

**UNIVERSIDADE FEDERAL DO PAMPA
BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO**

THIAGO PORTO MENDES

**MODELO DE INTELIGÊNCIA
ARTIFICIAL PARA PREVISÃO DO
ÍNDICE S&P500**

**Bagé
2023**

THIAGO PORTO MENDES

**MODELO DE INTELIGÊNCIA
ARTIFICIAL PARA PREVISÃO DO
ÍNDICE S&P500**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Engenharia de Computação como requisito para a obtenção do grau de Bacharel em Engenharia de Computação.

Orientador: Milton Roberto Heinen

**Bagé
2023**

Ficha catalográfica elaborada automaticamente com os dados fornecidos pelo(a) autor(a) através do Módulo de Biblioteca do Sistema GURI (Gestão Unificada de Recursos Institucionais).

M538m Mendes, Thiago Porto

Modelo de inteligência artificial para
previsão do índice S&P500 / Thiago Porto Mendes.

- 2023.

86 f.: il.

Orientador: Milton Roberto Heinen

Trabalho de Conclusão de Curso (Graduação)
- Universidade Federal do Pampa, Campus Bagé,
Bacharelado em Engenharia de Computação, 2023.

1. S&P500. 2. Machine Learning. 3. Facebook
Prophet. 4. Trading. I. Milton Roberto Heinen.
II. Título.



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
Universidade Federal do Pampa

THIAGO PORTO MENDES

MODELO DE INTELIGÊNCIA ARTIFICIAL PARA PREVISÃO DO ÍNDICE S&P500

Trabalho de Conclusão de Curso apresentado ao Curso de Bacharelado em Engenharia de Computação da Universidade Federal do Pampa, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Computação.

Trabalho de Conclusão de Curso defendido e aprovado em: 03 de fevereiro de 2023.

Banca examinadora:

Prof. Dr. Milton Roberto Heinen
Orientador
UNIPAMPA

Prof. Dr. Carlos Michel Betemps
UNIPAMPA

Prof. Dra. Sandra Dutra Piovesan
UNIPAMPA



Assinado eletronicamente por **SANDRA DUTRA PIOVESAN, PROFESSOR DO MAGISTERIO SUPERIOR**, em 07/02/2023, às 13:39, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **CARLOS MICHEL BETEMPS, PROFESSOR DO MAGISTERIO SUPERIOR**, em 07/02/2023, às 21:26, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



Assinado eletronicamente por **MILTON ROBERTO HEINEN, PROFESSOR DO MAGISTERIO SUPERIOR**, em 08/02/2023, às 12:39, conforme horário oficial de Brasília, de acordo com as normativas legais aplicáveis.



A autenticidade deste documento pode ser conferida no site https://sei.unipampa.edu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1049217** e o código CRC **D2ED799A**.

Referência: Processo nº 23100.002573/2023-40 SEI nº 1049217

Dedico este trabalho à meus pais, meus colegas e amigos da faculdade que enfrentaram as mesmas dificuldades que eu, sempre de frente e motivando uns aos outros a não desistir.

AGRADECIMENTO

Aos meus pais, Sérgio e Maria Luiza e a minha irmã Mariane, pela confiança no meu progresso e pelo apoio emocional.

Aos meus gatos que acima de tudo, sempre presentes nos momentos difíceis com um miado de incentivo.

Ao meu orientador Milton que apesar da intensa rotina de sua vida acadêmica aceitou me orientar nesta monografia. As suas valiosas indicações fizeram toda a diferença.

Aos meus professores pelos ensinamentos dispensados ao longo dos anos que se tornou essencial para que o projeto fosse concluído.

Aos meus colegas do curso pelas trocas de ideias e ajuda mútua. Juntos conseguiremos avançar e ultrapassar todos os obstáculos.

“Until you spread your wings, you’ll have no
idea how far you can fly.”

— Napoleon Bonaparte

RESUMO

O mercado financeiro mundial é responsável pela circulação de uma enorme quantidade de dinheiro, sendo estimado em mais de 30 trilhões de dólares em 2022. Atualmente, após a pandemia, guerras e instabilidade inflacionária, existe uma grande incerteza envolta do futuro e da segurança que as moedas e negócios que atualmente dominam o mercado podem prover aos investidores e pessoas no geral. Em resposta a essas tensões, mais e mais pessoas buscam aplicar seu dinheiro, e diversificar tais aplicações, no mercado de ações de modo a proteger suas economias e rendas, ou para os mais aventureiros, buscar novas fontes de renda nas subidas e descidas do mercado. Em vista dessa entrada de novos ingressantes ao mercado sem experiência e de outros investidores mais antigos que podem estar com dificuldade de encontrar investimentos lucrativos nessa nova economia, esse trabalho teve como objetivo o desenvolvimento de um modelo capaz de prever o valor do índice S&P500, o qual avalia e estima o valor de mercado das 500 maiores empresas negociadas publicamente que atuam nos Estados Unidos da América. Além de atuar como estudo de caso, a escolha do índice vem de sua amplitude e diversidade de ativos que o compõe, o que o torna amplamente usado como um medidor do estado geral do mercado, que tende a seguir com otimismo ou pessimismo de acordo com às tendências apresentadas pelo índice. O modelo proposto se diferencia dos previamente desenvolvidos por outros autores pela sua ênfase em desenvolver um sistema leve que pode ser retreinado localmente pelo usuário sem a necessidade de recursos intensivos que foi possível pelo uso da ferramenta Facebook Prophet em conjunto à Árvores de decisão. As tecnologias utilizadas consistem primariamente no Facebook Prophet, que foi ferramenta que se mostrou mais adequada para a aplicação dessa solução e de árvores de decisão, que complementam às necessidades multivariadas do modelo, e para funcionalidades diversas, bibliotecas Python como Sklearn. Os resultados do modelo desenvolvido mostraram uma melhora leve comparado ao melhor dos outros modelos testados com componentes diferentes, porém dado a performance do mesmo em conjunto aos resultados simulados, concluí-se que os objetivos propostos foram alcançados de maneira satisfatória.

Palavras-chave: S&P500; Machine Learning; Facebook Prophet; Trading.

ABSTRACT

The global financial market is responsible for the circulation of a huge amount of money, estimated at more than 30 trillion dollars in 2022. Currently, after the pandemic, wars and inflationary instability, there is great uncertainty surrounding the future and security that currencies and businesses that currently dominate the market can provide for investors and people in general. In response to these tensions, more and more people are looking to invest their money, and diversify such investments, in the stock market in order to protect their savings and income, or for the more adventurous, seek new sources of income in the ups and downs of the market. . In view of this entry of new entrants to the market with no experience and other older investors who may be having difficulty finding profitable investments in this new economy, this work aimed to develop a model capable of predicting the value of the S&P500 index , which evaluates and estimates the market value of the 500 largest publicly traded companies operating in the United States of America. In addition to acting as a case study, the choice of the index comes from its breadth and diversity of assets that compose it, which makes it widely used as a gauge of the general state of the market, which tends to follow with optimism or pessimism according to trends shown by the index. The proposed model differs from those previously developed by other authors due to its emphasis on developing a lightweight system that can be locally retrained by the user without the need for intensive resources, which was made possible by the use of the Facebook Prophet tool in conjunction with Decision Trees. The technologies used primarily consist of Facebook Prophet, which was the tool that proved to be the most suitable for the application of this solution and decision trees, which complement the multivariate needs of the model, and for various functionalities, Python libraries such as Sklearn. The results of the model developed showed a slight improvement compared to the best of the other models tested with different components, but given its performance in conjunction with the simulated results, it was concluded that the proposed objectives were achieved satisfactorily.

Keywords: S&P500, Machine Learning, Facebook Prophet, Trading.

LISTA DE FIGURAS

Figura 1	Exemplo de gráfico de linha	25
Figura 2	Exemplo de gráfico de barra	26
Figura 3	Exemplo de gráfico de vela.....	26
Figura 4	Exemplo de linhas de Tendência	28
Figura 5	Ilustração da hierarquia de IA, ML e DL	35
Figura 6	Exemplo de árvore de decisão	41
Figura 7	Fluxo de funcionamento do projeto.....	52
Figura 8	Dados a serem inseridos no modelo Prophet.....	58
Figura 9	Aproximação da curva e previsão do Prophet	58
Figura 10	Demonstração do ponto de previsão para uma semana	59
Figura 11	Funcionamento da tabela de comparação	61
Figura 12	Matriz de confusão	67
Figura 13	Diagrama de mudança	69

LISTA DE TABELAS

Tabela 1	Strings usadas	18
Tabela 2	Trabalhos analisados.....	50
Tabela 3	Tabela estatística de resultados.....	65
Tabela 4	Tabela estatística de resultados.....	68
Tabela 5	Tabela de resultados do simulador.....	70
Tabela 6	Segunda tabela de resultados do simulador.....	71
Tabela 7	Tabela de resultados da acurácia	80

LISTA DE ABREVIATURAS E SIGLAS

AI	Artificial Intelligence (Inteligência Artificial)
ARIMA	Autoregressive Integrated Moving Average (Modelo auto-regressivo integrado de médias móveis)
DL	Deep Learning (Aprendizado Profundo)
IDE	Integrated Development Environment (Ambiente de desenvolvimento integrado)
KNN	K-Nearest Neighbours
LSTM	Long short-term memory (Memória de longo curto prazo)
MA	Moving Averages (Médias móveis)
MACD	Moving Average Convergence / Divergence (Convergência/Divergência de Médias Móveis)
MAE	Mean Absolute Error (Média do Erro Absoluto)
MAPE	Mean Absolute Percentage Error (Média do Erro Absoluto Percentual)
MSCI	Morgan Stanley Capital International
OBV	On-Balance Volume
RMS	Root Mean Squared Error (Raiz do Erro Quadrático Médio)
RNN	Recurrent Neural Network (Rede Neural Recorrente)
RSI	Relative strength index (Índice de força relativa)
ML	Machine Learning (Aprendizado de máquina)
S&P500	Standard & Poor's 500
UNIPAMPA	Universidade Federal do Pampa
XGBoost	eXtreme Gradient Boosting

SUMÁRIO

1 INTRODUÇÃO	15
1.1 Problema de pesquisa	15
1.2 Objetivo Geral	15
1.2.1 Objetivos específicos	16
1.3 Justificativa e importância de pesquisa.....	16
1.4 Metodologia de pesquisa	17
1.5 Organização do texto	18
2 REVISÃO BIBLIOGRÁFICA	20
2.1 Investidores	20
2.2 Mercado	22
2.2.1 Ursos e touros	22
2.2.2 Preços de abertura, fechamento, volume e etc	23
2.3 Gráficos	23
2.3.1 Tipos de gráficos.....	24
2.4 Tendências.....	27
2.4.1 Linhas de tendência	27
2.5 Indicadores	28
2.5.1 Médias Móveis	29
2.5.2 MACD - Convergência/Divergência de médias móveis	30
2.5.3 MACD - Histograma.....	31
2.5.4 Estocástico	32
2.5.5 Índice de Força Relativa.....	32
2.6 Risco	33
2.6.1 Controle de riscos.....	34
2.7 Revisão de técnicas e metodologias de inteligência artificial	35
2.8 Diferença de Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo	35
2.9 Inteligência Artificial	36
2.9.1 Inteligência artificial forte versus fraca	36
2.10 Aprendizado de Máquina	37
2.11 Aprendizado profundo.....	37
2.12 Facebook PROPHET	38
2.12.1 Sazonalidade.....	38
2.12.2 Modelo.....	39
2.13 Árvores de decisão	40
3 TRABALHOS CORRELATOS	43
3.1 Conclusão.....	49
4 DESENVOLVIMENTO	51
4.1 Fluxo de funcionamento	51
4.2 Ferramentas e Softwares	53
4.3 Configuração de parâmetros.....	55
4.4 Aquisição e preparação dos dados.....	56
4.5 Predição do modelo Prophet	57
4.6 Preparação dos dados para árvore de decisão	59
4.7 Montagem da árvore de decisão	61
4.8 Simulação de <i>trading</i>	62
5 RESULTADOS	64
5.1 Comparação do modelo.....	64

5.2	Análise dos resultados.....	67
5.3	Resultados da simulação de <i>trading</i>	69
5.4	Conclusão dos resultados	72
6	CONSIDERAÇÕES FINAIS	74
6.1	Problemas Encontrados.....	74
6.2	Conclusão.....	75
6.3	Trabalhos Futuros.....	76
	REFERÊNCIAS.....	77
	APÊNDICE A – TABELA DE RESULTADOS DA ACURÁCIA.....	80
	APÊNDICE B – CÓDIGO FONTE DO MODELO	81

1 INTRODUÇÃO

Com a inflação global em constante aumento, causada por eventos como a pandemia e problemas internacionais que afetam a disponibilidade de produtos e transações comerciais, cada vez mais pessoas buscam formas de se proteger contra os riscos no mercado financeiro, fazendo uso de investimentos seguros que ofereçam proteção contra a inflação, como títulos de renda fixa. No entanto, a entrada segura no mercado financeiro requer muito estudo ou a contratação de indivíduos ou empresas especializadas, o que pode não ser viável para todos devido à falta de tempo ou capital. Isso cria uma oportunidade para mini-cursos e ferramentas de baixo custo que possam ajudar os investidores iniciantes a entrar no mercado de forma segura e bem-sucedida.

Diante desse problema, busca-se desenvolver um modelo que possa fornecer esse suporte de maneira automatizada e sem custo para o usuário. A funcionalidade idealizada é a de prever os preços do índice S&P 500 com uma precisão que permita lucros em diferentes períodos de tempo. Para alcançar esse objetivo, usar-se-á a ferramenta de aprendizado de máquina (*machine learning*) Facebook Prophet, que simplifica o processo de treinamento e permite o rápido re-treinamento dos dados mesmo com capacidade de processamento limitada, o que é essencial para manter as previsões atualizadas com o estado atual do mercado e pode ser executado diretamente no computador pessoal do usuário. O modelo desenvolvido será disponibilizado em um repositório Git pronto para execução do usuário.

1.1 Problema de pesquisa

Será possível implementar um modelo capaz de solucionar o problema do auxílio de investidores, especialmente os pouco experientes, e que cumpra os requisitos de prever as subidas e quedas do índice S&P500 utilizando um conjunto de ferramentas de desenvolvimento, automação e aprendizado de máquina?

1.2 Objetivo Geral

O presente trabalho possui como objetivo geral o desenvolvimento de um modelo capaz de automaticamente e continuamente prever as mudanças de preço

do índice S&P500 com o propósito de auxiliar investidores, especialmente os com menos conhecimento e experiência, a identificar o estado geral do mercado ou investir diretamente em fundos atrelados ao índice utilizando ferramentas de aprendizado de máquina e técnicas já consolidadas de investimento e *trading*.

1.2.1 Objetivos específicos

Relativo aos objetivos específicos, o trabalho busca:

1. Estudar o conhecimento acumulado e difundido sobre o funcionamento e conceitos do mercado de ações, técnicas de *trading* e linhas de raciocínios apresentadas por *traders* de sucesso;
2. Buscar por uma tecnologia baseada em aprendizado de máquina que apresente as características e requisitos ideais para a implementação da solução;
3. Examinar na documentação online e buscar as ferramentas da linguagem Python (escolhida pela simplicidade de codificar e disponibilidade de bibliotecas diversas) e nas suas bibliotecas maneiras de construir uma interface com dados do mercado financeiro disponíveis de maneira gratuita;
4. Definir as funcionalidades e métricas de validação da solução a ser implementada, documentando o processo de desenvolvimento da mesma.

1.3 Justificativa e importância de pesquisa

A importância do trabalho na área do conhecimento vem da escassa aplicação do conceito de retreinamento e atualização constante baseado na lógica por trás de Médias Móveis Exponenciais, onde os valores mais recentes tem um peso consideravelmente maior comparado aos antigos. Isso, combinado com o uso de ideias estabelecidas de aprendizagem de máquina e uma validação mais rigorosa do modelo através de testes robustos, justifica o desenvolvimento deste trabalho como solução para problemas reais.

1.4 Metodologia de pesquisa

A pesquisa bibliográfica feita no trabalho com o propósito de estabelecer uma base de conhecimento ao leitor é classificada como pesquisa aplicada por cumprir a descrição: "objetiva gerar conhecimentos para aplicação prática dirigidos à solução de problemas específicos"(PRODANOV; FREITAS, 2013).

Sob o ponto de vista dos objetivos, segundo a classificação descrita em Prodanov and Freitas (2013), a pesquisa é classificada como exploratória dado sua finalidade de proporcionar mais informações sobre o assunto que vai ser investigado, facilitando assim a delimitação do problema de pesquisa e a expansão do conhecimento adquirido pelo pesquisador sobre o tema.

Quanto aos procedimentos técnicos, utilizou-se de pesquisa bibliográfica para obter os dados necessários para a elaboração da pesquisa, essa classificação sendo justificada pelos dados utilizados terem sido adquiridos a partir de: livros, revistas, publicações em periódicos e artigos científicos, jornais, boletins, monografias, dissertações, teses, material cartográfico, internet (fontes confiáveis e fidedignas).(PRODANOV; FREITAS, 2013)

O método utilizado para a formulação do trabalho foi o científico seguindo o raciocínio hipotético-dedutivo, onde se busca estabelecer uma teoria e a partir dela, tentar comprovar sua veracidade ou falsidade perante os dados adquiridos através de meios sistemáticos científicos.

Para estabelecer a posição do trabalho proposto dentro da área do conhecimento e analisar os tópicos já abordados e explorados por outros pesquisadores, utilizou-se de strings de busca nos sites da "IEEE"¹, "Arxiv"² e "Google Scholar"³. As strings usadas em cada um dos sites podem ser vistas na Tabela 1.

Assim que as ferramentas e metodologias foram definidas, as strings de busca foram elaboradas de maneira que objetivavam buscar as aplicações da mesma tecnologia em circunstâncias com o máximo possível de similaridades com o presente trabalho. Enquanto que as outras fontes de dados apresentaram pouca variedade de trabalhos, no caso do Google Scholar, a maior base de dados permitiu uma filtragem maior dos artigos em busca de selecionar com mais precisão os de maior relevância, limitando-se inclusive a apenas incluir os lançados nos últimos cinco anos para manter uma consistência dos

¹<https://ieeexplore.ieee.org/Xplore/>

²<https://arxiv.org/>

³<https://scholar.google.com.br/>

Tabela 1 – Strings usadas

<i>Site</i>	<i>Strings de busca</i>	<i>Parâmetros Adicionais</i>
Google Scholar	Stock Price Prediction facebook prophet S&P500 indicator	Ano: 2017-2022
Arxiv	Facebook Prophet	Nenhum
IEEE	"Facebook Prophet"financial	Nenhum

Fonte: O autor

trabalhos mais atuais, filtro esse que não é tão restritivo como parece dado que o próprio Prophet foi lançado em 2017.

A linguagem principal utilizada no trabalho é o Python, escolhida por ter acesso a diversas bibliotecas úteis e ferramentas de integração com as mais diversas áreas de desenvolvimento e automação, entre elas há toda uma gama de ferramentas de *trading* e matemática avançada. A IDE (Ambiente de Desenvolvimento Integrado) utilizada foi o Visual Studio Code ⁴ com extensões de suporte a Notebooks Python e a tecnologia principal a ser utilizada consiste no Facebook PROPHET com técnicas de previsão temporal e indicadores técnicos junto aos dados históricos de preço do índice S&P500.

1.5 Organização do texto

No Capítulo 1, que cobre toda a introdução, foram estabelecidos os objetivos, justificativa e metodologia do trabalho. Com os pontos de interesse já definidos, no Capítulo 2 foi explorada a bibliografia da área sobre os tópicos necessários para estabelecer os conhecimentos requeridos para compreender o trabalho proposto, primeiramente a revisão sobre *trading* e em seguida brevemente sobre conceitos básicos de inteligência artificial.

Após adquirir a estrutura de conhecimentos necessária, exploramos trabalhos similares dentro da área no Capítulo 3, onde analisamos quais conceitos já foram desenvolvidos anteriormente e quais conclusões relevantes foram tiradas. No capítulo 4, descrevemos detalhadamente o desenvolvimento e funcionamento do projeto. No capítulo 5, analisamos os resultados obtidos pelo modelo e determinamos sua capacidade de cumprir os requisitos estabelecidos. Por fim, no Capítulo 6, discutimos as

⁴<https://code.visualstudio.com/>

considerações finais do trabalho, incluindo a conclusão, problemas encontrados durante o desenvolvimento e possíveis trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Nesta seção, primeiramente, serão apresentados conceitos sobre *trading*, seguidos por conceitos básicos sobre aprendizado de máquina, que são fundamentais para compreender o trabalho realizado.

2.1 Investidores

Na visão de Elder (2014), *traders* (ou investidores) tendem a ser pessoas focadas e perspicazes, com a mente aberta a novas ideias e com o objetivo não de fazer dinheiro, mas sim de aperfeiçoar a habilidade de fazer bons investimentos, pois sabem que quando realmente se desenvolverem nesse meio, o dinheiro virá inevitavelmente.

Traders são divididos em três grupos: Compradores, vendedores e indecisos. Compradores querem pagar o mínimo possível, enquanto que os vendedores querem cobrar o máximo possível. Com isso em mente, o comprador tem a escolha de esperar o preço baixar ou pagar o que o vendedor demanda, enquanto que o vendedor tem uma escolha similar, esperar o preço subir ou aceitar uma oferta mais baixa. A transação ocorre quando existe esse consenso dos dois lados sobre o valor daquele ativo, porém o que pressiona os dois lados a não somente aguardarem as circunstâncias desejadas inicialmente sem buscar esse meio termo é a presença dos indecisos, que podem a qualquer momento passar a frente e, no caso dos compradores, aceitar o preço que o vendedor está oferecendo e perder a chance da compra, e no caso dos vendedores, oferecer um preço menor que o vendedor e perder a chance de vender a sua parte mais cedo.

Apesar de na literatura os termos *trader* e investidor costumarem ser usados intercambiavelmente, alguns autores buscaram descrever certos padrões de comportamentos que caracterizam cada um. Em seu livro, Graham (2003) diferencia investidores de traders de modo onde investidores buscam segurar seus ativos a longo prazo de modo a capitalizar os ganhos com o mínimo possível de taxas (tanto para as corretoras quanto para o governo). Já traders buscam tirar uma liquidez mediana porém em um número maior de *trades*, na expectativa de que apesar das taxas aplicadas em cada trade, esse volume maior trará lucros que não apenas as compensarão, mas também trarão mais lucro que investimentos a longo prazo (que tendem a ser mais seguros) trariam.

Para Graham (2003), há dois tipos de investidores, os "defensivos" e os "agressivos". O investidor defensivo (ou passivo) tem como objetivo principal evitar

perdas e erros sérios, e seu objetivo secundário é liberdade de esforço, problemas e a necessidade de fazer decisões constantes. O traço determinante dos investidores agressivos é sua disposição a dedicar tempo e cuidado à seleção de títulos que aparentam tanto serem mais arriscados e de maior retorno imediato que a maioria, objetivando a longo prazo retornos maiores quando comparado ao investidor defensivo, enquanto que o conservador tende a sofrer menos de *drawdowns* (quedas bruscas de preço comparado ao pico ou onde foi adquirido).

Como observado por Malkiel (1973), há duas correntes de pensamento utilizadas no mercado de ações, uma delas sendo a análise técnica e a outra sendo de análise fundamentalista. A análise técnica essencialmente é baseada em interpretar gráficos de preços, acreditando que o mercado é apenas 10% lógico e 90% psicológico, onde o que importa acima de tudo é prever como os outros investidores vão se comportar. O meio para essa compreensão, neste caso, viria dos gráficos de preços, os quais são compostos em grande parte do que esses mesmos investidores estiveram fazendo no passado.

Já a análise fundamentalista toma o lado contrário, acreditando que o mercado é 90% lógico e apenas 10% psicológico, pouco se importando com o padrão particular dos movimentos dos preços, mas sim buscando determinar o valor real daquele ativo. Valor nesse caso estaria relacionado com crescimento, pagamento de dividendos, taxas de interesse e risco. Com esses valores o investidor seria capaz de estimar o valor de um ativo, e caso esse valor seja maior que o atualmente estimado pelo mercado ele deveria comprar, acreditando que eventualmente o mercado vai refletir mais fielmente o valor ideal daquele ativo.

De acordo com Bodie, Kane and Marcus (2009), a definição de investimento é a renúncia atual à algum recurso de valor imediato na esperança que esse ato traga benefícios futuros, já no mercado financeiro, esses investimentos tratam-se quase que exclusivamente de ativos financeiros, os quais podem ser distinguidos em três tipos abrangentes:

1. Títulos de renda fixa: Promete uma renda contínua fixa ou uma renda variável determinada por uma fórmula específica. Atuam como o mais seguro dos investimentos, sendo garantidos os pagamentos a menos que a instituição declare falência (independe do sucesso direto da mesma). Esses títulos são oferecidos com uma grande variedade de limites de maturidade e pagamentos, isto sendo em quanto tempo o valor do título deve ser ressarcido ao credor e quanto que o credor receberá durante esse período. Um dos principais pontos a serem avaliados é a segurança

do título em questão, a qual depende de diversos fatores, desde a confiabilidade da instituição (que pode ser tanto governamental quanto privada) quanto ao período até o título ser pago, o que implica no mesmo estar sujeito a mudanças na taxa de juros e outros fatores.

2. Títulos de participação no capital: Não promete nenhum tipo de pagamento em particular, mas sim representa a posse de parte da instituição, e com isso tem direito a receber parte de quaisquer dividendos que venham a ser pagos. Por essa razão, o valor desse título é diretamente ligado ao sucesso ou fracasso da instituição em questão, sendo assim um investimento mais arriscado que os títulos de renda fixa.
3. Derivativo: De acordo com Gordon (2022), derivativos são um contrato financeiro entre duas partes para a compra ou venda de ativos a um preço predeterminado em um período específico de tempo. O valor de um derivativo provém da performance de um ativo subjacente ou de um grupo de ativos, esses que são negociados separadamente dos derivativos. Dependendo do tipo de derivativo, há vários possíveis usos do mesmo, mas no geral são utilizados para projetar o preço futuro de um ativo, evitando quaisquer problemas de taxa de câmbio e cobrindo contra mudanças nos valores dos ativos.

2.2 Mercado

Nesta subseção serão introduzidos alguns conceitos e terminologia utilizada relativo ao mercado financeiro propriamente dito.

2.2.1 Ursos e touros

Como explanado por Kramer (2022), no meio de investidores e traders, os termos *Bull/Bullish* (Da palavra em inglês Touro) e *Bear/Bearish* (Da palavra em inglês Urso) são comumente usados para descrever as condições do mercado, isto é, se ele está ganhando ou perdendo valor. E para cada investidor, a direção do mercado é um fator que apresenta um grande impacto na composição do seu portfólio (conjunto de todos os investimentos financeiros de um determinado investidor/trader).

A terminologia usada não tem uma origem exata, mas tende a ser atribuída ao modo que ursos e touros tendem a atacar suas presas, o qual ursos jogam suas presas ao

chão com as patas, já de modo contrário, touros levantam o alvo no ar com os chifres. Esse paralelo de derrubar e levantar é usado para representar a tendência geral do mercado, tornando fácil a comunicação da situação vista por cada trader.

2.2.2 Preços de abertura, fechamento, volume e etc

Nos gráficos que serão vistos na próxima seção, os termos preço de abertura, preço de fechamento, ponto alto, ponto baixo e volume serão utilizados regularmente por serem exatamente as informações que os gráficos buscam representar. Dito isso, as definições de cada um desses termos como apresentada por Smigel (2022) se da como nos seguintes itens:

1. Preço de abertura: Como o nome indica, o preço de abertura consiste no preço estimado para determinado ativo no horário de abertura da bolsa de valores naquele dia específico. Geralmente o preço tende a ser o mesmo/similar ao preço de fechamento do dia anterior, porém fatores como relatórios de rendimentos entre outros que podem ser divulgados após o horário de fechamento da bolsa, podem influenciar diretamente no valor;
2. Preço de fechamento: O preço de fechamento refere-se ao preço de um ativo específico quando a bolsa de valores fecha as transações do dia, representando a última ordem de compra e venda executada entre dois traders;
3. Ponto alto e baixo: O valor de ponto alto e baixo consiste no maior e menor preço ,respectivamente que uma venda daquele ativo foi efetuada no período de tempo. Os usos mais comuns de análise de ponto alto e baixo são diários e os de 52 semanas, que consiste no maior e menor valor de transação no período de um ano.
4. Volume: Volume ou volume de negociação é uma medida da quantidade de um certo ativo financeiro que foi vendido em um certo período de tempo. Para ações, volume é uma medida do número de ações vendidas, já para títulos seria baseado em quantos contratos mudaram de mãos.

2.3 Gráficos

De acordo com HAYES, as transições entre tendências ascendentes (touros) e descendentes (ursos) são sinalizadas por padrões de preços, sendo uma configuração

reconhecível que é definida por uma série de linhas de tendência ou curvas.

A análise técnica de gráficos busca padrões de preços antigos para examinar o movimento atual e prever as futuras tendências do mercado. Para isso traça-se linhas de tendência que ajudam a identificar áreas de suporte e resistência em um gráfico de preço.

Quando um padrão de preço muda a direção da tendência, é conhecido como um padrão de reversão, enquanto que uma tendência que continua no mesmo sentido após uma breve interrupção é conhecido como um padrão de continuação.

Para Elder (2002), "Fazer *trades* sem gráficos é como jogar poker sem olhar para as cartas". Os gráficos são onde os traders conseguem visualizar a constante luta entre os touros e ursos, mas apenas os que aprendem a ler suas pegadas vão conseguir prever seus movimentos.

2.3.1 Tipos de gráficos

Ao explicar as bases da análise técnica, Schwager (1999) descreve que os gráficos de preços são a ferramenta primária do analista técnico. E apesar de existir uma variedade de formatos, a maioria dos gráficos usam um sistema de grade, onde o eixo X mede tempo enquanto que o eixo Y mede o preço.

O incremento de tempo do eixo X pode variar de acordo com o quão distante são os dados que o trader quer relevar em sua análise, podendo ser construídos com qualquer intervalo: anual, mensal, semanal, diário (considerado o mais comum) e intra-diário (ex: 60 minutos, 30 minutos, etc.).

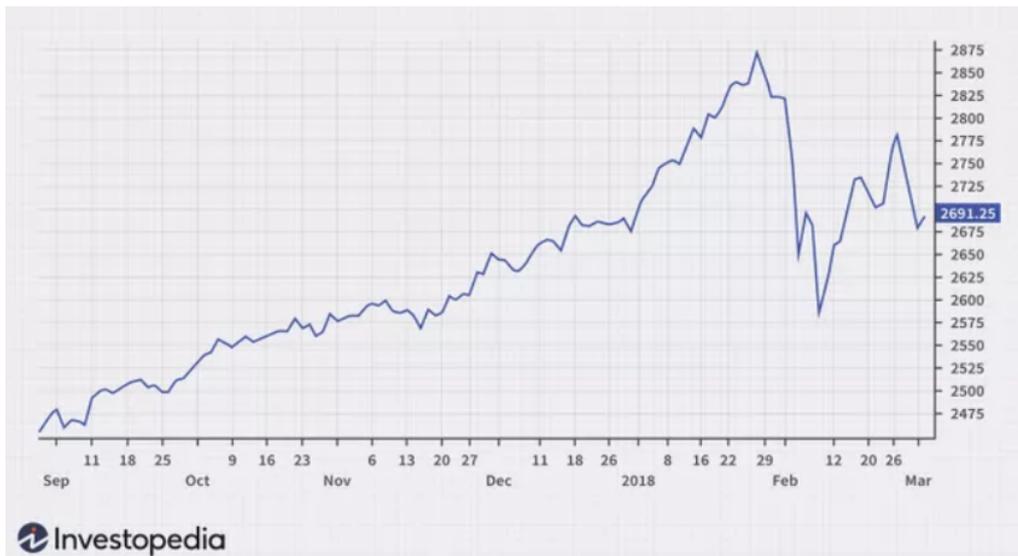
Há três tipos principais de gráficos usados para visualizar o histórico de preços:

1. Gráfico de linha:

De acordo com Peters (2022), o gráfico de linha é a representação gráfica da história de preço de ativos que conecta uma série de pontos de dados em uma linha contínua, sendo o tipo mais básico de gráfico usado na área financeira e tipicamente apenas representa o preço de fechamento ao longo do tempo. Um exemplo pode ser visto na figura 1.

Gráficos de linha podem ser usados em qualquer formato de tempo, mas geralmente é usado em mudanças de preço diárias. Seu principal diferencial é a simplicidade e a remoção de "ruídos" que geralmente consistem de dados menos críticos do dia (preços de abertura, alta e baixa). Essa redução pode ser útil para traders iniciantes

Figura 1 – Exemplo de gráfico de linha



Fonte: (PETERS, 2022)

ou em casos de conflitos de leitura, onde a análise não prossegue por excesso de dados, efeito esse chamado "paralisia por análise".

2. Gráfico de barra:

De acordo com Mitchell (2021), o gráfico de barras consiste em múltiplas barras de preço, sendo que cada barra ilustra como o preço de um ativo ou título mudou durante um período específico de tempo, com cada barra mostrando o preço de abertura, alta, baixa e fechamento, podendo porém ser ajustada para omitir o preço de entrada. Um exemplo pode ser visto na figura 2.

As barras são compostas por uma linha vertical que mostra o maior e o menor preço alcançado naquele período, o preço de abertura é representado por uma pequena linha horizontal à esquerda da linha vertical e o preço de fechamento é representado por uma linha a direita da linha vertical. Em alguns casos, quando o preço de fechamento está acima do preço de abertura, a barra pode ser colorida de preto ou verde, caso contrário ela pode ser vermelha, sendo essa codificação por cores uma ferramenta disponível em grande parte das ferramentas de trading.

Figura 2 – Exemplo de gráfico de barra



Fonte: (MITCHELL, 2022)

3. Gráfico de Velas:

De acordo com Mitchell (2022), gráficos de velas são capazes de mostrar os quatro pontos de preço (abertura, fechamento, alta e baixa) no período que o trader especificar. O gráfico de vela é composto por uma área mais larga chamada de "corpo real", sendo que esse corpo real representa a faixa de preço entre a abertura e o fechamento daquele dia. Além da codificação de cores semelhante a do gráfico de barras (preto ou verde quando o fechamento está acima do preço de abertura e vermelho caso contrário), outro padrão possibilitado pelo modelo de vela é o de que quando o "corpo real" da vela está preenchida, significa que o fechamento foi mais alto que a abertura, e caso contrario o corpo fica vazio (sem preenchimento). Um exemplo pode ser visto na figura 3.

Figura 3 – Exemplo de gráfico de vela



Fonte: (MITCHELL, 2022)

2.4 Tendências

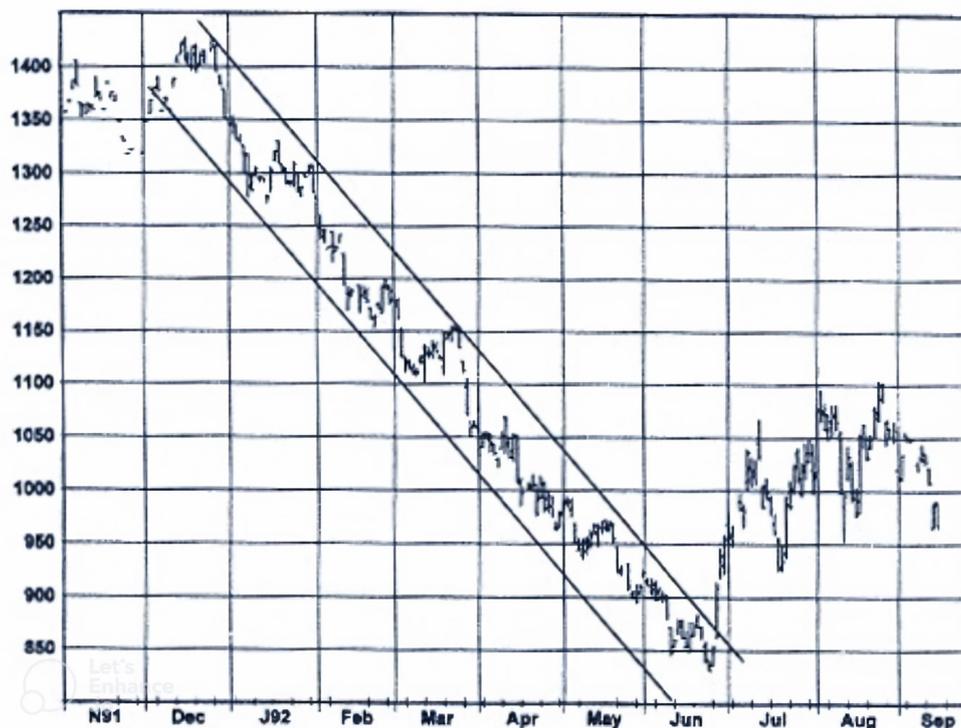
Um dos campos de análise mais relevantes ao se tratar do mercado financeiro através da corrente de pensamento técnica é justamente a identificação e análise de tendências. Schwager (1999) descreve que o propósito mais básico de análise de gráficos é definir e identificar tendências de preço, onde uma tendência de crescimento é uma sequência de altas relativas mais altas que as anteriores e baixas relativas mais significativas que as baixas relativas anteriores.

De maneira similar, tendências de queda consistem em uma sequência de altas relativas menores que as anteriores assim como baixas relativas cada vez menores. Tendências de crescimento e queda são consideradas intactas até que um ponto de baixa relativa quebre a barreira de uma baixa anterior (no caso de crescimento) ou que um ponto de alta quebre a barreira de uma alta anterior (no caso de queda), sendo que a violação dessa condição pode indicar que essa tendência pode ter terminado.

2.4.1 Linhas de tendência

Linhas de tendência costumam ser traçadas conectando pontos baixos (para tendências de crescimento) ou altos (para tendências de queda) significantes, existindo espaço de debate sobre o número de pontos necessários para a criação de uma linha de tendência válida, com dois sendo o absoluto mínimo e a significância da linha traçada aumentando proporcionalmente ao número de pontos que ela conecta. (SCHWAGER, 1999)

Figura 4 – Exemplo de linhas de Tendência



Fonte: Schwager (1999)

Na Figura 4 podemos observar um exemplo de linhas de tendência de queda, o qual é traçado a partir de quatro pontos de alta e de três pontos de baixa.

2.5 Indicadores

De acordo com Elder (2014), indicadores servem para ajudar a identificar tendências e viradas do mercado, provendo dados mais objetivos que os gráficos e mais percepção do balanço de poder entre touros e ursos. Porém o uso de indicadores diferentes implica na eventual contradição dos mesmos, com alguns funcionando melhor nos mercados em alta e outros em mercados estáveis, por isso é recomendado que cada trader selecione um número pequeno de indicadores que funciona melhor para si e combiná-los de maneiras inteligentes para cada tipo de mercado.

A maioria dos indicadores são baseados nas mesmas cinco categorias de dados: abertura, alta, baixa, fechamento e volume. Por causa disso, mesmo que sejam usados dez, vinte ou cinquenta indicadores, a análise não será sempre aprimorada, já que a base de dados que está sendo analisada é sempre a mesma.

Para Elder (2014), existem três tipos de indicadores:

- Indicadores de tendência: Funcionam melhor quando o mercado está se movendo, porém a qualidade do sinal deteriora rapidamente quando o mercado estabiliza. Ex: MA (Médias móveis - Sigla do inglês *Moving Averages*), Linhas MACD (Médias móveis convergentes-divergentes), Sistema de movimento direcional, OBV (On-Balance Volume), Acumulação/Distribuição, e outros.
- Indicadores osciladores: Buscam identificar os pontos de virada em mercados estáveis, mas dão sinais prematuros e perigosos quando surgem tendências. Ex: Histograma-MACD, Force Index, Stochastic, Taxa de mudança, Momento, Índice de força relativa, Elder-ray, Williams %R e outros.
- Indicadores miscelâneos: Buscam identificar a intensidade do mercado em tendências de altas ou quedas. Ex: New High–New Low Index, the Put-Call Ratio, Bullish Consensus, Commitments of Traders, e outros.

2.5.1 Médias Móveis

A média móvel reflete o valor médio dos dados na janela de tempo definida, com a MA de 5 dias mostrando o preço médio dos últimos 5 dias, a MA de 20 dias o preço dos últimos 20 dias, etc.

De acordo com Elder (2006), médias móveis refletem o consenso médio do valor de um ativo em uma janela de tempo. Ela reflete duas importantes mensagens ao trader: A primeira é sobre a direção da curva, onde uma subida reflete um crescente otimismo (touro), enquanto que uma descida reflete um crescente pessimismo (urso).

A segunda mensagem seria sobre o que Elder chama de "valor da transação" e "teoria do maior tolo". Onde se o trader comprar perto da MA, ele está comprando no valor ideal, já alguém que compra muito acima da MA, está dizendo "Sou um tolo, mas espero achar um tolo ainda maior depois.". Este tipo de pensamento pode funcionar algumas vezes, porém a longo prazo sempre buscar comprar no valor ideal vai se provar uma estratégia muito mais sensata.

Há dois tipos principais de médias móveis: simples e exponencial:

- Simples: Consiste na média simples dos valores assumidos no período analisado. Apresenta o grande problema de responder duas vezes a cada mudança de preço, uma vez para o valor que entrou e uma para o valor que saiu, podendo apresentar

comportamentos inconsistentes com a realidade em espaços menores de tempo (já que cada valor tem um peso maior na média final).

A equação da média móvel simples pode ser vista na equação 1:

$$MA = \frac{P_1 + P_2 + \dots + P_N}{N} \quad (1)$$

Onde:

P são os preços do intervalo.

N é o número de dias do intervalo.

- Exponencial: Consiste em uma função recursiva que dá maior peso aos últimos dias calculados e responde mais rápido que a média móvel simples sem dar saltos ao descartar dados antigos.

A equação da média móvel exponencial pode ser vista na equação 2 e 3:

$$EMA = P_{tod} \cdot K + EMA_{yest} \cdot (1 - K) \quad (2)$$

$$K = \frac{2}{N + 1} \quad (3)$$

Onde:

P_{tod} = É o preço do dia atual.

N = É o número de dias do intervalo.

EMA_{yest} = É a média exponencial média do dia anterior.

2.5.2 MACD - Convergência/Divergência de médias móveis

Como mostrado por Elder (2014), médias móveis ajudam a identificar tendências e reversões nos gráficos. Com isso em mente, o analista Gerald Appel criou o indicador avançado de Convergência/Divergência de médias móveis (MACD), que consiste de três médias móveis exponenciais simultâneas cujos cruzamentos dão sinais de compra e venda.

Um MACD consiste de dois tipos de linha: as sólidas (chamada de linha MACD) e a pontilhada (chamada de linha de sinal). As sólidas são compostas pela diferença de duas EMAs (Média exponencial aritmética), as quais respondem as mudanças de preço de forma relativamente rápida. Já a linha de sinal é outra EMA, porém é designada para

responder mais devagar as mudanças. No sistema original os sinais de compra e venda eram dados quando a linha MACD rápida cruzava acima ou abaixo da linha de sinal mais lenta.

O indicador MACD já é muito difundido no mercado financeiro, por isso é incluído na maioria dos programas de análise técnica. Mas para criá-lo a manualmente basta seguir os seguintes passos:

1. Calcule o EMA do preço de fechamento dos últimos 12 dias.
2. Calcule o EMA do preço de fechamento dos últimos 26 dias.
3. Subtraia o EMA de 26 dias do EMA de 12 dias e trace a diferença em uma linha sólida. Essa é a linha MACD.
4. Calcule o EMA de nove dias da linha MACD e trace o resultado numa linha pontilhada. Essa é a linha de sinal.

2.5.3 MACD - Histograma

Histograma MACD oferece um entendimento no equilíbrio de poder entre touros e ursos melhor que a MACD original. Mostrando não apenas se os touros ou ursos estão no controle mas se estão crescendo ou diminuindo. (ELDER, 2014)

$$\text{Histograma MACD} = \text{Linha MACD} - \text{Linha de sinal} \quad (4)$$

Como visto na equação 4, histograma-MACD mede a diferença entre a linha MACD e a linha de sinal e traça a diferença como um histograma. Como descrito por (CHEN, 2022), histogramas são uma representação de agrupamento de dados por certos intervalos de tempo, condensando esses dados em barras verticais facilmente interpretáveis.

De maneira prática, quando as barras estão crescendo acima do zero pode-se entender que os compradores estão dominando o mercado, sendo assim uma boa hora de investir a longo prazo, já se a barra está crescendo para abaixo da linha do zero entende-se que os vendedores estão dominando, passando a ideia que é melhor investir a curto prazo. Nesse mesmo sentido, quando as barras estão crescendo na mesma direção dos preços, o investimento tende a ser seguro, enquanto que em casos onde as barras crescem na direção oposta dos preços, o risco tende a ser consideravelmente maior. Elder (2014)

2.5.4 Estocástico

Como explicado por Schwager (1999), estocástico é um oscilador que mede o momento comparando o preço de fechamento recente com o intervalo de preço absoluto (Ponto mais alto do intervalo subtraído do ponto mínimo) em um período de N dias. A equação do estocástico pode ser vista na equação 5

$$\%K = \frac{C_t - L_n}{H_n - L_n} \cdot 100 \quad (5)$$

Onde:

C_t = É o preço de fechamento do dia atual.

H_n = É o maior preço nos últimos N dias.

L_n = É o preço mais baixo nos últimos N dias.

Já a segunda linha do indicador Estocástico consiste na suavização da linha %K (geralmente em um período de três dias) para o qual há vários métodos, um exemplo sendo visto na equação 6:

$$\%D = \frac{\text{Soma de 3 dias de } (C_t - L_n)}{\text{Soma de 3 dias de } (H_n - L_n)} \cdot 100 \quad (6)$$

Essa fórmula normalmente calculada num período de 5 dias consiste no que é chamado de Estocástico Rápido, a qual apesar de sensível tende a apresentar muito ruído dos dados, e para solucionar esse problema foi criado o chamado Estocástico Lento (que normalmente é o usado e chamado somente de "Estocástico").

O Estocástico Lento utiliza o %D do rápido como seu %K, e em seguida aplica-se uma suavização no novo %K para obter-se o novo %D. Esse modelo é capaz de melhor filtrar os ruídos do mercado e apresenta um gráfico mais suave.

2.5.5 Índice de Força Relativa

Larson (2012) explica, que o Índice de Força Relativa (mais comumente referenciado pela sigla RSI de *relative strength index*) compara os ganhos com as perdas de determinadas ações, sendo utilizado de maneira similar ao Estocástico, onde busca-se comprar quando o valor passa do nível dos 20% e considera-se vender quando cruza os 80%.

De acordo com Jack Schwager (1999), o indicador RSI foi introduzido por Welles

Wilder Jr. em seu livro *New Concepts in Technical Trading Systems (1978)*, onde o autor mostra a fórmula vista na equação 7:

$$RSI = 100 - \left[\frac{100}{1 + RS} \right] \quad (7)$$

Onde:

RS = Força relativa = Média dos preços de fechamento em alta no período dividido pela média dos preços de baixa.

O valor da força relativa é tudo o que importa para o RSI, o resto da equação apenas normaliza o valor na escala de 0 a 100. Os fechamentos de alta e baixa são classificados a cada dia, consistindo na comparação do valor de fechamento do dia com o fechamento do dia anterior, com o valor absoluto de cada sendo a diferença dos mesmos.

2.6 Risco

Como visto em Bodie, Kane and Marcus (2009), risco é um fenômeno inevitável em todas as transações de ativos, tanto para os investidores quanto para as empresas. Ainda que o nível de risco possa ser escolhido de acordo com o investimento desejado, como por exemplo quando uma empresa automobilística constrói uma nova fábrica, não há certeza alguma que essa fábrica irá gerar um fluxo monetário que eventualmente justifique sua construção. Com isso, o mercado financeiro permite que a empresa ofereça esse risco maior aos investidores interessados que acreditam que esse risco vai resultar em retornos maiores a longo prazo, enquanto que os investidores mais conservadores podem evitar esse tipo de investimento e comprar apenas os de baixo risco (como títulos de renda fixa).

Esse sistema beneficia até mesmo empresas menores que precisam levantar capital para investimentos iniciais, onde cada uma pode oferecer diferentes tipos de riscos e seguranças para variados tipos de investidores, facilitando assim a construção do estoque de ativos da empresa.

Em seu livro, Douglas (2013) foca no lado psicológico adotado pelos traders de sucesso, observando que aprender a aceitar riscos em geral pode ser difícil, e ainda mais para traders quando consideramos o que está em jogo, que seria perder dinheiro por um erro absolutamente próprio. Esse pensamento que leva traders iniciantes a querer fazer de tudo para evitar os riscos é justamente uma das principais coisas que os impede de

realmente alcançar o sucesso na área. Aceitar o risco e manter-se focado no que está acontecendo sem a resposta emocional de perdas e riscos é um traço fundamental do profissional da área.

2.6.1 Controle de riscos

De acordo com Elder (2006), novos traders são facilmente seduzidos pelas possibilidades do uso de indicadores técnicos, com a mentalidade que é possível ganhar lucro em todas as operações, mas os traders mais experientes sabem que o controle de risco é tão importante quanto, já que quando inevitavelmente uma perda ocorrer, não é viável todo o lucro de semanas ou meses desapareça de uma vez só.

Com isso em mente, os dois pilares mais populares para controle de risco são a regra dos 2% e a regra dos 6%:

1. **Regra dos 2%:** Essa regra diz que em um único trade, não mais que 2% do valor total da conta pode ser posto em risco. Isso não quer dizer por exemplo, que alguém com uma conta de R\$100,000 pode comprar apenas R\$2,000 de ações, mas sim que calculando-se a diferença da ordem de parada pelo preço de entrada, esse valor não pode passar dos 2%.

Ex: Uma ação cujo preço unitário está sendo vendido a R\$24, o trader decide numa ordem de parada aos R\$20, e sabe que a conta atual que está trabalhando tem um saldo de R\$200,000. Como em cada ação o risco está em R\$4 e o seu limite de risco está em R\$4,000. Nesta trade individual ele pode comprar até 1,000 ações ou menos cumprindo a regra dos 2%.

2. **Regra dos 6%:** Essa regra tem como objetivo limitar o risco para a conta do trader como um todo, declarando que ao mesmo tempo, não mais que 6% da conta pode estar em risco de perda. Esta regra permite que o trader mantenha mais *trades* abertas quando está lucrando (o que implica que o mercado está em um bom estado), mas que vá mais devagar quando começa a perder dinheiro.

Ex: Com uma conta de R\$200,000 e risco de R\$2,000 em cada trade. Não mais que 6 *trades* simultâneas podem estar abertas, e digamos que duas dessas *trades* deram errado, agora até o fim do mês só é permitido quatro *trades* abertas, pois os 2% perdidos implicam que apenas 4% do risco está ainda aberto.

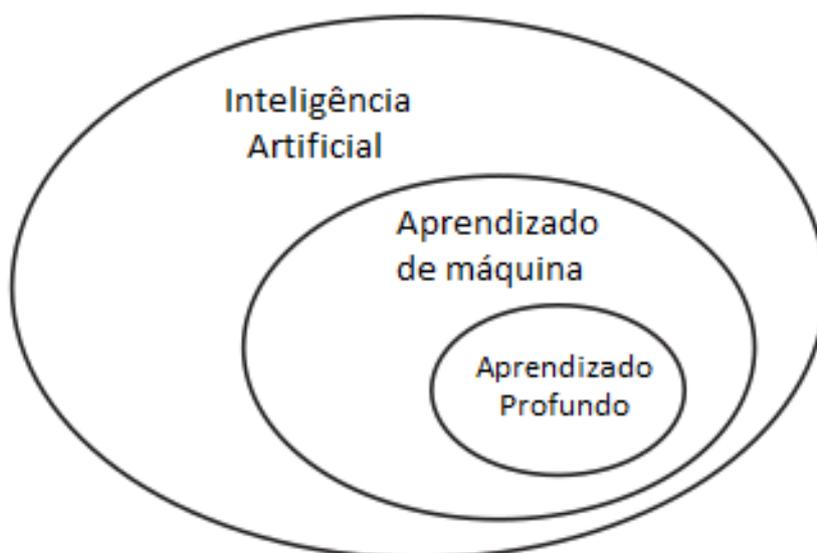
2.7 Revisão de técnicas e metodologias de inteligência artificial

Nesta sessão iremos explorar alguns conceitos básicos de inteligência artificial e seus campos relacionados, eventualmente aprofundando no funcionamento das tecnologias utilizadas nesse trabalho.

2.8 Diferença de Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo

Os termos Inteligência Artificial (do inglês *artificial intelligence* - IA), Aprendizado de Máquina (do inglês *machine learning* - ML) e Aprendizado Profundo (do inglês *Deep Learning* - DL) são extensamente utilizados na área de computação como exemplos das tecnologias revolucionárias que proporcionaram ao século 21, ferramentas e produtos antes considerados possíveis somente em trabalhos de ficção científica. Apesar de esses termos serem comumente usados intercambiavelmente, cada um tem significado diferente e definições específicas que são de suma importância na organização e compreensão da área de estudo como um todo.

Figura 5 – Ilustração da hierarquia de IA, ML e DL



Fonte: Chollet (2017)

Na Figura 5 tirada do livro de Chollet (2017), podemos observar um diagrama de Venn representando a relação dos campos de inteligência artificial, onde esta abrange as

outras duas e aprendizado de máquina engloba aprendizado profundo. Mas para descrever essa relação adequadamente, será necessário explorarmos cada uma individualmente e definir o que cada uma envolve.

2.9 Inteligência Artificial

Chollet (2017) descreve que historicamente o conceito de inteligência artificial surgiu por volta dos anos 50, onde pioneiros do campo de ciência da computação se questionaram se seria eventualmente possível fazer computadores "pensarem". A definição inicial concisa do que esse campo de pesquisa representaria é descrita como: "*A capacidade de automatizar tarefas intelectuais normalmente afetadas por humanos.*".

A partir dessa definição podemos começar a observar o porquê do campo de AI englobar ML e DL, porém mais do que isso, há diversos outros procedimentos que não envolvem nenhum tipo de aprendizado. Um grande exemplo disso seria os primeiros programas de xadrez, os quais eram programados diretamente com as regras do jogo e jogavam simulando posições internamente, não tendo qualquer tipo de aprendizado.

Inicialmente era teorizado que eventualmente a criação de uma inteligência comparável a humana seria possível através da criação de um conjunto de regras suficientemente grandes, porém esse conceito foi continuamente defasado com a aparição de problemas mais complexos e difíceis de definir, como classificação de imagem, reconhecimento de voz e tradução de idiomas.

2.9.1 Inteligência artificial forte versus fraca

Camposato (2020) descreve que atualmente existem dois campos principais relacionados à AI. O campo de inteligência artificial fraca que é associada ao *Massachusetts Institute of Technology* e classifica qualquer sistema que exiba comportamento inteligente como exemplo de AI. Por essa métrica, o foco está em se o programa executa suas tarefas corretamente, não importando se faz do mesmo jeito que humanos, e por essa característica tende a ser a abordagem utilizada dos campos de engenharia elétrica, robótica e outros campos que somente a performance é relevante.

Por outro lado, inteligência artificial forte é associada à *Carnegie-Mellon University* e é baseada em um conceito chamado plausibilidade biológica (*biological*

plausibility). Esse conceito diz que em casos que um sistema apresenta comportamento inteligente, sua performance precisa ser baseada nas mesmas metodologias utilizadas com humanos. Um exemplo de comparação seria um sistema de captação de sons: um sistema de AI forte buscaria executar a tarefa simulando o sistema auditivo humano, incluindo partes equivalentes a cóclea, canal auditivo, tímpano, etc. Já um sistema de AI fraco iria relevar somente a performance do sistema.

2.10 Aprendizado de Máquina

Em seu livro, Burkov (2019) deixa claro desde o começo que nenhuma máquina tem a real capacidade de aprender, e que de modo simples, o que uma típica "máquina que aprende" faz é buscar uma fórmula matemática que, quando recebe um conjunto de dados (chamado de dados de treinamento), produz um tipo específico de resultado. Essa fórmula quando encontrada tende a gerar resultados corretos para a maioria de outros dados (diferentes dos de treinamento) inseridos na máquina, na condição de que esses dados tenham a mesma ou similar distribuição estatística dos dados de treinamento.

O motivo da escolha do nome aprendizado de máquina é atribuído a Arthur Samuel, que popularizou o termo em 1959 enquanto trabalhava na IBM, porém apesar do nome, ao invés de qualquer relação com inteligência real, pode-se descrever aprendizado de máquina como a **ciência e engenharia de construção de máquinas capazes de desempenhar diversas funções úteis sem serem explicitamente programadas para tal.**

2.11 Aprendizado profundo

Bengio Yoshua; Courville (2016) aponta que a maior fonte de dificuldade em várias aplicações reais de inteligência artificial é que há muitos fatores de variação que influenciam cada parte dos dados que são observados como, por exemplo, os pixels em uma imagem de um carro vermelho que se tornam quase pretos a noite. Outro exemplo seria a forma do carro que muda de acordo com o ângulo observado, e justamente problemas assim que requerem uma avaliação dos fatores de variação que identifique os mais significativos e descartem aqueles que não são relevantes.

Deep learning então busca solucionar esse problema utilizando representações

expressas em termos de outras representações mais simples, permitindo ao computador construir conceitos complexos a partir de outros mais básicos. Sobre o exemplo anterior, um sistema de DL (*Deep Learning*) buscaria representar o conceito da imagem de uma pessoa combinando outros mais simples, como curvas, cantos e arestas.

2.12 Facebook PROPHET

Em seu trabalho, Taylor and Letham (2017) descrevem que o facebook PROPHET é um modelo de previsão de dados em séries temporais com base em um modelo aditivo em que as tendências não lineares são ajustadas à sazonalidade anual, semanal e diária, criado para tentar solucionar o grande problema atual de alta demanda de previsões de alta qualidade relativo á dificuldade de sua criação. Tais previsões requerem analistas com experiência extensa na área sobre os produtos e serviços específicos que estão trabalhando, além de conhecimento em previsões em séries temporais, definitivamente uma combinação que não é fácil de encontrar para todas as áreas que requerem tais leituras.

Dado esse problema, buscou-se desenvolver uma solução que fosse capaz de prover uma alternativa mais utilizável sem a especialização necessária mas que também produzisse as predições em escala. Definição essa de escala que é definida de três modos diferentes:

1. Um grande número de pessoas fazendo previsões, possivelmente sem o treinamento em métodos temporais.
2. Uma grande variedade de problemas de previsão com comportamentos potencialmente incomuns e extraordinários.
3. Em uma situação realista, um grande número de previsões vão ser criadas, requerendo meios automatizados eficientes de avaliá-las e compará-las, além de detectar quando eles estão possivelmente apresentando performance baixa.

2.12.1 Sazonalidade

Os dados extraídos tendem a apresentar tendências relativas a comportamentos temporais, definidas como sazonalidades, que podem ser em ciclos diários, semanais ou anuais, isto também sendo afetados por feriados e outros acontecimentos aparte do

ciclo. Essas tendências temporais tendem a aparecer em dados onde a fonte é relativa ao comportamento humano, seja por normalmente terem como público alvo a previsão de ações humanas como por exemplo compras em determinada loja, ou até mesmo hábitos menos conscientes como tráfego em certa rua.

Modelos que não consideram sazonalidades como o Auto-ARIMA (modelo auto-regressivo integrado de médias móveis) e outros que utilizam sazonalidades mas de maneira limitada como o Snaive e o TBATS, tendem a apresentar dificuldades em produzir previsões de um período maior por não conseguir adequar as características temporais dos problemas, como por exemplo respondendo muito fortemente às mudanças de fim de ano, tornando o modelo inconsistente por um certo período após a leitura.

2.12.2 Modelo

O modelo do facebook PROPHET foi criado para ser capaz de lidar com as características mais comuns de previsões temporais, e mais importante, também foi desenvolvido para utilizar parâmetros intuitivos que podem ser ajustados dado o conhecimento dos detalhes do funcionamento do modelo.

As previsões são calculadas a partir de três componentes principais como visto na Equação 8:

$$y(t) = g(t) + s(t) + h(t) + e_t \quad (8)$$

Onde:

$g(t)$ = É a função de tendência que modela mudanças não periódicas no valor da série temporal.

$s(t)$ = Representa mudanças periódicas (sazonalidades).

$h(t)$ = Representa os efeitos de feriados que ocorrem em datas irregulares durante um ou mais dias.

e_t = Representa quaisquer mudanças e irregularidades que não estão acomodadas pelo modelo.

Diferentemente de outros modelos de previsão como o ARIMA que explicitamente lida com a estrutura de dependência temporal dos dados, o modelo do PROPHET trata o problema de previsão como um exercício de ajustar uma curva nos dados. Tal abordagem acaba por não aproveitar de vantagens importantes destes outros

modelos, mas em troca tem uma série de vantagens, tais como:

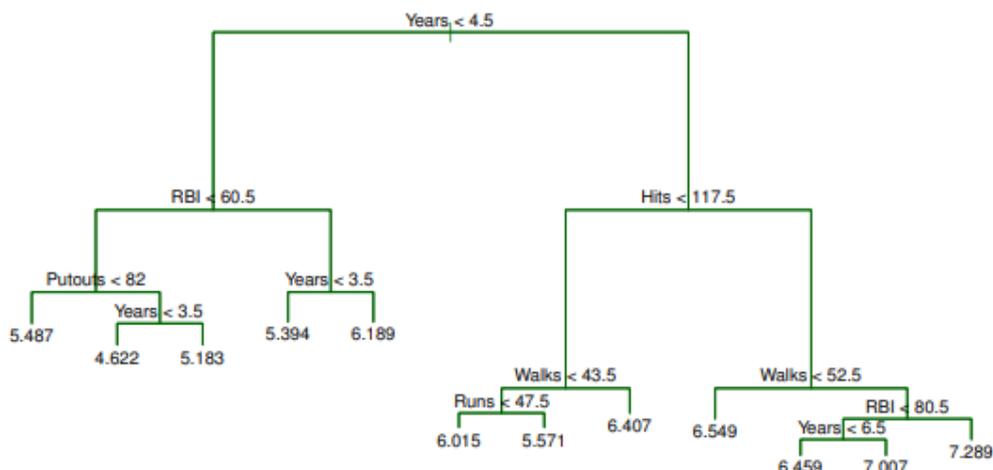
- **Flexibilidade:** Facilmente acomoda sazonalidades com múltiplos períodos, permitindo ao analista tirar conclusões sobre as tendências.
- **Preparação:** Diferentemente do modelo ARIMA, as medidas não tem que ter um espaçamento regular, não precisam da interpolação de valores faltantes e não apresentam a necessidade de remoção de outliers.
- **Velocidade:** O ajuste é bastante rápido, permitindo que o analista explore iterativamente diversos modelos.
- **Interpretabilidade:** Os parâmetros são facilmente interpretáveis, tornando possível que o analista tire suposições dos dados observados. Além disso, os mais experientes com regressão podem facilmente estender o modelo para incluir novos componentes.

2.13 Árvores de decisão

As árvores de decisão são um método popular de aprendizado de máquina, particularmente no campo da inteligência artificial, o qual pertence a família dos algoritmos de aprendizado supervisionado que são usados quando são empregados dados rotulados, ou seja, dados com um resultado definido. Eles são um tipo de algoritmo que permite que um computador faça previsões ou decisões com base em determinados dados de entrada.

Uma árvore de decisão é uma estrutura semelhante a um fluxograma, onde cada nó interno representa um "teste" em um atributo (por exemplo, se o lançamento de uma moeda sai cara ou coroa), cada ramo representa o resultado do teste e cada nó folha representa um rótulo de classe (decisão tomada após computar todos os atributos). O nó mais alto em uma árvore de decisão é conhecido como nó raiz, o qual aprende a particionar os dados em subconjuntos com base nos valores dos atributos. Um exemplo de árvore de decisão pode ser vista na Figura 6

Figura 6 – Exemplo de árvore de decisão



Fonte: (JAMES et al., 2013)

A ideia básica por trás das árvores de decisão é particionar recursivamente os dados em subconjuntos com base nos valores dos recursos de entrada. Cada partição é representada por um ramo na árvore, e a decisão final ou previsão é feita nos nós das folhas. O processo de construção de uma árvore de decisão envolve a seleção do melhor atributo para dividir os dados em cada etapa, e isso geralmente é feito usando uma medida de impureza, como entropia ou impureza de Gini. (MITCHELL, 1997)

Uma das principais vantagens das árvores de decisão é a sua interpretabilidade. A estrutura da árvore permite uma fácil visualização do processo de tomada de decisão, e as regras aprendidas pela árvore podem ser facilmente compreendidas pelo ser humano. Além disso, as árvores de decisão podem lidar com dados categóricos e numéricos e não são sensíveis ao dimensionamento dos recursos. Isso torna as árvores de decisão uma excelente opção para exploração de dados e para entender o relacionamento subjacente entre as entradas e as saídas.

No entanto, as árvores de decisão também apresentam algumas limitações. Em alguns casos elas são propensas a *overfitting* e além disso, as árvores de decisão podem ser computacionalmente caras para treinar e avaliar, especialmente para grandes conjuntos de dados.

Overfitting ocorre quando um modelo é muito bom em ajustar os dados de treinamento e não consegue generalizar para novos dados. Isso ocorre porque ele aprendeu o ruído nos dados e não os padrões subjacentes. No caso de árvores de decisão,

isso pode acontecer quando a árvore se torna muito profunda e complexa, capturando até mesmo as menores flutuações nos dados. (JAMES et al., 2013)

Existem várias maneiras de detectar o *overfitting*, como observar o desempenho do modelo em um conjunto de validação separado ou comparar o desempenho do modelo no conjunto de treinamento e no conjunto de teste. Algumas técnicas comuns para evitar o *overfitting* nas árvores de decisão são a poda, que envolve a remoção de ramos da árvore que não agregam muito valor às previsões, ou o uso de métodos *ensemble* como florestas aleatórias, que combinam várias árvores de decisão para reduzir o *overfitting*.

Outro aspecto importante das árvores de decisão é a capacidade de lidar com valores ausentes. No algoritmo da árvore de decisão, os valores ausentes são tratados usando *surrogate splits*, a qual consiste na divisão feita por variáveis que não são usadas na árvore de decisão, mas são usadas para lidar com valores omissos.

Em conclusão, as árvores de decisão são um método poderoso e interpretável para aprendizado de máquina e inteligência artificial. Eles são fáceis de entender, funcionam bem com dados categóricos e numéricos e não são sensíveis ao dimensionamento de recursos. Eles também são robustos a valores ausentes. No entanto, eles podem ser propensos a *overfitting* e computacionalmente caros.

3 TRABALHOS CORRELATOS

Na bibliografia podem ser encontrados trabalhos, artigos e projetos previamente feitos onde tecnologias, mercado alvo ou metodologia de seleção de indicadores tem relação com o presente trabalho. Nesta seção buscaremos explorar esses trabalhos e descobrir quais ideias e conceitos já foram executadas com sucesso, e esclarecer quais campos ainda tem potencial para serem explorados e quais as principais diferenças e críticas que podem ser observadas em cada um.

Os trabalhos buscados apresentaram o uso do Facebook Prophet em conjunto ou não de outras tecnologias aplicado na previsão de preços nas bolsas de valores. Buscou-se também selecionar os trabalhos com relação a previsão de índices ou de fundos de investimento relacionados ao S&P500, porém dado a escassez de trabalhos dessa natureza acabou por ser utilizado uma flexibilidade maior no tema de índice alvo.

O trabalho de Li and Bastos (2020) busca em 34 artigos, selecionados utilizando metodologia de pesquisa, revisar as técnicas de *Deep Learning* (Aprendizagem profunda) e análise técnica aplicados a provisão do mercado financeiro, através dos quais constatou-se que a metologia mais explorada atualmente é a aplicação de redes neurais recorrentes (ou também comumente chamadas LSTMs / *Long-Short Term Memory*), aproveitando-se da sua capacidade de resolver problemas como o gradiente explosivo e gradiente desaparecido. LSTMs também podem ser vistas em modelos híbridos, onde uma combinação do mesmo tratando indicadores técnicos e outras tecnologias como reconhecimento de texto aplicado a fontes de notícias resulta nesses modelos apresentando uma tendência de performance melhor quando comparada aos que utilizaram somente LSTMs. Apenas cerca de metade dos artigos fazia uso de quaisquer estratégias de *trading*, dado que a outra metade simplesmente tratava os dados na sua forma original e buscava prever as tendências futuras da mesma forma. Porém, foi observado que para previsões de mais longo prazo é necessário o uso de estratégias mais sofisticadas dado a volatilidade do mercado. A principal contribuição científica desse estudo foi mostrar o quão pequena é a porção de artigos (35,3%) que lida diretamente com a rentabilidade do método e apenas dois utilizam gerenciamento de riscos, apesar de diversos autores citarem a importância desses passos para a validação do modelo.

O trabalho de DUTT (2021) buscou fazer um estudo teórico da performance e aplicação de quatro modelos de aprendizado de máquina com dados de três diferentes fontes, sendo elas o preço do Bitcoin, o índice S&P500 e uma previsão de 10 anos

do tesouro dos Estados Unidos. O motivo da escolha dessas três fontes é a grande diferença do padrão de comportamento de seus gráficos, com o Bitcoin apresentando uma alta variabilidade, testando assim o quão bem os modelos conseguiriam prever algo com tamanha popularidade e volatilidade. A segunda classe de ativos é composta pelo S&P500, sendo um bom exemplo de como os modelos conseguem ser precisos lidando com dados relativamente estáveis e constantes, e por último a previsão do tesouro dos estados unidos serve de contraste a alta volatilidade do Bitcoin e a estabilidade do S&P500, sem uma tendência constante mas também evitando grandes variabilidades em curtos períodos.

Os quatro modelos de aprendizado de máquina são compostos pelo XGBoost, ARIMA, Facebook PROPHET e LSTM, sendo escolhidos por serem os mais conhecidos e utilizados em suas respectivas categorias. Começando pelo XGBoost, o qual é o que mais se assemelha a modelos de regressão clássicos utilizando gradient boosting (técnica de aprendizado de máquina que usa árvores de decisão), com sua maior dificuldade de uso sendo o grande número de hiperparâmetros configuráveis. O segundo é o modelo ARIMA (modelo auto-regressivo integrado de médias móveis), bem comumente usado por ser composto de outros modelos, os principais sendo o modelo auto-regressivo e o modelo de média móvel, que dão o nome ao mesmo. A combinação destes permite explicar o valor da variável futura a partir dos valores anteriores. O terceiro modelo é o LSTM, sendo o modelo mais avançado de RNN (Rede Neural Recorrente) por resolver o problema de perder dados importantes a longo prazo, para o qual é utilizado portões lógicos internos que regulam o fluxo de informações e decide o que é esquecido e o que precisa ser guardado a longo prazo. O quarto modelo é o Facebook Prophet, que consiste em um modelo temporal que acomoda sazonalidade em múltiplos períodos e permite ao analista fazer descobertas sobre as tendências. O modelo em si é feito de modo que o analista não precise saber o funcionamento do mesmo, com parâmetros intuitivos que podem ser ajustados por leigos em aprendizado de máquina sem problemas maiores.

Os resultados alcançados mostraram que o XGBoost teve uma performance baixa, não conseguindo capturar quaisquer padrões que incluíssem volatilidade, apenas mostrando uma performance razoável no S&P500 porém sem a precisão ideal. No caso do ARIMA, ele também não consegue capturar a volatilidade do Bitcoin, porém no caso dos títulos do tesouro ele mostrou uma capacidade excelente de predizer os valores, mostrando ser ideal para dados com baixa flutuação. O Facebook Prophet teve os melhores resultados de todos os modelos, o que segundo os autores é devido ao seu método de utilizar os

valores já previstos em conjunto aos parâmetros ajustados para reagir aos dados de treino e teste. Em conjunto aos dados recebidos para a descrição da sazonalidade, isso o tornou capaz de prever até o altamente volátil Bitcoin.

Por último, o LSTM obteve uma performance levemente inferior ao Prophet, esta diferença sendo vista melhor na análise do Bitcoin, onde apesar de conseguir seguir bem a tendência dos dados, se mostrou incapaz de prever a escala de sua volatilidade. A relevância deste trabalho é clara, dado que além de ter sido utilizada a tecnologia e o índice alvo, ele pôs essa aplicação em direta comparação com as outras tecnologias que já foram utilizadas na área (XGBoost e ARIMA), assim como os novos modelos considerados estado da arte (LSTM). Podemos observar que possivelmente por causa da multiplicidade de modelos e tecnologias utilizadas, o dataset foi utilizado apenas com os dados originais, ignorando indicadores e outras metodologias de Trading que poderiam melhorar ainda mais os resultados obtidos, porém, dado a natureza do trabalho essa escolha é compreensível e deixa espaço aberto para mais estudos.

A pesquisa realizada por Fang et al. (2019) (FANG et al., 2019) busca utilizar o Facebook Prophet em conjunto um modelo *Deep Learning* LSTM para prever a performance do índice MSCI (Taiwan Index Futures), composto pelas ações das 105 maiores empresas que atuam em Taiwan, o qual representa para esse país o mesmo que o S&P500 representa para os Estados Unidos (a principal diferença vem do fato das empresas no MSCI poderem estar situadas no exterior). Utilizando essa combinação de ferramentas, foi descoberto um padrão onde nas segundas-feiras o valor tendia a subir, assim como em geral na primeira metade do ano, enquanto que na segunda metade o preço tendia a cair. Além disso, a aplicação direta de LSTM foi capaz de prever as tendências de curto prazo com uma considerável acurácia, tendo um resultado satisfatório para os pesquisadores. A relevância desta pesquisa se apresenta pela tecnologia em comum que está sendo proposta e sobre as similaridades entre o S&P500 com o MSCI, fazendo um paralelo direto com o presente trabalho sem diminuir sua relevância acadêmica.

O artigo de Garlapati et al. (2021) propõe a utilização do algoritmo de previsão ARIMA em conjunto ao Facebook Prophet para previsão do valor de ações não especificadas no período de 2012 à 2020. Neste é explorado os fatores que podem afetar positivamente ou negativamente os valores das ações, além de análise/tratamento de dados, validação da correlação dos valores trabalhados, e a análise de médias móveis para a suavização dos dados e para a subsequente confirmação da aplicabilidade do modelo ARIMA. Os resultados do modelo ARIMA mostram uma confiabilidade superior

aos mínimos 95% requeridos para uma previsão de sucesso, e em seguida a inserção dos dados no Facebook Prophet também alcançam uma performance satisfatória com os testes executados, terminando com a validação dos resultados alcançados através do cálculo do parâmetro MAPE (Erro percentual absoluto médio) confirmando a baixa taxa de erro apresentada. É relevante especificar que os 95% de confiabilidade não significa que o modelo vai ser capaz de prever por volta disso em dados futuros, provavelmente muito menos dependendo da fonte dos dados que estão sendo trabalhados, e sim que o modelo conseguiu aprender padrões nos dados passados e se ajustar com 95% de precisão. Deste trabalho pode-se observar a falta de especificação da base de dados utilizada, o que impossibilita uma replicação exata dos testes realizados, porém os tópicos abordados abrangem vários dos temas que o presente trabalho propõe-se a executar tendo assim uma correlação relevante.

O artigo de Saldivar and Ortiz (2019) buscou comparar a performance de diferentes modelos de aprendizado de máquina na predição da cotação das ações do *SPDR S&P500 Exchange Traded Fund* (ETF), o qual consiste em um fundo de investimento negociado que busca seguir o S&P500. Os modelos comparados são uma simples regressão linear, o Facebook Prophet e uma LSTM, esses sendo escolhidos por representarem desde o mais simples modelo preditivo até o mais robusto modelo de predição, considerado atualmente o estado da arte. Os resultados alcançados mostraram como esperado que a regressão linear não é capaz de prever a curva de preço de forma alguma, apenas servindo de base para comparação do avanço representado pelas tecnologias seguintes. O Prophet mostrou um resultado superior à regressão linear, porém não foi capaz de realmente seguir a curva, apesar de a predição cruzar o preço real duas vezes. Por último, o LSTM mostrou os melhores resultados entre os experimentos, apesar da variância dos resultados obtidos a partir dos parâmetros. Algo que é dada importância no artigo é a ocasional melhora de performance ao aplicar LSTM nos dados organizados em reverso comparado ao normal, a explicação da qual não foi encontrada pelos autores em nenhum lugar da literatura.

Um dos problemas observados pelo próprio autor foi a escolha de utilizar o preço de fechamento dos últimos vinte anos, o que possivelmente causou os dados mais antigos a serem pouco relevantes comparados a dados mais novos, sugerindo assim que utilizar os dados dos preços a cada duas ou três horas em um período menor pode vir a dar melhores resultados. A relevância do trabalho é observada pela similaridade dos dados alvo, consistindo de um fundo de investimento negociado que busca seguir o

S&P500, sendo assim uma possível abordagem para utilizar quaisquer previsões feitas sobre o índice S&P500 em um ativo tangível. Além disso, as tecnologias comparadas são relevantes dado a consistirem na tecnologia alvo do presente trabalho (Facebook Prophet), um modelo considerado antiquado à complexidade do problema proposto mas que serve de base de comparação aos outros e um modelo atualmente considerado estado da arte.

No trabalho de Mohan et al. (2019), os autores propõe a utilização de notícias relacionadas às companhias para melhorar a acurácia das predições de ações de algumas companhias que compõe o S&P500, essas sendo mais especificamente a Apple, Amazon, American Airlines, Facebook e Microsoft. Foram comparados seis tipos diferentes de predições, essas sendo com o modelo ARIMA, com o Facebook Prophet, com uma LSTM usando como entrada somente preços, com LSTM e utilizando também polaridade textual (modelo RNN que extrai somente se o texto é positivo ou negativo), com LSTM e com o texto inteiro como entrada (basicamente aplicando o modelo de compreensão de texto e suas saídas como parte do modelo geral) e por último, uma LSTM multivariada (onde todos os dados de todas as empresas são inseridos em conjunto para capturar quaisquer co-relações entre as notícias de empresas afetarem umas as outras). Os resultados alcançados mostraram resultados baixos com o modelo ARIMA e o Facebook Prophet, já o modelo LSTM com o texto completo de entrada teve uma performance boa em ativos estáveis mas uma queda considerável nos ativos de baixo valor e voláteis. O modelo que utiliza polaridade textual teve a melhor performance de todos, e por último o modelo de LSTM multivariada teve uma performance baixa, indicando que os dados de cada empresa são mais importantes que os dados gerais. A importância desse trabalho se deu pela mudança de foco do mesmo, saindo de trabalhar somente com os dados passados dos preços das ações e buscando outras fontes de dados que podem influenciar os valores. As ações escolhidas tem certa similaridades com aquelas do presente trabalho assim como a utilização do Prophet, porém dado a natureza do mesmo ser muito pouco descrito quanto a sua utilização, o que torna os resultados difíceis de replicar e validar efetivamente.

No artigo de Kumar et al. (2021) busca-se comparar os resultados do valor previsto da ADANIPORT, usando o período de dezembro de 2007 a dezembro de 2020, para treinamento de seis diferentes modelos: Média móvel, Regressão linear, Modelo KNN (K-Nearest Neighbours), Auto ARIMA, Facebook Prophet e LSTM. Para a validação foram selecionadas três métricas distintas, essas sendo o RMS (raiz do erro quadrático médio), o R2 (R ao quadrado) e o MAE (média do erro absoluto).

Os resultados obtidos indicam que a performance mais baixa sendo a da Regressão

linear, seguido do KNN e Auto ARIMA respectivamente. O LSTM teve os melhores resultados com no mínimo o dobro de precisão comparado ao segundo melhor resultado, este sendo o modelo de Média móvel, e em terceiro ficou o Facebook Prophet.

A relevância do artigo consiste principalmente na comparação de alguns modelos de previsão pouco vistos na literatura da área, apesar de, por exemplo, a média móvel ser um dos indicadores mais populares de trading e no trabalho apresentar um resultado bom considerando a simplicidade do modelo.

No artigo de Demiray and Gürhanlı (2021) buscou-se comparar a performance de cinco modelos de aprendizagem de máquina utilizando o preço de fechamento das ações da Microsoft de 2014 até 2019 para treinamento, e para cada modelo foi feita a previsão para cinco, dez e vinte dias.

Os modelos utilizados no trabalho foram a Regressão Polinomial, ARIMA, XGBooster, LSTM, e o Facebook Prophet, os quais foram avaliados utilizando o RMS (raiz do erro quadrático médio), o MSE (média do erro quadrático) e o MAE (média do erro absoluto).

Os resultados obtidos indicaram uma melhor performance da regressão polinomial, seguido do ARIMA e XGBooster respectivamente, com o Prophet e LSTM apresentando as piores performances. Como esperado, os dados apresentam erros maiores conforme se preveem mais dias, com vinte dias apresentando mais erros que dez dias, por exemplo.

A relevância desse trabalho vem de, apesar de utilizar as tecnologias e metodologias similares a outros diversos trabalhos, os resultados encontrados vão contra a grande maioria dos mesmos, com as técnicas estatísticas e tecnologias mais antigas e pouco usadas apresentando melhores resultados que as mais recentes e complexas.

Observa-se também uma certa simplicidade na apresentação dos resultados, os quais apesar de mostrar a tabela com os valores das métricas realizadas, não foi bem explorado quais os métodos e parâmetros que foram utilizados em cada uma das técnicas. Problemas assim são especialmente complexos para o modelos como o Prophet e LSTM onde hiperparâmetros são extremamente importantes em definir se o modelo vai conseguir efetivamente prever os dados futuros.

No artigo de Kulshreshtha and A (2020) é proposto um modelo híbrido de ARIMA com LSTM, comparando-o com o Facebook Prophet na previsão do valor do índice S&P 500 e os resultados são mensurados usando RMSE, MSE, R-quadrado e MAPE.

Em teoria a ideia é que o ARIMA é ideal para se capturar quaisquer tendências

lineares, porém é incapaz de lidar com não-linearidade, enquanto que o LSTM é o melhor método atual para lidar com não linearidade. Com isso em mente, o objetivo do trabalho é criar e testar um modelo onde uma previsão é feita pelo ARIMA, e os resíduos que não puderam ser computados pelo mesmo são alimentados a uma LSTM, o resultado dos dois são combinados em uma previsão final.

Os resultados da comparação mostram que o modelo proposto foi capaz de alcançar resultados muito superiores ao Prophet, com 99% de previsão no coeficiente R-quadrado comparado ao 74% do Prophet.

A relevância do trabalho está na aplicação do Prophet junto a outra tecnologia na previsão do SP 500, apesar de que neste trabalho talvez por o tema do mesmo ser o desenvolvimento do outro modelo, a aplicação do Prophet foi muito pouco explorada, sendo apenas mostrado seu resultado final após os testes, sem as etapas de preparação e seleção de parâmetros que o modelo criado passou.

3.1 Conclusão

No geral o tema de comparação de múltiplos modelos de inteligência artificial para a previsão de diferentes tipos de ações e ativos já foi bastante explorado como observamos nesse capítulo, porém como já foi citado anteriormente, há uma visível falta de abordagens utilizando conhecimentos de análise técnica de *trading* e analisando a aplicabilidade real desses modelos desenvolvidos no mercado atual, abrindo um espaço ainda não explorado para este trabalho levantar questões e possibilidades de aplicação com sua utilização mais profunda das tecnologias escolhidas em conjunto de um conhecimento da área financeira em busca de resultados mais consistentes e aplicáveis no mercado real.

Na tabela 2 podemos visualizar um resumo dos trabalhos discutidos nesse capítulo e suas principais características:

Tabela 2 – Trabalhos analisados

Autor	Principais Características
A. W. Li e G. S. Bastos	Pesquisa em múltiplos artigos relacionados a inteligência artificial e mercado financeiro mapear o panorama atual de pesquisa na área.
Anav Dutt	Buscou comparar quatro modelos diferentes em três fontes de dados distintas, os modelos sendo o XGBoost, ARIMA, Facebook PROPHET e LSTM, enquanto que os dados são os preços do Bitcoin, S&P500 e tesouro dos EUA.
Fang et al.	Buscou utilizar o Facebook Prophet em conjunto um modelo Deep Learning LSTM para prever a performance do índice MSCI (Taiwan Index Futures).
Garlapati et al.	Faz uso do ARIMA em conjunto ao Facebook Prophet para previsão do valor de ações não especificada.
Frank Saldivar e Mauricio Ortiz	Buscou comparar a performance de uma simples regressão linear, o Facebook Prophet e uma LSTM na predição da cotação das ações do SPDR S&P500 Exchange Traded Fund (ETF), o qual consiste em um fundo de investimento negociado que busca seguir o S&P500.
Mohan et al.	Os autores propõe a utilização de notícias relacionadas às companhias para melhorar a acurácia das predições de ações de algumas companhias que compõe o S&P500, essas sendo mais especificamente a Apple, Amazon, American Airlines, Facebook e Microsoft.
Kumar et al.	Buscou comparar os resultados do valor previsto da ADANI PORT, utilizando seis diferentes modelos: Média móvel, Regressão linear, Modelo KNN (K-Nearest Neighbours), Auto ARIMA, Facebook Prophet e LSTM.
Salim Demiray e Ahmet Gürhanlı	Buscou comparar a performance de modelos baseados em Regressão Polinomial, ARIMA, XGBooster, LSTM, e o Facebook Prophet utilizando o preço de fechamento das ações da Microsoft, e para cada modelo foi feita a previsão para cinco, dez e vinte dias.
S. Kulshreshtha e Vijayalakshmi A	Propõe um modelo híbrido de ARIMA com LSTM, comparando-o com o Facebook Prophet na previsão do valor do índice S&P 500 e os resultados são mensurados usando RMSE, MSE, R-quadrado e MAPE.

Fonte: O Autor

4 DESENVOLVIMENTO

Neste trabalho, foi desenvolvido um sistema de inteligência artificial com a capacidade de prever o valor do índice S&P500 com a maior acurácia possível. O sistema proposto é baseado no padrão observado em ferramentas e sistemas de *trading*, onde uma importância muito mais elevada é dada aos dados mais recentes comparados aos mais antigos. Essa importância, teoricamente, torna sistemas incapazes de lidar com dados novos de maneira ágil e otimizada menos úteis para utilizações a longo prazo, um problema que é facilmente observado em modelos de redes neurais, incapazes de acompanhar essa constante entrada de dados novos com eficiência.

Com esse problema em mente, buscou-se tecnologias que fossem capaz de fazer previsões com baixo tempo de treinamento sem perder muito da performance dos modelos mais complexos e robustos. Para isso, bem cedo na elaboração do projeto foi selecionado o Facebook Prophet, o qual de maneira rápida e flexível pode fazer a leitura de conjuntos completamente novos de dados e produzir uma previsão em poucos segundos.

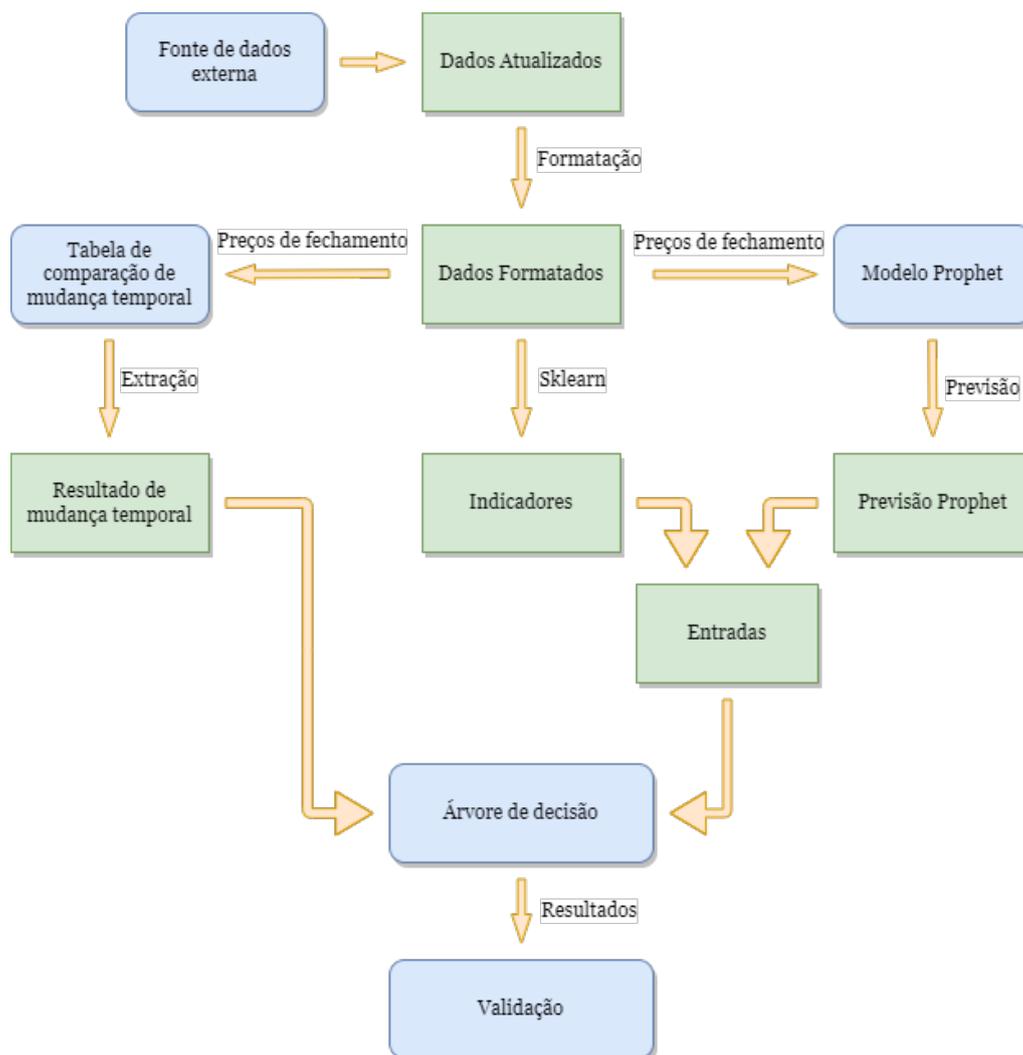
Para reforçar a previsão com conhecimentos de trading raramente utilizados neste tipo de software, foram usados como entrada secundária alguns dos principais indicadores disponíveis, cuja contribuição aos resultados foi avaliada em testes preliminares e eliminados conforme necessário, buscando-se tornar o processo o mais simples e computacionalmente viável sem perdas de performance.

4.1 Fluxo de funcionamento

Na Figura 7 pode-se observar o fluxo de funcionamento do projeto, onde cada passo do funcionamento e interações podem ser descritas como:

1. Fontes de dados externa: Para utilizar o software, o usuário deve providenciar uma fonte de dados histórica de preços para ser utilizado pelo programa, isto pode facilmente ser adquirido de um dos muitos sites disponíveis pela internet.
2. Dados formatados: Os dados inseridos no programa são preparados para a utilização dentro do programa, essa preparação inclui etapas como remover dias sem transações, converter o formato para um utilizável pelas bibliotecas e renomear as colunas de acordo com o padrão utilizado pelo programa.
3. Modelo Prophet: Os preços de fechamento e as datas dos mesmos são inseridos

Figura 7 – Fluxo de funcionamento do projeto



Fonte: Autor

no software do Facebook Prophet e uma previsão é gerada para um número determinado de dias (o usuário determina a quantidade de dias através de parâmetros)

4. Indicadores: Os indicadores são calculados utilizando a biblioteca chamada Sklearn, a qual disponibiliza diversas ferramentas que tornam o processo simples e rápido.
5. Tabela de comparação de mudança temporal: Consiste na criação de uma tabela com o formato desejado de saída para o programa, onde para cada uma das velas (oriundo de um gráfico de velas) uma leitura é feita sobre seu futuro que pode ser subida, descida ou mudança insuficiente. Assim como o período a ser previsto pelo programa, a escala de mudança a ser considerada significativa é determinada por uma entrada do usuário através de um parâmetro.
6. Entradas: A previsão do Prophet e os indicadores são reunidos em uma tabela única para atuar como entrada para a árvore de decisão, porém no caso dos dados do Prophet, os dados inseridos consistem nas previsões a frente da data que os indicadores foram extraídos, sendo assim um conjunto de indicadores e a média da tendência ao invés do valor de fechamento do dia.
7. Árvore de decisão: Na árvore de decisão são inseridos em uma tabela os indicadores e a previsão do Prophet, e em outra tabela as leituras de subidas e descidas dos preços já conhecidos. A partir desses dados a árvore vai extrair quaisquer padrões presentes e aprender a prever crescimentos e quedas utilizando dados novos.
8. Validação: O modelo gerado pela árvore foi testado em dados novos nunca antes vistos pela mesma e coeficientes de acurácia serão determinados para validar a qualidade de leitura da mesma.

4.2 Ferramentas e Softwares

A linguagem de programação Python foi selecionada por prover uma larga gama de ferramentas e pacotes que tornam o desenvolvimento mais simples, ágil e especialmente compreensível, sendo assim possível compreender seu funcionamento com o mínimo de conhecimento da mesma.

Inicialmente, foi utilizado o Jupyter notebook como ambiente de programação devido às suas várias funcionalidades que ajudam em projetos de *Data Science*. No

entanto, durante o desenvolvimento, foi feita uma mudança para o Visual Studio Code para aproveitar o suporte a plugins que permitem a visualização e controle de vários tipos de arquivos no mesmo programa, o que melhorou o desenvolvimento devido a requisitos não planejados durante a estimativa inicial, como o download manual de dados para obter a melhor qualidade e os mais recentes dados disponíveis para inserir no programa. Outras funcionalidades fornecidas pela IDE incluem o controle de versão Git, que será usado no projeto para evitar perdas locais e fornecer versionamento do código, além do suporte para intellisense (verificação de erros de tipo e escrita do código), instalação e reparo de bibliotecas/dependências dentro da IDE, entre outros.

Além disso, um conjunto de indicadores baseados nos dados de entrada será avaliado para complementar esses dados de entrada do modelo. Os indicadores selecionados inicialmente foram Média Móvel Simples e Exponencial, RSI, MACD, Histograma MACD e Estocástico. A seleção dos indicadores foi feita com base na popularidade de uso e na disponibilidade de material relacionado a cada um. No entanto, o uso deles foi eventualmente restrito à Média Móvel Simples e RSI, os quais apresentaram os melhores resultados quando usados em conjunto com o Prophet.

Para o desenvolvimento mais fluído do projeto, foram utilizadas diversas bibliotecas disponíveis para Python:

1. Pandas: Biblioteca open-source composta por várias ferramentas fundamentais de análise de dados e *Data Science*, além disso tem extensões como o "pandas-ta" que permitem a extração de indicadores diretamente de dados com funções simples.
2. Nbformat: Pacote unificado que contém toda a base para trabalhar com notebooks Jupyter mesmo em outras IDEs, além de APIs para lidar com tais notebooks.
3. Prophet: Biblioteca para utilizar a ferramenta do Facebook Prophet, a qual disponibiliza a ferramenta de previsão de séries temporais com especialidade em dados com tendências sazonais e temporadas, além de apresentar robustez quanto a dados ausentes e mudanças em tendências.
4. Plotly: Biblioteca para criação de gráficos interativos e de alta qualidade.
5. Sklearn: Nome completo "Scikit-learn", consiste numa biblioteca composta de ferramentas de *Machine Learning* pré prontas para uso com simples configurações iniciais e bons resultados. Requer as bibliotecas Numpy e Matplotlib para funcionar.
6. Datetime: Biblioteca interna do Python para conversões, mensuração e alterações

de datas e horários.

7. Matplotlib: Biblioteca básica para criação de gráficos e visualizações em Python, sendo o modelo preferido por outras bibliotecas por sua leveza e performance com poucas configurações.
8. Numpy: Biblioteca para configuração e manipulação de grandes quantidades de dados em formato de tabela e matrizes, utilizado extensamente por outras bibliotecas e uma ferramenta fundamental para desenvolvimento de projetos de *Data Science*.

4.3 Configuração de parâmetros

Para cada execução do programa, vários parâmetros podem ser escolhidos para determinar a configuração da previsão, expandindo assim a utilidade do programa para possivelmente outros tipos de dados ou para se adaptar melhor a épocas diferentes com padrões diferentes. Isso aumenta a flexibilidade do programa e permite ao usuário otimizar as configurações para atender às suas necessidades específicas.

Os parâmetros disponíveis são:

- **TIMEFRAME**: Especifica o período que cada vela dos gráficos representa, não afeta todos os gráficos do programa, mas somente alguns específicos. Ex: '1d' = Um dia
- **WARMUP**: Comunica ao programa o período de aquecimento dos dados, isto é, o período que os dados serão utilizados para extrair até o mais longo indicador. Ex: Caso o maior indicador utilize dados de até 60 dias atrás, deve-se configurar esse parâmetro para 60.
- **DAYS_AHEAD**: Seleciona o período que o Prophet e a árvore devem tentar prever a frente. Ex: Para prever o dia seguinte o parâmetro é configurado como "1". Para prever a semana seguinte o parâmetro é configurado "7".
- **PERCENTAGE_TEST_DATA**: Seleciona a porcentagem de dados que o Facebook Prophet deve separar para testes.
- **PERCENTAGETOINCREASE**: Seleciona o limite de significância para este treinamento, seu funcionamento será explicado em detalhes nas seções subsequentes.
- **PERCENTAGE_TEST_TREE**: Seleciona a porcentagem de dados que a árvore de

decisão deve separar para testes.

- `QUANTIDADE_TESTES`: Seleciona quantos testes serão executados por vez, em busca do treinamento que alcançou o melhor resultado.

Há mais três parâmetros a serem configurados, porém como sua relevância é somente para questões de validação sua explicação será dada posteriormente no capítulo.

4.4 Aquisição e preparação dos dados

Na proposta inicial do projeto, dados seriam buscados em tempo real de uma das diversas ferramentas de trading, dentre as quais, as duas mais promissoras para o projeto consistiam na API da própria NASQUAD, que oferecia a vantagem de ter a melhor fidelidade dos dados já que seriam adquiridos diretamente de uma fonte confiável e mais robusta, e a segunda fonte consistia no servidor do Yahoo Finance, o qual é amplamente utilizado em diversos projetos abertos relativos ao mercado financeiro, muitos desses projetos de natureza similar a esse, e por esse motivo também foi um dos principais candidatos ao projeto.

Ambos acabaram sendo descartados por razões de que a API da NASQUAD requisitava além de uma chave específica de conta que poderia ser criada gratuitamente na mesma, uma outra chave referente á fonte de dados que se estaria buscando, no caso do S&P500 os dados somente estavam disponíveis através de uma empresa terceira cujos serviços eram pagos, e por isso foi descartado para este projeto, não sendo viável esse controle de chaves para possíveis usuários futuros.

Já o Yahoo Finance passou por diversos testes para confirmar a qualidade dos dados obtidos e se cumpria os requisitos do projeto, porém logo a inviabilidade do mesmo neste projeto em particular foi determinada por utilizar dados antigos e com pouca precisão quando comparados a fontes estáticas, sendo descartado por não cumprir o requisito de qualidade e contemporaneidade dos dados.

Enfim, para a adquirir os dados optou-se por fazer a aquisição manual dos mesmos em tabelas CSV (*Comma-separated values*), os quais além de serem o padrão de download de grande parte dos sites de dados financeiros, é um modelo de pequeno tamanho, facilmente lido por qualquer programa de texto ou tabela e pode ser lido e convertido rapidamente pelo Pandas dentro do código.

A preparação dos dados inicia com a remoção dos dias onde o volume de

transações é zero, que consiste nos fins de semana onde não há mudanças significativas nos dados. Em seguida separa a coluna de datas e cria um índice independente (no caso o índice identifica numericamente os dados em ordem de 0,1,2,3,4,etc.), já que as datas serão inseridas no modelo como uma entrada temporal.

Em seguida são criadas três colunas para armazenar os dados dos indicadores selecionados, no caso do projeto final estes foram:

1. SMA50 - Para armazenar a média móvel simples dos últimos cinquenta dias.
2. SMA200 - Para armazenar a média móvel simples dos últimos duzentos dias.
3. RSI - Para armazenar o índice de força relativa dos últimos quatorze dias.

As colunas criadas são preenchidas através de funções disponibilizadas pela biblioteca do Pandas_ta, que consiste em uma extensão do Pandas para o cálculo de diversos indicadores.

Por fim, remove-se as entradas mais antigas de acordo com o indicador de mais alcance, no caso deste projeto, o indicador seria a média móvel simples de duzentos dias, descartando esses valores para que todas as entradas tenham igualmente todas as colunas preenchidas. Depois disso ajusta-se os índices para refletir o novo começo da tabela e com isso, os dados estão prontos para serem utilizados.

4.5 Predição do modelo Prophet

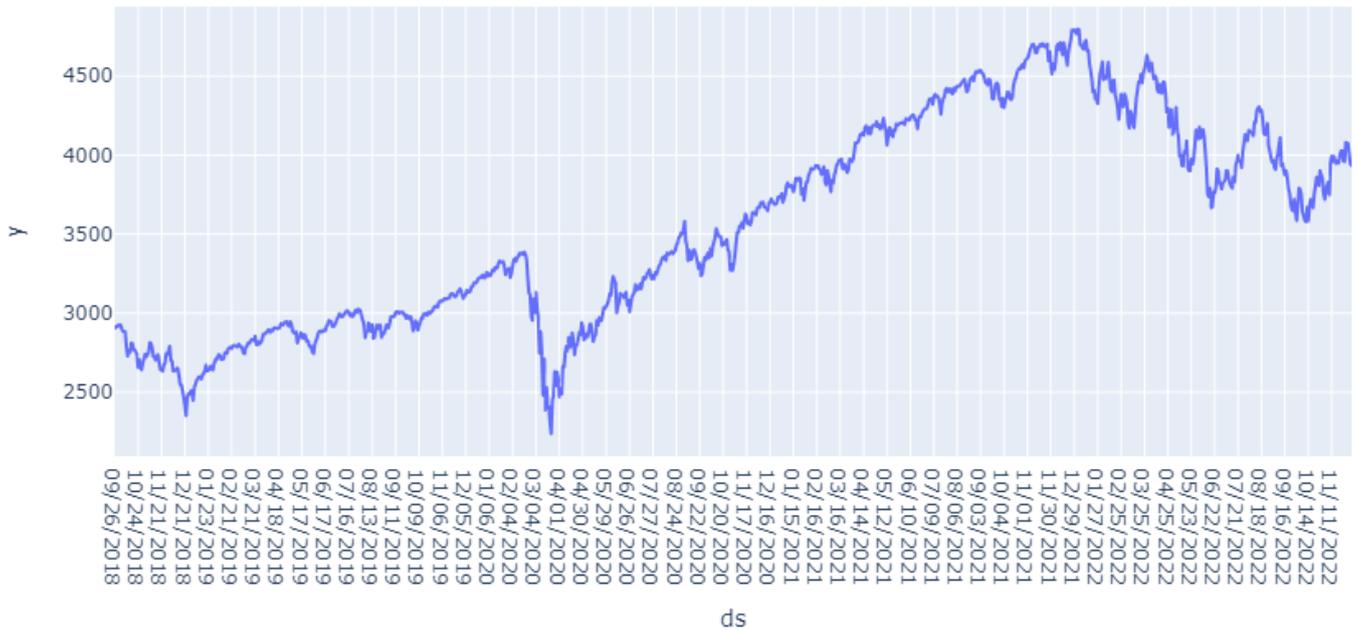
Com os dados devidamente preparados, inicia-se os procedimentos para adquirir a predição do Prophet. Inicialmente, cria-se uma tabela separada com apenas as colunas: "ds" e "y", sendo assim formatadas pelas especificações do Prophet que requer esse formato. Em seguida, os dados inseridos nesta tabela são respectivamente as datas e os valores de fechamento de cada dia.

Uma representação gráfica pode ser visualizada na Figura 8:

A tabela de entrada do Prophet é então separada em duas sub tabelas, uma para treinamento e uma para testes, as quais a porcentagem de cada é dada por um hiper parâmetro que é inserido de acordo com a porcentagem dos dados que o usuário quer utilizar para cada.

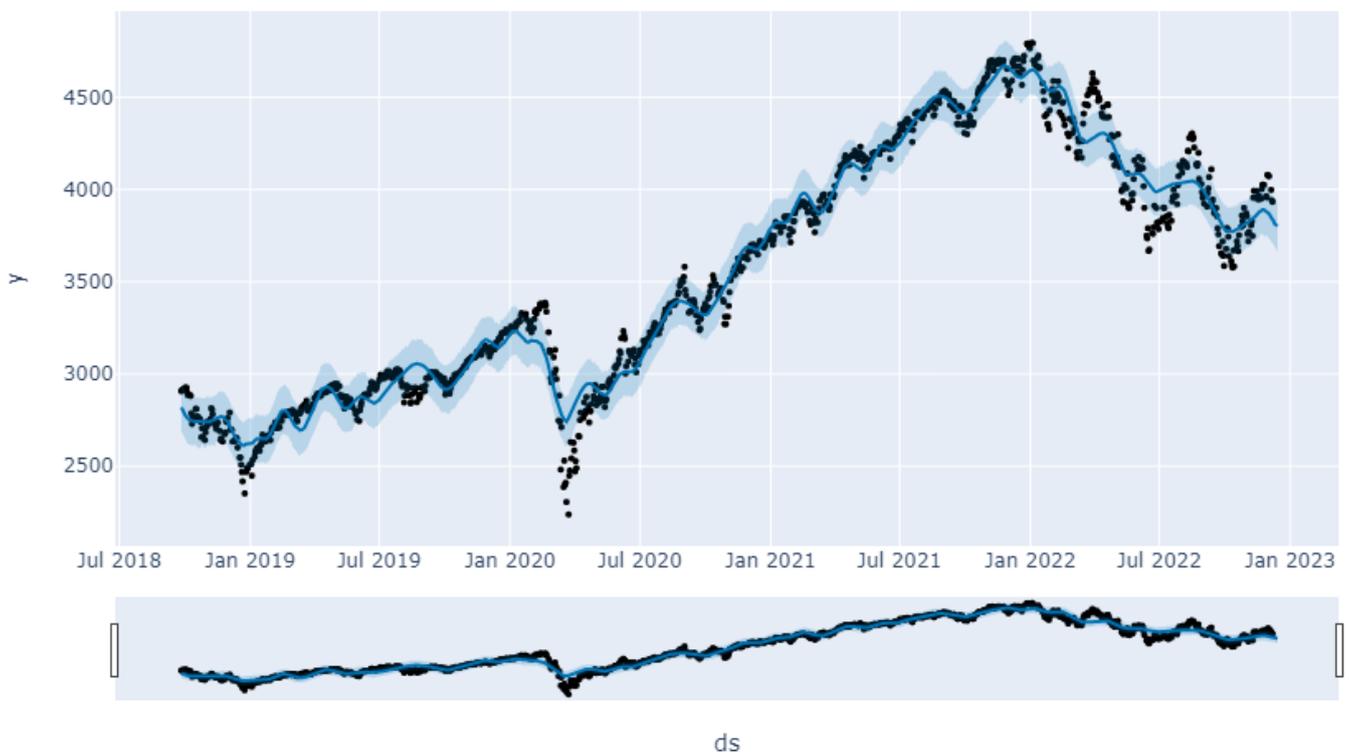
Após isso inserimos os dados de treinamento no Prophet e através de outro hiper parâmetro selecionamos quantos dias a frente a previsão deve tentar prever. O resultado deste procedimento pode ser observado na figura 9 gerada pelo Plotly:

Figura 8 – Dados a serem inseridos no modelo Prophet



Fonte: Autor / Plotly

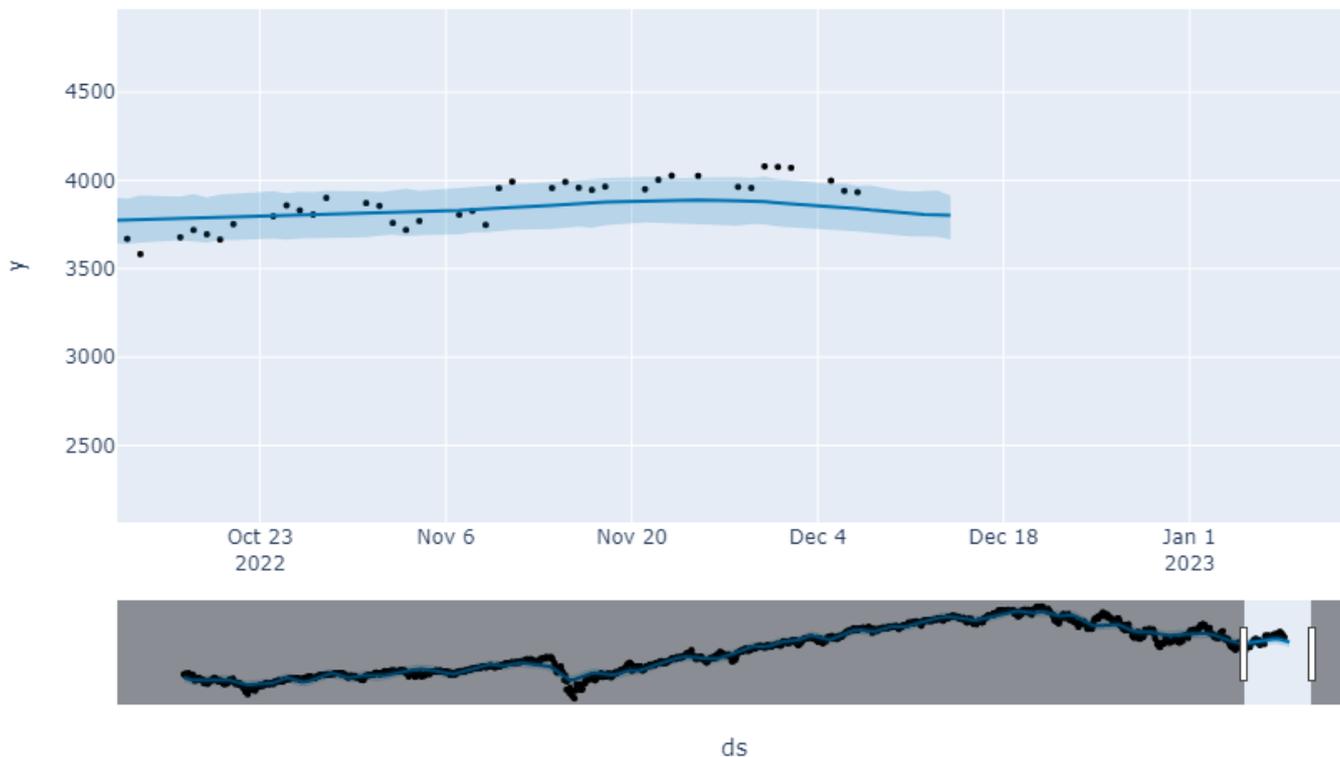
Figura 9 – Aproximação da curva e previsão do Prophet



Fonte: Autor / Plotly

Na Figura podemos observar nos pontos, as marcações do valor de fechamento do

Figura 10 – Demonstração do ponto de previsão para uma semana



Fonte: Autor / Plotly

dia, e na linha azul, a média prevista pelo Prophet, com a área azul mais clara envolta da mesma sendo a máxima e a mínima prevista em cada ponto da previsão.

Fica claro que durante o treinamento, o Prophet sozinho não é capaz de acompanhar com precisão certas mudanças bruscas nos preços, tendendo a contra-balancear a tendência antes do gráfico apresentar a mudança prevista. Isso significa que, embora o Prophet cumpra a tarefa de ferramenta preditiva, ele é menos eficaz em previsões a curto prazo. Vale ressaltar que neste gráfico, o período a ser previsto pelo Prophet é de sete dias seguintes, então em um gráfico de cerca de cinco anos, a previsão parece seguir os pontos. No entanto, quando visto de perto, como mostrado na imagem 10, a previsão se estende além do período especificado no parâmetro.

4.6 Preparação dos dados para árvore de decisão

Com a leitura do Prophet devidamente preparada em outra tabela, é criada uma nova pra formatar os dados que serão inseridos na árvore de decisão sem afetar as anteriores. Primeiramente busca-se na tabela inicial as colunas com os valores de datas

e indicadores com os mesmos nomes das colunas. Evita-se de utilizar a primeira tabela do projeto para não perder os dados originais ou precisar apagar as colunas que não são relevantes nessa etapa do processo, dando-se preferência pela criação de uma nova e a duplicação dessas colunas em específico.

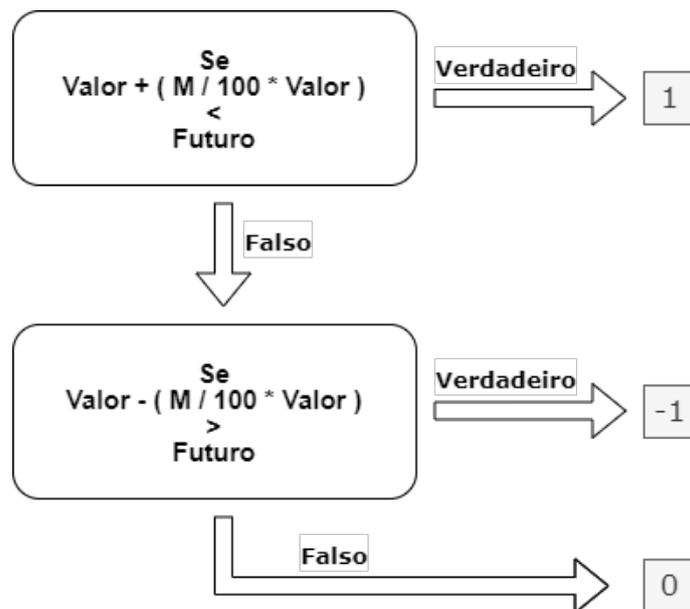
Em seguida, são removidos os dados finais da tabela de acordo com o hiper parâmetro de previsão, este passo sendo necessário pois para estes dias não existem as informações exatas do futuro para o treinamento, no caso, a informação que desejamos seria se o valor da ação subiu ou desceu.

Com a tabela formatada, é inserido então os dados de previsão gerados pelo Prophet com os primeiros valores excluídos, isto se dá para relacionar as previsões dos dias a frente extraídas pelo Prophet com a leitura do dia adquirida pelos indicadores, sendo essa combinação de dados que quando inseridos na árvore de decisão resultará em uma leitura de subida, descida ou sem mudança.

Antes de concluir a preparação da tabela de inserção, uma última alteração importante é a conversão da data presente na tabela que está como texto, para um formato reconhecido pelo Sklearn, para isso utiliza-se a biblioteca Datetime que oferece tal suporte.

Concluídos esses passos a entrada da árvore está preparada, e agora para a saída esperada utilizaremos outro hiper parâmetro para determinar um índice de mudança relevante que preencherá a tabela de saída. Esse parâmetro é inserido como uma porcentagem de mudança mínima necessária para o software identificar como subida, descida ou sem mudança o valor de fechamento no período da previsão. O funcionamento do mesmo pode ser entendido através da figura 11:

Figura 11 – Funcionamento da tabela de comparação



Fonte: Autor

Onde:

- Parâmetro de dias a serem previstos - N
- Parâmetro de mudança mínima necessária - M
- Valor de fechamento do dia - Valor
- Valor de fechamento N dias depois - Futuro

Para cada uma das entradas, é criada uma saída equivalente que determina se a partir daquele dia haverá uma subida ou descida no período determinado, e é exatamente esse padrão que precisa ser capturado pela árvore de decisão.

4.7 Montagem da árvore de decisão

O primeiro passo para montar a árvore de decisão consistiu em converter ambas as tabelas previamente preparadas para o formato Numpy (mais precisamente um ndarray, abreviação para array n-dimensional), sendo este o único formato aceito pelo Sklearn. Em seguida, todos esses dados são separados em dados para treinamento e teste utilizando o mesmo hiper parâmetro utilizado anteriormente.

Com os dados já devidamente preparados basta inserir as entradas e saídas na árvore de decisão e avaliar os resultados obtidos, os resultados da qual serão avaliados

em testes de acurácia, matriz de confusão e poderemos tirar certas conclusões sobre os resultados e onde ainda há espaço para aprimoramentos.

4.8 Simulação de *trading*

A fim de reforçar a validade do modelo como uma ferramenta eficaz para o auxílio na negociação de ações, foi desenvolvida uma função adicional, consistindo em um simulador de transações em um período específico, utilizando o modelo treinado para identificar os momentos oportunos para compra e venda.

A validação é crucial devido ao uso prático das previsões de trading e à importância única de cada previsão correta e incorreta. Isso significa que, em teoria, um software com uma porcentagem de acertos mais alta não necessariamente é uma ferramenta de trading superior.

Sua operação é relativamente simples, consistindo na aplicação de técnicas de Swing trade para aproveitar ao máximo cada oportunidade encontrada. Para sua utilização, são empregados três parâmetros específicos que configuram sua operação e podem ser ajustados para imitar o modelo de investimento do usuário. Estes são:

- **DAYS_BETWEEN_TRADES** - Especifica o período de tempo a ser aguardado após cada transação antes de iniciar a busca por novas oportunidades de compra/venda. Essa configuração ajuda a adaptar o uso do algoritmo para imitar o comportamento humano, uma vez que muitos *traders* (especialmente os adeptos de *Swing Trade*) costumam evitar realizar múltiplas transações consecutivas e preferem esperar a oportunidade ideal e evitar as armadilhas mais imediatas do mercado.
- **INITIAL_MONEY** - Especifica o montante inicial a ser simulado pelo usuário. Essa configuração permite obter uma noção mais precisa dos lucros que o usuário poderia obter com o capital de que dispõe atualmente ou que dispunha no período que a simulação é executada.
- **PAST_PERIOD_TO_TRADE** - Especifica o período histórico a ser utilizado para simular as transações, sendo configurado como os N dias anteriores em relação ao final da tabela de dados. Exemplo: Se o parâmetro for estabelecido em 120, os dados utilizados serão as últimas 120 entradas da tabela para realizar as transações. Essa configuração é utilizada para limitar o período de operações simuladas, pois

para muitos *traders* não é relevante ponderar os possíveis lucros obtidos ao começar a investir anos atrás, quando não havia conhecimento ou capital de investimento. Ainda assim, essa configuração adiciona mais uma possibilidade de personalização para a simulação.

Uma vez estabelecidos os parâmetros, a simulação não adota nenhuma estratégia específica de gestão de riscos, baseando-se somente nos resultados obtidos a partir do modelo e investindo todo o capital disponível em cada oportunidade identificada como *Bullish*, e recuperando todo o capital juntamente com quaisquer lucros obtidos quando é reconhecido um momento *Bearish*.

Para aplicar com o dinheiro real do usuário, é necessário um sistema de proteção de capital para lidar com eventos imprevisíveis que podem afetar o mercado, como a pandemia COVID-19 e a guerra na Ucrânia, que tiveram impacto significativo no mercado de ações. No entanto, para este protótipo acadêmico, a escolha foi optar por uma simulação que busca obter o máximo de lucro possível de cada previsão, em vez de se preocupar com eventos imprevisíveis.

Efetivamente, o funcionamento do simulador inicialmente verifica se o usuário está em um período de tempo específico, estabelecido pelo parâmetro, em que não são realizadas transações. Se as transações estiverem disponíveis, ela coleta os valores atuais de capital e ações do usuário, identificando qual dos dois ele possui no momento e, com base nessa informação, utiliza a previsão para determinar se uma transação de compra ou venda deve ser realizada.

A combinação necessária para uma transação é intuitiva, mas para esclarecer, em situações em que o usuário possui capital e não possui ações, ele buscará a próxima previsão de crescimento (no código do programa, um "1") e realizará a compra de ações utilizando todo o capital disponível. Por outro lado, quando o usuário possui ações e não possui capital, a simulação buscará a próxima previsão de queda nos preços e venderá todas as ações. Ao final da execução, caso o usuário possua ações e não possua dinheiro, a simulação venderá todas as ações acumuladas e apresentará o montante final.

No próximo capítulo, analisaremos os resultados obtidos com o modelo e as validações realizadas para avaliar se o programa atingiu o desempenho esperado.

5 RESULTADOS

Nesta secção serão apresentados os resultados adquiridos da execução do programa e as conclusões que puderam ser tiradas.

Esses resultados serão divididos em diversas abordagens utilizadas para mensurar a eficácia da solução no problema proposto, divisões essas consistindo em: Comparação entre o modelo proposto com a utilização de somente o Facebook Prophet e de somente uma Árvore de decisões, validações e análises internas dos resultados, exemplo de retornos financeiros e implicações possíveis dos resultados.

5.1 Comparação do modelo

Para comprovar a utilidade do modelo como uma ferramenta superior às alternativas disponíveis e com seu espaço de utilização justificado, foram executadas diversas execuções do programa com as mesmas configurações porém diferentes componentes de predição para definir quais as vantagens, desvantagens e o mais importante, a eficácia geral de cada um dos modelos.

Para isso foram utilizados além do modelo descrito ao longo do trabalho, um alternativo que faz uso somente do Facebook Prophet e depende unicamente dos resultados gerados pelo mesmo para fazer a predição de movimento dos preços. O outro modelo alternativo consiste em uma árvore de decisão extremamente similar à utilizada no modelo principal, diferenciando-se da mesma somente na utilização dos preços de fechamento do dia em vez da previsão futura do Prophet.

Cada um dos modelos foi testado com diversas configurações para garantir a correta execução dos passos planejados. No entanto, para selecionar os resultados de forma imparcial, todos os modelos foram executados exatamente trinta vezes, com as mesmas configurações e em sequência, a fim de obter um cenário o mais semelhante possível.

Os resultados foram avaliados pela sua acurácia na previsão do movimento do preço de fechamento no conjunto de testes. A acurácia foi calculada como uma porcentagem, onde quanto mais próximo de 1, mais acertos e menos erros e vice-versa (sendo mais específico, um resultado de 0.6805 significa uma taxa de acerto de 68.05%). Os resultados de cada execução foram registrados em um arquivo de *log* com os detalhes da execução específica. Eles foram submetidos a tratamento estatístico e apresentados em

Tabela 3 – Tabela estatística de resultados

Estatística			
	Prophet + Árvore	Prophet Puro	Árvore Pura
Média	0,68056872	0,43544829	0,67788310
Desvio Padrão	0,03373331	0,03099663	0,03621119
Desvio Médio	0,02603475	0,02425100	0,02731964
Máximo	0,75829384	0,49814841	0,74407583
Mínimo	0,61611374	0,38583193	0,60663507
Variância	0,00113794	0,00096079	0,00131125

Fonte: Autor

tabela, como mostrado na tabela 2.

Dado sua pouca relevância na análise de performance, os resultados completos serão registrados no apêndice A.

Na tabela mensuramos os resultados obtidos através de seis métricas diferentes:

- **Média:** É a soma de todos os valores de uma amostra ou população dividida pelo número de valores. Ela representa o valor central de uma distribuição de dados.
- **Desvio Padrão:** É uma medida da dispersão dos dados em torno da média. Ele é calculado como a raiz quadrada da variância. Ele fornece uma medida da variação dos dados e ajuda a indicar se os dados estão concentrados ou espalhados.
- **Desvio Médio:** É a diferença entre cada valor da amostra ou população e a média aritmética. Ele é usado para medir a precisão dos resultados ou para avaliar a estabilidade dos processos.
- **Máximo:** É o valor mais alto da amostra. Ele é usado para estabelecer o limite superior da amostra
- **Mínimo:** É o valor mais baixo da amostra. Ele é usado para estabelecer o limite inferior da amostra
- **Variância:** É uma medida da dispersão dos dados em torno da média. Ela é calculada como a média dos quadrados das diferenças entre cada valor da amostra ou população e a média aritmética.

A média é utilizada para mensurar a qualidade geral das previsões de cada um dos modelos, provendo um indicador genérico de o quão útil é cada um em situações onde uma quantidade limitada de iterações de treinamento está disponível.

Para propósitos de análise, podemos juntar os valores do desvio padrão, desvio

médio e variância em uma representação da variação de valores em volta da média, mensurando a inconsistência do modelo em alcançar resultados similares em treinos distintos.

Por último, o máximo e o mínimo oferecem o melhor e o pior resultado alcançado com cada um dos modelos, variações essas que acontecem dado as diferenças individuais em cada treinamento, seja pela ordem que o modelo identificou características úteis nos dados quando pela compatibilidade da base de testes com as *features* extraídas, entre outros motivos.

Nesses resultados podemos concluir que o modelo que utiliza apenas o Facebook Prophet apresentou a menor precisão dos três por uma grande margem, apresentando performance superior somente nos quesitos de precisão e distribuição. Pode-se inferir que esse resultado é causado pela incapacidade do Facebook Prophet de sozinho interpretar os dados desta natureza e aprender suas características, porém é possível que em outros tipos de ações ou em outros *timeframes* (em vez de utilizar dados diários como foi testado no modelo, experimentar utilizar dados semanais ou dados coletados a cada hora) o modelo fosse capaz de alcançar um resultado adequado, no entanto para o índice S&P500 com dados diários ele se mostra uma ferramenta pouco proveitosa.

Quando comparamos os resultados do modelo que faz uso do Prophet em conjunto da árvore de decisão com o que utiliza a mesma árvore porém com os dados atuais de fechamento é visível uma pequena melhoria em todas as métricas apresentadas pelo que faz uso do Prophet, tanto uma média maior quanto dispersão menor e máximos e mínimos maiores. Essas métricas apontam esse modelo como ligeiramente superior ao que utilizam somente a árvore para fazer o treinamento, porém as causas não ficam completamente claras do porquê, podendo ser suposto que as lacunas de aprendizado do Prophet são cobertas pela árvore e vice-versa, esse auxílio na previsão não sendo encontrado no outro modelo cuja árvore precisa identificar os padrões dos dados apenas com o auxílio dos indicadores e mais nada, tornando assim, ligeiramente menos capaz de identificar os padrões apresentados.

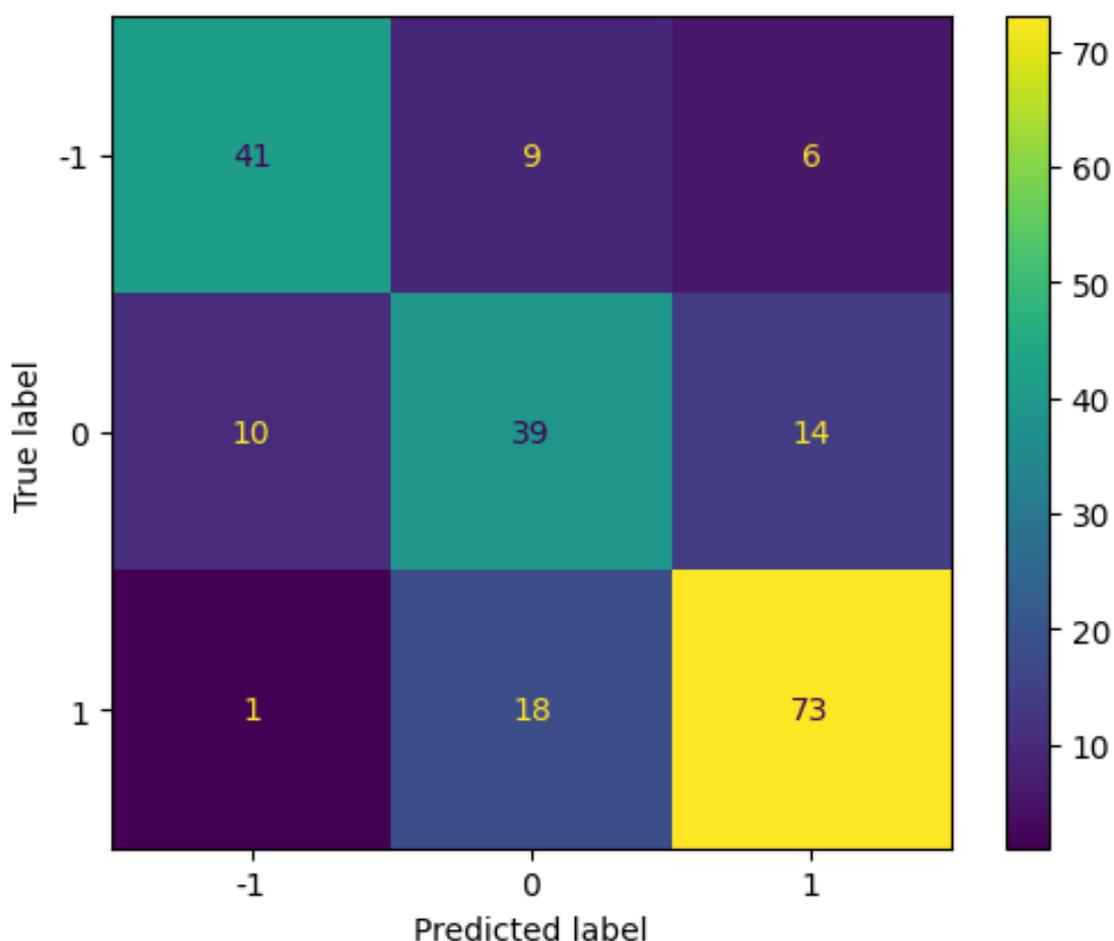
O último ponto a ser avaliado dessa última comparação seria o tempo de treinamento dos modelos, onde o modelo que não faz uso do Prophet apresentou um tempo de treinamento cerca de dez vezes menor que sua alternativa, porém como o ponto principal de um sistema que lida com dinheiro real seria a consistência da previsão, é discutível o peso que isto teria na avaliação de cada desde que nenhum deles ultrapasse um lapso de tempo aceitável, o que não é o caso dado que o mais lento deles (o que utiliza

o Prophet) consegue executar cada ciclo do programa em cerca de meio segundo.

5.2 Análise dos resultados

Outra representação que podemos utilizar para analisar os resultados obtidos a partir dos modelos é uma matriz de confusão, através da qual podemos observar quais são as ocorrências de previsões corretas e erradas mais comuns no conjunto de testes. Houve uma grande similaridade na distribuição entre os modelos testados, conseqüentemente será avaliado somente uma das matrizes, a qual foi extraída do modelo que utiliza o Prophet em conjunto da árvore de decisão. A matriz pode ser vista na Figura 12:

Figura 12 – Matriz de confusão



Fonte: Autor

Para lembrar, na matriz de confusão o "1" representa uma previsão de aumento significativo, enquanto que o "-1" representa uma queda significativa e o "0" representa a

Tabela 4 – Tabela estatística de resultados

Diferença	Quantidade
0	153
1	28
-1	23
-2	6
2	1

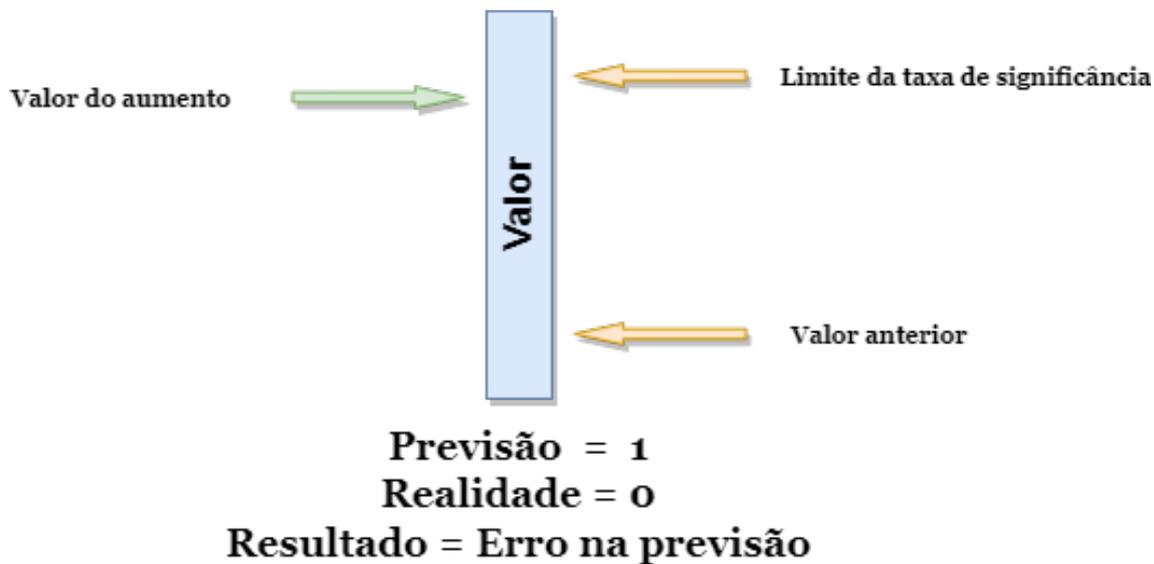
Fonte: Autor

ausência de mudança significativa. Dessa forma, na matriz, podemos ver por exemplo, que em 73 leituras o modelo previu um aumento significativo e acertou essa previsão, mas em 6 ocasiões ele previu um aumento e ocorreu uma queda.

Alguns pontos interessantes que podem ser observados na matriz são que há uma predominância de leituras de crescimento, enquanto que às distribuições das leituras de decréscimo e de estabilidade se mantêm similares. Isto pode ser facilmente explicado pela tendência apresentada pelo índice no geral onde há uma propensão do mesmo a eventualmente sempre crescer em valor.

Outro ponto interessante é o quão pouco a predição erra entre subidas e descidas do preço, isto pode ser visto melhor na tabela 4 onde podemos observar que apesar da quantidade considerável de erros envolvendo predições ou valores estáveis, erros com diferença de dois ou dois negativo (que consiste em prever um subida no preço quando ocorre uma queda e vice-versa) são substancialmente menos comuns. Isto pode ser explicado pela própria geração destes valores, onde a utilização de uma porcentagem rígida que define o quanto uma mudança no valor é considerada uma alteração significativa e insignificante abre espaço para mudanças muito próximas, porém insuficientes para serem categorizadas da mesma maneira. Uma representação gráfica disso pode ser vista na figura 13, onde apesar do aumento significativo no valor, por não alcançar o limite de significância a previsão acaba por ser contabilizada como incorreta.

Figura 13 – Diagrama de mudança



Fonte: Autor

5.3 Resultados da simulação de *trading*

Utilizando os parâmetros para:

- Dias entre trades: 2
- Capital Inicial: 10000
- Período para trades: 120

Em um modelo utilizando Árvore de decisão + Prophet com acurácia de 75,8% nos dados de teste, os resultados obtidos estão registrados na tabela 5.

Podemos comparar esta tabela com uma segunda tabela apresentada na Figura 7, onde as configurações utilizadas foram:

- Dias entre trades: 0
- Capital Inicial: 10000
- Período para trades: 150

Este modelo também utiliza a Árvore de decisão + Prophet, porém desta vez com acurácia de 72,0% nos dados de teste, este foi o modelo com a melhor performance em trinta execuções, mesmo número de execuções que alcançou o resultado do modelo anterior.

Tabela 5 – Tabela de resultados do simulador

Tabela de Transações	
Transação de compra	
Valor da ação - 3749.63	Ações Compradas - 2.66693
Transação de venda	
Valor da ação - 3911.74	Capital Adquirido - 10432.33599
Transação de compra	
Valor da ação - 3818.83	Ações Compradas - 2.731815
Transação de venda	
Valor da ação - 3831.39	Capital Adquirido - 10466.647583
Transação de compra	
Valor da ação - 3854.43	Ações Compradas - 2.715485
Transação de venda	
Valor da ação - 4280.15	Capital Adquirido - 11622.683939
Transação de compra	
Valor da ação - 3986.16	Ações Compradas - 2.91576
Transação de venda	
Valor da ação - 3979.87	Capital Adquirido - 11604.343811
Transação de compra	
Valor da ação - 3693.23	Ações Compradas - 3.142058
Transação de venda	
Valor da ação - 3719.04	Capital Adquirido - 11685.440335
Transação de compra	
Valor da ação - 3639.66	Ações Compradas - 3.210586
Transação de venda	
Valor da ação - 3859.11	Capital Adquirido - 12390.003366
Transação de compra	
Valor da ação - 3856.1	Ações Compradas - 3.213092
Transação de venda	
Valor da ação - 4026.12	Capital Adquirido - 12936.293237
Quantia final: 12936.293237	

Fonte: Autor

Tabela 6 – Segunda tabela de resultados do simulador

Tabela de Transações	
Transação de compra	
Valor da ação - 3930.08	Ações Compradas - 2.544477
Transação de venda	
Valor da ação - 4023.89	Capital Adquirido - 10238.697431
Transação de compra	
Valor da ação - 3923.68	Ações Compradas - 2.609463
Transação de venda	
Valor da ação - 4132.15	Capital Adquirido - 10782.692164
Transação de compra	
Valor da ação - 3900.86	Ações Compradas - 2.764183
Transação de venda	
Valor da ação - 3911.74	Capital Adquirido - 10812.766479
Transação de compra	
Valor da ação - 3821.55	Ações Compradas - 2.829419
Transação de venda	
Valor da ação - 3831.39	Capital Adquirido - 10840.607962
Transação de compra	
Valor da ação - 3854.43	Ações Compradas - 2.812506
Transação de venda	
Valor da ação - 4207.27	Capital Adquirido - 11832.972621
Transação de compra	
Valor da ação - 3986.16	Ações Compradas - 2.968514
Transação de venda	
Valor da ação - 4006.18	Capital Adquirido - 11892.402276
Transação de compra	
Valor da ação - 3693.23	Ações Compradas - 3.220055
Transação de venda	
Valor da ação - 3719.04	Capital Adquirido - 11975.511885
Transação de compra	
Valor da ação - 3639.66	Ações Compradas - 3.290283
Transação de venda	
Valor da ação - 3859.11	Capital Adquirido - 12697.564518
Transação de compra	
Valor da ação - 3856.1	Ações Compradas - 3.292851
Transação de venda	
Valor da ação - 4026.12	Capital Adquirido - 13257.415123
Quantia final: 13257.415123	

Fonte: Autor

Ao comparar esses dois resultados, podemos concluir que o aumento do período utilizado para transações é refletido com precisão nas duas tabelas, com uma pequena variação no início da segunda, onde os trinta dias adicionais de simulação se refletem em uma transação alternativa e quatro transações adicionais, todas no início das tabelas. Ao observarmos a segunda tabela, notamos que a terceira compra do usuário, que precede a venda cujo preço de fechamento foi de 3911.74 dólares, é equivalente à primeira compra da primeira tabela, cujo preço de compra é consideravelmente inferior (3900 dólares para 3749,63 dólares).

Esta interação nos fornece uma visão valiosa sobre o funcionamento do simulador. No entanto, é possível observar que existe uma ineficiência intrínseca nas transações executadas pelo simulador, devido ao fato de que ele não utiliza técnicas de controle de risco. Como resultado, o simulador mantém sua posição de compra e venda até que uma oportunidade seja identificada, mesmo que essa oportunidade possa não ser a mais lucrativa. Um trader, por outro lado, poderia utilizar previsões de maneira mais produtiva e identificar mais pontos de venda e compra para maximizar os lucros em menos tempo.

Os parâmetros de previsão podem ser ajustados para se adaptar melhor a cada conjunto de dados. Como mencionado anteriormente, a solução inteira é flexível e, teoricamente, pode lidar com qualquer conjunto histórico de preços, não apenas o S&P500. No entanto, para alcançar resultados desejados com cada conjunto, seria necessário realizar ajustes diferentes, os quais também afetariam o simulador. Nos testes realizados com diferentes parâmetros, as mudanças foram mínimas para o S&P500, o que pode ser explicado pela natureza de períodos de constante aumento e quedas bruscas nos preços.

5.4 Conclusão dos resultados

Com base nas validações apresentadas nesta seção, podemos chegar a algumas conclusões sobre o desempenho do modelo desenvolvido. Em primeiro lugar, devemos observar que, mesmo nos dados de teste, a performance do modelo não é perfeita, apresentando cerca de 25% ou mais de taxa de erro ao ser aplicado na carga de testes. No entanto, é importante lembrar que, com a aplicação adequada de métodos de controle de riscos, uma taxa de acerto superior a 50% é mais do que suficiente para gerar resultados satisfatórios, como comprovado pelo simulador. Portanto, podemos afirmar que o desenvolvimento da solução proposta foi bem-sucedido.

Além disso, como foi mostrado na matriz de confusão, essa margem de erro pode ser exagerada, dependendo do limite da taxa de significância utilizada. Por essa razão, medir a qualidade da previsão apenas com a acurácia nos testes pode não ser a melhor estratégia. Em vez disso, é importante compreender completamente o modelo para buscar a melhor configuração e desempenho possíveis.

6 CONSIDERAÇÕES FINAIS

Neste estudo, apresentamos um método detalhado para desenvolver um modelo de previsão de preços eficiente e robusto para o índice S&P500, utilizando a ferramenta de previsão Prophet do Facebook, bem como outras técnicas avançadas de inteligência artificial e aprendizado de máquina.

6.1 Problemas Encontrados

Durante o processo de desenvolvimento, foram identificados diversos desafios decorrentes de preconceitos em relação às tecnologias e ferramentas utilizadas, o que resultou em um prazo de desenvolvimento mais longo e a necessidade de buscar soluções alternativas para atender às necessidades da ferramenta.

Os principais problemas encontrados foram relacionados à coleta automatizada de dados, conforme descrito na seção de desenvolvimento, bem como com o desempenho básico do Facebook Prophet para lidar com previsões multivariadas. Cada caso pode ser analisado em detalhes para identificar onde foram cometidos erros no projeto da ferramenta e como evitá-los no futuro.

- Adquirir dados: Em muitos outros projetos, incluindo outros desenvolvidos na disciplina, o Yahoo Finance foi frequentemente utilizado devido à sua praticidade e disponibilidade gratuita. No entanto, para este modelo específico, as limitações da ferramenta gratuita, como a falta de acesso a dados históricos e recentes, a tornam inadequada para atender ao requisito original de manter os dados históricos constantemente atualizados com entradas relevantes e recentes. Diante disso, foram procuradas outras ferramentas que atendessem aos requisitos necessários para o uso. A API da NASQUAD foi avaliada e testada como a principal opção para ser incluída no modelo. No entanto, posteriormente foi descoberto que o módulo do S&P500 necessário só pode ser acessado por meio de uma API de terceiros que tem custo, tornando-a também inviável para este trabalho. Finalmente, foi decidido descontinuar a automatização da entrada de dados, optando por permitir que o usuário insira manualmente o conjunto de dados desejado. Isso sacrifica parte da facilidade de uso, mas permite maior flexibilidade e reutilização do software a longo prazo.

- Modelo multivariado pelo Prophet: Um dos requisitos essenciais do projeto era a aplicação de previsão multivariada, que consistiria na utilização da ferramenta Prophet para prever o preço utilizando o valor de fechamento, considerando os dados de indicadores. Como o diferencial do trabalho estava justamente na aplicação dessas informações adicionais na previsão, era imprescindível implementá-lo. Por essa razão entre outras, o Prophet foi escolhido como a ferramenta ideal, pois sua implementação incluía essa funcionalidade. No entanto, descobriu-se que sua implementação não era adequada para este problema específico, pois ele exige que os dados adicionais estejam disponíveis para o período a ser previsto, o que não é possível neste caso. Uma das soluções consideradas e testadas foi realizar múltiplas previsões, primeiro para os indicadores e depois para o modelo principal, utilizando as informações previstas. No entanto, os resultados obtidos foram insatisfatórios, então essa abordagem foi descartada. Por fim, foi utilizada outra tecnologia auxiliar já estabelecida como uma solução confiável para a análise multivariada, esta sendo a árvore de decisão, em conjunto com o Prophet para compensar a falta da funcionalidade de análise multivariada nele. Dessa forma então, foi possível alcançar os resultados desejados.

Estes acontecimentos ilustram um grande problema com a profundidade de conhecimento prático das ferramentas selecionadas no período de planejamento do modelo, sendo assim para trabalhos futuros fica a recomendação de um mais extenso uso das ferramentas com ambientes similares de modo a melhor simular as situações que serão enfrentadas durante o desenvolvimento.

6.2 Conclusão

Este trabalho requereu uma ampla gama de conhecimentos em várias áreas, especialmente no campo do *trading* financeiro. Foram estudados temas como tipos de investidores, investimentos, tendências *bullish* e *bearish*, leitura de gráficos e outras técnicas avançadas de previsão de preços manuais utilizadas por profissionais da área. Além disso, foi fundamental um conhecimento aprofundado em modelos de inteligência artificial, com ênfase no Facebook Prophet e na teoria geral de aprendizado de máquina.

Neste trabalho, foram apresentados diversos tópicos relevantes para garantir que o leitor tenha uma compreensão completa dos temas utilizados a seguir. Em seguida,

foi apresentado o desenvolvimento completo do modelo, incluindo os resultados finais obtidos e a análise dos mesmos. Com base nos resultados, conclui-se que o modelo apresentou resultados satisfatórios e atingiu as expectativas.

O modelo apresentou uma precisão satisfatória ao prever as variações de preços do índice S&P500, embora não tenha alcançado uma precisão absoluta, o que é compreensível devido à natureza imprevisível dos dados. Assim, os resultados são considerados positivos para utilizar o modelo como uma ferramenta de auxílio para investidores iniciantes, fornecendo indicações que, quando utilizadas em conjunto com técnicas de gerenciamento de risco, podem auxiliar na tomada de decisões. Além disso, a estrutura do modelo pode ser facilmente analisada por traders experientes e pode revelar padrões úteis para a análise manual de investimentos potenciais.

6.3 Trabalhos Futuros

Existem várias oportunidades para melhorias futuras no modelo. Por exemplo, pode-se implementar a coleta automatizada de dados utilizando serviços pagos, expandir o modelo para prever as variações de outros ativos, e utilizar diversas outras ferramentas de aprendizado de máquina, como o NeuralProphet, para aprimorar o desempenho do modelo. Essas são apenas algumas das possíveis áreas de trabalho futuro que podem ser exploradas.

REFERÊNCIAS

- BENGIO YOSHUA; COURVILLE, A. G. I. J. **Deep learning: adaptive computation and machine learning**. [S.l.]: The MIT Press, 2016. (Adaptive Computation and Machine Learning series). ISBN 0262035618; 9780262035613.
- BODIE, Z.; KANE, A.; MARCUS, A. J. **Investments**. Boston: McGraw-Hill/Irwin, 2009.
- BURKOV, A. **The Hundred-Page Machine Learning Book**. [S.l.: s.n.], 2019. ISBN 9781999579500; 199957950X.
- CAMPESATO, O. **Artificial Intelligence, Machine Learning, and Deep Learning**. [S.l.]: Mercury Learning and Information, 2020. ISBN 1683924673; 9781683924678.
- CHEN, J. **Histogram**. 2022. Available from Internet: <<https://www.investopedia.com/terms/h/histogram.asp>>.
- CHOLLET, F. **Deep Learning with Python**. [S.l.]: Manning Publications Co, 2017. ISBN 1617294438; 9781617294433.
- DEMIRAY, S.; GÜRHANLI, A. Stock closing prediction with machine learning algorithms. **2021 International Journal of Engineering Research and Applications**, Vol. 11, Issue 1, (Series-III), p. 59–62, 2021.
- DOUGLAS, M. **Trading - Atitude mental do trader de sucesso**. [S.l.]: Bookout, 2013. ISBN 9789898694010.
- DUTT, A. Time series forecasting using machine learning. **IPHS Senior Research Seminar**, 2021. Available from Internet: <digital.kenyon.edu/dh_iphs_ss/6/>.
- ELDER, A. **Come Into My Trading Room: A Complete Guide to Trading**. Wiley, 2002. (Wiley Trading). ISBN 9780471273820. Available from Internet: <<https://books.google.com.br/books?id=mna7CdjGdcYC>>.
- ELDER, A. **Entries and Exits: Visits to Sixteen Trading Rooms**. [S.l.]: Wiley, 2006. (Wiley Trading). ISBN 9780471678052.
- ELDER, D. A. **The New Trading for a Living**. 1. ed. Hoboken, New Jersey: John Wiley Sons, Inc., Hoboken, 2014. 303 p.
- FANG, W.-X. et al. Combine facebook prophet and lstm with bpnn forecasting financial markets: the morgan taiwan index. In: **2019 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS)**. [S.l.: s.n.], 2019. p. 1–2.
- GARLAPATI, A. et al. Stock price prediction using facebook prophet and arima models. In: **2021 6th International Conference for Convergence in Technology (I2CT)**. [S.l.: s.n.], 2021. p. 1–7.
- GORDON, J. **Derivative Securities - Explained**. 2022. Available from Internet: <https://thebusinessprofessor.com/en_US/investments-trading-financial-markets/derivative-securities-explained>.

GRAHAM, B. **The Intelligent Investor Rev Ed.** [S.l.]: Harper Paperbacks, 2003.

HAYES, A. **Introduction to Technical Analysis Price Patterns.** 2022. Available from Internet: <<https://www.investopedia.com/articles/technical/112601.asp>>.

JAMES, G. et al. **An Introduction to Statistical Learning: with Applications in R.** Springer, 2013. Available from Internet: <<https://faculty.marshall.usc.edu/gareth-james/ISL/>>.

KRAMER, L. **An Overview of Bull and Bear Markets.** 2022. Available from Internet: <<https://www.investopedia.com/insights/digging-deeper-bull-and-bear-markets>>.

KULSHRESHTHA, S.; A, V. An arima-lstm hybrid model for stock market prediction using live data. **Journal of Engineering Science and Technology Review**, Eastern Macedonia and Thrace Institute of Technology, v. 13, n. 4, p. 117 – 123, 2020. ISSN 17919320.

KUMAR, S. et al. A survey on artificial neural network based stock price prediction using various methods. **2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS)**, p. 1866–1872, 2021.

LARSON, M. **12 Simple Technical Indicators (That Really Work) II.** [S.l.]: John Wiley Sons, Inc., 2012. ISBN 9781119204428; 1119204429; 9781592802906; 1592802907.

LI, A. W.; BASTOS, G. S. Stock market forecasting using deep learning and technical analysis: A systematic review. **IEEE Access**, v. 8, p. 185232–185242, 2020. Available from Internet: <ieeexplore.ieee.org/abstract/document/9220868>.

MALKIEL, B. G. **A Random Walk Down Wall Street.** [S.l.]: Norton, New York, 1973.

MITCHELL, C. **Bar Chart.** 2021. Available from Internet: <<https://www.investopedia.com/terms/b/barchart.asp>>.

MITCHELL, C. **Understanding Basic Candlestick Charts.** 2022. Available from Internet: <<https://www.investopedia.com/trading/candlestick-charting-what-is-it/>>.

MITCHELL, T. M. **Machine learning.** [S.l.]: McGraw-hill New York, 1997.

MOHAN, S. et al. Stock price prediction using news sentiment analysis. In: **2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService).** [S.l.: s.n.], 2019. p. 205–208.

PETERS, K. **Line Chart.** 2022. Available from Internet: <<https://www.investopedia.com/terms/l/linechart.asp>>.

PRODANOV, C. C.; FREITAS, E. C. de. **Metodologia do trabalho científico: Métodos e Técnicas da Pesquisa e do Trabalho Acadêmico.** [S.l.]: Feevale, 2013.

SALDIVAR, F.; ORTIZ, M. Stock market price prediction using various machine learning approaches. In: . [S.l.: s.n.], 2019.

SCHWAGER, J. D. **Getting Started in Technical Analysis.** [S.l.]: Wiley, 1999. ISBN 0471295426; 9780471295426.

SMIGEL, L. **What Is Open High Low Close in Stocks**. 2022. Available from Internet: <<https://analyzingalpha.com/open-high-low-close-stocks>>.

TAYLOR, S.; LETHAM, B. Forecasting at scale. 09 2017.

APÊNDICE A – TABELA DE RESULTADOS DA ACURÁCIA

Tabela 7 – Tabela de resultados da acurácia

Prophet + Árvore	Prophet Puro	Árvore Pura
0,66824645	0,44995319	0,66350711
0,71090047	0,4798434	0,69668246
0,71090047	0,47804059	0,68720379
0,75829384	0,38770357	0,69668246
0,65402844	0,44464601	0,67772512
0,73933649	0,43327451	0,68246445
0,67298578	0,42780052	0,61137441
0,66350711	0,3999706	0,68246445
0,71563981	0,44245561	0,68720379
0,68246445	0,42971175	0,66824645
0,69194313	0,42412863	0,68720379
0,63981043	0,44748552	0,74407583
0,68246445	0,42350119	0,71090047
0,71563981	0,38802683	0,69668246
0,64454976	0,44205428	0,61137441
0,69668246	0,39037784	0,71090047
0,72037915	0,38583193	0,67298578
0,69194313	0,46498293	0,66350711
0,66350711	0,44318651	0,72511848
0,61611374	0,44307917	0,72037915
0,62559242	0,39500363	0,61137441
0,69668246	0,43337296	0,64454976
0,67298578	0,42781631	0,63033175
0,68720379	0,46634527	0,69194313
0,66824645	0,49814841	0,66824645
0,61611374	0,41390427	0,60663507
0,68246445	0,42861285	0,67772512
0,65876777	0,4655116	0,72985782
0,67298578	0,49430853	0,67298578
0,69668246	0,41437023	0,70616114

Fonte: Autor

APÊNDICE B – CÓDIGO FONTE DO MODELO

```

1
2 '''
3 Sistema de previsão do índice S&P500 utilizando o Prophet e Árvore de
4   ↳ decisão
5 '''
6 #Importando todos os pacotes necessários.
7 import sys
8 from datetime import datetime
9 from pandas import read_csv
10 from prophet.plot import plot_plotly
11 from prophet import Prophet
12 from sklearn.model_selection import train_test_split
13 from sklearn import tree
14 from sklearn import metrics
15 from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
16 import matplotlib.pyplot as plt
17 import pandas as pd
18 import plotly.express as px
19 import numpy as np
20 import pandas_ta as ta
21 pd.options.mode.chained_assignment = None
22 original_stdout = sys.stdout # Save a reference to the original
23   ↳ standard output
24
25 #Parâmetros de previsão
26 TIMEFRAME = '1d' # Período a ser visualizado por cada vela
27 WARMUP = 200 # Dias necessários para calculo de indicadores
28 DAYS_AHEAD = 7 # Dias a frente a serem previstos pelo Prophet
29 PERCENTAGE_TEST_DATA = 5 # Porcentagem dos dados que serão separados
30   ↳ para teste
31 PERCENTAGETOINCREASE = 1 # Porcentagem de diferença esperada entre o
32   ↳ dia atual e o futuro
33 PERCENTAGE_TEST_TREE = 0.2 # Porcentagem dos dados a serem usados para
34   ↳ teste na árvore de decisão
35 QUANTIDADE_TESTES = 30 # Quantas vezes executar o programa
36
37 #Parâmetros de simulação
38 DAYS_BETWEEN_TRADES = 2 # Quantos dias não fazer trade após cada
39   ↳ transação
40 INITIAL_MONEY = 10000 # Capital inicial da simulação de trade
41 PAST_PERIOD_TO_TRADE = 120 # Quantidade de dias a partir do mais
42   ↳ recente para simular as transações

```

```

38 resultados = np.empty(QUANTIDADE_TESTES, dtype=float)
39
40 for log_number in range(0, QUANTIDADE_TESTES):
41
42     FILENAME = 'Logs/Log{}.txt'.format(log_number)
43
44     #Configura para salvar em arquivo de texto
45     sys.stdout = open(FILENAME, 'w', encoding="utf-8") # Change the
46     → standard output to the file we created.
47     #print('This message will be written to a file.')
48     #sys.stdout = original_stdout # Reset the standard output to its
49     → original value
50
51     #Utilização de dados locais
52     df = read_csv('Dados/HistoricalData.csv') #Busca os dados e insere
53     → em um dataframe
54     df = df[df['Volume']!=0] #Remove os dias onde não houveram
55     → transações (fins de semana)
56     df = df.reindex(index=df.index[:-1]).reset_index().drop('index',
57     → axis=1) #Separa o index da data e cria uma tabela separada
58     #Calcula os indicadores
59     df['SMA50'] = ta.sma(close=df['Close'], length=50)
60     df['SMA200'] = ta.sma(close=df['Close'], length=200)
61     df['RSI14'] = ta.rsi(close=df['Close'], length=14)
62     #Remove as primeiras velas (usadas para aquecer os indicadores)
63     df.drop(df.index[:WARMUP], inplace=True)
64     df = df.reset_index().drop('index', axis=1)
65
66     #Cria um dataframe formatado para o modelo Prophet
67     ProphetDataframe = df[['Date', 'Close']]
68     ProphetDataframe = ProphetDataframe.rename(columns={'Date': 'ds',
69     → 'Close': 'y'})
70
71     #Visualiza dos dados a serem inseridos no Prophet
72     fig = px.line(ProphetDataframe, x='ds', y='y')
73     fig.update_xaxes(rangeslider_visible=False)
74     fig.show()
75
76     # Separa os N primeiros valores para ser os dados de treinamento,
77     → onde N é o parametro definido acima
78     train_data = ProphetDataframe[:int(-ProphetDataframe.size*
79     (PERCENTAGE_TEST_DATA / 100))]
80     # Os outros valores são separados para os testes e avaliação do
81     → modelo
82     test_data = ProphetDataframe.drop(train_data.index)
83
84     # Visualizar a quantidade de entradas

```

```

78 print(f'training data size : {train_data.shape}')
79 print(f'testing data size : {test_data.shape}')
80
81 # Definir o algoritmo do Prophet e inserir os dados a serem
  → previstos
82 fbp = Prophet(daily_seasonality=True)
83 fbp.fit(ProphetDataframe)
84 future = fbp.make_future_dataframe(periods=DAYS_AHEAD)
85 forecast = fbp.predict(future)
86 plot_plotly(fbp, forecast)
87
88 #Arvore de decisao
89
90 dytree = [] #Cria uma lista para armazenar as mudanças de valor
  → para cima e baixo
91 dftree = df[['Date', 'SMA50', 'SMA200', 'RSI14']] #Insere a data e
  → indicadores na tabela que será inserida na árvore de decisão
92 dftree = dftree.iloc[0:-DAYS_AHEAD,:] #Remove os ultimos N dados,
  → sendo um passo necessário por não haver a informação correta
  → de se esse valor subiu ou desceu
93 ''' Remove tanto os N primeiros quanto os N ultimos valores da
  → previsão do Prophet, os N primeiros por não estarem ligados a
94 previsão em si, sendo que utilizaremos os valores previstos para o
  → dia para cada previsão, e os N ultimos pela mesma razão da linha
  → anterior,
95 que não temos a previsão correta para fazer o treinamento, sendo
  → dados inúteis'''
96 dftree['Trend'] =
  → forecast['trend'].iloc[DAYS_AHEAD:-DAYS_AHEAD].reset_index()
97 .drop('index', axis=1)
98 dftree.loc[:, 'Date'] = [datetime.timestamp(x) for x in
  → pd.to_datetime(dftree.loc[:, 'Date'])] #Converte toda a coluna
  → pra o tipo datetime, que é aceita pelo sklearn
99 #Separa os valores N dias a frente em uma lista para comparar com
  → os dados de treinamento e visualizar o aumento ou diminuição
100 Values = [df.loc[x + DAYS_AHEAD, 'Close'] for x in
  → range(0, len(df.loc[:, 'Date']) - DAYS_AHEAD)]
101 for x, value in enumerate(Values):
102     #Caso os valores apresentem um aumento nesses N dias, é salvo o
  → o valor 1
103     if df.loc[x, 'Close'] + (df.loc[x, 'Close'] *
  → (PERCENTAGETOINCREASE / 100)) < Values[x]:
104         dytree.append(1)
105     #Caso os valores apresentem uma queda nesses N dias, é salvo o
  → valor -1
106     elif df.loc[x, 'Close'] - (df.loc[x, 'Close'] *
  → (PERCENTAGETOINCREASE / 100)) > Values[x]:
107         dytree.append(-1)

```

```

108         else:
109             #Caso nenhuma mudança relevante seja apresentada nesses N
110             ↪ dias, é salvo o valor 0
111             dytree.append(0)
112         print(dftree)
113         print(dytree)
114
115     print(f'Output for tree shape : {pd.DataFrame(dytree).shape}')
116
117     #Separa as entradas da saída e converte para arrays numpy
118     X = dftree[["Date", "SMA50", "SMA200", "RSI14", "Trend"]].to_numpy()
119     y = np.array(dytree)
120
121     #Divide os arquivos de treino e teste
122     X_train, X_test, y_train, y_test = train_test_split(X, y,
123         ↪ test_size=PERCENTAGE_TEST_TREE, shuffle=True, stratify=y)
124
125     #Faz a classificação
126     clf = tree.DecisionTreeClassifier()
127
128     clf.fit(X_train, y_train)
129
130     #Calcula o acerto na base de treino
131     y_pred_train = clf.predict(X_train)
132     print("Accuracy train:", metrics.accuracy_score(y_train,
133         ↪ y_pred_train), end='\t')
134
135     #Calcula o acerto na base de testes
136     y_pred_test = clf.predict(X_test)
137     print("Accuracy test:", metrics.accuracy_score(y_test,
138         ↪ y_pred_test))
139     resultados[log_number] = metrics.accuracy_score(y_test,
140         ↪ y_pred_test)
141
142     #Matriz de confusão
143     cm = confusion_matrix(y_test, y_pred_test, labels=clf.classes_)
144     disp = ConfusionMatrixDisplay(confusion_matrix=cm,
145         ↪ display_labels=clf.classes_)
146     disp.plot()
147     plt.show()
148
149     #Mostra as saídas desejadas e obtidas
150     x = np.asmatrix(np.transpose(np.array([y_test, y_pred_test, y_test -
151         y_pred_test])))
152     pd_x = pd.DataFrame(x, columns=['y_test', 'y_pred', 'difference'])
153     print(pd_x)
154
155     #Podemos contar quantas vezes a diferença chegou a 2 ou -2, o que
156     ↪ implicaria que a previsão errou completamente a direção

```

```

150     contagemDiferencas = pd_x["difference"].value_counts()
151     print(contagemDiferencas)
152
153     #-----Simulação de compras e
154     ↪ vendas-----
155
156     FILENAME = 'Transacoes/Log{}.txt'.format(log_number)
157     sys.stdout = open(FILENAME, 'w', encoding="utf-8")
158
159
160     updated_values = df[['Close']]
161     updated_values = updated_values.iloc[-PAST_PERIOD_TO_TRADE -
162     ↪ DAYS_AHEAD:-DAYS_AHEAD ,:].reset_index(drop = True)
163     complete_prediction = clf.predict(X)
164     complete_prediction = complete_prediction[-PAST_PERIOD_TO_TRADE:]
165     TRADE_OPEN = True
166     DAYS_UNTIL_OPEN = 0
167     MONEY = INITIAL_MONEY
168     STOCKS = 0
169     for x in range(PAST_PERIOD_TO_TRADE):
170         if TRADE_OPEN is True or DAYS_UNTIL_OPEN == 0:
171             if complete_prediction[x] == 1 and MONEY is not 0:
172                 STOCKS = MONEY / updated_values.iloc[x]
173                 MONEY = 0
174                 TRADE_OPEN = False
175                 DAYS_UNTIL_OPEN = DAYS_BETWEEN_TRADES
176                 print(f"Transação de compra: Stock Price -
177                 ↪ {updated_values.iloc[x].to_string(index=False)}
178                 ↪ Stock Bought - {STOCKS.to_string(index=False)}")
179             elif complete_prediction[x] == -1:
180                 if MONEY is 0:
181                     MONEY = STOCKS * updated_values.iloc[x]
182                     STOCKS = 0
183                     TRADE_OPEN = False
184                     DAYS_UNTIL_OPEN = DAYS_BETWEEN_TRADES
185                     print(f"Transação de venda: Stock Price -
186                     ↪ {updated_values.iloc[x].to_string(index=False)}
187                     ↪ Money Gained -
188                     ↪ {MONEY.to_string(index=False)}")
189
190         else:
191             DAYS_UNTIL_OPEN -= 1
192             if DAYS_UNTIL_OPEN == 0:
193                 TRADE_OPEN = True
194
195     if MONEY is 0:
196         MONEY = STOCKS * updated_values.iloc[PAST_PERIOD_TO_TRADE - 1]
197     print(f"Quantia final: {MONEY.to_string(index=False)}")

```

```
191
192 with open('Logs/Resultados.txt', 'w') as f:
193     sys.stdout = f #
194     print(resultados)
195     sys.stdout = original_stdout
196
```
