



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

ANDRIUS DE FREITAS PINHEIRO

GERAÇÃO DE CENÁRIOS PARA TOMADA DE DECISÃO DE ATIVOS
FINANCEIROS POR MEIO DE ANÁLISE FUNDAMENTALISTA E TÉCNICA

Sorocaba

2023

ANDRIUS DE FREITAS PINHEIRO

GERAÇÃO DE CENÁRIOS PARA TOMADA DE DECISÃO DE ATIVOS
FINANCEIROS POR MEIO DE ANÁLISE FUNDAMENTALISTA E TÉCNICA

Trabalho de Conclusão de Curso apresentado ao Instituto de Ciência e Tecnologia de Sorocaba (ICTS), Universidade Estadual Paulista “Júlio de Mesquita Filho” (UNESP), como parte dos requisitos para obtenção do grau de bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Galdenoro Botura Junior

Sorocaba

2023

P654g Pinheiro, Andrius de Freitas
Geração de cenários para tomada de decisão de ativos financeiros por meio de análise fundamentalista e técnica / Andrius de Freitas Pinheiro. -- Sorocaba, 2023
79 f.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (Unesp), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: Galdenoro Botura Junior

1. Mercado financeiro. 2. Ações. 3. Dados. 4. Indústria 4.0. 5. Python. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca do Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Instituto de Ciência e Tecnologia
Câmpus de Sorocaba

GERAÇÃO DE CENÁRIOS PARA TOMADA DE DECISÃO DE ATIVOS
FINANCEIROS POR MEIO DE ANÁLISE FUNDAMENTALISTA E TÉCNICA

ANDRIUS DE FREITAS PINHEIRO

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO
PARTE DO REQUISITO PARA A OBTENÇÃO DO GRAU DE **BACHAREL EM**
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Prof^o. Dr. Everson Martins

Coordenador

BANCA EXAMINADORA:

Prof. Dr. GALDENORO BOTURA JR

Orientador/UNESP-Campus de Sorocaba

Prof. Dr. EDUARDO VERRI LIBERADO

UNESP-Campus de Sorocaba

Prof. Dr. JOSÉ ROBERTO RIBEIRO BORTOLETO

UNESP-Campus de Sorocaba

Novembro de 2023

AGRADECIMENTOS

Gostaria de iniciar o presente trabalho prestando meus agradecimentos a quem, não somente na graduação, mas na vida toda, me ajudou e apoiou e teve grande contribuição para essa formação.

Primeiramente agradeço à Deus por me proporcionar saúde e condições para que eu pudesse ingressar e estudar em uma faculdade renomada como a Unesp, não deixando me faltar nada em todo esse período.

Meus agradecimentos também vão à minha família, meu pai Antonio Carlos Pinheiro, minha mãe, Raquel de Freitas Pinheiro e aos meus irmãos Wesley de Freitas Pinheiro e Alexander de Freitas Pinheiro, por todo apoio e por sempre estarem ao meu lado nos bons momentos e nas situações mais difíceis, e que certamente cada um teve uma contribuição muito grande para a minha formação.

Também agradeço aos amigos que fiz na Unesp, os quais compartilhei anos de estudo, por toda amizade e colaboração e que também têm um papel importante na minha formação como engenheiro de controle e automação.

Também presto meus agradecimentos ao corpo docente, aos colaboradores da Unesp Sorocaba, por proporcionarem anos de ensino de qualidade, uma ótima estrutura para estudos e ao meu orientador, Prof. Dr. Galdenoro Botura Junior, o qual sempre se mostrou disposto a ajudar no projeto com ideias inovadoras e orientações de suma importância para a conclusão deste trabalho.

Por fim, presto meus agradecimentos ao Itaú Unibanco S.A., por toda estrutura oferecida em relação ao estágio que foi de extrema importância para meu desenvolvimento profissional e pessoal proporcionado nesse período, em especial aos coordenadores Juliano Drusian Braquini e André Araújo pelo acolhimento, ensinamentos e intermediação com a universidade.

RESUMO

O mercado financeiro, por definição, pode ser definido como um ambiente de negociação, ou seja, compra e venda de valores mobiliários, composto de diversas instituições e órgãos que regularizam e intermediam as transações financeiras, unindo agentes superavitários, que são pessoas ou empresas que possuem o capital e o interesse de investi-lo para uma valorização futura de seu patrimônio, e os agentes deficitários que são os que necessitam do dinheiro do superavitário, por meio de um crédito, empréstimos, entre outros instrumentos que destinarão o capital a outros projetos ou investimentos fomentando o desenvolvimento econômico da sociedade no geral. Um dos produtos mais negociados na bolsa de valores são as ações, que representam a menor fração do capital social de uma empresa, que por sua vez é reconhecida pela sua variação de preço com o tempo, explicado pela lei da procura e oferta. Portanto, se há mais investidores interessados na compra de uma ação do que em sua venda, há uma maior procura por uma mesma quantidade de ativos em circulação, fazendo com que o preço fique mais valorizado e se eleve, e caso haja mais investidores interessados em vendê-las, consequentemente, a mesma se desvalorizará, abaixando seu preço. O objetivo do presente trabalho, promovendo uma maior contextualização sobre o mercado financeiro, consiste em análise de dados financeiros por meio da linguagem de programação Python, que por sua vez tem capacidade de manusear e analisar diferentes parâmetros e indicadores financeiros de empresas de capital aberto na bolsa de valores, aprofundando e dando maior ênfase no mercado de ações, assim como sua dinâmica de preços no decorrer do tempo. Com isso, a ideia é trazer uma resposta abstraída ao usuário sobre qual cenário é o mais indicado, uma suposta compra ou venda do mesmo. Portanto, para a realização deste projeto, foram utilizados processos, tecnologias, ferramentas e conceitos de análise de dados atuais, totalmente aderentes e largamente utilizadas para o mercado de trabalho atualmente, tais como linguagem Python e suas respectivas bibliotecas.

Palavras-chave: Mercado financeiro; ações; dados; Industria 4.0; Python.

ABSTRACT

The financial market, by definition, can be defined as a negotiation environment, that is, the purchase and sale of securities, made up of several institutions that regularize and intermediate financial transactions, bringing together surplus agents, which are people or companies that have the capital and the will in investing it for the future appreciation of their assets, and the deficit agents who are those who need the money from the surplus, through credit, loans, among other instruments that will allocate this capital to other projects or investments promoting the economic development of society in general. One of the most traded products on the stock exchange are shares, which represent the smallest fraction of a company's capital, which is recognized by its price variation over time, explained by the law of demand and supply. Therefore, if there are more investors interested in buying a share than in selling it, there is a greater demand for the same amount of assets in circulation, causing the price to become more valued and rise, and if there are more investors interested in selling them, consequently, it will depreciate, decreasing its price. The work's purpose, promoting contextualization of the financial market, consists of analyzing financial data using the Python programming language, which has the capacity to handle and analyze different parameters and financial indicators of publicly traded companies in the stock exchange, deepening and placing greater emphasis on the stock market, as well as its price dynamics over time. Therefore, the idea is to provide an abstracted answer to the user about which scenario is the most suitable, a supposed purchase or sale of the same. Therefore, to carry out this project, current data analysis processes, technologies, tools and concepts were used, fully adherent and widely used in the job market today, such as the Python language and its respective libraries.

Key Words: Financial market; shares; data; Industry 4.0; Python.

LISTA DE ILUSTRAÇÕES

Figura 1 – Evolução da indústria no decorrer do tempo.	22
Figura 2 - Etapas do Business Intelligence.	23
Figura 3 - Os 5 V's do Big Data	25
Figura 4 - Logo da Linguagem de programação Python	26
Figura 5 - Exemplo de um DataFrame no Python utilizando Pandas	27
Figura 6 - Exemplos de gráficos gerados a partir do Matplotlib	28
Figura 7 - Etapas do Machine Learning	29
Figura 8 - Etapas do processo de ETL	32
Figura 9 - Simplificação do fluxo de ETL	33
Figura 10 - Fluxograma das etapas da análise fundamentalista	35
Figura 11 - Comando de instalação da biblioteca Fundamentus	35
Figura 12 - Importação das bibliotecas no código	36
Figura 13 - Dados financeiros da Petrobrás no site Fundamentus	36
Figura 14 - Lista com as colunas extraídas do site Fundamentus	37
Figura 15 - Lista completa dos indicadores financeiros da bolsa de valores	38
Figura 16 - Definição dos filtros dos indicadores financeiros	40
Figura 17 - comando utilizado para mostrar quantidade de dados	41
Figura 18 – Ativos que cumpriram os primeiros requisitos de seleção	41
Figura 19 – Fluxograma das etapas da análise técnica	42
Figura 20 – Instalação da biblioteca yfinance	43
Figura 21 – importação das bibliotecas necessárias no Python	43
Figura 22 – Extração dos dados históricos do índice Ibovespa	44
Figura 23 – Demonstração das últimas cotações obtidas do Ibovespa	44
Figura 24 – Cotação histórica do índice Ibovespa através da biblioteca matplotlib	45
Figura 25 – Carteira de ações	46
Figura 26 – Dados históricos das ações selecionadas	46
Figura 27 – Gráfico de evolução de cada ação da carteira	47
Figura 28 – Valor simulado da carteira	48
Figura 29 – Gráfico da evolução das ações normalizadas.	49

Figura 30 – Valores normalizados para o índice Ibovespa	49
Figura 31 - Extração de dados do Yahoo Finance	50
Figura 32 – Sinais de compra e venda, respectivamente	51
Figura 33 - Criação das médias móveis	51
Figura 34 - Definição de parâmetros para médias móveis	52
Figura 35 - Definição de regras das médias	52
Figura 36 - Exemplo da representação das médias móveis	53
Figura 37 - Definição do período a ser analisado no gráfico	53
Figura 38 - Definição de parâmetros para o gráfico de médias móveis	54
Figura 39 - Parâmetros de indicação de compra e venda	54
Figura 40 - Exemplo de gráfico de cotações com médias	54
Figura 41 - Parâmetros e resultados da carteira de ações	56
Figura 42 - Evolução individualizada das ações	58
Figura 43 - Comparação da performance: carteira x Ibovespa	58
Figura 44 – Cruzamento de médias móveis para PETR4	59
Figura 45 – Cruzamento de médias móveis para CMIG4.	60
Figura 46 – Cruzamento de médias móveis para GGBR3.	61

LISTA DE TABELAS

Tabela 1 - Parâmetros definidos para os indicadores financeiros	39
Tabela 2 - Ticker das empresas Petrobrás e Gerdau	42
Tabela 3 - Resultados finais dos indicadores fundamentalistas de ações selecionadas.	57
Tabela 4 - Comparação dos resultados finais: carteira x Ibovespa.	57
Tabela 5 - Aproveitamento do método para PETR4	60
Tabela 6 - Aproveitamento do método para CMIG4	61
Tabela 7 - Aproveitamento do método para GGBR3	61

LISTA DE ABREVIATURAS E SIGLAS

B3	Brasil Bolsa Balcão (Bolsa de Valores Brasileira)
BACEN	Banco Central
BDR	<i>Brazilian Depositary Receipt</i> (Certificado de Depósitos de Ações)
BI	<i>Business Intelligence</i>
CNN	Redes Neurais Convolucionais
CVM	Comissão de Valores Mobiliários
DY	<i>Dividend Yield</i>
EBITDA	<i>Earnings Before Interest, Taxes, Depreciation and Amortization</i> (Lucros antes de juros, impostos, depreciação e amortização)
EDA	<i>Exploratory Data Analysis</i> (Análise Exploratoria de Dados)
ETF	<i>Exchange Traded Funds</i> (Fundo de Índice)
ETL	Extract Transform Load (Extração, Transformação, Carregamento)
IA	Inteligência Artificial
IoT	<i>Internet of Things</i> (Internet das Coisas)
IPO	<i>Initial Public Offering</i> (Oferta Pública Inicial)
LAJIDA	Lucros antes de juros, impostos, depreciação e amortização
LPA	Lucro por ação
NOPLAT	<i>Net Operating Profit Less Adjusted Taxes</i> (Lucro Operacional Descontados os Impostos)
P/L	Preço/Lucro
P/VPA	Preço/Valor Patrimonial da Ação
PL	Patrimonio Líquido
RNN	Redes Neurais Recorrentes
ROE	<i>Return On Equity</i> (Retorno Sobre Patrimônio)
ROIC	<i>Return On Invested Capital</i> (Retorno Sobre Capital Investido)
S.A.	Sociedade Anônima
SVM	<i>Support Vector Machine</i> (Máquina de Vetores de Suporte)
VPA	Valor Patrimonial da Ação

SUMÁRIO

1	INTRODUÇÃO	11
2	LEVANTAMENTO BIBLIOGRÁFICO.....	13
2.1	CONCEITUAÇÃO	13
2.1.1	Mercado Financeiro.....	13
2.1.2	Mercado de Ações.....	14
2.1.3	Análise Fundamentalista	14
2.1.4	Indicadores Fundamentalistas	15
2.1.5	Análise Técnica de Ações	20
2.1.6	Indústria 4.0.....	21
2.1.7	Business Intelligence.....	22
2.1.8	Big Data	23
2.1.9	Python	25
2.1.10	Machine Learning.....	28
2.1.11	ETL	31
3	DESENVOLVIMENTO.....	34
3.1	ANÁLISE DE INDICADORES FUNDAMENTALISTAS	34
3.2	ANÁLISE TÉCNICA	42
4	RESULTADOS	56
4.1	RESULTADOS DA ANÁLISE FUNDAMENTALISTA.....	56
4.2	RESULTADOS DA ANÁLISE TÉCNICA	59
5	CONCLUSÃO.....	63
	REFERÊNCIAS.....	65
	APÊNDICE A – Construção da análise fundamentalista.....	68
	APÊNDICE B – Construção da análise técnica	70

1 INTRODUÇÃO

Apesar do mercado financeiro ainda ser um assunto distante de grande parte da população brasileira, têm-se observado uma grande ascensão do número de investidores nos últimos anos. Entre julho de 2021 e junho de 2022, por exemplo, houve a entrada de aproximadamente 1,25 milhão de pessoas na Bolsa de Valores de São Paulo (B3), aumentando cerca de 40% de pessoas na bolsa (MENDES, 2022) e mais recentemente um crescimento de 5,6% no 1º semestre 2023 em relação ao mesmo período de 2022 (FERRARI, 2023). Isso se deve à alguns acontecimentos importantes como por exemplo a crise causada pela pandemia do novo coronavírus, alta da taxa de juros no país, fazendo com que as pessoas procurem novos meios pra rentabilizar seu dinheiro, e também devido à um rápido crescimento de fontes de educação financeira, democratizando cada vez mais o tema. Entretanto, grande parte dos novos investidores não tem um conhecimento suficiente e não estão devidamente embasados para analisar e escolherem boas oportunidades na bolsa, fazendo com que muitas vezes se prendam a aplicações ruins ou falsas promessas de ganhos rápidos.

Com o decorrer dos anos, também têm se observado uma acentuada transformação da tendência do mercado de trabalho como um todo, com mudanças perceptíveis desde processos de produção, desenvolvimento de tecnologia, e até mudança no *Mindset* das pessoas em uma forma geral. Grande parte da contribuição para tal feito deve-se à revolução gerada pela indústria 4.0 onde organizações tiveram de se adaptar e superar o novo contexto de forma ágil e assertiva de forma conectada embasada em tecnologia.

Assim como nos demais setores da economia, o mercado financeiro é um dos mais impactados e beneficiados pela chegada da transformação digital, com as organizações buscando uma melhora cada vez maior na experiência dos seus clientes por meio de uma forte base tecnológica, dando maior velocidade, praticidade e confiança e controle para suas operações principalmente em áreas de cibersegurança, Inteligência Artificial (IA) e ciência de dados, assim como benefícios como eliminação de burocracias, visibilidade e transparência e descentralização muito impulsionada pela computação em nuvem (SACRAMENTO, 2021), sendo cada vez mais importante uma mão de obra especializada afim de manusear ferramentas ainda pouco exploradas.

Com isso, buscando uma melhor compreensão da dinâmica de preços dos produtos financeiros, com o decorrer do tempo foram desenvolvidos diversos padrões grafistas, programas, ferramentas e tecnologias que realizassem cálculos para determinação do quão barato ou caro estaria o produto naquele determinado momento e, conseqüentemente um possível indicador de compra ou venda por parte do investidor.

Outra abordagem muito utilizada para escolha e tomada de decisão com ativos financeiros é a estratégia denominada de análise fundamentalista. Essa por sua vez se trata de uma abordagem para avaliar o valor intrínseco de um ativo financeiro, como ações, com base em uma análise aprofundada dos fundamentos financeiros da empresa. Os fundamentos podem incluir dados como balanços, demonstrações de resultados, fluxo de caixa, métricas de *Valuation*, concorrência de mercado, perspectivas econômicas e muito mais.

Portanto, o presente trabalho visa explorar simultaneamente duas áreas do conhecimento em alta no mercado de trabalho em que, através da tecnologia (linguagens de programação e ferramentas de dados) é possível abstrair dados financeiros fundamentalistas, de demonstrativos complexos de empresas listadas na bolsa de valores, e realizar análises simplificadas para o usuário que queira tomar decisão com seu ativo também através da análise técnica de ações. Para isso, inicialmente o objetivo é a contextualização sobre o mercado financeiro como um todo tendo o cenário do mercado de investimentos no país e posteriormente construção de uma análise completa através de um programa em Python que auxiliará investidores iniciantes a tomar decisões mais embasadas sobre seus recursos.

2 LEVANTAMENTO BIBLIOGRÁFICO

2.1 CONCEITUAÇÃO

Para a elaboração do trabalho, é de suma importância que sejam reunidos todos os conceitos utilizados na experimentação de forma teórica e explicativa. Neste capítulo é apresentada uma breve descrição de conceitos relacionados ao mercado financeiro, assim como vários índices fundamentalistas que são base para análises de *Valuation* de empresas, e também uma explanação de temas sobre tecnologia como a linguagem de programação, Big Data, BI, entre outros.

Com isso é possível definir uma linha de raciocínio ou uma estratégia para justificação dos objetivos propostos inicialmente.

2.1.1 Mercado Financeiro

O termo mercado financeiro traz um conceito amplo devido à sua abrangência e complexidade. Trata-se de um ambiente onde ocorre negociação de produtos e serviços financeiros onde coexistem duas partes que são denominados os agentes superavitários (ou credores) que oferecem o capital em esperando obter algum lucro com a operação, e os agentes deficitários (ou captadores) que captam os recursos dos investidores, e que não se limita somente ao âmbito dos investimentos, abrangendo toda a parte de produtos bancários, seguros, cartões, etc.

Pode-se dizer que é o ambiente responsável pelo fluxo da economia onde são negociados valores mobiliários, ou seja, compra e venda de ações, títulos de renda fixa, fundos de investimentos, entre outros, que são intermediados por instituições financeiras (bancos, bancos de investimentos, corretoras, *Assets*, *fintechs*, etc.) propiciando negociações ou transações mais seguras.

Para estabelecimento de regras e normas entre os integrantes, o mercado financeiro conta com o Banco Central (Bacen), instituição a qual está ligada ao Ministério da Economia, que atua como uma autoridade monetária nacional sendo o principal responsável pelo controle da inflação e taxas de juros do país. A Comissão de Valores Mobiliários (também conhecida como CVM) também atua como uma autoridade, tendo como função fiscalizar, normatizar e disciplinar o mercado financeiro como um todo, garantindo a segurança dos participantes do sistema por meio

de aplicação de leis, supervisão da atuação de profissionais, garantindo a transparência por ambas as partes (REIS, 2020).

2.1.2 Mercado de Ações

A denominação mercado de ações diz respeito à um ambiente de negociações, onde são realizadas operações de compra e venda de ações e demais produtos como ETFs (*Exchange-Traded Fund*), BDRs (*Brazilian Depositary Receipt*) e opções, intermediado pela bolsa de valores, tendo uma elevada importância na economia como um todo, por ser a área responsável por atrair investimentos e fomentar o crescimento de um país. É por meio deste que empresas conseguem captar recursos para financiar seu crescimento através de projetos, novos produtos e serviços.

Quando se trata de empresas listadas em bolsa de valores (B3, no Brasil), são companhias que optaram por abrir seu capital, registrando-se devidamente na Comissão de Valores Mobiliários (CVM) para captar recursos para seus projetos e desenvolvimento, ou seja, há disponibilidade de parte de suas ações que serão negociadas publicamente que são realizadas por meio de um IPO (Oferta Pública Inicial em português). Uma ação representa a menor parcela do capital de uma empresa, que por sua vez sofre variação no preço de acordo com os resultados apresentados pela empresa, assim como o volume de negociação por parte dos investidores. Vale ressaltar que com a compra de pelo menos uma única ação, o investidor também está comprando os direitos e deveres daquela companhia tornando-se um sócio da mesma (INFOMONEY, 2022).

2.1.3 Análise Fundamentalista

A análise fundamentalista no mercado financeiro, é um conceito que descreve o estudo das perspectivas e situação financeira das empresas de capital aberto que estão listadas na bolsa de valores. Diferentemente da análise técnica, onde há o estudo de preço de ativos através de padrões grafistas no decorrer do tempo, a análise fundamentalista ajuda no entendimento das expectativas e do negócio para obtenção de um preço justo e realizar análises de resultados a médio e longo prazo por meio do potencial de crescimento do lucro da empresa no futuro (REIS, 2021).

Tal análise é realizada por meio de indicadores financeiros da empresa e do seu papel (ação) na bolsa de valores, também sendo realizada de forma comparativa setorialmente (entre

empresas do mesmo setor), assim como análise macroeconômica (INFOMONEY, 2022), tendo como resultado final a estimação de um preço justo, ou valor intrínseco para o respectivo ativo.

Apesar de ser a principal técnica utilizada como base de estudo por instituições financeiras como bancos, bancos de investimentos, corretoras, *Asset Management*, etc. na tomada de decisão de compra ou venda dos seus ativos devido à sua abrangência e completude, também pode auxiliar o investidor pessoa física nas suas escolhas, se realizada da forma correta. Portanto, as tomadas de decisão iniciam com a comparação da estimativa de preço justo com o preço atual do ativo.

2.1.4 Indicadores Fundamentalistas

A mensuração de indicadores fundamentalistas tem papel importante na análise de determinada companhia ou setor, e suas ações listadas na bolsa de valores. É através deles que é possível extrair informações da saúde financeira, perspectivas futuras e pontos de atenção, impactando diretamente na tomada de decisão para o respectivo ativo financeiro. Dentre os diversos indicadores existentes para as análises, geralmente estão envolvidos nos cálculos, fatores como cotação momentânea da ação, lucro e patrimônio da empresa, seu número de ações negociadas, dividendos, entre outros (DAMODARAN, 2007).

O primeiro indicador a ser estudado é o Lucro por ação cuja sigla se dá por LPA, trata-se de um dado financeiro que leva em consideração o lucro líquido de determinada empresa pela quantidade de ações emitidas. Este, por sua vez é representado pela equação 1:

$$LPA = \frac{\text{Lucro no período}}{\text{Número total de ações}} \quad (1)$$

Com isso quanto maior o LPA de determinada ação, indica um desempenho maior quando se olha para o lucro. Mas que também é impactada diretamente pelo número de ações que essa empresa tem disponível. Portanto, por utilizar o lucro em sua fórmula, pode-se ter LPA negativo caso a empresa não seja lucrativa no período analisado. Esse é um indicador de alta importância por ser base de cálculo para outros indicadores a seguir.

Assim como todos os outros indicadores, o LPA não deve ser analisado de forma isolada, ou seja, o cálculo deste indicador somente pode trazer uma falsa percepção da situação atual da

empresa. Outro ponto de atenção é se há mudança no número de ações negociadas, podendo também impactar e influenciar na análise.

Tendo em vista o conhecimento do lucro por ação, é possível relacioná-lo com o a cotação atual do ativo, obtendo-se a métrica Preço pelo lucro por ação, também descrito como P/LPA. Tal relação, também conhecido como P/L, diz respeito ao Preço sobre Lucro (por ação), o qual passa a informação da divisão entre o preço ou cotação atual do ativo e o quanto de resultado foi gerado no período (últimos 12 meses). Esse índice, por sua vez, é um dos mais utilizados para avaliar se o preço atual da ação está barato ou caro no momento.

Uma outra forma de interpretar esse indicador é o número de vezes que a empresa está valendo em relação ao seu lucro anual junto com a expectativa de retorno futuro, ou seja, quanto maior a relação, maior expectativa de crescimento do resultado da empresa por necessitar de maior período para o retorno para o investidor. Por outro lado, menores valores para esse indicador (próximo de 1), pode indicar problemas e riscos na companhia que está afastando seus investidores, diminuindo sua perspectiva de retorno futuro, consequentemente diminuindo seu preço.

Já quando se observa o patrimônio Líquido (PL) de uma companhia, fala-se de um indicador contábil que corresponde ao que realmente pertence aos acionistas de uma empresa, ou seja, a real riqueza de uma companhia. Também pode ser entendido como a diferença entre o ativo e o passivo da organização (equação 2), sendo ativos os bens e direitos pertencentes à organização em um determinado período e passivos as obrigações que a empresa possui perante terceiros como dívidas, salários e contas em geral.

$$PL = Ativos - passivos \quad (2)$$

Tendo como base o patrimônio líquido visto anteriormente, é possível também encontrar o Valor Patrimonial da Ação, também conhecido como VPA, que por sua vez é obtido a partir da divisão entre o PL da empresa pelo seu número de ações. Através deste indicador, representado pela expressão matemática 3 abaixo, é possível ter uma métrica de quanto o mercado está disposto a pagar pelo patrimônio líquido da empresa.

$$VPA = \frac{PL}{N^{\circ} \text{ ações}} \quad (3)$$

Com isso, relaciona-se também esse indicador com sua cotação atual, através da métrica de Preço por Valor Patrimonial da ação, resumido pela sigla P/VPA, sendo um dos mais importantes indicadores fundamentalistas por estar relacionado com a estrutura financeira da empresa.

Para este cálculo, um maior valor pode indicar em uma maior disposição por parte do mercado em pagar mais por seu patrimônio líquido. Ao se analisar isoladamente, não recomendado para este ou qualquer outro indicador, um P/VPA menor ou igual a um, tem-se um indicativo que existem problemas precificados pelo mercado, além de ter um indicativo que suas ações estão sendo negociadas abaixo do seu patrimônio líquido. Caso essa relação seja maior do que 1, entende-se que a empresa é vista com maior otimismo e que tem um potencial de crescimento no longo prazo.

Para tratar a questão de receita e lucro de uma companhia, utiliza-se a métrica *Earnings Before Interest, Taxes, Depreciation and Amortization* (EBITDA), que em português significa Lucros Antes de Juros, Impostos, Depreciação e Amortização, também podendo ser encontrado como LAJIDA (abreviação em português).

Trata-se também de um indicador financeiro usado para mensurar a quantidade de recursos que determinada empresa gera advindos apenas das suas atividades principais, desconsiderando-se outras entradas ou saídas de dinheiro como rentabilidades ou descontos de impostos.

Portanto, esse indicador, representado pela equação 4, visa mostrar o desempenho e eficiência de uma organização em comparação com outras do mesmo setor sem que fatores externos interfiram nessa avaliação

$$EBITDA = Lucro Op.Liquido + Depreciações + Amortizações \quad (4)$$

O próximo dado a ser estudado e explicado é o *Return On Equity* (ROE), traduzido como retorno sobre o patrimônio. Trata-se de um dos mais importantes indicadores fundamentalistas por conseguir revelar informações sobre a saúde financeira de uma empresa, por meio da mensuração da capacidade que a companhia como um todo de gerar valor ao negócio e a seus

investidores através dos recursos que a mesma possui inicialmente, ou seja, traz a informação do quanto de lucro é gerado em relação ao capital investido.

A determinação do ROE é realizada pela divisão entre a soma de todo o lucro líquido dos últimos 12 meses por seu patrimônio líquido. A fórmula matemática do ROE é representada pela equação 5.

$$ROE = \left(\frac{\text{Lucro Líquido}}{\text{Patrimônio Líquido}} \right) \times 100 \quad (5)$$

Portanto conclui-se que é um indicador de eficiência de gestão, por mostrar como os recursos da respectiva empresa estão sendo aplicados. Por isso, quanto maior essa relação, significa que a mesma gera mais lucro com determinado patrimônio, ou seja, mais eficiente.

Semelhantemente como visto para o ROE, também é utilizado como métrica para análises o *Return On Invested* (ROIC) Capital, que em português significa retorno sobre o capital investido. Trata-se também de um importante indicador fundamentalista por trazer a informação de quanto dinheiro uma determinada companhia consegue gerar em razão de todo o capital investido. Tal indicador é resumido pela equação 6:

$$ROIC = \left(\frac{NOPLAT}{\text{Capital Investido}} \right) \times 100 \quad (6)$$

Para essa equação considera-se a razão entre o NOPLAT (*Net Operating Profit Less Adjusted Taxes*), traduzindo para o português, lucro operacional descontados os impostos, em relação ao capital investido que é todo o valor alocado pela empresa levando em conta o capital dos acionistas e de terceiros. Com isso se essa relação é grande, significa que a empresa obteve grande retorno, ou lucro em relação ao que foi investido, entretanto, para baixas porcentagens de ROIC, significa que determinada companhia não teve grande retorno com o que foi investido.

Quando se trata de dividendos, não é considerado um indicador fundamentalista, mas é de suma importância o entendimento para os conceitos subsequentes. O dividendo é uma parcela do lucro apurado da empresa no período, que é distribuída aos seus acionistas periodicamente, sendo o valor a ser recebido pelo investidor proporcional à quantidade de ações que este detém.

Vale ressaltar que o valor dessa distribuição é realizado de forma deliberada pela própria empresa como gratificação aos seus acionistas, sem haver algum tipo de obrigatoriedade.

O seu pagamento geralmente tem alguns objetivos principais como atrair novos investidores, conseqüentemente valorizando mais o ativo em questão, e também recompensá-los por ter a posse da determinada ação.

Com isso, o primeiro indicador sobre a distribuição de recursos para seus acionistas ser visto é *Dividend Yield* (DY), rendimento de dividendos em português. Este indicador, por sua vez, tem como objetivo a mensuração da performance da empresa de acordo com o retorno do investimento em uma ação na forma de dividendos. Como visto, a remuneração da carteira de investimentos através de proventos é uma das estratégias largamente utilizadas por investidores, dando a devida importância a este indicador. O *Dividend Yield* traz uma relação dos dividendos pagos nos últimos 12 meses por sua cotação momentânea. Tal expressão é resumida pela equação 7:

$$DY = \frac{\text{Dividendos pagos nos últimos 12 meses}}{\text{cotação da ação}} \quad (7)$$

Portanto empresas que apresentam um elevado DY, optam por distribuir uma maior parcela do seu lucro aos seus acionistas ao invés de reinvesti-lo, ocorrendo geralmente com empresas mais consolidadas e de maior porte.

O indicador *Dividend Payout* tem como objetivo apresentar a parcela de lucro da empresa que será distribuída aos acionistas. Assim como o DY, o dividend payout também é expresso em porcentagem, em uma relação do total de dividendos pagos com o lucro líquido da companhia (equação 8).

$$\text{Dividend Payout} = \left(\frac{\text{Dividendos totais}}{\text{lucro líquido}} \right) \times 100 \quad (8)$$

Vale ressaltar que tanto o *Dividend Yield* quanto o *Dividend Payout* variam entre empresas e também entre setores da economia. Geralmente empresas em um setor mais estável e com balanços sólidos tendem a distribuir uma maior parcela de seu lucro para os acionistas, devido a um menor potencial e perspectiva de crescimento. Por outro lado, empresas com grande

potencial de expansão, ou inseridas em setores mais voláteis e sensíveis à economia, tendem a reinvestir uma maior parcela do lucro líquido, distribuindo menos dividendos.

Portanto, vale ressaltar que, apesar de ser uma métrica muito utilizada no mercado financeiro para escolha de ativos, somente a análise deste dado de forma separada pode não traduzir a realidade financeira de uma companhia. Por exemplo, altos índices de *Dividend Yield* e *Dividend Payout*, podem parecer atrativos para um investidor, entretanto essa alta distribuição de recursos aos acionistas pode significar pouco reinvestimento da companhia em melhorias internas e novos projetos, também resultando em um baixo crescimento no futuro. Assim como uma baixa distribuição de lucros pode significar um alto reinvestimento em projetos para um crescimento elevado no futuro.

2.1.5 Análise Técnica de Ações

A análise técnica de ações, também conhecida como análise gráfica, é largamente utilizada no mercado financeiro para prever movimentos de preços das ações por meio de gráficos (INFOMONEY, 2022). Diferentemente da análise fundamentalista, onde se utiliza dos diversos indicadores financeiros para encontrar um valor intrínseco do ativo de determinada empresa, a análise técnica se preocupa mais com os históricos de preços passados.

Geralmente essa técnica é utilizada como estratégia para quem deseja realizar operações de curto prazo tais como *Day Trade* (operações realizadas dentro de um mesmo dia) ou *Swing Trade* (operações de algumas semanas ou poucos meses). E o objetivo central é identificar a tendência no momento e operar antes que a mesma seja revertida. Existem três tendências principais que um ativo pode ter: de alta, caracterizada por um gráfico crescente, ou seja, máximos e mínimos ascendentes no tempo. A tendência de baixa é quando ocorre o oposto, ou seja, quando os máximos e mínimos são decrescentes no tempo. E a última tendência é a de lateralização, que é caracterizada por uma oscilação dentro de um intervalo definido, porém mais a longo prazo o preço tende a se manter o mesmo.

Assim como a análise fundamentalista, a análise técnica também conta com indicadores tais como suporte e resistência que tem como objetivo uma previsão de até onde um ativo pode chegar, linhas de tendência, também largamente utilizada, representa de forma gráfica a tendência de sentido e direção de um ativo de mercado. O método Estocástico é um cálculo realizado com base na relação entre o preço de fechamento e o intervalo de movimentação no período de

análise, com o objetivo de identificar tendências ou reversões. E por fim, a estratégia que será utilizada neste relatório, também amplamente utilizada no mercado, médias móveis.

As médias móveis simples é formada pelo somatório de um conjunto de valores numéricos que são as cotações das ações em um determinado período, pela quantidade de elementos somados, e com isso, indica o preço médio das ações dentro de um período determinado.

Quanto mais períodos ou dados uma média abranger, menos esta irá oscilar, portanto a estratégia a ser utilizada é o cruzamento de duas médias móveis de diferentes periodicidades indicando um momento de tendência de alta ou baixa no mercado.

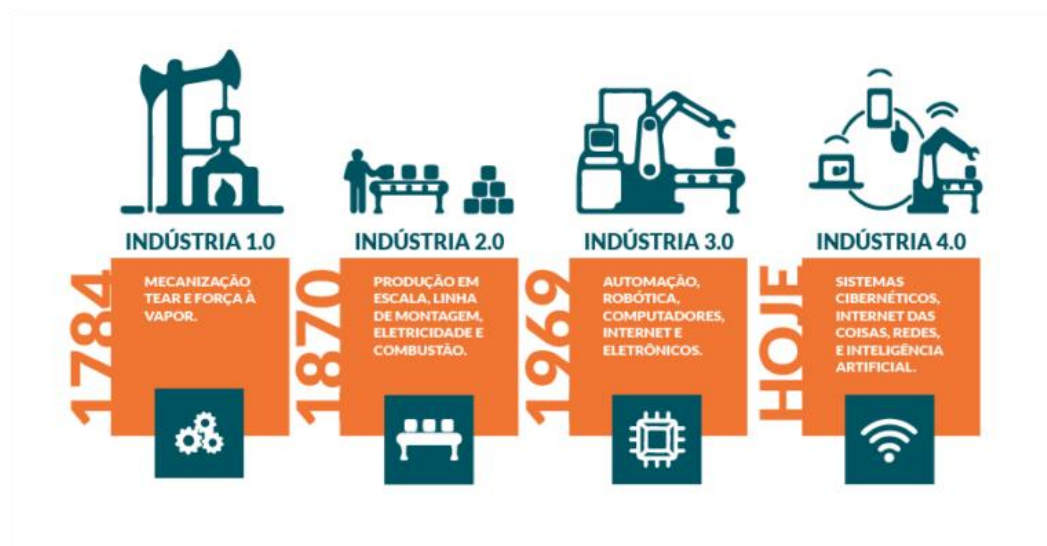
2.1.6 Indústria 4.0

Com um rápido avanço das ferramentas, processos e metodologias de trabalho potencializadas pela internet, foi também acompanhado por um enorme avanço no campo tecnológico nos últimos anos, denominando-se como quarta revolução industrial ou indústria 4.0 quando se diz respeito à automação e tecnologia da informação, entre outras inovações. Tal revolução é marcada pela transformação de tecnologias digitais, as quais tem promovido mudanças relevantes dentro das companhias.

O termo que surgiu em 2011 em Hannover, Alemanha é utilizado para definir o desenvolvimento e a evolução da sociedade destacando as mudanças do modo como as pessoas lidam com a tecnologia diariamente. Também pode ser caracterizada pela fusão de tecnologias digitais e físicas criando sistemas mais inteligentes e conectados (BRAZILLAB, 2019).

Entre seus principais pilares estão: inteligência artificial, Big Data, IoT (Internet das coisas, em português) e *Machine Learning* (Aprendizado de Máquina, em português), temas que serão de extrema importância para o presente trabalho. Vale ressaltar que para todos estes conceitos, independentemente da área de atuação ou aplicação, necessita-se de uma massiva quantidade de dados coletados para análises e tomada de decisão, sendo realizadas tanto por colaboradores, ou máquinas (robôs). Na figura 1 é possível notar a evolução da indústria no decorrer do tempo.

Figura 1 – Evolução da indústria no decorrer do tempo.



Fonte: Silveira (2016).

2.1.7 Business Intelligence

Com a constante evolução no mercado de trabalho, as empresas que desejam se manter competitivas em seus respectivos ramos ou linha de negócios, vêm enfrentando uma revolução tanto em seus processos internos, quanto na sua forma de tomada de decisões. O que tem-se observado, é uma busca constante de uma mudança para a cultura orientada por dados, ou também designada data-driven, ou seja, tomada de decisões com um maior embasamento nos dados ao invés de intuição ou experiências passadas dos executivos.

O termo *Business Intelligence*, traduzindo para o português significa Inteligência de Negócios, e foi criado na década de 1980 pelo Gartner Group (Instituto de Pesquisa e Análise de TI). Trata-se de uma metodologia de que oferece apoio à tomada de decisões de forma inteligente, totalmente pautada nos dados. Para isso o BI engloba uma enorme variedade de ferramentas e tecnologias que permitem as organizações atuarem desde a captação, passando por todo o tratamento e por fim, realizando a análise e dando a informação abstraída para auxiliar os times de negócios ou comerciais nas decisões.

Estão entre os ganhos e vantagens da implementação do *Business Intelligence*: antecipação e solução dos problemas de forma preventiva, otimização de preços, rápidas reações às novas demandas do mercado, e até novas oportunidades de negócio, e estão entre as ferramentas mais utilizadas o Microsoft Power BI, Tableau para visualização de Dashboards e relatórios gerenciais

e ferramentas de Data Warehouse, o qual tem como objetivo centralizar e consolidar grandes quantidades de dados de diversas fontes, fornecendo suporte às demais atividades do BI. A figura 2 ilustra as principais etapas e termos do *Business Intelligence*.

Figura 2 - Etapas do *Business Intelligence*.



Fonte: O que [...] (2022).

Entretanto, vale ressaltar que o BI não se resume somente às ferramentas que cumprem a função de estruturação e análise dos dados, também diz respeito à cultura *data-driven* em todo o seu processo pautado nos dados, incluindo também os analistas e tomadores de decisões nos negócios.

2.1.8 Big Data

Atualmente, com o surgimento de uma infinidade de programas, aplicativos e tecnologias em geral, a todo instante é gerada uma quantidade massiva de dados, em que em sua maioria podem trazer informações importantes para as companhias, gerando uma vantagem competitiva nos negócios se tratado e analisado de forma adequada e eficiente.

Apesar de ser um conceito relativamente novo, as origens de grandes conjuntos de dados começaram a surgir aproximadamente nas décadas de 1960 e 1970, com o surgimento dos

primeiros data centers e dos primeiros bancos de dados relacionais para o mundo de dados. Portanto, com o rápido aumento do volume de dados, foi necessário a criação e aperfeiçoamento de tecnologias para armazenar, manipular e analisa-los.

A principal definição para Big Data é conhecida como os dados que contêm os “5 Vs”, ou seja, conta com uma enorme variedade de dados e tipos de dados (estruturados e não estruturados), com volume expressivo e com maior velocidade do que as fontes primitivas, trazendo uma maior confiança e segurança em suas análises através de sua veracidade, agregando valor aos negócios.

a) Volume: refere-se à grande quantidade de dados. Atualmente com boa parte da população com acesso à internet, redes sociais, smartphones e também através de diversos dispositivos IoT, novas tecnologias industriais, etc. tem-se uma quantidade massiva de dados e informações geradas em um curto espaço de tempo. Estima-se que diariamente são gerados 2,5 quintilhões de bytes, sendo estes de diversas fontes utilizando internet.

b) Velocidade: trata-se de um crescimento da taxa nos quais os dados são extraídos, armazenados e tratados de forma satisfatória. Como visto, com uma quantidade massiva de dados gerados a todo momento, necessita-se de um tratamento eficiente que suporte esse grande volume. Com isso, através de soluções que tragam mais agilidade e escalabilidade nos processos, como por exemplo a crescente utilização da computação em nuvem (*cloud computing*), evita-se que com uma maior demanda de dados impacte na velocidade de execução das atividades.

c) Variedade: trata-se da enorme variedade de tipos de dados disponíveis atualmente, podendo ser estruturados, semiestruturados e não estruturados. Os dados estruturados são aqueles que estavam previamente organizados em estruturas antes da sua extração, geralmente aparecem em forma de linhas e colunas ou agrupados segundo algum critério ou vinculados a categorias específicas. Dados não estruturados não possuem uma estrutura definida, portanto são mais complexos. Com a presença de diversas fontes como redes sociais, smartphones, plataformas na internet, sistemas empresariais, têm-se diferentes tipos de dados em que uma solução de Big Data deve ser capaz de lidar como por exemplo texto, foto, áudio e vídeos.

d) Veracidade: neste aspecto, refere-se ao quanto pode ser assegurado que tais dados podem atender às necessidades do negócio. Trata-se de um dos aspectos mais importantes do Big Data, pois é onde há a distinção do que é confiável, e o que não é, e a separação dos dados que serão relevantes para as etapas posteriores, como nas análises por exemplo.

e) Valor: para esse aspecto, sua importância é o quanto os dados conseguem agregar valor ao negócio ou serviço a ponto destes se tornarem competitivos, através de tendências e padrões no decorrer do tempo com o objetivo de uma tomada de decisão mais embasada e fundamentada, ou algo que trará algum valor como melhoria de processos e resolução de problemas.

A figura 3 destaca os principais pilares do *Big Data*.

Figura 3 - Os 5 V's do Big Data



Fonte: Gutta (2020).

2.1.9 Python

O Python é uma das linguagens de programação mais populares e utilizadas no Brasil e no mundo, tratando-se de uma linguagem em alto nível e *Open-Source*, ou seja, com sintaxe mais próxima da linguagem humana. Sua utilização começou na década de 90 pelo programador holandês Guido Van Rossum, que teve por objetivo a criação de uma linguagem mais legível, simples e versátil comparada às existentes até então.

Dentre os principais usos, estão construção de aplicativos, criação de sites, automatização de tarefas, ciência e análise de dados e também em aplicações para inteligência artificial. Sua popularidade também se deve à gratuidade, simplicidade, acessibilidade e também por ser extensível, ou seja, sendo possível incluir bibliotecas já prontas para determinada aplicação desejada, economizando tempo e esforço para o programador.

Portanto, explica-se a alta utilização do Python no mercado financeiro, onde frequentemente há a necessidade de automação de tarefas executadas por analistas, e a sua capacidade de

manusear e analisar grandes quantidades de dados de forma relativamente simples. A figura 4 mostra o logo da linguagem python utilizado no presente trabalho.

Figura 4 - Logo da Linguagem de programação Python



Fonte: Python [...] (2022).

Dentre as muitas bibliotecas existentes em Python, o presente trabalho se concentra em algumas que foram suficientes para realização do experimento. A primeira e mais conhecida é o Pandas. Trata-se de uma das bibliotecas mais utilizadas no Python para Ciência de Dados, devido à sua flexibilidade e por ter uma abordagem rápida e intuitiva para tratar estruturas robustas de dados estruturados. Através dessa biblioteca é possível fazer limpeza e tratamento dos dados, análise exploratória (EDA), queries em bancos de dados relacionais e até auxílio em atividades de *Machine Learning*.

A facilidade em utilizar essa biblioteca é também devido ao seu manuseamento de dados através de *DataFrames*, os quais são objetos bidimensionais e de tamanho variável com os dados sendo organizados em linhas e colunas tendo formato de uma tabela. A figura 5 mostra um exemplo de *DataFrame* com dados financeiros de empresas da bolsa de valores, resultado do código utilizado no experimento deste trabalho.

Figura 5 - Exemplo de um *DataFrame* no Python utilizando Pandas

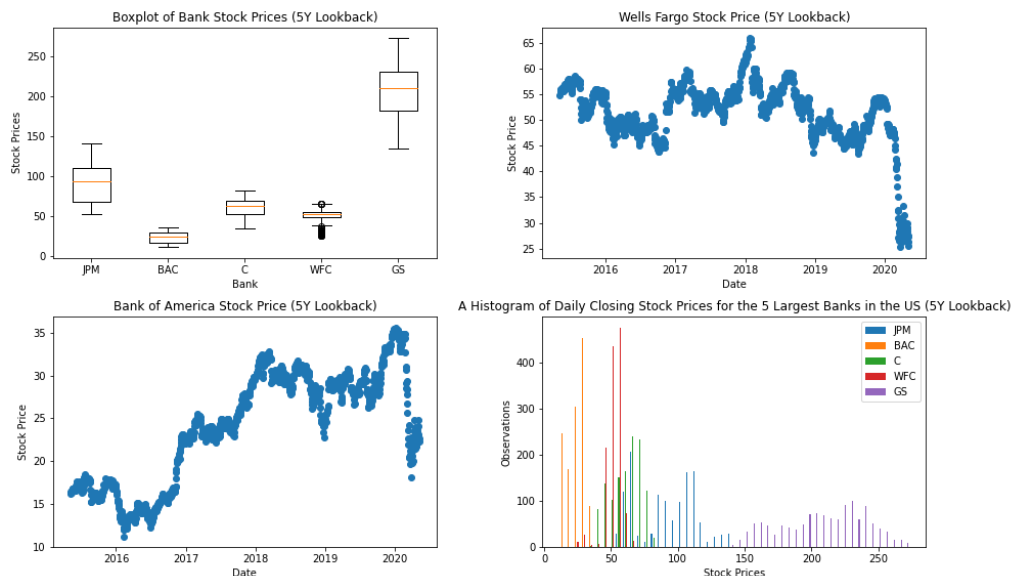
Multiples papel	cotacao	pl	pvp	psr	dy	pa	pcg	pebit	pac1	evebit	evebitda	mrgebit	mrqliq	roic	roe	liq
AALR3	10.41	-4.30	1.42	1.087	0.0000	0.488	-3.83	-726.93	-0.99	-1255.63	18.52	-0.0015	-0.2460	-0.0007	-0.3311	0.55
ABCB3	0.00	0.00	0.00	0.000	0.0000	0.000	0.00	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0000	0.1489	0.00
ABCB4	19.81	5.55	0.83	0.000	0.0609	0.000	0.00	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0000	0.1489	0.00
ABEV3	12.98	14.32	2.40	2.472	0.0587	1.534	1010.62	12.55	-16.22	12.03	8.81	0.1970	0.1779	0.1576	0.1678	1.01
ABYA3	4.91	-214.00	1.76	2.055	0.0000	0.527	1.98	19.96	-2.75	33.67	33.67	0.1029	-0.0096	0.0278	-0.0082	2.09
ACES3	95.27	8.79	2.28	1.693	0.0000	1.438	6.40	7.20	20.77	6.54	6.54	0.2352	0.1927	0.2690	0.2595	2.05
ACES4	94.73	8.74	2.27	1.683	0.0000	1.430	6.37	7.16	20.65	6.50	6.50	0.2352	0.1927	0.2690	0.2595	2.05
ADHM3	1.56	-4.09	-1.29	770.924	0.0000	105.126	-1.43	-5.69	-1.27	-5.69	-5.81	-135.4850	-188.6060	5.2170	0.3147	0.00

Fonte: Autoria Própria

A segunda biblioteca a ser destacada é a NumPy, uma simplificação para *Numerical Python*, também largamente utilizada no campo de ciência de dados. Vale ressaltar sua grande importância por fornecer um grande conjunto de funções e operações que auxiliam em cálculos numéricos que podem ser utilizados em modelos de *Machine Learning*, processamento de imagem e computação gráfica, tarefas matemáticas sendo eficiente em executar tarefas como integração numérica, diferenciação, interpolação e funções incorporadas para álgebra linear.

Por fim, também será tratada a biblioteca Matplotlib, uma extensão do NumPy que é responsável pela visualização de dados no Python, permitindo apresentar insights encontrados na análise de dados de uma forma muito mais visual oferecendo recursos de plotagem para criação de gráficos de 2 ou 3 dimensões em diferentes tipos de relatórios de visualização como histogramas, gráficos de linhas, barras, pizza, dispersão, entre outros. Vale ressaltar sua alta capacidade de funcionar bem com diferentes sistemas operacionais. A figura 6 ilustra diferentes tipos de gráficos apropriados para diferentes aplicações no que tange ao mercado financeiro e análise de ações, gerados com o auxílio da biblioteca Matplotlib.

Figura 6 - Exemplos de gráficos gerados a partir do Matplotlib



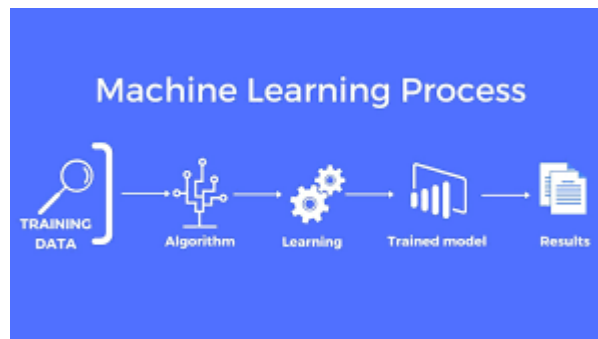
Fonte: McCullum (2020).

2.1.10 Machine Learning

Machine Learning, traduzido como Aprendizado de Máquina, pode ser definido como a capacidade de uma máquina, através de um processo de testes e aprendizagem, aprender determinado padrão ou um processo específico através de dados coletados e tratados e através da sua própria experiência com esses insights já utilizados (DUQUE, 2018), conseguir realizar análises prescritivas, ou seja, realizar previsões e recomendações futuras.

Trata-se de um ramo da Inteligência Artificial (IA) que se concentra no uso dos dados e algoritmos se baseando na forma como os humanos aprendem, incrementando e melhorando cada vez mais sua precisão. São tecnologias cada vez mais utilizadas no mercado de trabalho por conseguirem lidar com quantidades expressivas de dados, extraindo informações valiosas destes, desenhando padrões e por fim capaz de realizar análises prescritivas, ou seja, através do aprendizado dos dados coletados, poder entregar recomendações assertivas de negócios, que seriam realizadas por meio da intuição ou experiências passadas dos colaboradores. A figura 7 tem como objetivo ilustrar os principais pilares e conceitos dentro do *Machine Learning*.

Figura 7 - Etapas do *Machine Learning*



Fonte: Uso [...] (2023).

Entretanto, o *Machine Learning* não traz ganhos somente para o mundo dos negócios, vê-se também uma melhoria em vários aspectos da vida cotidiana das pessoas que são simplificados por meio dessa tecnologia, como por exemplo recomendação de conteúdos nas redes sociais, plataformas de streaming, recomendação de produtos, tradução de textos, entre muitos outros.

Quando aplicada ao mercado financeiro, na análise de ações com *Machine Learning* também é uma abordagem que utiliza algoritmos de aprendizado de máquina para tomar decisões relacionadas a oportunidades de entrada, saída de mercado, e até escolha dos ativos mais promissores no momento pautado em dados. Para isso, o objetivo é extrair informações úteis a partir de dados financeiros, notícias, indicadores econômicos, performances passadas e históricos e outros inputs para prever tendências de mercado e tomar decisões de investimento mais embasadas. Aqui estão algumas etapas típicas envolvidas na análise de ações com *Machine Learning*:

Coleta de dados: A primeira etapa envolve a coleta de dados financeiros, que por sua vez podem ser preços de ações históricos, indicadores fundamentalistas, dados macroeconômicos, notícias e até mesmo dados não financeiros, como sentimentos e tendências nas redes sociais. Os dados são cruciais para treinar e testar os modelos de *Machine Learning*.

Pré-processamento de dados: Posteriormente, após a coleta dos dados, é necessário realizar o pré-processamento para limpar e transformar os dados em um formato adequado para análise. Para isso podem ter etapas de remoção ou adição de valores ausentes, normalização e padronização de dados e a criação de recursos relevantes. Tal etapa se configura como uma das mais importantes de todo o ciclo por prover informações confiáveis para análises nas etapas posteriores.

Escolha de algoritmos: nessa etapa da análise de ações por meio da inteligência artificial, há a seleção dos algoritmos de *Machine Learning* mais apropriados para a resolução do problema desejado. Algoritmos comuns usados na análise de ações incluem regressão linear, árvores de decisão, modelos de séries temporais, redes neurais e algoritmos de aprendizado profundo.

Treinamento do modelo: Com os dados devidamente preparados e os algoritmos selecionados, a próxima etapa se resume no treinamento o modelo com um conjunto de dados históricos de teste, para que ele aprenda a fazer previsões com base nos dados disponíveis.

Avaliação do modelo: tendo por concluído os treinamentos do modelo com os dados históricos, é importante avaliar o desempenho do mesmo usando dados de teste ou validação para verificar o quão precisas estão sendo as previsões. Métricas de avaliação comuns incluem erro médio, precisão, recall e F1-score. Para esta etapa, caso o desempenho ainda não tenha sido o desejado ou não atender às expectativas, realiza-se uma otimização do modelo por meio de ajustes de alguns parâmetros, com experimentação de diferentes algoritmos ou até mesmo considerando a inclusão de mais dados.

Implantação e monitoramento: por fim, uma vez satisfeito com o desempenho do modelo, pode ser realizada a implantação para fazer previsões em tempo real ou tomar decisões de investimento automatizadas. Vale considerar a importância de monitorar o modelo continuamente e fazer ajustes conforme necessário, devido à rápida mudança das condições de mercado.

É fundamental lembrar que a análise de ações com *Machine Learning* não é uma garantia de sucesso nos investimentos. Como citado, o mercado de ações é complexo, dinâmico e influenciado por muitos fatores imprevisíveis que vão além de números ou dados quantitativos. Além disso, modelos de *Machine Learning* estão sujeitos a *overfitting* e a limitações inerentes aos dados usados para treinamento. Portanto, é aconselhável sempre combinar a análise de *Machine Learning* com uma abordagem diversificada com outras técnicas de análise.

Dentre os diversos modelos de *Machine Learning* existentes, cada um com suas características e aplicações, abaixo está uma breve explicação de cada um:

O modelo de Regressão Linear é bastante utilizado para prever um valor numérico com base em uma relação linear entre variáveis. É adequado para problemas de regressão simples. Já o modelo de Regressão Logística é mais utilizado classificação binária, previsões de probabilidades entre duas classes. É amplamente usado em problemas de classificação.

Outro modelo muito conhecido do *Machine Learning* são as Árvores de Decisão, que são modelos que tomam decisões com base em regras hierárquicas e são usados tanto para classificação quanto para regressão. Já o *Random Forest*, pode-se dizer que se trata de uma extensão das árvores de decisão que combina várias árvores para melhorar o desempenho e reduzir o *overfitting*.

O modelo SVM (*Support Vector Machine*) é comumente utilizado para problemas de classificação. Ele tenta encontrar um hiperplano que melhor separa os pontos de dados em classes diferentes.

Quando são abordados modelos de Redes Neurais, tem-se algumas subclassificações que são utilizadas para diferentes aplicações. Por exemplo, redes neurais artificiais são modelos inspirados na estrutura do cérebro humano. Já as redes neurais profundas (*Deep Learning*) são usadas para tarefas complexas, como visão computacional e processamento de linguagem natural. As redes neurais convolucionais (CNNs) são mais indicadas para usos em tarefas de visão computacional, sendo as mais especializadas para esse tipo de aplicação, como reconhecimento de imagem por exemplo. E por fim, as redes neurais recorrentes (RNNs) são projetadas para tarefas envolvendo sequências de dados, como análise de séries temporais ou processamento de linguagem natural.

O modelo Naive Bayes é mais utilizado para classificação e é baseado no teorema de Bayes. É muito eficaz em problemas de processamento de linguagem natural, como filtragem de spam. E por fim, o modelo XGBoost é uma implementação eficaz de *Gradient Boosting* que é frequentemente usada em competições de ciência de dados.

Cada modelo tem suas próprias vantagens e desvantagens, e a escolha do modelo depende da natureza do problema que você está tentando resolver. Em muitos casos, também é benéfico experimentar vários modelos para determinar qual funciona melhor para sua aplicação específica.

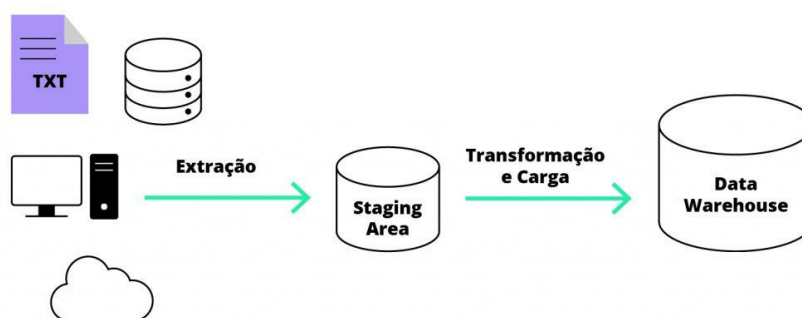
2.1.11 ETL

Como citado anteriormente, o manuseamento dos dados vem se tornando cada vez mais importante para o mercado de trabalho atual, surgindo novos desafios diariamente em como tratá-los e manuseá-los de forma eficiente, auxiliando em grandes tomadas de decisão.

Para isso, para ajudar na gestão de grandes quantidades de dados, em diversos formatos, advindos de diferentes tipos de fontes, é preciso criar etapas ou padrões para sua devida análise.

Para isso surgiu o conceito de ETL, abreviação para *Extract, Transform and Load*, que traduzindo para o português seria extração, transformação e carregamento como é possível observar na figura 8. Trata-se de um processo de manipulação de dados com capacidade de lê-los, transportá-los entre diferentes ambientes.

Figura 8 - Etapas do processo de ETL



Fonte: MJV (2021).

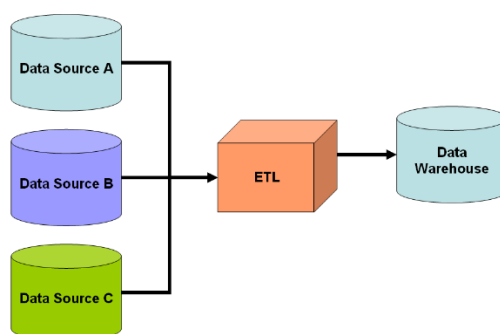
A etapa de extração diz respeito ao transporte desses dados de suas respectivas fontes, podendo estar na forma estruturada ou não estruturada como por exemplo e-mails, bancos de dados, sensores, documentos, entre outros, para um armazenamento de destino, que geralmente se caracteriza por uma área de preparação. Vale ressaltar que comumente essas áreas são temporárias, ou seja, que seu conteúdo pode ser apagado depois da conclusão da etapa de extração.

Com os dados devidamente extraídos de forma bruta, ou seja, formato original, tais dados necessitam de um tratamento ou transformação. Para essa etapa ocorrem validações, autenticações, limpeza, revisão de formato, agregações ou exclusões, para que finalmente se obtenha dados confiáveis para uma futura utilização. Vale ressaltar a grande importância dessa etapa devido ao objetivo de atender requisitos da regra de negócio, viabilizando um dado apropriado para a aplicação desejada.

Por fim, com os dados devidamente tratados e preparados, as ferramentas de ETL realiza o transporte dos dados transformados da área de preparação para o armazenamento de dados de destino, geralmente consistidos por *data warehouses* de forma automatizada, podendo ocorrer de

em tempo real ou por meio de lotes programados quando se trata de uma quantidade massiva de informações. Para essa etapa do processo de ETL podem ocorrer dois tipos de carregamento: o carregamento completo, em que todos os dados da origem são tratados e transportados para data Warehouse, ocorrendo geralmente na primeira carga, e o carregamento incremental, em que há uma carga do delta, ou seja, da diferença do que ainda não foi carregado anteriormente. A figura 9 resume o processo como um todo do ETL.

Figura 9 - Simplificação do fluxo de ETL



Fonte: Santos (2018).

Portanto, o processo de ETL também tem grande importância para o *Business Intelligence*, devido à uma maior confiabilidade, precisão e eficiência dos dados para suas análises.

3 DESENVOLVIMENTO

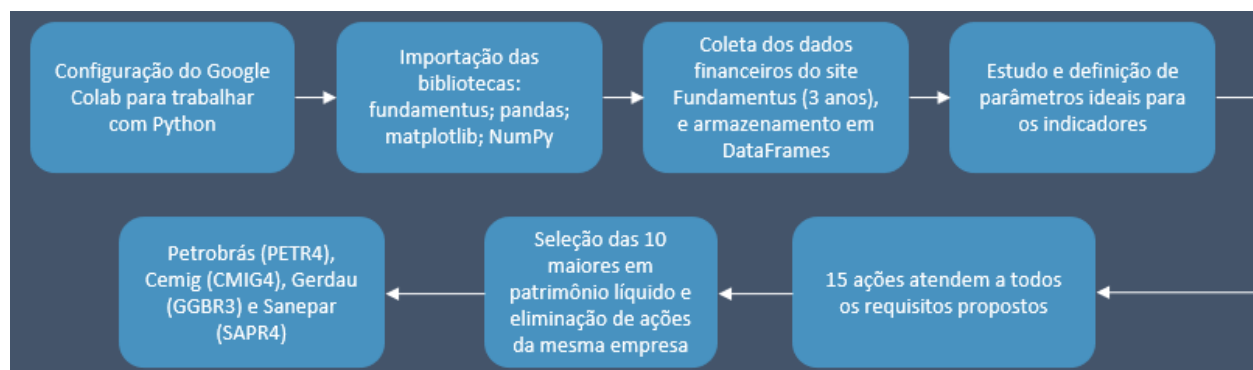
Para o desenvolvimento do presente trabalho, foi determinado a utilização da linguagem de programação Python devido à sua completude no tratamento e análise de dados com auxílio de suas bibliotecas, e sua versatilidade para lidar com diferentes tipos de fontes de dados.

Com isso foi escolhida uma estratégia dividida em duas grandes etapas. Primeiro foram coletados, tratados e analisados os dados fundamentalistas das companhias listadas na bolsa de valores. Posteriormente, com *insights* gerados na primeira etapa, foram utilizadas métricas de análise gráfica de ações para uma análise final.

3.1 ANÁLISE DE INDICADORES FUNDAMENTALISTAS

Para uma análise de indicadores financeiros mais eficaz, foi construído e seguido uma sequência de passos que levassem desde dados brutos em sites de investimentos, até uma resposta abstraída para o usuário final, como mostra o fluxograma representado na figura 10.

Figura 10 - Fluxograma das etapas da análise fundamentalista



Fonte: Autoria Própria.

Como primeira etapa da experimentação para o presente trabalho, priorizou-se a coleta dos dados financeiros das empresas para a análise. Tal passo consistiu em extrair dados financeiros fundamentalistas atualizados das ações listadas da bolsa.

Para isso, primeiramente, foi utilizada a ferramenta de serviço de armazenamento em nuvem Google *Colaboratory*, ou também denominado como Google Colab, a qual permite a criação e execução de códigos em Python diretamente no navegador, não havendo a necessidade de instalação de programas ou *softwares* no computador, necessitando somente ter uma conta ativa no Google.

Posteriormente, realizou-se a importação de algumas bibliotecas do Python essenciais para coleta e tratamento dos dados a serem analisados (ver seção 2.1.8). A primeira biblioteca utilizada foi o Pandas, que permite manuseio de dados e tabelas com maior facilidade, sendo possível realizar análises e manipulações de uma quantidade grande de informações com poucas linhas de código. Também foram utilizadas as bibliotecas NumPy e Matplotlib que permitem uma maior versatilidade e flexibilidade na realização de cálculos matemáticos entre os dados fundamentalistas caso necessário, e também a construção de gráficos ou formas que permitissem uma melhor visualização dos dados e análises realizadas, respectivamente.

Finalmente foi utilizada a biblioteca Fundamentus no Python. Trata-se de uma pequena API (*Application Programming Interface*, ou Interface de programação de aplicações em português), que por sua vez são mecanismos que permitem que dois componentes de *software* se comuniquem através da utilização de um conjunto de definições e protocolos. Vale ressaltar que foi necessária a instalação da biblioteca Fundamentus no Python para utilização no google colab através do comando “!pip install fundamentus” como mostra a figura 11. Com isso também foi necessária a importação dessas bibliotecas no início do código para futuras utilizações no mesmo, como mostra a figura 12.

Figura 11 - Comando de instalação da biblioteca Fundamentus

```
!pip install fundamentus

Collecting fundamentus
  Downloading fundamentus-0.2.0-py3-none-any.whl (13 kB)
Requirement already satisfied: requests>=2.25.1 in /usr/local/lib/python3.10/dist-packages (from fundamentus) (2.31.0)
Collecting requests-cache>=0.5.2 (from fundamentus)
  Downloading requests_cache-1.1.0-py3-none-any.whl (60 kB)
    60.1/60.1 kB 2.0 MB/s eta 0:00:00
Requirement already satisfied: pandas>=1.1.5 in /usr/local/lib/python3.10/dist-packages (from fundamentus) (1.5.3)
Requirement already satisfied: lxml>=4.6.2 in /usr/local/lib/python3.10/dist-packages (from fundamentus) (4.9.3)
Requirement already satisfied: tabulate>=0.8.7 in /usr/local/lib/python3.10/dist-packages (from fundamentus) (0.9.0)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->fundamentus) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->fundamentus) (2023.3.post1)
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->fundamentus) (1.23.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25.1->fundamentus) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25.1->fundamentus) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25.1->fundamentus) (2.0.6)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25.1->fundamentus) (2023.7.22)
Requirement already satisfied: attrs>=21.2 in /usr/local/lib/python3.10/dist-packages (from requests-cache>=0.5.2->fundamentus) (23.1.0)
Collecting cattrs>=22.2 (from requests-cache>=0.5.2->fundamentus)
  Downloading cattrs-23.1.2-py3-none-any.whl (50 kB)
    50.8/50.8 kB 4.4 MB/s eta 0:00:00
Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests-cache>=0.5.2->fundamentus) (3.11.0)
Collecting url-normalize>=1.4 (from requests-cache>=0.5.2->fundamentus)
  Downloading url_normalize-1.4.3-py2.py3-none-any.whl (6.8 kB)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from cattrs>=22.2->requests-cache>=0.5.2->fundamentus) (1.1.3)
Requirement already satisfied: typing_extensions>=4.1.0 in /usr/local/lib/python3.10/dist-packages (from cattrs>=22.2->requests-cache>=0.5.2->fundamentus) (4.5.0)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas>=1.1.5->fundamentus) (1.16.0)
Installing collected packages: url-normalize, cattrs, requests-cache, fundamentus
Successfully installed cattrs-23.1.2 fundamentus-0.2.0 requests-cache-1.1.0 url-normalize-1.4.3
```

Fonte: Autoria Própria.

Figura 12 - Importação das bibliotecas no código

```
[ ] import fundamentus
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

Fonte: Autoria Própria.

A presente biblioteca tem como função principal o mapeamento e extração de todas as ações da bolsa de valores brasileira, assim como seus principais indicadores fundamentalistas atualizados do site Fundamentus. Vale ressaltar que na figura 13, página do site citado, estão presentes os indicadores de uma só empresa, Petrobrás (cujo ticker é resumido pela sigla PETR4) e que foi utilizado para exemplificar a visão do site.

Figura 13 - Dados financeiros da Petrobrás no site Fundamentus

PÁGINA INICIAL	INVESTIMENTO CONSCIENTE	MAIS OPÇÕES ▼	ENTRE EM CONTATO	FUNDAMENTUS MOBILE	DETALHES	GRÁFICO
Papel	PETR4		Cotação		36,28	
Tipo	PN		Data últ cot		13/10/2023	
Empresa	PETROBRAS PN		Min 52 sem		17,40	
Sector	Petróleo, Gás e Biocombustíveis		Max 52 sem		36,28	
Subsetor	Exploração, Refino e Distribuição		Vol \$ méd (2m)		1.593.820.000	
Valor de mercado	473.254.000.000		Últ balanço processado		30/06/2023	
Valor da firma	691.644.000.000		Nro. Ações		13.044.500.000	
Oscilações		Indicadores fundamentalistas				
Dia	3,30%	P/L	3,03	LPA	11,99	
Mês	4,73%	P/VP	1,28	VPA	28,42	
30 dias	8,20%	P/EBIT	1,81	Marg. Bruta	50,6%	
12 meses	54,50%	PSR	0,81	Marg. EBIT	45,0%	
2023	82,75%	P/Ativos	0,48	Marg. Líquida	27,0%	
2022	46,00%	P/Cap. Giro	-32,01	EBIT / Ativo	26,4%	
2021	22,76%	P/Ativ Circ Liq	-0,98	ROIC	29,0%	
2020	-6,10%	Div. Yield	25,5%	ROE	42,2%	
2019	36,87%	EV / EBITDA	2,06	Liquidez Corr	0,90	
2018	41,83%	EV / EBIT	2,64	Div Br / Patrim	0,75	
		Cres. Rec (5a)	19,4%	Giro Ativos	0,59	
Dados Balanço Patrimonial						
Ativo	990.459.000.000		Div. Bruta		279.375.000.000	
Disponibilidades	60.985.000.000		Div. Líquida		218.390.000.000	
Ativo Circulante	136.816.000.000		Patrim. Lq		370.683.000.000	
Dados demonstrativos de resultados						
Últimos 12 meses			Últimos 3 meses			
Receita Líquida	581.563.000.000		Receita Líquida		113.840.000.000	
EBIT	261.516.000.000		EBIT		49.813.000.000	
Lucro Líquido	156.375.000.000		Lucro Líquido		28.782.000.000	

Fonte: Autoria Própria.

Portanto, os dados trazidos através da API Fundamentus são de diversos indicadores financeiros que podem ser utilizados para diferentes análises de sua ação ou saúde financeira da empresa. Os indicadores trazidos do site são os mostrados na figura 14:

Figura 14 - Lista com as colunas extraídas do site Fundamentus

```
df = fundamentus.get_resultado()
display(df.columns)
```

Index(['cotacao', 'pl', 'pvp', 'psr', 'dy', 'pa', 'pcg', 'pebit', 'pacl',
'evebit', 'evebitda', 'mrgebit', 'mrgliq', 'roic', 'roe', 'liqc',
'liq2m', 'patrliq', 'divbpatr', 'c5y'],
dtype='object', name='Multiples')

Fonte: Autoria Própria.

Como observado na figura 14, os dados da lista representam os indicadores que estão dispostos em colunas na figura 15. Para isso, foi criado um DataFrame através da biblioteca pandas, e este foi exibido de forma completa, listando 967 empresas ao todo com 20 dados financeiros.

São eles na ordem mostrada na : cotação (preço) da ação no dia, pl (preço da ação pelo lucro da ação), pvp (preço da ação pelo seu valor patrimonial), PSR (*Price Sales Ratio*) que é seu preço dividido pela receita líquida por ação, dy (dividend yield), pa, pcg (preço da ação por seu capital de giro), pemit (preço pelo seu EBIT), pacl (preço por seu ativo circulante líquido), evebit (*Enterprise Value* pelo EBIT da empresa), evebitda (*Enterprise Value* pelo EBITDA), mrgebit (EBIT dividido por sua receita líquida), mrgliq (margem líquida), roic, roe, liqc (liquidez corrente), liq2m (volume médio de negociação nos últimos 2 meses), patrliq (patrimônio líquido da empresa), divpatr (dívida bruta da companhia por seu patrimônio) e por fim c5y (crescimento de receita nos últimos 5 anos).

Figura 15 - Lista completa dos indicadores financeiros da bolsa de valores

display(df)

	Multiples	cotacao	pl	pvp	psr	dy	pa	pcg	pebit	pacl	evebit	evebitda	mrgbit	mrgliq	roic	roe	liqc	liq2m	patrlq	divbpatr	c5y
papel																					
AALR3	10.41	-4.30	1.42	1.087	0.0000	0.488		-3.83	-726.93	-0.99	-1255.63	18.52	-0.0015	-0.2460	-0.0007	-0.3311	0.55	2989320.0	8.653780e+08	1.15	0.0306
ABCB3	0.00	0.00	0.00	0.000	0.0000	0.000		0.00	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0000	0.1489	0.00	0.0	5.420810e+09	0.00	0.1728
ABCB4	19.81	5.55	0.83	0.000	0.0609	0.000		0.00	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0000	0.1489	0.00	14359700.0	5.420810e+09	0.00	0.1728
ABEV3	12.98	14.32	2.40	2.472	0.0587	1.534	1010.62	12.00	12.55	-16.22	12.03	8.81	0.1970	0.1779	0.1576	0.1678	1.01	359179000.0	8.507460e+10	0.05	0.1416
ABYA3	4.91	-214.80	1.76	2.055	0.0000	0.527		1.98	19.96	-2.75	33.67	33.67	0.1029	-0.0096	0.0278	-0.0082	2.09	0.0	2.920600e+08	1.31	0.1641
...	...	...	...	...	...	...		...	...	...	...	...	...	...	...	...	...	...	...	...	...
WMMM4	39.32	12.63	2.19	0.646	0.0279	1.516		4.12	10.70	4.89	10.68	10.36	0.0603	0.0511	0.1763	0.1735	2.48	105522.0	6.533200e+08	0.24	0.2526
WMBY3	25.39	-19.30	2.87	0.836	0.0000	0.182		1.20	8.62	-1.50	25.02	25.02	0.0970	-0.0705	0.0239	-0.1486	1.44	0.0	2.124390e+08	6.51	-0.1448
WSON33	67.00	8.07	0.98	1.067	0.0000	0.400	13.41	2.42	-0.89	5.08	5.08	0.4414	0.1358	0.1836	0.1217	1.26	0.0	2.148530e+09	1.21	0.0523	
YDUQ3	19.62	55.04	1.93	1.248	0.0140	0.624		6.57	8.10	-1.52	13.79	6.94	0.1542	0.0233	0.0899	0.0351	1.56	121096000.0	3.138850e+09	1.73	0.0849
ZAMP3	6.00	-14.82	1.21	0.437	0.0000	0.411		12.39	19.44	-0.93	28.08	4.69	0.0225	-0.0295	0.0254	-0.0815	1.18	23092700.0	1.367230e+09	0.85	0.1038

967 rows x 20 columns

Fonte: Autoria Própria.

Entretanto, para utilização de todos esses indicadores financeiros necessitaria de técnicas avançadas de *Valuation*, ou seja, avaliação de empresas, com um objetivo de chegar em um valor justo para suas ações. Tais técnicas, que são possíveis através da utilização de quantidades massivas de indicadores e de modelos avançados de análises, são geralmente realizadas pelas áreas de *Research* de bancos de investimentos ou corretoras que trabalham continuamente a fim de identificar oportunidades de entrada ou saída no mercado, demandando enormes esforços, conhecimento e tempo.

Vale esclarecer também, que para uma análise bem feita de determinada empresa ou ação na bolsa não considera-se apenas dados quantitativos tais como os indicadores fundamentalistas supracitados, mas também vale um estudo de dados qualitativos tais como governança da empresa, histórico de corrupção, envolvimento direto de políticos no comando, principais riscos da empresa e do setor, seus sócios majoritários, entre muitos outros fatores, que podem influenciar diretamente na sua confiabilidade de investimento.

Contudo, como a proposta deste trabalho é dar uma introdução à conceitos financeiros e tecnologia, foi decidido trabalhar avaliando 7 dos principais indicadores financeiros das companhias sendo possível tirar *insights* de sua saúde financeira, geração de lucro e rentabilidade para seus acionistas.

Os indicadores analisados são:

- P/VP – Preço da ação por seu valor patrimonial
- ROE – Retorno sobre o patrimônio líquido
- P/L – Preço da ação por seu lucro (por ação)
- DY – *Dividend Yield*
- Crescimento de receita (últimos 5 anos)
- Div Br/Patr. – Dívida bruta por seu patrimônio líquido
- ROIC – Retorno sobre capital investido

Para cada indicador citado, espera-se uma faixa de valores aproximados para que sejam minimamente aceitos. Vale ressaltar que esses valores foram escolhidos a partir de um consenso que analistas entendem que sejam bons. Entretanto, trata-se de uma aproximação baseada em dados históricos, portanto não devem ser considerados verdades absolutas para estes valores. Outro ponto relevante, é que indicadores financeiros devem ser analisados de forma conjunta com outros fatores externos e indicadores, com isso, apenas uma métrica não consegue resumir a saúde financeira de uma empresa assertivamente.

Dito isso, foram utilizados como métrica os seguintes valores expostos na tabela 1.

Tabela 1 – Parâmetros definidos para os indicadores financeiros.

Indicador	Valores Propostos
P/VP	< 1,5
ROE	> 0,15
P/L	< 10
DY	> 0,07
C. Receita	> 0
Div / Patrim	< 0,9
ROIC	> 0,1

Fonte: Autoria Própria.

A proposta da utilização dos valores da tabela 1, foi realizada baseado em histórico de análises de indicadores de empresas, pertencendo à faixas de valores que são aceitáveis pelo mercado financeiro.

- Como visto na seção 2.1.4, para o indicador P/VP, a escolha de valores menores que 1.5 significa que o preço do ativo está sendo negociado a um valor menor que 50% do acima de seu valor patrimonial. Valores muito elevados podem significar uma elevada valorização do preço da ação, não se destacando como uma boa oportunidade de investimento.
- Para os indicadores de eficiência ROE e ROIC, ou seja, que medem a capacidade de geração de lucro líquido a partir de seu patrimônio e valores investidos, respectivamente, valores acima de 10% indicam um coeficiente eficaz de geração de retorno para o investidor.
- A relação entre preço e lucro por ação (P/L), que foi determinada-se um valor ideal menor que 10 como visto na tabela 1, traduz a quantidade de tempo que o investidor terá lucro com a compra de determinado ativo, portanto não é desejável um alto valor significando um alto preço a ser pago pela quantidade de lucro gerada naquela ação.
- Para o indicador de crescimento de receita foi determinado um valor maior que zero por desejar empresas que tiveram crescimento positivo nos últimos anos.
- Para o próximo indicador analisado, a estratégia adotada é que as ações da carteira se limitem em 90% na relação entre dívida e patrimônio líquido, ou seja, a empresa conta com um caixa positivo apesar das dívidas contraídas.
- Por fim, a escolha de valores superiores a 0.07 para o parâmetro de *Dividend Yield*, se traduz em um retorno de pelo menos 7% além da valorização da ação para o investidor.

Portanto, tais valores foram definidos em Python através do filtro composto mostrado na figura 16 no Google Colab.

Figura 16 - Definição dos filtros dos indicadores financeiros

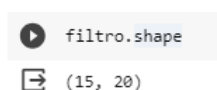
```
filtro = df[(df.pvp < 1.5) & (df.roe > 0.15) & (df.pl < 10) & (df.dy > 0.07) & (df.c5y > 0) & (df.divbpatr < 0.9) & (df.roic > 0.1)]
filtro.sort_values('patrlq', ascending = False, inplace=True)
display(filtro)
```

Fonte: Autoria Própria.

Com isso, como mostra o código da figura 16, das ações que atenderam aos requisitos propostos no trabalho, foram reordenadas de acordo com o seu patrimônio líquido de forma decrescente a fim de escolher quatro empresas para a carteira de investimentos. Vale ressaltar que

essa reordenação poderia ter sido feita com qualquer outro indicador dependendo do objetivo a ser alcançado. Tal estratégia foi adotada priorizando companhias de maior expressão, consequentemente, mais sólidas. Portanto, através do comando ‘.shape()’ como mostra a figura 17, somente 15 ações da bolsa (figura 18) atenderam a todos os requisitos propostos, ou seja, a partir da estratégia escolhida, são os ativos que se enquadram em todas as regras simultaneamente.

Figura 17 - comando utilizado para mostrar quantidade de dados



Fonte: Autoria Própria.

Figura 18 – Ativos que cumpriram os primeiros requisitos de seleção

Multiples	cotacao	pl	pvp	psr	dy	pa	pcg	pebit	pacl	evebit	evebitda	mrgebit	mrqliq	roic	roe	liqc	liq2m	patrliq	divbpatr	c5y
papel																				
PETR3	38.35	3.20	1.35	0.860	0.2412	0.505	-33.84	1.91	-1.04	2.75	2.14	0.4497	0.2699	0.2895	0.4219	0.90	4.747960e+08	3.706830e+11	0.75	0.1942
PETR4	35.35	2.95	1.24	0.793	0.2617	0.466	-31.19	1.76	-0.96	2.60	2.02	0.4497	0.2699	0.2895	0.4219	0.90	1.709210e+09	3.706830e+11	0.75	0.1942
GGBR3	20.27	3.73	0.73	0.467	0.1596	0.483	1.78	2.90	6.27	3.43	2.75	0.1613	0.1259	0.1939	0.1952	2.94	1.073140e+06	4.897690e+10	0.22	0.2138
GGBR4	21.76	4.00	0.78	0.501	0.1487	0.518	1.91	3.11	6.73	3.64	2.92	0.1613	0.1259	0.1939	0.1952	2.94	2.116460e+08	4.897690e+10	0.22	0.2138
CMIG4	11.91	3.96	1.11	0.731	0.0928	0.481	7.68	3.32	-1.51	4.34	3.77	0.2202	0.1849	0.1635	0.2807	1.34	1.079040e+08	2.360910e+10	0.50	0.1180
GOAU3	10.08	3.26	0.60	0.137	0.1151	0.139	0.50	0.85	1.56	1.30	1.04	0.1611	0.1261	0.1935	0.1839	3.03	5.574870e+05	1.740170e+10	0.61	0.2138
GOAU4	10.21	3.30	0.61	0.138	0.1136	0.141	0.50	0.86	1.58	1.31	1.05	0.1611	0.1261	0.1935	0.1839	3.03	7.906770e+07	1.740170e+10	0.61	0.2138
SAPR11	22.82	5.04	0.76	1.169	0.0803	0.390	8.31	2.80	-1.15	4.60	3.89	0.4183	0.2319	0.1520	0.1503	1.47	2.966700e+07	9.100570e+09	0.62	0.0745
SAPR3	4.45	4.92	0.74	1.140	0.0763	0.380	8.10	2.73	-1.12	4.53	3.83	0.4183	0.2319	0.1520	0.1503	1.47	1.224820e+06	9.100570e+09	0.62	0.0745
SAPR4	4.58	5.06	0.76	1.174	0.0815	0.391	8.34	2.81	-1.15	4.61	3.89	0.4183	0.2319	0.1520	0.1503	1.47	8.517270e+06	9.100570e+09	0.62	0.0745
EQMA3B	26.50	5.33	1.15	0.833	0.1053	0.499	7.11	3.55	-1.87	4.87	4.08	0.2346	0.1562	0.1696	0.2156	1.31	5.165560e+03	3.783950e+09	0.69	0.1066
POMO3	4.24	5.77	1.24	0.635	0.0882	0.551	1.80	5.93	18.54	7.51	6.26	0.1071	0.1089	0.1253	0.2141	2.13	1.948480e+06	3.250220e+09	0.74	0.0751
ROMI3	11.57	4.66	0.93	0.683	0.0812	0.451	1.28	4.45	4.20	6.74	5.45	0.1536	0.1472	0.1221	0.1989	2.31	6.917540e+06	1.107180e+09	0.74	0.2180
MTSA4	44.98	4.18	0.95	0.544	0.0746	0.722	1.29	3.53	1.35	2.69	2.45	0.1541	0.1302	0.2888	0.2264	3.60	1.881030e+05	4.175870e+08	0.09	0.3160
TRPN3	0.71	0.44	0.54	0.263	1.0474	0.399	0.59	0.36	0.59	-0.35	-0.35	0.7228	0.5929	5.2301	1.2235	3.61	0.000000e+00	5.935800e+07	0.00	0.0577

Fonte: Autoria Própria.

Com isso, com as manipulações realizadas, e com o objetivo de selecionar as quatro empresas com melhores indicadores levando em consideração os seus indicadores fundamentalistas, observa-se que na lista, houveram mais de um ativo pertencente à mesma empresa para alguns casos (ações ordinárias e ações preferenciais), como mostrado na tabela 2:

Tabela 2 – Ticker das empresas Petrobrás e Gerdau.

PETR3	Petrobrás
PETR4	
GGBR3	Gerdau S.A
GGBR4	
GOAU3	Gerdau (Metalúrgica)
GOAU4	

Fonte: Autoria Própria.

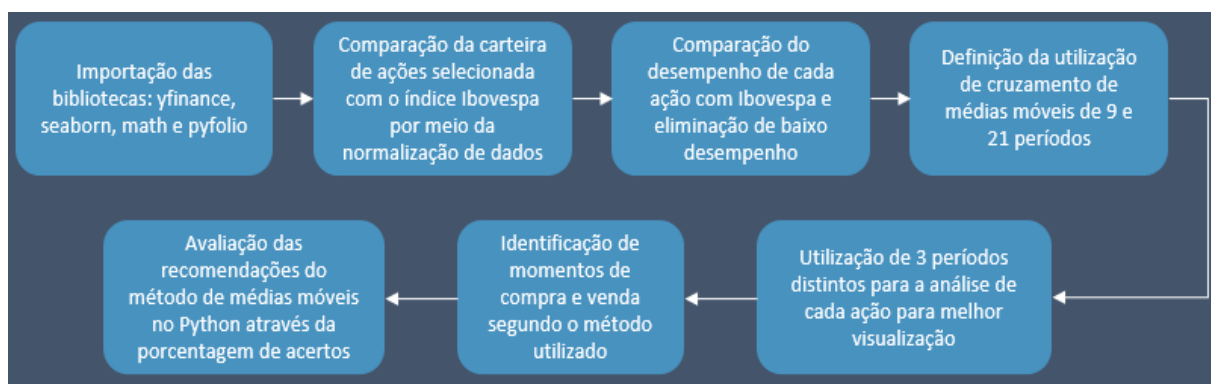
Entretanto, como o objetivo é avaliar as melhores empresas e organizações para um investimento diversificado, houve a escolha de apenas um ativo por companhia, baseado em seu histórico de pagamento de dividendos.

3.2 ANÁLISE TÉCNICA

A análise gráfica também se destaca pela sua importância devido ao seu estudo de cotações históricas, correlações com outros ativos e *benchmark*, entre outras métricas já explicadas anteriormente no presente trabalho, abstraindo uma visão de momento de compra ou venda de determinado ativo.

O experimento utiliza também dessa análise como um ajuste fino, ou seja, inicialmente foi realizada a análise fundamentalista e com isso foram selecionados os ativos. A partir dessa escolha, observou-se aspectos relacionados à dados históricos de cotação dos ativos. A figura 19 tem o objetivo de sintetizar todos os passos realizados no experimento da análise técnica.

Figura 19 - Fluxograma das etapas da análise técnica



Fonte: Autoria Própria.

Para isso foram necessárias novas bibliotecas no Python para que atendessem os requisitos propostos nesta etapa. Como destacado anteriormente, a utilização da biblioteca Fundamentus foi de extrema importância para coleta dos dados financeiros das empresas, contudo somente trazia a informação momentânea de cotação. Como para esta etapa serão analisados aspectos de preço, foi necessária instalação e importação da biblioteca yfinance, como mostra a figura 20, que tem como função a extração de dados de cotação histórica dos ativos da bolsa de valores.

Figura 20 – Instalação da biblioteca yfinance

```
!pip install yfinance --upgrade --no-cache-dir
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-packages (0.2.31)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.5.3)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.23.5)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.31.0)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.10/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.9.3)
Requirement already satisfied: appdirs>=1.4.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.4.4)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2023.3.post1)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.3.8)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.10/dist-packages (from yfinance) (3.17.0)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.11.2)
Requirement already satisfied: html5lib>=1.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (2.5)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.3.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2023.7.22)
```

Fonte: Autoria Própria.

Para as devidas análises também foi preciso a importação da biblioteca seaborn que tem como função a plotagem e visualização de dados para realização de análises. A figura 21 destaca a importação das bibliotecas para o experimento.

Figura 21 – importação das bibliotecas necessárias no Python

```
[ ] #Importando as bibliotecas
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader.data as web
import yfinance as yf
import seaborn as sns

yf.pdr_override()
```

```
[ ] !pip install pyfolio
import pyfolio as pf
```

Fonte: Autoria Própria.

A figura 22 mostra a parte do código do Python que coleta os dados históricos do Ibovespa, principal índice da Bolsa de Valores no Brasil, sendo este, composto pela média de desempenho de toda a carteira de ações listadas.

Figura 22 – Extração dos dados históricos do índice Ibovespa

```
[5] ibov = web.get_data_yahoo('^BVSP')
[*****100%*****] 1 of 1 completed
```

Fonte: Autoria Própria.

Com isso, tem-se disponível para utilização a base completa com as cotações históricas do índice ibovespa, adicionalmente, com comando `ibov.tail()` é possível trazer os dados mais recentes para a análise, como mostra a figura 23.

Figura 23 – Demonstração das últimas cotações obtidas do Ibovespa

`ibov.tail()`

	Open	High	Low	Close	Adj Close	Volume
Date						
2023-10-18	115907.0	115907.0	113952.0	114060.0	114060.0	15492200
2023-10-19	114059.0	115063.0	113768.0	114004.0	114004.0	11497600
2023-10-20	113996.0	114090.0	112533.0	113155.0	113155.0	0
2023-10-23	113145.0	113680.0	112164.0	112785.0	112785.0	10701800
2023-10-24	112814.0	114249.0	112814.0	113762.0	113762.0	10711800

Fonte: Autoria Própria.

Os dados coletados nessa etapa e representados na imagem 23 são:

- Data (*Date*)
- Cotação da ação de abertura no dia (*Open*)
- Cotação máxima da ação no dia (*High*)
- Cotação mínima da ação no dia (*Low*)
- Preço de fechamento no dia (*Close*)
- Preço de fechamento ajustado (*Adjusted Close*)
- Volume negociado no dia (*Volume*)

Como o Ibovespa se trata de um índice, os valores representados na figura 23 são quantificados por pontos, ou seja, uma unidade que representaria o real, porém comumente tratado como pontos. Essas mesmas métricas serão utilizadas para as carteiras de ações, mas neste caso, em unidade monetária (real).

Com os valores trazidos em forma estruturada na tabela, através da biblioteca matplotlib é possível plotar o gráfico de desempenho nesse período tornando-a mais fácil de ser visualizada como mostra o gráfico da figura 24.

Figura 24 – cotação histórica do índice Ibovespa através da biblioteca matplotlib



Fonte: Autoria Própria.

Portanto, a próxima etapa é a realização do tratamento dos dados da carteira de ações selecionadas na análise fundamentalista (seção 3.1) e também os dados da Ibovespa para o mesmo período a fim de que se obtenha insumos pra compará-las como mostra a figura 25. Para esta etapa, foi utilizada uma lista contendo os quatro *Tickers* das ações a serem analisadas, e por meio do comando `get_data_yahoo`, foram trazidos os dados de preço ajustado de fechamento (*Adjusted Close*, ou *Adj Close*) dos últimos 3 anos.

Figura 25 – carteira de ações

```
[22] # simulando carteira de ações
empresas = ['PETR4.SA', 'GGBR3.SA', 'CMIG4.SA', 'SAPR4.SA']
carteira = web.get_data_yahoo(empresas, period= '3y')['Adj Close']
ibov = web.get_data_yahoo('^BVSP', period= '3y')['Adj Close']

[*****100%*****] 4 of 4 completed
[*****100%*****] 1 of 1 completed
```

Fonte: Autoria Própria.

Com isso, como é possível observar na figura 26, os dados foram extraídos do site yahoo e apresentados no google colab por meio de um *DataFrame*. Os dados são trazidos de forma tabular contando com 747 linhas, ou seja, quantidade de dias que a bolsa de valores esteve aberta a negociações nesses três anos.

Figura 26 – Dados históricos das ações selecionadas

```
display(carteira)
```

	CMIG4.SA	GGBR3.SA	PETR4.SA	SAPR4.SA
Date				
2020-10-27	5.261428	12.906155	7.598260	4.202626
2020-10-28	5.124131	12.213568	7.135792	4.023097
2020-10-29	5.094710	12.380056	7.372759	3.990455
2020-10-30	5.026062	12.053740	7.238987	3.884370
2020-11-03	5.065289	12.646438	7.510353	3.900690
...	...	...	...	...
2023-10-23	11.910000	20.270000	35.349998	4.580000
2023-10-24	11.930000	20.139999	35.880001	4.640000
2023-10-25	11.850000	19.950001	36.070000	4.590000
2023-10-26	12.160000	20.290001	35.700001	4.600000
2023-10-27	11.930000	20.180000	35.439999	4.560000

747 rows × 4 columns

Fonte: Autoria Própria.

Portanto, por meio da biblioteca seaborn (sns), tem-se a evolução da rentabilidade das ações da carteira em função do tempo, como pode ser visto na figura 27. Vale citar que para este caso o gráfico considera as curvas de acordo com o seu real valor monetário.

Figura 27 – Gráfico de evolução de cada ação da carteira



Fonte: Autoria Própria.

Afim de que se elimine as distorções, impactando em uma análise mais assertiva, foi realizada uma normalização dos preços dos ativos, ou seja, foi feito uma divisão do valor atual da ação pelo valor do início da análise (dia 27 de outubro de 2020), assim é possível ter uma melhor visão de rentabilidade dos ativos no tempo.

Adicionalmente, com o objetivo de trazer a análise para um cenário mais realista, a carteira normalizada com todos os ativos escolhidos foi multiplicada por 1000, simulando o aporte de R\$ 1000,00 em cada ação no período inicial, como também pode ser visto na tabela expressa na figura 28. Com isso tem-se uma melhor noção da evolução de patrimônio em função do tempo. Para esta etapa, esse valor foi escolhido aleatoriamente, podendo ser qualquer outro valor desejado.

Finalmente, o último comando em Python demonstrado também na figura 28 nomeada como “saldo”, foi utilizado com objetivo de expressar o patrimônio total contido na carteira de ações naquele determinado período, ou seja, a soma de todas as ações no respectivo dia.

Figura 28 – Valor simulado da carteira

```

carteira_normalizada = (carteira/carteira.iloc[0])*1000
carteira_normalizada
carteira_normalizada['saldo'] = carteira_normalizada.sum(axis =1)

```

	CMIG4.SA	GGBR3.SA	PETR4.SA	SAPR4.SA	saldo
Date					
2020-10-27	1000.000000	1000.000000	1000.000000	1000.000000	4000.000000
2020-10-28	973.905054	946.336696	939.134993	957.281661	3816.658404
2020-10-29	968.313163	959.236638	970.322013	949.514639	3847.386453
2020-10-30	955.265781	933.952823	952.716417	924.272072	3766.207092
2020-11-03	962.721454	979.876501	988.430662	928.155413	3859.184030
...	...	...	...	...	...
2023-10-23	2263.644037	1570.568542	4652.380784	1089.794856	9576.388219
2023-10-24	2267.445374	1560.495745	4722.133938	1104.071631	9654.146688
2023-10-25	2252.240390	1545.774193	4747.139483	1092.174375	9637.328441
2023-10-26	2311.159656	1572.118225	4698.444264	1094.553781	9676.275925
2023-10-27	2267.445374	1563.595113	4664.225622	1085.035930	9580.302039

747 rows × 5 columns

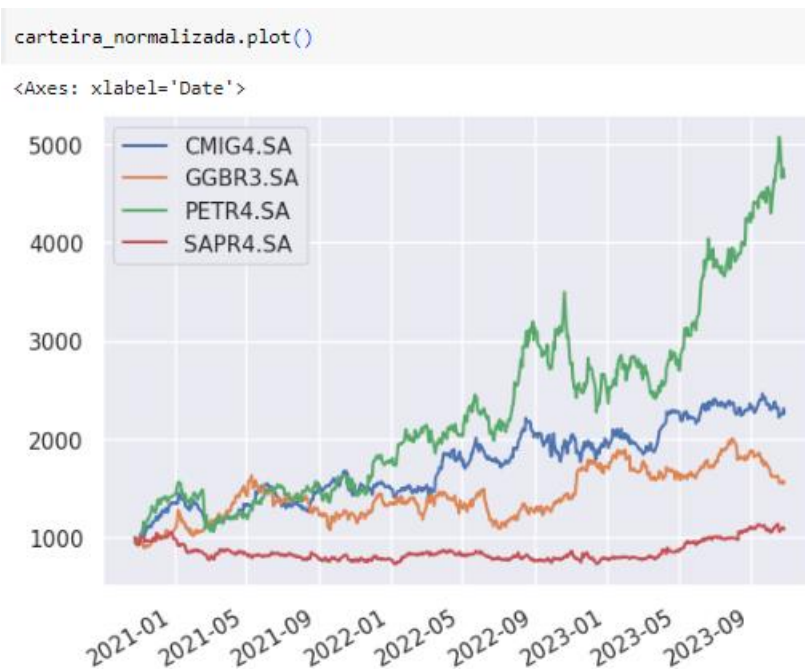
Fonte: Autoria Própria.

Tais valores da carteira normalizada podem ser melhor visualizados no gráfico da figura 29. O mesmo mostra a evolução e rentabilidade de cada ação separadamente da carteira ao longo dos 3 anos.

Assim como foi realizado para a carteira de ações, foram trazidos os dados de cotações do índice Ibovespa também dos últimos três anos com o objetivo de compará-los com os mesmos.

Assim como será visto na seção de resultados do presente trabalho, vale ressaltar a importância de se comparar ativos ou carteiras com seu *benchmark*, ou seja, o índice Ibovespa é uma média das cotações e desempenho de todas as ações listadas. Portanto, obtendo-se desempenhos melhores do que o índice usado pra comparação, é um bom indicativo de boa escolha para os investimentos.

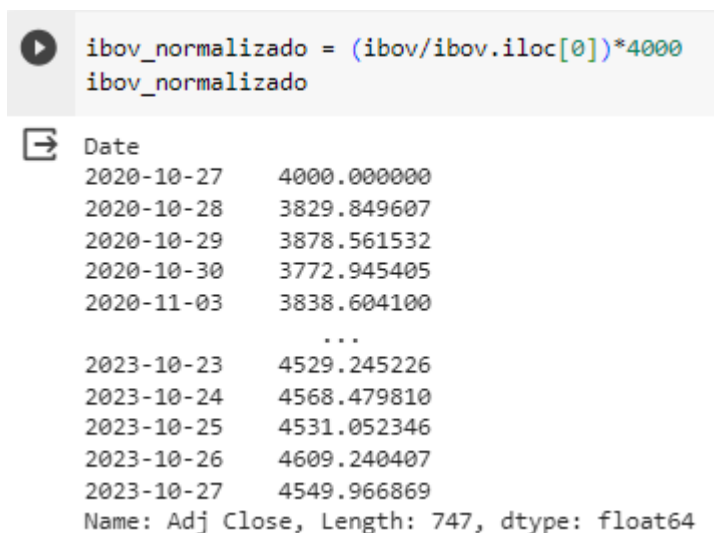
Figura 29 – Gráfico da evolução das ações normalizadas



Fonte: Autoria Própria.

A figura 30 representa a mesma lógica utilizada para a carteira de ações. Foi utilizada uma estratégia de normalização deste índice e a simulação de um investimento de R\$ 4000,00, mesmo valor anteriormente investido na carteira escolhida como um todo. Tal simulação foi realizada com o objetivo de equalizar os ganhos ou perdas percentuais em relação às ações propriamente ditas.

Figura 30 – Valores normalizados para o índice Ibovespa



Fonte: Autoria Própria.

Comparando devidamente cada ativo da carteira com o índice, é possível observar o desempenho individualizado com relação ao *Benchmark*, as ações CMIG4, GGBR3 e PETR4 tiveram um desempenho melhor nos últimos 3 anos como será melhor observado nos resultados (seção 4.2). Portanto, o experimento focou nas empresas que superaram seu índice no período estudado. Vale ressaltar que performance passada não é conclusivo sobre seu desempenho futuro. Entretanto, é um bom indicativo iniciar uma análise de ativos com empresas que performaram melhor do que a média das outras ações com o decorrer do tempo.

Com isso, como última etapa do experimento, foram extraídos novamente dados do Yahoo Finance, porém agora separadamente para estudo de cada ativo. A figura 31 representa a parte do código de coleta dos dados de uma empresa genérica (que será utilizada as ações da carteira, demonstrada nos resultados), em um período de 3 anos.

Figura 31 - Extração de dados do Yahoo Finance.

```
data = yf.download('Ticker', '2020-11-03', '2023-10-31')  
data.head()
```

Fonte: Autoria Própria.

Tendo os dados de abertura, fechamento, máxima e mínima do dia, foi escolhido a estratégia de experimento de cruzamento de médias móveis, explicada na seção 2.1.5 do presente trabalho, largamente utilizada no mercado financeiro.

Com isso, foram escolhidos dois diferentes períodos de análise para as curvas de médias móveis. A de curto prazo representando 9 períodos de cotação (dias), e a de mais longo prazo sendo de 21 períodos. Vale ressaltar que podem ser utilizados outros valores maiores em uma estratégia de mais longo prazo, entretanto optou-se por médias móveis de quantidades menores de dias devido à uma maior quantidade de cruzamentos no período de tempo analisado, e com isso, mais insumos para uma conclusão final.

Assim como visto na seção 2.1.5, a estratégia de análises de ações por meio de cruzamento de médias móveis consiste, primeiramente, na escolha da quantidade de períodos a serem levados em consideração no cálculo. Geralmente são escolhidas médias de mais curto prazo que

acompanha mais fielmente a cotação real das ações e posteriormente, seleciona-se o número de períodos a serem considerados na média de mais longo prazo.

Tendo definido a quantidade de períodos em ambas as médias, a plotagem de seu traçado no gráfico, representam informações valiosas para a análise em questão. Como ilustra a figura 32, no período em que há o cruzamento das curvas onde a média de curto prazo passa a ser maior que a média de longo prazo representa um sinal de compra. O oposto ocorre no momento que a média de curto prazo se torna menor que a de longo prazo, ou seja, com decaimento mais acentuado que a segunda, representando um sinal de venda. Tal fato ocorre devido à média de curto prazo performar mais similarmente com a cotação real do ativo, sofrendo maior oscilação em relação à média com mais períodos.

Figura 32 - Sinais de compra e venda, respectivamente



Fonte: Salas (2021).

Com isso, foram criadas as médias em Python para ambos os períodos, como mostra o trecho de código da figura 33.

Figura 33 - Criação das médias móveis

```
def crossover_sinal(data):
    #criar sinais de compra e venda
    data['MM9'] = data['Close'].rolling(9).mean()
    data['MM21'] = data['Close'].rolling(21).mean()
```

Fonte: Autoria Própria.

Após a criação e referenciação das médias móveis, foram criados mais dois parâmetros que as comparassem por meio da subtração da média de 9 dias em relação à de 21 dias, como

mostra a figura 34. Tal cálculo é realizado devido ao conceito apresentado na seção de análise técnica (2.1.5), onde diz que o ponto de encontro das médias móveis quando representa momento de compra ou venda do ativo.

O código implementado nessa etapa contou com a criação de parâmetros que representassem a diferença entre as médias em um período anterior e no período do cruzamento.

Figura 34 - Definição de parâmetros para médias móveis

```
# sinais de compra e sinais de venda

data['Anterior'] = data['MM9'].shift(1) - data['MM21'].shift(1)
data['Atual'] = data['MM9'] - data['MM21']
```

Fonte: Autoria Própria.

Portanto, o objetivo é identificar o momento exato onde o período anterior tivesse uma diferença entre médias negativa, ou seja, média móvel de 9 dias menor que a média móvel de 21 dias e o momento atual em que a diferença fosse positiva, em outras palavras, ponto de cruzamento das curvas ascendente, representando uma ordem de compra, como mostra a primeira linha de código da figura 35. Para uma ordem de venda, a estratégia foi inversa em que, o objetivo é a identificação onde o momento anterior tivesse uma diferença positiva e o momento posterior tivesse uma diferença negativa, ou seja, momento exato onde a média de curto prazo passa a ter valores menores que a curva de longo prazo.

Com os parâmetros das diferenças das médias móveis criados, foi definida a regra dos cruzamentos dos dados em função do tempo indicando oportunidades de compra ou venda dos ativos, como mostra o código na figura 35.

Figura 35 - Definição de regras das médias

```
data.loc[(data['Anterior']<0)&(data['Atual']>0),'Compra'] = data['Close']
data.loc[(data['Anterior']>0)&(data['Atual']<0),'Venda'] = data['Close']

columns = ['Close', 'MM9', 'MM21', 'Compra', 'Venda']

return data[columns]
```

Fonte: Autoria Própria.

A figura 36 exemplifica a coleta dos 10 últimos dados de fechamento e médias móveis para 9 e 21 dias respectivamente, em função do dia a ser analisado.

Figura 36 - Exemplo da representação das médias móveis

```
output_df = crossover_sinal(data)
output_df.tail(10)
```

	Close	MM9	MM21
Date			
2023-10-17	20.740000	20.970000	21.859048
2023-10-18	20.120001	20.843334	21.710476
2023-10-19	20.120001	20.748889	21.583333
2023-10-20	20.059999	20.641111	21.434286
2023-10-23	20.270000	20.566667	21.317619
2023-10-24	20.139999	20.478889	21.210000
2023-10-25	19.950001	20.376667	21.072381
2023-10-26	20.290001	20.298889	20.966191
2023-10-27	20.180000	20.207778	20.861429
2023-10-30	20.250000	20.153334	20.753334

Fonte: Autoria Própria.

Após isso, foram definidos também os períodos em que se desejaria realizar a análise do gráfico. Vale lembrar que os dados utilizados para todas as análises neste relatório foram dos últimos 3 anos. Entretanto esse período foi quebrado em 3 (anualmente) para uma melhor visualização do gráfico das médias. Portanto, os parâmetros “Data final” e “Data inicial” mostrados na figura 37 são genéricos, com o objetivo da demonstração no relatório, e foram variados três vezes para cada empresa analisada.

Figura 37 - Definição do período a ser analisado no gráfico

```
filtered_data = output_df.loc[(output_df.index<'Data final')&(output_df.index>'Data inicial')]
```

Fonte: Autoria Própria.

Portanto, tendo como concluída a definição dos parâmetros como período de análise, ações e periodicidade das médias móveis, foram definidos os dados a serem mostrados nos gráficos, assim como configuração de detalhes de cor para as análises, como mostram as figuras 38 e 39.

Figura 38 - Definição de parâmetros para o gráfico de médias móveis

```
fig, ax = plt.subplots()
ax.plot(filtered_data.index, filtered_data['Close'], label='Fechamento', alpha=0.5)
ax.plot(filtered_data.index, filtered_data['MM9'], label='MedMovel9', color='orange')
ax.plot(filtered_data.index, filtered_data['MM21'], label='MedMovel21', color='grey')
```

Fonte: Autoria Própria.

O gráfico para esta seção irá utilizar parâmetros como fechamento, ou seja, cotação do ativo no fechamento do respectivo dia, as duas médias móveis, e sinais de compra ou venda dos ativos em relação ao tempo.

Figura 39 - Parâmetros de indicação de compra e venda

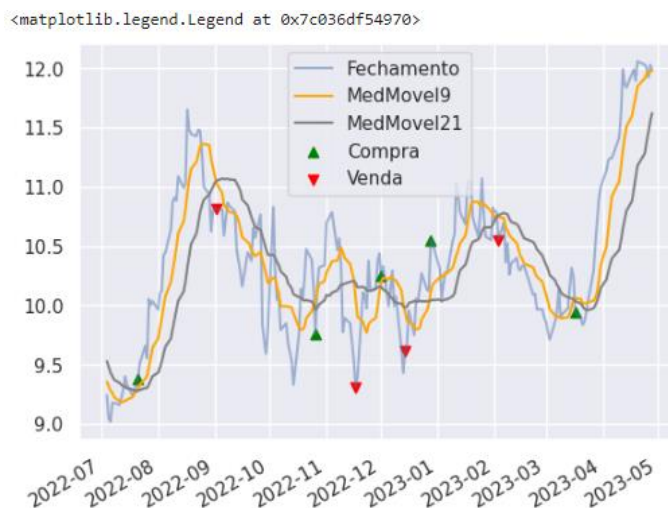
```
ax.scatter(filtered_data.index, filtered_data['Compra'], label='Compra', marker='^', color='green')
ax.scatter(filtered_data.index, filtered_data['Venda'], label='Venda', marker='v', color='red')

fig.autofmt_xdate()
ax.legend()
```

Fonte: Autoria Própria.

A figura 40 representa um exemplo de gráfico de recomendações de entrada e saída de um ativo financeiro que será melhor discutido nos resultados deste relatório.

Figura 40 - Exemplo de gráfico de cotações com médias



Fonte: Autoria Própria.

Como é possível observar, a cotação está em linha azul, tendo maior oscilação e volatilidade. A linha amarela representa a média móvel de curto prazo, ou seja, de 9 dias. Esta por sua vez, por representar uma média, já apresenta uma oscilação menor do que a cotação diária apesar de acompanhar bem todas as curvas devido à baixa quantidade de dados. E por fim, a linha cinza representa a média móvel de longo prazo, ou seja, 21 dias, apresentando a menor oscilação devido à uma maior quantidade de dados abordados em seu cálculo.

4 RESULTADOS

Este capítulo apresenta os resultados obtidos com os testes realizados e técnicas empregadas ao longo de todo o trabalho. Como visto na seção anterior, os experimentos foram divididos em duas grandes etapas. Primeiramente foram analisados dados visando mais a saúde financeira das empresas, com análises de balanços e resultados a fim de que, a partir de determinada estratégia, fossem selecionadas as mais compatíveis com o objetivo proposto.

Posteriormente, com as empresas selecionadas na primeira análise, o objetivo principal foi analisar e identificar boas oportunidades de compra ou venda do ativo no decorrer do tempo analisando métricas voltadas à histórico de preços.

4.1 RESULTADOS DA ANÁLISE FUNDAMENTALISTA

Com todas etapas de coleta, tratamento e análise dos dados devidamente realizados (ver **seção 3.1**), empregando as técnicas de seleção e filtragem dos indicadores que correspondiam ao objetivo final, foram selecionadas as 4 empresas que iriam compor a carteira de investimentos, que por sua vez atenderam a todos os requisitos solicitados. Essas, por sua vez são: Petrobrás (PETR4), Gerdau (GGBR3), Companhia Energética de Minas Gerais S.A (CMIG4) e Companhia de Saneamento do Paraná (SAPR4).

Através do comando exposto na figura 41 é possível obter tais dados, com os indicadores previamente selecionados, assim como mostra a tabela 3.

Figura 41 - Parâmetros e resultados da carteira de ações



```
selecao = filtro.loc[['PETR4','GGBR3','CMIG4','SAPR4'],['cotacao','pl','pvp','dy','roe','roic','c5y','divbpatr']]
display(selecao)
```

Fonte: Autoria Própria.

Como é possível observar, os demais dados da extração do site Fundamentus, não foram considerados para esta análise, portanto não foram trazidos para a tabela em questão.

Tabela 3 - Resultados finais dos indicadores fundamentalistas de ações selecionadas.

Ativo	cotação	pl	pvp	dy	roe	roic	c5y	divbpatr
PETR4	35.35	2.95	1.24	0.2617	0.4219	0.2895	0.1942	0.75
GGBR3	20.27	3.73	0.73	0.1596	0.1952	0.1939	0.2138	0.22
CMIG4	11.91	3.96	1.11	0.0928	0.2807	0.1635	0.1180	0.50
SAPR4	4.58	5.06	0.76	0.0815	0.1503	0.1520	0.0745	0.62

Fonte: Autoria Própria.

Vale ressaltar que devido ao fato da diferença de dias dos experimentos entre a análise fundamentalista e análise técnica, a cotação das ações sofreram leve alteração, porém sem impacto em qualquer análise realizada.

Com isso, através da coleta dos dados de cotação dos ativos e do índice Ibovespa em um período determinado de tempo (seção 3.2), foi possível obter a comparação de rentabilidade entre a carteira como um todo com o índice Ibovespa. O que foi observado é que na análise de três anos a carteira de ações teve um desempenho muito superior ao do índice.

Na figura 42 é possível notar que ambas tiveram desempenho similar do início da análise até aproximadamente setembro de 2021. Entretanto, após esse período fica evidente a diferença de performance entre as ações escolhidas em relação ao índice principal da bolsa em um período relativamente longo, portanto provando resultados sustentáveis ao longo do tempo.

Com as carteiras simuladas neste relatório, um investimento de R\$ 4000,00 no índice no início da análise realizada, traria um valor final de rentabilidade de R\$ 4549,97 no fim do período examinado (27/10/2023). Para os resultados da carteira selecionada, com os mesmos R\$ 4000,00 aportados na mesma data, traria um resultado final de R\$ 9580,30 no fim do período analisado, como é possível visualizar na tabela 4.

Tabela 4 - Comparação dos resultados finais: carteira x Ibovespa.

Ativo	Aporte (R\$)	Saldo Final (R\$)	Rentabilidade (R\$)	Rentabilidade Carteira (R\$)	Rentabilidade (%)	Rentabilidade Carteira (%)
PETR4	1.000,00	4.664,22	3.664,22	5.580,30	366,42%	139,5%
GGBR3	1.000,00	1.563,59	563,59		56,36%	
CMIG4	1.000,00	2.267,44	1.267,44		126,74%	
SAPR4	1.000,00	1.085,03	85,03		8,50%	
IBOV	4.000,00	4.549,97	549,97	549,97	13,75%	13,75%

Fonte: Autoria Própria.

Portanto é possível observar uma grande diferença de evolução e rentabilidade entre a carteira selecionada em relação ao índice Ibovespa. Quando se analisa individualmente as empresas em relação ao índice, o resultado também é satisfatório, ocorrendo que em 75% das ações (3 empresas das 4 previamente selecionadas) superaram o mesmo, como mostra a figura 42.

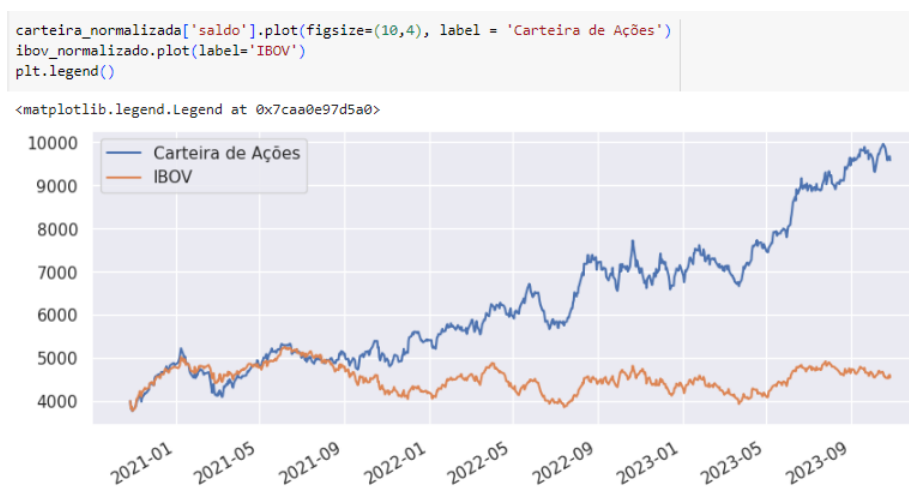
Figura 42 - Evolução individualizada das ações



Fonte: Autoria Própria.

A fim de facilitar a visualização dos resultados consolidados para devida comparação, a figura 43 representa a evolução da carteira como um todo em relação ao índice Ibovespa no período em questão.

Figura 43 - Comparação da performance: carteira x Ibovespa



Fonte: Autoria Própria.

4.2 RESULTADOS DA ANÁLISE TÉCNICA

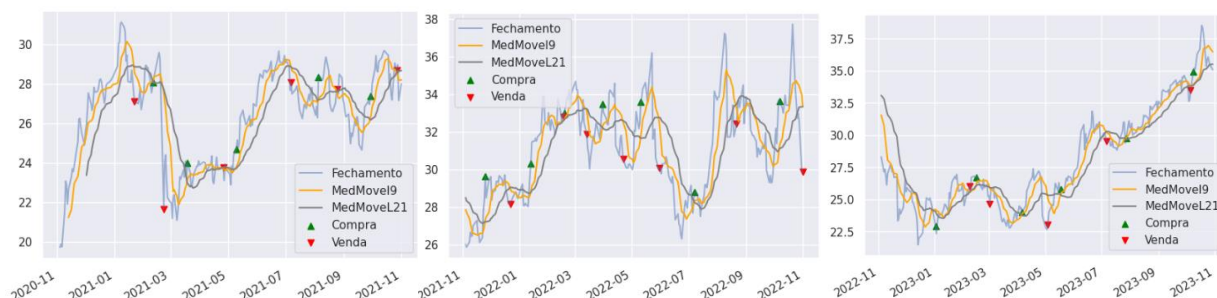
Tendo definido as ações por meio da análise fundamentalista, através da análise da saúde financeira das empresas, lucro gerado, dívidas, histórico de dividendos, entre outros, foi possível tirar informações valiosas de tomada de decisão através da análise técnica. Como visto na seção anterior, através da definição destes parâmetros, foi possível realizar a escolha de empresas com bons fundamentos garantindo uma vantagem na rentabilidade em relação ao seu *benchmark*.

Como visto na tabela 4 e também na figura 42, o único ativo que não superou o índice no período analisado foi o SAPR4. Vale ressaltar que em todo o tempo de análise, sua rentabilidade foi abaixo do Ibovespa, apesar de contar com bons fundamentos na análise fundamentalista, tendo uma evolução de somente 8,5% nos três anos. Portanto, foi decidido realizar o estudo de médias móveis, procurando ajustes finos de entrada e saída do ativo, somente com as ações que tiveram bons desempenhos no período. Com isso, a estratégia adotada para essa seção foi o estudo separadamente de cada ativo e seus respectivos momentos de compra e venda validando a estratégia de utilização de médias móveis.

Para todos os casos estudados (PETR4, CMIG4 e GGBR3) foram utilizadas estratégias de cruzamento de médias móveis e através da definição de parâmetros de indicação de compra ou venda, é possível observar com mais facilidade a eficácia desse experimento. Outro ponto a se destacar, é que o período de 3 anos inicialmente analisado, foi dividido em 3 partes devido à quantidade de dados a serem visualizados no mesmo gráfico, como mostra a figura 44.

Para isso, iniciou-se a etapa com a análise dos gráficos da Petrobrás que mostrou uma eficácia de 41,2% no método.

Figura 44 – Cruzamento de médias móveis para PETR4



Fonte: Autoria Própria.

Vale ressaltar, que o pior desempenho acontece no segundo período devido à uma grande lateralização dos preços dos ativos, ou seja, não há uma tendência clara de alta ou de baixa no período analisado, como mostra a tabela 5.

Para tais períodos, há um grande encontro entre as médias móveis devido à alta oscilação de preços, com isso diminui o desempenho do método. Diferentemente do observado no terceiro período onde há uma tendência de alta com ganhos muito mais expressivos.

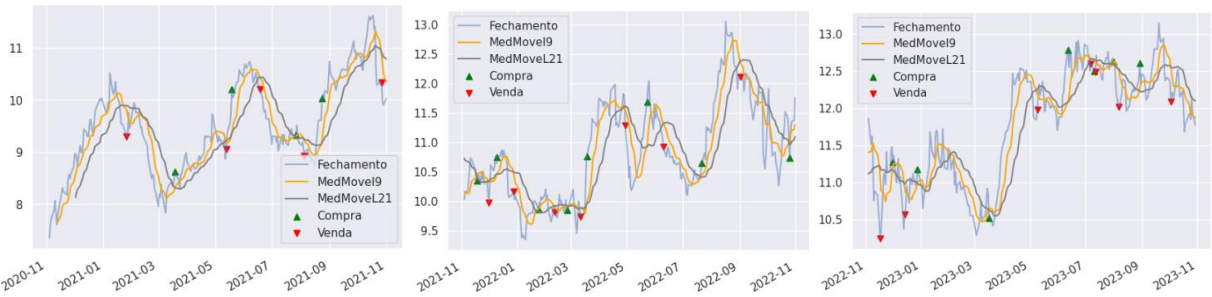
Tabela 5 – Aproveitamento do método para PETR4

PETR4	
Período	Aproveitamento
11/2020 – 11/2021	40%
11/2021 – 11/2022	28,5%
11/2022 – 11/2023	60%
TOTAL	41,2%

Fonte: Autoria Própria.

Para a análise da CMIG4, também é possível analisar tal fato. O pior desempenho do método se encontra no segundo período como mostra a figura 45. É possível ver uma alta de preços no geral, porém há uma grande lateralização de cotações abaixando o desempenho. Vale citar o primeiro período onde claramente observa-se uma tendência de alta, espaçando-se mais os cruzamentos de médias móveis, também obtendo-se melhores desempenhos e rentabilidades.

Figura 45 – Cruzamento de médias móveis para CMIG4.



Fonte: Autoria Própria.

A tabela 6 demonstra os valores de aproveitamento do método para a CMIG4. O aproveitamento final foi um melhor do que a PETR4 (tabela 5).

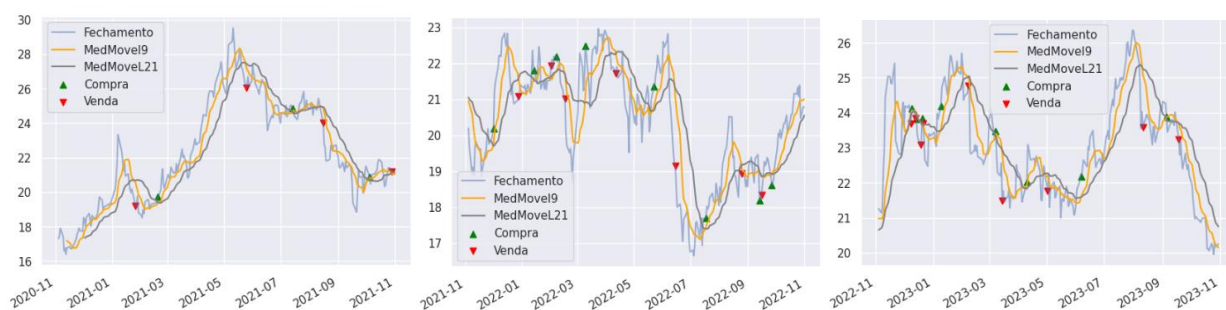
Tabela 6 – Aproveitamento do método para CMIG4

CMIG4	
Período	Aproveitamento
11/2020 – 11/2021	80%
11/2021 – 11/2022	28%
11/2022 – 11/2023	50%
TOTAL	50%

Fonte: Autoria Própria.

Por fim, o desempenho da estratégia utilizada para a empresa Gerdau foi similar à CMIG4. Para a GGBR3 houve grande lateralização de preços no primeiro e terceiro períodos, impactando o desempenho da análise, conforme ilustrado na figura 46.

Figura 46 – Cruzamento de médias móveis para GGBR3.



Fonte: Autoria Própria.

A tabela 7 explicita os valores de desempenho do método para as ações da Gerdau.

Tabela 7 – Aproveitamento do método para GGBR3

GGBR3	
Período	Aproveitamento
11/2020 – 11/2021	62,5%
11/2021 – 11/2022	66,7%
11/2022 – 11/2023	28,5%
TOTAL	50%

Fonte: Autoria Própria.

A eficiência do método foi medida através da quantidade de operações onde foram auferidos lucros em relação a quantidade de interações totais. Uma interação se resume a um par de recomendações (compra e venda), portanto, uma interação positiva trata-se de uma operação onde houve lucro, ou seja, a ordem de venda foi a um preço superior ao de compra. O oposto ocorre para uma interação negativa, onde há prejuízo financeiro, ou seja, a ordem de compra foi superior à ordem de venda.

5 CONCLUSÃO

Neste trabalho foi possível explorar importantes áreas do conhecimento no que se refere à análise de dados e mercado financeiro. Primeiramente, foi possível obter um desempenho eficaz na escolha de ações da bolsa de valores brasileira por meio de estratégias com indicadores fundamentalistas. Em que o desempenho da carteira de ativos selecionados na primeira análise teve uma rentabilidade 125,75% maior do que o índice de ações brasileiras no período. Tais resultados foram obtidos por meio da combinação de indicadores financeiros mais relevantes para a análise desejada, assim como a definição de valores eficazes para cada um desses indicadores.

Vale ressaltar que a análise realizada foi referente ao período que se iniciou no final de 2020 e foi até o final de 2023, que por sua vez ocorreram acontecimentos mundialmente conhecidos como, o auge da crise da pandemia do novo coronavírus e também a guerra entre Ucrânia e Rússia. Tais ocorridos impactaram o mundo como um todo de diversas formas, inclusive o desempenho de muitas das ações listadas na bolsa de valores, assim como na rentabilidade da carteira selecionada. Entretanto, todo o estudo e experimento acerca da análise fundamentalista estudada no presente trabalho evidencia que a escolha de ações baseadas em seus fundamentos, são capazes de trazer ganhos significativos no longo prazo.

Foi também possível observar a importância da análise técnica para a definição do melhor momento de entrada ou saída de um determinado ativo. Tendo como base as ações previamente selecionadas baseada no seu valor intrínseco, foi utilizado o método de cruzamento de médias móveis das cotações para o mesmo período estudado anteriormente.

Para esta etapa, obteve-se um aproveitamento de aproximadamente 50% de acertos de momento de compra e venda ao longo da análise das ações da Petrobrás, Gerdau e Cemig. Vale ressaltar que o desempenho foi diretamente impactado devido à lateralização de preços das ações em determinados períodos, ou seja, grande quantidade de cruzamentos de médias sem muita alteração no preço. Com isso, para períodos que em que houveram clara tendência de alta, mesmo no longo prazo, o desempenho do método se mostrou mais eficiente e com cruzamentos de médias mais distantes, ou seja, indicativo de venda com rentabilidade mais expressiva.

Com isso, foi possível concluir que trata-se de uma estratégia importante na análise da dinâmica de preços, podendo ser escalada para outros produtos de renda variável, tais como fundos imobiliários e ETF's, mas deve-se considerar sua utilização em conjunto com outras técnicas para uma boa maior efetividade nas tomadas de decisão.

Por fim, outra consideração realizada foi que, o trabalho como um todo seguiu uma linha sequencial de experimentos utilizando uma determinada estratégia. Primeiramente, seleção das empresas baseadas em seus fundamentos com valores adotados, com isso classificação das mesmas pelo seu patrimônio líquido, escolha dos ativos de melhor rendimento que o Ibovespa para a análise técnica e o experimento com médias móveis, e que tais critérios caso fossem realizados em ordem inversa, teriam resultados completamente diferentes segundo os critérios definidos.

Para trabalhos futuros, adicionalmente à todas as estratégias e métodos utilizados, recomenda-se a utilização do *Machine Learning* (Aprendizado de Máquina) por trabalhar com análises preditivas e prescritivas, ou seja, faz previsões de resultados futuros com a utilização de dados históricos com mais assertividade.

Para isso, é necessário o estudo do melhor método para aplicações na utilização de maior quantidade de indicadores financeiros na análise fundamentalista, e utilização de outras estratégias na análise grafista de ações visando um maior acerto na escolha de ações e momento da tomada de decisão.

REFERÊNCIAS

BRAZILLAB. **A Indústria 4.0 já é realidade, e vai revolucionar o setor de GovTechs.** [S. l.], 7 out. 2019. Disponível em: <https://brazillab.org.br/noticias/a-industria-40-ja-e-realidade-e-vai-revolucionar-o-setor-de-govtechs>. Acesso em: 3 ago. 2023.

DAMODARAN, A. **Avaliação de empresas.** 2. ed. São Paulo: Pearson Prentice Hall, 2007

DUQUE, Gabriel. **O que é machine learning e quais são as 7 etapas do aprendizado das máquinas.** [S. l.], 11 out. 2018. Disponível em: <https://blog.idwall.co/o-que-e-machine-learning/>. Acesso em: 1 set. 2023.

FERRARI, Hamilton; NASCIMENTO, Houldine. **Número de investidores na B3 bate recorde no 1º semestre:** Foram 6,2 milhões de contas ao todo, o que representa um crescimento de 5,6% em relação ao mesmo período em 2022. São Paulo, 8 set. 2023. Disponível em: <https://www.poder360.com.br/economia/numero-de-investidores-na-b3-bate-recorde-no-1o-semester/#:~:text=A%20quantidade%20de%20investidores%20na,203%2C1%20milh%C3%B5es%20em%202022>. Acesso em: 15 out. 2023.

GUTTA, Surya. **Data Science: The 5 V's of Big Data.** [S. l.], 4 maio 2020. Disponível em: <https://medium.com/analytics-vidhya/the-5-vs-of-big-data-2758bfcc51d>. Acesso em: 2 set. 2023.

INFOMONEY (São Paulo). **Análise Fundamentalista de Ações: como identificar empresas sólidas e rentáveis a longo prazo.** [S. l.], 14 out. 2022. Disponível em: <https://www.infomoney.com.br/guias/analise-fundamentalista/>. Acesso em: 4 jul. 2023.

INFOMONEY. **Entenda como funciona o mercado de ações e a bolsa de valores.** [S. l.], 8 nov. 2022. Disponível em: <https://www.infomoney.com.br/guias/mercado-de-acoes/>. Acesso em: 4 set. 2023.

MCCULLUM, Nick. **How to Embed Interactive Python Visualizations on Your Website with Python and Matplotlib.** [S. l.], 7 maio 2020. Disponível em:

<https://www.freecodecamp.org/news/how-to-embed-interactive-python-visualizations-on-your-website-with-python-and-matplotlib/>. Acesso em: 4 set. 2023.

MENDES, Diego. **Número de investidores na bolsa cresce 15% em 2022 apostando na diversificação**: De acordo com a B3, 48% dos investidores que entram no mercado de ações estão na faixa de 25 a 39 anos. São Paulo, 1 set. 2022. Disponível em: <https://www.cnnbrasil.com.br/economia/numero-de-investidores-na-bolsa-cresce-15-em-2022-apostando-na-diversificacao/#:~:text=Entre%20julho%20de%202021%20e,em%20renda%20fixa%20e%20vari%C3%A1vel>. Acesso em: 2 out. 2023

MJV. **O que é ETL e por que devemos integrar dados?**. [S. l.]: MJV, 21 jan. 2021. Disponível em: <https://www.mjvinnovation.com/pt-br/blog/o-que-e-etl-como-funciona/>. Acesso em: 5 ago. 2023.

O QUE é Business Intelligence e como aplicar na sua empresa. [S. l.]: Sankhya, 2022. Disponível em: <https://www.sankhya.com.br/blog/5-ferramentas-de-business-intelligence-e-como-aplicar-na-sua-empresa/>. Acesso em: 7 set. 2023.

PALEPU, K. G.; HEALY, P. G.; BERNARD, V. L. **Business analysis and valuation: using financial statements**. 2nd ed. Ohio: South-Western College, 2000.

PYTHON Logo. [S. l.]: Logosmarcas, 2022. Disponível em: <https://logosmarcas.net/python-logo/>. Acesso em: 1 set. 2023.

REIS, Tiago. **Agentes do mercado financeiro: saiba quem são os principais**. São Paulo, 30 out. 2020. Disponível em: <https://www.suno.com.br/artigos/agentes-do-mercado/>. Acesso em: 4 set. 2023.

REIS, Tiago. **Análise fundamentalista: o que é? Guia completo de análise de ações**. [S. l.], 20 set. 2021. Disponível em: <https://www.suno.com.br/guias/analise-fundamentalista/>. Acesso em: 3 jul. 2023.

SACRAMENTO, Gabriel. **Entenda como funciona a transformação digital no mercado financeiro.** [S. l.], 17 set. 2021. Disponível em: <https://rockcontent.com/br/blog/transformacao-digital-mercado-financeiro/>. Acesso em: 8 ago. 2023

SALAS, Isabelle. **MÉDIAS MÓVEIS NO SETUP: SAIBA QUAL O MELHOR JEITO DE USAR ESSE INDICADOR.** [S. l.], 14 jan. 2021. Disponível em: <https://yappi.com.br/medias-moveis-no-setup/>. Acesso em: 7 nov. 2023.

SANTOS, Jones. **Você precisa de um Data Warehouse?.** [S. l.], 30 ago. 2018. Disponível em: <https://www.linkedin.com/pulse/voc%C3%AA-precisa-de-um-data-warehouse-jones-santos/?originalSubdomain=pt>. Acesso em: 6 ago. 2023.

SILVEIRA, Cristiano Bertulucci. **Indústria 4.0: O que é, e como ela vai impactar o mundo.** [S. l.], 11 fev. 2016. Disponível em: <https://www.citisystems.com.br/industria-4-0/>. Acesso em: 5 set. 2023.

USO da Inteligência Artificial para indicação de livros a leitores. Micreiros, 20 maio 2023. Disponível em: <https://micreiros.com/uso-da-inteligencia-artificial-para-indicacao-de-livros-a-leitores/>. Acesso em: 7 set. 2023.

APÊNDICE A – Construção da análise fundamentalista

!pip install fundamentus

Collecting fundamentus

Downloading fundamentus-0.2.0-py3-none-any.whl (13 kB)

Requirement already satisfied: requests>=2.25.1 in /usr/local/lib/python3.10/dist-packages (from fundamentus) (2.31.0)

Collecting requests-cache>=0.5.2 (from fundamentus)

Downloading requests_cache-1.1.0-py3-none-any.whl (60 kB)

60.1/60.1 kB 2.1 MB/s eta 0:00:00

Requirement already satisfied: pandas>=1.1.5 in /usr/local/lib/python3.10/dist-packages (from fundamentus) (1.5.3)

Requirement already satisfied: lxml>=4.6.2 in /usr/local/lib/python3.10/dist-packages (from fundamentus) (4.9.3)

Requirement already satisfied: tabulate>=0.8.7 in /usr/local/lib/python3.10/dist-packages (from fundamentus) (0.9.0)

Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->fundamentus) (2.8.2)

Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->fundamentus) (2023.3.post1)

Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.1.5->fundamentus) (1.23.5)

Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25.1->fundamentus) (3.3.1)

Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25.1->fundamentus) (3.4)

Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25.1->fundamentus) (2.0.7)

Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.25.1->fundamentus) (2023.7.22)

Requirement already satisfied: attrs>=21.2 in /usr/local/lib/python3.10/dist-packages (from requests-cache>=0.5.2->fundamentus) (23.1.0)

Collecting cattr>=22.2 (from requests-cache>=0.5.2->fundamentus)

Downloading cattr-23.1.2-py3-none-any.whl (50 kB)

50.8/50.8 kB 4.2 MB/s eta 0:00:00

Requirement already satisfied: platformdirs>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests-cache>=0.5.2->fundamentus) (3.11.0)

Collecting url-normalize>=1.4 (from requests-cache>=0.5.2->fundamentus)

Downloading url_normalize-1.4.3-py2.py3-none-any.whl (6.8 kB)

Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from cattr>=22.2->requests-cache>=0.5.2->fundamentus) (1.1.3)

Requirement already satisfied: typing_extensions>=4.1.0 in /usr/local/lib/python3.10/dist-packages (from cattr>=22.2->requests-cache>=0.5.2->fundamentus) (4.5.0)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.1->pandas>=1.1.5->fundamentus) (1.16.0)

Installing collected packages: url-normalize, cattr, requests-cache, fundamentus

Successfully installed cattr-23.1.2 fundamentus-0.2.0 requests-cache-1.1.0 url-normalize-1.4.3

```
[ ] import fundamentus
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
```

```
[ ] df = fundamentus.get_resultado()
display(df.columns)
```

```
Index(['cotacao', 'pl', 'pvp', 'psr', 'dy', 'pa', 'pcg', 'pebit', 'pacl',
      'evebit', 'evebitda', 'mrgebit', 'mrgliq', 'roic', 'roe', 'liqc',
      'liq2m', 'patrliq', 'divbpatr', 'csy'],
      dtype='object', name='Multiples')
```

display(df)

	Multiples	cotacao	pl	pvp	psr	dy	pa	pcg	pebit	pacl	evebit	evebitda	mrgebit	mrgliq	roic	roe	liqc	liq2m	patrliq	divbpatr	c5y
	papel																				
	AALR3	10.06	-4.15	1.38	1.051	0.0000	0.472	-3.70	-702.49	-0.96	-1231.19	18.16	-0.0015	-0.2460	-0.0007	-0.3311	0.55	1474210.0	8.653780e+08	1.15	0.0306
	ABCB3	0.00	0.00	0.00	0.000	0.0000	0.000	0.00	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0000	0.1489	0.00	0.0	5.420810e+09	0.00	0.1728
	ABCB4	19.84	5.56	0.83	0.000	0.0608	0.000	0.00	0.00	0.00	0.00	0.00	0.0000	0.0000	0.0000	0.1489	0.00	11650400.0	5.420810e+09	0.00	0.1728
	ABEV3	13.51	14.11	2.36	2.582	0.0564	1.543	57.42	12.52	-24.18	11.70	8.61	0.2063	0.1882	0.1650	0.1671	1.11	330582000.0	9.025490e+10	0.04	0.1323
	ABYA3	4.91	-214.80	1.76	2.055	0.0000	0.527	1.98	19.96	-2.75	33.67	33.67	0.1029	-0.0096	0.0278	-0.0082	2.09	0.0	2.920600e+08	1.31	0.1641
	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***	***
	WLMM4	36.68	11.78	2.04	0.602	0.0260	1.415	3.84	9.98	4.57	9.97	9.66	0.0603	0.0511	0.1763	0.1735	2.48	99706.0	6.533200e+08	0.24	0.2526
	WMBY3	25.39	-19.30	2.87	0.836	0.0000	0.182	1.20	8.62	-1.50	25.02	25.02	0.0970	-0.0705	0.0239	-0.1486	1.44	0.0	2.124390e+08	6.51	-0.1448
	WSON33	67.00	8.07	0.98	1.067	0.0000	0.400	13.41	2.42	-0.89	5.08	5.08	0.4414	0.1358	0.1836	0.1217	1.26	0.0	2.148530e+09	1.21	0.0523
	YDUQ3	17.62	49.43	1.74	1.121	0.0156	0.560	5.90	7.27	-1.36	12.97	6.52	0.1542	0.0233	0.0899	0.0351	1.56	101289000.0	3.138850e+09	1.73	0.0849
	ZAMP3	6.40	-15.81	1.29	0.466	0.0000	0.439	13.22	20.73	-0.99	29.38	4.91	0.0225	-0.0295	0.0254	-0.0815	1.18	13905300.0	1.367230e+09	0.85	0.1038

967 rows × 20 columns

```
filtro = df[(df.pvp < 1.5) & (df.roe > 0.15) & (df.pl < 10) & (df.dy > 0.07) & (df.csy > 0) & (df.divbpatr < 0.9) & (df.roic > 0.1)]
filtro.sort_values('patrlq', ascending = False, inplace=True)
display(filtro)
```

<ipython-input-5-e4cde5f7e379>:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
filtro.sort_values('patrlq', ascending = False, inplace=True)

	Multiples	cotacao	pl	pvp	psr	dy	pa	pcg	pebit	pacl	evebit	evebitda	mrgebit	mrqliq	roic	roe	liqc	liq2m	patrlq	divbpatr	c5y
papel																					
PETR3	38.50	3.21	1.35	0.864	0.2403	0.507	-33.97	1.92	-1.04	2.76	2.15	0.4497	0.2699	0.2895	0.4219	0.90	4.597730e+08	3.706830e+11	0.75	0.1942	
PETR4	35.51	2.96	1.25	0.796	0.2605	0.468	-31.33	1.77	-0.96	2.61	2.03	0.4497	0.2699	0.2895	0.4219	0.90	1.736910e+09	3.706830e+11	0.75	0.1942	
GGBR3	21.07	4.55	0.73	0.513	0.1535	0.485	1.82	3.57	7.03	4.10	3.17	0.1437	0.1132	0.1602	0.1607	2.93	1.110810e+06	5.060090e+10	0.23	0.2096	
GGBR4	23.12	5.00	0.80	0.563	0.1399	0.532	2.00	3.92	7.71	4.45	3.43	0.1437	0.1132	0.1602	0.1607	2.93	2.046780e+08	5.060090e+10	0.23	0.2096	
CMIG4	12.68	4.21	1.18	0.778	0.0872	0.512	8.18	3.53	-1.61	4.56	3.95	0.2202	0.1849	0.1635	0.2807	1.34	1.094970e+08	2.360910e+10	0.50	0.1180	
GOAU3	10.69	4.03	0.62	0.153	0.1085	0.143	0.52	1.07	1.76	1.50	1.16	0.1434	0.1135	0.1599	0.1529	3.02	5.261200e+05	1.794810e+10	0.64	0.2096	
GOAU4	10.88	4.10	0.63	0.156	0.1066	0.145	0.53	1.09	1.79	1.52	1.17	0.1434	0.1135	0.1599	0.1529	3.02	7.732120e+07	1.794810e+10	0.64	0.2096	
SAPR11	22.92	5.07	0.76	1.175	0.0800	0.391	8.35	2.81	-1.15	4.62	3.90	0.4183	0.2319	0.1520	0.1503	1.47	3.062570e+07	9.100570e+09	0.62	0.0745	
SAPR3	4.50	4.97	0.75	1.153	0.0754	0.384	8.19	2.76	-1.13	4.56	3.85	0.4183	0.2319	0.1520	0.1503	1.47	1.232540e+06	9.100570e+09	0.62	0.0745	
SAPR4	4.61	5.09	0.77	1.181	0.0810	0.393	8.40	2.82	-1.16	4.63	3.91	0.4183	0.2319	0.1520	0.1503	1.47	8.799100e+06	9.100570e+09	0.62	0.0745	
CSMG3	18.13	5.33	0.87	0.985	0.0921	0.492	6.89	4.12	-2.02	6.12	4.18	0.2392	0.1847	0.1346	0.1638	1.58	3.915270e+07	7.891110e+09	0.58	0.0853	
EQMA3B	26.99	5.43	1.17	0.848	0.1034	0.508	7.24	3.62	-1.90	4.94	4.14	0.2346	0.1562	0.1696	0.2156	1.31	3.946680e+03	3.783950e+09	0.69	0.1066	
POMO3	4.49	5.34	1.26	0.662	0.0833	0.558	1.72	5.23	10.22	6.50	5.58	0.1265	0.1251	0.1458	0.2364	2.16	1.246310e+06	3.369750e+09	0.73	0.1055	
MTSA4	45.89	4.27	0.97	0.555	0.0731	0.736	1.32	3.60	1.37	2.76	2.52	0.1541	0.1302	0.2888	0.2264	3.60	2.067740e+05	4.175870e+08	0.09	0.3160	
TRPN3	0.71	0.44	0.54	0.263	1.0474	0.399	0.59	0.36	0.59	-0.35	-0.35	0.7228	0.5929	5.2301	1.2235	3.61	0.000000e+00	5.935800e+07	0.00	0.0577	

```
[ ] filtro.shape
(15, 20)
```

```
filtro.head(10)
```

	Multiples	cotacao	pl	pvp	psr	dy	pa	pcg	pebit	pacl	evebit	evebitda	mrgebit	mrqliq	roic	roe	liqc	liq2m	patrlq	divbpatr	c5y
papel																					
PETR3	38.50	3.21	1.35	0.864	0.2403	0.507	-33.97	1.92	-1.04	2.76	2.15	0.4497	0.2699	0.2895	0.4219	0.90	4.597730e+08	3.706830e+11	0.75	0.1942	
PETR4	35.51	2.96	1.25	0.796	0.2605	0.468	-31.33	1.77	-0.96	2.61	2.03	0.4497	0.2699	0.2895	0.4219	0.90	1.736910e+09	3.706830e+11	0.75	0.1942	
GGBR3	21.07	4.55	0.73	0.513	0.1535	0.485	1.82	3.57	7.03	4.10	3.17	0.1437	0.1132	0.1602	0.1607	2.93	1.110810e+06	5.060090e+10	0.23	0.2096	
GGBR4	23.12	5.00	0.80	0.563	0.1399	0.532	2.00	3.92	7.71	4.45	3.43	0.1437	0.1132	0.1602	0.1607	2.93	2.046780e+08	5.060090e+10	0.23	0.2096	
CMIG4	12.68	4.21	1.18	0.778	0.0872	0.512	8.18	3.53	-1.61	4.56	3.95	0.2202	0.1849	0.1635	0.2807	1.34	1.094970e+08	2.360910e+10	0.50	0.1180	
GOAU3	10.69	4.03	0.62	0.153	0.1085	0.143	0.52	1.07	1.76	1.50	1.16	0.1434	0.1135	0.1599	0.1529	3.02	5.261200e+05	1.794810e+10	0.64	0.2096	
GOAU4	10.88	4.10	0.63	0.156	0.1066	0.145	0.53	1.09	1.79	1.52	1.17	0.1434	0.1135	0.1599	0.1529	3.02	7.732120e+07	1.794810e+10	0.64	0.2096	
SAPR11	22.92	5.07	0.76	1.175	0.0800	0.391	8.35	2.81	-1.15	4.62	3.90	0.4183	0.2319	0.1520	0.1503	1.47	3.062570e+07	9.100570e+09	0.62	0.0745	
SAPR3	4.50	4.97	0.75	1.153	0.0754	0.384	8.19	2.76	-1.13	4.56	3.85	0.4183	0.2319	0.1520	0.1503	1.47	1.232540e+06	9.100570e+09	0.62	0.0745	
SAPR4	4.61	5.09	0.77	1.181	0.0810	0.393	8.40	2.82	-1.16	4.63	3.91	0.4183	0.2319	0.1520	0.1503	1.47	8.799100e+06	9.100570e+09	0.62	0.0745	

PETR3; GGBR4; GOAU3; GOAU4; SAPR11; SAPR3 | Ações pertencentes às mesmas empresas dos ativos escolhidos abaixo

```
[ ] selecao = filtro.loc[['PETR4','GGBR3','CMIG4','SAPR4'],['cotacao','pl','pvp','dy','roe','roic','c5y','divbpatr']]
display(selecao)
```

	Multiples	cotacao	pl	pvp	dy	roe	roic	c5y	divbpatr
papel									
PETR4	35.51	2.96	1.25	0.2605	0.4219	0.2895	0.1942		0.75
GGBR3	21.07	4.55	0.73	0.1535	0.1607	0.1602	0.2096		0.23
CMIG4	12.68	4.21	1.18	0.0872	0.2807	0.1635	0.1180		0.50
SAPR4	4.61	5.09	0.77	0.0810	0.1503	0.1520	0.0745		0.62

APÊNDICE B – Construção da análise técnica

```
!pip install yfinance --upgrade --no-cache-dir
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-packages (0.2.31)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.5.3)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.23.5)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.31.0)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.10/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.9.3)
Requirement already satisfied: appdirs>=1.4.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.4.4)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2023.3.post1)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.3.8)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.10/dist-packages (from yfinance) (3.17.0)
Requirement already satisfied: BeautifulSoup4>=4.11.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.11.2)
Requirement already satisfied: HTML5lib>=1.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from BeautifulSoup4>=4.11.1->yfinance) (2.5)
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.10/dist-packages (from HTML5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from HTML5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3.0->yfinance) (2.8.2)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.3.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2023.7.22)
```

```
[ ] #Importando as bibliotecas
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pandas_datareader.data as web
import yfinance as yf
import seaborn as sns
import math
```

```
yf.pdr_override()
```

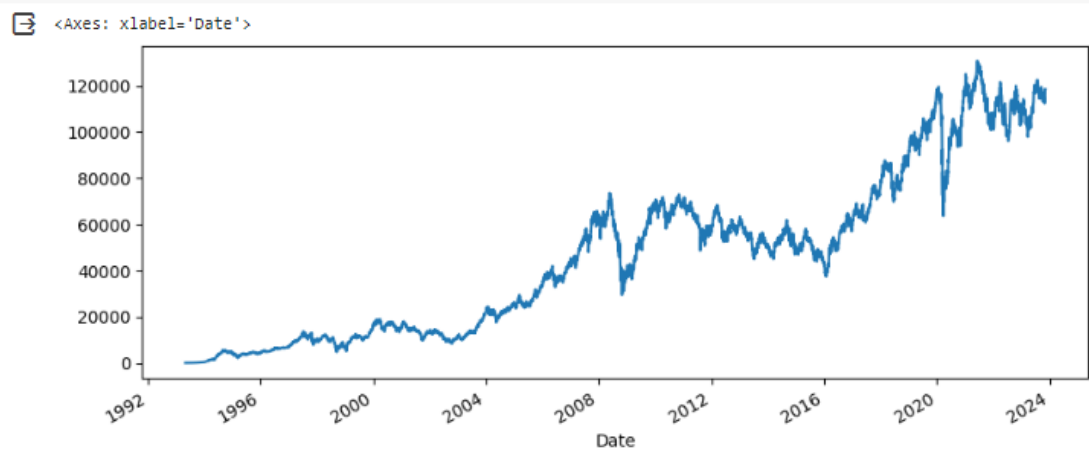
```
[ ] ibov = web.get_data_yahoo('^BVSP')
```

```
[*****100%*****] 1 of 1 completed
```

```
[ ] ibov.tail()
```

	Open	High	Low	Close	Adj Close	Volume
Date						
2023-10-30	113303.00000	114204.000000	112309.000000	112532.00	112532.00	9741800
2023-10-31	112534.00000	113597.000000	112098.000000	113144.00	113144.00	10557400
2023-11-01	113157.00000	115433.000000	113157.000000	115053.00	115053.00	13904100
2023-11-03	115062.00000	118502.000000	115062.000000	118160.00	118160.00	15714000
2023-11-06	118159.96875	118757.523438	118044.820312	118431.25	118431.25	0

```
ibov["Close"].plot(figsize=(10,4)) #gráfico total do ibovespa
```



```
[ ] # simulando carteira de ações
empresas = ['PETR4.SA','GGBR3.SA','CMIG4.SA','SAPR4.SA']
carteira = web.get_data_yahoo(empresas, period= '3y')['Adj Close']
ibov = web.get_data_yahoo('^BVSP', period= '3y')['Adj Close']
```

```
[*****100%*****] 4 of 4 completed
[*****100%*****] 1 of 1 completed
```

```
[ ] display(carteira)
```

	CMIG4.SA	GGBR3.SA	PETR4.SA	SAPR4.SA
Date				
2020-11-06	5.491892	12.519905	7.548573	4.235268
2020-11-09	5.609575	12.265217	8.259478	4.251589
2020-11-10	5.732162	11.682118	8.821321	4.251589
2020-11-11	5.536023	11.796056	8.744880	4.194466
2020-11-12	5.408533	11.541368	8.374138	4.039417
...	...	...	...	...
2023-10-30	11.770000	20.250000	35.080002	4.460000
2023-10-31	11.740000	20.059999	34.740002	4.360000
2023-11-01	12.240000	20.250000	35.119999	4.500000
2023-11-03	12.550000	20.950001	35.540001	4.570000
2023-11-06	12.660000	21.070000	35.509998	4.610000

745 rows x 4 columns

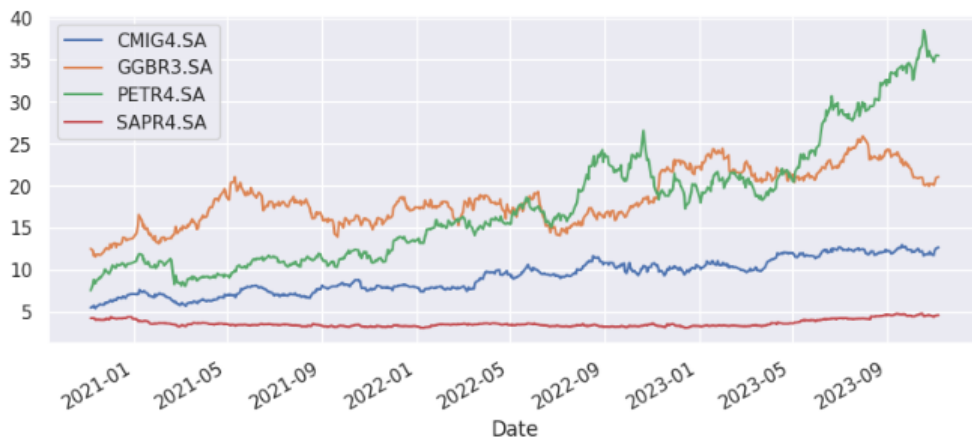
```
[ ] display(ibov)
```

```
Date
2020-11-06    100799.0
2020-11-09    103913.0
2020-11-10    105351.0
2020-11-11    104532.0
2020-11-12    102175.0
...
2023-10-30    112532.0
2023-10-31    113144.0
2023-11-01    115053.0
2023-11-03    118160.0
2023-11-06    118487.0
Name: Adj Close, Length: 745, dtype: float64
```

RESULTADOS

```
[ ] sns.set()
    carteira.plot(figsize=(10,4))
```

<Axes: xlabel='Date'>



```
#Normalização dos dados | Multiplicação por R$1000
carteira_normalizada = (carteira/carteira.iloc[0])*1000
carteira_normalizada
```

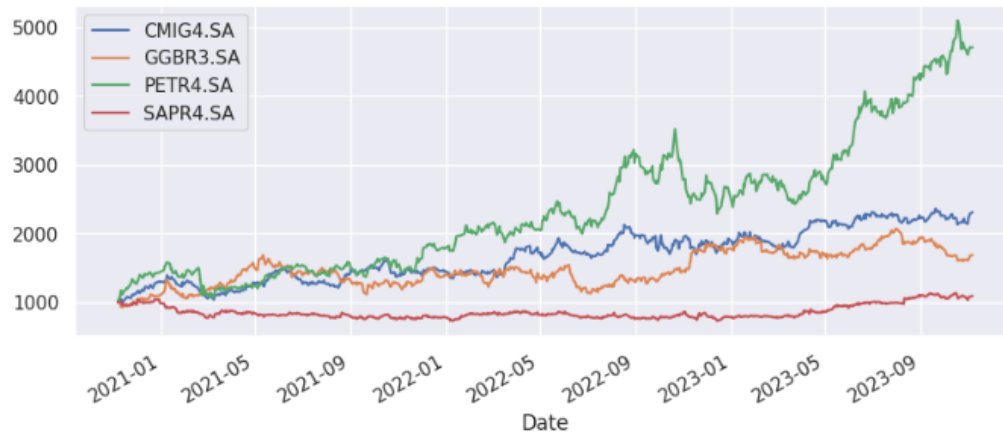


	CMIG4.SA	GGBR3.SA	PETR4.SA	SAPR4.SA
Date				
2020-11-06	1000.000000	1000.000000	1000.000000	1000.000000
2020-11-09	1021.428574	979.657333	1094.177280	1003.853636
2020-11-10	1043.749889	933.083624	1168.607624	1003.853636
2020-11-11	1008.035715	942.184123	1158.481100	990.366135
2020-11-12	984.821398	921.841532	1109.366940	953.757042
...	...	...	...	...
2023-10-30	2143.159544	1617.424402	4647.235913	1053.062030
2023-10-31	2137.696821	1602.248525	4602.194270	1029.450796
2023-11-01	2228.740128	1617.424402	4652.534543	1062.506523
2023-11-03	2285.187055	1673.335430	4708.174458	1079.034443
2023-11-06	2305.216521	1682.920081	4704.199853	1088.478936

745 rows x 4 columns

```
[ ] carteira_normalizada.plot(figsize=(10,4))
```

<Axes: xlabel='Date'>



```
[ ] carteira_normalizada['saldo'] = carteira_normalizada.sum(axis =1)
```

carteira_normalizada

	CMIG4.SA	GGBR3.SA	PETR4.SA	SAPR4.SA	saldo
Date					
2020-11-06	1000.000000	1000.000000	1000.000000	1000.000000	4000.000000
2020-11-09	1021.428574	979.657333	1094.177280	1003.853636	4099.116823
2020-11-10	1043.749889	933.083624	1168.607624	1003.853636	4149.294773
2020-11-11	1008.035715	942.184123	1158.481100	990.366135	4099.067072
2020-11-12	984.821398	921.841532	1109.366940	953.757042	3969.786911
...	...	...	...	...	...
2023-10-30	2143.159544	1617.424402	4647.235913	1053.062030	9460.881888
2023-10-31	2137.696821	1602.248525	4602.194270	1029.450796	9371.590412
2023-11-01	2228.740128	1617.424402	4652.534543	1062.506523	9561.205596
2023-11-03	2285.187055	1673.335430	4708.174458	1079.034443	9745.731385
2023-11-06	2305.216521	1682.920081	4704.199853	1088.478936	9780.815391

745 rows × 5 columns

```
[ ] ibov_normalizado = (ibov/ibov.iloc[0])*4000
      ibov_normalizado
```

```
Date
2020-11-06    4000.000000
2020-11-09    4123.572654
2020-11-10    4180.636713
2020-11-11    4148.136390
2020-11-12    4054.603716
...
2023-10-30    4465.599857
2023-10-31    4489.885812
2023-11-01    4565.640532
2023-11-03    4688.935406
2023-11-06    4701.911725
Name: Adj Close, Length: 745, dtype: float64
```

```

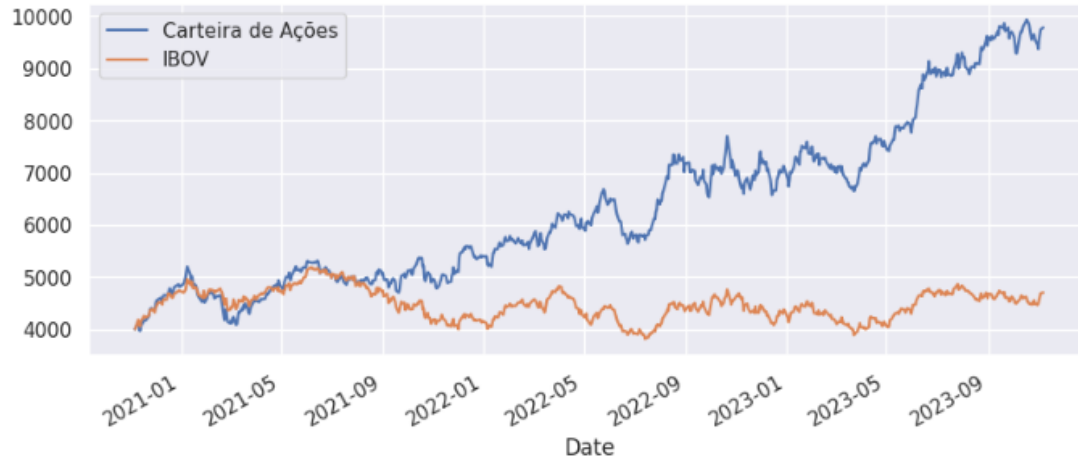
▶ carteira_normalizada['saldo'].plot(figsize=(10,4), label = 'Carteira de Ações')
ibov_normalizado.plot(label='IBOV')
plt.legend()

```

```

↳ <matplotlib.legend.Legend at 0x7cc595964fa0>

```



```

[ ] retorno = carteira.pct_change()
    retorno

```

	CMIG4.SA	GGBR3.SA	PETRA.SA	SAPR4.SA
Date				
2020-11-06	NaN	NaN	NaN	NaN
2020-11-09	0.021429	-0.020343	0.094177	0.003854
2020-11-10	0.021853	-0.047541	0.068024	0.000000
2020-11-11	-0.034217	0.009753	-0.008665	-0.013436
2020-11-12	-0.023029	-0.021591	-0.042395	-0.036965
...	...	...	...	...
2023-10-30	-0.013412	0.003469	-0.010158	-0.021930
2023-10-31	-0.002549	-0.009383	-0.009692	-0.022422
2023-11-01	0.042589	0.009472	0.010938	0.032110
2023-11-03	0.025327	0.034568	0.011959	0.015556
2023-11-06	0.008765	0.005728	-0.000844	0.008753

745 rows x 4 columns

```
[ ] #Volatilidade dos ativos
    retorno.plot()
```

<Axes: xlabel='Date'>



```
▶ retorno_acumulado = (1+retorno).cumprod()
   retorno_acumulado
```



	CMIG4.SA	GGBR3.SA	PETR4.SA	SAPR4.SA
Date				
2020-11-06	NaN	NaN	NaN	NaN
2020-11-09	1.021429	0.979657	1.094177	1.003854
2020-11-10	1.043750	0.933083	1.168607	1.003854
2020-11-11	1.008036	0.942184	1.158481	0.990366
2020-11-12	0.984821	0.921841	1.109367	0.953757
...	...	...	...	...
2023-10-30	2.143159	1.617424	4.647235	1.053062
2023-10-31	2.137697	1.602248	4.602194	1.029451
2023-11-01	2.228740	1.617424	4.652534	1.062507
2023-11-03	2.285187	1.673335	4.708174	1.079035
2023-11-06	2.305216	1.682920	4.704199	1.088479

745 rows × 4 columns

CRUZAMENTO DE MÉDIAS MÓVEIS (GERDAU, CEMIG e PETROBRÁS)

```
[ ] data = yf.download('CMIG4.SA', '2020-11-03', '2023-10-31')
data.head()
```

```
[*****100%*****] 1 of 1 completed

      Open      High      Low      Close  Adj Close  Volume
Date
2020-11-03  7.175087  7.244078  7.071600  7.126793  5.065289  15743886
2020-11-04  7.181986  7.499346  7.175087  7.354464  5.227104  19386668
2020-11-05  7.457951  7.651126  7.451052  7.623530  5.418339  20225035
2020-11-06  7.582135  7.789109  7.526942  7.727017  5.491892  11423916
2020-11-09  7.899495  8.002982  7.823604  7.892596  5.609575  17241757
```

```
def crossover_sinal(data):
    #criação de sinais de compra e venda
    data['MM9'] = data['Close'].rolling(9).mean()
    data['MM21'] = data['Close'].rolling(21).mean()

    # Identificação dos momentos de cruzamentos

    data['Anterior'] = data['MM9'].shift(1) - data['MM21'].shift(1)
    data['Atual'] = data['MM9'] - data['MM21']

    data.loc[(data['Anterior']<0)&(data['Atual']>0), 'Compra'] = data['Close']
    data.loc[(data['Anterior']>0)&(data['Atual']<0), 'Venda'] = data['Close']

    columns = ['Close', 'MM9', 'MM21', 'Compra', 'Venda']

    return data[columns]
```

```
output_df = crossover_sinal(data)
output_df.tail(10)
```

```

      Close      MM9      MM21  Compra  Venda
Date
2023-10-17  12.01  12.255555  12.478571    NaN    NaN
2023-10-18  11.68  12.196667  12.425714    NaN    NaN
2023-10-19  11.78  12.150000  12.371905    NaN    NaN
2023-10-20  11.81  12.115556  12.308095    NaN    NaN
2023-10-23  11.91  12.064444  12.258571    NaN    NaN
2023-10-24  11.93  12.002222  12.212857    NaN    NaN
2023-10-25  11.85  11.927778  12.160952    NaN    NaN
2023-10-26  12.16  11.928889  12.145714    NaN    NaN
2023-10-27  11.93  11.895556  12.124762    NaN    NaN
2023-10-30  11.77  11.868889  12.094762    NaN    NaN
```

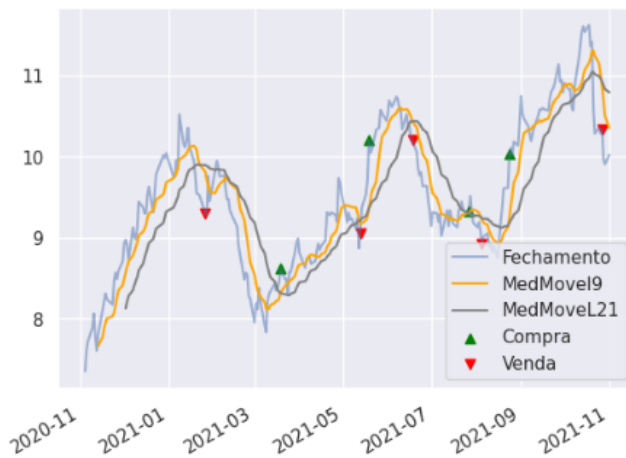
```
[ ] #representação gráfica de um período específico
filtered_data1 = output_df.loc[(output_df.index<'2021-11-03')&(output_df.index>'2020-11-03')]
```

```
fig, ax = plt.subplots()
ax.plot(filtered_data1.index, filtered_data1['Close'],label='Fechamento',alpha=0.5)
ax.plot(filtered_data1.index, filtered_data1['MM9'],label='MedMove19',color='orange')
ax.plot(filtered_data1.index, filtered_data1['MM21'],label='MedMoveL21',color='grey')

# Criação de sinais para compra e venda
ax.scatter(filtered_data1.index, filtered_data1['Compra'],label='Compra',marker= '^',color='green')
ax.scatter(filtered_data1.index, filtered_data1['Venda'],label='Venda',marker= 'v',color='red')

fig.autofmt_xdate()
ax.legend()
```

<matplotlib.legend.Legend at 0x7cc5873db700>



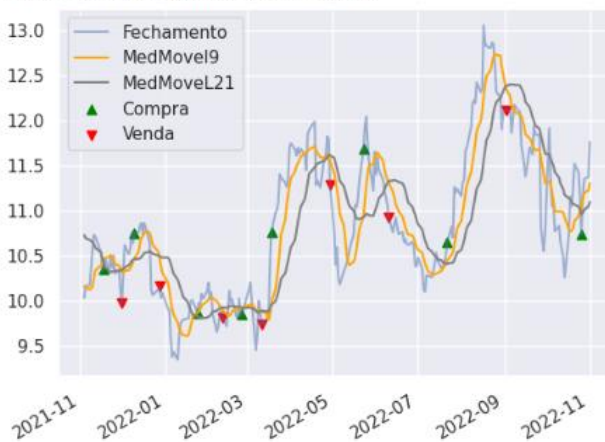
```
[ ] filtered_data2 = output_df.loc[(output_df.index<'2022-11-03')&(output_df.index>'2021-11-03')]
```

```
fig, ax = plt.subplots()
ax.plot(filtered_data2.index, filtered_data2['Close'],label='Fechamento',alpha=0.5)
ax.plot(filtered_data2.index, filtered_data2['MM9'],label='MedMove19',color='orange')
ax.plot(filtered_data2.index, filtered_data2['MM21'],label='MedMoveL21',color='grey')

ax.scatter(filtered_data2.index, filtered_data2['Compra'],label='Compra',marker= '^',color='green')
ax.scatter(filtered_data2.index, filtered_data2['Venda'],label='Venda',marker= 'v',color='red')

fig.autofmt_xdate()
ax.legend()
```

<matplotlib.legend.Legend at 0x7cc59a3cb880>



```
[ ] filtered_data3 = output_df.loc[(output_df.index<'2023-11-03')&(output_df.index>'2022-11-03')]
```

```
fig, ax = plt.subplots()
ax.plot(filtered_data3.index, filtered_data3['Close'],label='Fechamento',alpha=0.5)
ax.plot(filtered_data3.index, filtered_data3['MM9'],label='MedMove19',color='orange')
ax.plot(filtered_data3.index, filtered_data3['MM21'],label='MedMoveL21',color='grey')

ax.scatter(filtered_data3.index, filtered_data3['Compra'],label='Compra',marker='^',color='green')
ax.scatter(filtered_data3.index, filtered_data3['Venda'],label='Venda',marker='v',color='red')

fig.autofmt_xdate()
ax.legend()
```

```
<matplotlib.legend.Legend at 0x7cc5876604f0>
```

