, aproveitando-se da volatilidade do mercado (infomoney).

financeiro. O investidor pessoal que realiza esta modalidade de operações de *trading* é comumente denominado de *day trader*.

O aumento de investidores pessoa física realizando a prática de negociação *day trade* na B3⁴ vem aumentando com o passar dos anos. Segundo Chague e Giovannetti (2019), por causa da comodidade proporcionada com o avanço tecnológico referente ao compartilhamento de informação, tornou-se acessível para qualquer indivíduo participar à distância de cursos e acompanhamento de salas de *day trade* na internet. Outro fator que possibilitou esta expansão de *traders* individuais foi a disponibilização, por parte das corretoras de valores, diversos *softwares* de *trading*. Tais *softwares* acabam dispondo de instrumentos e indicadores técnicos utilizados para compor regras de negociação.

1.2 PROBLEMA

Os resultados nas negociações realizadas pelos *day traders* não têm demonstrado um cenário positivo. Conforme Chague e Giovannetti (2019), apenas no mini contrato futuro do dólar, um dos ativos que mais concentram investidores individuais, 14.748 pessoas iniciaram a atividade de *day trade* entre 2013 e 2015; dessas, 13.617 (92,3%) desistiram do *day trade* neste ativo. Das 1.131 pessoas que persistiram por mais 300 pregões, ao somarmos o lucro obtido por todos, chegamos ao valor negativo de -R\$42,0 milhões (-R\$38,1 milhões dos resultados das operações e -R\$3,9 milhões dos custos de emolumentos e taxa de registro variável cobrado pela B3).

Embora os dados apresentados pelos autores Chague e Giovannetti (2019) alertem que investidores individuais percam mais dinheiro do que ganham em operações *day trade*, com o passar dos anos, cada vez mais é notado um aumento pela procura por esta modalidade de investimento. Tal aumento pode ser visto a partir da procura do termo “*day trade*” em computadores localizados no Brasil e pelo aumento de novos *day traders* operando na B3. Conforme Figura 1, é traçado um paralelo entre o aumento pela busca do termo “*day trade*” e os novos *day traders*.

⁴ Bolsa de Valores de São Paulo

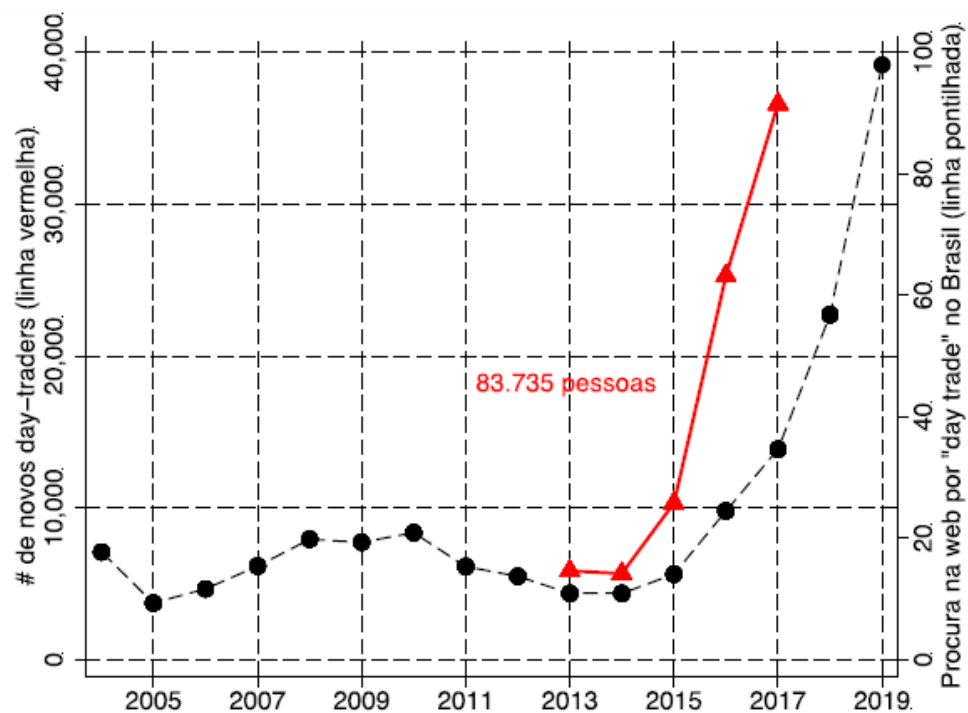


Figura 1 – Dinâmica entre o número de busca pelo termo “day trade” em relação ao aumento da quantidade de day traders.

Fonte: Chague (2019).

Uma das possíveis motivações para tal aumento é a facilidade e alcance de novas tecnologias. Cursos e salas de acompanhamento de *trade* são ofertados em abundância por corretoras e instituições de ensino. Somado a esses fatores, corretoras de valores disponibilizam pacotes e descontos em corretagem e maiores limites de alavancagem⁵.

Até o artigo Chague e Giovannetti (2019), não havia nenhuma evidência a respeito de qual é resultado obtido por *traders* que decidem investir na carreira de *day trade* no mini contrato futuro do dólar.

No entanto, Chague e Giovannetti (2019) apresentaram fortes evidências sobre a perspectiva pela busca de ganhos ligeiros em operações *day trade*. Ao menos economicamente, os autores indicaram que a chance de obter uma renda significativa é mínima.

A chance de obter uma renda significativa (por exemplo, maior do que R\$ 300,00 por dia) é remota para as pessoas que persistem na atividade. Por outro lado, a chance de se obter prejuízo é muito elevada. Além disso, os dados indicam também que à medida que o day-trader vai persistindo na atividade seu desempenho tende a ir piorando.

Sendo assim, apresentado este cenário, dentre os diversos fatores que impedem o *day trader* de não obter um resultado consistente e lucrativo no mini contrato futuro do

⁵ Alavancagem é uma técnica de maximizar a rentabilidade por meio de uma espécie de endividamento (*Easynvest*).

dólar, para muitos *traders*, é a falta de uma regra de negociação com alta assertividade. A busca por parametrizar uma regra de negociação que forneça um retorno recompensador concebeu diversos fóruns na *internet* provendo regras de negociação para alugar, ou até mesmo vender, para qualquer *trader* que tenha interesse no assunto. Cursos e metodologias de *trading* apresentam *setups*⁶ personalizados e adaptados para cada perfil de *day trader* prometendo retornos financeiros plausíveis.

Nesse sentido, esforços devem focar no estudo e desenvolvimento de regras de negociação para que novas práticas possam ser aplicadas de forma mais responsiva e eficiente. O devido empenho pode ajudar os *day traders*, que operam o mini contrato futuro do dólar, a promover novos meios que auxiliem o processo decisório.

1.3 OBJETIVOS

O objetivo deste trabalho é contribuir com melhorias na prática do day trading, visando prever se as operações de compra ou de venda, a partir da regra de negociação empregada pelo autor, serão lucrativas ou não. Para tal fim foi empregado o modelo de redes neurais artificiais profunda aspirando aferir a acurácia do conjunto de variáveis empregada na regra de negociação.

Para que o objetivo geral seja alcançado, devem-se atingir os seguintes objetivos específicos:

- Criação de um algoritmo computadorizado (robô de negociação) que identifique as entradas de compra e venda da regra de negociação;
- Determinar as variáveis de entrada que serão analisadas na ocorrência da regra de negociação;
- Identificar as operações de compra e de venda que atingiram os pontos necessários para serem operações lucrativas;
- Montar um banco de dados com as variáveis baseadas nas entradas de compra e venda do robô de negociação;
- Codificar o modelo de redes neurais artificiais profundas em *Python*;
- Medir a acurácia do modelo de redes neurais artificiais profunda e atestar a assertividade da regra de negociação.

1.4 JUSTIFICATIVA

Devido ao alto índice de desistência e de perda de capital financeiro por parte dos *day traders* pessoa física, que realizam negociações de compra e venda no mini contrato

⁶ Sinônimo para regra de negociação.

do dólar, justifica-se a necessidade em elaborar um estudo estatístico de uma das regras de negociação mais utilizadas e abordadas em livros de análise técnica, o cruzamento de médias móveis.

Dessa forma, faz-se necessário um alinhamento entre as ferramentas já existentes de controle quantitativo com a aplicação de redes neurais artificiais profunda, agregando valor a regra de negociação e provendo dados que auxiliam a tomada de decisão dos *day traders* pessoa física. Além de servir para a comunidade científica mais um estudo referente à prática de *day trade* no Brasil.

1.5 DELIMITAÇÃO DO TRABALHO

Este trabalho se restringe a analisar o desempenho da regra de negociação cruzamento de médias móveis. Não estando em seu escopo o estudo de outras técnicas e métodos. Da mesma forma, o mini contrato futuro do dólar consistirá na base para aplicabilidade do trabalho, sendo o ativo financeiro escolhido para a realização do estudo. Sendo assim, não foi possível garantir a replicação dos resultados para outros ativos financeiros devido às especificidades do ativo financeiro mini contrato futuro do dólar.

1.6 METODOLOGIA DE PESQUISA

Com o objetivo de nortear o pesquisador, as pesquisas científicas visam seguir padrões. Tais padrões adotam como premissa a classificação à sua natureza, abordagem, objetivos e procedimentos técnicos. A Figura 2 visa representar o fluxo classificatório adotado no trabalho.

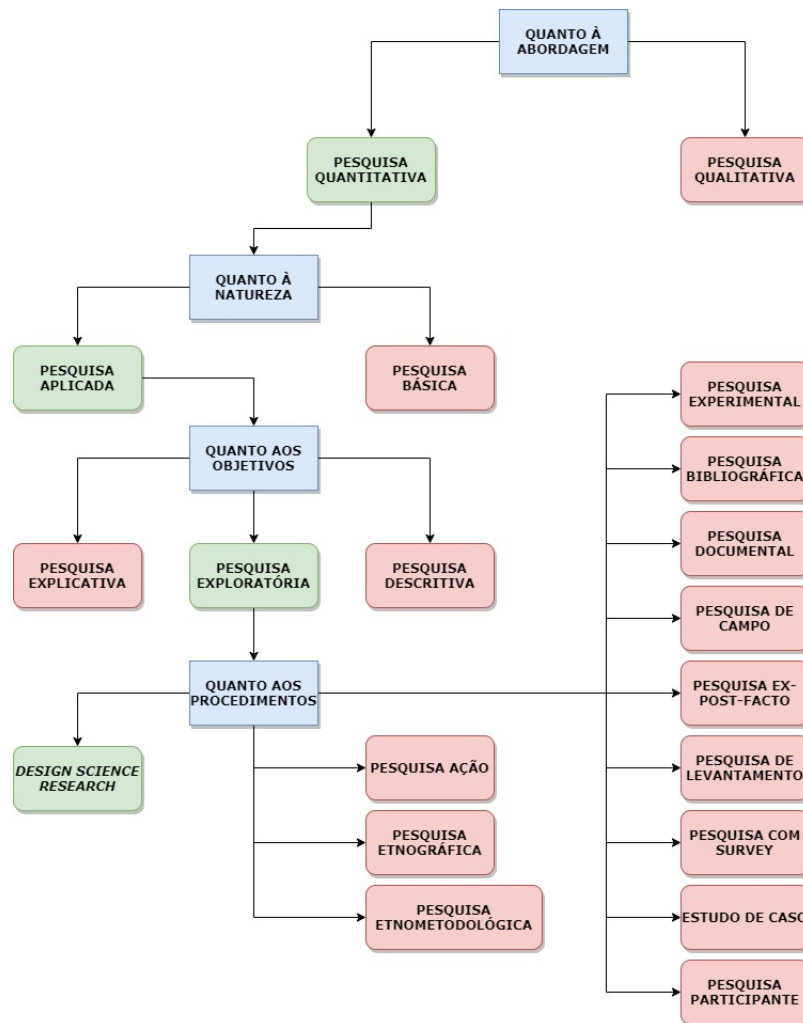


Figura 2 – Classificação da Metodologia de Pesquisa.

Fonte: Gil (2010).

No que se refere à abordagem e natureza, este trabalho é classificado como uma pesquisa quantitativa aplicada. A geração de conhecimentos para a aplicação, na prática, são direcionados às soluções de problemas específicos. E, de certa forma, enfatizando a objetividade na coleta e análise dos dados numéricos, assim como a utilização da estatística nesta análise (GIL, 2010).

Quanto aos objetivos, esta é uma pesquisa exploratória, pois, visa proporcionar maior familiaridade com o problema com vistas a torná-lo explícito, envolvendo levantamento bibliográfico e análise de exemplos que estimulem a compreensão (MELLO; TURRIONI, 2012).

O método adotado nessa dissertação foi a *Design Science Research* que, como método de pesquisa orientado à solução de problemas, busca, a partir do entendimento do problema, construir um artefato (modelo) que permite transformar situações, alterando suas condições para condições melhores ou desejáveis (DRESCH; LACERDA; V. JÚNIOR, 2015).

1.7 ESTRUTURA DO TRABALHO

Essa dissertação foi dividida em cinco capítulos, incluindo este primeiro, o qual apresenta os aspectos gerais do tema trabalhado: contextualização, o problema, objetivos da pesquisa, justificativas do trabalho a ser desenvolvido, a delimitação do trabalho, o método de pesquisa utilizado e as etapas que foram executadas.

No segundo é apresentado a sistemática para a seleção dos artigos científicos mais relevantes para o trabalho.

No terceiro capítulo são apresentadas as teorias e publicações sobre o tema, fundamentação teórica e trabalhos correlatos.

No quarto capítulo é realizado uma análise descritiva da regra de negociação estudada neste trabalho. Contudo, é apresentado, além da análise descritiva, as concepções que constituem a regra de negociação quanto aos métodos que auxiliarão na aferição da efetividade da mesma. Neste capítulo são detalhados alguns valores obtidos a partir da regra de negociação e relatado as nuances que ofereçam o suporte necessário para o processo de tomada de decisão, visando mostrar os padrões do *setup*. Por fim, é abordado a metodologia para o treinamento e todo o processo de construção da arquitetura da rede neural artificial profunda que mediu a acurácia da regra de negociação.

No quinto capítulo, é elucidado os resultados obtidos a partir do trabalho realizado e as conclusões gerais discutidas a partir deles e fez sugestões para trabalhos futuros de modo a fomentar a continuidade da linha de pesquisa estudada.

2 MAPEAMENTO BIBLIOMÉTRICO

A execução da revisão da literatura, de certa forma, torna-se um elemento indispensável quando se inicia um planejamento de pesquisa. Todavia, o mapeamento e a identificação de artigos científicos relevantes através das análises tradicionais, muitas das vezes, carecem de uma metodologia na procura do referencial bibliográfico. Podendo, por assim dizer, guiar o pesquisador para algumas avaliações tendenciosas. Isto, por outro lado, compromete o devido entendimento da realidade do tema estudado (TRANFIELD; DENYER; SMART, 2003).

Devido ao aumento da quantidade cada vez maior de materiais científicos, aguçado pelo surgimento de novas tecnologias de informação e de comunicação, cada vez mais há a preocupação por parte dos pesquisadores, assim sendo, de lidarem com este problema que não é atual. Esta preocupação pode ser verificada, por exemplo, num estudo elaborado na primeira metade do século XX por Bush (1945), que mostra os desafios, já existentes nesta época, do crescimento da publicação de literaturas científicas.

Partindo desta premissa, são adotadas metodologias que visam auxiliar os pesquisadores a enfrentarem esse desafio. Desafio este, mitigado quando aplicado os estudos bibliométricos. Estes estudos partem da premissa da utilização de ferramentas científicas e elementos técnicos para aferir os aspectos da produção científica de uma área da ciência, fundamentada em instrumentos estatísticos que possibilitam uma análise mais acertada das produções científicas, conforme abordado em literaturas que discorrem sobre indicadores bibliométricos (TODESCHINI; BACCINI, 2016), combinação de mapeamento científico e análise de citações (NOYONS; MOED; LUWEL, 1999), mensuração de pesquisas acadêmicas (CRAIG, 2010), mecanismos utilizados para medir o registro da informação e a difusão do conhecimento (VANTI, 2002) e indicadores cienciométricos (SPINAK, 1998).

Esses métodos de estudo permitem, dentre outros pontos: distinguir os temas despontantes e redes de colaboração; examinar o desenvolvimento da produção científica e o seu nível de amadurecimento; estabelecer as instituições que centralizam a maior quantidade de publicações, com redes ou grupos de pesquisa mais atuante; condensar os títulos dos periódicos mais usados para a divulgação; verificar quais são os autores que produzem mais material científico, e demonstrar os autores mais citados, o que pode acarretar o quanto são relevantes para o tema (CRAIG (2010) e TODESCHINI e BACCINI (2016)).

2.1 PROCEDIMENTO ADOTADO

Um campo de pesquisa cujo interesse da comunidade acadêmica cresceu nos últimos anos é a Rede Neural Artificial e Análise Técnica. Chen e He (2018) descreve:

Atualmente, o principal método para prever o movimento do preço das ações pode

ser concluído em 2 aspectos: usando técnicas de Machine Learning ou de Redes Neurais Artificiais Profunda.¹

No tocante a Rede Neural artificial, esse interesse tem sido estimulado pelas frequentes discussões sobre o avanço da produção de dados pelas empresas e pelo surgimento de novas tecnologias. No entanto, o interesse pela Análise Técnica é proveniente ao acréscimo de investidores individuais em operações *daytrade* na bolsa de valores. Nesse sentido, diante da atenção que esses temas têm recebido da comunidade acadêmica, dos desafios existentes para se mapear e identificar literaturas relevantes, e a importância das técnicas bibliométricas para minimizar esse hiato, esse capítulo tem o objetivo de analisar e identificar aspectos relevantes da produção científica nacional e internacional sobre os respectivos temas.

Para analisar e identificar aspectos relevantes da produção científica nacional e internacional sobre Redes Neurais Artificiais e da Análise Técnica, este estudo adotou técnicas bibliométricas conforme recomendações de Blagus, Leskosek e Stare (2015), Cobo et al. (2012), Noyons, Moed e Luwel (1999) e Quevedo-Silva et al. (2016).

Um dos principais fundamentos para estudos bibliométricos são os dados das plataformas bibliográficas. Numa pesquisa, Granda-Orive et al. (2013) afirmam, e de certa forma, apontam as vantagens da utilização de indicadores bibliométricos da plataforma bibliográfica Scopus em relação aos da *Web of Science*. Dessa maneira, foi utilizado tão-somente para este estudo, a coleção bibliográfica da Scopus, vista como uma das maiores bases de dados de citações e resumo de literatura científica revisada por pares (ELSEVIER, 2019).

Para compreender como o campo de pesquisa Rede Neural Artificial e Análise Técnica têm se desenvolvido na produção científica, foi efetuada uma pesquisa na base Scopus utilizando palavras-chave e filtros. Tais palavras-chave e filtros estão dispostos abaixo:

- **Rede Neural Artificial:** TITLE-ABS-KEY (“artificial neural network” AND “binary classification”) AND (LIMIT-TO (DOCTYPE , “ar”)) AND (LIMIT-TO (LANGUAGE , “English”)) AND (LIMIT-TO (EXACTKEYWORD , “Artificial Neural Network”)) AND (LIMIT-TO (EXACTKEYWORD , “Neural Networks (Computer)”) OR LIMIT-TO (EXACTKEYWORD , “Algorithm”) OR LIMIT-TO (EXACTKEYWORD , “Algorithms”) OR LIMIT-TO (EXACTKEYWORD , “Binary Classification”) OR LIMIT-TO (EXACTKEYWORD , “Machine Learning”) OR LIMIT-TO (EXACTKEYWORD, “Neural Networks”) OR LIMIT-TO (EXACTKEYWORD , “Prediction”) OR LIMIT-TO (EXACTKEYWORD , “Support Vector Machine”) OR LIMIT-TO (EXACTKEYWORD , “Accuracy”) OR LIMIT-TO (EXACTKEYWORD , “Classification Algorithm”) OR LIMIT-TO (EXACTKEYWORD , “Binary Classification Problems”) OR LIMIT-TO (EXACTKEYWORD , “Decision Tree”) OR LIMIT-TO (EXACTKEYWORD , “Learning

¹ Tradução livre realizada pelo autor.

Algorithms") OR LIMIT-TO (EXACTKEYWORD , "Data Mining") OR LIMIT-TO (EXACTKEYWORD , "Decision Trees") OR LIMIT-TO (EXACTKEYWORD , "Artificial Neural Networks") OR LIMIT-TO (EXACTKEYWORD , "Data Analysis") OR LIMIT-TO (EXACTKEYWORD , "Measurement Accuracy") OR LIMIT-TO (EXACTKEYWORD , "ROC Curve") OR LIMIT-TO (EXACTKEYWORD , "Deep Neural Networks") OR LIMIT-TO (EXACTKEYWORD , "Supervised Learning") OR LIMIT-TO (EXACTKEYWORD , "Support Vector Machines (SVMs)") OR LIMIT-TO (EXACTKEYWORD , "K Nearest Neighbor") OR LIMIT-TO (EXACTKEYWORD , "Multilayer Neural Networks") OR LIMIT-TO (EXACTKEYWORD , "Network Learning") OR LIMIT-TO (EXACTKEYWORD , "Neural Network")).

- **Análise Técnica:** TITLE-ABS-KEY (("technical analysis" AND ("stock market" OR "index"))) AND (LIMIT-TO (DOCTYPE , "ar")) AND (LIMIT-TO (LANGUAGE , "English")) AND (LIMIT-TO (EXACTKEYWORD , "Technical Analysis") OR LIMIT-TO (EXACTKEYWORD , "Stock Market") OR LIMIT-TO (EXACTKEYWORD , "Forecasting") OR LIMIT-TO (EXACTKEYWORD , "Neural Networks") OR LIMIT-TO (EXACTKEYWORD , "Data Mining") OR LIMIT-TO (EXACTKEYWORD , "Moving Average") OR LIMIT-TO (EXACTKEYWORD , "Technical Indicator") OR LIMIT-TO (EXACTKEYWORD , "Stock Markets") OR LIMIT-TO (EXACTKEYWORD , "Algorithms") OR LIMIT-TO (EXACTKEYWORD , "Trading Rules") OR LIMIT-TO (EXACTKEYWORD , "Artificial Intelligence") OR LIMIT-TO (EXACTKEYWORD , "Trading Strategies") OR LIMIT-TO (EXACTKEYWORD , "Machine Learning") OR LIMIT-TO (EXACTKEYWORD , "Moving Averages") OR LIMIT-TO (EXACTKEYWORD , "Relative Strength Index") OR LIMIT-TO (EXACTKEYWORD , "Stock Trading") OR LIMIT-TO (EXACTKEYWORD , "Support Vector Machines") OR LIMIT-TO (EXACTKEYWORD , "Technical Indicators") OR LIMIT-TO (EXACTKEYWORD , "Stock Price Prediction") OR LIMIT-TO (EXACTKEYWORD , "Trading Strategy") OR LIMIT-TO (EXACTKEYWORD , "Artificial Neural Networks") OR LIMIT-TO (EXACTKEYWORD , "Learning Systems") OR LIMIT-TO (EXACTKEYWORD , "Neural Network") OR LIMIT-TO (EXACTKEYWORD , "Stock Price Movements") OR LIMIT-TO (EXACTKEYWORD , "Technical Trading") OR LIMIT-TO (EXACTKEYWORD , "Deep Learning") OR LIMIT-TO (EXACTKEYWORD , "Artificial Neural Network") OR LIMIT-TO (EXACTKEYWORD , "Classification (of Information)") OR LIMIT-TO (EXACTKEYWORD , "MACD") OR LIMIT-TO (EXACTKEYWORD , "RSI") OR LIMIT-TO (EXACTKEYWORD , "Regression Analysis") OR LIMIT-TO (EXACTKEYWORD , "Stochastic Systems") OR LIMIT-TO (EXACTKEYWORD , "Stock Price") OR LIMIT-TO (EXACTKEYWORD , "Technical Analysis Indicators") OR LIMIT-TO (EXACTKEYWORD , "Trade") OR LIMIT-TO (EXACTKEYWORD , "Trading Indicators") OR LIMIT-TO (EXACTKEYWORD , "Clustering Algorithms")

OR LIMIT-TO (EXACTKEYWORD , “Exponential Moving Average”)).

Como resultado, para o assunto Redes Neurais Artificiais e seus respectivos filtros foram contabilizados 96 artigos do período de 1990 a 2019. Em contrapartida, para o tema Análise Técnica e seus respectivos filtros de busca, foram contabilizados 280 artigos do período de 1990 a 2019. No entanto, devido a grande quantidade de artigos encontrados no período de 1990 a 2019, o estudo do tema 'Análise Técnica' através do Mapeamento Bibliométrico se ateve nos períodos de 2006 a 2019.

As informações extraídas para análise do mapeamento bibliométrico são: título do artigo, autores, palavras-chave, resumo, revista onde foi publicado, ano da publicação, instituição a que pertencem os autores, quantidade de citações e *h-index*, que é um indicador para quantificar a produtividade e o impacto científico, a partir da quantidade de citações e publicações. Por exemplo, o *h-index* será 9, se os 9 artigos mais citados de um autor, tema, revista, entre outros, tiverem no mínimo 9 citações cada um (HIRSCH, J. E., 2005).

Contudo, a partir da base bibliográfica da Scopus, os dados obtidos foram transferidos para serem processados pela ferramenta de elaboração de mapas bibliométricos *Science Mapping Analysis software Tool (SciMAT)*, e para a confecção de gráficos e tabelas com o uso do *software* Microsoft Excel. Todo esse processo foi realizado para se analisar o desempenho bibliométrico e a progresso conceitual dessas áreas de pesquisa através do mapeamento bibliométrico.

Enquanto a análise do desempenho bibliométrica visa mensurar o impacto da citação da produção científica de diferentes atores, o mapeamento bibliométrico destina-se a demonstrar a estrutura conceitual, social ou intelectual da pesquisa científica, bem como a sua evolução e aspectos dinâmicos (GUTIÉRREZ-SALCEDO et al., 2018), por uma representação espacial de como temas, especialidades ou autores, por exemplo, estão relacionados entre si.

Para melhorar a qualidade dos dados, como o agrupamento de palavras que representam o mesmo conceito, ou termos redigidos, por exemplo, no singular e no plural, os dados foram tratados com um recurso do *SciMAT*, específico para esse fim.

Em seguida, também com o uso do *SciMAT*, foram definidos os períodos para demonstrar a evolução conceitual da Rede Neural Artificial e da Análise Técnica pela análise de mapeamento bibliométrico. Assim, para o tema Rede Neural Artificial, o período total (1990-2019) foi subdividido em três, um entre 1990 e 2000, um segundo em 2001 e 2010 e outro entre 2011 e 2019. Para o tema Análise Técnica, o período total (2006 - 2019) foi subdividido em três também, um entre 2006 e 2010, um segundo em 2011 e 2015 e outro entre 2016 e 2019. Cobo et al. (2012) afirma que a subdivisão dos períodos fica a critério do pesquisador.

A proposta inicial era escolher períodos menores para Redes Neurais Artificiais, por exemplo, de 4 anos, semelhante à divisão realizada para Análise Técnica. No entanto,

verificou-se que com períodos deste tamanho não eram gerados dados suficientes para obter os resultados adequados para a análise de mapeamento bibliométrico.

Com o auxílio do *SciMAT*, a etapa da análise da evolução conceitual elaborou-se um mapeamento científico sob uma estrutura longitudinal que utiliza técnicas como a combinação da análise de coocorrência de palavras, cocitação de autor ou de periódico, com o indicador *h-index*². Essa análise foi elaborada pelas seguintes etapas (COBO et al., 2012):

- **Identificação dos temas abordados:** essa identificação ocorreu a partir da análise de coocorrência das palavras que compõe as 'palavras-chaves' dos artigos, dentro de cada um dos períodos definidos. O período definido para o tema 'Redes Neurais Artificiais' foi '1990 - 2000', '2001 - 2010' e '2011 - 2019'. Para o tema 'Análise Técnica' foi o período de '2006 - 2010', '2011 - 2015' e '2016 - 2019'. Um conjunto de palavras-chave fortemente relacionado entre si, representará um tema;
- **Visualização de temas de pesquisa e redes temáticas:** nesta etapa, os temas identificados são visualizados através de um diagrama estratégico e de redes temáticas. O diagrama estratégico demonstra os temas identificados num determinado período em um espaço bidimensional, caracterizando-os de acordo com as medidas de densidade e centralidade de Callon, Courtial e Laville (1991). A centralidade mede o grau de interação de uma rede com outras redes, e a densidade mede a força interna da rede. De acordo com a centralidade e a densidade, é possível representar um tema de pesquisa em um diagrama de dois eixos. Esse diagrama se divide em quadrantes, conforme demonstrado na Figura 3, sendo que a representação de cada um deles é a seguinte: (i) temas motores (alta densidade e alta centralidade) – os temas localizados neste quadrante se identificam como aqueles muito desenvolvidos ou essenciais para construir a área de investigação; (ii) temas básicos e transversais (baixa densidade e alta centralidade) - nesse quadrante se incluem os temas importantes para o campo científico, porém pouco desenvolvidos. São temas transversais básicos e genéricos; (iii) temas emergentes ou decadentes (baixa densidade e baixa centralidade): os temas que se situam nesse quadrante são pouco desenvolvidos ou estão desaparecendo, e (iv) temas altamente desenvolvidos e isolados (alta densidade e baixa centralidade): neste quadrante estão os temas muito especializados, porém de importância marginal para o campo científico, de escassa relevância.

² Índice para quantificar a produtividade e o impacto de cientistas baseando-se nos seus artigos mais citados.

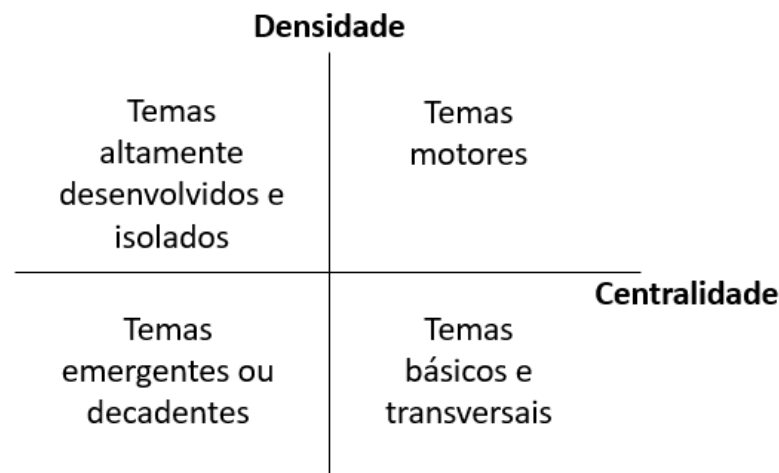


Figura 3 – Diagrama estratégico do Mapeamento Bibliométrico.

Fonte: Elaborado pelo autor (2019).

Nos diagramas estratégicos são incorporadas figuras esféricas cujos volumes podem representar diferentes indicadores bibliométricos, como a quantidade de citações recebidas ou o *h-index* do tema.

2.2 ANÁLISE DO MAPEAMENTO BIBLIOMÉTRICO

Para analisar os artigos científicos mais destacados de Redes Neurais Artificiais e Análise Técnica, foram apresentados os diagramas estratégicos de cada período selecionado, elaborados pelo *SciMAT* com base nas medidas de densidade e centralidade de Callon, Courtial e Laville (1991). Os volumes das esferas que foram apresentadas nesses diagramas equivalem à quantidade de citações recebidas pelos temas que elas representam.

2.2.1 Mapeamento Bibliométrico para Redes Neurais Artificiais

Visando analisar a evolução dos temas e sua respectivas relevâncias, conforme citado por Cobo et al. (2012), foi adotado para a seção 'Redes Neurais Artificiais' a divisão dos temas em três períodos de tempo: '1990 - 2000', '2001 - 2010' e '2011 - 2019'.

- Primeiro Período (1990 - 2000)

De acordo com o diagrama estratégico apresentado na Figura 4, observou-se que a produção científica se concentra no tema: *Artificial Neural Network* (redes neurais artificiais). Por fazer parte do quadrante dos temas motores, que são aqueles muito desenvolvidos ou essenciais para construir a área de investigação, o tema em questão se destaca como o mais importante desse período dentre os temas analisados.

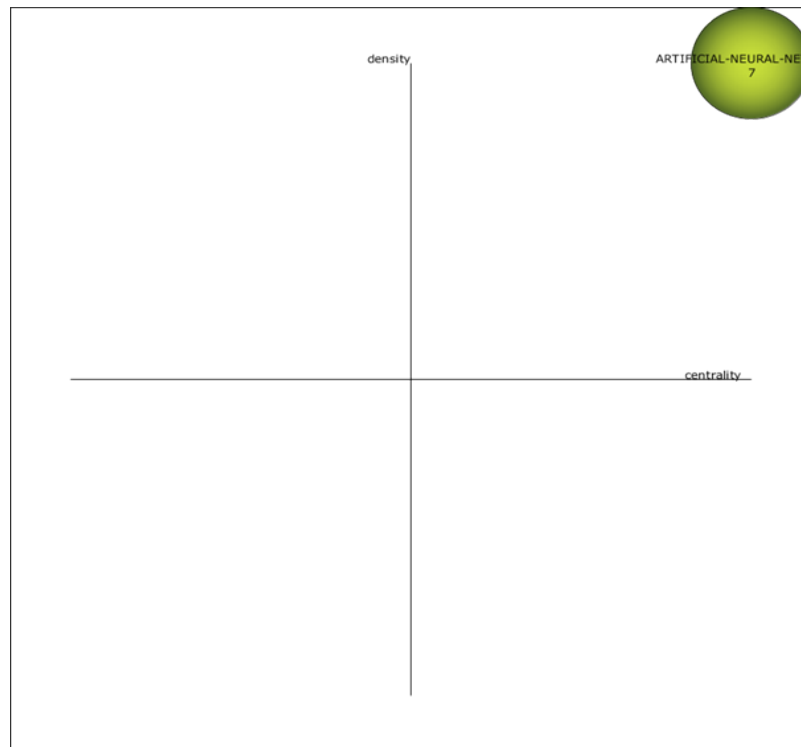


Figura 4 – Diagrama estratégico do período de 1990 a 2000.

Fonte: SciMAT (2019).

Na Tabela 1, são apresentados os dados sobre a quantidade de publicações dos temas que se sobressaíram neste período, as citações obtidas por essas publicações e os valores *h-index*. Observa-se que o principal tema, Artificial Neural Network (redes neurais artificiais), atingiu quantidades significantes de citações.

Tabela 1 – Desempenho dos temas nos períodos de 1990 a 2000.

Tema	Publicações	Citações	<i>H-index</i>
<i>Artificial Neural Network</i> (redes neurais artificiais)	7	82	5

Fonte: Elaborado pelo Autor (2019).

- Segundo Período (2001 - 2010)

De acordo com o diagrama estratégico apresentado na Figura 5, observou-se que a produção científica sobre *Artificial Neural Network* (redes neurais artificiais) aumentou com o passar dos anos. Por fazer parte do quadrante dos temas motores, que são aqueles muito desenvolvidos ou essenciais para construir a área de investigação, o tema em questão se destaca como o mais importante desse período dentre os temas analisados. Todavia, o tema *Binary Classification* (classificação binária) começa a possuir relevância nos artigos científicos.

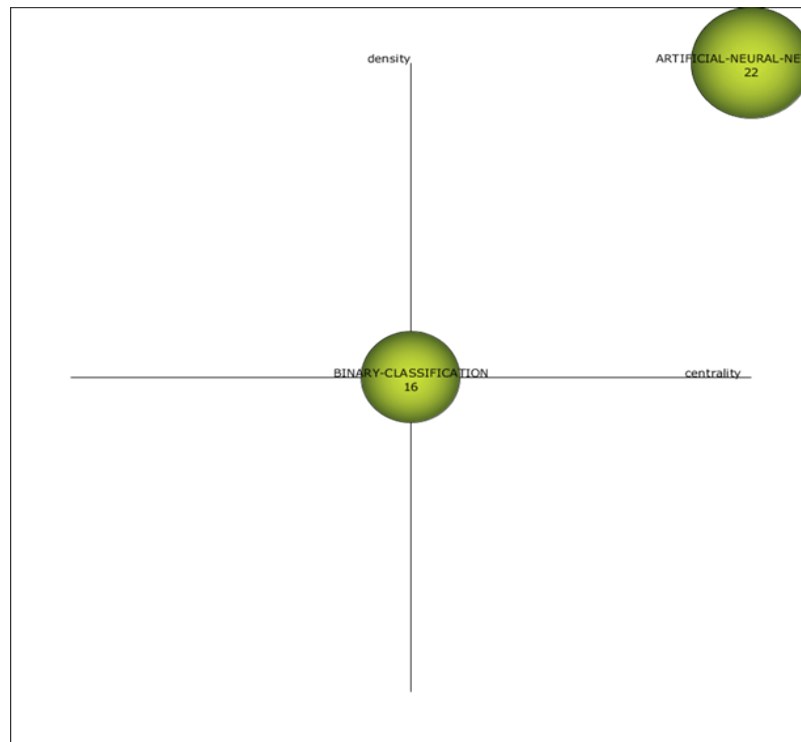


Figura 5 – Diagrama estratégico do período de 2001 a 2010.

Fonte: SciMAT (2019).

Na Tabela 2, são apresentados os dados sobre a quantidade de publicações dos temas que se sobressaíram neste período, as citações obtidas por essas publicações e os valores *h-index*. Observa-se que o principal tema, *Artificial Neural Network* (redes neurais artificiais), obteve mais citações neste período do que o anterior. Por outro lado, o tema *Binary Classification* (classificação binária) é o segundo com maior significância entre as publicações científicas. De certa forma, apresentando um número elevado de citações, conclui-se a ocorrência de uma eclosão de pesquisas sobre o tema.

Tabela 2 – Desempenho dos temas nos períodos de 2001 a 2010.

Temas	Publicações	Citações	<i>H-index</i>
<i>Artificial Neural Network</i> (redes neurais artificiais)	22	697	13

Temas	Publicações	Citações	<i>H-index</i>
<i>Binary Classification</i> (classificação binária)	16	457	10

Fonte: Elaborado pelo Autor (2019).

- Terceiro Período (2011 - 2019)

De acordo com o diagrama estratégico apresentado na Figura 6, observou-se que a produção científica se concentra em três temas: *Artificial Neural Network* (redes neurais artificiais), *Binary Classification* (classificação binária) e *Machine Learning* (aprendizado de máquina). Por fazer parte do quadrante dos temas motores, que são aqueles muito desenvolvidos ou essenciais para construir a área de investigação, *Artificial Neural Network* (redes neurais artificiais) e o tema *Machine Learning* (aprendizado de máquina) se destacam como os mais importantes desse período dentre os temas analisados. Por estar no quadrante emergente, considerados pouco desenvolvidos ou estão desaparecendo, o tema *Binary Classification* (classificação binária) também se apresenta como relevante.

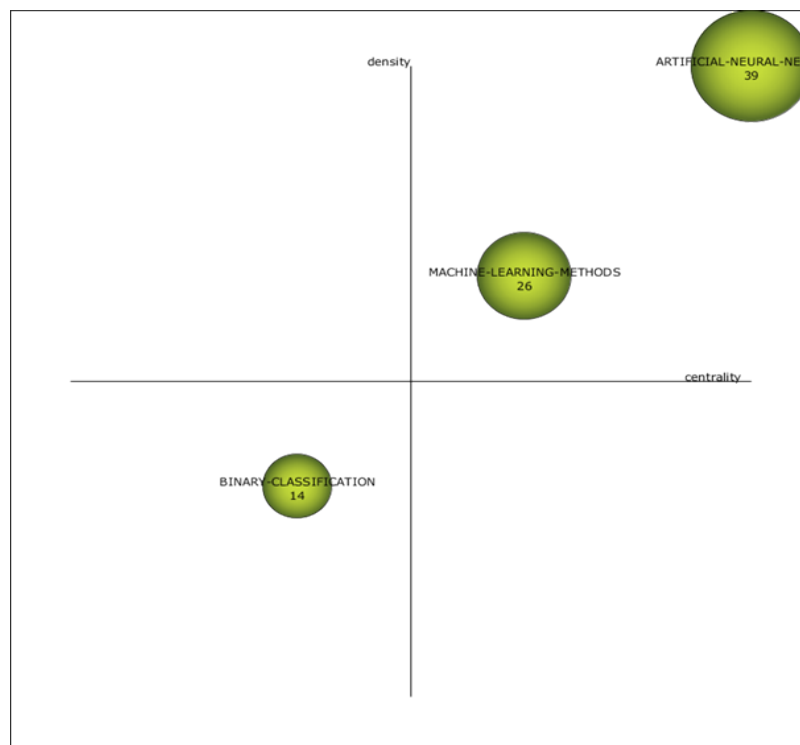


Figura 6 – Diagrama estratégico do período de 2011 a 2019.

Fonte: Elaborado pelo autor (2019).

Na Tabela 3, são apresentados os dados sobre a quantidade de publicações dos temas que se sobressaíram neste período, as citações obtidas por essas publicações e os valores *h-index*.

Observa-se que os principais temas, *Artificial Neural Network* (redes neurais artificiais) e *Machine Learning* (aprendizado de máquina), obtiveram mais citações neste período do que o anterior. Por outro lado, o tema *Binary Classification* (classificação binária) mudou de panorama.

Tabela 3 – Desempenho dos temas nos períodos de 2010 a 2019.

Tema	Publicações	Citações	H-index
Artificial Neural Network (<i>redes neurais artificiais</i>)	39	2.108	17
<i>Machine Learning</i> (aprendizado de máquina)	26	797	13
<i>Binary Classification</i> (classificação binária)	14	366	9

Fonte: Elaborado pelo Autor (2019).

2.2.2 Mapeamento Bibliométrico da Análise Técnica

Visando analisar a evolução dos temas e sua respectivas relevâncias, conforme citado por Cobo et al. (2012), foi adotado para a seção 'Análise Técnica' a divisão dos temas em três períodos de tempo: '2006 - 2010', '2011 – 2015' e '2016 - 2019'. Por ser um tema mais abrangente, em relação aa seção Redes Neurais Artificiais, julgou a necessidade numa maior divisão dos períodos, visando filtrar mais acertado os referidos temas.

- Primeiro Período (2006 - 2010)

De acordo com o diagrama estratégico apresentado na Figura 7, observou-se que a produção científica se concentra em três temas: *Stock* (ações), *Technical Analysis* (análise técnica) e *Statistical Analysis* (análise estatística). Por fazer parte do quadrante dos temas motores, que são aqueles muito desenvolvidos ou essenciais para construir a área de investigação, *Stock* (ações) e *Technical Analysis* (análise técnica) se destacam como os mais importantes desse período dentre os temas analisados. Por estar no quadrante emergente ou decadentes, considerados pouco desenvolvidos ou estão desaparecendo, o tema *Statistical Analysis* (análise estatística) também se apresenta como relevante.

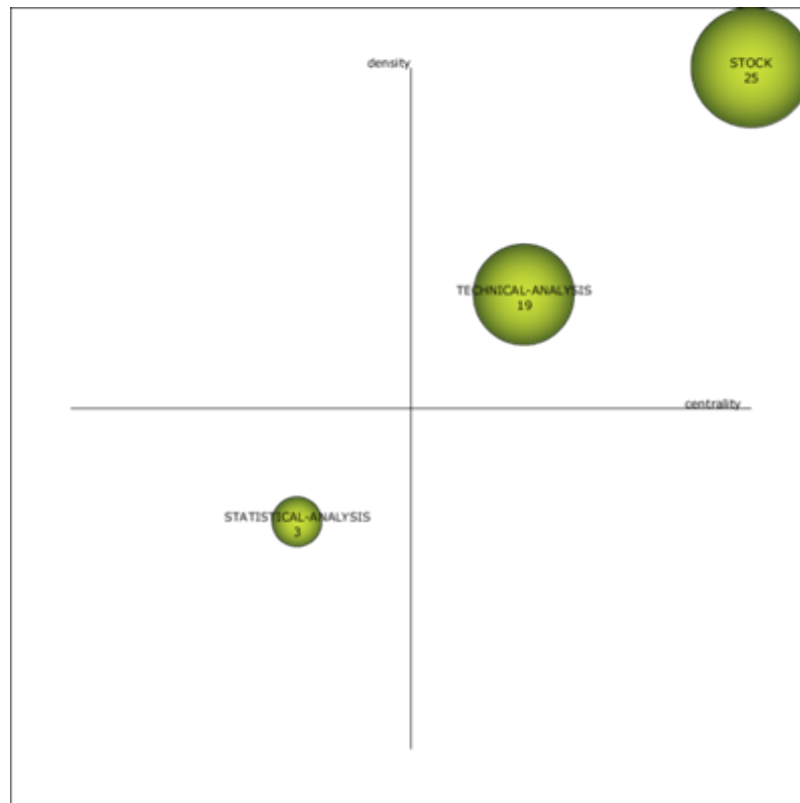


Figura 7 – Diagrama estratégico do período de 2006 a 2010.

Fonte: SciMAT (2019).

Na Tabela 4, são apresentados os dados sobre a quantidade de publicações dos temas que se sobressaíram neste período, as citações obtidas por essas publicações e os valores *h-index*. Observa-se que os principais temas, *Stock* (ações) e *Technical Analysis* (análise técnica), obtiveram a maior quantidade de citações. Por outro lado, o tema *Statistical Analysis* (análise estatística) necessita ser avaliado para que se comprove se o mesmo requer mais desenvolvimento científico ou está obsoleto.

Tabela 4 – Desempenho dos temas nos períodos de 2006 a 2010.

Tema	Publicações	Citações	<i>H-Index</i>
<i>Stock</i> (ações)	19	444	9
<i>Technical Analysis</i> (análise técnica)	25	577	13

Tema	Publicações	Citações	H-Index
<i>Statistical Analysis</i> (análise estatística)	3	49	3

Fonte: Elaborado pelo autor (2019).

- Segundo Período (2011 - 2015)

De acordo com o diagrama estratégico apresentado na Figura 8, observou-se que a produção científica se concentra em quatro temas: *Stock* (ações), *Technical Analysis* (análise técnica), *Forecasting* (previsão) e *Investiment* (investimento). Por fazer parte do quadrante dos temas motores, que são aqueles muito desenvolvidos ou essenciais para construir a área de investigação, *Stock* (ações) se destaca como o mais importante desse período dentre os temas analisados. Por estar no quadrante dos básicos e transversais, considerados como importantes para o campo científico, apesar de ser pouco desenvolvidos, o tema *Technical Analysis* (análise técnica) também se apresenta como relevante. Por outro lado, o tema *Investiment* (investimento), por estar no quadrante de temas altamente desenvolvidos e isolados, não apresenta relevância para o estudo.

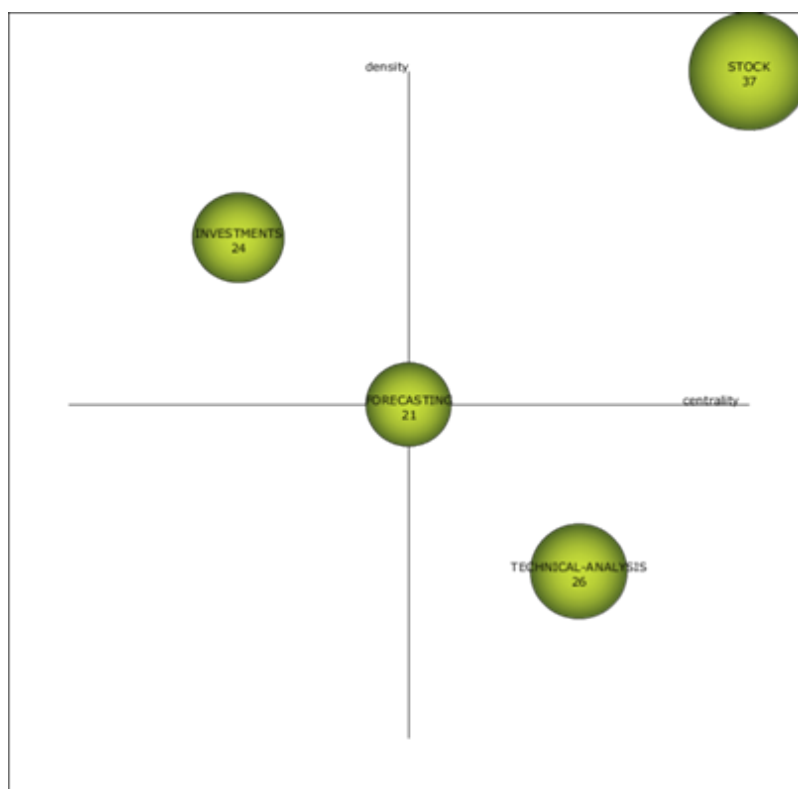


Figura 8 – Diagrama estratégico do período de 2011 a 2015.

Fonte: SciMAT (2019).

Na Tabela 5, são apresentados os dados sobre a quantidade de publicações dos temas que se sobressaíram neste período, as citações obtidas por essas publicações e os valores *h-index*. Observa-se que os principais temas, *Stock* (ações) e *Technical Analysis*

(análise técnica), obtiveram a maior quantidade de citações. Por outro lado, o tema *Statistical Analysis* (análise estatística) necessita ser avaliado para que se comprove se o mesmo requer mais desenvolvimento científico ou está obsoleto.

Tabela 5 – Desempenho dos temas nos períodos de 2011 a 2015.

Tema	Publicações	Citações	<i>H-Index</i>
<i>Technical Analysis</i> (análise técnica)	26	338	11
<i>Stock</i> (ações)	37	527	14
<i>Investiment</i> (investimento)	24	268	8
<i>Forecasting</i> (previsão)	21	212	8

Fonte: Elaborado pelo autor (2019).

- Terceiro Período (2016 - 2019)

De acordo com o diagrama estratégico apresentado na Figura 9, observou-se que a produção científica se concentra em três temas: *Stock* (ações), *Technical Analysis* (análise técnica) e *Financial Market* (mercado financeiro). Por fazer parte do quadrante dos temas motores, que são aqueles muito desenvolvidos ou essenciais para construir a área de investigação, *Stock* (ações) se destaca como o mais importante desse período dentre os temas analisados. Por estar no quadrante dos básicos e transversais, considerados como importantes para o campo científico, apesar de ser pouco desenvolvidos, o tema *Technical Analysis* (análise técnica) também se apresenta como relevante. Por outro lado, o tema *Genetic Algorithm* (algoritmo genético) necessita ser avaliado para que se comprove se o mesmo requer mais desenvolvimento científico ou está obsoleto.

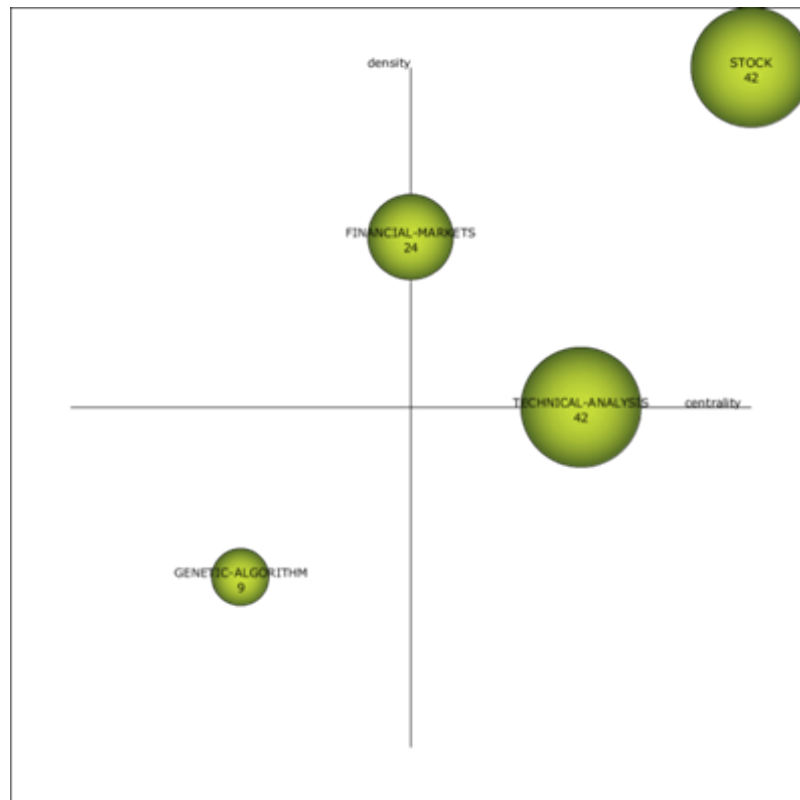


Figura 9 – Diagrama estratégico do período de 2016 a 2019.

Fonte: SciMAT (2019).

Na Tabela 6, são apresentados os dados sobre a quantidade de publicações dos temas que se sobressaíram neste período, as citações obtidas por essas publicações e os valores *h-index*. Observa-se que os principais temas, *Stock* (ações), *Technical Analysis* (análise técnica) e *Financial Market* (mercado financeiro), obtiveram a maior quantidade de citações. Por outro lado, o tema *Genetic Algorithm* (algoritmo genético) necessita ser avaliado para que se comprove se o mesmo requer mais desenvolvimento científico ou está obsoleto.

Tabela 6 – Desempenho dos temas nos períodos de 2016 a 2019.

Tema	Publicações	Citações	<i>H-Index</i>
<i>Technical Analysis</i> (análise técnica)	42	157	6
<i>Financial Market</i> (mercado financeiro)	24	172	7

Tema	Publicações	Citações	H-Index
<i>Stock</i> (ações)	42	177	6
<i>GeneticAlgorithm</i> (algoritmo genético)	9	58	3

Fonte: Elaborado pelo autor (2019).

2.3 Ponderação Sobre a Relevância dos Temas para a Pesquisa Científica

A escolha de um tema ou área visando a realização de um trabalho científico é um dos principais obstáculos para o pesquisador. Interrogações sobre a importância da pesquisa, e logo, sua relevância para o âmbito acadêmico fez-se da revisão de literatura o ponto central na organização da pesquisa acadêmica.

Não obstante, as Seções 2.2.1 e 2.2.2 procuraram corroborar, e de certa forma enfatizar, aspectos relevantes da produção científica nacional e internacional sobre 'Redes Neurais Artificiais' e 'Análise Técnica'. Ambos os temas eminentes na concepção deste trabalho.

Entre os anos de 1990 a 2019, período escolhido para analisar Redes Neurais Artificiais, foram contabilizados 124 publicações diretamente ligados ao tema. Referente à Análise Técnica, entre o período de 2006 a 2019, foram contabilizados 271 artigos publicados. Para ambos os temas, verificou-se uma crescente produção de artigos científicos nos últimos períodos conforme explanado nas Tabelas 3 e 6. Tal evolução, permitiu a evidenciação, de certa forma, da elevação do grau de maturação desses campos de pesquisa no decorrer dos anos. Fato este corroborado quando observado nos diagramas estratégicos, obtidos pelo *SciMAT*.

Embora o mapeamento bibliométricos dos temas 'Redes Neurais' e 'Análise Técnica' estão sendo apresentados separadamente neste trabalho, foram utilizados palavras-chave que visavam encontrar artigos demonstrando uma interseção entre os mesmos. Porém, não foi encontrado nenhum material científico relevante pelo *SciMAT*.

2.3.1 Relevância do Tema Redes Neurais Artificiais

Pesquisas científicas utilizando algoritmos de Redes Neurais Artificiais como base operacional para soluções de problemas têm aumentado com o passar dos anos graças ao grande interesse da comunidade científica em diversas áreas como medicina, ciência de dados, biologia, etc.

A partir do mapeamento bibliométrico foi possível determinar que no período entre 1990 e 2000, as publicações realizadas sobre Redes Neurais Artificiais expressaram foco apenas em um tema: '*Artificial Neural Network*' (redes neurais artificiais).

No segundo Período, entre os anos de 2001 e 2010, o foco passou a ser em dois temas. Dentre eles, destaque para o surgimento e eclosão do tópico *Binary Classification*

(classificação binária), obtendo 457 citações em trabalhos científicos neste período. Contudo, ainda não ficou claro a relevância deste tema já que o mesmo aparece bem centralizado nos quadrantes de importância. O tópico '*Artificial Neural Network*' (redes neurais artificiais), seguindo a tendência do período de 1990 e 2000, continua entre os temas motores para a produção científica. Desta vez obtendo 697 citações em artigos científicos.

No terceiro período, entre os anos de 2011 e 2019, dois temas se destacaram como muito desenvolvido e essencial: *Artificial Neural Network* (redes neurais artificiais) e *Machine Learning* (aprendizado de máquina). Mais uma vez, seguindo a disposição dos períodos anteriores, a seção *Artificial Neural Network* (redes neurais artificiais) continua entre os tópicos motores para a produção científica. O interesse neste assunto, mais uma vez, aumentou consideravelmente em relação aos períodos anteriores. Obtendo 2108 citações no decorrer dos anos. Outro assunto que apresentou relevância entre a comunidade acadêmica foi *Machine Learning* (aprendizado de máquina). Dispondo de 797 citações. O tema *Binary Classification* (classificação binária) se apresenta como emergente ou decadente, pouco desenvolvido ou que está desaparecendo.

2.3.2 Relevância do Tema Análise Técnica

A Análise Técnica, por se tratar de uma método onde é empregado ferramentas didáticas e de fácil uso, está sendo cada vez mais utilizado por investidores individuais. De certa forma, este recurso possibilitou aos mesmos investir em ativos financeiros sem precisar possuir uma base acadêmica sobre economia e derivativos financeiros. Não diferente, artigos científicos sobre a Análise Técnica, com o passar dos anos, tem aumentado consideravelmente.

A partir do mapeamento bibliométrico foi possível determinar que no período entre 2006 e 2010, as publicações realizadas sobre Análise Técnica teve foco em três temas: *Stock* (ações), *Technical Analysis* (análise técnica) e *Statistical Analysis* (análise estatística). Neste período, *Stock* (ações), *Technical Analysis* (análise técnica) estavam no auge da pesquisa acadêmica. Cada uma, respectivamente, foram citadas 577 e 444 vezes em trabalhos científicos, sendo de certa forma, temas motores para pesquisas científicas. *Statistical Analysis* (análise estatística) se apresenta como emergente ou decadente, pouco desenvolvido ou que está desaparecendo.

No segundo Período, entre os anos de 2011 e 2015, o foco passou a ser em quatro assuntos: *Stock* (ações), *Technical Analysis* (análise técnica), *Investment* (investimento) e *Forecasting* (previsão). Dentre eles, destaque para a mudança de quadrante do tópico *Technical Analysis* (análise técnica). No período anterior, o mesmo estava entre os temas motores nas pesquisas científicas. Porém, neste período, a temática encontra-se no quadrante dos temas básicos e transversais. Nesse quadrante se incluem os temas importantes para o campo científico, ainda assim, pouco desenvolvidos. O tópico *Technical Analysis*

(análise técnica) teve 338 citações. No período entre 2006 e 2010, este número foi de 444. *Stock* (ações) continua entre os temas motores com 527 citações. O tema *Forecasting* (previsão) foi citado 212 vezes em artigos científicos. Contudo, ainda não ficou claro a relevância deste assunto já que o mesmo aparece bem centralizado nos quadrantes de importância. Já o tema *Investment* (investimento) é um tema altamente desenvolvido e isolado. Contudo, de importância marginal e restrita a uma área.

No terceiro período, entre os anos de 2016 e 2019, três tópicos se destacaram como muito desenvolvido e essencial: *Stock* (ações), *Technical Analysis* (análise técnica) e *Financial Market* (mercado financeiro). Mais uma vez, seguindo a disposição dos períodos anteriores, os assuntos *Stock* (ações) e *Technical Analysis* (análise técnica) continuam entre os temas motores para a produção científica. Outra temática que apresentou relevância entre a comunidade acadêmica foi *Financial Market* (mercado financeiro). Dispondo de 172 citações. O assunto *Genetic Algorithm* (algoritmo genético) se apresenta como emergente ou decadente, pouco desenvolvido ou que está desaparecendo.

3 FUNDAMENTAÇÃO TEÓRICA

Para este trabalho, os temas centrais motivadores para o estudo são a Análise Técnica e Redes Neurais Artificiais Profunda devido ao significativo grau de importância dos mesmos conforme demonstrado no Capítulo 2.

Pelo fato da regra de negociação manusear informações de indicadores técnicos; a primeira seção, no caso a Análise Técnica, foi o arcabouço de toda a pesquisa por propiciar os instrumentos fundamentais para a arrecadação de dados oriundos do *software* ou plataforma de negociação eletrônica. Em contrapartida, a Redes Neurais Artificiais Profunda resumir-se-á tal como dispositivo validador, provendo o ensaio preditivo das variáveis empregadas na regra de negociação. Além disso é descrito as particularidades do mini contrato futuro do dólar e os modelos probabilísticos empregados na análise.

A Figura 10 visa representar o roteiro da Fundamentação Teórica deste trabalho.

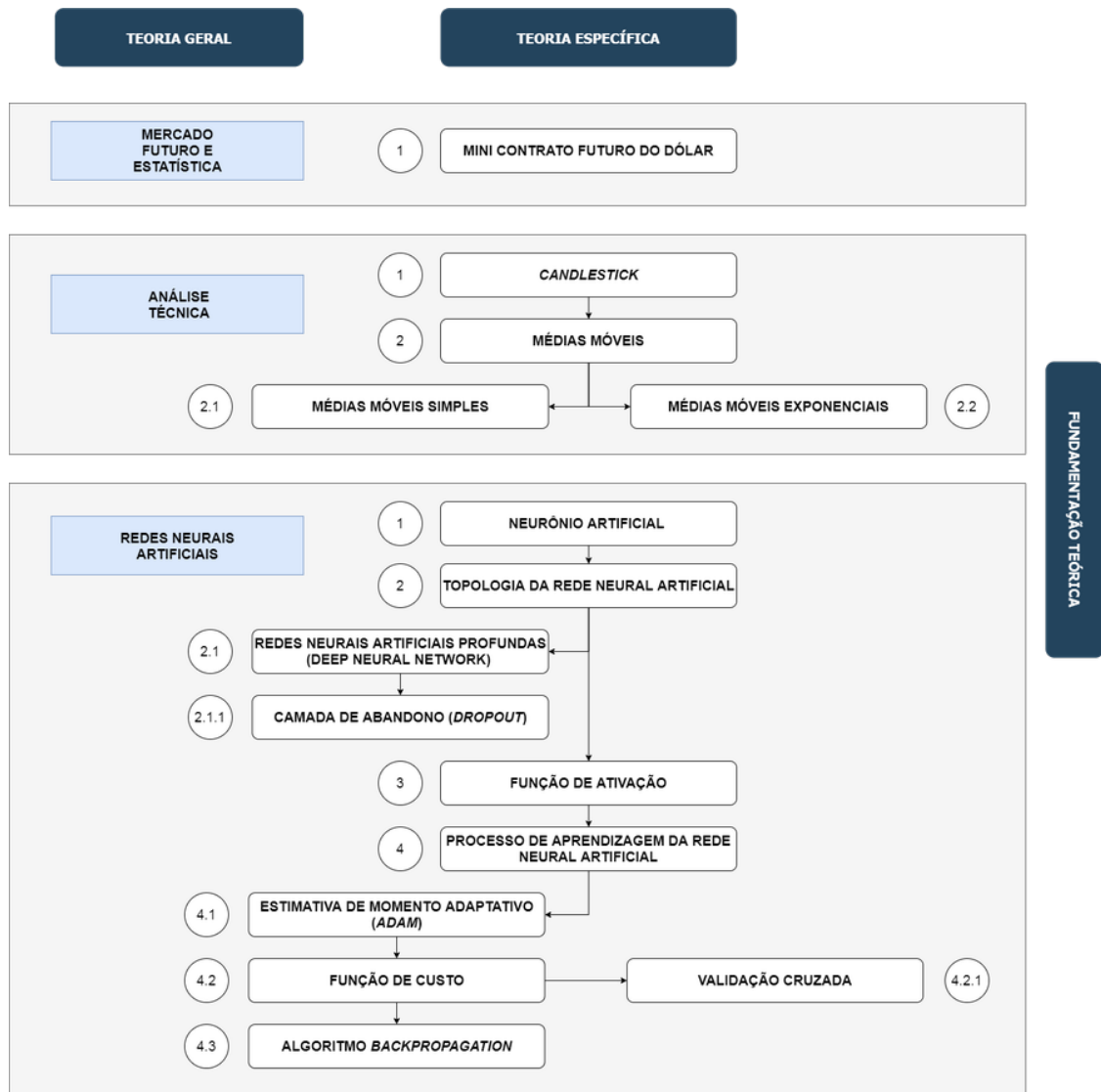


Figura 10 – Roteiro da Fundamentação Teórica.

Fonte: Elaborado pelo autor (2019).

3.1 MINI CONTRATO FUTURO DO DÓLAR

Negociações em derivativos financeiros têm aumentado cada vez mais entre investidores individuais (pessoa física ou *traders*) e instituições financeiras ao redor do mundo. Motivado pelo crescimento econômico nos últimos anos, este modelo de investimento tem possibilitado para estes investidores oportunidades de ganhos expressivamente lucrativas. Para Leung, Daouk e Chen (2000), pode-se afirmar que existem dois motivos cruciais para o aumento das negociações em derivativos da bolsa de valores. Segundo os autores, o progresso e a popularização da realização de *trading* neste modelo de instrumento, primeiramente, possibilitou ao investidor o acréscimo de mais uma ferramenta de proteção para sua carteira de investimento (*hedge*). Por fim, criou oportunidades de ganhos rápidos e de baixo custo para os especuladores do mercado financeiro (*day trading*).

No Brasil, acompanhando a tendência mundial, as negociações no segmento mercado futuro na bolsa de valores aumentaram com o passar do tempo. Um dos produtos que melhor representa esta gradativa evolução é o mini contrato futuro do dólar - mini dólar ou WDO. Derivativo financeiro da taxa de câmbio de reais por dólar comercial, o mini contrato futuro do dólar tornou-se um dos principais ativos para se negociar na bolsa de valores brasileira devida sua alta liquidez e baixo custo. Segundo B3¹, o Contrato Futuro de Dólar Comercial é o segundo contrato derivativo mais negociado no mercado brasileiro, perdendo apenas para o Contrato Futuro de DI. Por servir para proteção ou especulação sobre o preço da moeda em data futura, é bastante demandado pelo mercado.

Todavia, o mini dólar possibilita que os investidores negociem as expectativas futuras da moeda em relação ao real. Logo, este ativo financeiro enquadra-se numa categoria de negócio denominado mercado futuro. Distinto do mercado de ações, onde uma fatia da empresa é distribuída entre os acionistas, no mercado futuro são negociados contratos futuros onde a parte compradora e vendedora compromete-se a negociar um bem num determinado valor, e conseqüentemente, numa determinada data futura. Outra dissemelhança entre as ações de empresas em relação à contratos futuros negociados na B3, a cotação do contrato futuro é efetuada em pontos, diferentemente da ação, a qual é cotada em reais.

Partindo dessa premissa, o mini dólar proporcionou uma guinada entre o grupo de investidores. Entre os principais grupos de investidores que negociam este ativo, até agosto de 2019, 26% corresponde o investidor pessoa física; 60% correspondem ao investidor não residente no Brasil e 12% corresponde ao investidor institucional, segundo a B3. Dentre este grupo de investidores que aplicam e investem no mini dólar, destaca-se, conforme Figura 11, o aumento de investidores individuais ao passar dos anos. Em um mercado financeiro ocupado por grandes instituições financeiras, sendo estas, munidas com ferramentas de alta tecnologia, para muitos, não soa plausível que um indivíduo utilizando um *desktop* de sua casa consiga consistência e lucro no *day trading*. Mesmo assim, este perfil de investidor

¹ Bolsa de Valores brasileira.

vem crescendo ano após ano em virtude das baixas taxas e facilidades oferecidas pelas corretoras e pela aspiração em um retorno rápido em operações *day trade*.

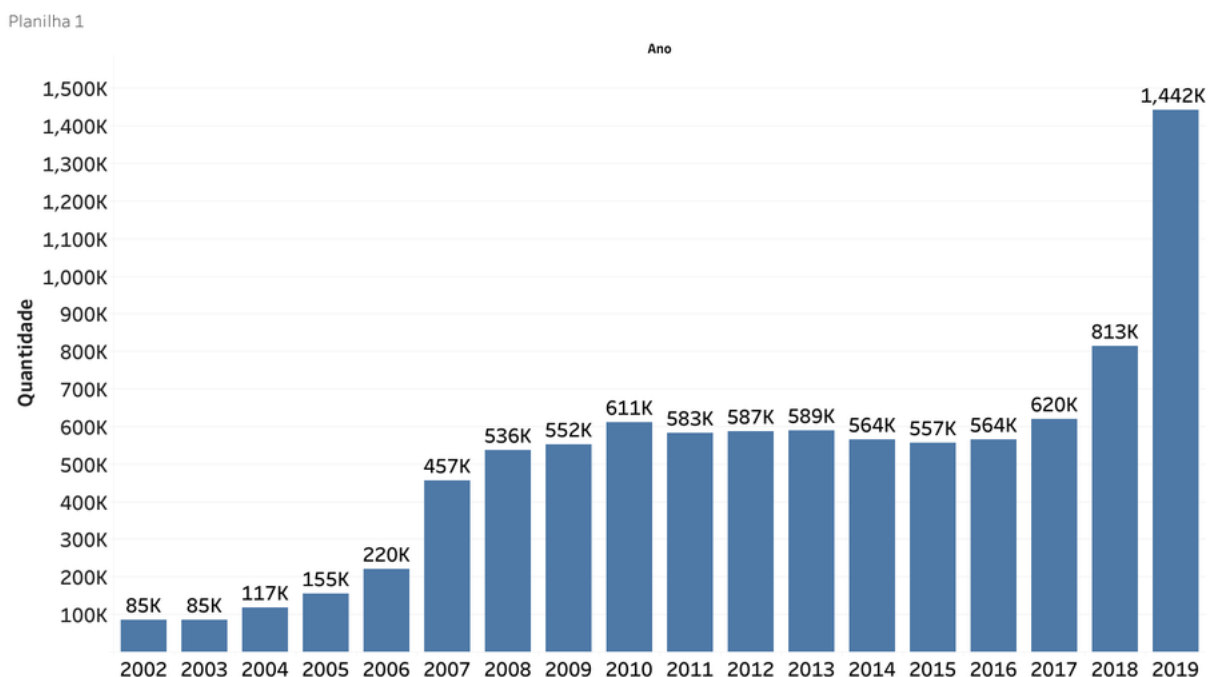


Figura 11 – Evolução no número de investidores pessoas físicas na B3.

Fonte: B3 (2019).

Não obstante, para que o investidor ou *trader* negocie contrato futuro é necessário dispor de uma margem de garantia. A margem de garantia é uma quantia depositada pelas partes envolvidas em um contrato futuro com o objetivo de garantir o cumprimento do mesmo. Para o mini dólar, a margem garantia é de 15%. Para uma melhor elucubração, um mini contrato futuro do dólar equivale a US\$ 5.000,00. Caso o dólar esteja precificado em relação ao real num patamar de R\$ 3,00, este mini contratado do dólar valerá R\$ 15.000,00. No entanto, a margem garantia possibilita que o *trader* não precise reembolsar este montante na conta da corretora de valores para que o mesmo execute operações de compra e venda na B3. Como a margem garantia no mini dólar exige um percentual de 15%, com o montante financeiro de R\$ 2.250,00, o *trader* pode negociar um ativo que custa R\$ 15.000,00. Logo, por assim dizer, por exigir inicialmente um baixo custo financeiro, culminou na viabilização da participação de investidores pessoa física.

O mini dólar possui características técnicas específicas, tais como a forma na qual é cotada, o tamanho do lote padrão (o número permissível de contratos que podem ser negociados por operação) e data de vencimento. Possuem um período de pré-abertura de negociação, no início do dia, o qual tem duração variável. Possuem também um horário de encerramento de negociação, o qual é diferenciado no dia de sua expiração, ou seja, as negociações podem encerrar-se em um horário diferente do usual no dia do seu término.

Neste trabalho, são considerados os horários de negociação entre os períodos de pré-abertura e encerramento, e a cotação do contrato futuro com data de vencimento mais próxima, porque esse é o contrato mais negociado. A Tabela 7 visa apresentar os aspectos técnicos referente ao mini dólar.

Tabela 7 – Características técnicas do mini contrato do dólar futuro negociado na B3.

Sigla Ativo (mini dólar)	Lote Mínimo Padrão	Cotação de Fechamento do Dia 11/07/2019 (Em pontos)	Valor do Tick	Valor do Ponto de Contrato (R\$ por ponto)
WDO	01	3.761,50	0.50	5,00

Fonte: Elaborado pelo autor (2019).

O valor do *tick* representa o intervalo entre a variação da cotação. Como exemplo, o valor cotado no fechamento no dia 11/07/2019 foi 3.761,50. Contudo, no decorrer deste dia, o preço manteve-se variando. Esta variação, por conseguinte, é de 0.50 pontos. Sendo assim, quando a cotação estava numa tendência de alta, ou seja, a cotação dos preços estava em ascendência, o valor do ponto do ativo de 3.761,50 alternava de 0.50 em 0.50 pontos a mais. Quer dizer, o valor era 3.761,50, num determinado momento, atingiu a cotação para 3.762,00. Por outro lado, o valor do ponto de contrato é a variação monetária acarretada pela movimentação dos preços conforme Equação 3.1.

$$lucro = \frac{\text{valorizao em pontos} \times \text{reais por pontos} \times \text{contratos}}{\text{valor do tick}} \quad (3.1)$$

Caso o *trader* realizasse a compra de um lote mínimo de 01 contrato no valor de 3.761,50, e; por conseguinte, rematasse a operação de venda no valor 3.770,50, a diferença entre o preço de compra e o preço de venda seria de 09 pontos. De maneira que cada tick valha R\$ 5,00, como resultado da negociação de 01 contrato, o *trader* obteve um lucro - não foi considerado taxas operacionais e impostos - de R\$ 90,00.

Visando elucidar melhor o que foi descrito no parágrafo anterior, a Figura 12 descreve a matemática do lucro ao comprar um lote mínimo padrão (contrato) do ativo WDO.



Figura 12 – Representação de uma operação lucrativa.

Fonte: TradingView (2019).

Imaginando que a compra aconteceu no valor de 5.378,5 e o *trader* especulava uma valorização de 09 pontos do ativo, a venda seria no valor de 5.387,5. Como o valor do tick valha R\$ 5,00, ao encerrar a operação, o *trader* teria embolsado R\$ 90,00. Substituindo estes valores na Equação 3.1, o lucro foi calculado conforme Equação 3.2.

$$lucro = \frac{9 \times 5 \times 1}{0,5} = 90 \text{ reais} \quad (3.2)$$

3.2 ANÁLISE TÉCNICA

A aplicabilidade da Análise Técnica nos mercados financeiros ao redor do mundo vem crescendo com o passar dos anos e sendo adotada por um número maior de investidores (FERNANDES; HAMBERGER; VALLE, 2015). A aderência por esta técnica pode ser identificada tanto entre os pequenos e autônomos investidores quanto às grandes instituições financeiras como bancos e empresas de investimentos. Para o pequeno investidor, a Análise Técnica é mais didática por possuir ferramentas mais intuitivas e que não exige deste um grande conhecimento em economia, sendo algumas delas de fácil utilização e discernimento. Por outro lado, para as grandes instituições financeiras, a Análise Técnica pôde ser adicionada ao vasto conjunto de técnicas, funcionando como mais um instrumento corroborativo de inferências sobre o mercado

No entanto, a utilização da Análise Técnica como mecanismo de predição da tendência dos preços dos ativos a partir dos seus dados históricos foram, por muito tempo, depreciada pela comunidade científica financeira (BOAINAIN, 2009). Ao contrário da Análise Fundamentalista, que obteve uma rápida adesão por parte dos especialistas das finanças

quantitativas modernas, a Análise Técnica não era considerada uma ciência para os mesmos (LO; MAMAYSKY; WANG, 2000). Malkiel (2016), em seu livro “*A Random Walk Down Wall Street: The Time-Tested Strategy for Successful Investing*”, conclui que “Sob escrutínio científico, a leitura de gráficos deve compartilhar um pedestal com a alquimia”².

Com o passar dos anos, diversos artigos apresentam uma nova perspectiva científica a respeito da Análise Técnica e que esta, quando utilizada corretamente, pode ser uma forma eficiente de extrair informações pertinentes dos preços do mercado. Ao analisarem os índices de ações semanais nos EUA, os autores Lo e MacKinlay (1997), Lo e MacKinlay (1988), demonstraram que os preços passados podem ser usados para prever retornos futuros até certo ponto. Esta premissa, vai de certa forma, contra a teoria *Random Walk Hypothesis* proposta por Malkiel (2016) onde é implicado que as mudanças dos preços não podem ser previstas. Tabell e Tabell (1964) revisaram algumas das técnicas mais utilizadas por analistas técnicos demonstrando que é possível obter bons retornos financeiros. Treynor e Ferguson (1985) concluíram que o estudo dos dados históricos dos preços combinados com informações macroeconômicas, permite a obtenção de lucros extraordinários. Goh et al. (2013) também mostraram que em alguns títulos do tesouro, a análise técnica pode gerar melhor desempenho na previsão da tendência dos preços do que os preditores macroeconômicos.

Os *traders* técnicos baseiam suas análises na premissa de que os padrões de preços de mercado são recorrentes no futuro e, portanto, esses padrões podem ser usados para fins preditivos. A motivação por trás da Análise Técnica é ser capaz de identificar mudanças nas tendências em um estágio inicial e manter uma estratégia de investimento até que o peso da evidência indique que a tendência foi revertida (GENÇAY, 1998).

Segundo Fernandes, Hamberger e Valle (2015), a partir das informações passadas dos ativos, indicadores técnicos são calculados através de técnicas de inferência, que com base na análise computadorizada, tenta encontrar pontos claros de compra e venda baseados em observações estatísticas, como Índice de Força Relativa (IFR), Médias Móveis, Osciladores Estocástico. Todo analista técnico acredita que a demanda e oferta pode ser detectada a partir do gráfico de preços (BROCK; LAKONISHOK; LEBARON, 1992a), que está relacionada com a interpretação de algumas figuras típicas de alta ou de baixa comumente conhecidas como padrão de *Candlestick*.

3.2.1 Padrão de *Candlestick*

A Análise Técnica, como abordado anteriormente, busca sinais emitidos pelo mercado em uma série de preços passados. Para facilitar esta análise, os sinais normalmente são buscados dentro de gráficos. Os analistas técnicos em um mercado financeiro possuem diversos tipos gráficos para auxiliá-los em tomadas de decisões, dentre esses, um dos mais

² Tradução livre realizada pelo autor.

utilizados é o gráfico com padrão de *Candlestick*. O grande uso deste gráfico se deve ao fato de ele possuir um volume de informações superior aos outros, além de possuir uma visualização clara e fácil.

Fernandes, Hamberger e Valle (2015) citam que o gráfico de *Candlestick* surgiu durante o século XVIII no Japão como uma metodologia de registro dos preços do arroz no mercado futuro, bem como uma ferramenta de previsão de movimentos de preços. Nison (2001), ao escrever um dos mais amplos trabalhos sobre os padrões de *Candlesticks*, credita a origem dessa ferramenta técnica ao comerciante japonês Munehisa Homma, que difundiu essa metodologia no Japão após ter obtido grandes fortunas no mercado futuro de arroz utilizando essas figuras gráficas. Esta análise foi difundida nos mercados ocidentais a partir do trabalho de Nison (2001) denominado *“Japanese Candlestick Charting Techniques: a Contemporary Guide to the Ancient Investment Techniques of the far East”* e, como citou o próprio Nison (2001), “desde que foi introduzida no mundo ocidental, a técnica dos *Candlesticks* se tornou onipresente, disponível em todos os *softwares* e nos pacotes gráficos online”³.

A Figura 13 demonstra um gráfico de *Candlestick* semanal da ação da empresa Petrobras Preferencial (petr4) entre o intervalo de tempo de 28/12/2018 até 07/07/2019. Como se pode observar, os gráficos de *candlesticks* são formados por inúmeras barras em formato de velas, também chamadas de *candles*. Como se trata de um gráfico semanal, cada *candle* do gráfico representara uma semana de negociação do ativo. É possível observar que cada *candle* é formado por uma barra densa, verde ou vermelha (alguns gráficos utilizam respectivamente as cores branca e preta), denominada de corpo real do *candle*. Muitas vezes vemos linhas estendidas acima e abaixo do corpo real, chamadas de sombras.



Figura 13 – Exemplo de um gráfico padrão *Candlestick*.

Fonte: Metatrader 5 (2019).

³ Tradução livre realizado pelo autor.

Para operações intraday em que são executadas as estratégias de *day trade* os padrões de *candlestick* normalmente utilizados são de períodos de 01 minuto a 60 minutos. Na Figura 14, temos uma leitura do significado de cada *candle* dentro de um gráfico de *candlestick*. O corpo do *candle* representa a diferença entre a abertura e o fechamento de período de negociações. O corpo escuro significa que o fechamento se deu abaixo do preço de abertura, e o corpo claro, significa que o fechamento foi acima da abertura. As sombras representam o máximo e o mínimo que os preços atingiram num determinado período.

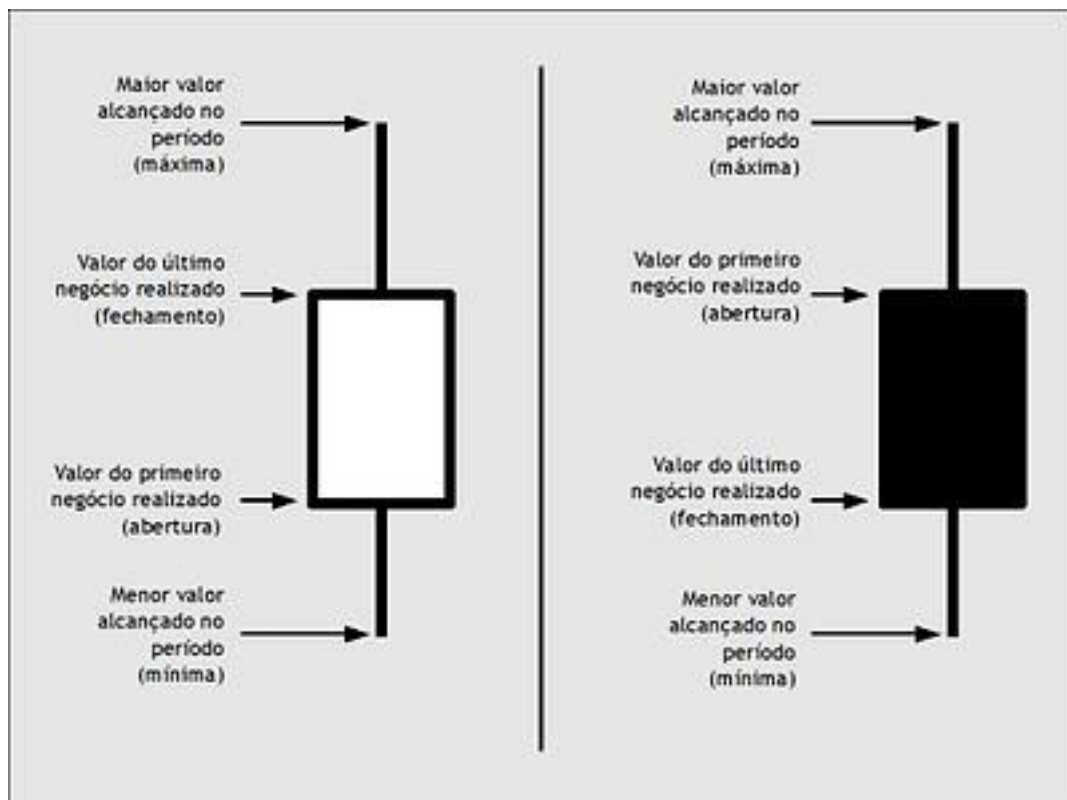


Figura 14 – Interpretando o padrão *Candlestick*.

Fonte: Fernandes, et al. (2015).

3.2.2 Média Móvel

Conforme mencionado anteriormente, os *traders* técnicos baseiam suas análises na premissa de que os padrões de preços de mercado são recorrentes no futuro, portanto, esses padrões podem ser usados para fins preditivos. A motivação por trás da análise técnica é ser capaz de identificar mudanças nas tendências em um estágio inicial e manter uma estratégia de investimento até que o peso da evidência indique que a tendência foi revertida.

Para tal tarefa, mecanismos de suavização dos preços são incrementados nos gráficos. Denominados como indicadores técnicos, estes desenvolvem o papel de orientar os *traders* realizando cálculos para que movimentos futuros sejam detectados. Programas grá-

ficos possuem uma infinidade de indicadores, desde osciladores, seguidores de tendência e volumes.

Sem sombra de dúvida, um dos indicadores técnicos mais utilizados pelos analistas técnicos é a média móvel, tendo como principal função indicar a direção da tendência dos preços (GENÇAY, 1998). Partindo desta premissa, diversas regras de negociação foram desenvolvidas a partir da utilização deste indicador técnico. No livro “Estratégias Operacionais de Análise Técnica de Ações”, Palex (2014) descreve com primazia diversas regras de negociação com base no indicador de médias móveis. Brock, Lakonishok e LeBaron (1992a) e Lo, Mamaysky e Wang (2000) encontraram fortes evidências de previsibilidade de retorno financeiro ao usar análise técnica com base em uma estratégia de média móvel.

3.2.2.1 Média Móvel Simples

Com objetivo de suavizar o ruído gerado pela movimentação dos preços, a média móvel simples (MMS) é utilizada em larga escala tal qual um indicador que visa apontar a tendência do ativo financeiro.

Brock, Lakonishok e LeBaron (1992b), aplicaram em suas análises, média móvel simples no ativo financeiro *Dow Jones Industrial Average*. Contudo, para calcular a média móvel simples, utiliza-se da Fórmula 3.3:

$$MMSn_t = \frac{C_t + C_{t-1} + \dots + C_{n-(t-1)}}{n} \quad (3.3)$$

onde $MMSn_t$ é a média móvel simples, n é o período de tempo estipulado (dias, semanas, minutos), e C_t é o preço de fechamento no período t . A Equação 3.3 pode ser escrita mais simplificada conforme Equação 3.4:

$$MMSn_t = \frac{1}{n} \sum_{i=0}^{t-1} C_{n-i} \quad (3.4)$$

Na Figura 15 é demonstrada a média móvel simples de 20 dias - 20 períodos - na ação da Petrobras Preferencial (petr4). Explicitamente o preço apresenta-se mais suavizado, indicando que o ativo financeiro encontra-se numa tendência de alta.



Figura 15 – Média móvel simples de 20 períodos na ação PETR4 (PN).

Fonte: Meatrader 5 (2019).

Uma das regras de negociação utilizadas pelos *traders* é o cruzamento de médias. A regra de negociação de cruzamento de médias utiliza-se da média móvel simples para determinar quando comprar e quando vender um ativo financeiro. A regra é definida da seguinte forma: quando a média curta⁴ é maior que a média longa⁵, caracteriza-se compra. Por outro lado, no sentido inverso, quando a média curta é menor que a média longa, caracteriza-se venda.

- Regra de negociação cruzamento de média para compra (Figura 16):

$$MMSn_t(C_t, 9, 0) > MMSn_t(C_t, 20, 0) \quad (3.5)$$

onde $MMSn_t(C_t, 9, 0)$ é a média curta de 9 períodos baseada no preço de fechamento C_t , representada pela cor vermelha. E $MMSn_t(C_t, 20, 0)$ é a média móvel longa de 20 períodos baseada no preço de fechamento C_t , representada na cor azul.

- Regra de negociação cruzamento de média para venda (Figura 16):

$$MMSn_t(C_t, 9, 0) < MMSn_t(C_t, 20, 0) \quad (3.6)$$

onde $MMSn_t(C_t, 9, 0)$ é a média curta de 9 períodos baseada no preço de fechamento C_t , representada pela cor vermelha. E $MMSn_t(C_t, 20, 0)$ é a média móvel longa de 20 períodos baseada no preço de fechamento C_t , representada na cor azul.

⁴ É denominada média curta a média móvel com menor período. Exemplo: A média móvel com 9 períodos em relação a média móvel com 20 períodos torna-se a média móvel curta.

⁵ É denominada média longa a média móvel com maior período. Exemplo: A média móvel com 20 períodos em relação a média móvel com 9 períodos torna-se a média móvel longa.

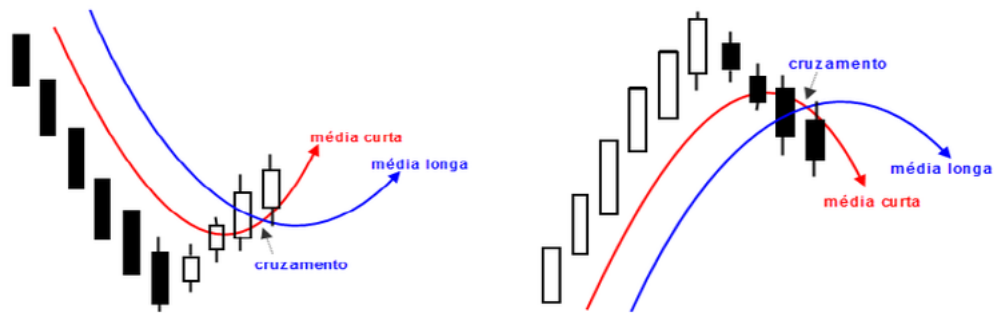


Figura 16 – Regra de negociação baseada em cruzamento de médias móveis simples.

Fonte: Palex (2014).

3.2.2.2 Média Móvel Exponencial

A média móvel exponencial (MME) tem a mesma função da média móvel simples (MMS), suavizar a movimentação dos preços e indicar a tendência. Contudo, em sua fórmula é acrescentado um multiplicador visando pôr mais peso para os dados mais recentes. Para se calcular a média móvel exponencial, utiliza-se da Fórmula 3.7:

$$MME_n = \left[(C_t - MME_{n-1}) \times \left(\frac{2}{n+1} \right) \right] + MME_{n-1} \quad (3.7)$$

onde MME é a média móvel exponencial, n é o período de tempo estipulado (dias, semanas, minutos), e C_t é o preço de fechamento. Na Figura 17 é demonstrada a média móvel simples de 20 dias - 20 períodos - e a média móvel exponencial de 20 dias - 20 períodos na ação da Petrobras Preferencial (petr4) visando apresentar a distinção entre os dois modelos. A média móvel simples está representada pela cor vermelha e a exponencial pela cor preta.

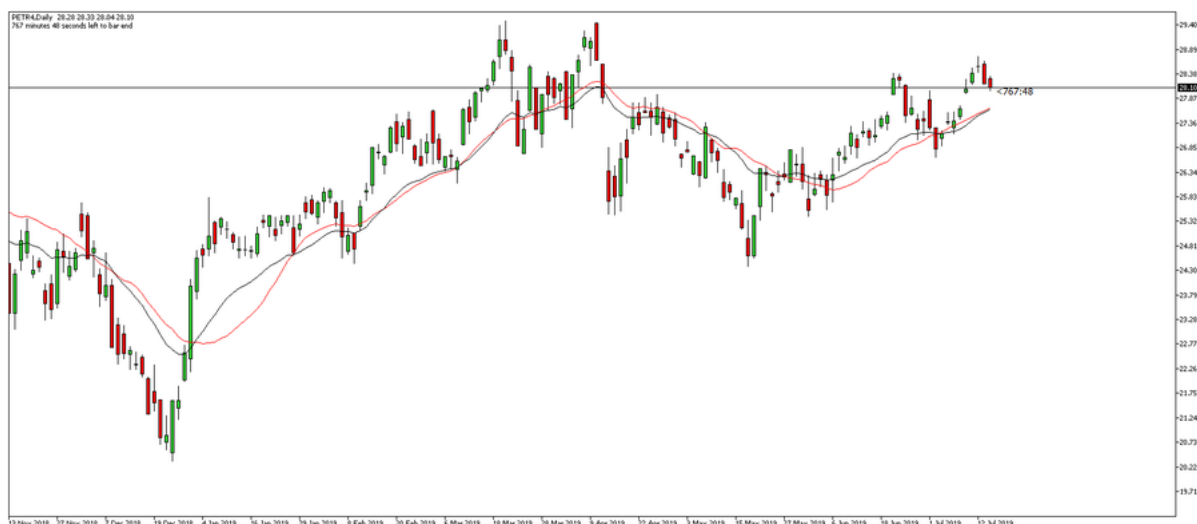


Figura 17 – Média móvel simples e exponencial de 20 períodos na ação PETR4 (PN)

Fonte: Metatrader 5 (2019)

A regra de negociação 9,1, criada pelo *trader* norte-americano Larry Williams, utiliza-se da média móvel exponencial para determinar quando comprar e quando vender um ativo

financeiro. A regra é definida da seguinte forma, quando o preço de fechamento é maior que a média exponencial de 9 períodos, caracteriza-se compra. Por outro lado, quando ao contrário, quando o preço de fechamento é menor que a média exponencial de 9 períodos, caracteriza-se venda.

- Regra de negociação 9,1 para compra (Figura 18):

$$C_t > MME(C_t, 9, 0) \quad (3.8)$$

onde C_t é o preço de fechamento e $MME(C_t, 9, 0)$ seria a média exponencial de 09 períodos do preço de fechamento C_t .

- Regra de negociação 9,1 para venda (Figura 18):

$$C_t < MME(C_t, 9, 0) \quad (3.9)$$

onde C_t é o preço de fechamento e $MME(C_t, 9, 0)$ seria a média exponencial de 09 períodos do preço de fechamento C_t .



Figura 18 – Regra de negociação 9,1 para compra, criada pelo trader Larry Williams.

Fonte: Palex (2014).

3.3 REDES NEURAIS ARTIFICIAIS

A Rede Neural Artificial (ANN) é um mecanismo de processamento artificial de informação que é inspirado nos sistemas nervosos biológicos. A Rede Neural Artificial compreende alguns elementos interconectados, ou neurônios, trabalhando juntos como um sistema para resolver problemas específicos. Logo, o neurônio é a parte estrutural tendo como principal função a manutenção do processamento das informações da rede neural artificial (HINTON, 2002).

Sendo uma classe de algoritmo de aprendizagem de máquina, este modelo visa traçar padrões complexos de dados empregando um conjunto de funções matemáticas nas diversas camadas do sistema. Os valores de entradas são introduzidos no conjunto neural artificial e são transformados e transportados através dessas camadas, fornecendo as informações necessárias para a aprendizagem do modelo para gerar uma previsão a partir da combinação de todos os neurônios (HAIKIN, 2001).

O funcionamento parte do princípio da utilizando de técnicas de otimização aplicadas em cada ciclo de treinamento, obtendo como resultado, métricas de erro onde são calculadas com base na diferença entre os valores de saídas reais dos dados de entradas com os valores de saídas previstos. A partir da técnica conhecida como *backpropagation*, as derivadas dos erros métricos gerados são calculadas e propagadas de volta através da rede neural artificial. Esse processo é repetido iterativamente até que o erro de rede diminua abaixo de um limite aceitável (HINTON, 2002).

Para Inthachot, Boonjing e Intakosum (2016), rede neural artificial é um dos algoritmos de aprendizado de máquina mais famosos e vem sendo utilizado largamente em previsões de séries temporais nos índices de bolsas de valores, ações, tendências e mercado financeiro. Kimoto et al. (1990) foi o primeiro a aplicar o algoritmo de rede neural modular visando prever a movimentação dos preços do índice da bolsa de valores de Tóquio. Wu e Lu (1993) utilizaram redes neurais artificiais para prever a movimentação do índice relacionado ao *S&P 500*⁶. Birgul, Ozturan e Badur (2003) aplicaram redes neurais artificiais para prever a tendência do índice *ISE*. Já Bollen, H.Mao e Zeng (2011) usaram dados postados no Twitter para antever a movimentação do índice *Dow Jones*⁷. Wang et al. (2016) empregaram redes neurais recorrentes (RNN) para prever a movimentação da tendência nos índices *SSE*⁸, *TWSE*⁹, *KOSPI*¹⁰ e *nikkei225*¹¹. Chen e He (2018), deixando o modelo de regressão à parte, aplicaram redes neurais artificiais no modelo de classificação binária tendo como *output* os valores zero e um quando a tendência é de baixa ou de alta numa ação chinesa.

3.3.1 Neurônio Artificial

Conforme descrito anteriormente na Seção 3.3, um neurônio é a base de toda a estrutura de uma rede neural artificial tornando-se a responsável, de fato, do processamento da troca de informação entre as camadas.

⁶ Trata-se de um índice composto por quinhentos ativos cotados nas bolsas de NYSE ou NASDAQ.

⁷ Índice criado em 1896 pelo editor do *The Wall Street Journal* e fundador do *Dow Jones & Company*, Charles Dow.

⁸ Índice do mercado de ações de todas as ações negociadas na Bolsa de Valores de Xangai.

⁹ Índice do mercado de ações para empresas negociadas na Bolsa de Valores de Taiwan.

¹⁰ Índice de Bolsa de Valores sul-coreana.

¹¹ Principal índice econômico da Bolsa de Valores de Tóquio.

O delineamento de um neurônio artificial é dada através da Figura 19.

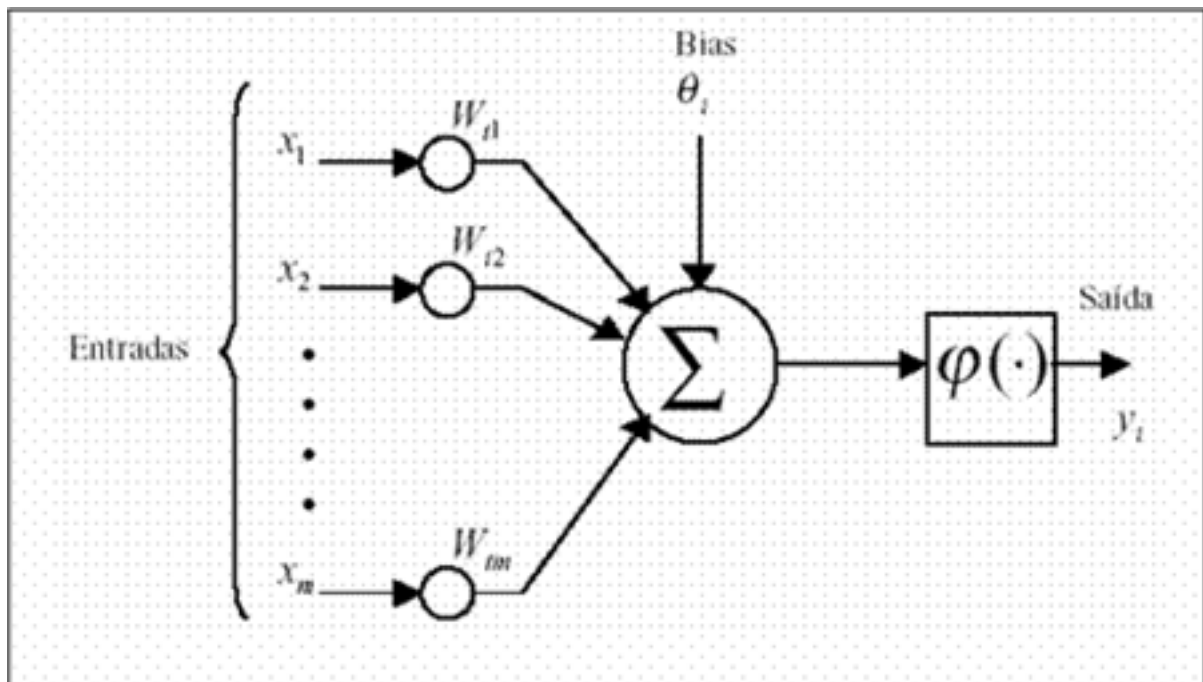


Figura 19 – Neurônio artificial.

Fonte: Haykin (2001)

O neurônio artificial, também conhecido por *perceptron*, demonstrado na Figura 19, tem como principais características transformar a cognição humana numa generalização de modelos matemáticos. Contudo, o neurônio artificial termina possuindo as seguintes características:

- A propagação dos sinais a partir das entradas x_m são transferidas através de conexões ou sinapses;
- As sinapses ponderam o sinal transmitido através da associação a um peso w_m ;
- Cada neurônio (ou unidade) aplica uma função de ativação $\varphi(\cdot)$ - geralmente não-linear - à sua entrada de rede para determinar sua saída;
- Sendo aplicado externamente, o viés θ tem a função de ponderar a entrada da função de ativação, dependendo se ele é positivo ou negativo. Termos de viés são constantes adicionais conectadas aos neurônios e adicionadas à entrada ponderada antes da função de ativação é aplicado. O viés auxilia os modelos a representar padrões que não passam necessariamente pela origem. Por exemplo, se todos os seus recursos fossem 0, sua saída também seria zero? É possível que exista algum valor base sobre o qual seus recursos têm um efeito? Os termos de viés normalmente acompanham os pesos e também devem ser aprendidos pelo seu modelo.

Na forma matemática, o neurônio artificial k pode ser descrito conforme Equação 3.10 e 3.11:

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (3.10)$$

e

$$y_k = \varphi(u_k + b_k) \quad (3.11)$$

onde, x_j são os sinais de entrada; w_{kj} são os pesos sinápticos do neurônio k , u_k é a saída do combinador linear devido aos sinais de entrada; b_k é o viés; φ é a função de ativação e y_k é o sinal de saída do neurônio. O uso do viés b_k tem o efeito de aplicar uma transformação afim à saída u_k do somador linear no modelo da Figura 19, como mostrado na Equação 3.12:

$$v_k = u_k + b_k \quad (3.12)$$

O viés b_k é um parâmetro externo do neurônio artificial k . É possível considerar sua presença como na Equação 3.11. Equivalentemente se podem formular as Equações 3.13 e 3.14 como segue:

$$v_k = \sum_{j=0}^m w_{kj} x_j \quad (3.13)$$

$$y_k = \varphi(v_k) \quad (3.14)$$

Na Equação 3.15 se adicionou uma nova sinapse. A sua entrada e o seu peso são dados, respectivamente, por:

$$x_0 = +1 \quad (3.15)$$

$$w_{k0} = b_k \quad (3.16)$$

É possível, portanto, reformular o modelo do neurônio k (da Figura 19) onde o efeito do viés é considerado de duas maneiras: (i) adicionando-se um novo sinal de entrada fixo em +1 e (ii) adicionando-se um novo peso sináptico igual ao viés b_k .

3.3.2 Topologia da Rede Neural Artificial

A estrutura de uma rede neural artificial é determinada pelo arranjo das conexões entre os neurônios e pela natureza dessas conexões. A maneira de treinar ou ajustar os pontos fortes dessas conexões de modo a atingir um comportamento geral desejável é conhecido como processo de aprendizagem (KERAMATI, A. et al., 2014).

Possuindo uma estrutura complexa, as redes neurais artificiais são bem conhecidas pelo seu poder de modelagem empírico bem estabelecido. Outra característica marcante; pode-se dizer, é a sua capacidade de aprender automaticamente a partir dos dados disponíveis, a fim de fornecer um meio para as previsões (KERAMATI, A. et al., 2014).

A composição básica da rede neural artificial, segundo Maharani e Murfi (2019), consiste em camada de entrada (*input layer*), camada oculta (*hidden layer*) e camada de saída (*output layer*).

- Camada de entrada (*input layer*): fornece os dados em que seu modelo irá treinar. Cada neurônio na camada de entrada representa um atributo exclusivo no seu conjunto de dados;
- Camada oculta (*hidden layer*): situa-se entre as camadas de entrada e saída. Aplica-se uma função de ativação de modo a transmitir os resultados de uma camada para outra. Muitas das vezes a rede neural baseia-se em mais de uma camada oculta. Porém, nas redes tradicionais, as camadas ocultas exercem a conexão da saída de um neurônio para entrada de outro neurônio.
- Camada de saída (*output layer*): é a camada final em uma rede neural artificial. Ele recebe informações da camada oculta anterior, podendo assim, aplicar opcionalmente uma função de ativação, e retorna uma saída que representa a previsão do seu modelo.

A Figura 20 exibe uma rede neural artificial com apenas uma camada oculta.

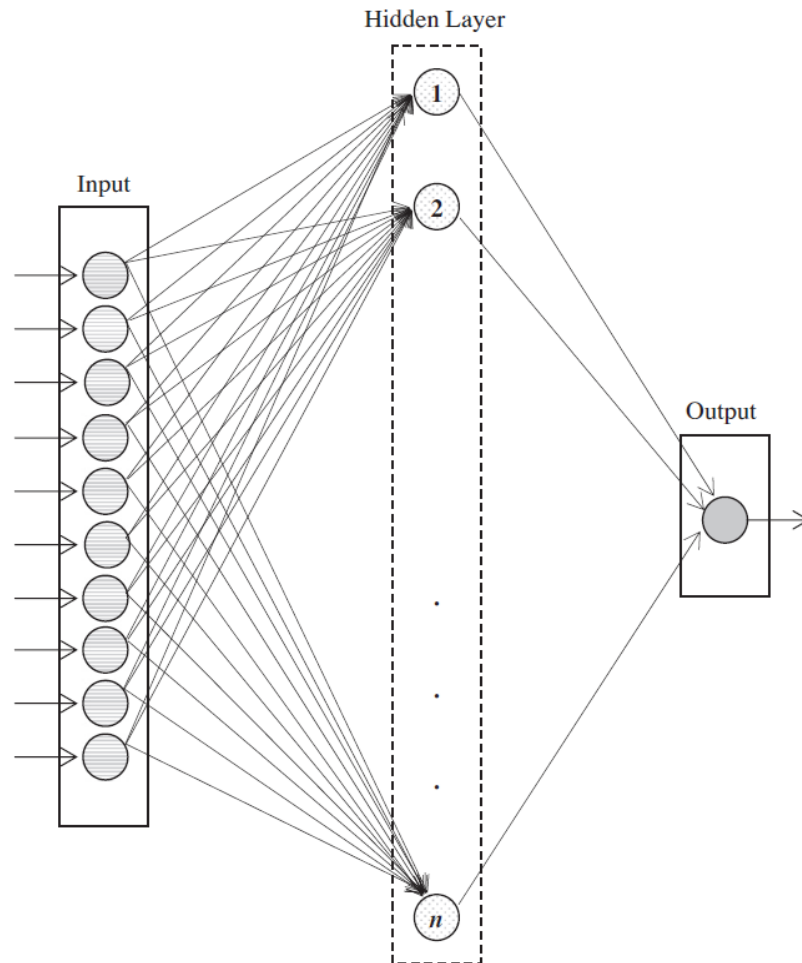


Figura 20 – Arquitetura ANN.

Fonte: Kara, Y. et al. (2011).

3.3.2.1 Redes Neurais Artificiais Profundas

Segundo Yin et al. (2015), o modelo de redes neurais artificiais profunda - ou *Deep Neural Network* (DNN) - é uma extensão do modelo rede neural artificial (ANN), onde o número de camadas na camada oculta (*hidden layer*) é maior que um. O modelo DNN tem sido amplamente utilizado para resolver problemas relacionados ao reconhecimento de fala, reconhecimento de padrões, reconhecimento de imagens e vários outros aplicativos. Além disso, o DNN pode aprender recursos de alto nível com mais complexidade e abstração do que a rede neural artificial. Através de um recurso conhecido como *feedforward Neural Network*, o neurônio é alimentado por um dado de entrada, para que em seguida, a rede neural retorna um vetor de saída.

Segundo Nielsen (2019), matematicamente, o processamento dos dados realizado pela ANN fundamenta-se através do *output* de ativação a_j^i do neurônio j^{th} referente ao *layer* i^{th} . Onde a_j^1 é o j^{th} elemento do vetor de entrada. Sendo assim, podemos relacionar a

entrada do próximo neurônio através da seguinte relação:

$$a_j^i = \sigma \left(\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i \right) \quad (3.17)$$

onde:

- σ é a função de ativação,
- w_{jk}^i refere-se ao peso k^{th} pertinente ao *layer* $(i - 1)^{th}$ para o neurônio j^{th} posicionado na camada i^{th} ,
- b_j^i corresponde ao viés do neurônio j^{th} na camada i^{th} , e
- a_j^i representa o valor de ativação do neurônio j^{th} na camada i^{th} .

Em alguns casos é escrito z_j^i para representar $\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i$, ou seja, em outras palavras, seria o valor de ativação antes de aplicar a função de ativação no neurônio no dado de saída do neurônio. O *feedforward Neural Network* de uma DNN pode ser representada na forma de um diagrama de rede, como mostra a Figura 21.

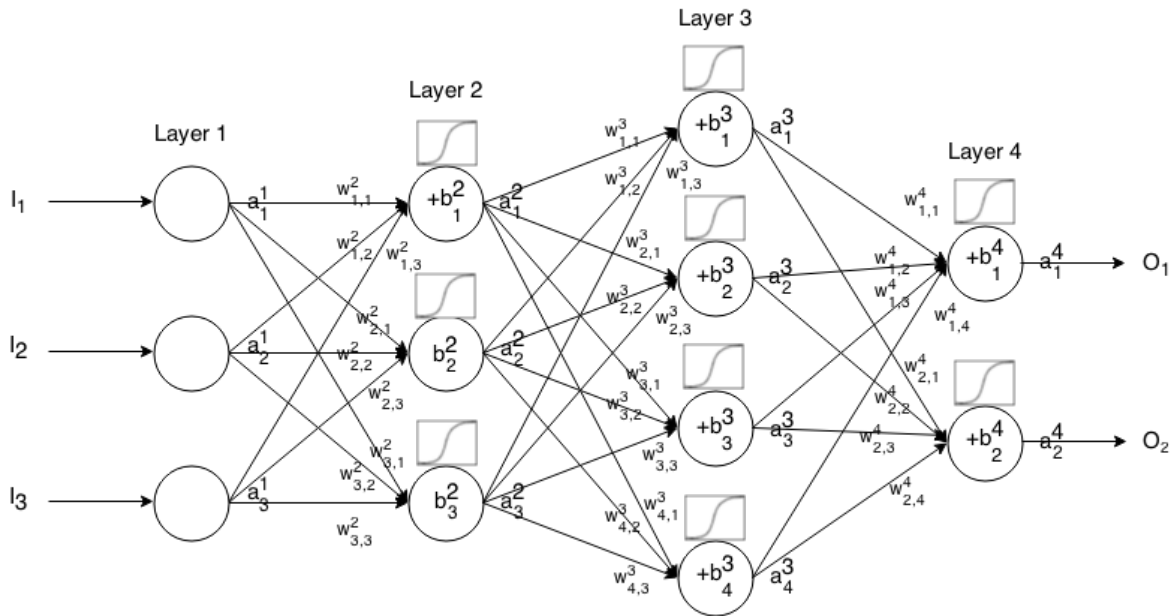


Figura 21 – Diagrama de rede para a rede neural de duas camadas correspondente a equação.

Fonte: Nielsen (2019).

Nielsen (2019) explica que pode ter uma notação mais concisa conforme equação abaixo:

$$a^i = \sigma (w^i \times a^{i-1} + b^i) \quad (3.18)$$

Para usar esta fórmula para calcular a saída de uma rede de *feedforward* para algum *input* $I \in R^n$, defina $a^1 = I$, e depois calcule a^2, a^3, \dots, a^m , onde m é o número de camadas.

3.3.2.1.1 Camada de Abandono (Dropout)

No modelo de redes neurais artificiais profunda (DNN), é conhecido que a quantidade das camadas ocultas é superior a um. Sendo esta a principal diferença entre uma rede neural artificial profunda (DNN) e a rede neural artificial (ANN), conforme explicitado na seção 3.3.2.1. Contudo, este aumento no número de camadas ocultas acarretou da utilização de uma maior porção de parâmetros tendendo a exigir maior empenho da função de custo (é abordado na Seção 3.3.4.2). Super ajustando (*overfitting*), por assim dizer, o modelo.

Partindo desta premissa, nos modelos de redes neurais artificiais profunda (DNN) adota-se a técnica da camada de abandono. Segundo Maharani e Murfi (2019), a técnica da camada de abandono exclui aleatoriamente alguns neurônios durante o processo de treinamento, minimizando a possibilidade de super ajustamento do modelo. (SRIVASTAVA et al., 2014) evidenciaram que a aplicação da camada de abandono melhora o desempenho da rede neural artificial profunda em estudos realizados nos campos da visão, reconhecimento de fala, biologia computacional e na classificação de documentos.

3.3.3 Função de Ativação das Camadas Ocultas

Segundo Maharani e Murfi (2019), função de ativação é o responsável pelo estímulo dos neurônios referente a camada de entrada para os neurônios referentes às camadas de saídas. Para tal ativação, esta conexão pode ser realizada através de funções como: funções lineares, funções *sigmóides*, funções *softmax*, funções *tanh* e funções *ReLU*.

As funções de ativação situam-se internamente das camadas da rede neural e têm como propósito alterar os dados recebidos antes de transferi-los para a camada seguinte. Logo, as funções de ativação possuem uma larga influência às redes neurais, permitindo que seja moldado ligações não lineares relevantes.

As funções de ativação geralmente têm as seguintes propriedades:

- **Não linearidade** - na regressão linear, estamos limitados a uma equação de previsão que se parece com uma linha reta. Isto é agradável para conjuntos de dados simples e com um relacionamento individual entre entradas e saídas. Mas, por outro lado, e se os padrões em nosso conjunto de dados não existam linearidades (por exemplo, 2, sin, log)? Para realizar este tipo de modelagem entre as camadas, é necessário aplicar uma equação não linear.
- **Continuamente diferenciáveis** - para melhorar nosso modelo, é preciso que a saída tenha uma boa inclinação para calcular as derivadas dos erros em relação aos pesos. Se o neurônio produzisse 0 ou 1 (*perceptron*), não seria possível saber em que direção atualizar os pesos para reduzir o erro.

- **Faixa fixa** - as funções de ativação geralmente comprimem os dados de entrada em uma faixa estreita que torna o treinamento do modelo mais estável e eficiente.

Sendo assim, a conexão entre os neurônios de uma camada para a outra adota uma associação de valores de ponderação variado dentro do espaço $[-1, 1]$ (HEIDARI, A. A. et al., 2019). Matematicamente, segue abaixo a definição das funções de ativação mais utilizadas:

- A função linear é definida pela Equação 3.19:

$$a_j^i = \sigma(z_j^i) = z_j^i \quad (3.19)$$

onde a_j^i representa o valor de ativação do neurônio j^{th} na camada i^{th} , σ é a função de ativação e z_j^i representa o valor de ativação de um neurônio antes de aplicar a função de ativação. Em outras palavras, escreve-se z_j^i para representar $\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i$.

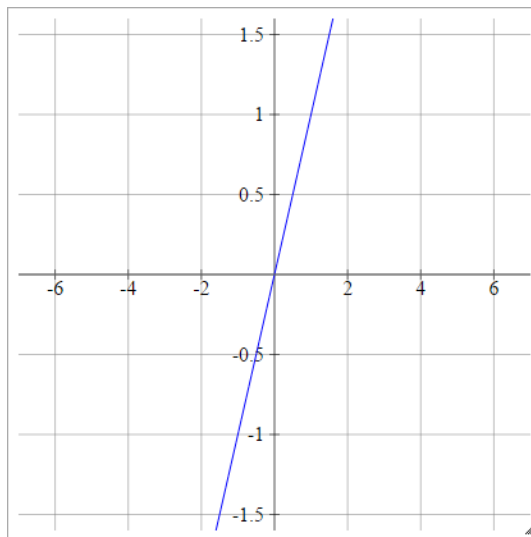


Figura 22 – Gráfico função linear.

Fonte: Nielsen (2019).

- A função *Rectified Linear Unit (ReLU)* é definida pela Equação 3.20:

$$a_j^i = \sigma(z_j^i) = \max(0, z_j^i) \quad (3.20)$$

onde a_j^i representa o valor de ativação do neurônio j^{th} na camada i^{th} , σ é a função de ativação e z_j^i representa o valor de ativação de um neurônio antes de aplicar a função de ativação. Em outras palavras, escreve-se z_j^i para representar $\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i$.

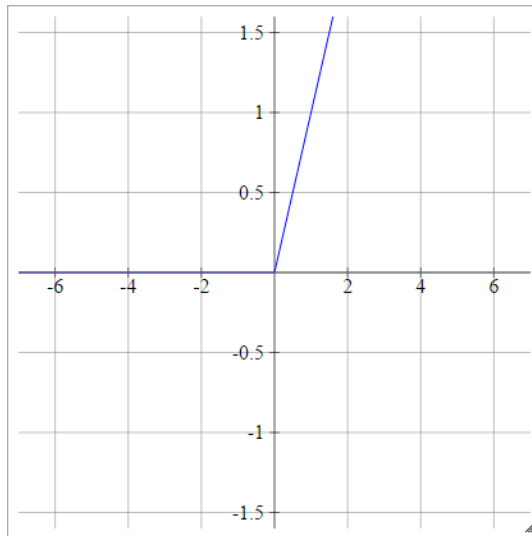


Figura 23 – Gráfico função *Relu*.

Fonte: Nielsen (2019).

- A função *sigmoid* é definida pela Equação 3.21:

$$a_j^i = \sigma(z_j^i) = \frac{1}{1 + \exp(-z_j^i)} \quad (3.21)$$

onde a_j^i representa o valor de ativação do neurônio j^{th} na camada i^{th} , σ é a função de ativação e z_j^i representa o valor de ativação de um neurônio antes de aplicar a função de ativação. Em outras palavras, escreve-se z_j^i para representar $\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i$.

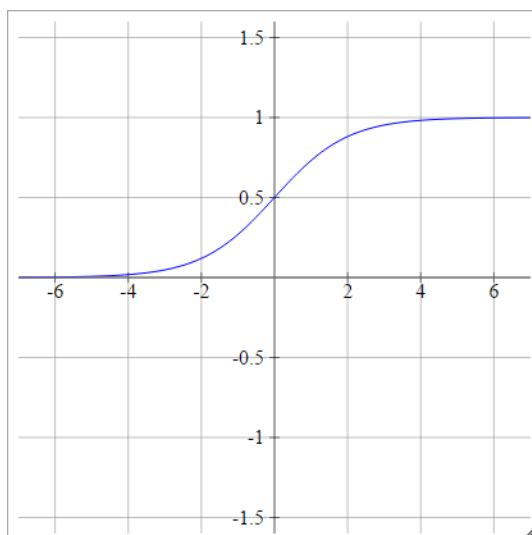


Figura 24 – Gráfico função *sigmoid*.

Fonte: Nielsen (2019).

- A função *tanh* é definida pela Equação 3.22:

$$a_j^i = \sigma(z_j^i) = \tanh(z_j^i) \quad (3.22)$$

onde a_j^i representa o valor de ativação do neurônio j^{th} na camada i^{th} , σ é a função de ativação e z_j^i representa o valor de ativação de um neurônio antes de aplicar a função de ativação. Em outras palavras, escreve-se z_j^i para representar $\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i$.

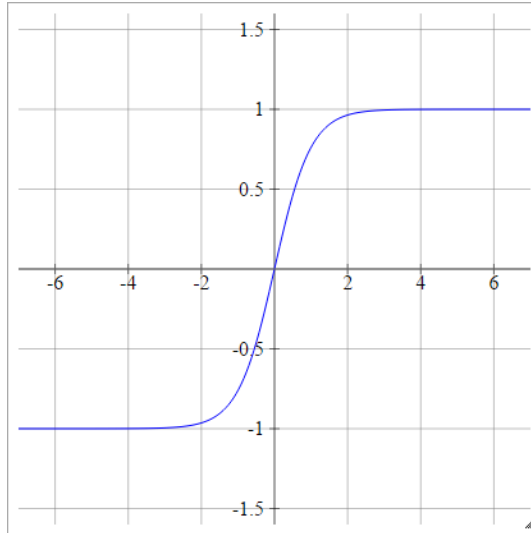


Figura 25 – Gráfico função \tanh .

Fonte: Nielsen (2019).

- A função *leaky Rectified Linear Unit (LReLU)* é definida por Equação 3.23:

$$a_j^i = \sigma(z_j^i) = 1(z_j^i < 0)(\alpha z_j^i) + 1(z_j^i \geq 0)(z_j^i) \quad (3.23)$$

onde a_j^i representa o valor de ativação do neurônio j^{th} na camada i^{th} , σ é a função de ativação e z_j^i representa o valor de ativação de um neurônio antes de aplicar a função de ativação. Em outras palavras, escreve-se z_j^i para representar $\sum_k (w_{jk}^i \cdot a_k^{i-1}) + b_j^i$. α é um hiperparâmetro com um valor muito pequeno e geralmente escolhemos 0,01 como seu valor inicial (CHEN; HE, 2018).

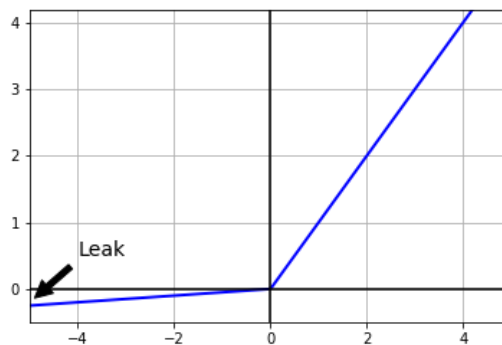


Figura 26 – Gráfico função $LReLU$.

Fonte: Nielsen (2019).

3.3.4 Processo de Aprendizagem da Rede Neural Artificial

A execução de uma tarefa a partir de uma rede neural devidamente treinada, é originada através dos ajustes dos pesos de cada sinapse, de tal maneira que, o erro gerado na camada de saída seja reduzido (HINTON, 2002). Para implementação deste procedimento o valor dos pesos passam por uma manutenção e são proporcionais à mudança da taxa de erro. O procedimento, agora conhecido como o algoritmo *backpropagation*¹² (o algoritmo *backpropagation* é abordado mais à frente na Seção 3.3.4.3), tornou-se uma das ferramentas mais importantes para o treinamento de redes neurais.

Utilizando da ideia de um neurônio artificial para exemplificar o processo de aprendizagem, segue abaixo os seguintes passos:

- 1) Conforme dito anteriormente, a rede neural artificial é alimentada com o fornecimento de dados através da camada de entrada. Conforme Figura 27, os dados de entradas são as variáveis independentes *Study Hrs*, *Sleep Hrs* e *Quiz*. Representadas por X_m , os valores destas variáveis independentes são transportadas através das sinapses com um peso W_m gerados aleatoriamente, *a priori*, até ao neurônio.

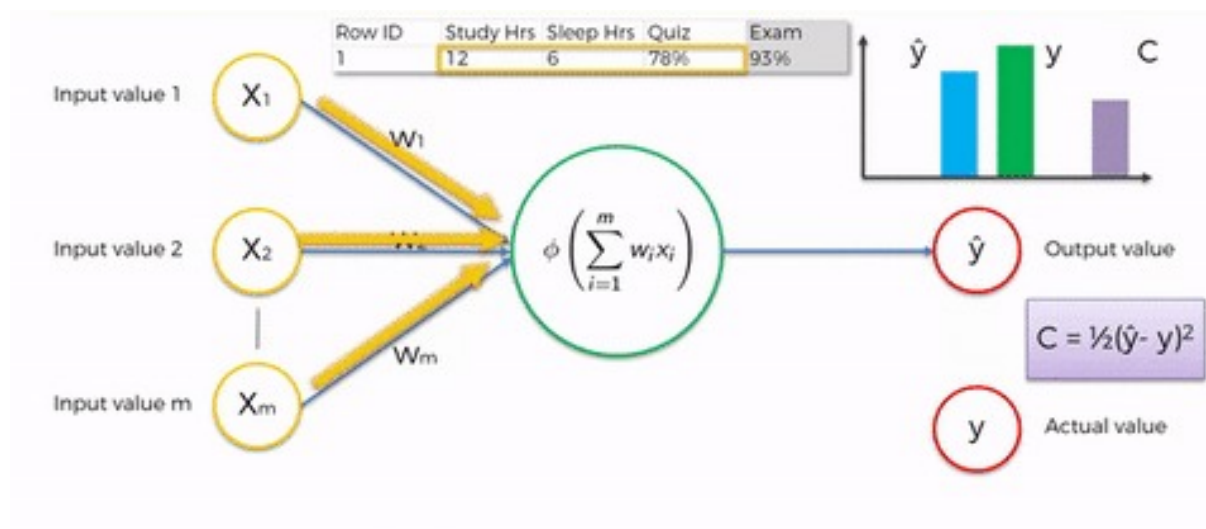


Figura 27 – Entrada das variáveis independentes no neurônio artificial.

Fonte: Super Data Science (2019).

- 2) Após as variáveis independentes serem remodeladas pela função de ativação, através da sinapse de saída, é produzido, então, o valor previsto \hat{y} como valor de saída da última camada do neurônio artificial. Logo, conforme Figura 28, o valor previsto \hat{y} é comparado com o valor atual y - variável dependente *Exam* - e aplicado a função de custo C (a função de custo é abordada mais à frente na Seção 3.3.4.2).

¹² Em 1974, Paul J. Werbos inventou o algoritmo *backpropagation* enquanto estava trabalhando num doutorado na Universidade de Harvard.

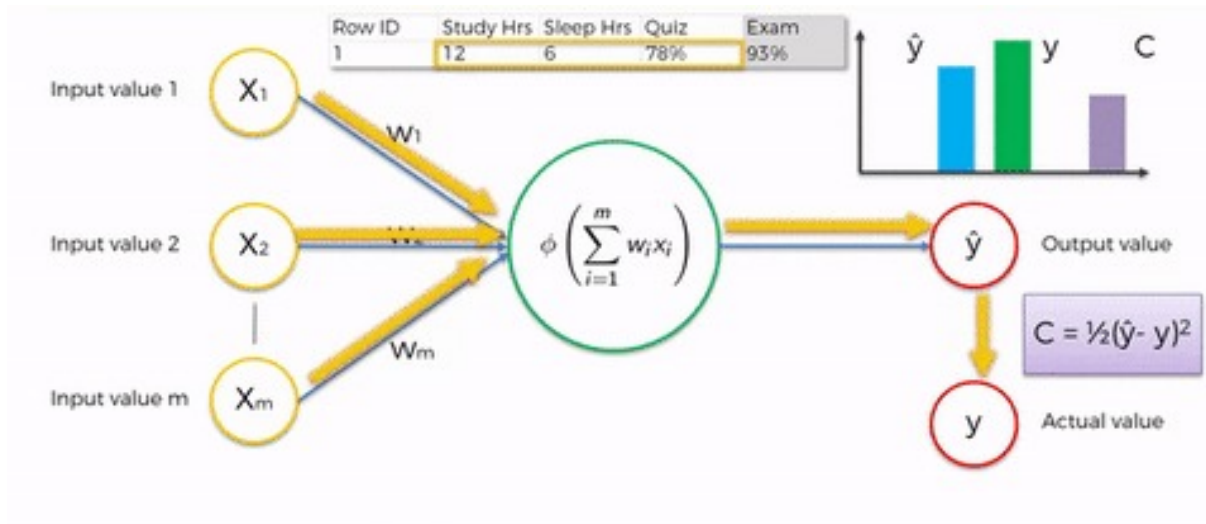


Figura 28 – Processo de saída do valor previsto no neurônio artificial.

Fonte: Super Data Science (2019).

- 3) Comparados o valor previsto \hat{y} com o valor atual y e calculado a função de custo C , é aplicado o algoritmo *backpropagation* (o algoritmo *backpropagation* é abordado mais à frente na Seção 3.3.4.3) ajustando os valores dos pesos W_m . Enquanto o valor gerado pela função de custo C não se aproxima de zero, os pesos W_m serão ajustados com novos valores conforme Figura 29. Repetem-se os passos 1 e 2.

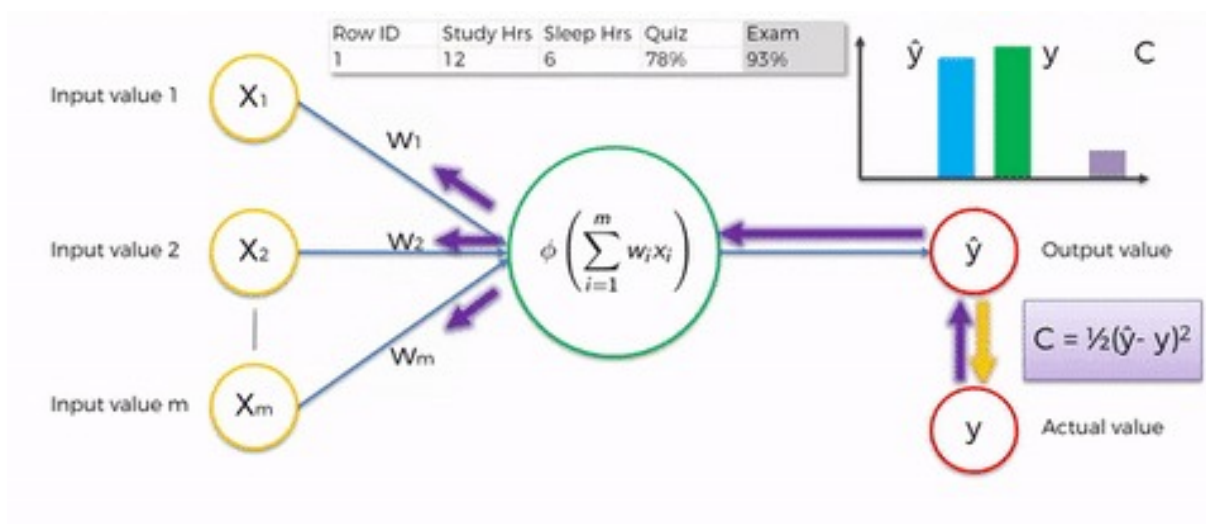


Figura 29 – Exemplificação do procedimento backpropagation.

Fonte: Super Data Science (2019).

- 4) É finalizado o processo de aprendizagem da rede neural artificial quando o valor da função de custo C aproxima-se de zero conforme Figura 30. Sendo assim, a rede neural aprende quando é encontrado os pesos W_m ótimo através da função de custo C .

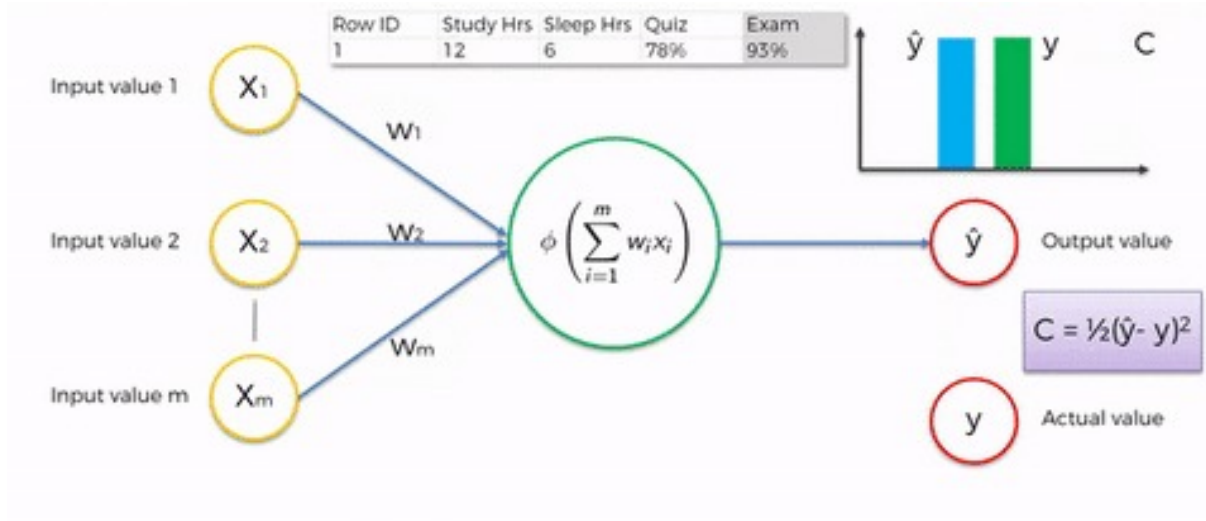


Figura 30 – Neurônio artificial treinado.
 Fonte: Super Data Science (2019).

3.3.4.1 Estimativa de Momento Adaptativo (*Adam*)

Uma rede neural artificial profunda, visando um perfeito desempenho, necessita atualizar e otimizar os pesos W_m . Para tal, é estipulado ao modelo uma taxa de aprendizado. A taxa de aprendizado, frequentemente representada por η , visa indicar a velocidade ou ritmo que os pesos serão atualizados.

A estimativa de momento adaptativo (*Adam*), atualmente, é dos algoritmos mais populares que tem como principal funcionalidade a otimização e atualização dos pesos W_m para cada dado de treinamento do modelo de redes neurais artificiais (LATHUILIERE et al., 2018).

Rather, Agarwal e Sastry (2014), utilizaram o método *Adam* como taxa de aprendizagem no modelo para predição do retorno em ações. Kingma e Ba (2014) propuseram a utilização do algoritmo *Adam* visando a otimização estocástica do modelo de redes neurais.

A estimativa de momento adaptativo (*Adam*) calcula taxas de aprendizado adaptável para cada parâmetro e funciona da seguinte maneira:

- Primeiramente, é necessário calcular a média ponderada exponencialmente dos gradientes anteriores (v_{dW}) conforme Equação 3.24;

$$v_{dW} = \beta_1 v_{dW} + (1 - \beta_1) \frac{\partial J}{\partial W} \quad (3.24)$$

- Segundo passo, calcula-se a média ponderada exponencialmente dos quadrados dos gradientes anteriores (s_{dW}) conforme Equação 3.25;

$$s_{dW} = \beta_2 s_{dW} + (1 + \beta_2) \left(\frac{\partial J}{\partial W} \right)^2 \quad (3.25)$$

- Terceiro, essas médias têm um viés em direção tendendo a zero e, para neutralizar isso, uma correção de viés é aplicada $(v_{dW}^{corrigido}, s_{dW}^{corrigido})$ conforme Equações 3.26 e 3.27;

$$v_{dW}^{corrigido} = \frac{v_d W}{1 - (\beta_1)^t} \quad (3.26)$$

$$s_{dW}^{corrigido} = \frac{s_{dW}}{1 - (\beta_1)^t} \quad (3.27)$$

- Por fim, os parâmetros são atualizados usando as informações das médias calculadas conforme Equação 3.28:

$$W = W - \alpha \frac{v_{dW}^{corrigido}}{\sqrt{s_{dW}^{corrigido} + \epsilon}} \quad (3.28)$$

onde:

- v_{dW} = a média ponderada exponencialmente dos gradientes anteriores;
- s_{dW} = a média ponderada exponencialmente dos quadrados dos gradientes passados;
- β_1 = hiperparâmetro a ser ajustado;
- β_2 = hiperparâmetro a ser ajustado;
- $\frac{\partial J}{\partial W}$ = gradiente de custo em relação à camada oculta atual;
- W = a matriz de pesos (parâmetro a ser atualizado);
- α = taxa de aprendizado;
- ϵ = valor muito pequeno para evitar dividir por zero.

3.3.4.2 Função de Custo (*Cost Function*)

Num primeiro treinamento da rede neural artificial, os parâmetros pesos W_m são gerados aleatoriamente adotando valores discrepantes do cenário ideal. Contudo, visando aperfeiçoar a rede, torna-se necessário alterar os valores dos pesos. É a partir deste ponto que a função de custo é aplicada à rede. A função de custo é a responsável por mostrar o quão adequado é o valor do peso atualmente. As funções de custos mais populares incluem a *Mean squared Error (MSE)* e *Cross-entropy Loss*.

- **Mean squared Error (MSE):** a função *MSE* é utilizada quando o problema a ser resolvido é um problema de regressão. A função *MSE* pode calcular o erro quadrático médio, uma métrica de risco que corresponde ao valor esperado de um erro ou perda quadrática. Quanto menor o valor do *MSE*, melhor o modelo usado.

$$MSE = \frac{1}{n} \sum_{i=1}^n (c_i - \hat{c}_i)^2 \quad (3.29)$$

onde c_i corresponde aos valores verdadeiros e \hat{c}_i aos valores pressupostos.

- **Cross-entropy Loss:** mede o desempenho de um modelo de classificação cuja saída é um valor de probabilidade entre 0 e 1. *Cross-entropy loss* aumenta à medida que a probabilidade prevista diverge do valor real. Então, prevendo uma probabilidade de 0,012 quando o valor de observação real é 1, seria ruim e resultaria em um alto valor de perda. Um perfeito modelo teria um *cross-entropy loss* de 0. A Figura 31 mostra a faixa de possíveis valores de perda, dada uma observação verdadeira. Como o previsto, quando a probabilidade se aproxima de 1, a perda de *cross-entropy loss* diminui lentamente. À medida que a probabilidade prevista diminui, no entanto, a *cross-entropy loss* aumenta rapidamente. A *cross-entropy loss* penaliza os dois tipos de erros, mas principalmente as previsões confiáveis e errado.

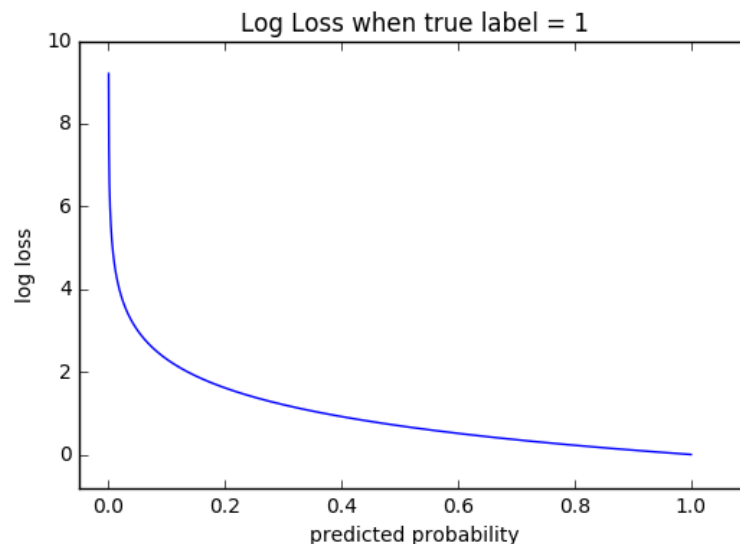


Figura 31 – Gráfico *Cross-entropy Loss*.

Fonte: Nielsen (2019).

Sendo assim, na classificação binária, onde o número de classes M é igual a 2, *cross-entropy loss* pode ser calculada pela Equação 3.30:

$$-(y \log(p) + (1 - y) \log(1 - p)) \quad (3.30)$$

onde M corresponde ao número de classes, y opera como indicador binário (0 ou 1) e p é a observação de probabilidade prevista.

3.3.4.2.1 Validação Cruzada (*K-fold Cross Validation*)

A validação cruzada é um método estatístico usado para estimar o desempenho dos modelos de aprendizado de máquina.

É comumente usado no aprendizado de máquina aplicado para comparar e selecionar um modelo para um determinado problema de modelagem preditiva, porque é fácil de entender, fácil de implementar e resulta em estimativas de predição que geralmente têm um viés menor do que outros métodos.

Conforme James et al. (2013), esta abordagem divide randomicamente o banco de dados de treinamento, em k grupos¹³, em dimensões iguais visando minimizar o problema de variância. A Figura 32 visa descrever o processo de validação cruzada.

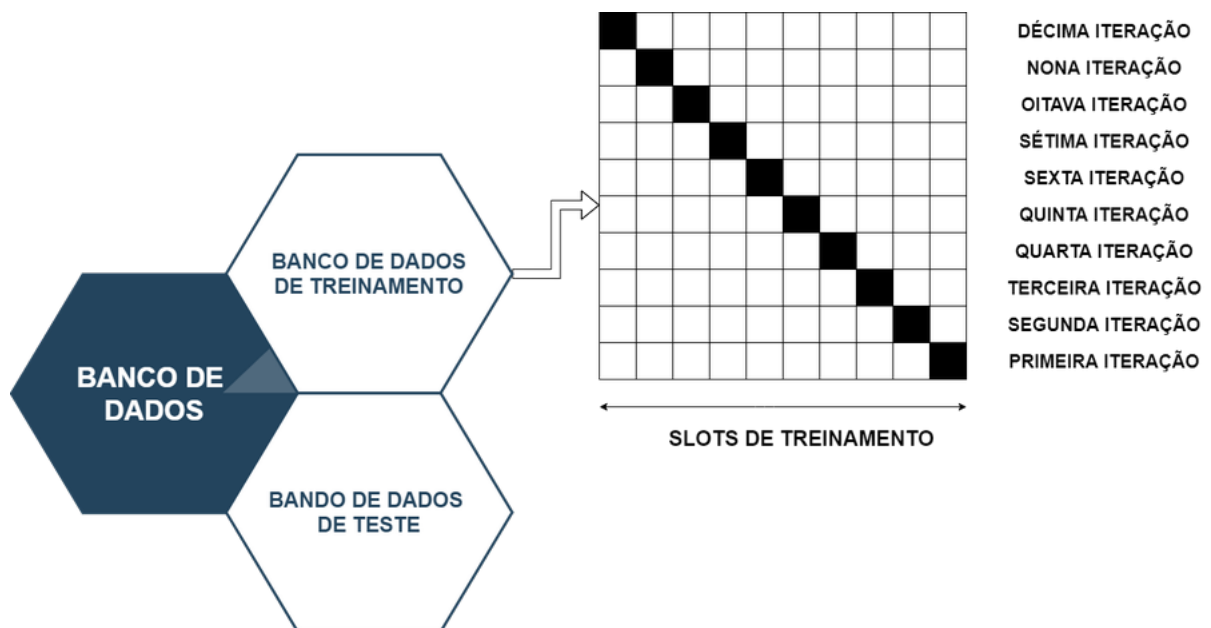


Figura 32 – Validação Cruzada com 10 Agrupamentos.

Fonte: Elaborado pelo autor (2019).

O primeiro agrupamento é tratado como um conjunto de validação e o método é adequado para os grupos $k - 1$ restantes. Logo, o *Mean squared Error (MSE)* do primeiro agrupamento é computado. Este processo é repetido k vezes, e cada vez, cada grupo de treinamento observado é considerado como um novo grupo de validação, portanto, resulta

¹³ Tradução livre para *k-folds*.

em k estimativas do erro de teste *Mean squared Error* ($MSE_1, MSE_2, \dots, MSE_k$). Após as devidas iterações, é estimado a validação cruzada (CV) mediante a média desses valores conforme a Equação 3.31:

$$CV_{(k)} = \frac{1}{k} \sum_{i=1}^k MSE_i \quad (3.31)$$

Contudo, na prática, normalmente é adotado os valores de $k = 5$ e $k = 10$ para CV. Porém, qual vantagem em adotar estes valores para k ao invés de $k = n$? James et al. (2013) explicita que a vantagem é puramente computacional. A validação cruzada é uma abordagem ampla que pode ser aplicada a praticamente qualquer método de aprendizado estatístico. Alguns métodos de aprendizado estatístico possuem procedimentos de ajuste computacionalmente intensivos, portanto, a execução do modelo pode apresentar problemas computacionais, especialmente se n for extremamente alto. Em contrapartida, quando CV é performedo com $k = 10$ requer a adaptação do processo de aprendizagem apenas dez vezes, o que pode ser muito mais viável computacionalmente.

Para problemas de classificação binária CV é calculado conforme Equação 3.32:

$$CV_{(n)} = \frac{1}{n} \sum_{i=1}^n Err_i \quad (3.32)$$

onde $Err_i = I(y_i \neq \hat{y}_i)$.

3.3.4.3 Algoritmo *Backpropagation*

O algoritmo *backpropagation* baseia-se no controle da mudança dos valores dos pesos sinápticos da rede neural, visando minimizar o erro obtido na saída através do aprendizado do vetor de treinamento (entrada-saída). Tendo como principal característica a capacidade de atribuir os erros obtidos na saída da rede às camadas ocultas, o algoritmo *backpropagation* é dado como inovador por possuir tais atributos (HINTON, 2002).

Pode-se dizer que o algoritmo funciona em dois estágios: o primeiro, conforme abordado na seção 3.3.2.1, seria o *forwardpropagation*, onde é realizado o cálculo na saída da rede neural, e o segundo seria o *backpropagation*, onde os erros e os pesos são recalculados, com objetivo de minimizar os erros.

Segundo Hinton (2002), para que seja explicado a implementação do algoritmo *backpropagation*, primeiramente é necessário descrever, novamente, uma rede neural em termos matemáticos. A etapa *forwardpropagation* é regida da seguinte maneira: sabido um exemplo de treinamento representado por (x, d) , como o vetor de entrada x aplicado à camada de entrada de nós da rede, e o vetor saída desejada apresentada à camada de nós de saída da rede. Primeiramente devem ser calculados, conforme Equação 3.33, os campos locais induzidos e os sinais funcionais, prosseguindo através da rede, camada por

camada. O campo local induzido $v_j^{(l)}$ para o neurônio j na camada l é:

$$v_j^{(l)} = \sum_{i=0}^{m_0} w_{ji}^{(l)} \cdot y_i^{(l-1)} \quad (3.33)$$

onde $y_i^{(l-1)}$ é o sinal (função) de saída do neurônio i na camada anterior $l - 1$, e $w_{ji}^{(l)}$ é o peso sináptico do neurônio j na camada l , que é alimentado pelo neurônio i da camada $l - 1$.

Prosseguindo, é calculado, conforme Equação 3.34, o sinal de saída do neurônio j da camada l usando a função de ativação. Normalmente, segundo Hinton (2002), utiliza-se a função *sigmoid* para tal.

$$y_j^{(l)} = \varphi_j \left(v_j^{(l)} \right) \quad (3.34)$$

Logo, se o neurônio j está na primeira camada da rede ($l = 1$), tem-se:

$$y_j^{(0)} = x_j \quad (3.35)$$

onde x_j é o j -ésimo elemento do vetor x . Sendo assim, se o neurônio j estiver posicionado na camada de saída da rede (ou seja, $l = L$, onde L é chamado de *profundidade* da rede), obtem-se:

$$y_j^{(L)} = o_j \quad (3.36)$$

O passo para a frente termina com o cálculo do erro na camada de saída:

$$e_j = d_j - o_j \quad (3.37)$$

O algoritmo *backpropagation* se refere à atribuição de “responsabilidades” pelo erro na saída. Ele se inicia com o cálculo dos gradientes locais em cada camada da rede, definidos por:

- Para o neurônio j na camada de saída L

$$\delta_j^{(l)} = e_j^{(L)} \cdot \varphi_j' \left(v_j^{(L)} \right) \quad (3.38)$$

- Para o neurônio j na camada de saída l

$$\delta_j^{(l)} = \varphi_j' \left(v_j^{(l)} \right)' \cdot \sum_k \delta_k^{(l+1)} \cdot w_{kj}^{(l+1)} \quad (3.39)$$

onde apóstrofe em $\varphi(\cdot)$ representa a diferenciação em relação ao argumento. Os pesos sinápticos da rede são ajustados então com base na Equação 3.40:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta \cdot \delta_j^{(l)}(n) \cdot y_i^{(l-1)}(n) \quad (3.40)$$

onde η é a taxa de aprendizagem.

Os passos de computação *forwardpropagation* e *backpropagation* são iterados sucessivamente, até que algum critério de parada seja alcançado (normalmente número de iterações ou erro préestabelecido).

Embora o algoritmo pareça razoavelmente simples (talvez aí resida sua força), ele está por trás da maior parte das aplicações contemporâneas de redes neurais, provando ser uma ferramenta surpreendentemente poderosa em treinamento de redes com várias camadas.

4 METODOLOGIA DO TRABALHO

Day trade é uma modalidade de investimento dinâmico, cujo fluxo de informação oriundo dos gráficos exige do *trader* excessiva concentração e fator psicológico.

Devido ao fato do pesquisador vivenciar e realizar operações *day trade* no derivativo mini contrato futuro do dólar (WDO) e por já ter estudado a aplicabilidade de diversos indicadores técnicos, se deparou com a necessidade de padronizar as negociações. Fazendo que o fator previsibilidade seja o ponto vital para a aplicação da regra de negociação.

Visando criar bases sólidas para o trabalho, neste capítulo foi elucidado as etapas do trabalho, tais como a análise da regra de negociação e o treinamento da rede neural artificial profunda.

O método *Design Science Research* tem como premissa doze passos que auxiliam, e, de certa forma, guiam o pesquisador no processo da criação de um conteúdo acadêmico consistente. Essa lógica pode ser observada conforme a Figura 33.

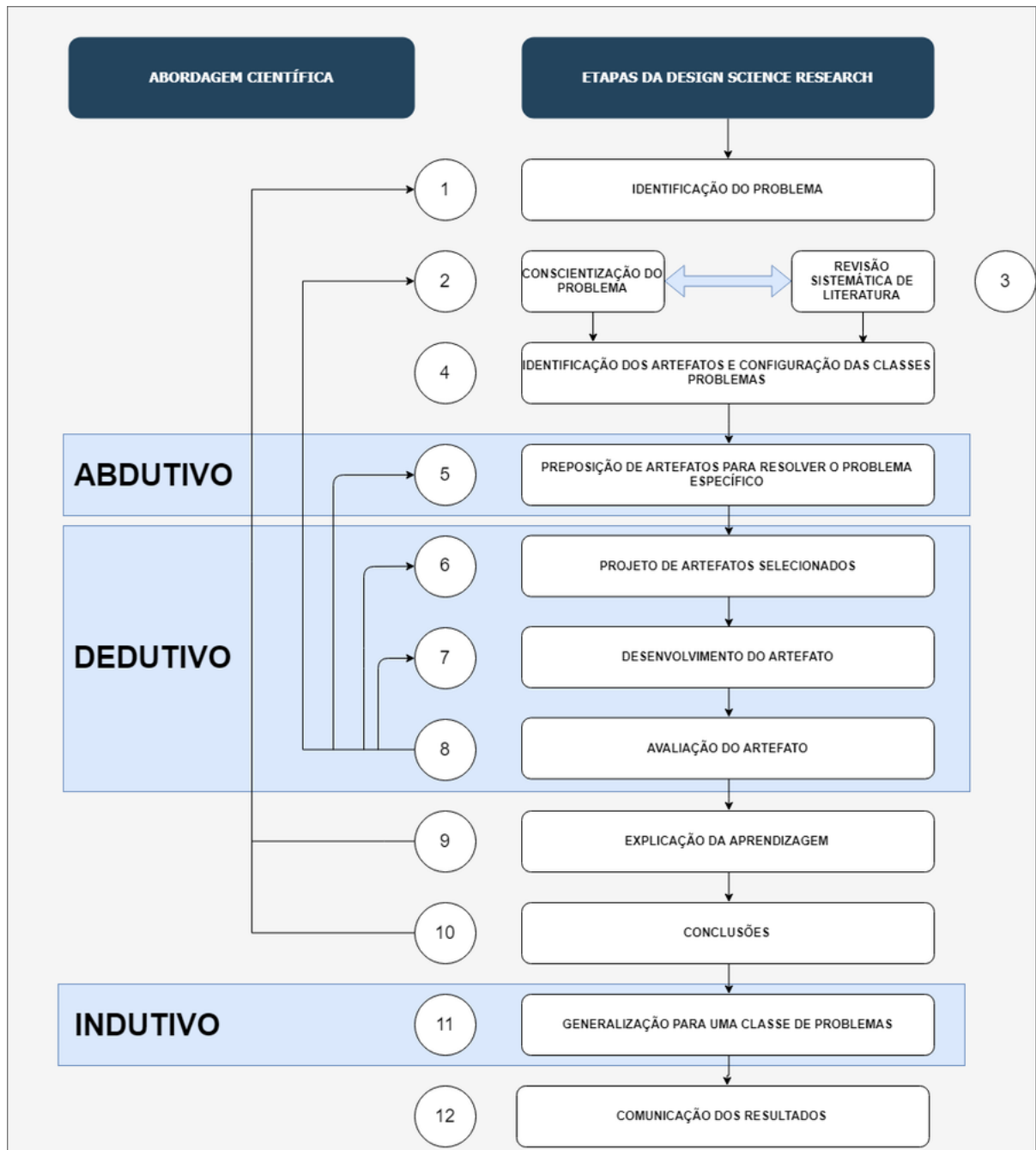


Figura 33 – Método Proposto para a Condução do *Design Science Research*.

Fonte: Dresch, A. et al. (2015).

A primeira etapa do método *Design Science Research* é a identificação do problema. Em relevância, o mesmo tem de ser distinguido e estudado, para que futuramente o pesquisador consiga, e logo, possa elaborar a questão da pesquisa.

No decorrer da compreensão do problema o pesquisador tem que procurar o maior número de informações possíveis, tendo em consideração o funcionalismo do artefato (modelo), além de formalizar os requisitos necessários para o que artefato (modelo) seja capaz de solucionar o problema.

A terceira etapa visa tratar da revisão sistemática da literatura, que deve ser elaborada em conjunto com a compreensão do problema. O empenho empregado nessa etapa auxiliará no melhor entendimento, enquadramento e definição do problema.

Após ter feito uma boa revisão sistêmica de literatura o pesquisador deve ter identificado na literatura os artefatos (modelos) e classe de problemas relacionados com o que ele está tentando resolver, é possível que seja encontrado um artefato (modelo) pronto e ideal. Da mesma forma em que pode encontrar uma classe de problema já estruturada que se enquadre em suas aspirações.

Segundo Dresch, Lacerda e V. Júnior (2015), o artefato é a organização dos componentes do ambiente interno para atingir objetivos em um determinado ambiente externo, porém, ainda não existe uma uniformidade de conceitos em relação ao tipo de artefatos (modelo) gerados a partir da aplicação da *Design Science Research*. A Figura 34 contém uma síntese dos principais autores que tipificam artefatos, extraídos por Dresch, Lacerda e V. Júnior (2015).

Autor	Produtos da <i>design Science research</i>				
Nunamaker Chen e Purdin (1991)	-	-	-	Software	Construção de teorias
Walls, Wydmeyer e Sawy (1992)	-	-	-	-	Teorias do design
March e Smith (1995)	Construção	Modelo	Método	Instanciação	-
Purao (2002)	Princípios operacionais	-	-	Artefato	Teorias emergente
Van Aken (2004)	-	-	-	-	Conhecimento em <i>design</i>
Venable (2006)	Parte de uma Solução tecnológica	Parte de uma Solução tecnológica	Parte de uma Solução tecnológica	Sistema de base computacional	Teoria do <i>design</i>
Gregor e Jones (2007)	Componente de uma teoria do <i>Design</i>	Componente de uma teoria do <i>Design</i>	Componente de uma teoria do <i>Design</i>	Componente de uma teoria do <i>Design</i>	Teoria do <i>design</i>

Figura 34 – Produtos da *Design Science Research*.

Fonte: Altyki, Gable e Bandara apud Dresch et al, (2015).

Este estudo irá considerar a tipificação de artefato dada por Dresch, Lacerda e V. Júnior (2015) que se baseia nos conceitos de March e Smith (1995), adicionando um quinto tipo que são as teorias fundamentadas em '*design proposition*'. As cinco categorias de artefatos estão exemplificados na Figura 35.

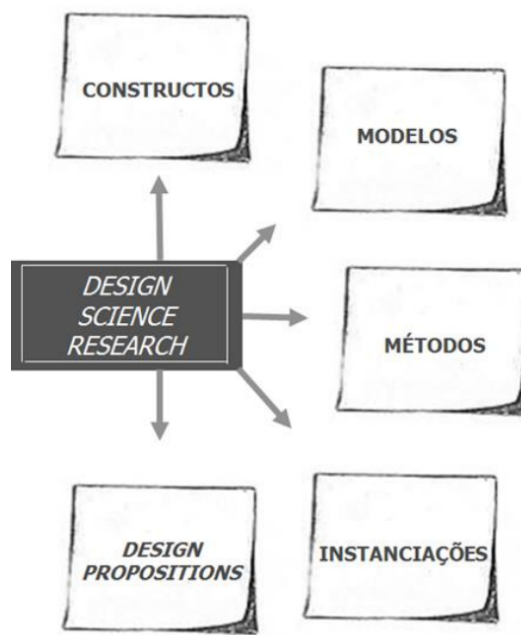


Figura 35 – Produtos da *Design Science Research* (artefatos).

Fonte: Dresch et al, (2015).

A Tabela 8 visa representar um empenho em transpor por outra forma os conceitos chaves que orientam cada modelo de artefato apresentado na Figura 35.

Tabela 8 – Artefatos e Conceitos Chaves.

ARTEFATOS	CONCEITO CHAVE
Constructor	Fundamentação Teórica.
Modelos	Parâmetros para organização dos dados (planilhas, etc.)
Métodos	Aplicação do modelo
Instanciações	Novas concepções
<i>Design Proposition</i>	Organização de uma solução através do “design” – como um software

Fonte: Elaborado pelo autor (2019).

Identificar artefatos existentes ajuda o pesquisador a ser mais assertivo na proposição de novos artefatos para as classes de problema inerente ao objeto de estudo, que foi feita na quinta etapa onde acontece a proposição de artefatos para a resolução do problema.

Após propor novos artefatos, na sexta etapa o pesquisador irá realizar o projeto do artefato, nesse momento deve ser informado o desempenho esperado que vá garantir a solução satisfatória para o problema garantindo o rigor da pesquisa, permitindo que possa ser replicada e confirmada posteriormente por outros pesquisadores.

Na sétima etapa acontece o desenvolvimento do artefato, onde se pode encontrar em seu estado funcional, ou a partir dos princípios da heurística de construção¹, que pode ser formalizada a partir do desenvolvimento do artefato.

A avaliação do artefato é a próxima etapa e pode ser conduzida em ambiente experimental ou em um contexto real. As saídas resultantes da etapa de avaliação são o artefato devidamente avaliado e a formalização das heurísticas contingenciais².

Na nona etapa, depois da avaliação do artefato e considerando-se que os resultados foram satisfatórios, é essencial que o pesquisador faça a explicitação das aprendizagens obtidas, assegurando que a pesquisa realizada possa servir de referência para a geração de conhecimentos tanto no campo prático como no teórico.

Posteriormente a explicitação das aprendizagens, na décima etapa, o pesquisador formaliza a conclusão, expondo os resultados obtidos na pesquisa apontando inclusive suas limitações.

Ao término da pesquisa, o artefato desenvolvido deve ser generalizado para uma classe de problemas, permitindo que haja o avanço do conhecimento em *Design Science Research*.

Na última etapa o pesquisador deve fazer a comunicação dos resultados em veículos científicos, com o intuito de atingir o maior número de pessoas possíveis interessados na temática, tanto na academia quanto nas organizações.

Por fim, a Figura 36 ilustra os capítulos da dissertação de acordo com as doze principais etapas retratadas pela metodologia *Design Science Research*.

¹ É a construção do artefato em si. Segundo Dresch, Lacerda e V. Júnior (2015), as heurísticas de construção definem quais os requisitos necessários para o funcionamento adequado do ambiente interno do artefato, com vistas do ambiente externo.

² De acordo com Dresch, Lacerda e V. Júnior (2015), as heurísticas contingenciais, é um conhecimento fundamental que explica os limites do artefato, quais são as suas condições de utilização e em que situações ele será útil.

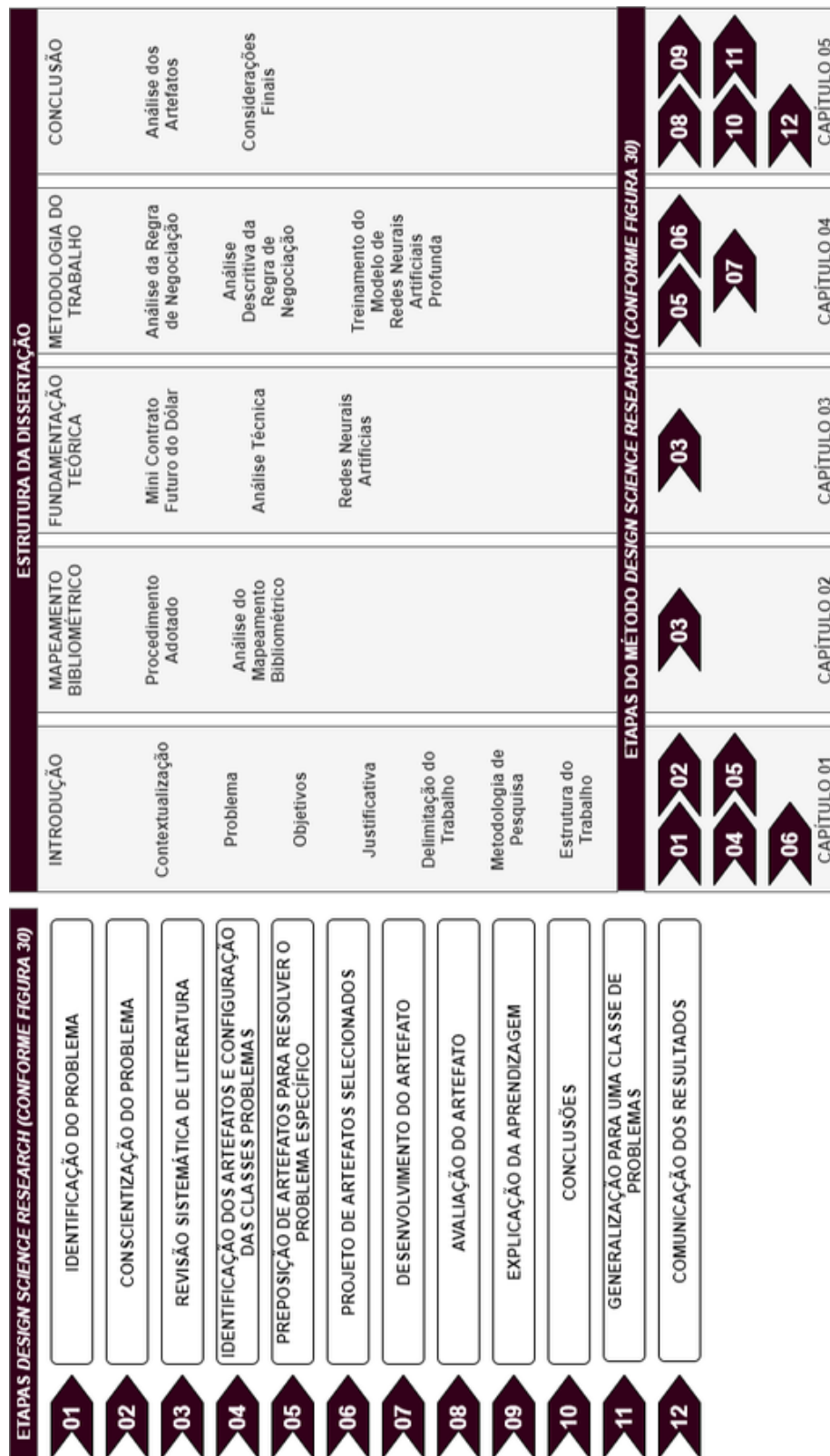


Figura 36 – Estrutura da dissertação conforme metodologia Design Science Research.

Fonte: Elaborado pelo autor (2019).

4.1 ANÁLISE DA REGRA DE NEGOCIAÇÃO

Como as operações de *day trade* são extremamente dinâmicas e ocorrem ao longo do dia, a liquidez dos ativos é de fundamental importância nas execuções das estratégias. Papéis que apresentam baixa liquidez ou elevado *spread*³ entre as ofertas de compra e venda não são favoráveis para estas operações.

Baseado nesta premissa, dentre os principais ativos listados na bolsa de valores de São Paulo - B3 - o mini contrato futuro do dólar é um dos derivativos financeiros que apresenta elevada liquidez diária.

Neste capítulo é explicitado as premissas para a obtenção e a análise exploratória dos dados obtidos a partir da ativação da regra de negociação visando organizar as etapas para o treinamento e validação do modelo de redes neurais artificiais profunda.

4.1.1 Levantamento dos Dados

Para realizar o levantamento dos dados da regra de negociação de cruzamento de médias móveis foi utilizado o formato de *backtesting*, que constitui em aplicar a estratégia em dados históricos disponíveis nas plataformas de negociação. Para realizar o *backtesting* foi utilizada a plataforma de negociação Metatrader 5, que possui a ferramenta de análise de automatização de operações (robô de negociação). Para tal, foi desenvolvido um código “mql5” com todos os parâmetros de entrada e saída das operações conforme as premissas adotadas para a regra de negociação testada. Este modelo de *backtesting* com o código de programação permitiu que a condição de automatização da análise dos pontos de entrada e saída das operações fosse realizado eliminando a análise manual. Após a automatização das operações temos um relatório com todas as entradas e saídas realizadas, por conseguinte, os valores finais das variáveis que foram utilizadas no banco de dados.⁴

Para a execução da regra de negociação foram utilizados os períodos gráficos de 02 e 05 minutos para cada candlestick. Sendo que o gráfico de 02 minutos é período onde as entradas e saídas da regra de negociação são aplicadas, extraído do gráfico de 05 minutos mais informações complementares para a análise final. Sendo assim, foram reportadas na Tabela 9 as variáveis utilizadas no estudo e na concepção do banco de dados que foram analisadas posteriormente pelo modelo de redes neurais artificiais profunda. Também foi mostrado na Tabela 9 os períodos gráficos utilizados para a coleta das variáveis. No total, o banco de dados é composto por 22 variáveis.

³ É o termo usado para medir a diferença entre o melhor preço de compra e o melhor preço de venda de um ativo. O *spread* acaba sendo uma medida de liquidez e de eficiência do mercado já que quanto menor o *spread*, maior a disposição dos players em negociar.

Fonte: <https://scalpertrader.com.br/spread/>

⁴ As únicas variáveis independentes que não foram possíveis serem coletadas pelo robô de negociação foram os 04 bancos e as 04 corretoras. Para a coleta destas, foi utilizado outra plataforma de negociação conhecida como *Tryd*.

Tabela 9 – Tabela com as variáveis coletadas a partir do *backtesting* da regra de negociação.

Variáveis	Descrição	Período Gráfico
Data	Dia e o mês do pregão onde ocorreram as entradas e saídas da regra de negociação.	02 minutos.
Horário	Horário onde ocorreram as entradas e saídas da regra de negociação.	02 minutos.
Tipo de negociação	Compra ou venda.	02 minutos.
Preços	<p>Valores dos preços quando acionada a regra de negociação. Foram utilizados os seguintes preços no estudo:</p> <ul style="list-style-type: none"> • Abertura; • Fechamento; • Máxima; • Mínima. 	02 minutos.
Médias móveis	Médias geradas a partir dos preços colhidos. Foram utilizadas as seguintes médias no estudo:	02 e 05 minutos.
	<div> <div> 1) Média aritmética: <ul style="list-style-type: none"> • Média aritmética de 20 períodos das máximas e mínimas dos preços. </div> <div> 1) Média exponencial: <ul style="list-style-type: none"> • Média exponencial de 9 períodos do preço de fechamento. </div> </div>	02 e 05 minutos.
Volume financeiro	Volume financeiro negociado no momento em que ocorreu a regra de negociação.	02 minutos.

Variáveis	Descrição	Período Gráfico
Corretoras	As 04 corretoras que tiveram o maior volume financeiro negociado no mini contrato futuro do dólar. No instante em que a regra de negociação foi ativada, foi verificado se as corretoras estavam contra ou a favor da entrada. Exemplo, caso a regra de negociação sinalizou uma entrada de compra, estas 04 corretoras estavam comprando também ou vendendo?	02 minutos.
Bancos	Os 04 bancos que tiveram o maior volume financeiro negociado no mini contrato futuro do dólar. No instante em que a regra de negociação foi ativada, foi verificado se os bancos estavam contra ou a favor da entrada. Exemplo, caso a regra de negociação sinalizou uma entrada de compra, estes 04 bancos estavam comprando também ou vendendo?	02 minutos.
Percentual de compra e venda	Contabilizado a ação do mercado no instante da entrada da regra de negociação. Se o mesmo estava com viés comprador ou vendedor.	02 minutos.
Pontos avançados	Após entrada da regra de negociação, foi verificado quantos pontos o mercado movimentou-se a favor. Variável que definiu os 2,5 de <i>ticks</i> para operações vencedoras descritas na Tabela 11.	02 minutos.
Pontos retornados	Após entrada da regra de negociação, caso o mercado não movimentasse a favor da entrada, foi verificado quantos pontos o mercado movimentou-se contra. Variável que definiu a premissa para as operações perdedoras descritas na Tabela 11.	02 minutos.
Amplitude	Diferença entre a máxima e a mínima do <i>candlestick</i> no instante da entrada da regra de negociação.	02 minutos.

Fonte: Elaborado pelo autor (2019).

A Tabela 10 apresenta o cenário resumido onde foi realizada as operações de entrada e saída no *backtesting*.

Tabela 10 – Resumo da estratégia.

Regra de Negociação	Operação	Período de Execução da Regra	Período Complementar da Regra
---------------------	----------	------------------------------	-------------------------------

Regra de Negociação	Operação	Período de Execução da Regra	Período Complementar da Regra
Cruzamento de Médias	Compra e Venda	Dois minutos	Cinco minutos

Fonte: Elaborado pelo autor (2019).

A época adotada para o estudo foi no período de 03/06/2019 a 27/12/2019, que totalizam 970 entradas e saídas da regra de negociação. Contudo, as premissas para execução da estratégia estão apresentadas na Tabela 11. Com relação à escolha das operações ganhadoras e perdedoras, em caso de execução da regra de negociação, foi utilizada os dados obtidos a partir da análise descritiva. Mais adiante, na Seção 4.2, foi abordado o meio de obtenção e definição das variáveis que foram utilizadas no modelo..

Tabela 11 – Premissas para a execução da regra de negociação.

Regra de negociação	Cruzamento de médias
Descrição da regra de negociação	Média exponencial cruzar média aritmética.
Horário inicial da execução	09:30
Sentido	Ponta compradora e vendedora.
Entrada na operação	Imediatamente após o fechamento do <i>candlestick</i> que gerou o cruzamento da média móvel.
Operação ganhadora	Diferença entre o preço de abertura da operação com o preço atual gerar o valor de <i>tick</i> de 2,5 pontos
Operação perdedora	Diferença entre o preço de abertura da operação com o preço atual não gerar o valor de <i>tick</i> de 2,5 pontos

Fonte: Elaborado pelo autor (2019).

4.1.2 Execução da Regra de Negociação

A Regra de Negociação é definida da seguinte forma: quando a média curta é maior que a média longa, caracteriza-se compra. Contudo, a média curta é ponderada a partir da média exponencial de 09 períodos do preço de fechamento e a média longa é ponderada a partir da média aritmética das máximas alcançada pelo preço. Por outro lado, no sentido inverso, quando a média curta é menor que a média longa, caracteriza-se venda. Porém, a média curta é ponderada a partir da média exponencial de 09 períodos do preço de fechamento e a média longa é ponderada a partir da média aritmética de 20 períodos das mínimas alcançadas pelo preço.

- Regra de negociação cruzamento de média para compra (Figura 37):

$$MME_{n_t}(C_t, 9, 0) > MMS_{n_t}(M_t, 20, 0) \quad (4.1)$$

onde MME é a média móvel exponencial, 9 corresponde a n períodos estipulados (dias, semanas, minutos), e C_t é o preço de fechamento. MMS_{n_t} é a média móvel simples, 20 corresponde a n períodos estipulados (dias, semanas, minutos), e M_t corresponde as máximas alcançadas pelo preço.

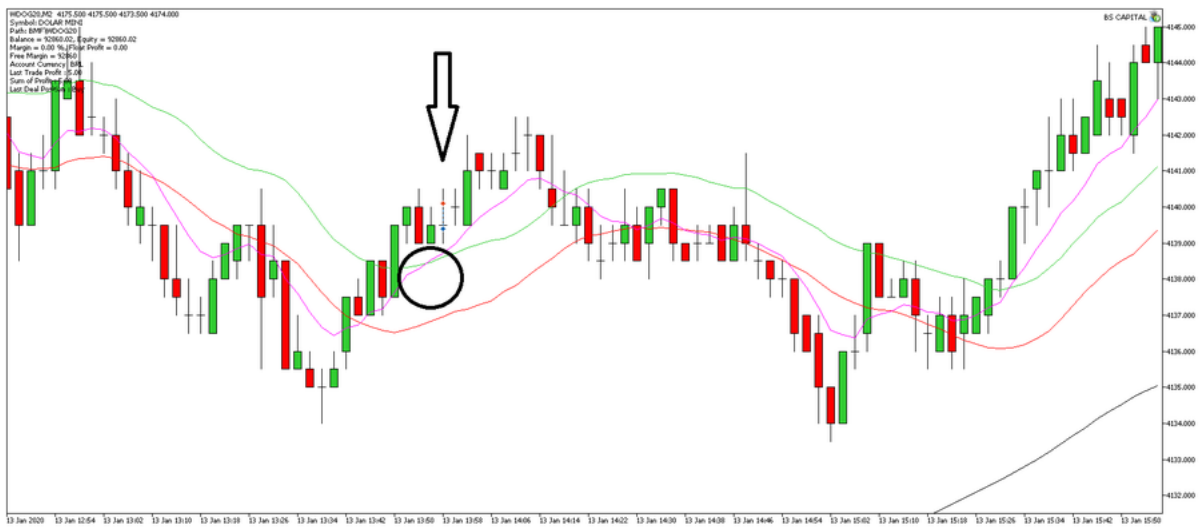


Figura 37 – Regra de negociação para a compra.

Fonte: Metatrader 5 (2019).

- Regra de negociação cruzamento de média para venda (Figura 38):

$$MME_{n_t}(C_t, 9, 0) < MMS_{n_t}(M_t, 20, 0) \quad (4.2)$$

onde MME é a média móvel exponencial, 9 corresponde a n períodos estipulados (dias, semanas, minutos), e C_t é o preço de fechamento. MMS_{n_t} é a média móvel simples, 20 corresponde a n períodos estipulados (dias, semanas, minutos), e M_t corresponde as máximas alcançadas pelo preço.

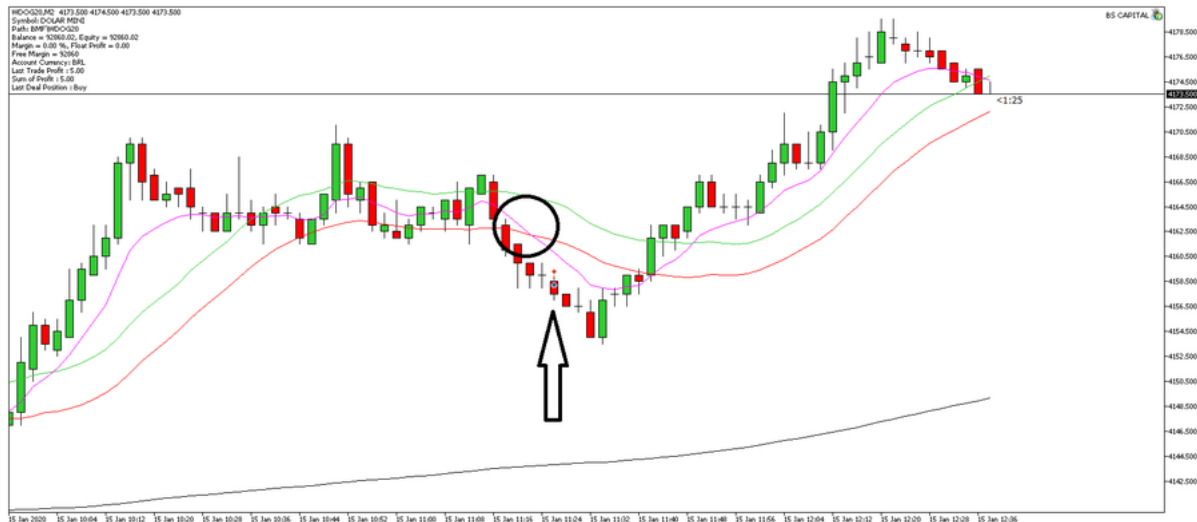


Figura 38 – Regra de negociação para a venda.

Fonte: Metatrader 5 (2019).

4.1.3 Matriz de Confusão (*Confusion Matrix*)

A principal tarefa deste trabalho é fazer a classificação binária das entradas e saídas realizadas pela regra de negociação de cruzamento de médias móveis que atenderam as premissas estipuladas na Tabela 12. Logo, no instante que se adota a classificação binária ao invés da regressão linear, a saída do nosso modelo apenas consiste em dois valores - um ou zero⁵.

Uma matriz de confusão é uma tabela frequentemente usada para descrever o desempenho de um modelo de classificação em um conjunto de dados de teste pelos quais os valores verdadeiros são conhecidos.

Tabela 12 – Tabela da matriz confusão (*confusion matrix*).

Classe Atual	Classe de Predição	
	Positivo	Negativo
Positivo	Verdadeiro positivo (VP)	Falso negativo (FN)
Negativo	Falso positivo (FP)	Verdadeiro negativo (VN)

Fonte: Elaborado pelo autor (2019).

- Verdadeiro positivo: estes são os valores positivos previstos corretamente, o que significa que o valor da classe atual é sim e o valor da classe de predição também é sim.

⁵ Ao finalizar a operação de compra ou de venda oriunda da regra de negociação com lucro, a variável independente é representada pelo valor um (*gain*). Por outro lado, quando a operação deu prejuízo, é representado por zero (*loss*).

- Falso positivo: quando a classe atual é não e a classe prevista é sim.
- Verdadeiro negativo: esses são os valores negativos previstos corretamente, o que significa que o valor da classe atual é não e o valor da classe de predição também é não.
- Falso negativo: quando a classe atual é sim, mas a classe de predição é não.

A partir dessas quatro classes de predição, pode-se calcular a acurácia (A), a precisão (P), *recall* (R) e a pontuação F1 (F_ω).

- **Acurácia:** para validar desempenho do modelo de redes neurais artificiais que desempenhará o papel a classificador binário na camada de saída, adota-se a acurácia (A) como validador. Fornecido um conjunto de dados de teste, a acurácia (A) é a razão entre o número de amostras classificadas corretamente pelo classificador e pelo número total da amostra. Pode-se pensar que, se tivermos alta acurácia (A), nosso modelo será o melhor. Contudo, a acurácia (A) é um ótimo validador, mas somente quando tem conjuntos de dados simétricos em que os valores de falso positivo (FP) e falso negativo (FN) são quase os mesmos. A acurácia (A) é definida pela seguinte equação:

$$A = \frac{VP + VN}{VP + VN + FP + FN} \quad (4.3)$$

Segundo Chen e He (2018), a acurácia (A) está longe de ser suficiente por isso é proposto outros três índices validadores para ajudar a fazer as avaliações do modelo. Para desempenhar esta tarefa, é utilizada a precisão (P), *recall* (R) e pontuação F1 (F_ω).

- **Precisão:** precisão (P) é a razão entre as observações positivas previstas corretamente e o total de observações positivas previstas. Alta precisão (P) refere-se à baixa taxa de falsos positivos.

$$P = \frac{VP}{VP + FP} \quad (4.4)$$

- **Recall:** *recall* (R) é a proporção de observações positivas previstas corretamente e todas as observações na classe atual - sim.

$$R = \frac{VP}{VP + FN} \quad (4.5)$$

- **Pontuação:** pontuação F1 (F_w) é a média ponderada de precisão (P) e *recall* (R). Portanto, essa pontuação leva em consideração tanto os falsos positivos quanto os falsos negativos. A precisão (P) funciona melhor se os falsos positivos e falsos negativos tiverem um custo semelhante. Se o custo de falsos positivos e falsos negativos for muito diferente, é melhor olhar para precisão (P) e *Recall* (R).

$$F_w = 2 * \frac{P \times R}{P + R} \quad (4.6)$$

De modo geral, quando o valor do *recall* (R) é muito alto, o valor da precisão (P) geralmente é baixo e vice-versa. Por isso, para julgar com propriedade qual o algoritmo é melhor para o conjunto de dados analisado, é bom, na prática, introduzir pontuação F1 (F_w).

4.2 ANÁLISE DESCRITIVA DA REGRA DE NEGOCIAÇÃO

Nesta etapa, foi verificado a singularidade e homogeneidade das variáveis de entrada que compuseram o banco de dados após análise do histórico das entrada e saídas obtidas no *backtesting* referente ao período de 03/06/2019 à 27/12/2019.

Durante o período analisado (03/06/2019 a 27/12/2019), no *backtesting* foram contabilizadas 970 entradas e saídas a partir da regra de negociação. Apresentando um total de 489 (50.46%) entradas para a posição compradora e 480 (49.54%) para a posição vendedora.

Com relação aos dias, a regra de negociação apresentou um equilíbrio (Figura 39), independentemente se nesta data ocorreu algum evento econômico de relevância ou alguma notícia - positiva ou negativa - que influenciasse o mercado.

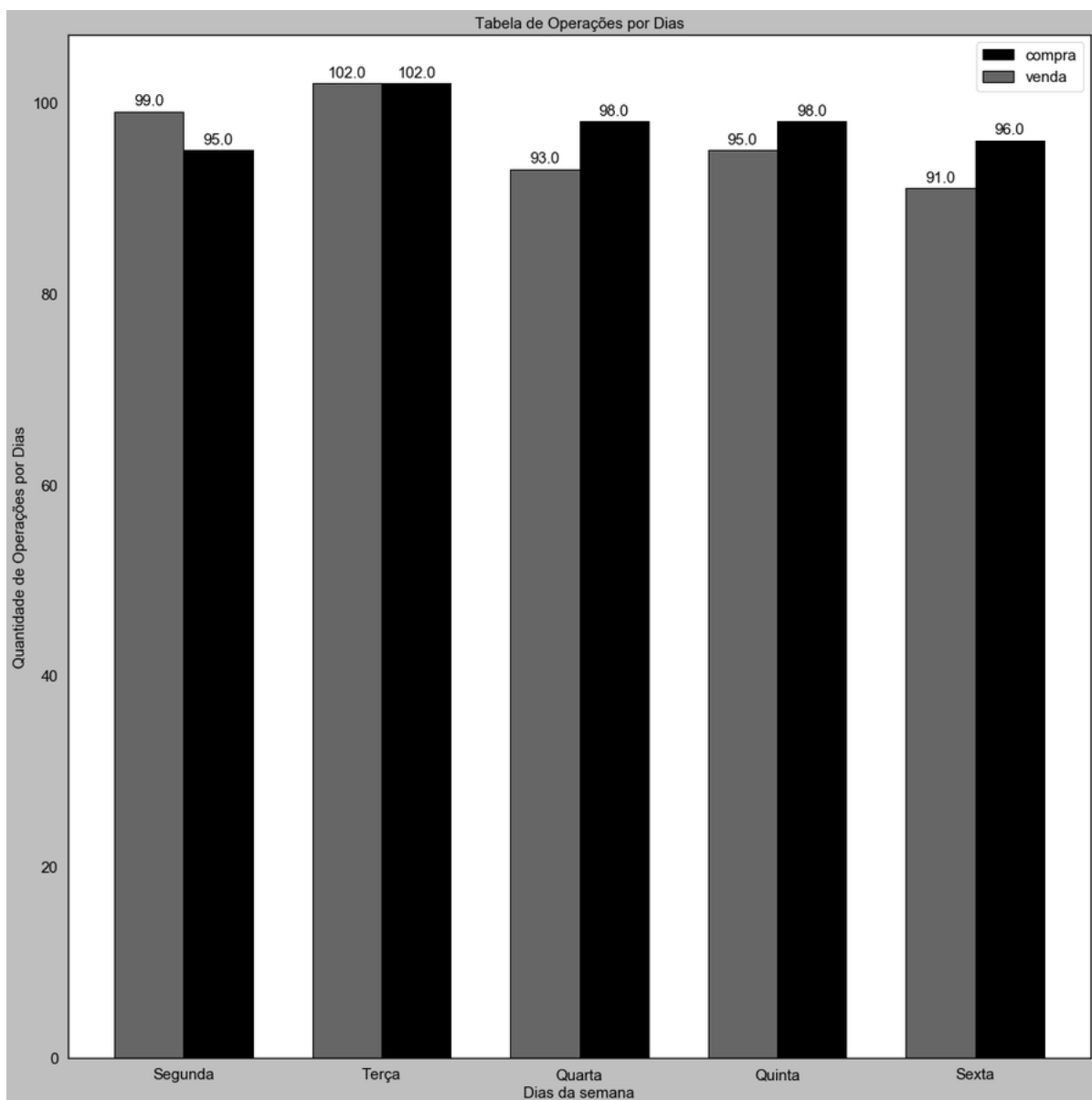


Figura 39 – Gráfico das operações por dias da semana.

Fonte: Elaborado pelo autor (2019).

Em relação aos meses, é possível notar a mesma estabilidade de números de entradas e saídas da regra de negociação. Fica evidente nos valores gerados pelo *backtesting* uma simetria na quantidade de entradas e saídas entre junho e outubro. Ocorrendo uma diminuição das entradas nos meses de novembro e dezembro proveniente aos feriados conforme Figura 40.

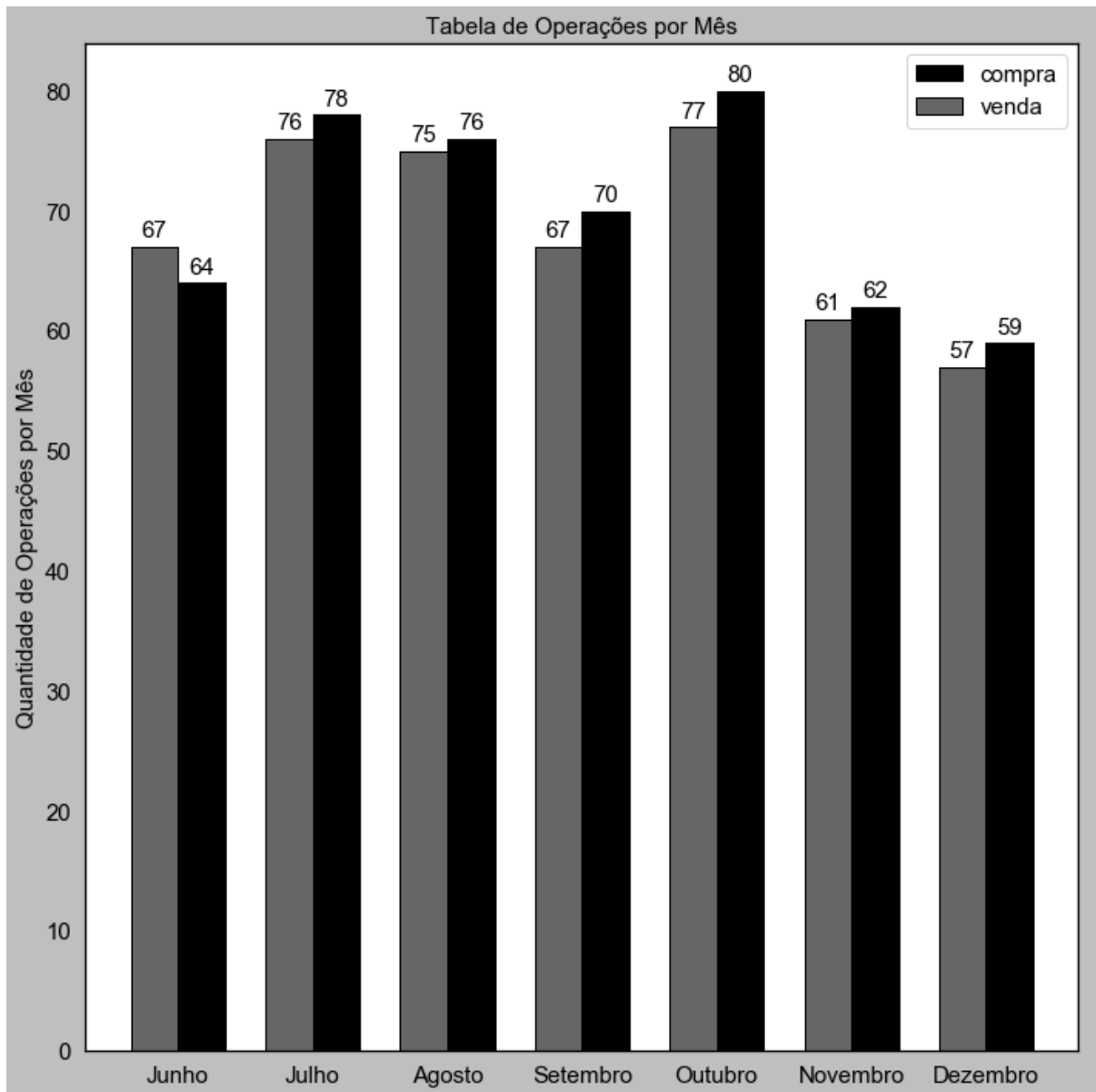


Figura 40 – Gráfico das operações por meses.

Fonte: Elaborada pelo autor (2019).

A regra de negociação obteve mais frequência de entradas e saídas entre as 09:00 e 10:00 conforme Figura 41. Possivelmente esta alta frequência é proveniente à abertura dos mercados internacionais e da bolsa de valores brasileiras.

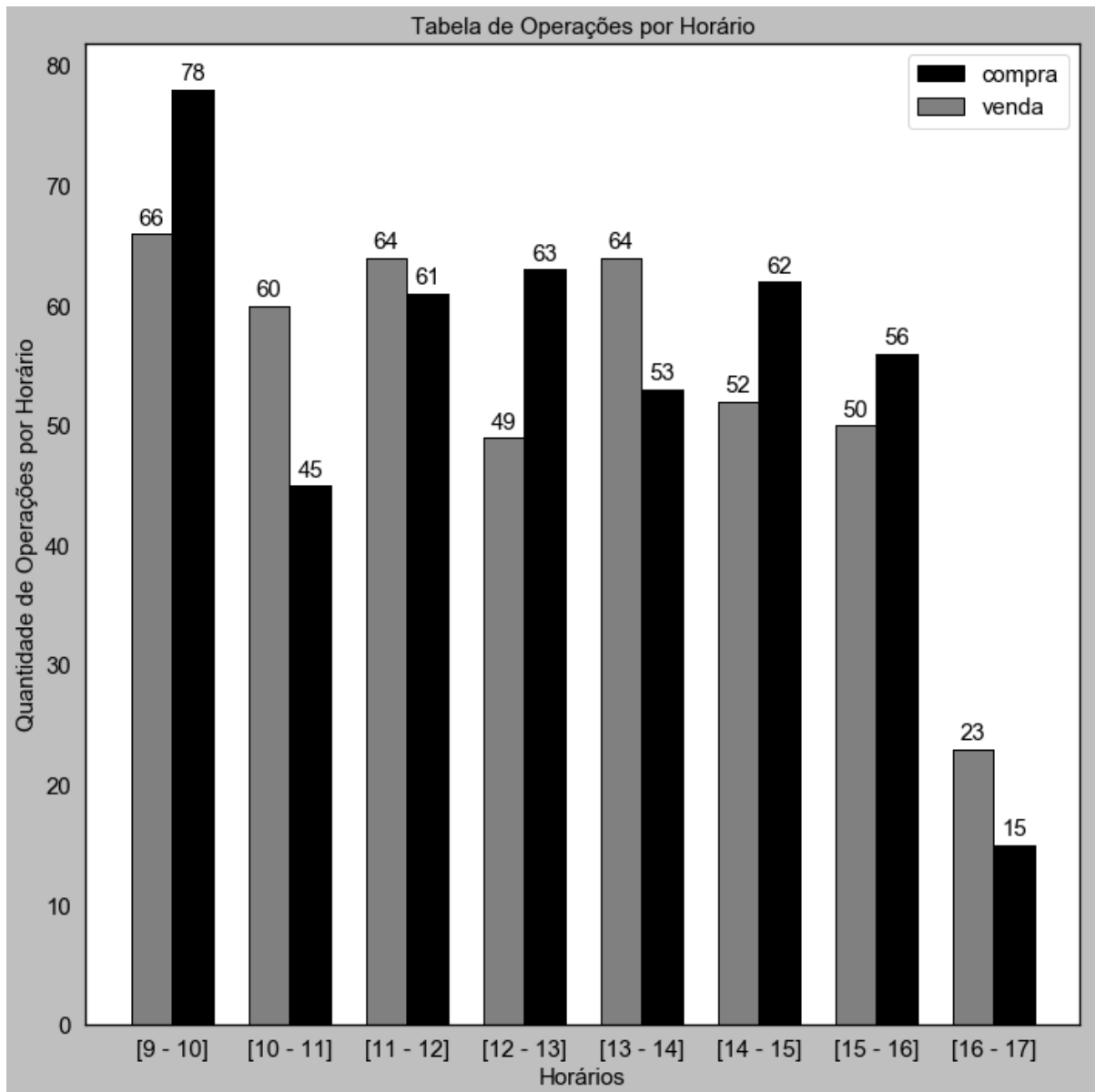


Figura 41 – Gráfico de operações por horário.

Fonte: Elaborado pelo autor (2019).

Com relação aos pontos avançados, conforme Figura 42, nota-se que quando ocorre a regra de negociação, em geral, a operação avançou entre 0 a 4 pontos. O pico no agrupamento 0-4 demonstra o quão comum são estes valores. Corroborado pelo fato da assimetria no histograma à direita demonstrando que a maioria das entradas e saídas avançam relativamente em poucos pontos.

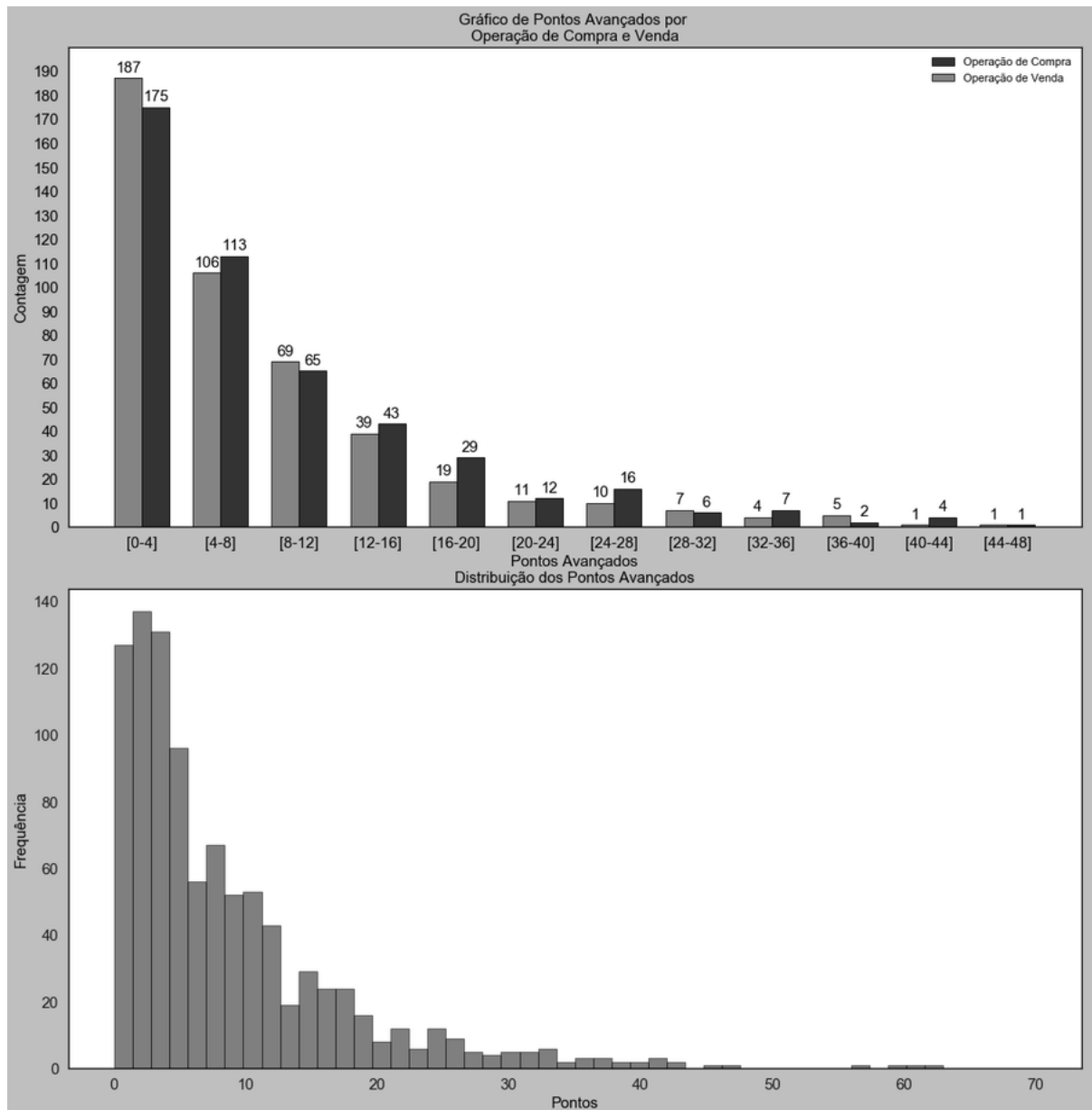


Figura 42 – Gráfico de barras e histograma dos pontos avançados.

Fonte: Elaborado pelo autor (2019).

Consequentemente, tornou-se possível definir com quantos pontos avançados necessitaria para se ter uma operação vencedora a partir da regra de negociação. Para este estudo, como houve uma maior frequência de pontos avançados no agrupamento 0-4, adotou-se a média deste agrupamento conforme Figura 43.

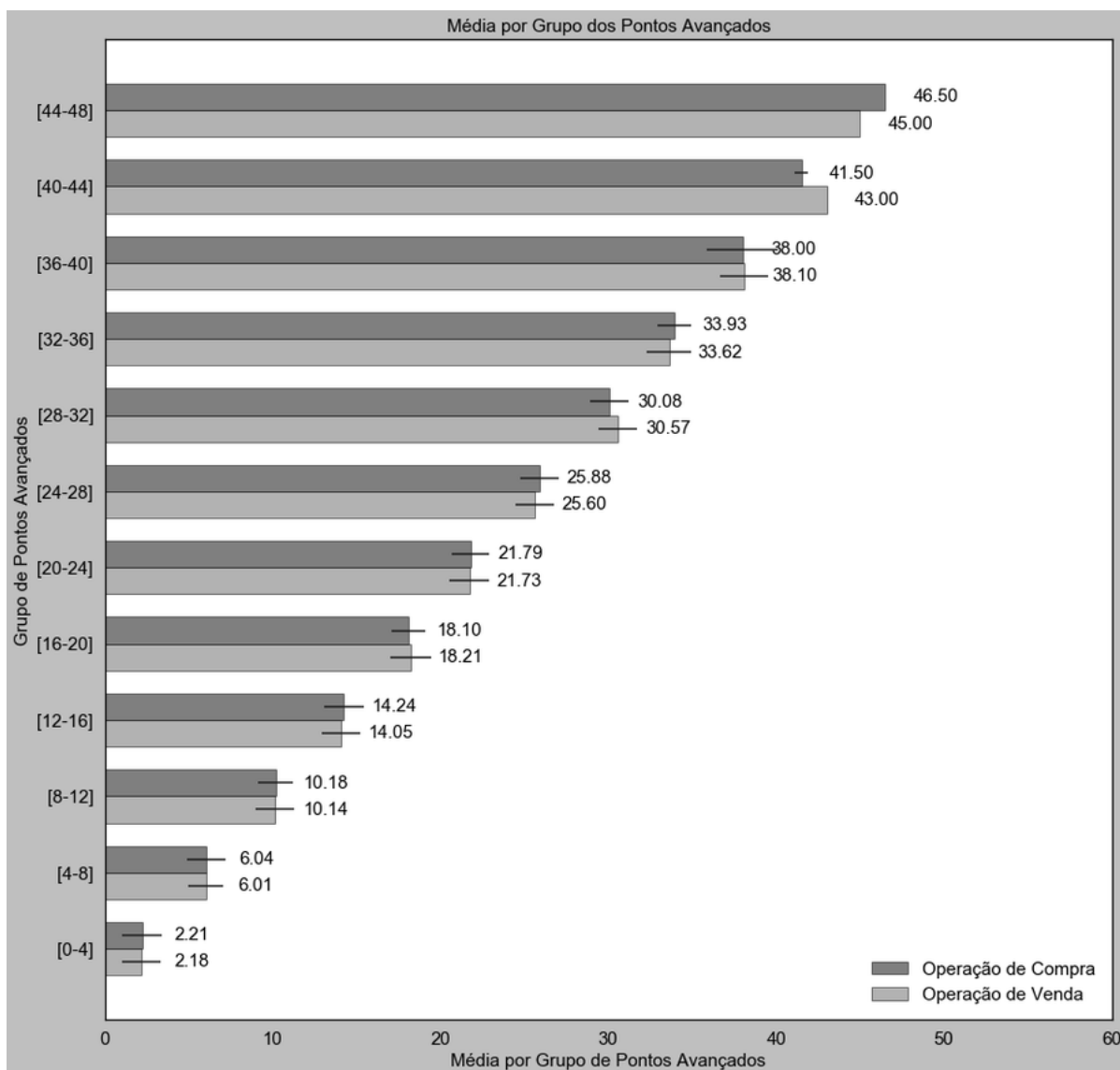


Figura 43 – Gráfico das médias por grupo dos pontos avançados.

Fonte: Elaborado pelo autor (2019).

Como o valor do *tick* do mini contrato futuro do dólar é de 0,5, foi necessário adotar 2,5 pontos para se ter uma operação vencedora conforme mencionado na Tabela 7. Adotando 2,5 pontos para as operações vencedoras a partir do banco de dados analisado, obteve-se 750 operações vencedoras. Foi averiguado que as 750 operações vencedoras correspondem a 77,4% das operações coletadas. E 219 operações perdedoras, representam 22.60% das operações coletadas.

Os bancos e corretoras exercem um papel fundamental na liquidez e direcionamento do mercado. Quanto aos quatro bancos e corretoras selecionadas para o estudo, baseado no volume financeiro negociado por tais instituições, notou-se que as quatro corretoras, durante o período analisado tanto nas operações de compra como nas operações de venda, na maioria das entradas atuavam contra tendência em relação à regra de negociação. Por

assim dizer, quando ativada uma entrada para uma operação de compra, na maior parte do tempo as corretoras estavam vendendo e vice-versa (Figura 44 e Figura 45).

Dos quatro bancos analisados, três bancos estavam acima de 50% de acerto em relação da tendência da regra de negociação (Figura 44 e Figura 45).

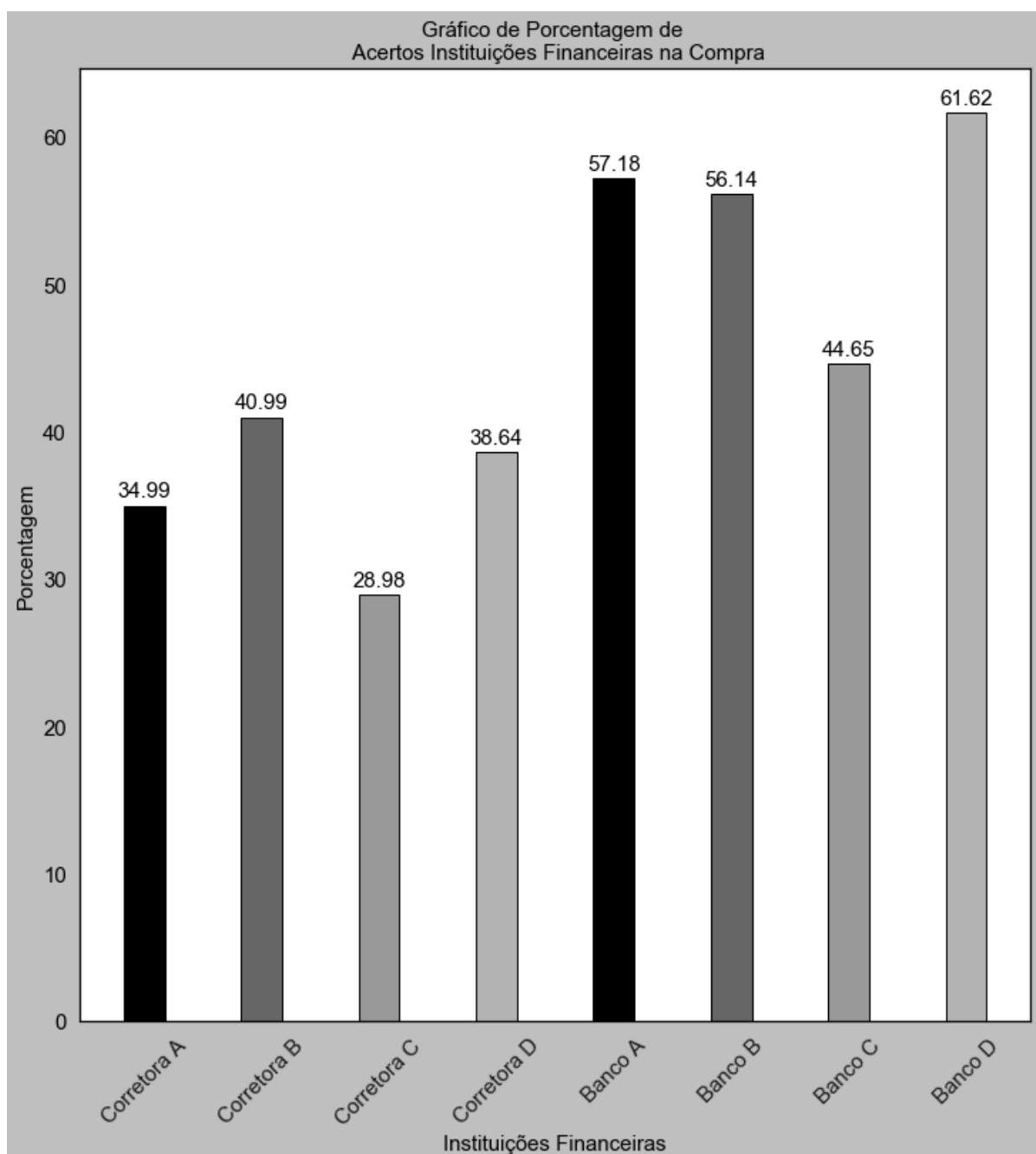


Figura 44 – Gráfico demonstrando a relação das instituições financeiras com as entradas no sentido da compra em relação à regra de negociação.

Fonte: Elaborado pelo autor (2019).

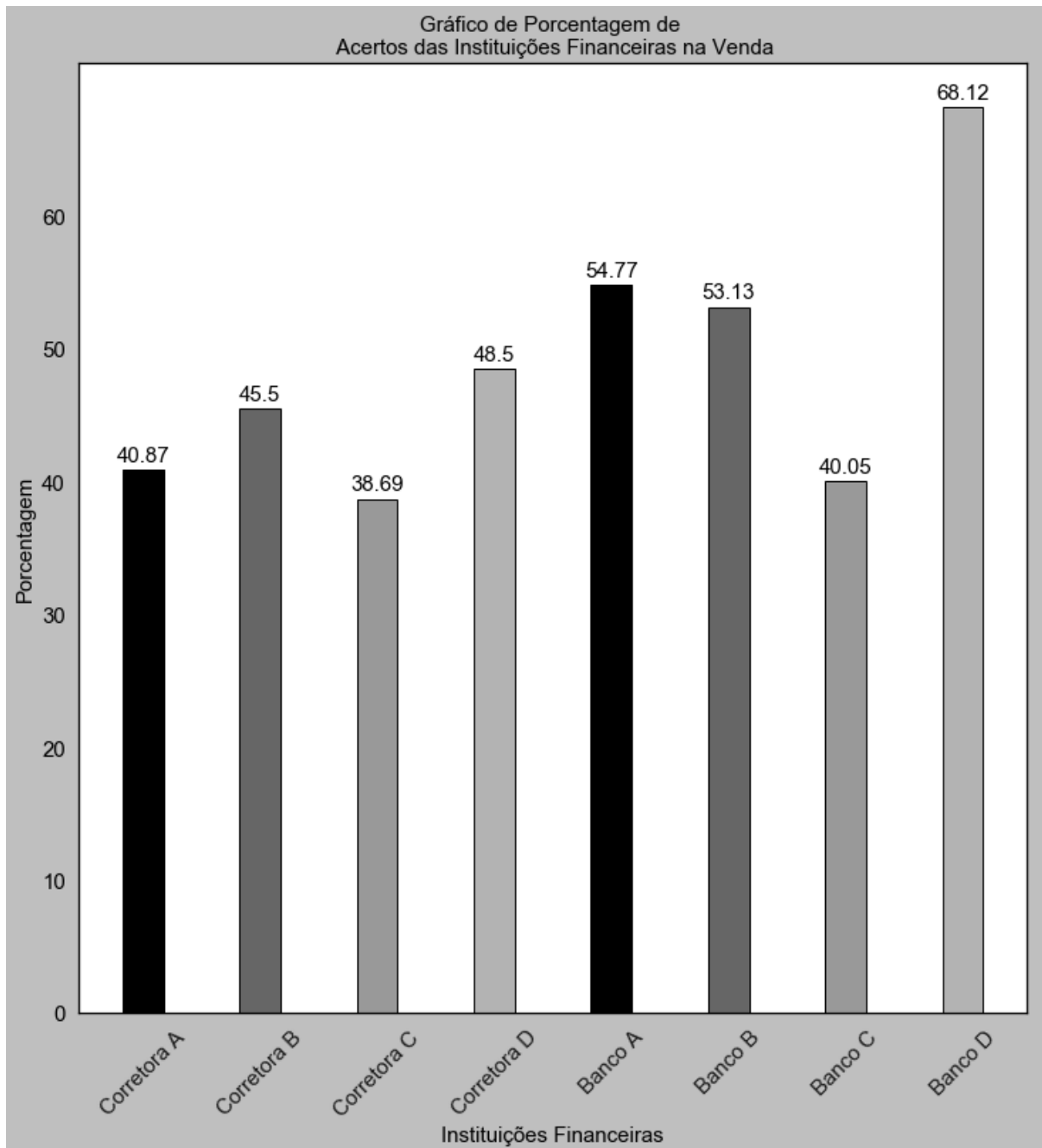


Figura 45 – Gráfico demonstrando a relação das instituições financeiras com as entradas no sentido da venda em relação à regra de negociação.

Fonte: Elaborado pelo autor (2019).

Empregando o coeficiente de correlação Pearson, foi possível verificar quais das variáveis de entrada teriam uma correlação com a variável de saída.

Sendo assim, a Figura 46 demonstra a correlação entre as variáveis de entrada com a variável de saída.

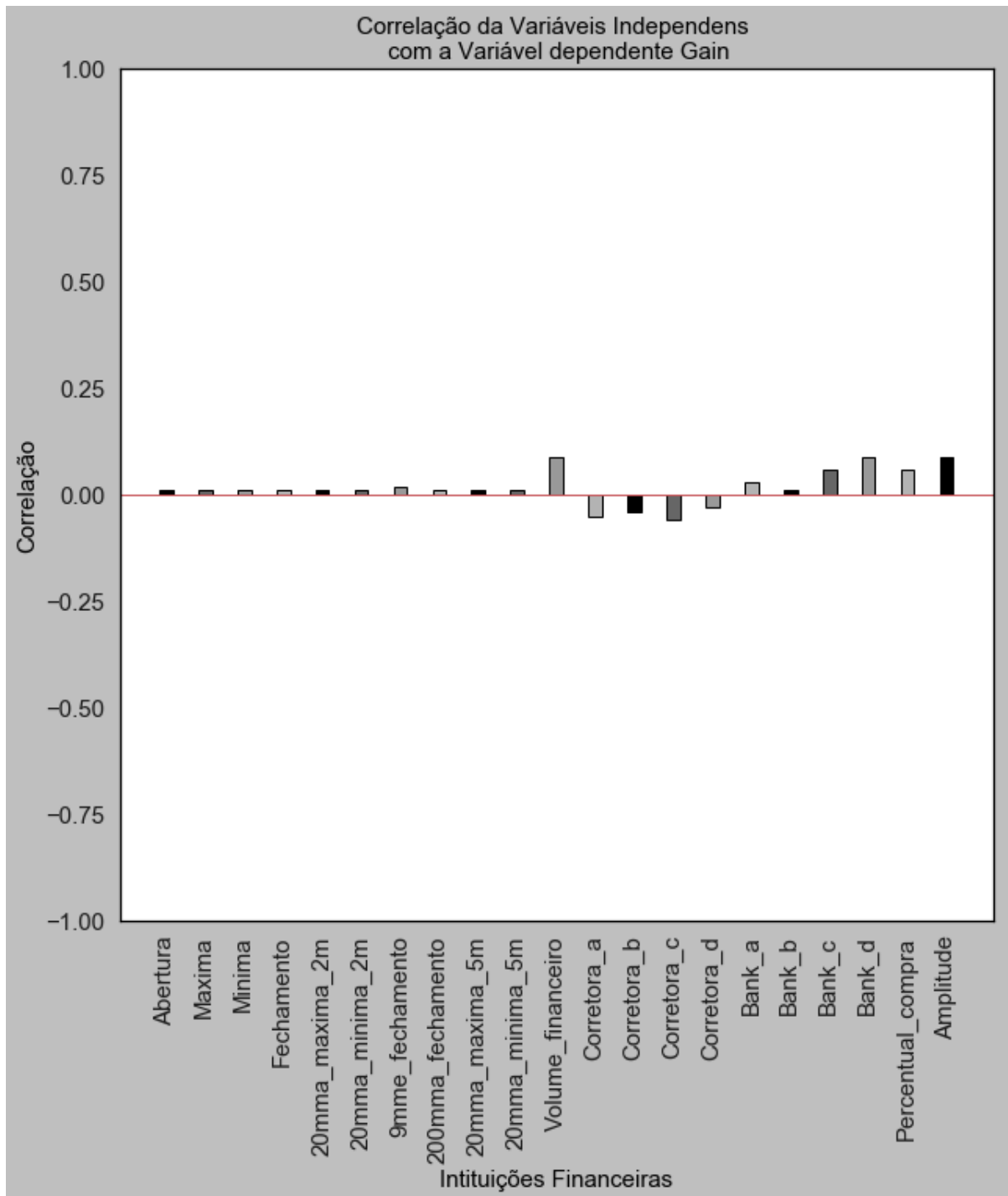


Figura 46 – Gráfico com a correlação das variáveis independentes com a variável dependente.

Fonte: Elaborado pelo autor (2019).

Os valores do coeficiente de correlação Pearson podem variar do valor +1 ao valor -1, onde o +1 indica a relação positiva perfeita entre as variáveis de entrada com a variável de saída, o -1 indica a relação negativa perfeita entre as variáveis de entrada com a variável de saída e um valor 0 indica que não existe relacionamento entre as variáveis de entrada com a variável de saída.

4.3 TREINAMENTO DO MODELO DE REDES NEURAIS ARTIFICIAIS PROFUNDA

Neste estudo, o treinamento dos modelos de redes neurais artificiais foi realizada em quatro etapas conforme Figura 47. Dentre estas etapas, foi realizado o pré-processamento de dados, arquitetura da rede neural artificial profunda, processo de aprendizado e, por fim, é apresentado os resultados finais do treinamento para avaliação dos modelos.

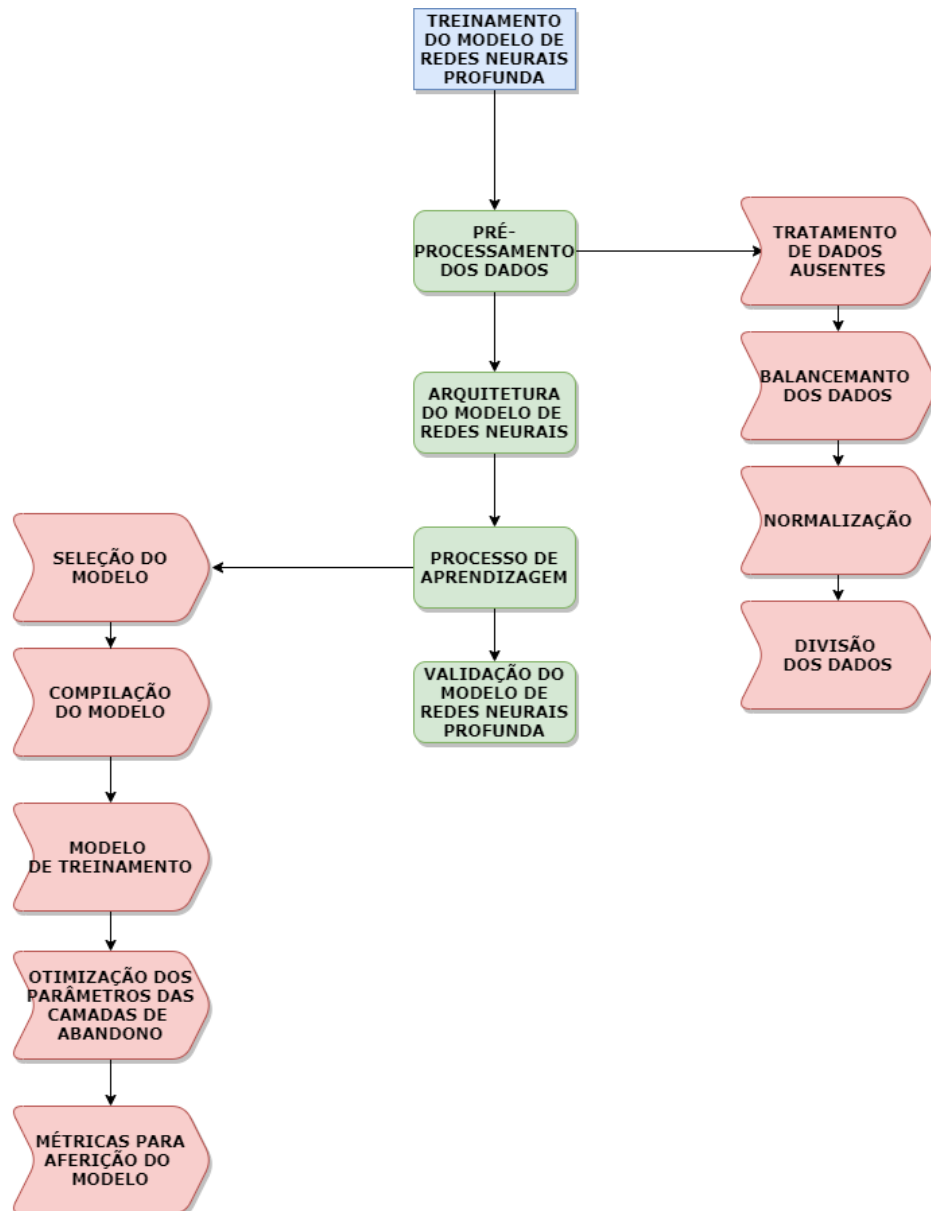


Figura 47 – Etapas do treinamento do modelo de redes neurais artificiais profunda.

Fonte: Elaborado pelo autor (2019).

4.3.1 Pré-processamento dos Dados

Conforme Chen e He (2018), o pré-processamento de dados é uma etapa importante quando deseja-se obter algumas informações dos conjuntos de dados para ajudar a fazer a

previsão. Como os dados iniciais podem apresentar grande ruído, é necessário reduzi-los para que eles não interfiram no resultado.

As variáveis utilizadas para elaboração do banco de dados foram descritas na Tabela 9. O período de dados analisados foi 03/06/2019 até 27/12/2019. Baseado em Chen, Zhou e Dai (2015), as primeiras variáveis escolhidas para compor o banco de dados do estudo foram os preços de fechamento, abertura, máximas, mínimas e o volume financeiro negociado. Segundo esses autores, após algumas experiências analisando ativos financeiros a partir da ótica destas variáveis, foram alcançados bons resultados em diferentes previsões introduzindo estes recursos como parâmetro de entrada do modelo. Contudo, por fazerem parte da regra de negociação estudada pelo trabalho, foram introduzidos os valores das médias móveis. Já as outras variáveis como bancos, corretoras, amplitude, percentual de compra e valores das médias do período gráfico de 5 minutos, foram introduzidas pelas suas relevâncias na movimentação dos preços e da regra de negociação.

1) Tratamento dos Dados Ausentes:

Um banco de dados pode apresentar dados faltantes (“NaN” - acrônimo em inglês para *Not a Number*) em suas colunas e linhas. No entanto, existem alguns métodos (na linguagem de programação) em *Python* para solucionar este problema de identificação e preenchimento de dados faltantes, tais como:

- *FillNaN 0*: preenche os dados ausentes com zero;
- *FillNaN mean*: preenche os dados ausentes com a média da coluna ou linha;
- *DropNaN*: apaga do banco de dados as linhas e colunas onde há dados ausentes.

Nesta trabalho, o método *DropNaN* foi escolhido para lidar com os dados ausentes.

2) Balanceamento e Reamostragem do Banco de Dados:

Bancos de dados desbalanceados são um problema comum quando é utilizado a classificação binária em algoritmos de aprendizado de máquina, onde ocorre uma razão desproporcional entre as observações de cada classe.

A maioria dos algoritmos de aprendizado de máquina têm um melhor desempenho quando o número das amostras de cada classe - no caso deste estudo a classificação binária - tenham praticamente o mesmo número. Isso porque a maioria destes algoritmos são projetados para desempenhar o papel de maximização da acurácia e redução dos erros.

Como o nosso estudo obteve 77.4% para operações vencedoras (classe 1) e 22.6% para operações perdedoras (classe 0), conforme descrito na Seção 4.1.1, nota-se um desbalanceamento entre as classes. Esta desproporcionalidade poderá influenciar negativamente

no treinamento do modelo. Para solucionar um problema desse tipo, são aplicadas técnicas de reamostragem.

Segundo Moore, Macbe e Duckworth (1996), os métodos de reamostragem (por exemplo: *Bootstrap*, Monte Carlo), permite quantificar a incerteza calculando os erros padrões e intervalos de confiança, bem como realizar testes de significância.

Para desempenhar a tarefa de balanceamento e reamostragem do banco de dados foi utilizado a biblioteca *scikit-learn*⁶. Esta biblioteca disponibiliza algumas ferramentas que se encarregam de realizar a tarefa de balanceamento e reamostragem do banco de dados. Como exemplo, pode-se citar a biblioteca *sklearn.utils.resample*. Esta biblioteca utiliza um algoritmo que tem como padrão a técnica estatística de reamostragem *bootstrapping*⁷.

3) Normalização:

A terceira etapa do pré-processamento de dados é a normalização dos dados. Normalização é o processo de dimensionamento dos dados para reduzir a redundância, aumentar a integridade e o desempenho do modelo e é utilizada quando as variáveis de entrada do conjunto de dados têm grandes diferenças entre seus intervalos ou simplesmente quando são medidos em unidades de medida diferentes. Essas diferenças nas amplitudes das variáveis de entrada causam problemas para muitos modelos de aprendizado de máquina.

4) Divisão dos Dados:

A quarta etapa do pré-processamento dos dados é a divisão dos dados. Esta divisão tem como o objetivo seccionar os dados em dados de treinamento e teste. Para este estudo foi utilizado a proporção de 70:30 onde 70% dos dados são destinados para treinamento e 30% para posterior teste.

4.3.2 Arquitetura da Rede Neural Profunda do Modelo

Neste estudo, a arquitetura da rede neural artificial profunda neste estudo adota a arquitetura geral proposta por Bishop (2006), em que existe uma camada de extração de recurso que é intercambiável entre a camada oculta, seguida por uma classificação binária ou regressão baseada no algoritmo de retro propagação.

⁶ A *scikit-learn* é uma biblioteca de aprendizado de máquina de código aberto para a linguagem de programação Python.

⁷ Conforme Maddala (2003), o método *Bootstrap* é uma técnica de reamostragem com o seguinte propósito: reduzir desvios e prover desvios padrão mais confiáveis. O seu funcionamento é dado da seguinte maneira: (y_1, y_2, \dots, y_n) seja a amostra dada. Retira-se dessa amostra uma amostra de tamanho n com reposição. Chama-se essa amostra de $\beta_j = (y_1^*, y_2^*, \dots, y_n^*)$, conhecida também como amostra *Bootstrap*. Cada y^* é uma escolha aleatória de (y_1, y_2, \dots, y_n) . E faz-se isso para $j = 1, 2, \dots, m$ e calcula-se de cada uma das amostras *Bootstrap* β_j . A distribuição $\hat{\theta}_j$ é a distribuição *Bootstrap* do estimador θ . As estimativas *Bootstrap* do desvio e variância de θ são derivadas dessa distribuição *Bootstrap*.

A arquitetura adotada, conforme Figura 48, mais especificamente, é formada por 08 camadas incluindo a camada de entrada, 03 camadas ocultas - por se tratar de um modelo de rede neural artificial profunda -, 03 camadas de abandono, e a última camada, a camada de saída.

A camada de entrada da arquitetura da rede neural artificial profunda é parametrizada com 22 dimensões. Isso ocorre porque, após o pré-processamento dos dados, foi averiguado a importância de 22 variáveis na análise da Regra de Negociação. As primeiras camadas ocultas são compostas por 12 neurônios, a partir da combinação das 22 variáveis da camada de entrada. Já a segunda e terceira camadas ocultas são compostas por 24 e 48 neurônios respectivamente. Cada camada oculta é interconectada com uma camada de abandono. Essa camada de abandono tem como objetivo evitar o *overfitting*⁸ ou *underfitting*⁹ durante o treinamento do modelo. Md. et al. (2018) recomenda que o valor de ajuste da camada de abandono seja entre 20%-50%. Corroborando Md. et al. (2018), Zhang e Wallace (2016) também usou um valor de ajuste para a camada de abandono de 50% como base dos parâmetros de seu estudo em uma rede neural convolucional. Para este estudo, foi adotado várias taxas para a camada de abandono entre a faixa de 20% a 50%. Foi escolhido para o modelo a faixa que obteve a melhor acurácia.

Maas, Hannun e Ng (2013) sugerem, para a primeira camada oculta, a aplicação da função de ativação *Relu*. Contudo, para as demais camadas de ativação, esses autores recomendam a aplicação da função de ativação *Lrelu*.

A camada de saída consiste apenas de um neurônio. Isso ocorre porque o problema a ser resolvido é uma classificação binária. Após o cálculo das camadas ocultas, a camada de saída - última camada - foi calculada pela função de ativação *sigmoid* visando obter a probabilidade em relação ao êxito ou não da regra de negociação.

⁸ O modelo *overfitting* ocorre quando o modelo se adaptou muito bem aos dados com os quais está sendo treinado; porém, não generaliza bem para novos dados. Ou seja, o modelo “decorou” o conjunto de dados de treino, mas não aprendeu de fato o que diferencia aqueles dados para quando precisar enfrentar novos testes.

Fonte: <https://pt.stackoverflow.com/questions/377643/o-que-%C3%A9-overfitting-e-underfitting-em-machine-learning>

⁹ O modelo *underfitting* ocorre quando o modelo não se adapta bem sequer aos dados com os quais foi treinado.

Fonte: <https://pt.stackoverflow.com/questions/377643/o-que-%C3%A9-overfitting-e-underfitting-em-machine-learning>

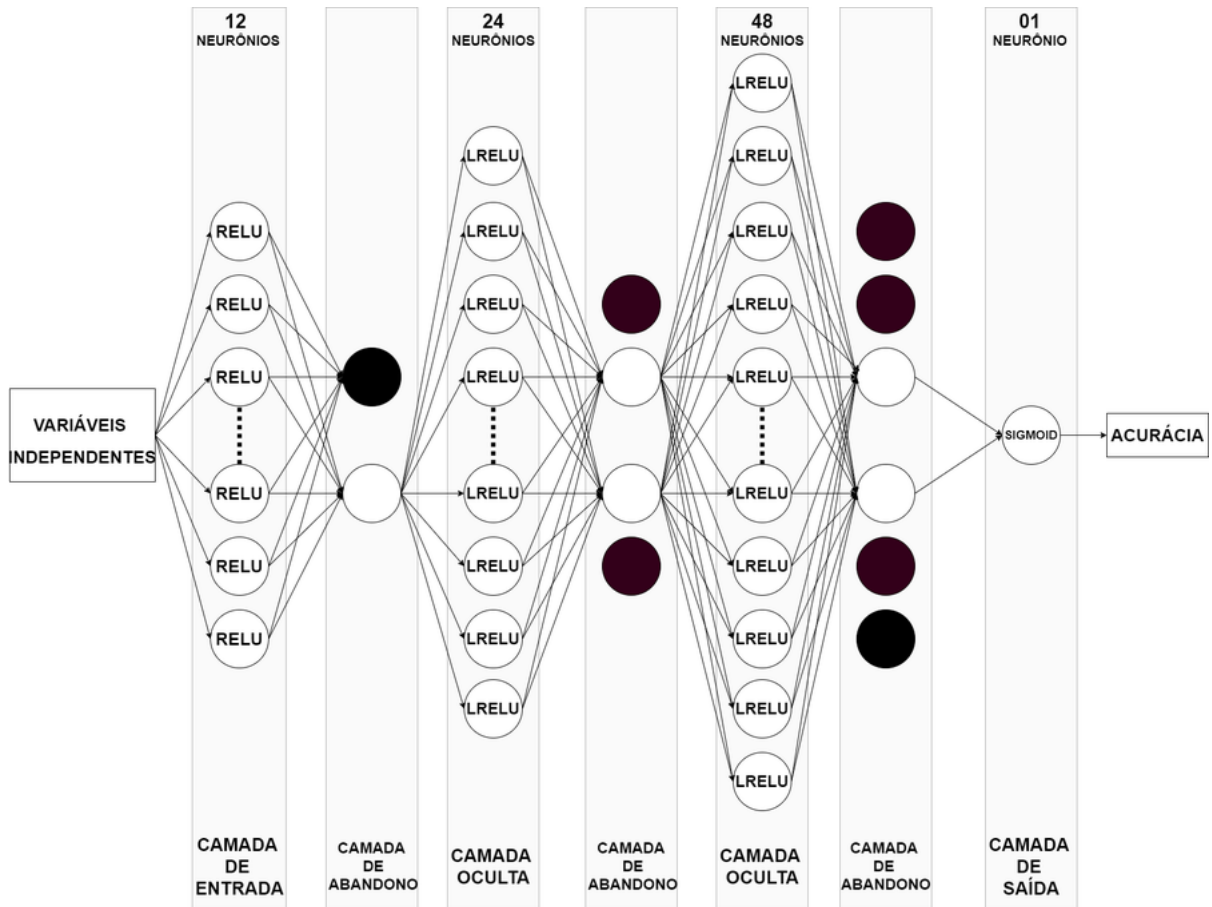


Figura 48 – Arquitetura do modelo de rede neural profundo do estudo.

Fonte: Elaborado pelo autor (2019).

4.3.3 Processo de Aprendizagem

Neste trabalho, o processo de aprendizagem foi realizado em quatro etapas: seleção do modelo, compilação, modelo de treinamento e validação do modelo.

1) **Seleção do modelo:**

Este trabalho utilizou o modelo de redes neurais artificiais profundas fornecidos pelo pacote *Keras*. *Keras* é um API (pacote), escrita em *Python*, desenvolvido com intuito de criar modelos de redes neurais artificiais.

2) **Compilação:**

No processo de aprendizado de uma rede neural artificial profunda, para minimizar a perda, a maneira mais usual é a utilização do algoritmo de retropropagação¹⁰(SAAD; PROKHOROV; WUNSCH, 1998). Para este estudo, o modelo criado foi compilado aplicando o algoritmo *Adam* como um otimizador com uma taxa de aprendizagem de 0,0001 (CHEN;

¹⁰ Tradução para algoritmo *backpropagation*.

HE, 2018). Outros parâmetros que foram necessários considerar na compilação do modelo foram os parâmetros de perda. Com o intuito de encontrar e minimizar os mesmos, foi utilizado a função de custo *binary cross-entropy loss*.

3) Modelo de Treinamento:

O último processo de aprendizado é o modelo de treinamento. Nesse processo de treinamento, foi determinado o número de amostras do banco de dados que foi atualizada a cada avanço do gradiente (tamanho do lote¹¹), o número de iterações em todo o banco de dados (época¹²), a distribuição dos dados de treinamento a serem usados como validação do banco de dados e etapa de validação.

Para este estudo, o banco de dados foi dividido entre banco de dados de treinamento e banco de dados de teste. Para o banco de dados de treinamento foram reservados 70% do banco de dados original. Para o banco de dados de teste, foram 30% do banco de dados original. A Tabela 13 contém os parâmetros do tamanho do lote e a época escolhida.

Tabela 13 – Parâmetros adotados para o treinamento do modelo.

Parâmetro	Tamanho
Tamanho do lote (<i>batch size</i>)	10
Época (<i>epoch</i>)	100

Fonte: Elaborado pelo autor (2019).

4) Otimização dos Parâmetros das Camadas de Abandono:

A otimização das camadas de abandono é uma atividade imprescindível na construção de uma rede neural artificial profunda. A utilização destas podem melhorar drasticamente a convergência e também reduzir o erro de teste (LI; GONG; YANG, 2016). O aprimoramento das camadas de abandono respeitam as seguintes condições:

- As camadas de abandono foram treinadas com os valores de 20, 30, 40 e 50%. Contudo, os valores adotados para treinamento podem aparecer repetidos entre as camadas de abandono, ou seja, ocorre a repetição de um mesmo valor para a primeira, segunda e terceira camada abandono. Sendo assim, num conjunto de valores $P=\{20, 30, 40, 50\}$, no primeiro treinamento, a primeira, segunda e terceira camada de abandono foram configuradas nos valores de 20%. No segundo treinamento, a primeira e segunda camada de abandono adotaram o valor de 20%. Sendo que a terceira camada adotou o valor de 30%. Conforme Equações 4.7 e 4.8 foi possível obter 64 agrupamentos ou combinações em relação ao conjunto P. A

¹¹ Na comunidade de ciências de dados utiliza-se o termo em inglês *batch size*.

¹² Na comunidade de ciências de dados utiliza-se o termo em inglês *epoch*.

simulação só finalizou quando as três camadas de abandono adotaram todos os valores do conjunto P.

$$A_{(n,p)} = n^p \quad (4.7)$$

onde n seria o número de elementos do conjunto e p quantidade de elementos por agrupamento. Como a arquitetura da rede neural artificial é composta por três camadas de abandono, logo $p = 3$.

$$A_{(4,3)} = 4^3 = 64 \quad (4.8)$$

- A configuração das camadas de abandono que encontrou a maior acurácia no treinamento foi utilizada no modelo final. Conforme Tabela 14, seguem as cinco configurações que encontraram as maiores acurácias conforme faixas adotadas na simulação.

Na Tabela 14, é apresentado as cinco melhores acurácias do modelo de redes neurais profunda. As Camadas de Abandono configuradas com 50% proporcionaram a melhor acurácia para o modelo, no caso, 65%. Os valores do *recall* e pontuação F1 também apresentaram uma pequena diferença entre as demais configurações. Por outro lado, a precisão foi demasiadamente menor do que o modelo de camada de abandono nos valores de 0,5 - 0,2 - 0,5.

Tabela 14 – Cinco maiores acurácias segundo os valores adotados para as camadas de abandono.

Modelo	Acurácia	Precisão	Recall	Pontuação F1
DNN <i>Dropout</i> 0,5 - 0,5 - 0,5	0,65	0,82	0,73	0,77
DNN <i>Dropout</i> 0,5 - 0,3 - 0,2	0,63	0,80	0,72	0,76
DNN <i>Dropout</i> 0,5 - 0,3 - 0,5	0,60	0,81	0,67	0,73
DNN <i>Dropout</i> 0,5 - 0,5 - 0,3	0,60	0,82	0,64	0,72
DNN <i>Dropout</i> 0,5 - 0,2 - 0,5	0,60	0,84	0,62	0,71

Modelo	Acurácia	Precisão	Recall	Pontuação F1
--------	----------	----------	--------	--------------

Fonte: Elaborado pelo autor (2019).

Sendo assim, mediante aos resultados da simulação, os valores para as camadas de abandono para o modelo final que aferirá a acurácia da regra de negociação foi de 50%. As Figuras 49 e 50 mostram a desempenho - acurácia e a perda gerada do banco de dados de treinamento e teste através da atualização dos pesos respectivamente - do modelo em questão.

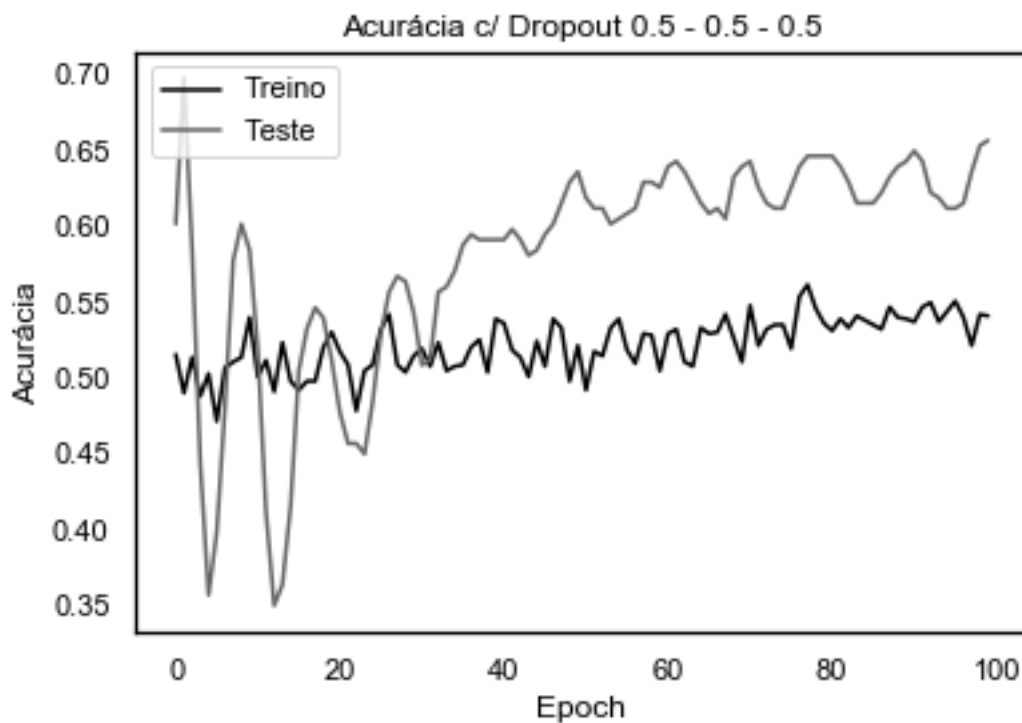


Figura 49 – Acurácia do banco de dados de treino e teste.

Fonte: Elaborado pelo autor (2019).

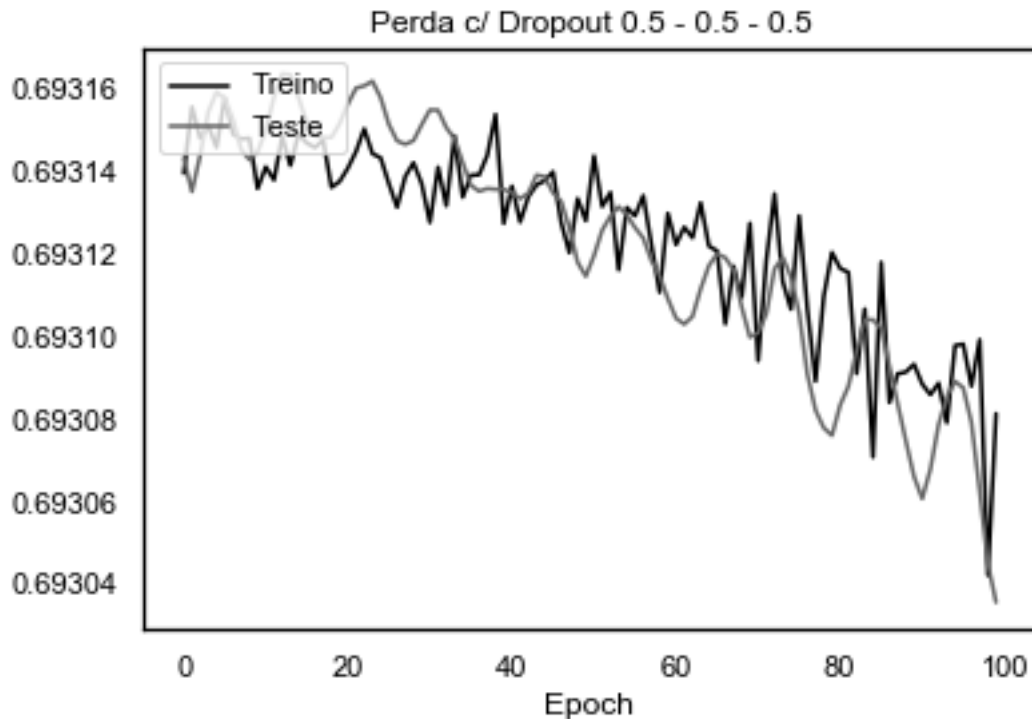


Figura 50 – Perda do modelo para que os pesos possam ser atualizados.

Fonte: Elaborado pelo autor (2019).

5) Métricas para Aferição do Modelo:

No processo de validação do modelo foi adotada a métrica acurácia (A). As métricas precisão (P), *recall* (R) e a pontuação F1 (F_w) serviram como métricas auxiliares para a verificação do desempenho do algoritmo de redes neurais artificiais profunda conforme abordadas na Seção 4.1.3.

4.3.4 Validação do Modelo de Redes Neurais Profunda

Cada vez que um modelo de rede neural profunda é treinado, os valores encontrados para acurácia podem variar. Tanto no conjunto de treinamento quanto no conjunto de teste podendo levar à superestimação das taxas de validação do modelo. Outro agravante para a superestimação é a utilização de 70% do banco de dados original para compor o conjunto de teste.

Partindo desta premissa, não é difícil ver que o modelo treinado fornecerá estimativas aproximadamente imparciais para a acurácia, já que cada conjunto de treinamento contém $n - 1$ observações, que é quase o número de observações no conjunto completo do banco de dados.

Visando minimizar esta superestimação no modelo de redes neurais profundas, foi aplicado a validação cruzada (CV) com $k = 10$. O principal motivo da utilização da

validação cruzada é a tentativa de suavizar o viés e variância. Uma vantagem menos óbvia, mas potencialmente mais importante do CV configurado com $k = 10$, é que ela geralmente fornece estimativas mais precisas da taxa de acurácia, performando o conjunto de treinamento com $\frac{(k-1)n}{k}$ observações. Após a definição do valor de k para a validação através da validação cruzada, o modelo foi treinado novamente. As acurácias obtidas pelo treinamento são mostradas conforme Tabela 15.

Tabela 15 – Valores da acurácia utilizando a validação cruzada.

Validação Cruzada ($k = 10$)	Acurácia (%)
$k = 01$	33
$k = 02$	36
$k = 03$	34
$k = 04$	35
$k = 05$	44
$k = 06$	49
$k = 07$	37
$k = 08$	56
$k = 09$	57
$k = 10$	55

Fonte: Elaborado pelo autor (2019).

É notório que os valores encontrados na Tabela 14 são maiores que os valores encontrados na Tabela 15. Concluindo que houve uma superestimação nas acurácias encontradas no primeiro treinamento do modelo.

Aplicando a equação 3.32, calcula-se a média das acurácias da Tabela 15. Sendo assim, a acurácia final para o modelo de redes neurais profunda deste estudo são 46%.

5 CONCLUSÃO

5.1 ANÁLISE DOS ARTEFATOS EMPREGADOS

No decorrer da elaboração da pesquisa, a concentração no aprimoramento de artefatos auxiliou na criação e validação de diferentes constructores, modelos, métodos, instâncias e *design propositions* que auxiliaram no fornecimento de respostas para os problemas inerentes na análise da desempenho da regra de negociação. A Tabela 16 retrata o protocolo de pesquisa, especificando os artefatos detectados e desenvolvidos durante a pesquisa.

Tabela 16 – Modelo de protocolo de pesquisa para a proposição de artefatos para resolver o problema específico.

ARTEFATO PROPOSTO	TIPIFICAÇÃO DE ARTEFATO	JUSTIFICATIVA	PRÓS	CONTRA
Mini Contrato Futuro do Dólar	Constructor	Um dos ativos financeiros que mais concentram investidores individuais.	Por possuir alta liquidez e baixo custo, o mini contrato futuro do dólar é um dos ativos mais negociados.	Possui características técnicas específicas. Dificultando o entendimento por parte dos investidores individuais.
Análise Técnica	Constructor	A Análise Técnica como materialização prática de um conceito de estudo de derivativos financeiros.	A Análise Técnica torna-se didática por utilizar ferramentas mais intuitivas e que não exige um grande conhecimento em economia.	Múltiplas interpretações. Os investidores têm diferentes análises a partir das ferramentas da Análise Técnica.

ARTEFATO PROPOSTO	TIPIFICAÇÃO DE ARTEFATO	JUSTIFICATIVA	PRÓS	CONTRA
Redes Neurais Artificiais Profunda	Constructor / Método	Classe de algoritmo de aprendizagem de máquina que visa encontrar padrões em complexos conjuntos de dados.	Apoio de uma ferramenta de aprendizado de máquina para aferir o desempenho da regra de negociação.	Experimentar diversos parâmetros para atingir a melhor acurácia do modelo.
Banco de Dados	Modelo	Artefato criado para organizar dados que servirão de <i>input</i> para o treinamento do modelo de redes neurais artificiais profunda.	Dados organizados.	Erros que podem ocorrer devido à falta de atenção na coleta dos dados que não são recolhidos pelo robô de negociação.
Matriz de Confusão	Método	Utilização de um método já existente com objetivo de orientar o pesquisador na decisão em quais modelos de redes neurais artificiais irão desempenhar um melhor desempenho.	Conhecimento da acurácia do modelo de redes neurais artificiais profunda.	<i>Inputs</i> de dados não reais.

ARTEFATO PROPOSTO	TIPIFICAÇÃO DE ARTEFATO	JUSTIFICATIVA	PRÓS	CONTRA
Classificação Binária da Regra de Negociação	Instanciação	Conceito baseado em um novo olhar para a aferição de uma regra de negociação a partir da aplicação do modelo de redes neurais profunda.	Aplicabilidade de mais um método de aferição do desempenho de uma regra de negociação.	Resistência em incorporar esse artefato em outras pesquisas.
<i>SoftwareMetatrader 5</i>	<i>Design Proposition</i>	Possibilita negociar (comprar e vender) ativos financeiros.	As operações de compra e de venda são realizadas através de um robô de negociação.	Por não expor todas as informações necessárias para a criação do banco de dados, há a necessidade de a utilização de outro <i>software</i> para realizar tal tarefa.
<i>SoftwareTryd</i>	<i>Design Proposition</i>	Possibilita negociar (comprar e vender) ativos financeiros.	Expõem informações do mercado que não são fornecidos no <i>softwareMetatrader 5</i> .	Erros que podem ocorrer devido à falta de atenção na coleta dos dados que não são recolhidos pelo robô de negociação.
<i>SoftwareSpyder</i>	<i>Design Proposition</i>	Possibilita a manipulação dos dados e a criação de um modelo de redes neurais artificiais a partir e bibliotecas cadastradas.	Otimização da manipulação dos dados.	Conhecimento avançado, dificultando a implementação da ferramenta por qualquer pessoa.

ARTEFATO PROPOSTO	TIPIFICAÇÃO DE ARTEFATO	JUSTIFICATIVA	PRÓS	CONTRA
<i>SoftwareMetaquotes</i>	<i>Design Proposition</i>	Possibilita a programação do robô de negociação para o envio de ordens de compra e de venda no <i>softwareMetatrader 5</i> . Além de extrair informações para a criação do banco de dados.	Automatização da coleta e registros dos dados.	Conhecimento avançado, dificultando a implementação da ferramenta por qualquer pessoa.

Fonte: Elaborado pelo autor (2019).

5.2 CONSIDERAÇÕES FINAIS

Neste estudo foi utilizado o modelo de redes neurais profunda (DNN) com o intuito de aferir a acurácia, e logo o desempenho, da regra de negociação conhecida como cruzamento de médias móveis em operações *daytrade* no ativo financeiro mini contrato futuro do dólar. Ativo este, negociado na bolsa de valores brasileira.

Todo o processo, desde da criação de um robô de negociação para a coleta dos dados, até a conclusão do modelo de redes neurais profunda, de fato, acarretou a aplicação de diversas técnicas e metodologias de análise de dados.

A análise de dados, em síntese, já é uma tarefa difícil para o pesquisador que visa realizar um trabalho científico com primazia. Somado a isso, para este estudo, a complexidade atingiu níveis mais elevados devido à aleatoriedade das informações que a modalidade de *daytrade* proporciona. Além de, em concordância com a análise de dados, possibilitar que o investidor individual esteja munido com variadas ferramentas de análise técnica.

Neste trabalho, esta complexidade inicia-se, logo, na escolha do tempo gráfico que a regra de negociação foi analisada. Quando é estudado operações *daytrade*, o *daytrader* pode realizar a compra e venda de um ativo financeiro tanto no tempo gráfico de um minuto ao tempo gráfico de quatro horas. No campo da coleta e manipulação dos dados, os métodos empregados pelo pesquisador pode alterar de forma significativa a métrica validadora do modelo de redes neurais artificiais profunda, que, neste contexto, foi a acurácia. Outro fator influenciável na acurácia final do modelo é a própria arquitetura empregada na rede neural artificial profunda, como por exemplo: a mudança da quantidade de camadas ocultas, ou a alteração da função de ativação, ou escolha de outro parâmetro para as camadas de

abandono, etc.

Isto posto, os processos empregados neste estudo para a avaliação do desempenho da regra de negociação são totalmente parametrizáveis. Esta generalização dos algoritmos permitiu que futuras mudanças dos atributos possam proporcionar, talvez, a variação da acurácia, tanto para a piora do modelo, quanto para a melhora do mesmo.

Sendo assim, para este estudo, que utilizou o tempo gráfico de dois e cinco minutos para aplicação da regra de negociação de cruzamento de médias móveis, mostrou que a otimização da arquitetura da rede neural artificial e a atualização dos parâmetros da camada de abandono afetaram suas precisões.

Após analisar todos os valores possíveis entre 20% - 50% para as três camadas de abandono, a arquitetura de rede neural artificial profunda que obteve a melhor acurácia foi a que empregou a taxa de 50% nas três camadas de abandono, performando uma acurácia de 65%. A precisão, o *recall* e a pontuação f-1, foram atribuídos respectivamente com os valores de 82%, 73% e 77%, demonstrando que a acurácia encontrada no modelo de redes neurais artificiais profunda aplicada no estudo estava coerente com divisão dos bancos de dados de treino e teste.

Contudo, quando aplicada a validação cruzada no modelo com $k = 10$, o modelo teve uma acurácia de 46%. De acordo com esta acurácia, conclui-se que, com as variáveis de entrada utilizadas no trabalho, com os tempos gráficos utilizados para a aplicação da regra de negociação e com as técnicas de manipulação dos dados, não é possível prever se uma operação é rentável ou não empregando a regra de negociação de cruzamento de médias móveis.

Destaca-se que os resultados discutidos nesta dissertação foram apresentados pelo autor no *40th International Symposium on Forecasting (ISF)*. Embora a acurácia encontrada pelo modelo de redes neurais profunda elimine qualquer indício de preditividade referente à regra de negociação, a generalização adotada pelo autor nos algoritmos tornou possível a alteração dos parâmetros. Possibilitando que outros pesquisadores possam realizar novos estudos sobre os dados coletados para que haja a possibilidade da criação de novos modelos preditivos.

Para que seja encontrada uma acurácia diferente deste estudo, de 46%, existem algumas, e pode-se dizer, muitas orientações interessantes para estudos futuros:

- 1) Alterar o tempo gráfico para aplicação da regra de negociação. Neste estudo foram utilizados os tempos gráficos de dois e cinco minutos. Talvez, futuramente, possa aplicar as técnicas utilizadas aqui em tempos gráficos maiores como 30 e 60 minutos;
- 2) Utilizar outras variáveis independentes para verificar se há uma alteração na acurácia, pois, a regra de negociação pode não apenas ser influenciado pelas variáveis como preço de abertura, preço de fechamento, preço máximo, preço mínimo, etc;

- 3) Usar mais conjuntos de dados para julgar se o resultado será melhor, pois, o número de conjuntos de dados é de grande importância para que o modelo de redes neurais profunda desempenhe um bom trabalho;
- 4) Alterar o método de reamostragem do conjunto de dados;
- 5) Alterar os parâmetros de treinamento do modelo;
- 6) Mudar as funções de ativação do modelo de redes neurais artificiais profunda;
- 7) Alterar a arquitetura do modelo de redes neurais artificiais profunda;
- 8) Aplicar outros modelos de aprendizado de máquina como: regressão logística, árvores de decisão, máquina de vetores de suporte, dentre outros.

Por fim, julga-se o cumprimento dos objetivos específicos apresentado na Seção 1.3 desta dissertação, conforme reportado na Tabela 17.

Tabela 17 – Cumprimento dos Objetivos Específicos.

OBJETIVOS ESPECÍFICOS	CUMPRIMENTO DOS OBJETIVOS
Criação de um algoritmo computadorizado (robô de negociação) que identifique as entradas de compra e venda da regra de negociação	A criação de um algoritmo computadorizado (robô de negociação) foi criado utilizando a plataforma de negociação Metatrader 5. Para tal, foi desenvolvido um código <i>mql5</i> com todos os parâmetros de entrada e saída da regra de negociação conforme Seção 4.1.1.

OBJETIVOS ESPECÍFICOS	CUMPRIMENTO DOS OBJETIVOS
Determinar as variáveis independentes que serão analisadas na ocorrência da regra de negociação	Ao todo, para o estudo da acurácia da regra de negociação, foram adotadas 22 variáveis conforme descrita na Tabela 9. As variáveis independentes como os preços de fechamento, abertura, máxima e mínima deram-se por terem alcançado bons resultados após algumas análises em outros ativos financeiros (CHEN; ZHOU; DAI, 2015). As outras variáveis foram introduzidas devido suas relevâncias a partir da ótica do pesquisador.
Identificar as operações de compra e de venda que atingiram os pontos necessários para serem operações lucrativas	A identificação das operações de compra e de venda que foram vitoriosas só foi possível após determinar os critérios para tal. Esta determinação está descrita na Tabela 11. Após definir os critérios para quais operações seriam “vencedoras ou “perdedoras”, foi possível identificar as operações de compra e de venda que atingiram os pontos necessários para serem operações lucrativas. As operações vencedoras correspondem 77,4% das operações realizadas. Por outro lado, as operações perdedoras correspondem 22,6%.
Montar um banco de dados com as variáveis independentes baseadas nas entradas de compra e venda do robô de negociação	O robô de negociação, além de abrir e fechar operações, quando estas aconteciam, o mesmo coletava automaticamente as variáveis independentes. Contudo, o robô só não coletava automaticamente os bancos e as corretoras. A coleta destes dados foi realizada pelo pesquisador com outra plataforma de negociação chamada <i>Tryd</i> .
Codificar o modelo de redes neurais profunda em <i>python</i>	Foram aplicadas diversas metodologias conforme Seção 4.3.2.
Medir a acurácia do modelo de redes neurais artificiais profunda e atestar a assertividade da regra de negociação	Foram aplicadas diversas metodologias conforme Seção 4.3.4.

OBJETIVOS ESPECÍFICOS	CUMPRIMENTO DOS OBJETIVOS
--------------------------	---------------------------

Fonte: Elaborado pelo autor (2019).

Com base na Tabela 17, é possível afirmar que os objetivos específicos de pesquisa foram todos cumpridos, satisfazendo assim o objetivo geral da dissertação.

REFERÊNCIAS

- BIRGUL, E.; OZTURAN, M.; BADUR, B. Stock market prediction using artificial neural networks. In: **Proceedings of the in 3rd Hawaii International Conference on Business**. [S.l.: s.n.], 2003.
- BISHOP. **Pattern Recognition and Machine Learning**. [S.l.]: Springer, 2006.
- BLAGUS, R.; LESKOSEK, B.; STARE, J. Comparison of bibliometric measures for assessing relative importance of researchers. **Scientometrics**, v. 105, n. 3, p. 1743 – 1762, 2015. Disponível em: <http://dx.doi.org/10.1007/s11192-015-1622-6>.
- BOAINAIN et al. “Ombro-Cabeca-Ombro”: Testando a Lucratividade do Padrão Gráfico de Análise Técnica no Mercado de Ações Brasileiro. **Revista Brasileira de Finanças**, Sociedade Brasileira de Finanças, Rio de Janeiro, Brasil, v. 7, n. 3, p. 265 – 303, 2009. ISSN 1679-0731.
- BOLLEN, J.; H. MAO; ZENG, X. Twitter mood predicts the stock market. **Journal of Computational Science**, v. 2, n. 1, p. 1 – 8, 2011.
- BROCK, W.; LAKONISHOK, J.; LEBARON, B. Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. **The Journal of Finance**, v. 47, n. 5, p. 1731 – 1764, 1992a.
- BROCK, W.; LAKONISHOK, J.; LEBARON, B. Simple Technical Trading Rules and the Stochastic Properties of Stock Returns. **Journal of Finance**, v. 47, n. 5, p. 1731 – 1764, Dezembro 1992b. Disponível em: <https://www.jstor.org/stable/2328994>.
- BUSH, V. As we may think. **Atlantic Monthly**, v. 176, n. 1, p. 101 – 108, 1945. Disponível em: <http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm>. Acesso em: 26 junho 2018.
- CALLON, M.; COURTIAL, J. P.; LAVILLE, F. Co-word analysis as a tool for describing the network of interactions between basic and technological research: The case of polymer chemistry. **Scientometrics**, v. 22, n. 1, p. 155 – 205, Setembro 1991.
- CHAGUE, F.; GIOVANNETTI, B. É possível viver de day-trading? 2019.
- CHEN, K.; ZHOU, Y.; DAI, F. A LSTM-based method for stock returns prediction: A case study of China stock market. In: **IEEE International Conference on Big Data (Big Data)**. Santa Clara, CA: [s.n.], 2015. p. 2823 – 2824.
- CHEN, S.; HE, H. Stock Prediction Using Convolutional Neural Network. 2018.
- COBO, M. J. et al. SciMAT: a new science mapping analysis software tool. **Journal of the American Society for Information Science and Technology**, Granada, Spain, v. 63, n. 8, p. 1609 – 1630, julho 2012.
- CRAIG, I. D. Measuring Academic Research: How to Undertake a Bibliometric Study. **Learned Publishing**, v. 23, n. 3, p. 270 – 271, 2010. Disponível em: <http://dx.doi.org/10.1087/20100315>.

DRESCH, A.; LACERDA, D. P.; V. JÚNIOR, A. **Design Science Research: Método de Pesquisa para Avanço da Ciência e Tecnologia**. Porto Alegre: Bookman Editora, 2015. 204 p.

ELSEVIER. 2019. Disponível em: https://www.elsevier.com/__data/assets/pdf_file/0008/208772/ACAD_R_SC_FS.pdf.

FERNANDES, M. da S.; HAMBERGER, P. A. do V.; VALLE, A. C. M. do. ANÁLISE TÉCNICA E EFICIÊNCIA DOS MERCADOS FINANCEIROS: UMA AVALIAÇÃO DO PODER DE PREVISÃO DOS PADRÕES DE CANDLESTICK. **REVISTA EVIDENCIAÇÃO CONTÁBIL & FINANÇAS**, v. 3, n. 3, p. 35 – 54, Dezembro 2015. Disponível em: <http://periodicos.ufpb.br/ojs2/index.php/recfin>.

GENÇAY, R. The predictability of security returns with simple technical trading rules. **Journal of Empirical Finance**, v. 5, p. 347 – 359, 1998.

GIL, A. C. **Como elaborar projetos de pesquisas**. 5ª. ed. São Paulo: Atlas, 2010.

GOH, J. et al. Forecasting government bond risk premia using technical indicators. **Singapore Management University and Washington University in St. Louis.**, 2013.

GRANDA-ORIVE, J. I. et al. Ciertas ventajas de Scopus sobre Web of Science en un análisis bibliométrico sobre tabaquismo. **Revista Española de Documentación Científica**, v. 36, n. 2, 2013.

GUTIÉRREZ-SALCEDO, M. et al. Some bibliometric procedures for analyzing and evaluating research field. **Applied Intelligence**, v. 48, n. 5, p. 1275 – 1287, 2018.

HAIKIN, S. **Redes Neurais: princípios e práticas**. 2. ed. Porto Alegre: Bookman, 2001.

HEIDARI, A. A. et al. Ant Lion Optimizer: Theory, Literature Review, and Application in Multi-layer Perceptron Neural Networks. **Nature-Inspired Optimizers**, v. 811, p. 23 – 46, Fevereiro 2019.

HINTON, G. How Neural Networks Learn from Experience. **Cognitive modeling**, 2002.

HIRSCH, J. E. An index to quantify an individual's scientific research output. v. 102, n. 46, p. 16569 – 16572, setembro 2005.

INTHACHOT, M.; BOONJING, V.; INTAKOSUM, S. Artificial Neural Network and Genetic Algorithm Hybrid Intelligence for Predicting Thai Stock Price Index Trend. **Hindawi Publishing Corporation**, Computational Intelligence and Neuroscience, v. 2016, Outubro 2016. Disponível em: <http://dx.doi.org/10.1155/2016/3045254>.

JAMES, G. et al. **An Introduction to Statistical Learning: with Applications in R**. 1st. ed. [S.l.]: Springer Texts in Statistics (Book 103), 2013. 440 p.

KERAMATI, A. et al. Improved churn prediction in telecommunication industry using data mining techniques. **Applied Soft Computing**, v. 24, p. 994 – 1012, novembro 2014.

KIMOTO, T. et al. Stock Market Prediction System with Modular Neural Networks. In: **Proceedings of the 1990 International Joint Conference on Neural Networks (IJCNN '90)**. Washington, USA: [s.n.], 1990. v. 1, p. 1 – 6.

KINGMA, D. P.; BA, J. Adam: A Method for Stochastic Optimization. **CoRR**, abs/1412.6980, 2014. Disponível em: <http://arxiv.org/abs/1412.6980>.

LATHUILIERE, S. et al. A Comprehensive Analysis of Deep Regression. [arXiv:1803.08450v1], 2018.

LEUNG, M. T.; DAOUK, H.; CHEN, A. Forecasting stock indices: a comparison of classification and level estimation models. **International Journal of Forecasting**, v. 16, n. 2, p. 173 – 190, Abril - Junho 2000.

LI, Z.; GONG, B.; YANG, T. Improved Dropout for Shallow and Deep Learning. **CoRR**, abs/1602.02220, 2016. Disponível em: <http://arxiv.org/abs/1602.02220>.

LO, A. W.; MACKINLAY, A. C. Stock market prices do not follow random walks: Evidence from a simple specification test. **Review of Financial Studies**, v. 1, p. 41 – 66, 1988.

LO, A. W.; MACKINLAY, A. C. Maximizing predictability in the stock and bond markets. **Macroeconomic Dynamics**, v. 1, p. 102 – 134, 1997.

LO, A. W.; MAMAYSKY, H.; WANG, J. Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation. **Journal of Finance**, v. 55, p. 1705 – 1765, 2000.

MAAS, A. L.; HANNUN, A. Y.; NG, A. Y. Rectifier nonlinearities improve neural network acoustic models. In: **ICML**. [S.l.: s.n.], 2013.

MADDALA, G. S. **Introdução à Econometria**. 3rd. ed. Rio de Janeiro: LTC, 2003. 318 p.

MAHARANI, D.; MURFI, H. Deep Neural Network For Structured Data - A Case Study Of Mortality Rate Prediction Caused By Air Quality. **Journal of Physics: Conference Series**, v. 1192, 2019.

MALKIEL, B. **A Random Walk down Wall Street**: Including a Life-Cycle Guide to Personal Investing. 11th. ed. [S.l.]: W. W. Norton & Company, 2016. 496 p.

MD., M. et al. Effect of Hyper-Parameter Optimization on the Deep Learning Model Propose for Distributed Attack Detection in Internet of Things Environment. In: ARXIV:1806.07057 (Ed.). [S.l.: s.n.], 2018.

MELLO, C. H. P.; TURRIONI João B. Pesquisa-ação na engenharia de produção: proposta de estruturação para sua condução. **Produção**, v. 22, n. 1, p. 1 – 13, jan/fev 2012. Disponível em: http://www.scielo.br/pdf/prod/v22n1/aop_t6_0010_0155.pdf. Acesso em: 14 de junho de 2018.

MOORE, D. S.; MACBE, G. P.; DUCKWORTH, W. M. **The Practice of Business Statistic**: Using Data for Decisions. 1st. ed. Rio de Janeiro: LTC, 1996. 785 p.

NIELSEN. **Neural network and Deep Learning**. [s.n.], 2019. Disponível em: <http://neuralnetworksanddeeplearning.com/chap2.html>. Acesso em: 14/10/2019.

NISON, S. **Japanese Candlestick Charting Techniques**: a Contemporary Guide to the Ancient Investment Techniques of the far East. 2nd. ed. [S.l.]: Prentice Hall Press, 2001. 336 p.

NOYONS, E. C. M.; MOED, H.; LUWEL, M. Combining Mapping and Citation Analysis for Evaluative Bibliometric Purposes: A Bibliometric Study. **JASIS**, v. 50, n. 2, p. 115 – 132, 1999. Disponível em: 3.0.CO;2-J.

PALEX. **Estratégias Operacionais de Análise Técnica de Ações**. 1st. ed. [S.l.: s.n.], 2014.

QUEVEDO-SILVA, F. et al. Estudo bibliométrico: orientações sobre sua aplicação. **Revista Brasileira de Marketing**, p. 246 – 262, 2016.

RATHER, A.; AGARWAL, A.; SASTRY, V. Recurrent neural network and a hybrid model for prediction of stock returns. **Expert Systems with Applications**, 2014.

SAAD, E. W.; PROKHOROV, D. V.; WUNSCH, D. C. Comparative Study of Stock Trend Prediction Using Time Delay, Recurrent and Probabilistic Neural Networks. **IEEE TRANSACTIONS ON NEURAL NETWORKS**, v. 9, n. 6, p. 1456 – 1470, 1998.

SPINAK, E. Indicadores cientímetricos. **Ciência da Informação**, scielo, v. 27, p. nd – nd, 00 1998. ISSN 0100-1965. Disponível em: <http://www.scielo.br/scieloOrg/php/articleXML.php?lang=pt&pid=S0100-19651998000200006>.

SRIVASTAVA, N. et al. Dropout: A simple way to prevent neural networks from overfitting. **J. Mach. Learn. Res.**, **15(1):1929–1958**, January 2014.

TABELL, E. W.; TABELL, A. W. The Case for Technical Analysis. **Financial Analysts Journal**, v. 20, n. 2, p. 67 – 76, 1964.

TODESCHINI, R.; BACCINI, A. **Handbook of Bibliometric Indicators**: Quantitative Tools for Studying and Evaluating Research. [S.l.]: Wiley-VCH Verlag GmbH & Co. KGaA, 2016. ISBN 9783527681969.

TRANFIELD, D.; DENYER, D.; SMART, P. Towards a Methodology for Developing Evidence-Informed Management Knowledge by Means of Systematic Review. **British Journal of Management**, v. 14, p. 207 – 222, 2003.

TREYNOR, J.; FERGUSON, R. In defense of technical analysis. **Journal of Finance**, n. 40, p. 757 – 773, 1985.

VANTI, N. A. P. Da bibliometria à webometria: uma exploração conceitual dos mecanismos utilizados para medir o registro da informação e a difusão do conhecimento. *Ci. Inf*, Brasília, v. 31, n. 2, p. 152 – 162, Mai/Ago 2002. ISSN 1518-8353. Disponível em: <http://dx.doi.org/10.1590/S0100-19652002000200016>. Acesso em: 12.nov.2019.

WANG, J. et al. Financial time series prediction using elman recurrent random neural networks. **Computational Intelligence and Neuroscience**, 2016.

WU, S.; LU, R. Combining artificial neural networks and statistics for stock-market forecasting. In: **Proceedings of the 21st Annual ACM Computer Science Conference**. Nova York, NY: [s.n.], 1993. p. 257 – 264.

YIN, S. et al. Noisy Training For Deep Neural Network In Speech Recognition. **EURASIP Journal on Audio, Speech, and Music Processing**, p. 1 – 14, Dezembro 2015.

ZHANG, Y.; WALLACE, B. C. A Sensitivity Analysis Of (Practitioners' Guide To) Convolution Neural Network For Sentence Classification. In: **Proceedings of the 8th International Joint Conference on Natural Language Processing**. [S.l.: s.n.], 2016. p. 253 – 263.

Apêndices

APÊNDICE A – CÓDIGO FONTES DAS APLICAÇÕES UTILIZADAS

Neste Apêndice encontram-se os códigos fontes das aplicações utilizadas neste trabalho tanto para a criação do robô de negociação, quanto para a análise dos dados.

• CÓDIGO FONTE PARA PROGRAMAÇÃO DO ROBÔ DE NEGOCIAÇÃO

```
//+-----+
//| tese_mestrado.mq5|
//| Copyright 2016, MetaQuotes Software Corp. |
//| https://www.mql5.com |
//+-----+
#property copyright "Copyright 2016, Vinicius Barbosa Paiva."
#property link "https://www.mql5.com"
#property version "1.00"
#include <Trade\Trade.mqh>;
CTrade Trade;
#include <Trade\TradeAlt.mqh>;
CTradeAlt TradeAlt;
#include <Trade\Pending.mqh>;
CPending Pending;
#include <Trade\lib_cisnewbar.mqh>;
CisNewBar current_chart;
#include <Trade\PositionInfo.mqh>;
CPositionInfo Position;
#include <Trade\AccountInfo.mqh>;
CAccountInfo Account;
#include <Trade\OrderInfo.mqh>;
COrderInfo OrderInfor;
#include <Trade\DealInfo.mqh>;
CDealInfo DealInfor;
#include <ChartObjects\ChartObject.mqh>;
CChartObject ChartObject;
#include <errordescription.mqh>
#include <MQL4MarketInfo.mqh>
#define MAX_RETRIES 5 // Max retries on error
#define RETRY_DELAY 3000 // Retry delay in ms
//— input parameters
sinput string Setup; // Which setup will use?
input bool titao= true; // Setup Titao?
```

```

input bool titao_tese= false;// Setup Tito Tese?
input bool titao_scalpe= false;// Setup Tito Scaper?
input bool nove_tres= false;// Setup 9.3?
input string BB; // Bollinger Bands
input int BandsPeriod= 20;
input int BandsShift = 0;
input double BandsDeviation=2;
input ENUM_APPLIED_PRICE BandsPrice=PRICE_CLOSE;
input string M20; // High Moving Average
input int MAPeriod20= 20; // Period High MA
input int MASHift20 = 0; // Shift High MA
input ENUM_MA_METHOD MAMethod20=MODE_SMA; // Method High MA
input ENUM_TIMEFRAMES MATime20 = PERIOD_M2; // TimeFrame High MA
input ENUM_APPLIED_PRICE MAPrice20=PRICE_HIGH; // Applied Price High MA
input string M17; // Low Moving Average
input int MAPeriod17= 20; // Period Low MA
input int MASHift17 = 0; // Shift Low MA
input ENUM_MA_METHOD MAMethod17=MODE_SMA; // Method Low MA
input ENUM_TIMEFRAMES MATime17 = PERIOD_M2; // TimeFrame Low MA
input ENUM_APPLIED_PRICE MAPrice17=PRICE_LOW; // Applied Price Low MA
input string M34; // Activation Moving Average
input int MAPeriod34= 9; // Period Activation MA
input int MASHift34 = 0; // Shift Activation MA
input ENUM_MA_METHOD MAMethod34=MODE_EMA; // Method Activation MA
input ENUM_TIMEFRAMES MATime34 = PERIOD_M2; // TimeFrame Activation MA
input ENUM_APPLIED_PRICE MAPrice34=PRICE_CLOSE; // Applied Price Activation MA
input string VolumeEa; // Define Volume
input double Volume=1;
input string Stop_Type; // Which Type of Stop Loss, point, price or band?
input string Stop_Style= "price";
input string P; // Period To Get High and Low Price When Stop_Style equals Price
input ENUM_TIMEFRAMES PP = PERIOD_M5;
input string StopLoss; // When Stop_Style equals Point
input int SL=50000;
input string TakeProfit;
input int TP=500;
input string Contrato; // WDO ou WIN?
input string Ativo="wdo"; // Digit Code (WDO or WIN)

```

```

input string BE; // Break Even
input bool UseBreakEven=false; // Use Breakeven?
input int BreakEvenProfit=150;
input int LockProfit=0;
input string TS; // Trailing Stop
input bool UseTrailingStop=false;
input string TrailingStop_Type; // Which Type of Trailing Stop, point, price, moving
average or band?
input string Type= "moving";
input int TrailingStop=300;
input int MinimumProfit=150;
input string Time;
input string StartHour="09:30";
input string EndHour="17:00";
input bool CloseAtEnd=true;
input string Media; // Exit Rule
input bool UseMovingAverage= true;
input string Maximum; // Maximum Profit/Loss
input bool UseLimit=true;
input double MaximumProfit=5;
input double MaximumLoss=-1000;
input string MG; // Martingale Activation
input bool Martingale = true;
input string MGType= "yes"; //Agressive martingale, yes or no?
input int MagicNumber=53435;
string IndiSymbol="";
string lastPosition;
ulong last_ticket = -1;
int TradeNow;
int AccountNumber=50390953;
double last_trade_profit =0;
static double Opa2;
double Ask,Bid,bsl,btp,btp2,btp3,stp2,stp3,ssl,stp,Upper[],Lower[],middle[],partial_lots,partial_lots
stopbuy, stopsell,
targetbuy, targetsell,VolumeLoss,fan[],Fan, Opa, divisao, divisao1, divisao2, Stop_loss_sell,
Stop_loss_buy, Take_Profit_buy, Take_Profit_Sell, oz[];
datetime StartDate = D'2020.01.01 08:00';
datetime EndDate = D'2020.12.31 20:00';
bool exptime;

```

```

bool Signal = false;
//+-----+
//| Expert initialization function |
//+-----+
int OnInit()
{
    //—
    //— set MagicNumber for your orders identification
    Trade.SetExpertMagicNumber(MagicNumber);
    int deviation=50;
    Trade.SetDeviationInPoints(deviation);
    IndiSymbol=Symbol();
    Opa2=0;
    // Selecting Stock
    if(Ativo == "wdo")
    {divisao = 1000;
    divisao1 = 5;
    divisao2 = 0.5;}
    if(Ativo == "win")
    {divisao = 1;
    divisao1 = 1;
    divisao2 = 5;}
    /*
    if(TimeCurrent()>StartDate && TimeCurrent()<EndDate){exptime=false;}else{exptime=true;}
    if(exptime==true)
    {
        Alert("Hiring time expired. Please, contact the developer Vinicius B. Paiva");
        ExpertRemove();
    }
    else
    {return(0);}
    */
    if
    (AccountInfoInteger(ACCOUNT_LOGIN)!=AccountNumber)
    {
        Alert("The current Account ID:"+IntegerToString(AccountInfoInteger(ACCOUNT_LOGIN))+“
is not registered with this EA.”);
        ExpertRemove();
    }

```

```

else
{return(0);}
//—
return(INIT_SUCCEEDED);
}
//+-----+
//| Expert deinitialization function |
//+-----+
void OnDeinit(const int reason)
{
}
//+-----+
//| Expert Ontrade function |
//+-----+
void OnTrade()
{
static long nDaysSince1970 = 0;
long n = (long)TimeCurrent() / (86400);
if ( n > nDaysSince1970 )
{
nDaysSince1970 = n;
Opa2 = 0;
}
static int previous_open_positions = 0;
int current_open_positions = PositionsTotal();
if(current_open_positions < previous_open_positions) // a position just got closed:
{
previous_open_positions = current_open_positions;
HistorySelect(TimeCurrent()-300, TimeCurrent()); // 5 minutes ago
int All_Deals = HistoryDealsTotal();
if(All_Deals < 1) Print("Some nasty shit error has occurred :s");
// last deal (should be an DEAL_ENTRY_OUT type):
ulong temp_Ticket = HistoryDealGetTicket(All_Deals-1);
string symbol=HistoryDealGetString(temp_Ticket,DEAL_SYMBOL);
ENUM_DEAL_ENTRY entry_type=(ENUM_DEAL_ENTRY)HistoryDealGetInteger(temp_Ticket,D
ENUM_DEAL_TYPE entry_position=(ENUM_DEAL_TYPE)HistoryDealGetInteger(temp_Ticket,D
// here check some validity factors of the position-closing deal
// (symbol, position ID, even MagicNumber if you care...)
if (symbol == _Symbol)

```

```

{
double LAST_TRADE_PROFIT = HistoryDealGetDouble(temp_Ticket , DEAL_PROFIT);
Print("Last Trade Profit : ", DoubleToString(LAST_TRADE_PROFIT));
Opa = LAST_TRADE_PROFIT;
Opa2 += Opa;
}
}

else if(current_open_positions > previous_open_positions) // a position just got
opened:
previous_open_positions = current_open_positions;
}
//+-----+
//| Expert tick function |
//+-----+
void OnTick()
{
//—
int period_seconds=PeriodSeconds(_Period); // Number of seconds in current chart
period
datetime new_time=TimeCurrent()/period_seconds*period_seconds; // Time of bar
opening on current chart
datetime new_time2=new_time+period_seconds;
if(current_chart.isNewBar(new_time)) OnNewBar(); // When new bar appears - launch
the NewBar event handler
// Taking Pending Orders
ulong tickets[];
Pending.GetTickets(_Symbol, tickets);
int numTickets = ArraySize(tickets);
if(Pending.TotalPending(_Symbol) > 1)
{
Delete(tickets[0]);
}
// Get Bollinger Bands
double bbUpper[], bbLower[], bbMiddle[];
ArraySetAsSeries(bbUpper,true);
ArraySetAsSeries(bbLower,true);
ArraySetAsSeries(bbMiddle,true);
int bbHandle=iCustom(_Symbol,_Period,"Examples\\BB",BandsPeriod,BandsShift,BandsDeviation);
CopyBuffer(bbHandle,0,0,3,bbMiddle);

```



```

CopyBuffer(bbHandle,1,0,3,bbUpper);
CopyBuffer(bbHandle,2,0,3,bbLower);
double bbMid= bbMiddle[1];
double bbUp = bbUpper[1];
double bbUp2 = bbUpper[2];
double bbLow = bbLower[1];
double bbLow2= bbLower[2];
// New rule
if(last_time() >= new_time && last_time() <= new_time2)
Signal = true;
else Signal = false;
// New candle
if(NewCandle()){TradeNow=1;}
Ask = SymbolInfoDouble(Symbol(),SYMBOL_ASK);
Bid = SymbolInfoDouble(Symbol(),SYMBOL_BID);
// Get high 2 minutes
ArraySetAsSeries(High,true);
CopyHigh(_Symbol,PP,0,10,High);
HighCandle=High[1];
// Get low 2 minutes
ArraySetAsSeries(Low,true);
CopyLow(_Symbol,PP,0,10,Low);
LowCandle=Low[1];
if(Stop_Style == "point")
{
Stop_loss_buy = iCloseTito(_Symbol,PERIOD_CURRENT,1)-SL*Point();
Stop_loss_sell = iCloseTito(_Symbol,PERIOD_CURRENT,1)+SL*Point();
Take_Profit_buy = Ask+TP*Point();
Take_Profit_Sell = Bid-TP*Point();
}
if(Stop_Style == "price")
{
Stop_loss_buy = LowCandle - (SL*Point());
Stop_loss_sell = HighCandle + (SL*Point());
Take_Profit_buy = HighCandle + (TP*Point());
Take_Profit_Sell = LowCandle - (TP*Point());
}
if(Stop_Style == "band")
{

```

```

long digits;
double tickSize;
SymbolInfoInteger(_Symbol,SYMBOL_DIGITS,digits);
SymbolInfoDouble(_Symbol,SYMBOL_TRADE_TICK_SIZE,tickSize);
Stop_loss_buy = NormalizeDouble(MathRound(bbLow/tickSize)*tickSize,digits);
Stop_loss_sell = NormalizeDouble(MathRound(bbUp/tickSize)*tickSize,digits);
/*
Stop_loss_buy = LowCandle - (SL*Point());
Stop_loss_sell = HighCandle + (SL*Point());
*/
Take_Profit_buy = (NormalizeDouble(MathRound(bbUp/tickSize)*tickSize,digits)+TP*Point());
Take_Profit_Sell = (NormalizeDouble(MathRound(bbLow/tickSize)*tickSize,digits)-
TP*Point());
}
// Get High moving average
double ma20[];
ArraySetAsSeries(ma20,true);
int ma20Handle=iMA(_Symbol,MATime20,MAPeriod20,MAShift20,MAMethod20,MAPrice20);
CopyBuffer(ma20Handle,0,0,3,ma20);
double currentMA20=ma20[1];
// Get Low moving average
double ma17[];
ArraySetAsSeries(ma17,true);
int ma17Handle=iMA(_Symbol,MATime17,MAPeriod17,MAShift17,MAMethod17,MAPrice17);
CopyBuffer(ma17Handle,0,0,3,ma17);
double currentMA17=ma17[1];
// Get Activation moving average
double ma34[];
ArraySetAsSeries(ma34,true);
int ma34Handle=iMA(_Symbol,MATime34,MAPeriod34,MAShift34,MAMethod34,MAPrice34);
CopyBuffer(ma34Handle,0,0,3,ma34);
double currentMA34=ma34[1];
// Get 34 moving average 2 minutes
double matwo34[];
ArraySetAsSeries(matwo34,true);
int ma34twoHandle=iMA(_Symbol,PERIOD_M2,MAPeriod34,MAShift34,MAMethod34,MAPrice34);
CopyBuffer(ma34twoHandle,0,0,3,matwo34);
double currenttwoMA34=matwo34[1];
// Get 17 moving average 2 minutes

```

```

double matwo17[];
ArraySetAsSeries(matwo17,true);
int ma17twoHandle=iMA(_Symbol,PERIOD_M2,MAPeriod17,MAShift17,MAMethod17,MAPrice1
CopyBuffer(ma17twoHandle,0,0,3,matwo17);
double currenttwoMA17=matwo17[1];
// open position
if(!WorkingHour() && CloseAtEnd)
{CLOSEALL(0);CLOSEALL(1);
for (int i=1; i<numTickets; i++)
{Delete(tickets[i]);}}
if(UseLimit==true)
{
double float_profit=AccountInfoDouble(ACCOUNT_PROFIT);
if (last_profit() >= MaximumProfit || last_profit() <=MaximumLoss || (last_profit()+float_profit)
>= MaximumProfit || (last_profit()+float_profit)<= MaximumLoss)
{
TradeNow = 0;
Print("Profit limit was reached");
Comment("Profit limit was reached");
if(OpenOrders(_Symbol) > 0)
{
CLOSEALL(0);CLOSEALL(1);
}
}
}
// Order Rules
if((WorkingHour() && titao==true /*&& last_position()!="Sell"*/) && ma34[2] < ma20[2]
&& currentMA34 > currentMA20 &&
iCloseTito(_Symbol,PERIOD_CURRENT, 2) < bbUp &&
iCloseTito(_Symbol,PERIOD_CURRENT, 1) < bbUp)
{
if(Martingale == false)
{OrderEntry(0);}
if(Martingale == true)
{
if (MGType == "yes")
{OrderEntry2(0);}
if (MGType == "no")
{OrderEntry3(0);}
}
}

```

```

    }
    }
    if((WorkingHour() && titao==true /*&& last_position()!="Buy"*/ ) && ma34[2] > ma17[2]
&& currentMA34 < currentMA17 &&
    iCloseTito(_Symbol,PERIOD_CURRENT, 2) > bbLow &&
    iCloseTito(_Symbol,PERIOD_CURRENT, 1) > bbLow )
    {
    if(Martingale == false)
    {OrderEntry(1);}
    if(Martingale == true)
    {
    if (MGType == "yes")
    {OrderEntry2(1);}
    if (MGType == "no")
    {OrderEntry3(1);}
    }
    }
    if((WorkingHour() /*&& Signal == false*/ && titao_tese==true && last_position()!="Sell")
&& ma34[2] < ma20[2] && currentMA34 > currentMA20)
    {
    if(Martingale == false)
    {OrderEntry(0);}
    if(Martingale == true)
    {
    if (MGType == "yes")
    {OrderEntry2(0);}
    if (MGType == "no")
    {OrderEntry3(0);}
    }
    }
    if((WorkingHour() /*&& Signal == false*/&& titao_tese==true && last_position()!="Buy")
&& ma34[2] > ma17[2] && currentMA34 < currentMA17)
    {
    if(Martingale == false)
    {OrderEntry(1);}
    if(Martingale == true)
    {
    if (MGType == "yes")
    {OrderEntry2(1);}

```

```

    if (MGType == "no")
    {OrderEntry3(1);}
    }
    }

    if((WorkingHour() && nove_tres == true) && nove_ponto_tres_compra()==true &&
currentMA34 > currentMA20)
    {
        if(Martingale == false)
        {PendingOrderEntry(0);}
        if(Martingale == true)
        {
            if (MGType == "yes")
            {PendingOrderEntry2(0);}
            if (MGType == "no")
            {PendingOrderEntry3(0);}
        }
    }

    if((WorkingHour() && nove_tres == true) && nove_ponto_tres_venda()==true &&
currentMA34 < currentMA17)
    {
        if(Martingale == false)
        {PendingOrderEntry(1);}
        if(Martingale == true)
        {
            if (MGType == "yes")
            {PendingOrderEntry2(1);}
            if (MGType == "no")
            {PendingOrderEntry3(1);}
        }
    }

    if((WorkingHour() && titao_scalpe == true) && currentMA34 < currentMA20
&& ma34[2] > ma20[2] && iCloseTito(_Symbol,PERIOD_CURRENT,1) < currentMA20
&& iCloseTito(_Symbol,PERIOD_CURRENT,1) > currentMA17 )
    {
        if(Martingale == false)
        {//Take_Profit_buy = Ask+TP*Point();
        PendingOrderEntry(0);}
        if(Martingale == true)
        {

```

```

    if (MGType == "yes")
    {PendingOrderEntry2(0);}
    if (MGType == "no")
    {PendingOrderEntry3(0);}
  }
}

if((WorkingHour() && titao_scalpe == true) && ma34[2] < ma17[2]
&& currentMA34 > currentMA17 && iCloseTito(_Symbol,PERIOD_CURRENT,1) >
currentMA17
&& iCloseTito(_Symbol,PERIOD_CURRENT,1) < currentMA20)
{
  if(Martingale == false)
  //{Take_Profit_Sell = Bid-TP*Point();
  PendingOrderEntry(1);}
  if(Martingale == true)
  {
    if (MGType == "yes")
    {PendingOrderEntry2(1);}
    if (MGType == "no")
    {PendingOrderEntry3(1);}
  }
}

// Exit Rule
bool mediaSignal=false;
if(((PositionTypeTito(_Symbol) == POSITION_TYPE_BUY) && (ma34[2] > ma17[2]
&& currentMA34 < currentMA17))
|| ((PositionTypeTito(_Symbol) == POSITION_TYPE_SELL) && (ma34[2] < ma20[2]
&& currentMA34 > currentMA20)))
{
  mediaSignal=true;
}
if(UseMovingAverage==true && mediaSignal==true){CLOSEALL(0);CLOSEALL(1);}
// Trailing Stop Price
if(UseTrailingStop==true && PositionTypeTito(_Symbol)!=-1 && Type=="price")
{
  if(PositionTypeTito() == POSITION_TYPE_BUY)
  {TrailingStopPrice(_Symbol,LowCandle,MinimumProfit);}
  if(PositionTypeTito() == POSITION_TYPE_SELL)
  {TrailingStopPrice(_Symbol,HighCandle,MinimumProfit);}
}

```

```

}
// Tralling Stop Moving Average
if(UseTrailingStop==true && PositionTypeTito(_Symbol)!=-1 && Type=="moving")
{if(PositionTypeTito() == POSITION_TYPE_BUY)
{TrailingStopPrice(_Symbol,ma20[1],MinimumProfit);}
if(PositionTypeTito() == POSITION_TYPE_SELL)
{TrailingStopPrice(_Symbol,ma17[1],MinimumProfit);}
}
// Tralling Stop Bolling
if(UseTrailingStop==true && PositionTypeTito(_Symbol)!=-1 && Type=="band")
{if(PositionTypeTito() == POSITION_TYPE_BUY)
{TrailingStopBand(_Symbol,bbLow,MinimumProfit);}
if(PositionTypeTito() == POSITION_TYPE_SELL)
{TrailingStopBand(_Symbol,bbUp,MinimumProfit);}
}
// Tralling Stop Points
if(UseTrailingStop==true && PositionTypeTito(_Symbol)!=-1 && Type=="point")
{TrailingStopTito(_Symbol,TrailingStop,MinimumProfit);}
// Break even
if(UseBreakEven==true && PositionTypeTito(_Symbol)!=-1)
{
BreakEven(_Symbol,BreakEvenProfit,LockProfit);
}
string com="";
//— Symbol description
string symbol=SymbolInfoString(_Symbol,SYMBOL_DESCRIPTION);
StringAdd(com,"Symbol: "+symbol);
StringAdd(com,"\r\n");
//— Path to symbol
string symbol_path=SymbolInfoString(_Symbol,SYMBOL_PATH);
StringAdd(com,"Path: "+symbol_path);
StringAdd(com,"\r\n");
//— Get account currency
string account_currency=AccountInfoString(ACCOUNT_CURRENCY);
//— Get values of balance and equity
double balance=AccountInfoDouble(ACCOUNT_BALANCE);
double equity=AccountInfoDouble(ACCOUNT_EQUITY);
//— Write values of balance and equity using text formatting
string format=StringFormat("Balance = %.2f, Equity = %.2f",balance,equity);

```

```

StringAdd(com,format);
StringAdd(com,“\r\n”);
//— Get values of margin and profit on account
double margin=AccountInfoDouble(ACCOUNT_MARGIN);
double float_profit=AccountInfoDouble(ACCOUNT_PROFIT);
//— Write values of margin and profit using text formatting
format=StringFormat(“Margin = %.2f %%, Float Profit = %.2f ”,margin,float_profit);
StringAdd(com,format);
StringAdd(com,“\r\n”);
//— Get values of free margin and write them using text formatting
double free_margin=AccountInfoDouble(ACCOUNT_FREEMARGIN);
format=StringFormat(“Free Margin = %G ”,free_margin);
StringAdd(com,format);
StringAdd(com,“\r\n”);
//— Add account currency name to the com string
StringAdd(com,“Account Currency: ”+account_currency);
StringAdd(com,“\r\n”);
//— Get values of latest profit and write them using text formatting
format=StringFormat(“Last Trade Profit : %.2f ”,last_profit_unit());
StringAdd(com,format);
StringAdd(com,“\r\n”);
//— Get values of sum of profit and write them using text formatting
format=StringFormat(“Sum of Profit : %.2f ”,last_profit());
StringAdd(com,format);
StringAdd(com,“\r\n”);
//— Get values of sum of profit and write them using text formatting
StringAdd(com,“Last Deal Position : ”+last_position());
StringAdd(com,“\r\n”);
//— Print the com string on the chart
Comment(com);
}
//+-----+
//|Close price |
//+-----+
double iCloseTito(string symbol,ENUM_TIMEFRAMES timeframe,int index)
{
double Close[];
double close=0;
ArraySetAsSeries(Close,true);

```



```

int copied=CopyClose(symbol,timeframe,0,Bars(symbol,timeframe),Close);
if(copied>0 && index<copied) close=Close[index];
return(close);
}
//+-----+
//|open price |
//+-----+
double iOpenTito(string symbol,ENUM_TIMEFRAMES timeframe,int index)
{
double Open[];
double open=0;
ArraySetAsSeries(Open,true);
int copied=CopyOpen(symbol,timeframe,0,Bars(symbol,timeframe),Open);
if(copied>0 && index<copied) open=Open[index];
return(open);
}
//+-----+
//|New candle |
//+-----+
bool NewCandle()
{
static int BarsOnChart=0;
if(Bars(_Symbol,PERIOD_CURRENT)==BarsOnChart)
return (false);
BarsOnChart=Bars(_Symbol,PERIOD_CURRENT);
return(true);
}
//+-----+
//|Last Time |
//+-----+
datetime last_time()
{
int period_seconds=PeriodSeconds(_Period);
datetime new_time=TimeCurrent()/period_seconds*period_seconds;
double deal_price=0;
datetime date;
// — time interval of the trade history needed
datetime end=new_time+period_seconds; // current server time
datetime start=new_time;// decrease 1 day

```

```

//— request of trade history needed into the cache of MQL5 program
HistorySelect(start,end);
//— get total number of deals in the history
int deals=HistoryDealsTotal();
//— get ticket of the deal with the last index in the list
ulong deal_ticket=HistoryDealGetTicket(deals-1);
if(deal_ticket>0) // deal has been selected, let's proceed ot
{
//— ticket of the order, opened the deal
ulong order=HistoryDealGetInteger(deal_ticket,DEAL_ORDER);
long order_magic=HistoryDealGetInteger(deal_ticket,DEAL_MAGIC);
long pos_ID=HistoryDealGetInteger(deal_ticket,DEAL_POSITION_ID);
date = (datetime)HistoryDealGetInteger(deal_ticket,DEAL_TIME);
deal_price=HistoryDealGetDouble(deal_ticket,DEAL_PRICE);
double deal_volume=HistoryDealGetDouble(deal_ticket,DEAL_VOLUME);
PrintFormat("Deal: #%d opened by order: #%d with ORDER_MAGIC: %d was in
position: #%d price: #%d volume:",
deals-1,order,order_magic,pos_ID,deal_price,deal_volume);
}
else // error in selecting of the deal
{
PrintFormat("Total number of deals %d, error in selection of the deal"+
" with index %d. Error %d",deals,deals-1,GetLastError());
}
return(date);
}
//+-----+
//|Last Profit |
//+-----+
datetime last_profit()
{
// — determine the time intervals of the required trading history
datetime end=StringToTime(TimeToString(TimeCurrent(), TIME_DATE) + " " + EndHour);
datetime gi_time = StringToTime(TimeToString(TimeCurrent(), TIME_DATE) + " " +
StartHour);
datetime start=end-/*gi_time;*/ PeriodSeconds(PERIOD_D1);// set the beginning
time to 24 hours ago
//— request in the cache of the program the needed interval of the trading history
HistorySelect(gi_time,end);

```

```

//— obtain the number of deals in the history
int deals=HistoryDealsTotal();
double result=0;
int returns=0;
double profit=0;
double loss=0;
//— scan through all of the deals in the history
for(int i=0;i<deals;i++)
{
//— obtain the ticket of the deals by its index in the list
ulong deal_ticket=HistoryDealGetTicket(i);
if(deal_ticket>0) // obtain into the cache the deal, and work with it
{
string symbol =HistoryDealGetString(deal_ticket,DEAL_SYMBOL);
datetime time =(datetime)HistoryDealGetInteger(deal_ticket,DEAL_TIME);
ulong order =HistoryDealGetInteger(deal_ticket,DEAL_ORDER);
long order_magic =HistoryDealGetInteger(deal_ticket,DEAL_MAGIC);
long pos_ID =HistoryDealGetInteger(deal_ticket,DEAL_POSITION_ID);
ENUM_DEAL_ENTRY entry_type=(ENUM_DEAL_ENTRY)HistoryDealGetInteger(deal_ticket,DEAL_ENTRY);
if(symbol==_Symbol)
{
if(entry_type==DEAL_ENTRY_OUT)
{
result+=HistoryDealGetDouble(deal_ticket,DEAL_PROFIT);
}
}
}
}
return(result);
}
//+-----+
//|Last Profit |
//+-----+
datetime last_profit_unit()
{
// — determine the time intervals of the required trading history
datetime end=StringToTime(TimeToString(TimeCurrent(), TIME_DATE) + “ ” + EndHour);
datetime gi_time = StringToTime(TimeToString(TimeCurrent(), TIME_DATE) + “ ” +
StartHour);

```

```

datetime start=end-/*gi_time;*/ PeriodSeconds(PERIOD_D1);// set the beginning
time to 24 hours ago
//— request in the cache of the program the needed interval of the trading history
HistorySelect(gi_time,end);
//— obtain the number of deals in the history
int deals=HistoryDealsTotal();
double result=0;
int returns=0;
double profit=0;
double loss=0;
//— scan through all of the deals in the history
for(int i=0;i<deals;i++)
{
//— obtain the ticket of the deals by its index in the list
ulong deal_ticket=HistoryDealGetTicket(i);
if(deal_ticket>0) // obtain into the cache the deal, and work with it
{
string symbol =HistoryDealGetString(deal_ticket,DEAL_SYMBOL);
datetime time =(datetime)HistoryDealGetInteger(deal_ticket,DEAL_TIME);
ulong order =HistoryDealGetInteger(deal_ticket,DEAL_ORDER);
long order_magic =HistoryDealGetInteger(deal_ticket,DEAL_MAGIC);
long pos_ID =HistoryDealGetInteger(deal_ticket,DEAL_POSITION_ID);
ENUM_DEAL_ENTRY entry_type=(ENUM_DEAL_ENTRY)HistoryDealGetInteger(deal_ticket,DE
if(symbol==_Symbol)
{
if(entry_type==DEAL_ENTRY_OUT)
{
result=HistoryDealGetDouble(deal_ticket,DEAL_PROFIT);
}
}
}
}
return(result);
}
//+-----+
//|Function Last Sell or Buy |
//+-----+
string last_position()
{

```

```

// — determine the time intervals of the required trading history
datetime end=StringToTime(TimeToString(TimeCurrent()), TIME_DATE) + “ ” + EndHour);
datetime gi_time = StringToTime(TimeToString(TimeCurrent()), TIME_DATE) + “ ” +
StartHour);
    datetime start=end-/*gi_time;*/ PeriodSeconds(PERIOD_D1);// set the beginning
time to 24 hours ago
    //— request in the cache of the program the needed interval of the trading history
    HistorySelect(gi_time,end);
    //— obtain the number of deals in the history
    int deals=HistoryDealsTotal();
    lastPosition=“No Orders”;
    //— scan through all of the deals in the history
    for(int i=0;i<deals;i++)
    {
        //— obtain the ticket of the deals by its index in the list
        ulong deal_ticket=HistoryDealGetTicket(i);
        if(deal_ticket>0) // obtain into the cache the deal, and work with it
        {
            string symbol =HistoryDealGetString(deal_ticket,DEAL_SYMBOL);
            datetime time =(datetime)HistoryDealGetInteger(deal_ticket,DEAL_TIME);
            ulong order =HistoryDealGetInteger(deal_ticket,DEAL_ORDER);
            long order_magic =HistoryDealGetInteger(deal_ticket,DEAL_MAGIC);
            long pos_ID =HistoryDealGetInteger(deal_ticket,DEAL_POSITION_ID);
            ENUM_DEAL_ENTRY entry_type=(ENUM_DEAL_ENTRY)HistoryDealGetInteger(deal_ticket,DE
            ENUM_DEAL_TYPE entry_position=(ENUM_DEAL_TYPE)HistoryDealGetInteger(deal_ticket,DE
            if(symbol==_Symbol)
            {
                if(entry_type==DEAL_ENTRY_OUT)
                {
                    if(entry_position == DEAL_TYPE_BUY)
                    {
                        lastPosition=“Buy”;
                    }
                    else if(entry_position == DEAL_TYPE_SELL)
                    {
                        lastPosition=“Sell”;
                    }
                }
            }
            else
            {

```

```

lastPosition="No Orders";
}
}
}
}
}
return(lastPosition);
}
//+-----+
//|Working Hour |
//+-----+
bool WorkingHour()
{
    datetime gi_time_01 = StringToTime(TimeToString(TimeCurrent(), TIME_DATE) + " "
+ StartHour);
    datetime gi_time_02 = StringToTime(TimeToString(TimeCurrent(), TIME_DATE) + " "
+ EndHour);
    datetime datetime_0 = TimeCurrent();
    if( gi_time_01 < gi_time_02 && gi_time_01 <= datetime_0 && datetime_0 <=
gi_time_02 ) return (true);
    if( gi_time_01 > gi_time_02 && (datetime_0 >= gi_time_01 || datetime_0 <= gi_time_02)
) return (true);
    return (false);
}
//+-----+
//|Close all |
//+-----+
void CLOSEALL(int type)
{
    int _tp=PositionsTotal();
    for(int i=_tp-1; i>=0; i--)
    {
        string _p_symbol=PositionGetSymbol(i);
        ulong tick=PositionGetTicket(i);
        if(PositionSelectByTicket(tick))
        {
            if(MagicNumber==PositionGetInteger(POSITION_MAGIC))
            {
                if(type==0 && PositionGetInteger(POSITION_TYPE)==POSITION_TYPE_BUY)

```

```

{Trade.PositionClose(tick,-1);}
if(type==1 && PositionGetInteger(POSITION_TYPE)==POSITION_TYPE_SELL)
{
Trade.PositionClose(tick,-1);
}
}
}
}
}
//+-----+
//|Open order |
//+-----+
int OpenOrders(string symbol)
{
int tnum=0;
int _tp=PositionsTotal();
for(int i=_tp-1; i>=0; i--)
{
string _p_symbol=PositionGetSymbol(i);
ulong tick=PositionGetTicket(i);
if(PositionSelectByTicket(tick))
{
if(_p_symbol==symbol)
{
if(MagicNumber==PositionGetInteger(POSITION_MAGIC))
{
tnum++;
}
}
}
}
return(tnum);
}
//+-----+
//|Entry |
//+-----+
void OrderEntry(int direction)
{
//double lotsize=NormalizeDouble(LotSize/4,2);

```

```

if(direction==0 && TradeNow==1 && OpenOrders(Symbol())<1)
{
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Take_Profit_buy;}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(Volume,NULL,Ask,bsl,btp,"EA Trade");
TradeNow=0;
}
if(direction==1 && TradeNow==1 && OpenOrders(Symbol())<1)
{
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(Volume,NULL,Bid,ssl,stp,"EA Trade");
TradeNow=0;
}
}
//+-----+
//|Pending Entry |
//+-----+
void PendingOrderEntry(int direction)
{
//double lotsize=NormalizeDouble(LotSize/4,2);
int period_seconds=PeriodSeconds(_Period); // Number of seconds in current chart
period
datetime new_time=TimeCurrent()/period_seconds*period_seconds; // Time of bar
opening on current chart
datetime expiration=new_time+period_seconds;
if(direction==0 && TradeNow==1 && OpenOrders(Symbol())<1)
{
if(SL>0)
{bsl=Stop_loss_buy;}
else

```



```

    {bsl=0;}
    if(TP>0)
    {btp=Take_Profit_buy;}else{btp=0;}
    btp=NormalizeDouble(btp,Digits());
    ssl=NormalizeDouble(bsl,Digits());
    Trade.BuyStop(Volume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");
    TradeNow=0;
}
if(direction==1 && TradeNow==1 && OpenOrders(Symbol())<1)
{
    if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
    if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
    stp=NormalizeDouble(stp,Digits());
    ssl=NormalizeDouble(ssl,Digits());
    Trade.SellStop(Volume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");
    TradeNow=0;
}
}
//+-----+
//|Pending Entry |
//+-----+
void PendingOrderEntry2(int direction)
{
    //double lotsize=NormalizeDouble(LotSize/4,2);
    int period_seconds=PeriodSeconds(_Period); // Number of seconds in current chart
period
    datetime new_time=TimeCurrent()/period_seconds*period_seconds; // Time of bar
opening on current chart
    datetime expiration=new_time+period_seconds;
    if(direction==0 && TradeNow==1 && OpenOrders(Symbol())<1)
    {
        if(last_profit() >= 0 && last_profit_unit() >= 0)
        {
            if(SL>0)
            {bsl=Stop_loss_buy;}
            else
            {bsl=0;}

```

```

        if(TP>0)
        {btp=Take_Profit_buy;}else{btp=0;}
        btp=NormalizeDouble(btp,Digits());
        ssl=NormalizeDouble(bsl,Digits());
        Trade.BuyStop(Volume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");
        TradeNow=0;
    }
    if (last_profit() >= 0 && last_profit_unit() < 0)
    {
        double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit ;
        double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
        newvolume = MathCeil(newvolume);
        if(SL>0)
        {bsl=Stop_loss_buy;}
        else
        {bsl=0;}
        if(TP>0)
        {btp=Take_Profit_buy;}else{btp=0;}
        btp=NormalizeDouble(btp,Digits());
        ssl=NormalizeDouble(bsl,Digits());
        Trade.BuyStop(newvolume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"
Trade");//newvolume
        TradeNow=0;
    }
    if (last_profit_unit() >= 0 && last_profit() < 0)
    {
        double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
        double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
        newvolume = MathCeil(newvolume);
        if(SL>0)
        {bsl=Stop_loss_buy;}
        else
        {bsl=0;}
        if(TP>0)
        {btp=Take_Profit_buy;}else{btp=0;}
        btp=NormalizeDouble(btp,Digits());
        ssl=NormalizeDouble(bsl,Digits());
        Trade.BuyStop(newvolume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"

```

```

Trade");//newvolume
    TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit_unit()<last_profit())
{
    double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit ;
    double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
    newvolume = MathCeil(newvolume);
    if(SL>0)
    {bsl=Stop_loss_buy;}
    else
    {bsl=0;}
    if(TP>0)
    {btp=Take_Profit_buy;}else{btp=0;}
    btp=NormalizeDouble(btp,Digits());
    ssl=NormalizeDouble(bsl,Digits());
    Trade.BuyStop(newvolume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,“
Trade");//newvolume
    TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()<last_profit_unit())
{
    double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
    double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
    newvolume = MathCeil(newvolume);
    if(SL>0)
    {bsl=Stop_loss_buy;}
    else
    {bsl=0;}
    if(TP>0)
    {btp=Take_Profit_buy;}else{btp=0;}
    btp=NormalizeDouble(btp,Digits());
    ssl=NormalizeDouble(bsl,Digits());
    Trade.BuyStop(newvolume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,“
Trade");//newvolume
    TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()==last_profit_unit())
{

```

```

double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Take_Profit_buy;}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.BuyStop(newvolume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"
Trade");//newvolume
TradeNow=0;
}
}
if(direction==1 && TradeNow==1 && OpenOrders(Symbol())<1)
{
if (last_profit() >= 0 && last_profit_unit() >= 0)
{
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(Volume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");
TradeNow=0;
}
if (last_profit() >= 0 && last_profit_unit() < 0)
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(newvolume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"E
Trade");//newvolume

```

```

TradeNow=0;
}
if (last_profit_unit() >= 0 && last_profit() < 0)
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
Comment(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(newvolume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"E
Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit_unit()<last_profit())
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(newvolume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"E
Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()<last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(newvolume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"E

```

```

Trade");//newvolume
    TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()==last_profit_unit())
{
    double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
    double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
    newvolume = MathCeil(newvolume);
    if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
    if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
    stp=NormalizeDouble(stp,Digits());
    ssl=NormalizeDouble(ssl,Digits());
    Trade.SellStop(newvolume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"E
Trade");//newvolume
    TradeNow=0;
}
}
}
//+-----+
//|Pending Entry |
//+-----+
void PendingOrderEntry3(int direction)
{
    //double lotsize=NormalizeDouble(LotSize/4,2);
    int period_seconds=PeriodSeconds(_Period); // Number of seconds in current chart
period
    datetime new_time=TimeCurrent()/period_seconds*period_seconds; // Time of bar
opening on current chart
    datetime expiration=new_time+period_seconds;
    if(direction==0 && TradeNow==1 && OpenOrders(Symbol())<1)
    {
        if(last_profit() >= 0 && last_profit_unit() >= 0)
        {
            if(SL>0)
            {bsl=Stop_loss_buy;}
            else
            {bsl=0;}
            if(TP>0)
            {btp=Take_Profit_buy;}else{btp=0;}

```

```

        btp=NormalizeDouble(btp,Digits());
        ssl=NormalizeDouble(bsl,Digits());
        Trade.BuyStop(Volume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");
        TradeNow=0;
    }
    if (last_profit() >= 0 && last_profit_unit() < 0)
    {
        double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit ;
        double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
        if(SL>0)
        {bsl=Stop_loss_buy;}
        else
        {bsl=0;}
        if(newtp>0)
        {btp=HighCandle+newtp*Point();}else{btp=0;}
        btp=NormalizeDouble(btp,Digits());
        ssl=NormalizeDouble(bsl,Digits());
        Trade.BuyStop(Volume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
        TradeNow=0;
    }
    if (last_profit_unit() >= 0 && last_profit() < 0)
    {
        double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
        double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
        if(SL>0)
        {bsl=Stop_loss_buy;}
        else
        {bsl=0;}
        if(newtp>0)
        {btp=HighCandle+newtp*Point();}else{btp=0;}
        btp=NormalizeDouble(btp,Digits());
        ssl=NormalizeDouble(bsl,Digits());
        Trade.BuyStop(Volume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
        TradeNow=0;
    }
    if (last_profit() < 0 && last_profit_unit() < 0 && last_profit_unit()<last_profit())

```

```

{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit ;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(newtp>0)
{btp=HighCandle+newtp*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.BuyStop(Volume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()<last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(newtp>0)
{btp=HighCandle+newtp*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.BuyStop(Volume,HighCandle,NULL,bsl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()==last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}

```



```

        if(newtp>0)
        {btp=HighCandle+newtp*Point();}else{btp=0;}
        btp=NormalizeDouble(btp,Digits());
        ssl=NormalizeDouble(ssl,Digits());
        Trade.BuyStop(Volume,HighCandle,NULL,ssl,btp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
        TradeNow=0;
    }
}
if(direction==1 && TradeNow==1 && OpenOrders(Symbol())<1)
{
    if (last_profit() >= 0 && last_profit_unit() >= 0)
    {
        if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
        if(TP>0){stp=Take_Profit_Sell;}else{stp=0;}
        stp=NormalizeDouble(stp,Digits());
        ssl=NormalizeDouble(ssl,Digits());
        Trade.SellStop(Volume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");
        TradeNow=0;
    }
    if (last_profit() >= 0 && last_profit_unit() < 0)
    {
        double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit;
        double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
        if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
        if(newtp>0){stp=LowCandle-newtp*Point();}else{stp=0;}
        stp=NormalizeDouble(stp,Digits());
        ssl=NormalizeDouble(ssl,Digits());
        Trade.SellStop(Volume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
        TradeNow=0;
    }
    if (last_profit_unit() >= 0 && last_profit() < 0)
    {
        double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
        double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
        if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
        if(newtp>0){stp=LowCandle-newtp*Point();}else{stp=0;}
    }
}

```

```

stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(Volume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit_unit()<last_profit())
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(newtp>0){stp=LowCandle-newtp*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(Volume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()<last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(newtp>0){stp=LowCandle-newtp*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(Volume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA
Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()==last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(newtp>0){stp=LowCandle-newtp*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.SellStop(Volume,LowCandle,NULL,ssl,stp,ORDER_TIME_GTC,Time[0]+Period()*60,"EA

```

```

Trade");//newvolume
    TradeNow=0;
}
}
}
//+-----+
//|Position Type |
//+-----+
long PositionTypeTito(string pSymbol=NULL)
{
    if(pSymbol==NULL) pSymbol=_Symbol;
    bool select=PositionSelect(pSymbol);
    if(select == true) return(PositionGetInteger(POSITION_TYPE));
    else return(WRONG_VALUE);
}
//+-----+
//|Trailing Stop |
//+-----+
bool TrailingStopTito(string pSymbol,int pTrailPoints,int pMinProfit=0)
{
    if(PositionSelect(pSymbol)==true && pTrailPoints>0)
    {
        enum ENUM_CHECK_RETCODE
        {
            CHECK_RETCODE_OK,
            CHECK_RETCODE_ERROR,
            CHECK_RETCODE_RETRY
        };
        MqlTradeRequest request;
        MqlTradeResult result;
        ZeroMemory(request);
        request.action = TRADE_ACTION_SLTP;
        request.symbol = pSymbol;
        long posType=PositionGetInteger(POSITION_TYPE);
        double currentStop=PositionGetDouble(POSITION_SL);
        double openPrice=PositionGetDouble(POSITION_PRICE_OPEN);
        double point=SymbolInfoDouble(pSymbol,SYMBOL_POINT);
        int digits=(int)SymbolInfoInteger(pSymbol,SYMBOL_DIGITS);
        double minProfit = pMinProfit * point;
    }
}

```

```

double trailStop = pTrailPoints * point;
currentStop=NormalizeDouble(currentStop,digits);
double trailStopPrice;
double currentProfit;
btp=openPrice+TP*Point();
btp=NormalizeDouble(btp,Digits());
stp=openPrice-TP*Point();
stp=NormalizeDouble(stp,Digits());
// Order loop
int retryCount=0;
int checkRes=0;
do
{
if(posType==POSITION_TYPE_BUY)
{
trailStopPrice = SymbolInfoDouble(pSymbol,SYMBOL_BID) - trailStop;
trailStopPrice = NormalizeDouble(trailStopPrice,digits);
currentProfit=SymbolInfoDouble(pSymbol,SYMBOL_BID)-openPrice;
if(trailStopPrice>currentStop && currentProfit>=minProfit)
{
request.sl= trailStopPrice;
request.tp = btp;
bool sent = OrderSend(request,result);
}
else return(false);
}
else if(posType==POSITION_TYPE_SELL)
{
trailStopPrice = SymbolInfoDouble(pSymbol,SYMBOL_ASK) + trailStop;
trailStopPrice = NormalizeDouble(trailStopPrice,digits);
currentProfit=openPrice-SymbolInfoDouble(pSymbol,SYMBOL_ASK);
if((trailStopPrice<currentStop || currentStop==0) && currentProfit>=minProfit)
{
request.sl= trailStopPrice;
request.tp = stp;
bool sent = OrderSend(request,result);
}
else return(false);
}
}

```

```

checkRes=CheckReturnCode(result.retcode);
if(checkRes==CHECK_RETCODE_OK) break;
else if(checkRes==CHECK_RETCODE_ERROR)
{
string errDesc=TradeServerReturnCodeDescription(result.retcode);
Alert("Trailing stop: Error ",result.retcode," - ",errDesc);
break;
}
else
{
Print("Server error detected, retrying...");
Sleep(RETRY_DELAY);
retryCount++;
}
}
while(retryCount<MAX_RETRIES);
if(retryCount>=MAX_RETRIES)
{
string errDesc=TradeServerReturnCodeDescription(result.retcode);
Alert("Max retries exceeded: Error ",result.retcode," - ",errDesc);
}
string errDesc=TradeServerReturnCodeDescription(result.retcode);
Print("Trailing stop: ",result.retcode," - ",errDesc," Old SL: ",currentStop," New SL:
",request.sl," Bid: ",SymbolInfoDouble(pSymbol,SYMBOL_BID)," Ask: ",SymbolInfoDou-
ble(pSymbol,SYMBOL_ASK)," Stop Level: ",SymbolInfoInteger(pSymbol,SYMBOL_TRADE_STOPS_LE
if(checkRes == CHECK_RETCODE_OK) return(true);
else return(false);
}
else return(false);
}
//+-----+
//|Check Return |
//+-----+
int CheckReturnCode(uint pRetCode)
{
enum ENUM_CHECK_RETCODE
{
CHECK_RETCODE_OK,
CHECK_RETCODE_ERROR,

```

```

CHECK_RETCODE_RETRY
};
int status;
switch(pRetCode)
{
case TRADE_RETCODE_REQUOTE:
case TRADE_RETCODE_CONNECTION:
case TRADE_RETCODE_PRICE_CHANGED:
case TRADE_RETCODE_TIMEOUT:
case TRADE_RETCODE_PRICE_OFF:
case TRADE_RETCODE_REJECT:
case TRADE_RETCODE_ERROR:
status=CHECK_RETCODE_RETRY;
break;
case TRADE_RETCODE_DONE:
case TRADE_RETCODE_DONE_PARTIAL:
case TRADE_RETCODE_PLACED:
case TRADE_RETCODE_NO_CHANGES:
status=CHECK_RETCODE_OK;
break;
default: status=CHECK_RETCODE_ERROR;
}
return(status);
}
//+-----+
//|Break Even |
//+-----+
bool BreakEven(string pSymbol,int pBreakEven,int pLockProfit=0)
{
if(PositionSelect(pSymbol)==true && pBreakEven>0)
{
enum ENUM_CHECK_RETCODE
{
CHECK_RETCODE_OK,
CHECK_RETCODE_ERROR,
CHECK_RETCODE_RETRY
};
MqlTradeRequest request;
MqlTradeResult result;

```

```

ZeroMemory(request);
request.action = TRADE_ACTION_SLTP;
request.symbol = pSymbol;
long posType=PositionGetInteger(POSITION_TYPE);
double currentSL = PositionGetDouble(POSITION_SL);
double openPrice = PositionGetDouble(POSITION_PRICE_OPEN);
double point=SymbolInfoDouble(pSymbol,SYMBOL_POINT);
int digits=(int)SymbolInfoInteger(pSymbol,SYMBOL_DIGITS);
double breakEvenStop;
double currentProfit;
btp=openPrice+TP*Point();
btp=NormalizeDouble(btp,Digits());
stp=openPrice-TP*Point();
stp=NormalizeDouble(stp,Digits());
int retryCount=0;
int checkRes=0;
double bid=0,ask=0;
do
{
if(posType==POSITION_TYPE_BUY)
{
bid=SymbolInfoDouble(pSymbol,SYMBOL_BID);
breakEvenStop = openPrice + (pLockProfit * point);
currentProfit = bid - openPrice;
breakEvenStop = NormalizeDouble(breakEvenStop, digits);
currentProfit = NormalizeDouble(currentProfit, digits);
if(currentSL<breakEvenStop && currentProfit>=pBreakEven*point)
{
request.sl = breakEvenStop;
request.tp = btp;
bool sent=OrderSend(request,result);
}
else return(false);
}
else if(posType==POSITION_TYPE_SELL)
{
ask=SymbolInfoDouble(pSymbol,SYMBOL_ASK);
breakEvenStop = openPrice - (pLockProfit * point);
currentProfit = openPrice - ask;

```

```

breakEvenStop = NormalizeDouble(breakEvenStop, digits);
currentProfit = NormalizeDouble(currentProfit, digits);
if((currentSL>breakEvenStop || currentSL==0) && currentProfit>=pBreakEven*point)
{
    request.sl = breakEvenStop;
    request.tp = stp;
    bool sent=OrderSend(request,result);
}
else return(false);
}
checkRes=CheckReturnCode(result.retcode);
if(checkRes==CHECK_RETCODE_OK) break;
else if(checkRes==CHECK_RETCODE_ERROR)
{
    string errDesc=TradeServerReturnCodeDescription(result.retcode);
    Alert("Break even stop: Error ",result.retcode," - ",errDesc);
    break;
}
else
{
    Print("Server error detected, retrying...");
    Sleep(RETRY_DELAY);
    retryCount++;
}
}
while(retryCount<MAX_RETRIES);
if(retryCount>=MAX_RETRIES)
{
    string errDesc=TradeServerReturnCodeDescription(result.retcode);
    Alert("Max retries exceeded: Error ",result.retcode," - ",errDesc);
}
string errDesc=TradeServerReturnCodeDescription(result.retcode);
Print("Break even stop: ",result.retcode," - ",errDesc," SL: ",request.sl," Bid: ",bid,"
Ask: ",ask," Stop Level: ",SymbolInfoInteger(pSymbol,SYMBOL_TRADE_STOPS_LEVEL));
if(checkRes == CHECK_RETCODE_OK) return(true);
else return(false);
}
else return(false);
}

```



```

//+-----+
//|Ensure Function |
//+-----+
void func_EnsureLotWithinAllowedLimits(double &chng_Lot,string _symbol)
{
//get minimum, maximum and step-size permitted for a lot
double lcl_MinPermittedLot,lcl_MaxPermittedLot,lcl_MinPermittedLotStep;
lcl_MinPermittedLot = MarketInfo( _symbol, MODE_MINLOT );
lcl_MaxPermittedLot = MarketInfo( _symbol, MODE_MAXLOT );
lcl_MinPermittedLotStep=MarketInfo(_symbol,MODE_LOTSTEP);
int _LotDigits=4;
double micro_lot=0.01,mini_lot=0.1,lot1=1;
//Print("Lots1: ",chng_Lot);
if(NormalizeDouble(chng_Lot,_LotDigits)!=NormalizeDouble(MathRound(chng_Lot/lcl_MinPermittedLotStep,_LotDigits))
{
//ensure given lot follows lot-step
chng_Lot=MathRound(chng_Lot/lcl_MinPermittedLotStep)*lcl_MinPermittedLotStep;
//Print("Lots2: ",chng_Lot);
}
if((NormalizeDouble(chng_Lot,_LotDigits)<NormalizeDouble(lcl_MinPermittedLot,_LotDigits)))
{
// lot must not be below the minimum allowed limit
chng_Lot=lcl_MinPermittedLot;
}
else if(NormalizeDouble(chng_Lot,_LotDigits)>NormalizeDouble(lcl_MaxPermittedLot,_LotDigits))
{
// lot must not be above the maximum allowed limit
chng_Lot=lcl_MaxPermittedLot;
}
// normalize the lot
double _LotMicro=0.01,// micro lots
_LotMini=0.10,// mini lots
_LotNrml= 1.00;
//
if(lcl_MinPermittedLot==_LotMicro)
_LotDigits=2;
else if(lcl_MinPermittedLot==_LotMini)
_LotDigits=1;
else if(lcl_MinPermittedLot==_LotNrml)

```

```

_LotDigits=0;
chng_Lot=NormalizeDouble(chng_Lot,_LotDigits);
}
//+-----+
//| New bar event handler function |
//+-----+
void OnNewBar()
{
PrintFormat("New bar: %s",TimeToString(TimeCurrent(),TIME_SECONDS));
}
//+-----+
//| Trailing Stop Price |
//+-----+
bool TrailingStopPrice(string pSymbol,double pTrailPrice,int pMinProfit=0)
{
if(PositionSelect(pSymbol) == true && pTrailPrice > 0)
{
enum ENUM_CHECK_RETCODE
{
CHECK_RETCODE_OK,
CHECK_RETCODE_ERROR,
CHECK_RETCODE_RETRY
};
MqlTradeRequest request;
MqlTradeResult result;
ZeroMemory(request);
request.action = TRADE_ACTION_SLTP;
request.symbol = pSymbol;
long posType = PositionGetInteger(POSITION_TYPE);
double currentStop = PositionGetDouble(POSITION_SL);
double openPrice = PositionGetDouble(POSITION_PRICE_OPEN);
double point = SymbolInfoDouble(pSymbol,SYMBOL_POINT);
long digits;
double tickSize;
SymbolInfoInteger(pSymbol,SYMBOL_DIGITS,digits);
SymbolInfoDouble(pSymbol,SYMBOL_TRADE_TICK_SIZE,tickSize);
/*
int digits = (int)SymbolInfoInteger(pSymbol,SYMBOL_DIGITS);
*/

```

```

double minProfit = pMinProfit * point;
currentStop=NormalizeDouble(MathRound(currentStop/tickSize)*tickSize,digits);
pTrailPrice=NormalizeDouble(MathRound(pTrailPrice/tickSize)*tickSize,digits);
/*
currentStop = NormalizeDouble(currentStop,digits);
pTrailPrice = NormalizeDouble(pTrailPrice,digits);
*/
double currentProfit;
btp=openPrice+TP*Point();
btp=NormalizeDouble(btp,Digits());
stp=openPrice-TP*Point();
stp=NormalizeDouble(stp,Digits());
int retryCount = 0;
int checkRes = 0;
double bid = 0, ask = 0;
do
{
if(posType == POSITION_TYPE_BUY)
{
bid = SymbolInfoDouble(pSymbol,SYMBOL_BID);
currentProfit = bid - openPrice;
if(pTrailPrice > currentStop && currentProfit >= minProfit)
{
request.sl = pTrailPrice;
request.tp = btp;
bool sent = OrderSend(request,result);
}
else return(false);
}
else if(posType == POSITION_TYPE_SELL)
{
ask = SymbolInfoDouble(pSymbol,SYMBOL_ASK);
currentProfit = openPrice - ask;
if((pTrailPrice < currentStop || currentStop == 0) && currentProfit >= minProfit)
{
request.sl = pTrailPrice;
request.tp = stp;
bool sent = OrderSend(request,result);
}
}
}

```

```

else return(false);
}
checkRes = CheckReturnCode(result.retcode);
if(checkRes == CHECK_RETCODE_OK) break;
else if(checkRes == CHECK_RETCODE_ERROR)
{
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Alert("Trailing stop: Error ",result.retcode," - ",errDesc);
break;
}
else
{
Print("Server error detected, retrying...");
Sleep(RETRY_DELAY);
retryCount++;
}
}
while(retryCount < MAX_RETRIES);
if(retryCount >= MAX_RETRIES)
{
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Alert("Max retries exceeded: Error ",result.retcode," - ",errDesc);
}
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Print("Trailing stop: ",result.retcode," - ",errDesc," Old SL: ",currentStop," New SL:
",request.sl," Bid: ",bid," Ask: ",ask," Stop Level: ",SymbolInfoInteger(pSymbol,SYMBOL_TRADE_STOP
if(checkRes == CHECK_RETCODE_OK) return(true);
else return(false);
}
else return(false);
}
//+-----+
//|Function Calcprofit |
//+-----+
bool CalcProfit()
{
// — determine the time intervals of the required trading history
datetime end=StringToTime(TimeToString(TimeCurrent(), TIME_DATE) + " " + EndHour);
datetime gi_time = StringToTime(TimeToString(TimeCurrent(), TIME_DATE) + " " +

```

StartHour);

datetime start=end-/*gi_time;*/ PeriodSeconds(PERIOD_D1);// set the beginning time to 24 hours ago

//— request in the cache of the program the needed interval of the trading history

HistorySelect(gi_time,end);

//— obtain the number of deals in the history

int deals=HistoryDealsTotal();

double result=0;

int returns=0;

double profit=0;

double loss=0;

//— scan through all of the deals in the history

for(int i=0;i<deals;i++)

{

//— obtain the ticket of the deals by its index in the list

ulong deal_ticket=HistoryDealGetTicket(i);

if(deal_ticket>0) // obtain into the cache the deal, and work with it

{

string symbol =HistoryDealGetString(deal_ticket,DEAL_SYMBOL);

datetime time =(datetime)HistoryDealGetInteger(deal_ticket,DEAL_TIME);

ulong order =HistoryDealGetInteger(deal_ticket,DEAL_ORDER);

long order_magic =HistoryDealGetInteger(deal_ticket,DEAL_MAGIC);

long pos_ID =HistoryDealGetInteger(deal_ticket,DEAL_POSITION_ID);

ENUM_DEAL_ENTRY entry_type=(ENUM_DEAL_ENTRY)HistoryDealGetInteger(deal_ticket,DEAL_ENTRY);

//— proceed deal with specified DEAL_MAGIC

if(order_magic==MagicNumber)

{

//... processing of deal with some DEAL_MAGIC

if(symbol==_Symbol)

{

if(entry_type==DEAL_ENTRY_OUT)

returns++;

{

result+=HistoryDealGetDouble(deal_ticket,DEAL_PROFIT);

if(result>0)

{

profit=result;

}

if(result<0)

```

{
    loss=result;
}
if((result>=MaximumProfit) || (result+Position.Profit())>= MaximumProfit))
{
    return(true);
}
if((result<=MaximumLoss) || (result+Position.Profit())<=MaximumLoss))
{
    return(true);
}
}
}
}
}
}
else // unsuccessful attempt to obtain a deal
{
    PrintFormat("We couldn't select a deal, with the index %d. Error %d",
i,GetLastError());
}
}
//— output the results of the calculations
PrintFormat("The total number of %d deals with a financial result. Profit=%.2f , Loss=
%.2f",
returns,profit,loss,result);
return(false);
}
//+-----+
//|Setup 9.3 Compra |
//+-----+
bool nove_ponto_tres_compra()
{
    double matwo9[];
    ArraySetAsSeries(matwo9,true);
    int ma9twoHandle=iMA(_Symbol,PERIOD_CURRENT,9,MAShift34,MAMethod34,MAPrice34);
    CopyBuffer(ma9twoHandle,0,0,3,matwo9);
    double currenttwoMA9=matwo9[1];
    if(LowCandle <= currenttwoMA9 && iOpenTito(_Symbol,PERIOD_CURRENT,1) >=
currenttwoMA9 &&

```

```

        iCloseTito(_Symbol,PERIOD_CURRENT,1) > currenttwoMA9 && currenttwoMA9 >
matwo9[2])
    {
        return(true);
    }
    return(false);
}
//+-----+
//|Setup 9.3 Venda |
//+-----+
bool nove_ponto_tres_venda()
{
    double matwo9[];
    ArraySetAsSeries(matwo9,true);
    int ma9twoHandle=iMA(_Symbol,PERIOD_CURRENT,9,MAShift34,MAMethod34,MAPrice34);
    CopyBuffer(ma9twoHandle,0,0,3,matwo9);
    double currenttwoMA9=matwo9[1];
    if(HighCandle >= currenttwoMA9 && iOpenTito(_Symbol,PERIOD_CURRENT,1) <=
currenttwoMA9 &&
        iCloseTito(_Symbol,PERIOD_CURRENT,1) < currenttwoMA9 && currenttwoMA9 <
matwo9[2])
    {
        return(true);
    }
    return(false);
}
//+-----+
//|Log Trade Request |
//+-----+
void LogTradeRequest()
{
    MqlTradeRequest request;
    MqlTradeResult result;
    Print("MqlTradeRequest - action:",request.action," , comment:",request.comment," ,
deviation:",request.deviation," , expiration:",request.expiration," , magic:",request.magic," , or-
der:",request.order," , position:",request.position," , position_by:",request.position_by," , price:",request.pric
ls:",request.sl," , stoplimit:",request.stoplimit," , symbol:",request.symbol," , tp:",request.tp," ,
type:",request.type," , type_filling:",request.type_filling," , type_time:",request.type_time," , vo-
lume:",request.volume);

```

```

Print("MqlTradeResult - ask:",result.ask," bid:",result.bid," comment:",result.comment,"
deal:",result.deal," order:",result.order," price:",result.price," request_id:",result.request_id,"
retcode:",result.retcode," retcode_external:",result.retcode_external," volume:",result.volume);
}
//+-----+
//|Delete Pending Order |
//+-----+
bool Delete(ulong pTicket)
{
enum ENUM_CHECK_RETCODE
{
CHECK_RETCODE_OK,
CHECK_RETCODE_ERROR,
CHECK_RETCODE_RETRY
};
MqlTradeRequest request;
MqlTradeResult result;
ZeroMemory(request);
ZeroMemory(result);
request.action = TRADE_ACTION_REMOVE;
request.order = pTicket;
// Order loop
int retryCount = 0;
int checkCode = 0;
do
{
bool sent = OrderSend(request,result);
checkCode = CheckReturnCode(result.retcode);
if(checkCode == CHECK_RETCODE_OK) break;
else if(checkCode == CHECK_RETCODE_ERROR)
{
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Alert("Delete order: Error ",result.retcode," - ",errDesc);
LogTradeRequest();
break;
}
else
{
Print("Server error detected, retrying...");

```



```

Sleep(RETRY_DELAY);
retryCount++;
}
}
while(retryCount < MAX_RETRIES);
if(retryCount >= MAX_RETRIES)
{
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Alert("Max retries exceeded: Error ",result.retcode," - ",errDesc);
}
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Print("Delete order #",pTicket," : ",result.retcode," - ",errDesc);
if(checkCode == CHECK_RETCODE_OK)
{
Comment("Pending order ",pTicket," deleted");
return(true);
}
else return(false);
}
//+-----+
//|Entry |
//+-----+
void OrderEntry2(int direction)
{
//double lotsize=NormalizeDouble(LotSize/4,2);
if(direction==0 && TradeNow==1 && OpenOrders(Symbol())<1)
{
if(last_profit() >= 0 && last_profit_unit() >= 0)
{
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Ask+TP*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(Volume,NULL,Ask,bsl,btp,"EA Trade");
TradeNow=0;

```

```

}
if (last_profit() >= 0 && last_profit_unit() < 0)
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit ;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Ask+TP*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(newvolume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit_unit() >= 0 && last_profit() < 0)
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Ask+TP*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(newvolume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit_unit()<last_profit())
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit ;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0)

```

```

{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Ask+TP*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(newvolume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()<last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Ask+TP*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(newvolume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()==last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Ask+TP*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());

```

```

Trade.Buy(newvolume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
}
if(direction==1 && TradeNow==1 && OpenOrders(Symbol())<1)
{
if (last_profit() >= 0 && last_profit_unit() >= 0)
{
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Bid-TP*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(Volume,NULL,Bid,ssl,stp,"EA Trade");
TradeNow=0;
}
if (last_profit() >= 0 && last_profit_unit() < 0)
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Bid-TP*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(newvolume,NULL,Bid,ssl,stp,"EA Trade"); //newvolume
TradeNow=0;
}
if (last_profit_unit() >= 0 && last_profit() < 0)
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
Comment(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Bid-TP*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(newvolume,NULL,Bid,ssl,stp,"EA Trade");//newvolume
TradeNow=0;
}

```

```

}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit_unit()<last_profit())
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Bid-TP*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(newvolume,NULL,Bid,ssl,stp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()<last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Bid-TP*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(newvolume,NULL,Bid,ssl,stp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()==last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newvolume = (newprofit/divisao1*divisao2)/(TP/divisao);
newvolume = MathCeil(newvolume);
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Bid-TP*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(newvolume,NULL,Bid,ssl,stp,"EA Trade");//newvolume
TradeNow=0;
}
}
}
}

```

```

//+-----+
//|Entry |
//+-----+
void OrderEntry3(int direction)
{
//double lotsize=NormalizeDouble(LotSize/4,2);
if(direction==0 && TradeNow==1 && OpenOrders(Symbol())<1)
{
if(last_profit() >= 0 && last_profit_unit() >= 0)
{
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(TP>0)
{btp=Ask+TP*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(Volume,NULL,Ask,bsl,btp,"EA Trade");
TradeNow=0;
}
if (last_profit() >= 0 && last_profit_unit() < 0)
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit ;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(newtp>0)
{btp=Ask+newtp*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(Volume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit_unit() >= 0 && last_profit() < 0)
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;

```

```

double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(newtp>0)
{btp=Ask+newtp*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(Volume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit_unit()<last_profit())
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit ;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(newtp>0)
{btp=Ask+newtp*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(Volume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()<last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(newtp>0)
{btp=Ask+newtp*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());

```

```

Trade.Buy(Volume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()==last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit ;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0)
{bsl=Stop_loss_buy;}
else
{bsl=0;}
if(newtp>0)
{btp=Ask+newtp*Point();}else{btp=0;}
btp=NormalizeDouble(btp,Digits());
ssl=NormalizeDouble(bsl,Digits());
Trade.Buy(Volume,NULL,Ask,bsl,btp,"EA Trade");//newvolume
TradeNow=0;
}
}
if(direction==1 && TradeNow==1 && OpenOrders(Symbol())<1)
{
if (last_profit() >= 0 && last_profit_unit() >= 0)
{
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(TP>0){stp=Bid-TP*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(Volume,NULL,Bid,ssl,stp,"EA Trade");
TradeNow=0;
}
if (last_profit() >= 0 && last_profit_unit() < 0)
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(newtp>0){stp=Bid-newtp*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(Volume,NULL,Bid,ssl,stp,"EA Trade"); //newvolume

```



```

TradeNow=0;
}
if (last_profit_unit() >= 0 && last_profit() < 0)
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(newtp>0){stp=Bid-newtp*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(Volume,NULL,Bid,ssl,stp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit_unit()<last_profit())
{
double newprofit = (MaximumProfit - (last_profit_unit())) + MaximumProfit;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(newtp>0){stp=Bid-newtp*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(Volume,NULL,Bid,ssl,stp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()<last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;
if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(newtp>0){stp=Bid-newtp*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(Volume,NULL,Bid,ssl,stp,"EA Trade");//newvolume
TradeNow=0;
}
if (last_profit() < 0 && last_profit_unit() < 0 && last_profit()==last_profit_unit())
{
double newprofit = (MaximumProfit - (last_profit())) + MaximumProfit;
double newtp = (((newprofit*divisao2)/divisao1)*divisao)/Volume;

```

```

if(SL>0){ssl=Stop_loss_sell;}else{ssl=0;}
if(newtp>0){stp=Bid-newtp*Point();}else{stp=0;}
stp=NormalizeDouble(stp,Digits());
ssl=NormalizeDouble(ssl,Digits());
Trade.Sell(Volume,NULL,Bid,ssl,stp,"EA Trade");//newvolume
TradeNow=0;
}
}
}
//+-----+
//|Trailing Stop Band |
//+-----+
bool TrailingStopBand(string pSymbol,double pTrailPrice,int pMinProfit=0)
{
if(PositionSelect(pSymbol) == true && pTrailPrice > 0)
{
enum ENUM_CHECK_RETCODE
{
CHECK_RETCODE_OK,
CHECK_RETCODE_ERROR,
CHECK_RETCODE_RETRY
};
MqlTradeRequest request;
MqlTradeResult result;
ZeroMemory(request);
request.action = TRADE_ACTION_SLTP;
request.symbol = pSymbol;
long posType = PositionGetInteger(POSITION_TYPE);
double currentStop = PositionGetDouble(POSITION_SL);
double currentTStop = PositionGetDouble(POSITION_TP);
double openPrice = PositionGetDouble(POSITION_PRICE_OPEN);
double point = SymbolInfoDouble(pSymbol,SYMBOL_POINT);
long digits;
double tickSize;
SymbolInfoInteger(pSymbol,SYMBOL_DIGITS,digits);
SymbolInfoDouble(pSymbol,SYMBOL_TRADE_TICK_SIZE,tickSize);
/*
int digits = (int)SymbolInfoInteger(pSymbol,SYMBOL_DIGITS);
*/

```

```

double minProfit = pMinProfit * point;
currentStop=NormalizeDouble(MathRound(currentStop/tickSize)*tickSize,digits);
pTrailPrice=NormalizeDouble(MathRound(pTrailPrice/tickSize)*tickSize,digits);
/*
currentStop = NormalizeDouble(currentStop,digits);
pTrailPrice = NormalizeDouble(pTrailPrice,digits);
*/
double currentProfit;
btp=openPrice+TP*Point();
btp=NormalizeDouble(btp,Digits());
stp=openPrice-TP*Point();
stp=NormalizeDouble(stp,Digits());
int retryCount = 0;
int checkRes = 0;
double bid = 0, ask = 0;
do
{
if(posType == POSITION_TYPE_BUY)
{
bid = SymbolInfoDouble(pSymbol,SYMBOL_BID);
currentProfit = bid - openPrice;
if(pTrailPrice > currentStop && currentProfit >= minProfit)
{
request.sl = pTrailPrice;
request.tp = currentTStop;
bool sent = OrderSend(request,result);
}
else return(false);
}
else if(posType == POSITION_TYPE_SELL)
{
ask = SymbolInfoDouble(pSymbol,SYMBOL_ASK);
currentProfit = openPrice - ask;
if((pTrailPrice < currentStop || currentStop == 0) && currentProfit >= minProfit)
{
request.sl = pTrailPrice;
request.tp = currentTStop;
bool sent = OrderSend(request,result);
}
}
}

```

```

else return(false);
}
checkRes = CheckReturnCode(result.retcode);
if(checkRes == CHECK_RETCODE_OK) break;
else if(checkRes == CHECK_RETCODE_ERROR)
{
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Alert("Trailing stop: Error ",result.retcode," - ",errDesc);
break;
}
else
{
Print("Server error detected, retrying...");
Sleep(RETRY_DELAY);
retryCount++;
}
}
while(retryCount < MAX_RETRIES);
if(retryCount >= MAX_RETRIES)
{
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Alert("Max retries exceeded: Error ",result.retcode," - ",errDesc);
}
string errDesc = TradeServerReturnCodeDescription(result.retcode);
Print("Trailing stop: ",result.retcode," - ",errDesc," Old SL: ",currentStop," New SL:
",request.sl," Bid: ",bid," Ask: ",ask," Stop Level: ",SymbolInfoInteger(pSymbol,SYMBOL_TRADE_STOP
if(checkRes == CHECK_RETCODE_OK) return(true);
else return(false);
}
else return(false);
}
//+-----+

```

• CÓDIGO FONTE GERADO PARA A ANÁLISE DOS DADOS

```

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import warnings
import calendar

```

```

import seaborn as sns
sns.set(style='white', palette='deep')
plt.style.use('grayscale')
warnings.filterwarnings('ignore')
width = 0.35
# Funções
def autolabel(rects,ax, df): #autolabel
for rect in rects:
height = rect.get_height()
ax.annotate('{} ({:.2f}%)'.format(height, height*100/df.shape[0]),
xy = (rect.get_x() + rect.get_width()/2, height),
xytext= (0,3),
textcoords="offset points",
ha='center', va='bottom', fontsize=15)
def autolabel_without_pct(rects,ax): #autolabel
for rect in rects:
height = rect.get_height()
ax.annotate('{}'.format(height),
xy = (rect.get_x() + rect.get_width()/2, height),
xytext= (0,3),
textcoords="offset points",
ha='center', va='bottom', fontsize=15)
def autolabel_horizontal(rects,ax):
"""
Attach a text label above each bar displaying its height
"""
for rect in rects:
width = rect.get_width()
ax.text(rect.get_x() + rect.get_width()+3, rect.get_y() + rect.get_height()/2.,
'%.2f' % width,
ha='center', va='center', color='black', fontsize=15)
# Importando o Arquivo
df = pd.read_excel('Banco de Dados - WDO.xlsx')
# Verificando Null Values
df.isnull().sum()
null_values = (df.isnull().sum()/len(df)*100)
null_values = pd.DataFrame(null_values, columns= ['% Null Values'])
null_values
# Deletando Null Values

```

```

df_feature = df.copy()
df_feature.dropna(inplace=True)
df_feature.isnull().sum()
# Alterando nome de colunas
bank = ['corretora_', 'bank_']
letters = ['abcdefghijklmnopqrstuvwxyz']
new_columns = np.array([])
for i in bank:
    for j in range(0,4):
        new_columns = np.append(new_columns, i+list(letters[0])[j])
df_feature.columns
count = 0
for i in df_feature.loc[:, ['win_xp_(5m)', 'win_rico_(5m)', 'win_clear_(5m)',
'win_modal_(5m)', 'win_ubs_(5m)', 'win_btg_(5m)', 'win_bradesco_(5m)',
'win_genial(5m)']]:
    df_feature.rename(columns={i:new_columns[count]}, inplace=True)
    count+=1
# Verificando erro de digitação
df_feature.columns
df_feature.set_index('data', inplace=True)
max_value = np.array([])
min_value = np.array([])
max_index = np.array([])
min_index = np.array([])
max_time = np.array([])
min_time = np.array([])
count = 0
value_error_final = pd.DataFrame()
for i in df_feature.loc[:,['abertura', 'maxima', 'minima',
'fechamento', '20mma_maxima_2m', '20mma_minima_2m', '9mme_fechamento',
'200mma_fechamento', '20mma_maxima_5m', '20mma_minima_5m',
'volume_financeiro', 'corretora_a', 'corretora_b', 'corretora_c',
'corretora_d', 'bank_a', 'bank_b', 'bank_c', 'bank_d', 'gain', 'quantas_correcoes',
'quantos_pontos_avancou', 'quantos_pontos_retornados']]:
    max_value = np.append(max_value, df_feature[i].max())
    min_value = np.append(min_value, df_feature[i].min())
    max_index = np.append(max_index, df_feature.loc[:,i].idxmax())
    min_index = np.append(min_index, df_feature.loc[:,i].idxmin())
    max_time = np.append(max_time, df_feature[df_feature[i] == df_feature[i].max()][0]['horario'])

```

```

min_time = np.append(min_time,df_feature[df_feature[i] == df_feature[i].min()][‘horario’])
print(‘O máximo valor para a coluna |{ }| foi de { }, no dia { } e no horário { }’.for-
mat(i,max_value[count],
max_index[count],max_time[count]))
print(‘O mínimo valor para a coluna |{ }| foi de { }, no dia { } e no horário { }’.for-
mat(i,min_value[count],
min_index[count], min_time[count]))
print(‘***100)
valer_error = pd.DataFrame({'valor_max':[max_value[count]],
'dia_max': [max_index[count]],
'horario_max': [max_time[count]],
'valor_min':[min_value[count]],
'dia_min': [min_index[count]],
'horario_min': [min_time[count]]}, index=[i])
value_error_final = pd.concat([valer_error,value_error_final])
count+=1
df_feature = df_feature.drop(‘gain’, axis=1)
#Pela amplitude podemos verificar erros de digitação nas colunas |máximas| e
|mínimas|
df_feature[‘amplitude’] = df_feature[‘maxima’]-df_feature[‘minima’] # Criando coluna
amplitude
amplitude_error = df_feature[df_feature[‘amplitude’] <0][[‘maxima’, ‘minima’, ‘hora-
rio’]]
#Verificando se a ME9 está menor que as MA20 de ativação
nove_compra_error = df_feature[df_feature[‘tipo_de_negociacao’]==‘compra’][[‘20mma_maxima_2m’,
‘9mme_fechamento’,
‘horario’]]
nove_venda_error = df_feature[df_feature[‘tipo_de_negociacao’]==‘venda’][[‘20mma_minima_2m’,
‘9mme_fechamento’,
‘horario’]]
nove_compra_error[‘error’] = nove_compra_error[‘9mme_fechamento’]-nove_compra_error[‘20mma_maxima_2m’]
nove_venda_error[‘error’] = nove_venda_error[‘9mme_fechamento’]-nove_venda_error[‘20mma_minima_2m’]
nove_compra_error = nove_compra_error[nove_compra_error[‘error’].values<0]
nove_venda_error = nove_venda_error[nove_venda_error[‘error’].values>0]
nove_compra_error[[‘20mma_maxima_2m’, ‘9mme_fechamento’]] = nove_compra_error[[‘9mme_fechamento’,
‘20mma_maxima_2m’]].values)
nove_venda_error[[‘20mma_minima_2m’,‘9mme_fechamento’]] = nove_venda_error[[‘9mme_fechamento’,
‘20mma_minima_2m’]].values)
df_feature.groupby(df_feature.index)[‘horario’].get_group(‘2019-06-03 ’)[0]

```

```

df_feature.groupby(df_feature.index)['horario'].get_group('2019-06-19 ')[0]
df_feature.groupby(df_feature.index)['horario'].value_counts()
for i in range(0, len(nove_compra_error)):
    df_feature.loc[(df_feature.index == nove_compra_error.index[i]) & (df_feature['horario']==nove_co
'20mma_maxima_2m'] = nove_compra_error['20mma_maxima_2m'].values[i]
    df_feature.loc[(df_feature.index == nove_compra_error.index[i]) & (df_feature['horario']==nove_co
'9mme_fechamento'] = nove_compra_error['9mme_fechamento'].values[i]
    for i in range(0, len(nove_venda_error)):
        df_feature.loc[(df_feature.index == nove_venda_error.index[i]) & (df_feature['horario']==nove_ven
'20mma_minima_2m'] = nove_venda_error['20mma_minima_2m'].values[i]
        df_feature.loc[(df_feature.index == nove_venda_error.index[i]) & (df_feature['horario']==nove_ven
'9mme_fechamento'] = nove_venda_error['9mme_fechamento'].values[i]
        nove_venda_error['20mma_minima_2m'][(1)]
        df_feature.loc[(df_feature.index == nove_venda_error.index[1]) & (df_feature['horario']==nove_ven
'20mma_minima_2m']
        #Verificando se M20 2m high tem divergência com M20 2m Low
        df_feature.columns
        m20_error_high_2m = df_feature[df_feature['20mma_maxima_2m']<df_feature['20mma_minima
'20mma_minima_2m', 'horario']]
        m20_error_high_2m['error'] = m20_error_high_2m['20mma_maxima_2m']-m20_error_high_2m['2
        m20_error_high_2m[['20mma_maxima_2m', '20mma_minima_2m']] = m20_error_high_2m[['20m
'20mma_maxima_2m']].where(m20_error_high_2m['error']<0,
        m20_error_high_2m[['20mma_maxima_2m', '20mma_minima_2m']].values)
        for i in range(0, len(m20_error_high_2m)):
            df_feature.loc[(df_feature.index == m20_error_high_2m.index[i]) & (df_feature['horario']==m20_er
'20mma_maxima_2m'] = m20_error_high_2m['20mma_maxima_2m'].values[i]
            df_feature.loc[(df_feature.index == m20_error_high_2m.index[i]) & (df_feature['horario']==m20_er
'20mma_minima_2m'] = m20_error_high_2m['20mma_minima_2m'].values[i]
            #Verificando se M20 5m high tem divergência com M20 5m Low
            df_feature.columns
            m20_error_high_5m = df_feature[df_feature['20mma_maxima_5m']<df_feature['20mma_minima
'20mma_minima_5m', 'horario']]
            m20_error_high_5m['error'] = m20_error_high_5m['20mma_maxima_5m']-m20_error_high_5m['2
            m20_error_high_5m[['20mma_maxima_5m', '20mma_minima_5m']] = m20_error_high_5m[['20m
'20mma_maxima_5m']].where(m20_error_high_5m['error']<0,
            m20_error_high_5m[['20mma_maxima_5m', '20mma_minima_5m']].values)
            for i in range(0, len(m20_error_high_5m)):
                df_feature.loc[(df_feature.index == m20_error_high_5m.index[i]) & (df_feature['horario']==m20_er
'20mma_maxima_5m'] = m20_error_high_5m['20mma_maxima_5m'].values[i]

```



```

df_feature.loc[(df_feature.index == m20_error_high_5m.index[i]) & (df_feature['horario']==m20_error_high_5m['20mma_minima_5m'])] = m20_error_high_5m['20mma_minima_5m'].values[i]
#Salvando planilha tratada
df_feature.to_excel('WDO Tratado.xlsx')
#Quais foram as operações com maior frequência? PLOT
df_feature.columns
df_compra = df_feature[df_feature['tipo_de_negociacao']=='compra']['tipo_de_negociacao']
df_venda = df_feature[df_feature['tipo_de_negociacao']=='venda']['tipo_de_negociacao']
labels = [df_compra.values[0],df_venda.values[0]]
ind = np.arange(len(labels))
values = [len(df_compra), len(df_venda)]
fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(1,1,1)
ax.set_title('Quantidade de Operações de Compra e Venda \n Realizado pela Estratégia de negociação', fontsize=15)
ax.set_xlabel('Tipo de Operação', fontsize=15)
ax.set_ylabel('Quantidade de Operações Realizadas', fontsize=15)
ax.set_xticklabels(['Compra', 'Venda'], fontsize=15)
ax.set_yticklabels(np.arange(0,501,100), fontsize=15)
rects1= ax.bar('Compra', len(df_compra), width, edgecolor='black')
rects2=ax.bar('Venda', len(df_venda), width, edgecolor='black')
ax.set_xticks(ind)
autolabel(rects1,ax,df_feature)
autolabel(rects2,ax,df_feature)
plt.tight_layout()
#Quais foram os dias com maiores operações? PLOT
df_feature.columns
df_compra = df_feature[df_feature['tipo_de_negociacao']=='compra']['horario','tipo_de_negociacao']
df_venda = df_feature[df_feature['tipo_de_negociacao']=='venda']['horario','tipo_de_negociacao']
df_compra['data'] = df_compra.index
df_compra['dia'] = df_compra['data'].apply(lambda x: x.weekday())
df_compra['mes'] = df_compra['data'].apply(lambda x: x.month)
df_compra['hora'] = df_compra['horario'].apply(lambda x: x.hour)
df_venda['data'] = df_venda.index
df_venda['dia'] = df_venda['data'].apply(lambda x: x.weekday())
df_venda['mes'] = df_venda['data'].apply(lambda x: x.month)
df_venda['hora'] = df_venda['horario'].apply(lambda x: x.hour)
dias = {}
for i,v in enumerate(list(calendar.day_name)):

```

```

dias[i]=v
meses = {}
for i,v in enumerate(list(calendar.month_name)[1:],1):
    meses[i]=v
dias_nomes_compra = np.array([])
for i in df_compra['dia']:
    for j in range(0,len(dias)):
        if i == list(dias.keys())[j]:
            dias_nomes_compra = np.append(dias_nomes_compra,dias[j])
dias_nomes_venda = np.array([])
for i in df_venda['dia']:
    for j in range(0,len(dias)):
        if i == list(dias.keys())[j]:
            dias_nomes_venda = np.append(dias_nomes_venda,dias[j])
def compra_venda(x):
    for i in range(6,len(meses)+1):
        if x == i:
            return meses[x]
df_compra['dia'] = dias_nomes_compra
df_venda['dia'] = dias_nomes_venda
df_compra['mes'] = df_compra['mes'].apply(compra_venda )
df_venda['mes'] = df_venda['mes'].apply(compra_venda )
fig = plt.figure(figsize=(15,15))
ax = fig.add_subplot(1,1,1)
labels = np.array([])
for i in range(0,5):
    labels = np.append(labels,dias[i])
len_dia_compra = np.array([])
len_dia_venda = np.array([])
for i in labels:
    len_dia_compra = np.append(len_dia_compra, len(df_compra[df_compra['dia']==i]))
    len_dia_venda = np.append(len_dia_venda, len(df_venda[df_venda['dia']==i]))
ind = np.arange(len(labels))
ax.set_title('Tabela de Operações por Dias', fontsize=15)
ax.set_xticks(ind)
ax.set_xticklabels(['Segunda', 'Terça', 'Quarta', 'Quinta', 'Sexta'], fontsize=15)
ax.set_xlabel('Dias da semana', fontsize=15)
ax.set_yticklabels(np.arange(0,150,20),fontsize=15)
ax.set_ylabel('Quantidade de Operações por Dias', fontsize=15)

```

```

for i in range(0,len(labels)):
    rects1 = ax.bar(ind+width/2, len_dia_compra, width=width, edgecolor='black')
    rects2 = ax.bar(ind-width/2, len_dia_venda, width=width, edgecolor='black')
    ax.legend(['compra','venda' ], fontsize=15, loc='best')
    autolabel_without_pct(rects1,ax)
    autolabel_without_pct(rects2,ax)
    plt.tight_layout()
#Quais foram os mesmos com maiores operações? PLOT
labels = []
len_mes_compra = []
len_mes_venda = []
[labels.append(meses[i]) for i in range(6,13)]
[len_mes_compra.append(len(df_compra[df_compra['mes']==i])) for i in labels]
[len_mes_venda.append(len(df_venda[df_venda['mes']==i])) for i in labels]
ind=np.arange(len(labels))
fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(1,1,1)
ax.set_title('Tabela de Operações por Mês', fontsize=15)
ax.set_xticks(ind)
ax.set_xticklabels(['Junho', 'Julho', 'Agosto', 'Setembro', 'Outubro', 'Novembro', 'De-
zembro'], fontsize=15)
ax.set_yticklabels(np.arange(0,110,10), fontsize=15)
ax.set_ylabel('Quantidade de Operações por Mês', fontsize=15)
for i in range(0,len(len_mes_compra)):
    rects1= ax.bar(ind+width/2, len_mes_compra, width=width, edgecolor='black')
    rects2= ax.bar(ind-width/2, len_mes_venda, width=width, edgecolor='black')
    ax.legend(['compra','venda' ], fontsize=15, loc='best')
    autolabel_without_pct(rects1,ax,)
    autolabel_without_pct(rects2,ax,)
    plt.tight_layout()
#Quais horários obteve mais sinais? PLOT
bins = np.arange(9,18)
time = list(np.arange(9,18))
time_string = [str(time[i]) for i in range(0,len(time))]
len_time_compra = list(df_compra.groupby(pd.cut(df_compra['hora'], bins))['hora'].value_counts())
len_time_venda = list(df_venda.groupby(pd.cut(df_venda['hora'], bins))['hora'].value_counts()).valu
labels=[]
count1=0
count2=1

```

```

while count2 != len(time_string):
    labels.append('['+time_string[count1]+' - '+time_string[count2]+']')
    count1+=1
    count2+=1
ind = np.arange(len(labels))
fig = plt.figure(figsize=(10,10))
ax = fig.add_subplot(1,1,1)
ax.set_title('Tabela de Operações por Horário', fontsize=15)
ax.set_xticklabels(labels, fontsize=15)
ax.set_xticks(ind)
ax.set_xlabel('Horários', fontsize=15)
ax.set_yticklabels(np.arange(0,100,10), fontsize=15)
ax.set_ylabel('Quantidade de Operações por Horário', fontsize=15)
ax.set_yticks(np.arange(0,100,10))
for i in range(0,len(len_time_compra)):
    rects1 = ax.bar(ind+width/2, len_time_compra, width=width, edgecolor='black', color='black')
    rects2 = ax.bar(ind-width/2, len_time_venda, width=width, edgecolor='black', color='gray')
    ax.legend(['compra','venda'],loc='best', fontsize=15)
    autolabel_without_pct(rects1,ax)
    autolabel_without_pct(rects2,ax)
plt.tight_layout()
#Quantos pontos avançaram?. PLOT
estatistica = df_feature.describe()
df_feature.columns
bins = np.arange(0,52,4)
df_feature.groupby(pd.cut(df_feature['quantos_pontos_avancou'], bins))['tipo_de_negociacao'].value_counts()
len_pontos_compra = []
len_pontos_venda = []
for i in range(0, 12):
    grouped = df_feature.groupby(pd.cut(df_feature['quantos_pontos_avancou'], bins))['tipo_de_negociacao'].value_counts()
    grouped_values = df_feature.groupby(pd.cut(df_feature['quantos_pontos_avancou'], bins))['tipo_de_negociacao'].get_group(grouped).value_counts()
    if grouped_values.index[0] == 'compra':
        len_pontos_compra.append(grouped_values.values[0])
    if grouped_values.index[1] == 'compra':
        len_pontos_compra.append(grouped_values.values[1])
    if grouped_values.index[0] == 'venda':

```

```

len_pontos_venda.append(grouped_values.values[0])
if grouped_values.index[1] == 'venda':
len_pontos_venda.append(grouped_values.values[1])
pontos = np.arange(0,49,4)
pontos_string = [str(i) for i in pontos]
count=0
count1=1
labels = []
while count1 != len(pontos_string):
labels.append('['+pontos_string[count]+'-'+pontos_string[count1]+'')
count+=1
count1+=1
ind =np.arange(len(labels))
fig=plt.figure(figsize=(15,15))
ax1 = fig.add_subplot(2,1,1)
ax2 = fig.add_subplot(2,1,2)
ax1.set_title('Gráfico de Pontos Avançados por \nOperação de Compra e Venda',
fontsize=15)
ax1.set_yticklabels(np.arange(0,200,10), fontsize=15)
ax1.set_yticks(np.arange(0,200,10))
ax1.set_ylim(0, 200)
ax1.set_ylabel('Contagem', fontsize=15)
ax1.set_xticklabels(labels, fontsize=15)
ax1.set_xticks(ind)
ax1.set_xlabel('Pontos Avançados', fontsize=15)
rects1 = ax1.bar(ind+width/2,len_pontos_compra, width=width, edgecolor='black',
alpha=0.8, label='Operação de Compra')
rects2 = ax1.bar(ind-width/2,len_pontos_venda, width=width, edgecolor='black', alpha=0.8,
label='Operação de Venda')
autolabel_without_pct(rects1,ax1)
autolabel_without_pct(rects2,ax1)
ax1.legend(loc='best',frameon=False )
plt.tight_layout()
ax2.hist(df_feature['quantos_pontos_avancou'].values, bins=50,
density=False, histtype='barstacked', align='mid', color='black',
alpha=0.5, range=(0,70))
ax2.set_ylabel('Frequência', fontsize=15)
ax2.tick_params(axis='both', labelsiz=15, labelcolor='k', labelrotation=0)
ax2.set_xlabel('Pontos', fontsize=15)

```

```

ax2.set_title('Distribuição dos Pontos Avançados', fontsize=15)
#Média por grupo de pontos. PLOT
df_compra['quantos_pontos_avancou'] = df_feature[df_feature['tipo_de_negociacao']=='compra']
df_venda['quantos_pontos_avancou'] = df_feature[df_feature['tipo_de_negociacao']=='venda']['q
mean_pontos_compra = np.round(list(df_compra.groupby(pd.cut(df_compra['quantos_pontos_av
bins))['quantos_pontos_avancou'].mean().values),decimals=2 )
mean_pontos_venda = np.round(list(df_venda.groupby(pd.cut(df_venda['quantos_pontos_avanco
bins))['quantos_pontos_avancou'].mean().values),decimals=2 )
std_pontos_compra = np.round(list(df_compra.groupby(pd.cut(df_compra['quantos_pontos_avanco
bins))['quantos_pontos_avancou'].std().values),decimals=2 )
std_pontos_venda = np.round(list(df_venda.groupby(pd.cut(df_venda['quantos_pontos_avancou']
bins))['quantos_pontos_avancou'].std().values),decimals=2 )
fig = plt.figure(figsize=(15,15))
ax = fig.add_subplot(1,1,1)
ax.set_title('Média por Grupo dos Pontos Avançados', fontsize=15)
ax.set_yticklabels(labels, fontsize=15)
ax.set_yticks(ind)
ax.set_ylabel('Grupo de Pontos Avançados', fontsize=15)
ax.set_xlabel('Média por Grupo de Pontos Avançados', fontsize=15)
ax.set_xticklabels(np.arange(0,70,10), fontsize=15)
ax.set_xticks(np.arange(0,70,10))
ax.set_xlim(0,60)
rects1 = ax.barh(ind+width/2,mean_pontos_compra, width, edgecolor='black',align='center',
alpha=0.5, label='Operação de Compra', xerr=std_pontos_compra)
rects2 = ax.barh(ind-width/2,mean_pontos_venda, width, edgecolor='black', align='center',
alpha=0.5, label='Operação de Venda', xerr=std_pontos_venda)
ax.legend(loc='best', frameon=False, fontsize=15)
autolabel_horizontal(rects1,ax)
autolabel_horizontal(rects2,ax)
#Analisando a maior porcentagem de acertos
pontos_avancados = list(np.arange(1,max(df_feature['quantos_pontos_avancou']),0.5))
for i in np.arange(0.5,len(pontos_avancados),0.5):
soma = sum(df_feature['quantos_pontos_avancou'].apply(lambda x: x>i))
print('Acimda de {:.2f} pontos avançou {:.2f}%'.format(i,(soma/len(df_feature))*100))
#Analisando diferença de máxima e mínima
estatistica.columns
estatistica.loc['max',['abertura', 'maxima', 'minima', 'fechamento']] - estatistica.loc['min',['abertura',
'maxima', 'minima', 'fechamento']]
#Analisando média dos preços

```

```

estatistica.columns
labels = ['abertura', 'maxima', 'minima', 'fechamento', '20mma_maxima_2m',
'20mma_minima_2m', '9mme_fechamento', '200mma_fechamento',
'20mma_maxima_5m', '20mma_minima_5m']
media_colunas = np.round(list(estatistica.loc['mean',labels]), decimals=2)
ind = np.arange(len(labels))
fig = plt.figure(figsize=(15,15))
ax = fig.add_subplot(1,1,1)
for i in range(0,len(labels)):
    rects0 = ax.bar(labels[i], media_colunas[i],width=width, edgecolor='black')
    autolabel_without_pct(rects0,ax)
ax.set_xticks(ind)
labels = [labels[i].capitalize() for i in range(0,len(labels))]
ax.set_xticklabels(labels, fontsize=15)
ax.set_title('Media de Pontos por Colunas', fontsize=15)
ax.set_xlabel('Colunas', fontsize=15)
ax.set_ylabel('Média das Coluna', fontsize=15)
ax.set_yticks(np.arange(0,5000,500))
ax.set_yticklabels(np.arange(0,5000,500), fontsize=15)
ax.tick_params(axis='x', labelsize=15, labelcolor='k', labelrotation=45)
plt.tight_layout()
#Definindo a coluna gain
df_feature.columns
df_feature['gain'] = df_feature['quantos_pontos_avancou'].apply(lambda x: x>=2.5)
df_feature['gain'] = df_feature['gain'].apply(lambda x: 1 if x==True else 0)
gain = df_feature[df_feature['gain'] ==1]
loss = df_feature[df_feature['gain'] ==0]
print('Total de {:.2f} operações com Gain'.format(len(gain)))
print('Porcentagem de {:.2f}% gain em relação ao banco de dados'.format((len(gain)/len(df_feature))))
print('Total de {:.2f} operações com Loss'.format(len(loss)))
print('Porcentagem de {:.2f}% loss em relação ao banco de dados'.format((len(loss)/len(df_feature))))
#####
#Definindo os principais bancos e corretoras a favor da operação
df_feature.columns
df_compra = df_feature[df_feature['tipo_de_negociacao']=='compra'][['corretora_a',
'corretora_b', 'corretora_c',
'corretora_d', 'bank_a', 'bank_b', 'bank_c', 'bank_d','gain']]
df_venda = df_feature[df_feature['tipo_de_negociacao']=='venda'][['corretora_a', 'cor-
retora_b', 'corretora_c',

```

```

'corretora_d', 'bank_a', 'bank_b', 'bank_c', 'bank_d', 'gain']]
df_compra_gain = df_compra[df_compra['gain']==1]
df_venda_gain = df_venda[df_venda['gain']==1]
bancos = ['corretora_a', 'corretora_b', 'corretora_c',
'corretora_d', 'bank_a', 'bank_b', 'bank_c', 'bank_d']
df_compra_gain_porcentagem = pd.DataFrame(np.round([(len(df_compra_gain[df_compra_gain[
for i in bancos], decimals=2),
index= [i for i in bancos], columns=['% Gain'])
df_venda_gain_porcentagem = pd.DataFrame(np.round([(len(df_venda_gain[df_venda_gain[i]==
for i in bancos], decimals=2),
index = [i for i in bancos], columns=['% Loss'])
fig = plt.figure(figsize=(10,10))
ind=np.arange(len(bancos))
ax = fig.add_subplot(1,1,1)
for i in range(len(bancos)):
    rects = ax.bar(bancos[i],df_compra_gain_porcentagem.values[i][0], width=width, ed-
gecolor='black')
    autolabel_without_pct(rects, ax )
plt.tight_layout()
ax.set_xticklabels(['Corretora A', 'Corretora B', 'Corretora C', 'Corretora D','Banco A',
'Banco B','Banco C','Banco D']
, fontsize=15)#[bancos[i].capitalize() for i in range(len(bancos))]
ax.set_xlabel('Instituições Financeiras', fontsize=15)
ax.tick_params(axis='x', labelsize=15, labelcolor='k', labelrotation=45)
ax.set_xticks(ind)
ax.set_ylabel('Porcentagem', fontsize=15)
ax.set_yticklabels(np.arange(0,70,10), fontsize=15)
ax.set_title('Gráfico de Porcentagem de \n Acertos Instituições Financeiras na Com-
pra', fontsize=15)
fig = plt.figure(figsize=(10,10))
ind=np.arange(len(bancos))
ax = fig.add_subplot(1,1,1)
for i in range(len(bancos)):
    rects = ax.bar(bancos[i],df_venda_gain_porcentagem.values[i][0], width=width, edge-
color='black')
    autolabel_without_pct(rects, ax)
plt.tight_layout()
ax.set_xticklabels(['Corretora A', 'Corretora B', 'Corretora C', 'Corretora D','Banco A',
'Banco B','Banco C','Banco D']

```



```

, fontsize=15)#[bancos[i].capitalize() for i in range(len(bancos))]
ax.set_xlabel('Instituições Financeiras', fontsize=15)
ax.tick_params(axis='x', labelsiz=15, labelcolor='k', labelrotation=45)
ax.set_xticks(ind)
ax.set_ylabel('Porcentagem', fontsize=15)
ax.set_yticklabels(np.arange(0,70,10), fontsize=15)
ax.set_title('Gráfico de Porcentagem de \n Acertos das Instituições Financeiras na
Venda', fontsize=15)
#Correlação entre as variáveis dependentes e independentes
df_feature.columns
df2 = df_feature.drop(['tipo_de_negociacao','percentual_venda',
'quantas_correcoes','quantos_pontos_avancou','quantos_pontos_retornados','gain'],
axis=1)
correlacao = df2.corrwith(df_feature.gain)
labels = [i.capitalize() for i in correlacao.index]
ind = np.arange(len(labels))
fig= plt.figure(figsize=(10,10))
ax=fig.add_subplot(1,1,1)
for i in range(len(labels)):
    rects = ax.bar(labels[i], np.round(correlacao.values[i], decimals=2), width=width, ed-
gecolor='black')
ax.axhline(y=0, color='r', linestyle='-', linewidth =1)
ax.tick_params(axis='x', labelsiz=15, labelcolor='k', labelrotation=90)
ax.set_xlabel('Intituições Financeiras', fontsize=15)
ax.set_xticks(ind)
ax.set_ylabel('Correlação', fontsize=15)
ax.tick_params(axis='y', labelsiz=15, labelcolor='k', labelrotation=0)
ax.set_ylim(-1,1)
ax.set_title('Correlação da Variáveis Independens \n com a Variável dependente
Gain', fontsize=15)
#Separando em X e y
df_feature.columns
X = df_feature.drop(['horario','percentual_venda',
'quantas_correcoes','quantos_pontos_avancou',
'quantos_pontos_retornados','gain'],axis=1)
y = df_feature['gain']
#Transformando as variáveis Dummies
X = pd.get_dummies(X)
#Evitando dummies trap

```

```

X.columns
X = X.drop('tipo_de_negociacao_venda',axis=1)
#Splitting train and test set
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3, random_state=0)
X_train.shape, X_test.shape,y_train.shape, y_test.shape
len(y_train[y_train==1])
len(y_train[y_train==0])
# Balancing the Training Set Upsample
from sklearn.utils import resample
X = pd.concat([X_train, y_train], axis=1)
not_gain = X[X.gain==0]
gain = X[X.gain==1]
gain_upsampled = resample(not_gain,
replace=True, # sample with replacement
n_samples=len(gain), # match number in majority class
random_state=0) # reproducible results
upsampled = pd.concat([gain, gain_upsampled])
upsampled.gain.value_counts()
y_train = upsampled.gain
X_train = upsampled.drop('gain', axis=1)
len(y_train[y_train==1])
len(y_train[y_train==0])
#Normalizando os dados
from sklearn.preprocessing import StandardScaler
sc_x = StandardScaler()
X_train = pd.DataFrame(sc_x.fit_transform(X_train), columns=X_train.columns.values)
X_test = pd.DataFrame(sc_x.transform(X_test), columns=X_train.columns.values)
#Importing Keras libraries e packages
import keras
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import Activation
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import GridSearchCV
from keras.layers import LeakyReLU
leaky_relu_alpha = 0.1
import time

```

```

from keras.optimizers import Adam, Adamax, Nadam, SGD
from keras import regularizers
#How many layer and neurons I will use in my model?
"""

def create_model(layers,activation,optimizer):
    classifier = Sequential()
    for i, nodes in enumerate(layers):
        if i==0:
            classifier.add(Dense(nodes,input_dim = int(X_train.shape[1])))
            classifier.add(Activation(activation))
        else:
            classifier.add(Dense(nodes))
            classifier.add(Activation(activation))
        classifier.add(Dense(1)) #Note: no activation beyond this point.
    classifier.compile(optimizer=optimizer, loss='binary_crossentropy', metrics=['accuracy'])
    return classifier

classifier = KerasClassifier(build_fn=create_model, batch_size = 10, epochs = 100,
verbose=0 )

classifier
layers = [[24,48], [48,24],[22,44],[32,64]]
activation = ['sigmoid', 'relu']
parameters = {'batch_size': [128, 256],
'layers' : layers,
'activation' : activation,
'epochs': [100, 500],
'optimizer': ['adam', 'sgd']}
grid_search = GridSearchCV(estimator = classifier,
param_grid = parameters,
scoring = 'accuracy',
cv = 10)
t0 = time.time()
grid_search = grid_search.fit(X_train, y_train)
t1 = time.time()
print("Took %0.2f seconds" % (t1 - t0))
best_parameters = grid_search.best_params_
best_accuracy = grid_search.best_score_
best_accuracy, best_parameters
"""

def create_model(layers,dropout):

```

```

classifier = Sequential()
for i, nodes in enumerate(layers):
    if i==0:
        classifier.add(Dense(nodes,init = 'uniform', activation='relu', input_dim = int(X_train.shape[1])))
        classifier.add(Dropout(p= dropout))
    else:
        classifier.add(Dense(nodes, init = 'uniform'))
        classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
        classifier.add(Dropout(p= dropout ))
    classifier.add(Dense(1,init = 'uniform', activation = 'sigmoid')) #Note: no activation
beyond this point.
    classifier.compile(optimizer='sgd', loss='binary_crossentropy', metrics=['accuracy'])
    return classifier
classifier = KerasClassifier(build_fn=create_model, batch_size = 10, epochs = 100,
verbose=0 )
classifier
int((X_test.shape[1]/2)+1)
layers = [[int((X_test.shape[1]/2)+1),24,48], [int((X_test.shape[1]/2)+1),48,24],[int((X_test.shape[1]
dropout = [0.2,0.3,0.4,0.5]
parameters = {
'layers' : layers,
'dropout': dropout,
}
#'batch_size': [128, 256],
#'epochs': [100, 500]
grid_search = GridSearchCV(estimator = classifier,
param_grid = parameters,
scoring = 'accuracy',
cv = 10)
t0 = time.time()
grid_search = grid_search.fit(X_train, y_train)
t1 = time.time()
print("Took %0.2f seconds" % (t1 - t0))
best_parameters = grid_search.best_params_
best_accuracy = grid_search.best_score_
best_accuracy, best_parameters
results2 = pd.DataFrame()
for layer1 in np.arange(len(dropout)):
for layer2 in np.arange(len(dropout)):

```

```

for layer3 in np.arange(len(dropout)):
#Initialising the ANN
classifier = Sequential()
#Adding the input layer and the first hidden layer
classifier.add(Dense(output_dim =int((X_test.shape[1]/2)+1) , init = 'uniform',
activation = 'relu', input_dim = int(X_test.shape[1])))
#classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
classifier.add(Dropout(p= dropout[layer1] ))
#output_dim = média da soma do número de variáveis (no caso 30) + 1
#init = Peso. Gera aleatoriamente. Sempre 'uniform'
#activation = função de ativação linear retificada. A mais utilizada.
#Input_dim = número de variáveis (no caso 30)
#Dropout = evita overfitting. Começa com 0.1. Se continuar com overfitting, tentar de
0.2 a 0.5. Nunca maior que 0.5 senão será underfitting
#Adding the second hidden layer
classifier.add(Dense(output_dim = 24, init = 'uniform'))#activation = 'relu'
classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
classifier.add(Dropout(p= dropout[layer2]))
#output_dim = média da soma do número de variáveis (no caso 30) + 1
#init = Peso. Gera aleatoriamente. Sempre 'uniform'
#activation = função de ativação linear retificada. A mais utilizada.
#Input_dim = Não será mais necessário pq já foi feito no input layer
#Dropout = evita overfitting. Começa com 0.1. Se continuar com overfitting, tentar de
0.2 a 0.5. Nunca maior que 0.5 senão será underfitting
#Adding the third hidden layer
classifier.add(Dense(output_dim = 48, init = 'uniform'))#, activation = 'relu'
classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
classifier.add(Dropout(p= dropout[layer3]))
#Adding the output layer
classifier.add(Dense(output_dim = 1, init = 'uniform', activation = 'sigmoid'))
#output_dim = Como neste caso queremos 1 ou 0, só teremos 1 output_layer.
#init = Peso. Gera aleatoriamente. Sempre 'uniform'
#activation = função de ativação sigmoid. A mais utilizada para output layer quando
são binárias.
#Input_dim = Não será mais necessário pq já foi feito no input layer
#Compiling the ANN
adam=Adam(lr=0.0001)
classifier.compile(optimizer = adam, loss = 'binary_crossentropy', metrics = ['accu-
racy'])

```

#optimizer = algoritmo para selecionar o melhor peso do ANN. 'Adam' é um dos mais utilizados

#loss = Algoritmo que minimiza as perdas do gradiente descendente estocástico. Como a saída é binária, utilizou binary_crossentropy. Se houver mais que uma variável categorical_crossentropy

#metrics = Padrão

#Fit classifier to the training test

history = classifier.fit(X_train, y_train, batch_size = len(X_train), epochs = 100, validation_data=(X_test, y_test))

#batch_size = não tem um valor certo.

#epochs = não tem um valor certo

#Predicting the test set result

y_pred = classifier.predict(X_test)

y_pred = (y_pred > 0.5) #converte em verdadeiro ou falso

from sklearn.metrics import confusion_matrix, accuracy_score, f1_score, precision_score, recall_score

acc = accuracy_score(y_test, y_pred)

prec = precision_score(y_test, y_pred)

rec = recall_score(y_test, y_pred)

f1 = f1_score(y_test, y_pred)

results_final = pd.DataFrame([['ANN Dropout {} - {} - {}'.format(dropout[layer1], dropout[layer2], dropout[layer3]), acc, prec, rec, f1]],

columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])

results2 = results2.append(results_final)

Plot training & validation accuracy values

plt.plot(history.history['acc'])

plt.plot(history.history['val_acc'])

plt.title('Acurácia c/ Dropout {} - {} - {}'.format(dropout[layer1], dropout[layer2], dropout[layer3]))

plt.ylabel('Acurácia')

plt.xlabel('Epoch')

plt.legend(['Treino', 'Teste'], loc='upper left')

plt.savefig('Accuracy {} - {} - {}.png'.format(dropout[layer1], dropout[layer2], dropout[layer3]))

plt.show()

Plot training & validation loss values

plt.plot(history.history['loss'])

plt.plot(history.history['val_loss'])

plt.title('Perda c/ Dropout {} - {} - {}'.format(dropout[layer1], dropout[layer2], dropout[layer3]))

plt.ylabel('Perda')

plt.xlabel('Epoch')

```

plt.legend(['Treino', 'Teste'], loc='upper left')
plt.savefig('Loss {} - {} - {}.png'.format(dropout[layer1], dropout[layer2], dropout[layer3]))
plt.show()
results2.sort_values(by='Accuracy', ascending=False)
# Salvando planilha tratada
results2.sort_values(by='Accuracy', ascending=False).to_excel('Resultado Final.xlsx')
#####
# Initialising the ANN
classifier = Sequential()
# Adding the input layer and the first hidden layer
classifier.add(Dense(output_dim = int((X_test.shape[1]/2)+1) , init = 'uniform', in-
put_dim = int(X_test.shape[1]),
activation = 'relu')) # kernel_regularizer=regularizers.l1(0.001)
classifier.add(Dropout(p= 0.5))
# classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
# output_dim = média da soma do número de variáveis (no caso 30) + 1
# init = Peso. Gera aleatoriamente. Sempre 'uniform'
# activation = função de ativação linear retificada. A mais utilizada.
# Input_dim = número de variáveis (no caso 30)
# Dropout = evita overfitting. Começa com 0.1. Se continuar com overfitting, tentar de
0.2 a 0.5. Nunca maior que 0.5 senão será underfitting
# Adding the second hidden layer
classifier.add(Dense(output_dim = 24, init = 'uniform',
)) # activation = 'relu', kernel_regularizer=regularizers.l1(0.001)
classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
classifier.add(Dropout(p= 0.5))
# output_dim = média da soma do número de variáveis (no caso 30) + 1
# init = Peso. Gera aleatoriamente. Sempre 'uniform'
# activation = função de ativação linear retificada. A mais utilizada.
# Input_dim = Não será mais necessário pq já foi feito no input layer
# Dropout = evita overfitting. Começa com 0.1. Se continuar com overfitting, tentar de
0.2 a 0.5. Nunca maior que 0.5 senão será underfitting
# Adding the third hidden layer
classifier.add(Dense(output_dim = 48, init = 'uniform',
)) #, activation = 'relu', kernel_regularizer=regularizers.l1(0.001)
classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
classifier.add(Dropout(p= 0.5))
# Adding the output layer
classifier.add(Dense(output_dim = 1, init = 'uniform', activation = 'sigmoid'))

```

```

#output_dim = Como neste caso queremos 1 ou 0, só teremos 1 output_layer.
#init = Peso. Gera aleatoriamente. Sempre 'uniform'
#activation = função de ativação sigmoid. A mais utilizada para output layer quando
são binárias.

#Input_dim = Não será mais necessário pq já foi feito no input layer
#Compiling the ANN
sgd = SGD(lr=0.0001)
adam=Adam(lr=0.0001)
adamax=Adamax(lr=0.0001)
nadam = Nadam(lr=0.0001)
classifier.compile(optimizer = adam, loss = 'binary_crossentropy', metrics = ['accu-
racy'])

#optimizer = algoritmo para selecionar o melhor peso do ANN. 'Adam' é um dos
mais utilizados

#loss = Algoritmo que minimiza as perdas do gradiente descendente estocástico.
Como a saída é binária, utilizou binary_crossentropy. Se houver mais que uma variável
categorical_crossentropy

#metrics = Padrão
#Fit classifier to the training test
history = classifier.fit(X_train, y_train, batch_size = 10, epochs = 100, validation_data=(X_test,
y_test))

#batch_size = não tem um valor certo.
#epochs = não tem um valor certo
#Predicting the test set result
y_pred = classifier.predict(X_test)
y_pred = (y_pred > 0.5) #converte em verdadeiro ou falso
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score, preci-
sion_score, recall_score
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
results_final = pd.DataFrame(['ANN W/ Dropout', acc, prec, rec, f1]),
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
# Plot training & validation accuracy values
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model accuracy')
plt.ylabel('Accuracy')

```



```

plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.savefig('accuracy.png')
plt.show()
# Plot training & validation loss values
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')
plt.savefig('loss.png')
plt.show()
#####
# Evaluating the ANN
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import cross_val_score
from keras.models import Sequential
from keras.layers import Dense
def build_classifier():
    classifier = Sequential()
    classifier.add(Dense(units = int((X_test.shape[1]/2)+1), kernel_initializer = 'uniform',
activation = 'relu',
    input_dim = int(X_test.shape[1])))
    classifier.add(Dropout(p= 0.5))
    classifier.add(Dense(units = 24, kernel_initializer = 'uniform'))
    classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
    classifier.add(Dropout(p= 0.5))
    classifier.add(Dense(units = 48, kernel_initializer = 'uniform'))
    classifier.add(LeakyReLU(alpha=leaky_relu_alpha))
    classifier.add(Dropout(p= 0.5))
    classifier.add(Dense(units = 1, kernel_initializer = 'uniform', activation = 'sigmoid'))
    classifier.compile(optimizer = adam, loss = 'binary_crossentropy', metrics = ['accu-
racy'])
    return classifier
classifier = KerasClassifier(build_fn = build_classifier, batch_size = 10, epochs = 100)
accuracies = cross_val_score(estimator = classifier, X = X_train, y = y_train, cv = 10,
n_jobs = 1)
accuracies.mean()

```

```

accuracies.std()
accuracies_data = pd.Series(accuracies,name='Acurácia (%)')
accuracies_data = np.round(accuracies_data*100,0)
accuracies_data.to_excel('Cross_validation.xlsx')
print("ANN Accuracy: %0.3f (+/- %0.3f)" % (accuracies.mean(), accuracies.std() * 2))
import pydotplus
from keras.utils.vis_utils import model_to_dot
keras.utils.vis_utils.pydot = pydot
plot_model(classifier, to_file='model.png', show_shapes=True, show_layer_names=True)
##### Model Building #####
### Comparing Models
## Logistic Regression
from sklearn.linear_model import LogisticRegression
lr_classifier = LogisticRegression(random_state = 0, penalty = 'l2')
lr_classifier.fit(X_train, y_train)
# Predicting Test Set
y_pred = lr_classifier.predict(X_test)
from sklearn.metrics import confusion_matrix, accuracy_score, f1_score, precision_score, recall_score
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame([[ 'Logistic Regression (Lasso)', acc, prec, rec, f1]],
columns = [ 'Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)
## K-Nearest Neighbors (K-NN)
#Choosing the K value
error_rate= []
for i in range(1,40):
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=i)
knn.fit(X_train, y_train)
pred_i = knn.predict(X_test)
error_rate.append(np.mean(pred_i != y_test))
plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')

```

```

plt.xlabel('K')
plt.ylabel('Error Rate')
print(np.mean(error_rate))
from sklearn.neighbors import KNeighborsClassifier
kn_classifier = KNeighborsClassifier(n_neighbors=15, metric='minkowski', p= 2)
kn_classifier.fit(X_train, y_train)
# Predicting Test Set
y_pred = kn_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame([[ 'K-Nearest Neighbors (minkowski)', acc, prec, rec,
f1]],
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)
## SVM (Linear)
from sklearn.svm import SVC
svm_linear_classifier = SVC(random_state = 0, kernel = 'linear', probability= True)
svm_linear_classifier.fit(X_train, y_train)
# Predicting Test Set
y_pred = svm_linear_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame([[ 'SVM (Linear)', acc, prec, rec, f1]],
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)
## SVM (rbf)
from sklearn.svm import SVC
svm_rbf_classifier = SVC(random_state = 0, kernel = 'rbf', probability= True)
svm_rbf_classifier.fit(X_train, y_train)
# Predicting Test Set
y_pred = svm_rbf_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

```

```

model_results = pd.DataFrame([[ 'SVM (RBF)', acc, prec, rec, f1]],
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)

## Naive Bayes
from sklearn.naive_bayes import GaussianNB
gb_classifier = GaussianNB()
gb_classifier.fit(X_train, y_train)

# Predicting Test Set
y_pred = gb_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame([[ 'Naive Bayes (Gaussian)', acc, prec, rec, f1]],
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)

## Decision Tree
from sklearn.tree import DecisionTreeClassifier
dt_classifier = DecisionTreeClassifier(criterion='entropy', random_state=0)
dt_classifier.fit(X_train, y_train)

#Predicting the best set result
y_pred = dt_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame([[ 'Decision Tree', acc, prec, rec, f1]],
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)

from sklearn.externals.six import StringIO
from IPython.display import Image
from sklearn.tree import export_graphviz
import pydotplus
dot_data = StringIO()
export_graphviz(dt_classifier, out_file=dot_data,
filled=True, rounded=True,
special_characters=True)
graph = pydotplus.graph_from_dot_data(dot_data.getvalue())
Image(graph.create_png())

```

```

graph.write_pdf('mestrado.pdf')
graph.write_png('mestrado.png')
## Random Forest
from sklearn.ensemble import RandomForestClassifier
rf_classifier = RandomForestClassifier(random_state = 0, n_estimators = 100,
criterion = 'gini')
rf_classifier.fit(X_train, y_train)
# Predicting Test Set
y_pred = rf_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame(['Random Forest Gini (n=100)', acc, prec, rec, f1]),
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)
## Ada Boosting
from sklearn.ensemble import AdaBoostClassifier
ad_classifier = AdaBoostClassifier()
ad_classifier.fit(X_train, y_train)
# Predicting Test Set
y_pred = ad_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame(['Ada Boosting', acc, prec, rec, f1]),
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)
## Gradient Boosting
from sklearn.ensemble import GradientBoostingClassifier
gr_classifier = GradientBoostingClassifier()
gr_classifier.fit(X_train, y_train)
# Predicting Test Set
y_pred = gr_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)

```

```

model_results = pd.DataFrame(['Gradient Boosting', acc, prec, rec, f1]),
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)
##Ensemble Voting Classifier
from sklearn.ensemble import VotingClassifier
from sklearn.metrics import accuracy_score
voting_classifier = VotingClassifier(estimators= [('lr', lr_classifier),
('kn', kn_classifier),
('svc_linear', svm_linear_classifier),
('svc_rbf', svm_rbf_classifier),
('gb', gb_classifier),
('dt', dt_classifier),
('rf', rf_classifier),
('ad', ad_classifier),
('gr', gr_classifier)], voting='soft')
for clf in (lr_classifier, kn_classifier, svm_linear_classifier, svm_rbf_classifier,
gb_classifier, dt_classifier, rf_classifier, ad_classifier, gr_classifier, voting_classifier):
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
print(clf.__class__.__name__, accuracy_score(y_test, y_pred))
# Predicting Test Set
y_pred = voting_classifier.predict(X_test)
acc = accuracy_score(y_test, y_pred)
prec = precision_score(y_test, y_pred)
rec = recall_score(y_test, y_pred)
f1 = f1_score(y_test, y_pred)
model_results = pd.DataFrame(['Ensemble Voting', acc, prec, rec, f1]),
columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F1 Score'])
results_final = results_final.append(model_results, ignore_index = True)
#The Best Classifier
print('The best classifier is:')
print('{}'.format(results_final.sort_values(by='Accuracy', ascending=False).head(5)))
#Applying K-fold validation
from sklearn.model_selection import cross_val_score
accuracies = cross_val_score(estimator=rf_classifier, X=X_train, y=y_train, cv=10)
accuracies.mean()
accuracies.std()
print("Random Forest Gini (n=100) Accuracy: %0.3f (+/- %0.3f)" % (accuracies.mean(),
accuracies.std() * 2))

```