Criação de uma Carteira Digital de Investimentos utilizando *Python*

Juliano dos Santos[1], Fábio José Parreira[2], Sidnei Renato Silveira[2]

[1] Curso de Bacharelado em Sistemas de Informação
[2] Departamento de Tecnologia da Informação (DTecInf)

Universidade Federal de Santa Maria (UFSM) - Campus Frederico Westphalen
Linha 7 de Setembro, s/n, CEP: 98400-000, BR 386 Km 40- Frederico Westphalen – RS
juliano431santos@hotmail.com,
fabiojparreira@gmail.com,sidneirenato.silveira@gmail.com

Resumo. *Este artigo apresenta o estudo e o desenvolvimento de uma carteira digital, utilizando a linguagem de programação Python, aliado a suas ferramentas para análise de dados. Foi criada uma carteira de investimentos diversificada em igual proporção entre as diferentes classes de ativos, sendo ações, fundos imobiliários e ETF, tendo como base a taxa de risco e o retorno financeiro, obtidos nas abordagens na teoria de Markowitz, Critério de Kelly e o Modelo KMV-Merton.*

Palavras-Chave: *Investimentos, Carteira Digital, Ativos.*

Abstract. *This paper presents a proposal for the development of a digital portfolio, using Python combined with its tools for data analysis, through an investment portfolio diversified in equal proportion between the different asset classes. Stocks, real estate funds and ETF, based on the financial risk and return rate, obtained in the approaches in the Markowitz theory, Kelly criterion and the KMV-Merton model.*

Keywords: *investments, Digital Wallet, assets*

1. Introdução

Este artigo apresenta o estudo e o desenvolvimento de uma carteira digital de investimentos, utilizando recursos de *Data Science*. A partir do interesse de um dos autores deste trabalho no mercado financeiro e de pesquisas realizadas na *web*, foram encontrados materiais sobre computação quântica, além de informações sobre o investidor Jim Simons, matemático e gestor de *Hedge Fund (*Fundos de cobertura - são fundos americanos de investimento) com o seu fundo *Medallion Fund*. A estratégia deste investidor consiste em aplicar modelos quantitativos, *Data Science* e *Machine Learning* no mercado financeiro. Os investimentos do referido investidor têm obtido altos retornos, superando o lendário investidor Warren Buffett, referência em análise fundamentalista que gere um conjunto de outras empresas subsidiarias, aplicando suas estratégias de investimentos e análises de empresas (MEIRELLES, 2021).

Neste sentido, entendemos que, com educação financeira e conhecimento, por meio da poupança mensal de parte dos recursos e com aportes mensais em ações, FII

(Fundos de Investimento Imobiliário) e ETFs (*Exchange Traded Fund*), é possível constituir um plano de aposentadoria por meio do mercado financeiro, com uma carteira previdenciária, com ações que pagam dividendos recorrentes.

Neste contexto, o objetivo geral deste trabalho foi o de desenvolver um protótipo para otimizar o desempenho de carteiras de investimento, constituídas por múltiplos ativos tais como ações, fundos imobiliários e ETF, tendo como base a taxa de risco e o retorno financeiro, obtidos nas abordagens da teoria de *Markowitz*, Critério de *Kelly* e modelo *KMV-Merton*.

Para dar conta desta proposta, este artigo está estruturado como segue: a seção 2 apresenta o referencial teórico, destacando conceitos como otimização de carteiras de investimentos, teoria de *Markowitz*, Critério de *Kelly*, carteiras de investimentos e o Modelo KMV Merton (COLMAN, WIENANDTS, DE PIETRO, 2013; INFOMONEY, 2022; EDWARD, 2008; POUNDSTONE, WILLIAM, 2005). A seção 3 apresenta alguns trabalhos relacionados. A carteira digital implementada é apresentada na seção 4. Encerrando o artigo são apresentadas as considerações finais e as referências empregadas.

2. Referencial Teórico

Esta seção apresenta um referencial teórico sobre as áreas envolvidas no desenvolvimento deste trabalho, destacando conceitos referentes a carteiras de investimento e critérios para a otimização de carteiras.

É consenso geral que o funcionamento do mercado financeiro é bastante complexo, já que os fatos e agentes que o influenciam não são claramente compreendidos ou seus efeitos facilmente mensurados. O principal fenômeno que explica a dificuldade em tornar esse mercado algo palpável é o fato de ele tratar de futuro ou perspectivas dele.

Harry Markowitz, na década de 50, trouxe à tona o conceito de risco e uma maneira de mensurá-lo e propôs que a diversificação, ou a composição de uma carteira de investimentos com vários ativos minimizaria o risco dos ativos individuais. Sendo assim, uma carteira composta de dois ativos correlacionados negativamente faria com que a perda de uma seja compensada pelos ganhos do outro. Essa diversificação dá ao investidor uma gama maior de opções de escolha (COLMAN, WIENANDTS, DE PIETRO, 2013).

2.1 Carteiras de Investimento

Carteira de investimentos é o conjunto de aplicações realizadas ou de ativos mantidos por uma pessoa. Ela pode incluir renda fixa e renda variável, tais como títulos públicos, LCA (Letras de Crédito do Agronegócio), ações, fundos, Fundos imobiliários, *stocks* (ações no exterior) (INFOMONEY, 2022).

A composição de uma carteira de investimentos é bastante particular. O conjunto ideal de ativos para uma pessoa pode ser completamente diferente do que vale para a outra. As melhores aplicações para cada um dependem de fatores como os objetivos para o dinheiro, o volume de patrimônio disponível e o perfil de risco.

Uma carteira de investimentos, em tese, tem que ser bem diversificada, em diferentes classes de ativos, países e moedas diferentes para tentar fugir de riscos sistêmicos (taxa de juros, crises políticas, insegurança jurídica do país) e riscos não sistêmicos (crédito, liquidez, mercado).

2.2 Critérios para a Otimização de Carteiras

Na literatura, existem diversos economistas que desenvolveram estudos para calcular o risco e retorno sobre uma carteira de investimentos. Em geral, os trabalhos científicos pesquisados focam, em sua grande maioria, no modelo de *Markowitz*, por ser um modelo para seleção de carteira que leva em consideração o risco, tratado neste caso como sendo a variância da carteira (COLMAN, WIENANDTS, DE PIETRO, 2013).

Entre os critérios para a otimização de carteiras de investimentos, o critério de *Kelly* é uma ferramenta para conseguir maximizar os lucros ou rendimentos e mitigar o risco de falência. Ele permite ao utilizador estimar a percentagem do capital a aplicar em um determinado investimento, no sentido de maximizar o rendimento e mitigar o risco (EDWARD, 2008; POUNDSTONE; WILLIAM, 2005).

O modelo KMV *Merton* fala sobre a avaliação do risco de crédito. É uma ferramenta utilizada pelas áreas de gerenciamento para medir a probabilidade de *default* (Calote na Dívida) de empresas, que surge quando o tomador não possui capacidade ou vontade de cumprir o contrato assumido. O modelo KMV possibilitou o cálculo da PD (*Probability of Default*) diária de cada um dos ativos em análise que, como o próprio nome sugere, descreve a chance de a firma não arcar com suas obrigações no curto prazo (LOPES, 2013).

2.2.1 Modelo de Markowitz

O modelo de *Markowitz* (Modelo Média-Variância) (MARKOWITZ, 1952) é uma metodologia de sistema fechado que busca, sob a estratégia de diversificação de carteira, os pontos ótimos e de maior eficiência, por meio da relação retorno esperado e risco associado (ELTON *et al*., 2014). Harry Markowitz trouxe à tona o conceito de risco e uma maneira de mensurá-lo (fato que antes era pouco ou nada explorado) e propôs que a diversificação minimizaria o risco dos ativos individuais. Sendo assim, uma carteira composta de dois ativos correlacionados negativamente faria com que a perda de um seja compensada pelos ganhos do outro.

Essa diversificação dá ao investidor uma gama maior de opções de escolha que, segundo outra hipótese de Markowitz, será feita com base na composição que gerar um maior retorno dentro de níveis de risco aceitáveis.

A diversificação, porém, só minimizará o chamado “risco diversificável”, que se refere ao conjunto restrito de empresas, enquanto o risco “não diversificável” refere-se a acontecimentos que afetam o mercado como um todo.

Para o desenvolvimento do método, Markowitz realizou uma série de suposições sobre o comportamento do investidor e do mercado, como a racionalidade do investidor ou a eficiência sem falhas do mercado (MANGRAM, 2013). Muitos estudiosos em finanças criticam o modelo pela perspectiva de que as suposições não são adequadas ao mundo real.

A teoria de Harry Markowitz é baseada no pressuposto de que um investidor é avesso ao risco (investidor conservador). Assim, irá escolher um portfólio de investimentos baseado no retorno esperado e no risco das ações em conjunto na carteira, A diversificação dos ativos muda a atratividade de um investimento devido à correlação de um novo ativo com o resto da carteira.

Considerando o risco de uma carteira dado pela sua variância, a adição de um novo ativo no portfólio pode diminuir o risco da carteira. Com isto, um investidor racional escolherá investir em uma combinação de ações que, dado um determinado risco possui um maior retorno.

Harry Markowitz foi o vencedor do prêmio de economia de 1990 pela sua contribuição à área de seleção de portfólio. Markowitz propôs um modelo de planejamento de carteira de investimentos em sistema fechado que se tornou uma das principais metodologias estudadas (OGATA, 2021).

A partir dos valores de retorno esperado da carteira e do risco associado podemos definir a Fronteira Eficiente como o subconjunto das carteiras consideradas ótimos com os maiores retornos, considerando um determinado potencial de risco. A Figura 1 ilustra um exemplo de uma fronteira eficiente.

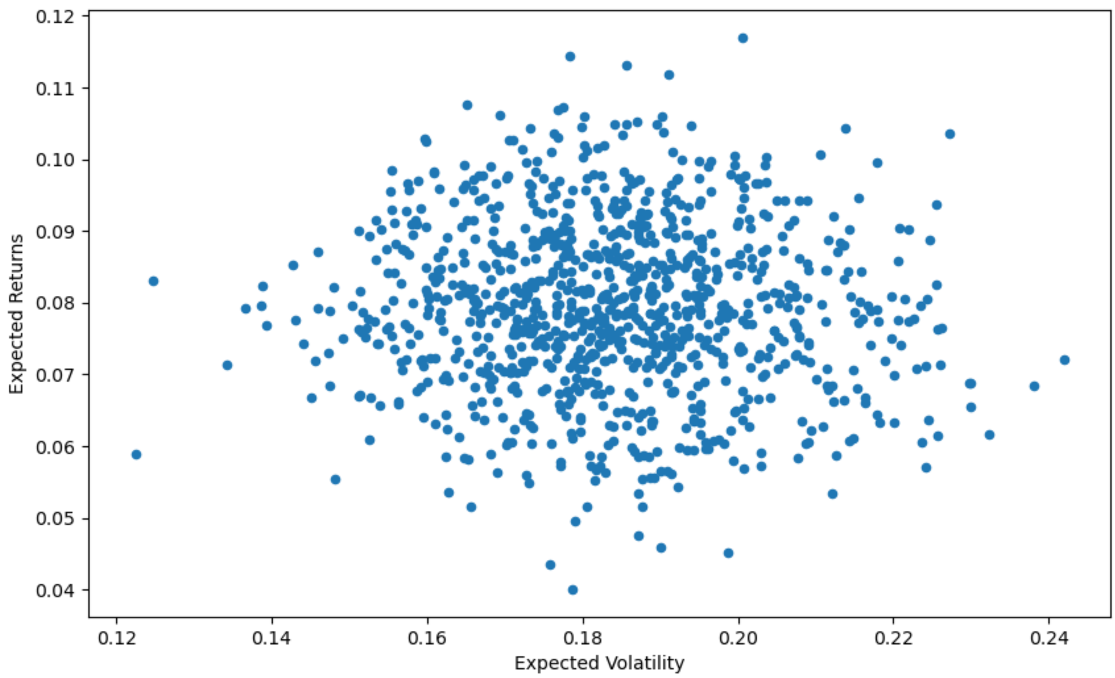

Figura 1: Fronteira Eficiente, Fonte: UDEMY, 2022

Mesmo que a contribuição dos estudos de Markowitz tenha um enorme impacto teórico, muitos estudiosos a criticam. Mangram (2013) mostra que as suposições sobre o perfil dos investidores e do mercado não se alinham ao mundo real. Por exemplo: as crises que o mercado sofre por ações externas, como o cenário mais recente da pandemia, demonstram que o mercado financeiro está longe de ser eficiente. Outro exemplo são os investidores que realizam aportes em ações que estão em alta no mercado, demonstrando que nem todo investidor opera sobre a racionalidade.

Dadas as curvas de indiferença do investidor, agora podemos analisar quais são as combinações de média e desvio padrão que um portfólio pode assumir. Desse modo, a escolha individual de investimento será dada no ponto em que a curva de indiferença do investidor tangencia a fronteira eficiente de investimento. A fronteira eficiente é construída maximizando o retorno, dado um determinado risco. Sendo assim, todas as combinações de investimento abaixo desta fronteira são consideradas como ineficientes, não sendo racional o investimento, uma vez que o investidor pode escolher outra combinação de ativos que o levar a um retorno maior sem aumentar seu risco (COLMAN, WIENANDTS, DE PIETRO, 2013).

2.2.2 Critério de Kelly

Desenvolvido no contexto das apostas, o critério de *Kelly* (KELLY, 1956) formulou uma estratégia para alocação do tamanho do aporte em apostas, popularizado como o "Sistema Cientifico de Apostas".

O critério utiliza as probabilidades do apostador vencer ou perder em cada partida para adequar o tamanho da fração a ser investida, impedindo que o jogador arrisque desnecessariamente parte de sua fortuna em cenários desfavoráveis. Por si só essa é uma propriedade, de grande interesse para uma ampla gama de investidores.

Esse critério, além de maximizar a soma dos retornos, maximiza a utilidade dos investidores com funções de utilidades logarítmicas. Isso implica, de certa forma, que investidores que gostam de usar estratégias que usam expressões logarítmicas estejam "mais satisfeitos". Sob condições favoráveis, também maximiza a mediana dos retornos logarítmicos, algo útil para manter o investidor na estratégia.

O investidor que utiliza a estratégia *Kelly*, sob os pressupostos do modelo, está fora de risco de uma falência, apesar de teoricamente poder atingir níveis arbitrariamente pequenos em cenários adversos. Quanto maior a probabilidade de vitória, maior será o tamanho do investimento a ser alocado. A Figura 2 ilustra um exemplo do Critério de *Kelly*.

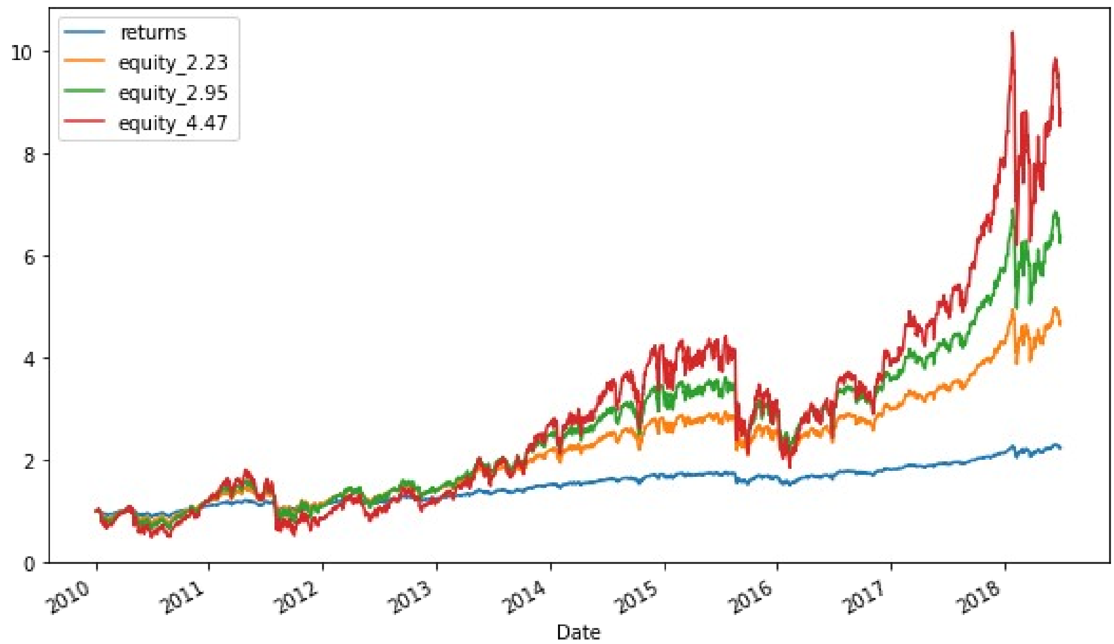

Figura 2: Critério de *Kelly*, Fonte: Quant Platform, 2022

2.2.3 Modelo KMV Merton

Robert Merton desenvolveu um modelo de avaliação do risco de crédito de empresas, suportado na teoria de avaliação de opções financeiras de Black e Scholes (1973). Este modelo encara o valor do capital próprio como uma opção (*call option*) sobre o valor dos ativos da empresa, com um preço de exercício igual ao valor facial da dívida para um determinado período de tempo (LOPES, 2013).

Nesta perspectiva, é possível calcular o valor da empresa transpondo a metodologia de avaliação de opções financeiras para a avaliação de empresas. A implementação do modelo de Merton recebeu muita atenção da indústria, e diversos modelos, baseados em sua teoria, foram desenvolvidos.

Um deles foi o modelo KMV-Merton, entendido como uma extensão ao modelo proposto por Black Scholes (1973) e Merton (1974). Esse modelo se torna especialmente importante porque criou o conceito de *Distance to Default*, adicionando uma etapa ao modelo de Merton e elaborando uma definição de "ponto de *default*" mais realista (que não fosse somente o valor de face da dívida).

A DD (*Distance to Default*) pode ser caracterizada como o número de desvios padrões entre o valor de mercado dos ativos e seu ponto de *default* e é inversamente proporcional ao PD (ponto de *Default*). Quanto maior a DD menor a PD.

Como é comum nos modelos de avaliação de empresas, um dos fatores mais importantes é poder obter previsões futuras para os ativos avaliados. Neste sentido testa-se o cálculo do valor da empresa para o futuro por meio da aplicação do modelo clássico de opções financeiras, transposto para a avaliação de empresas.

Calcular a probabilidade de não cumprimento das empresas para com suas dívidas, o não cumprimento das empresas é, sem dúvida, um fator importante para todos os *stakeholders* da organização, gestores, detentores de capital, credores e investidores.

Procura-se fazer a ligação da avaliação dos ativos das empresas com o risco de crédito associado.

Nos anos 80, uma empresa norte-americana especializada na análise de risco de crédito, a KMV *Corporation*, aplicou o modelo de Merton (1974) com algumas variantes para, depois de calcular o valor estimado da empresa e da dívida, atribuir probabilidades de incumprimento (EDF – *Estimated Default Frequency*), por meio de uma metodologia própria.

Rangarajan (2001) descreve o modelo de Merton (1974) e a sua variante com maior sucesso comercial, o modelo KMV-Merton. Este difere do modelo de Merton, tendo como principal objetivo o cálculo de uma probabilidade da distância até a insolvência, a DD e a consequente atribuição de uma probabilidade estimada de incumprimento, a EDF. A EDF é atribuída tendo por base uma distribuição de probabilidades definida pela experiência passada na KMV.

Merton analisa o risco de crédito das empresas tendo em conta uma estrutura de dívida simplificada (D). Esta simplificação implica a homogeneidade da dívida, ou seja, a empresa tem uma única emissão de dívida de um valor certo, a pagar em um período de tempo determinado. Corresponde à emissão de dívida/obrigações de culpa zero e o valor a pagar aos credores é único, conhecido e é liquidado no final do prazo contratualizado. Com esta simplificação da dívida o risco de crédito da empresa pode ser corretamente avaliado e ser assim determinado o valor da empresa. A Figura 3 apresenta um exemplo de retorno diário sobre o patrimônio líquido.

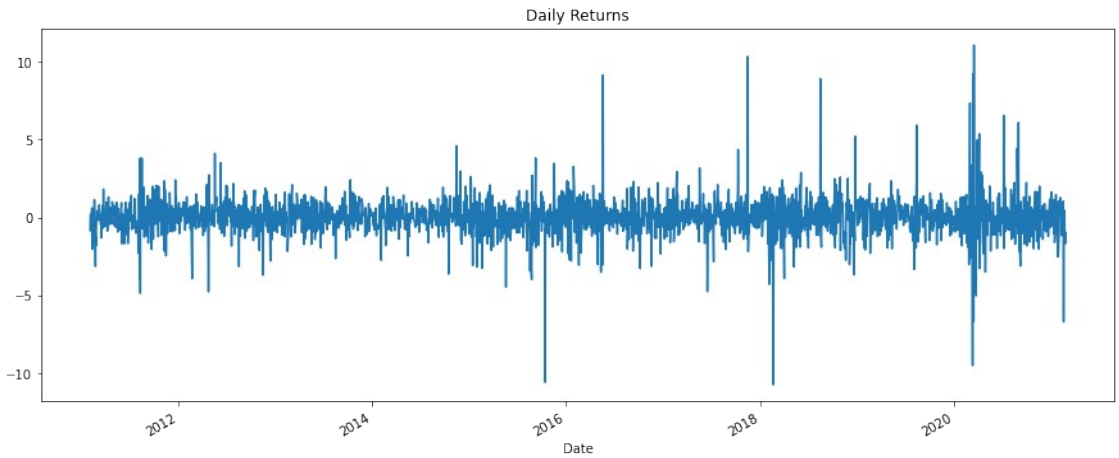

Figura 3: KMV-Merton, Fonte: FONSECA, 2022

3. Trabalhos Relacionados

Nessa seção apresentamos alguns trabalhos relacionados ao trabalho desenvolvido. No final da seção apresentamos um estudo comparativo entre os trabalhos estudados e a carteira digital implementada.

3.1 Relação Risco x Retorno nos Fundos de Pensão Brasileiros Análise com Base no Índice de Sharpe e Alfa de Jensen

Os planos de benefícios oferecidos pelas EFPCs (Entidades Fechadas de Previdência Complementar) têm se tornado uma importante opção de planejamento previdenciário para os trabalhadores que vêm reduzir a capacidade de o governo garantir, por meio do INSS (Instituto Nacional de Seguridade Social), uma aposentadoria tranquila. Nesse sentido, a atuação dos administradores das EFPCs é extremamente importante no sentido de garantir que os recursos acumulados pelos participantes durante seu período laboral sejam bem geridos e permitam o pagamento dos benefícios futuros aos trabalhadores. Por isso, os gestores dos investimentos dos planos de benefícios devem observar os limites legais estabelecidos, no sentido de não correr riscos que possam prejudicar os benefícios futuros esperados pelos participantes.

Nesse sentido, o trabalho de Figueiredo (2018) analisou os retornos obtidos por 831 planos de benefícios entre 2010 e 2015, com o objetivo de verificar se esses são essencialmente explicados pelos riscos assumidos nesses investimentos. Para tanto, foram utilizados o índice de *Sharpe*, como medida dos retornos após deduzidos os riscos incorridos e o *Alfa de Jensen* como medida do retorno adicional oferecido pelo plano a seus participantes. Os resultados apontam que a maior parte do retorno obtido pelos planos no período analisado foi decorrente dos riscos incorridos pelos gestores dos investimentos e que a atuação da administração na maioria dos planos gerou retorno negativo em relação aos retornos de mercado de amostra.

3.2 Risco de Crédito e o Retorno Acionário: Análise Empírica da Correlação Para o Mercado Brasileiro.

O trabalho de Rodrigues (2018) teve, como objetivo, o de analisar a evolução do risco de crédito das empresas brasileiras e quantificar a relação existente com o retorno acionário das mesmas. Para isso, foram analisadas as probabilidades de *default* e *distance to default* por meio de modelos elaborados tendo como base os resultados encontrados por Robert C. Merton (1974) e a evolução do retorno das ações no mesmo período. Foi, então, analisada empiricamente e estatisticamente a existência dessa relação. Os resultados encontrados não indicam a existência de uma anomalia, como proposto por diversos estudos na literatura, ou seja, foi encontrada uma relação positiva entre risco de crédito e retorno. Contudo, não foi possível encontrar um alfa estatisticamente significante para uma estratégia de compra e venda de portfólios de risco, o que indica que os mesmos estariam integralmente explicados pelos fatores do modelo de precificação propostos por Fama e French (1994).

3.3 Aplicação de Estratégias Quantitativas com Ativos do Mercado Financeiro Brasileiro

A aplicação de técnicas quantitativas para a tomada de decisões de investimentos é um assunto de constante estudo no mercado financeiro global, tanto para pesquisadores quanto para gestores profissionais. O trabalho de Meirelles (2021) apresenta um modelo de seguidor de tendências que busca uma previsão de altas e baixas no preço dos ativos para identificar um bom momento de compra ou venda. Um método de *Pairs Trading* que envolve a busca por arbitragens estatísticas por meio de distorções nos preços entre diversos ativos, de modo que, quando tais distorções são corrigidas, é possível obter lucros com risco reduzido, e aplicação de inteligência artificial através de redes neurais artificiais por meio do treinamento de rede para interpretar padrões de acordo com os dados inseridos.

Este projeto teve o intuito de desenvolver as estratégias mencionadas para buscar melhores retornos de investimentos e obter uma diversificação maior nas carteiras dos investidores com a crescente demanda do mercado brasileiro por fundos de investimentos quantitativos que ainda representam uma pequena parte do mercado de fundos total no país, e menor ainda quando comparada com o mercado quantitativo *offshore*, evidenciando o seu potencial de crescimento. Trabalhando com os ativos do mercado de ações brasileiro e com dados de janeiro de 2010 a janeiro de 2020, foram gerados os rendimentos para a estratégia do modelo de seguidor de tendências para 17 ativos analisados, os rendimentos da estratégia de *Pairs Trading* de 14 pares com maiores índices de cointegração, dadas todas as combinações geradas com os 17 ativos, e os relatórios das estratégias de 4 modelos de Redes Neurais Artificiais (RNA) utilizadas em apenas 5 ativos.

Os resultados obtidos mostram que, na maioria dos casos, a estratégia de Seguidor de Tendências não apresentou um retorno maior do que o do ativo durante o período analisado. A estratégia de *Pairs Trading* apresentou resultados mistos, visto que para casos de queda dos preços dos ativos no período a estratégia reportou ganhos, evidenciando uma boa diversificação, sendo possível obter lucros em mercados de baixa. Por fim, nos modelos de RNA observou-se qualitativamente que a estratégia depende dos dados de entrada e de outros parâmetros configurados para o treinamento da rede e que uma rede que apresenta um bom desempenho para um ativo específico não necessariamente apresenta o mesmo para outros.

3.4 Estudo Comparativo

Esta seção apresenta algumas características que permitem comparar os trabalhos estudados ao trabalho desenvolvido, como mostra o Quadro 1.

Quadro 1 – Estudo Comparativo

Características	**Trabalho 1 (FIGUEIREDO, 2018)**	**Trabalho 2 (RODRIGUES, 2018)**	**Trabalho 3 (MEIRELLES, 2021)**	**Trabalho Desenvolvido**
Estratégia	*Risco x retorno*	*Risco Crédito*	*Estratégia Quantitativa*	*Inovações no mercado financeiro para simulação de portfólios*
Finanças	*Outros métodos de otimização*	*Tema complementar*	*Linguagem de Programação*	*Análise Quantitativa de investimentos*
Mercado Financeiro	*Carteira Previdenciária*	*Análise Empírica*	*Algoritmos de Negociação*	*Análise de Dados*
Informação	*Fundos de*	*Precificação*	*Redes Neurais*	*Melhor capacidade*

	Pensão			*na Análise de investimentos*

Analisando as informações apresentadas no Quadro 1, a proposta de implementação de uma carteira digital tem características semelhantes aos trabalhos relacionados estudados. Em relação ao trabalho de Figueiredo (2018). O tema também trata de taxas de risco e retorno, como no trabalho desenvolvido. O trabalho de Rodrigues (2018) refere-se ao risco de crédito das empresas, as empresas selecionadas são escolhidas de forma criteriosa sem empresas com muita alavancagem operacional. O trabalho de Meirelles (2021) se assemelha ao trabalho proposto, pois trata do tema mais relacionado à técnica aplicada ao mercado financeiro, aplica Redes Neurais Artificiais e algoritmos de negociação. A carteira digital implementada neste trabalho tenta, por meio de uma relação de informações, unir os temas estudados para agregar conhecimento ao trabalho.

A implementação da carteira digital foi realizada com a instalação do serviço *Anaconda navigator,* que é uma interface de usuário que permite iniciar aplicativos de forma fácil, sem a necessidade de utilizar linhas de comando, utilizando o *Jupyter Lab* para a execução dos códigos. E o *Jupyter Notebook* utilizando a linguagem de programação *Python* na versão 3.7 e suas diversas bibliotecas para *Data Science* foram empregadas na organização dos dados e plotagem dos gráficos.

4. Solução Desenvolvida

Este trabalho pretendeu desenvolver uma aplicação para a análise e a otimização de uma carteira de investimentos, considerando que uma carteira de investimentos é uma seleção de diferentes classes de ativos, diversificadas em setores com empresas de diferentes setores da atividade econômica. Para compor a carteira digital implementada neste trabalho, foram selecionados dez ativos negociados no mercado financeiro brasileiro. Os ativos foram escolhidos baseados na montagem de uma carteira de investimentos diversificada, porém apenas ativos negociados na bolsa brasileira, com um alto valor de mercado, empresas pagadoras de dividendos e com maior liquidez nas negociações (é a facilidade com que um ativo pode ser convertido em dinheiro novamente), as chamadas *Small Caps*.

A carteira digital foi composta por três classes de ativos: Ações, Fundos de Investimentos Imobiliários (FII) e ETF (*Exchange Traded Fund*). A porcentagem de ativos na carteira é igual para as três classes de ativos: 33% para cada, formando uma carteira diversificada.

As ações foram selecionadas de diferentes setores: Aliansce Sonae, ALSO3 do setor de *shopping centers*, do setor financeiro de intermediários financeiros Banrisul BRSR6, GRND3, Grendene do setor de calçados, Camil CAML3 no setor de alimentos, CSMG3 uma empresa do setor de saneamento, TAEE11 do setor de utilidade pública, energia elétrica. Já os FII selecionados são de setores diferentes com imóveis em cidades diferentes com bons inquilinos (FUNDSEXPLORER, 2022).

Em se tratando de ETF, foi selecionado o IVVB11, que replica o índice S&P 500, e o *iShares* S&P 500, Fundo de Investimento em Cotas de Fundo de Índice –

Investimento no Exterior. Este é um fundo de índice constituído no Brasil que busca retornos de investimentos que correspondam, de forma geral, à performance do Índice S&P 500 em reais (S&P 500 *Brazilian Real Index*). A gestão da carteira do Fundo é realizada pela BlackRock Gestora de Investimentos (BLACKROCK, 2022) sendo, portanto, uma parte dolarizada da carteira de investimentos, possível proteção para o risco Brasil. Além disso foi selecionado o SMAL11 ETF que replica o índice de *small caps* (empresas com baixo e médio valor de mercado).

Aliado com estudos dos métodos de precificação de ativos, foi implementada uma carteira digital, utilizando a linguagem de programação *Python* e a teoria de *Markowitz*, junto com o Critério de *Kelly*. Essas teorias foram estudadas com KMV – Merton descrito nesse trabalho como método para descrever os riscos de *default* (calote) de uma empresa, ele é utilizado para modelagem risco e retorno nos investimentos. A Figura 4 mostra uma simulação dos retornos da carteira de investimentos que foi utilizada no trabalho, a partir da implementação em *Python*.

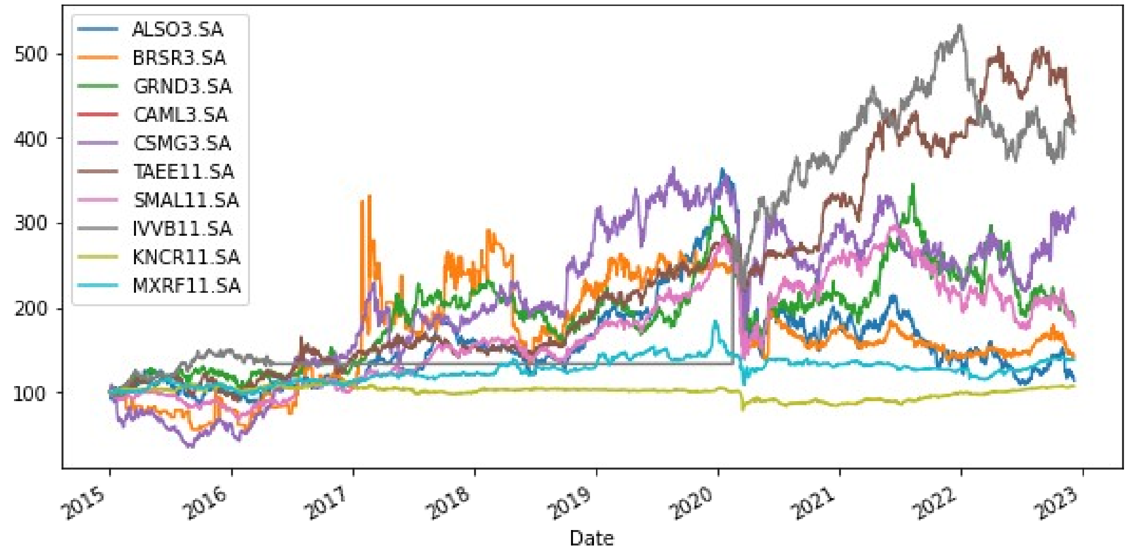

Figura 4: Desempenho das ações. Fonte: Os autores, 2022

O desenvolvimento desta aplicação envolveu, inicialmente, a separação das empresas para realizar a simulação da carteira digital. O estudo das ferramentas que foram utilizadas para a aplicação foi realizado em etapas. Esta parte do trabalho foi baseada em um processo de experimentação com base em dados históricos de ações da Bolsa de Valores de São Paulo e aplicação das metodologias de Markowitz, Critério de Kelly. Foram utilizados dados do S&P 500 e uma descrição simples do Modelo KMV-Merton utilizando a *Stock* (ação negociada nos Estados Unidos) do Wallmart, computando e analisando as métricas financeiras e computacionais para seleção dos portfólios. O detalhamento do desenvolvimento deste projeto foi organizado nas seguintes etapas:

• Etapa 1 - Aquisição dos Dados: foi utilizada a API (*Application Program Interface*) do *Yahoo Finance*, que possui dados de múltiplas ações, classes de ativos, de diferentes Índices e mercados (PYPI.ORG, 2022);

• Etapa 2 - Pré-processamento dos Dados, utilizando a biblioteca *NumPy* para organizar e processar os dados (NUMPY.ORG, 2022);

• Etapa 3 - Análise Exploratória dos Dados: Utilizando o *Pandas*, que é uma biblioteca para manipulação e análise de dados, por meio do pacote pandas que permite organizar os dados em uma tabela e anexar rótulos descritivos as linhas e colunas da tabela facilitando o trabalho com vários formatos de dados é perfeito para trabalhar com séries temporais (PANDAS.PYDATA.ORG, 2022);

• Etapa 4 - Experimentação com Regras de *Markowitz*: a teoria de Harry Markowitz é baseada no pressuposto de um investidor conservador. Assim, irá escolher um portfólio de investimentos baseado no retorno esperado e no risco das ações em conjunto na carteira. A diversificação dos ativos muda a atratividade de um investimento devido à correlação de um novo ativo com o resto da carteira;

Iniciamos a implementação da carteira digital importando as bibliotecas e os módulos para desenvolver a análise. Também importamos o *%matplotlib inline*, que é um recurso fornecido pela linguagem de programação *Python*, disponível exclusivamente para se trabalhar no *Jupyter Notebook.* Também importamos o *JupyterLab*, que facilita a plotagem de gráficos usando o *matplotlib*, logo abaixo das células de código e o armazenamento delas no documento dentro do *Jupyterlab*. O projeto *Jupyter* é uma organização sem fins lucrativos, criada para desenvolver *software* de código aberto, padrões abertos e serviços para computação interativa em dezenas de linguagens de programação (JUPYTER.ORG, 2022).

Os ativos analisados estão representados na Figura 5, que estão armazenados em uma lista chamada *assets*. Utilizamos os dados dos últimos anos, começando em 2015, já que alguns ativos não têm histórico suficiente para anos anteriores. Começamos com a normalização dos dados para base 100 (técnica que ajuda a obter o resultado mais rápido) e plotar os dados da carteira em um gráfico para ver como os 10 ativos estão se comportando no período analisado, como mostrado na Figura 4. Então se quisermos obter uma fronteira eficiente composta por esses ativos, nós vamos precisar de seus retornos logarítmicos, sua média, sua covariância e as matrizes de correlação (para retornar o logaritmo de cada elemento do *array*).

Out[5]:

Date	ALSO3.SA	BRSR3.SA	GRND3.SA	CAML3.SA	CSMG3.SA	TAEE11.SA	SMAL11.SA	IVVB11.SA	KNCR11.SA	MXRF11.SA
2022-12-02	17.780001	11.02	6.59	9.00	15.930000	35.259998	99.209999	234.449997	99.559998	10.01
2022-12-05	17.139999	10.82	6.40	8.73	15.550000	34.700001	95.910004	233.000000	99.750000	10.00
2022-12-06	17.410000	10.96	6.38	8.76	15.780000	34.980000	96.519997	227.500000	99.459999	10.00
2022-12-07	17.389999	11.07	6.50	8.78	16.049999	35.419998	96.099998	225.899994	99.110001	10.00
2022-12-08	16.799999	10.64	6.28	8.53	15.560000	35.270000	93.790001	226.899994	99.190002	10.01

Figura 5: Dados dos últimos anos. Fonte: Os autores, 2022

A Figura 6 mostra o resultado dos retornos logarítmicos obtidos a partir da sua média e de sua covariância e as matrizes de correlação. Podemos ver, facilmente, que os ativos tiveram uma taxa de retorno consistente durante o período de tempo analisado e os seus retornos estão bem correlacionados. Agora, vamos abordar a otimização do portfólio por uma perspectiva mais técnica. Primeiro devemos criar uma variável que vai carregar o número de ativos da carteira. Empregamos essa variável em nossas fórmulas, para que elas possam responder a uma mudança no número de ativos que compõem a carteira. Calculamos o número de elementos da lista chamada *assets* com a ajuda da função *len* que, nesse caso, são todos os ativos da carteira já descritos no texto.

```
Out[8]:  ALSO3.SA     0.016922
         BRSR3.SA     0.042348
         GRND3.SA     0.077951
         CAML3.SA     0.025615
         CSMG3.SA     0.142547
         TAEE11.SA    0.183252
         SMAL11.SA    0.074725
         IVVB11.SA    0.177832
         KNCR11.SA    0.007923
         MXRF11.SA    0.040755
         dtype: float64
```

Figura 6: Retornos. Fonte: Os autores, 2022.

Após isso são criados 10 pesos aleatórios, utilizando o método *random.random* da biblioteca *Numpy que vai* gerar 10 números decimais. Executando o código *arr = np.random.random (10), o arr é utilizado para ler* o resultado que gera dez novos valores. Na próxima célula de código do *Jupyterlab* a ser executado, devemos combinar os dez valores da matriz para ver o valor da soma por meio do código *arr [0] + arr [1]*. Em vez disso o que se pode fazer é, na primeira linha, atribuir dez números aleatórios para a nova matriz *Numpy* chamada *weigths*.

Adicionamos uma linha de código que vem depois disso *weigths = np.random.random (num_assets)*, após adicionamos uma linha de código que vem depois chamada *weights /= np.sum (weights)* para ler essa informação com o comando *weights*. Sendo assim, na seção de *loops*, nós introduzimos o conceito de incrementar por meio da sintaxe "+=", para designar que queremos que a variável do lado esquerdo seja atribuída com a mesma variável mais o valor da expressão no lado direito. Essa operação aritmética não vai ser uma soma, mas uma operação de divisão com o nome de *weights* (uma variável igual a *weights* dividido pela soma dos pesos). Não podemos esquecer que *weigths* é uma matriz, isso significa que com a segunda linha de código *(weights /= np.sum(weights)*, que nós transformamos os valores da nossa matriz.

A seguir escrevemos a fórmula *np.sum (weights * log_returns.meam ( )) * 250*, para o retorno esperado de um portfólio. A fórmula é dada pela soma da média dos retornos logarítmicos anualizados ponderados pelos seus pesos. A biblioteca *Numpy* tem um método análogo que permite somar valores em objetos multidimensionais. A sintaxe correta é *Numpy ponto sum*, código que fornecerá a variância e a volatilidade esperadas do portfólio, respectivamente. As próximas células do *Jupyterlab* (são onde os códigos são executados) mostrarão o retorno e a volatilidade na simulação das carteiras de mínima a variância, *np .dot (weights.T, np.dot (log_returns.cov( ) * 250, weights))*. Esse é o trecho de código que fornecerá a variância e a volatilidade esperadas do portfólio). Passamos por esses códigos executados, porque precisamos das fórmulas para o retorno e a volatilidade na simulação da carteira de mínima variância.

O objetivo é o de criar um gráfico onde 1000 simulações de média variância serão plotadas, para mostrar os retornos hipotéticos da carteira versus a volatilidade. Não estamos considerando 1000 investimentos diversos compostos de diferentes ações. Portanto, estamos considerando 1000 combinações dos mesmos dez ativos, isso significa que simulamos 1000 combinações dos valores de seus pesos onde armazenaremos esses dados. Dessa forma, os retornos do portfólio começam como uma lista vazia e pretendemos preenchê-la com retornos esperados gerados aleatoriamente, fazendo a mesma coisa para as volatilidades do portfólio. Para isso foi preciso escrever um *loop* que será repetido mil vezes, utilizando a função *range*. No corpo do *loop* geramos aleatoriamente 10 pesos, onde a soma vai ser igual a 1. O método *append* adiciona na lista *portfólio returns* cada valor de retorno do portfólio gerado aleatoriamente. Essa operação será repetida para cada passagem do *loop* até que a nossa lista *portfólio returns* acumule 1000 observações.

Dissemos que nosso objetivo é o criar um gráfico que mostre os retornos hipotéticos da carteira versus a volatilidade. Portanto, precisamos de dois objetos onde armazenaremos esses dados. Assim os retornos do portfólio começam como uma lista vazia que logo terá os retornos esperados gerados aleatoriamente.

A mesma operação foi realizada para as volatilidades do portfólio. Como estamos calculando o retorno futuro do nosso portfólio, precisamos escrever um *loop* que será repetido 1000 vezes utilizando a função *range*. No corpo do *loop* foram gerados aleatoriamente dez pesos. Precisamos de 10 pesos porque como já mencionamos, o portfólio possui 10 ativos onde a soma vai ser igual a 1. Teremos que inserir a fórmula já mencionada anteriormente. Na célula seguinte, repetimos o mesmo procedimento para a volatilidade de carteira. Além disso, executamos a fórmula para o desvio padrão, que mostra como ficou o retorno e a volatilidade do nosso portfólio.

Temos, assim, duas listas longas contendo 1000 valores cada. Por meio da biblioteca *Numpy* é possível converter as listas em *arrays* (pois os dados ficam organizados e de fácil compreensão) e utilizando os mesmos nomes, como mostra a Figura 7.

```
Out[14]:  array([0.05737617, 0.12284075, 0.17812432, 0.06736948, 0.11647786,
                 0.11670916, 0.07444692, 0.11711769, 0.04288205, 0.1066556 ])
```

Figura 7: Exemplo de *Array*. Fonte: Os autores, 2022

Primeiro criamos um objeto *data frame* contendo duas colunas, uma para os retornos e outra para as respectivas volatilidades, que será chamado de *portfólios*. Seus dados são compostos pelas palavras chave *Return* e V*olatility*. Esses nomes foram atribuídos automaticamente como nomes das colunas, como mostra a Figura 8.

Out[21]:

	Returns	Volatility
0	0.092350	0.184716
1	0.099817	0.174064
2	0.074807	0.192772
3	0.073950	0.177198
4	0.073881	0.200217

Figura 8: Retornos Logarítmicos. Fonte: Os autores, 2022

Os dados contidos em *portfólio returns* e *portfólio volatilityes* constituem os dois pares de valores-chave e preenchem as duas colunas após uma verificação rápida com os métodos *head* (usado para obter o retorno de linhas superiores ou série de dados) e *tail* (usado para obter o retorno de linhas inferiores de um quadro ou série de dados).

Primeiro, vamos criar um objeto do tipo *data frame* contendo duas colunas, uma para os retornos e outra para as respectivas volatilidades. Vamos chamar esse objeto de *portfólios*, com os nomes de *portfólio returns* e *portfólio volatilities*. Os dados contidos nessas colunas vão constituir os dois pares de valores-chave em nosso estudo e vão preencher as duas colunas. Plotamos os dados em um gráfico com um eixo X correspondendo à coluna *volatility* e um eixo Y correspondente à coluna *return*. Uma especificação importante é com relação ao tipo de gráfico utilizado, o tipo *scatter* ou dispersão, ajustando seu tamanho, atribuindo rótulos para x e y ao gráfico, por exemplo *Expected Volatility* e *Expected Return.* Agora vem a sintaxe do *Matplotlib* que precisamos a partir do *data frame* "portfólios", utilizando o trecho de código a seguir:

portfolios.plot(x='Volatility', y='Return', kind='scatter', figsize=(10, 6)); plt.xlabel('Expected Volatility'); plt.ylabel('Expected Return').

Assim obtemos a fronteira eficiente (conceito apresentado por Harry Markowitz, em que o risco de uma carteira não é dado simplesmente pela média dos ativos individuais, mas pela diversificação da carteira de investimento como um todo).

Após o desenvolvimento destas etapas iniciais seguimos com a implementação e análise da próxima teoria, o critério de *Kelly.*

• Etapa 5 - Experimentação com Critério de *Kelly:* esse critério, além de maximizar o retorno, maximiza a utilidade dos investidores com funções de utilidades logarítmicas. Sob condições favoráveis, também maximiza a mediana dos retornos logarítmicos, algo útil para manter o investidor na estratégia. Para obter o Critério de *Kelly*, instalamos a biblioteca *cufflinks* (HILPISCH, 2018). Os dados utilizados foram extraídos de um arquivo CSV (*Comma-Separated Values*) chamado *tr_eikon_eod_data.csv* retirado do *site Quant Platform* (QUANTPLATFORM, 2022) que é uma tabela contendo dados de determinados ativos entre ações e índices. O ativo escolhido para a análise foi o SPX que é o código de S&P 500. A seguir foi realizada a normalização dos dados. A Figura 9 mostra um extrato da leitura dos dados do arquivo mencionado.

Out[30]:

Date	.SPX
2018-06-25	2717.07
2018-06-26	2723.06
2018-06-27	2699.63
2018-06-28	2716.31
2018-06-29	2718.37

Figura 9: Leitura dos dados do S&P 500. Fonte: Os autores, 2022

A variável *mu* recebe a média dos dados das últimas 252 semanas de negociação, m*u = data.returns.mean( ) * 252*. Depois foram lidos os dados com o comando *data.tail* (que é utilizado para obter as últimas *n* linhas, sendo útil para verificar rapidamente os dados) e também ler o *um.* É preciso gerar a variável *sigma* (utilizada para realizar uma série de adições)*,* ao utilizar a fórmula *f = (mu - r) / sigma ** 2*, será obtido o *f* que esta na fórmula da célula do *Jupyterlab.* A Figura 10 apresenta os dados obtidos por meio da execução destes comandos.

A variável *eqs* será incorporada na estratégia do Critério de *Kelly* que mostrará o retorno do capital investido. A seguir foi criado um *data frame* com os dados utilizados no estudo. Por último são mostrados os resultados por meio da plotagem em um gráfico apresentado na Figura 11.

Date	equity_2.23	equity_2.95	equity_4.47
2018-06-25	4.707070	6.367340	8.794342
2018-06-26	4.730248	6.408727	8.880952
2018-06-27	4.639340	6.246147	8.539593
2018-06-28	4.703365	6.359932	8.775296
2018-06-29	4.711332	6.374152	8.805026

Figura 10: Dados dos obtidos. Fonte: Os autores, 2022

Figura 11: Critério de *Kelly*

- Etapa 6 - Modelo *KMV-Merton*: entendido como uma extensão ao modelo proposto por *Black Scholes* e *Merton* (conforme Figura 12). Esse modelo se torna especialmente importante porque criou o conceito de *Distance to Default*, adicionando uma etapa ao modelo de *Merton* e elaborando uma definição de "ponto de *default*" mais realista (que não fosse somente o valor de face da dívida). Para obter o *KMV-Merton* é necessária instalação da biblioteca *Quantlib* (QUANTLIB.ORG, 2022)**.** Como já é de praxe, foram importadas as bibliotecas para serem utilizadas na análise. Para realizar o estudo utilizamos um *dataset* do *Walmart* (GITHUB, 2022)**.** Por meio do comando *dataset.info(5)*, limitamos a 5 linhas a leitura das informações do *dataset* para simplificar (vide Figura 13). Este arquivo contém algumas informações, tais como; preço de fechamento da ação, dívida e taxa livre de risco, como mostra a Figura 14.

```
In [35]: dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 2535 entries, 2011-02-01 to 2021-02-26
Data columns (total 8 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Close share price (USD)  2535 non-null   float64
 1   Book value/share (USD)   2535 non-null   float64
 2   Nº Shares (Million)      2535 non-null   int64
 3   ST Debt (Million)        2535 non-null   int64
 4   LT Debt (Million)        2535 non-null   int64
 5   30-day Implied Vol       2535 non-null   float64
 6   Risk-free rate (%)       2535 non-null   float64
 7   SP 500 close spot        2535 non-null   float64
dtypes: float64(5), int64(3)
memory usage: 178.2 KB
```

Figura 12: Dados de 2011 até 2021. Fonte: Os autores, 2022

Figura 13: Evolução do preço das ações do *Walmart*. Fonte: Os autores, 2022

Out[34]:

Date	Close share price (USD)	Book value/share (USD)	Nº Shares (Million)	ST Debt (Million)	LT Debt (Million)	30-day Implied Vol	Risk-free rate (%)	SP 500 close spot
2011-02-01	56.330002	20.26	3670	58603	43842	0.1453	3.443	1307.59
2011-02-02	55.860001	20.26	3670	58603	43842	0.1453	3.489	1304.03
2011-02-03	55.919998	20.26	3670	58603	43842	0.1453	3.541	1307.10
2011-02-04	56.029999	20.26	3670	58603	43842	0.1453	3.652	1310.87
2011-02-07	56.070000	20.26	3670	58603	43842	0.1453	3.644	1319.05

Figura 14: Dados da Tabela. Fonte: Os autores, 2022

O próximo comando executado foi o *dataset ['30-day Implied Vol']***.***plot (figsize=(15,6), title ='30-day Option Implied Firm Volatility')*, que permite plotar em um gráfico o resultado de 30 dias de volatilidade implícita da opção do *Walmart,* como mostra a Figura 15.

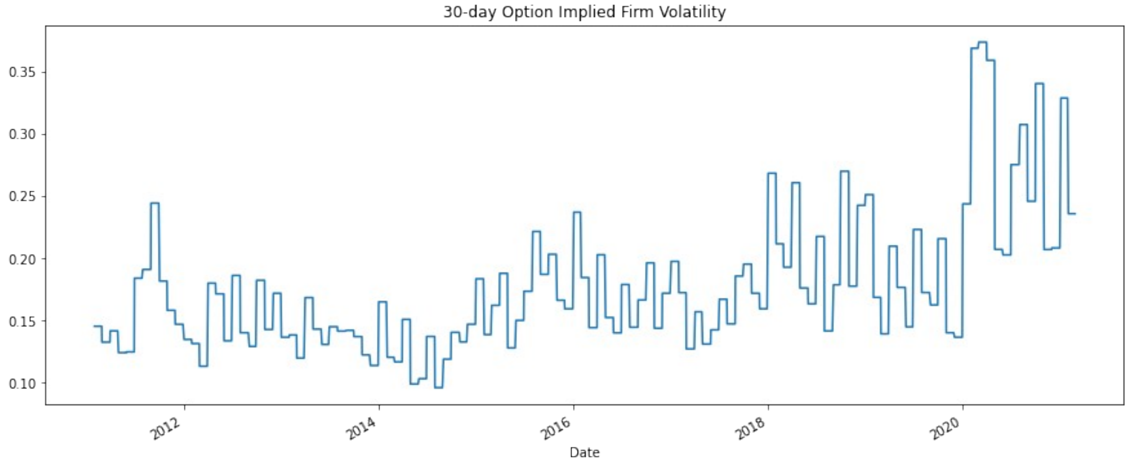

Figura 15: Volatilidade implícita da opção do *Walmart*. Fonte: Os autores, 2022

Na próxima célula de código do *Jupyterlab* foi plotada a evolução da dívida da empresa em milhões, que está no *dataset* como ST, LT *debit*, como mostra a Figura 16.

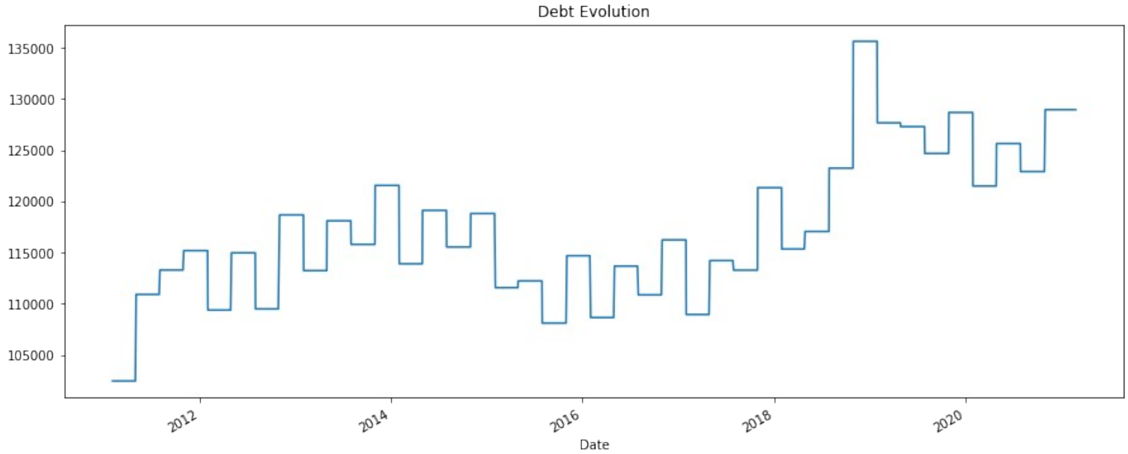

Figura 16: Evolução da dívida. Fonte: Os autores, 2022

Na célula seguinte do *Jupyterlab*, por meio dos comandos começados em plt (abreviação de *pyplot* da biblioteca *matplotlib*) como está exemplificado no código, que após plotado em um gráfico a comparação da dívida sobre o patrimônio líquido, pelo valor da dívida, mostrado na Figura 17.

plt.figure(figsize=(15, 6))

plt.title('Debt Ratios Evolution')

plt.plot(dataset['Debt/Equity (%)'], label='Debt to Equity')

plt.plot(dataset['Debt/Value (%)'], label='Debt to Value') *plt.legend( );*

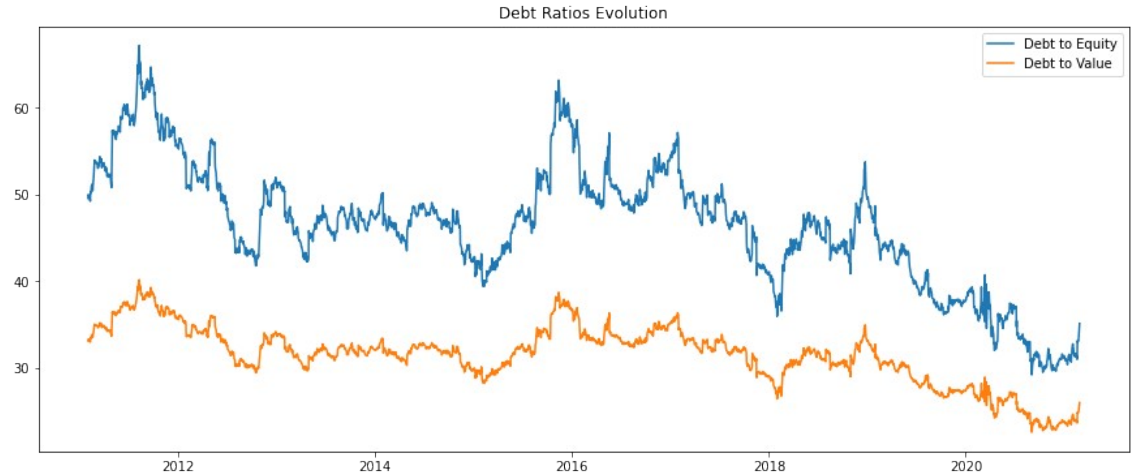

Figura 17: Evolução da comparação do endividamento. Fonte: Os autores, 2022

Foi plotada a evolução do patrimônio, da dívida e do valor da empresa, mostrada na Figura 18. No código o *plt* vai plotar o valor da companhia e o valor da ação. Notamos que o valor patrimonial e o valor dos ativos subiram ao longo do tempo e a dívida permaneceu de lado ao longo do mesmo período.

plt.figure(figsize=(15, 6))

plt.title('Equity, Debt and Company Value Evolution')

plt.plot(dataset['Company Value'], label='Assets Value')

plt.plot(dataset['Market Cap (Million)'], label='Equity Value')

plt.plot(debt_evolution, label='Debt Value')

plt.legend()

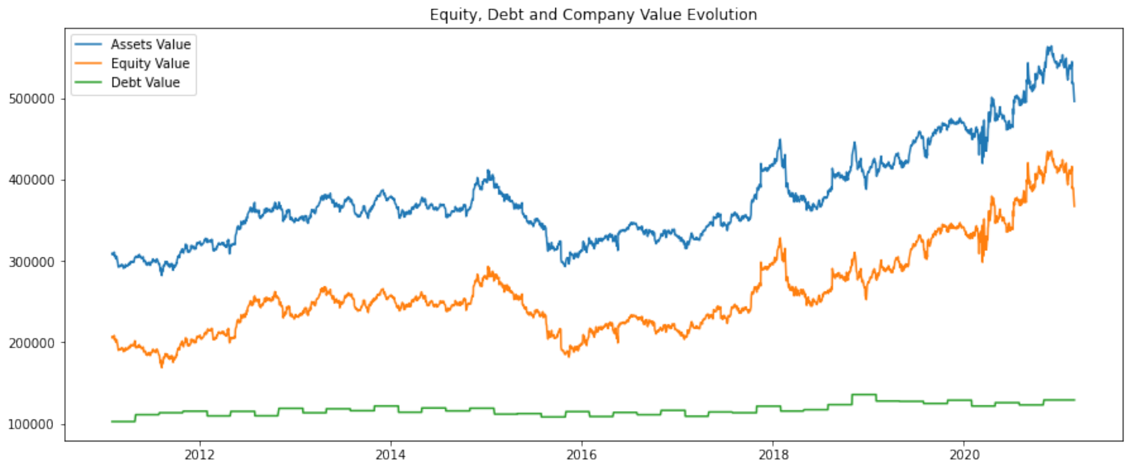

Figura 18: Evolução do Patrimônio, da Dívida e do Valor da Empresa. Fonte: Os autores, 2022

Nesta célula (a partir da utilização dos dados que estão no arquivo CSV) por meio dos comandos presentes nos códigos em *Python* é possível visualizar o preço de fechamento da ação no período, o valor da ação, o número de ações em milhões, a taxa livre de risco e o preço de fechamento do S&P 500 entre tantos outros dados, como mostra a tabela na Figura 19.

df = dataset

df.drop(columns =['Close share price (USD)', 'Book value/share (USD)','Nº Shares (Million)',

'Risk-free rate (%)','SP 500 close spot']).describe().transpose()

Out[47]:

	count	mean	std	min	25%	50%	75%	max
ST Debt (Million)	2535.0	72861.710454	7035.295888	58603.000	67942.0000	71818.0000	78521.0000	88121.0000
LT Debt (Million)	2535.0	44289.821696	3528.927534	36305.000	41836.0000	43842.0000	47079.0000	51568.0000
30-day Implied Vol	2535.0	0.176460	0.054706	0.096	0.1401	0.1662	0.1962	0.3735
Market Cap (Million)	2535.0	260494.368836	55269.576462	168709.000	222292.0000	248419.0000	283313.0000	435299.0000
Company Value	2535.0	377645.900986	60566.051843	282001.000	334270.0000	364135.0000	410319.5000	564269.0000
Equity book value	2535.0	79250.469034	4022.885091	68644.000	76546.0000	79664.0000	82414.0000	88063.0000
Price to book	2535.0	3.284521	0.658508	2.190	2.8200	3.1300	3.5600	5.1500
Debt/Equity (%)	2535.0	46.391065	7.074871	29.220	42.8800	46.9000	50.4050	67.1500
Debt/Value (%)	2535.0	31.528012	3.356602	22.620	30.0100	31.9200	33.5100	40.1700
Daily Return on Equity (%)	2534.0	0.032968	1.230023	-10.740	-0.5200	0.0400	0.6100	11.0700

Figura 19: Preço de fechamento, valor da ação, número de ações, taxa livre de risco e preço de fechamento do S&P 500. Fonte: Os autores, 2022

Na próxima célula do *Jupyterlab* o código executado *dataset ['Risk-free rate (%)'].plot(figsize=(15,6), title ='Risk-free rate evolution');* mostra a evolução da taxa livre de risco (Figura 20).

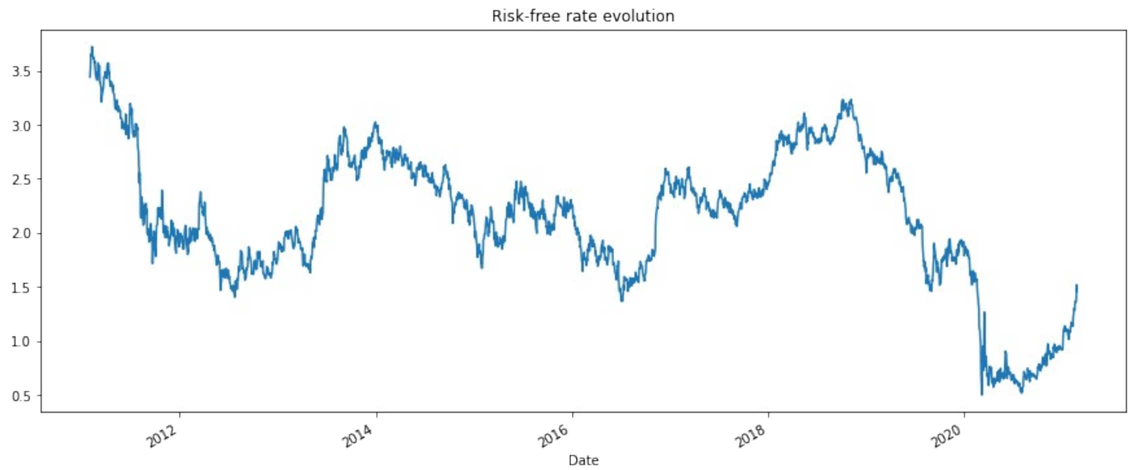

Figura 20: Taxa livre de Risco. Fonte: Os autores, 2022

Na próxima célula do *Jupyterlab,* o comando *print* mostra a rentabilidade média anual da empresa. Utilizando a média das últimas 252 semanas, o resultado é 8,31 % ao ano (Figura 21).

```
Out[50]: count    2535.000000
         mean        2.139153
         std         0.639445
         min         0.499000
         25%         1.774000
         50%         2.182000
         75%         2.592000
         max         3.725000
         Name: Risk-free rate (%), dtype: float64
```

Figura 21: Dados da Taxa livre de Risco. Fonte: Os autores, 2022

Avançando um pouco mais nas células do *Jupyterlab*, os próximos comandos vão calcular e plotar no gráfico o último dia útil de cada mês, como exemplificado na Figura 22.

```
# Compute the last working day of each month
d1 = ql.Date('2011-02-01','%Y-%m-%d')
d2 = ql.Date('2021-02-26','%Y-%m-%d')
date = ql.UnitedStates(ql.UnitedStates().NYSE).businessDayList(d1,d2)
dateEndMonth = [ ]
for i in date:
    dateEndMonth.append((ql.UnitedStates(ql.UnitedStates( ).NYSE).endOfMonth(i)).ISO ( ))
dateEndMonth = list(dict.fromkeys(dateEndMonth))
dateEndMonth
```

```
Out[53]: ['2011-02-28',
          '2011-03-31',
          '2011-04-29',
          '2011-05-31',
          '2011-06-30',
          '2011-07-29',
          '2011-08-31',
          '2011-09-30',
          '2011-10-31',
          '2011-11-30',
          '2011-12-30',
          '2012-01-31',
          '2012-02-29',
          '2012-03-30',
          '2012-04-30',
          '2012-05-31',
          '2012-06-29',
          '2012-07-31',
          '2012-08-31',
          '2012-09-28',
          '2012-10-31',
```

Figura 22: Último dia útil de cada mês. Fonte: Os autores, 2022

Agora, entrando no estudo do modelo KMV-Merton, abordamos a questão do *Distance to Default*, no período de 2011 a 2021, como mostra a Figura 23.

Date: 2011-02-28   Previous Year Date: 2010-02-28

```
Equity: 190767.0        Book Liabilities: 80524.0       Int. Rate:
0.034140000000000004         Tau: 1 Sigma of Equity:
0.047932255435261036
The company value E fitted:  268588.3081        Sigma E fitted:  0.034
d1: 36.40389484539158 DD: 36.36985057653445 Prob. Default:
6.380383700527487e-290
======================================================================
=========================================================
======================================================================
=========================================================
Date:  2011-03-31       Previous Year Date:  2010-03-31

Equity: 191023.0        Book Liabilities: 80524.0       Int. Rate:
0.03454      Tau: 1 Sigma of Equity: 0.05743061240251595
The company value E fitted:  268813.1858        Sigma E fitted:
0.0408
d1: 30.404313357211088      DD: 30.363502234644645         Prob.
Default: 8.334289955111116e-203
```

```
=======================================================================
=========================================================
=======================================================================
=========================================================
Date:  2011-04-29       Previous Year Date:  2010-04-29

Equity: 201777.0        Book Liabilities: 80524.0       Int. Rate:
0.032959999999999996         Tau: 1 Sigma of Equity:
0.06454560974771961
The company value E fitted:  279690.1914        Sigma E fitted:
0.0466
d1: 27.470566308562198         DD:  27.42400114504638  Prob. Default:
7.0959461483903e-166
```

Figura 23: *Distance to Default* Fonte: Os autores, 2022

Na próxima célula do *Jupyterlab* executamos os comandos para mostrar toda a distância até o padrão e probabilidade de inadimplência para o final de cada mês, conforme Figura 24.

Out[55]:

	End of Month	Distance to Default	Probability of Default
0	2011-02-28	36.369851	6.380384e-290
1	2011-03-31	30.363502	8.334290e-203
2	2011-04-29	27.424001	7.095946e-166
3	2011-05-31	24.632350	2.844645e-134
4	2011-06-30	21.286187	7.621835e-101
...	...	...	...
116	2020-10-30	6.347695	1.092824e-10
117	2020-11-30	6.360780	1.003661e-10
118	2020-12-31	6.232555	2.294443e-10
119	2021-01-29	6.138148	4.174452e-10
120	2021-02-26	5.863373	2.267785e-09

Figura 24: Distância até o padrão e probabilidade de inadimplência para o final de cada mês Fonte: Os autores, 2022

Na célula seguinte do *Jupyterlab* a linha de comando *dataset.head(1)* vai ler apenas uma linha, pois assim foi estipulado no código fonte, como mostram as Figuras 25 e 26.

Out[56]:

Date	Close share price (USD)	Book value/share (USD)	Nº Shares (Million)	ST Debt (Million)	LT Debt (Million)	30-day Implied Vol	Risk-free rate (%)	SP 500 close spot	Market Cap (Million)	Company Value	Equity book value	Price to book	Debt/
2011-02-01	56.330002	20.26	3670	58603	43842	0.1453	3.443	1307.59	206731.0	309176.0	74354.0	2.78	

Figura 25: Linha do *Dataset.* Fonte: Os autores, 2022

In [60]:

```
# Create a DataFrame to store all results
df_D = pd.DataFrame(columns=['End of Month','Distance to Default','Probability of Default','Drift Miu'])
df_D_extra = pd.DataFrame(columns=['Asset Value(iter k)','S&P 500','risk-free'])
df_E = pd.DataFrame(columns=['End of Month','Market Value of the firm','sigma_V','Liabilities(X)','Drift Miu'])

for idx in np.arange(0,len(dateEndMonth)):
    adjustedDate = ql.Date(dateEndMonth[idx],'%Y-%m-%d') - ql.Period('1Y')
    adjustedDate = adjustedDate.ISO()
    print("Date: ",dateEndMonth[idx],"\tPrevious Year Date: ",adjustedDate)

    tau = 1
    equity_iter = np.array(dataset['Market Cap (Million)'].loc[(dataset.index >= adjustedDate) & (dataset.index <= dateEndMonth[idx]
    debt_iter = np.array(dataset['ST Debt (Million)'].loc[(dataset.index >= adjustedDate) & (dataset.index <= dateEndMonth[idx])] +
                         (dataset['LT Debt (Million)'].loc[(dataset.index >= adjustedDate) & (dataset.index <= dateEndMonth[idx])] *
    r_iter = np.array(dataset.loc[((dataset.index >= adjustedDate) & (dataset.index <= dateEndMonth[idx])), 'Risk-free rate (%)'] *
    assetValue = equity_iter + debt_iter # iter k# iter k
    sigmaEquity_iter = sigmaValues(((dataset['Daily Return on Equity (%)'].loc[(dataset.index >= adjustedDate) & (dataset.index <= d

    iterResults = itertive_function(equity_iter, debt_iter, r_iter, assetValue)
    print('\nEquity:',equity_iter,'\nBook Liabilities:',debt_iter,'\nAsset Value:',assetValue,'\nInt. Rate:',r_iter,'\nTau:',tau,'\n
    print('\nResults', iterResults)
```

Figura 26: Criação do *Data Frame.* Fonte: Os autores, 2022

O código vai lendo o DF para cada período do *dataframe,* chamados no texto de *Question*, começando em A e terminado em F e calculando o DD e o PD (Figura 27), computando os dados, lendo e mostrando as informações, um exemplo é o df_D, apresentado na Figura 28.

```
# DD and PD computation
V_t = iterResults[0][-1]
sigma_V = iterResults[1]
X = debt_iter[-1]
tau = 1
drift_R = r_iter[-1]
print("\nMarket Value of the firm: ",V_t,"\nsigma_V: ",sigma_V,"\nLiabilities(X): ",X,"\nT: ",tau,"\ndrift r: ", drift_R)

resultDD_PD = DD_PD(V_t,sigma_V,X,tau,drift_R)
d1 , d2, probDef = resultDD_PD
print("\nDistance to Default = {}\nProbability of Default (%): {}".format(d2,probDef*100))
```

Figura 27: DD *and* PD *Computation*. Fonte: Os autores, 2022

Out[61]:

	End of Month	Distance to Default	Probability of Default	Drift Miu
0	2011-02-28	36.049015	7.145019e-285	0.051505
1	2011-03-31	30.266608	1.577372e-201	0.037159
2	2011-04-29	27.702237	3.281015e-169	0.035110
3	2011-05-31	23.773817	3.116373e-125	0.033811
4	2011-06-30	20.593919	1.555765e-94	0.037291
...	...	...	...	...
116	2020-10-30	6.521840	3.472507e-11	0.029421
117	2020-11-30	6.555712	2.768858e-11	0.029419
118	2020-12-31	6.345008	1.112071e-10	0.030300
119	2021-01-29	6.212167	2.612944e-10	0.032223
120	2021-02-26	5.837145	2.655140e-09	0.036006

Figura 28: DD e PD. Fonte: Os autores, 2022

O comando *df_E.iloc[0]* (Figura 29), mostra algumas informações importantes, tais como o valor de mercado da empresa, o valor do seu passivo entre outras, como mostra a Figura 30.

```
In [64]: for i in np.arange(0,len(df_E)):
             tau = 1
             resultDD_PD = DD_PD(df_E['Market Value of the firm'].iloc[i], df_E['sigma_V'].iloc[i],df_E['Liabilities(X)'].iloc[i],tau,df_E['Drift Miu'])
             d1 , d2, probDef = resultDD_PD
             df_E['Distance to Default'] = d2
             df_E['Probability of Default'] = probDef
         df_E
```

Figura 29: Informações da Empresa: Os autores, 2022

```
End of Month                2011-02-28
Market Value of the firm      268588.0
sigma_V                       0.034347
Liabilities(X)                 80524.0
Drift Miu                     0.051505
```

Figura 30: Comando *df_iloc[0].* Fonte: Os autores, 2022

Ao executar a célula do *Jupyterlab* com os comandos abaixo, será gerada uma tabela com as informações contidas no código, como mostra a Figura 31.

for i in np.arange(0,len(df_E)):

tau = 1

resultDD_PD = DD_PD(df_E['Market Value of the firm'].iloc[i], df_E['sigma_V'].iloc[i],df_E['Liabilities(X)'].iloc[i],tau,df_E['Drift Miu'])

d1 , d2, probDef = resultDD_PD

df_E['Distance to Default'] = d2

df_E['Probability of Default'] = probDef

df_E

Out[64]:

	End of Month	Market Value of the firm	sigma_V	Liabilities(X)	Drift Miu	Distance to Default	Probability of Default
0	2011-02-28	268588.0	0.034347	80524.0	0.051505	5.984923	1.082461e-09
1	2011-03-31	268813.0	0.040942	80524.0	0.037159	5.927476	1.538127e-09
2	2011-04-29	279690.0	0.046098	80524.0	0.035110	5.919274	1.616825e-09
3	2011-05-31	277947.0	0.050300	86559.5	0.033811	5.914071	1.668765e-09
4	2011-06-30	270550.0	0.056793	86559.5	0.037291	5.928007	1.533164e-09
...	...	...	...	...	...	...	...
116	2020-10-30	496719.0	0.239015	102436.5	0.029421	5.896493	1.856546e-09
117	2020-11-30	542932.0	0.242366	108545.5	0.029419	5.896483	1.856654e-09
118	2020-12-31	518238.0	0.243163	108545.5	0.030300	5.900012	1.817373e-09
119	2021-01-29	507622.0	0.245233	108545.5	0.032223	5.907713	1.734445e-09
120	2021-02-26	474126.0	0.249732	108545.5	0.036006	5.922863	1.581920e-09

Figura 31: Tabela. Fonte: Os autores, 2022

Na próxima célula do *Jupyterlab* o código *df_E.to_csv ("DF_Question_E.csv")*, cria um *Data frame* para fazer o mesmo estudo de DD e PD e mostrar as informações (Figura 32).

```
# Main function for question 2. f)
df_F = pd.DataFrame(columns=['End of Month','Distance to Default','Probability of Default'])

for idx in np.arange(0,len(dateEndMonth)):
    adjustedDate = ql.Date(dateEndMonth[idx],'%Y-%m-%d') - ql.Period('1Y')
    adjustedDate = adjustedDate.ISO()
    print("Date: ",dateEndMonth[idx],"\tPrevious Year Date: ",adjustedDate)

    equity_E = dataset['Market Cap (Million)'].loc[dateEndMonth[idx]]
    debt_X = dataset['ST Debt (Million)'].loc[dateEndMonth[idx]] + (dataset['LT Debt (Million)'].loc[dateEndMonth[idx]] * 0.5)
    r = dataset.loc[dateEndMonth[idx], 'Risk-free rate (%)'] * 0.01
    tau = 1
    sigmaEquity = sigmaValues(((dataset['Daily Return on Equity (%)'].loc[((dataset.index >= adjustedDate) & (dataset.index <= dateEndMonth[idx]))])*0.01
    sigmaNaive = sigmaNaiveFunction(equity_E, debt_X, sigmaEquity)

    print('\nEquity:',equity_E,'\tBook Liabilities:',debt_X,'\tInt. Rate:',r,'\tTau:',tau,'\tSigma of Equity:',sigmaEquity, '\tNaive Sigma:',sigmaNaive)

    # F-Solve function
    x_b = primarySystemEquation(equity_E, debt_X, r, tau, sigmaNaive)
    print('The company value E fitted: ', x_b[0].round(4), '\tSigma E fitted: ', x_b[1].round(4))
```

Figura 32: Plotagem dos Estudos. Fonte: Os autores, 2022

Após é criada uma tabela com essas informações o código vai plotar esses estudos conforme ilustrado na Figura 33.

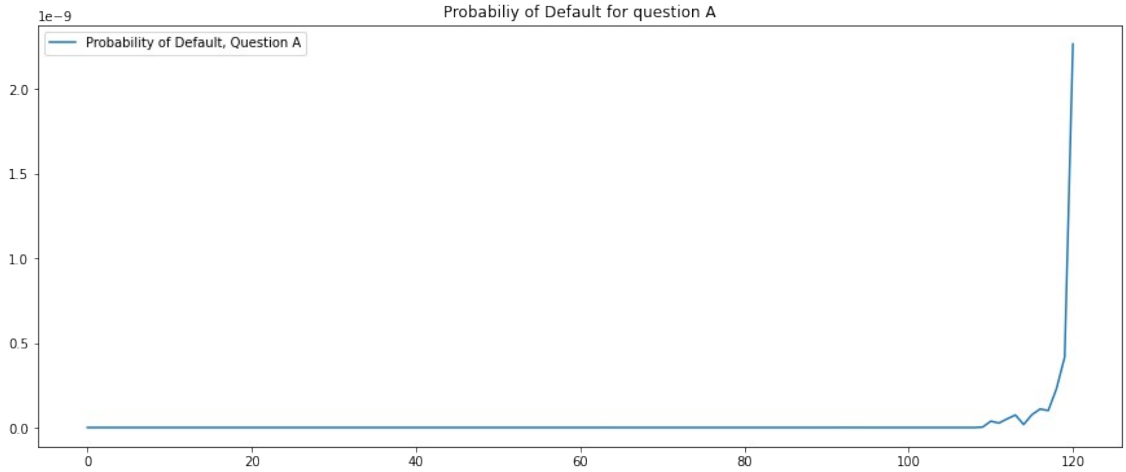

Figura 33: Gráfico da *Probability of Default* Questão A (*Data Frame Question A*). Fonte: Os autores, 2022

Nesta célula, que será executada no *Jupyter lab,* utilizamos os seguintes comandos (cujos resultados são apresentados na Figura 34):

plt.figure(figsize=(15, 6))
plt.title('Probabiliy of Default for question C')
plt.plot(df_C['Probability of Default'], label='Probability of Default, Question C')
plt.legend( );

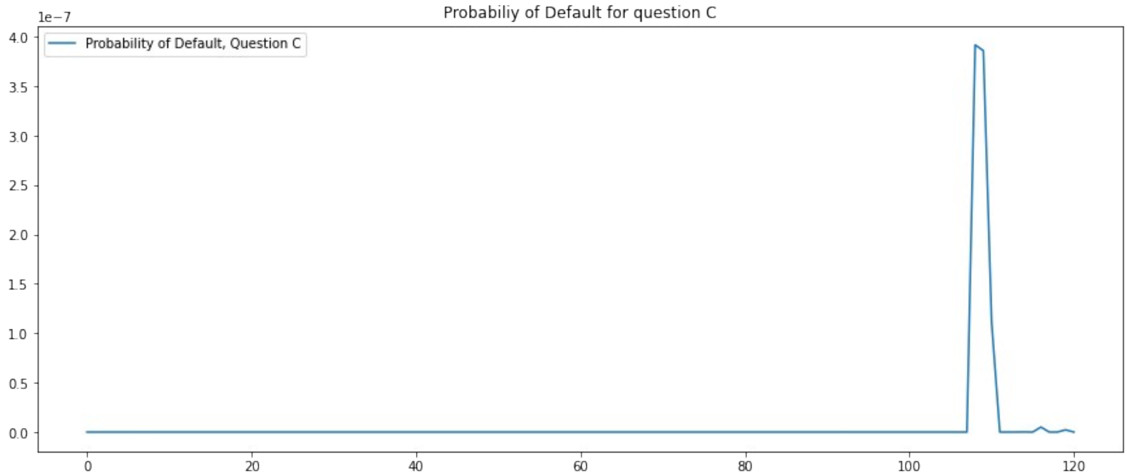

Figura 34: *Data Frame* questão C (Data frame no código fonte chamado *Question C*). Fonte: Os autores, 2022

Posteriormente executamos, no *Jupyter lab*, os comandos a seguir (cujos resultados são apresentados na Figura 35):

plt.figure(figsize=(15, 6))
plt.title('Probabiliy of Default for question D')
plt.plot(df_D['Probability of Default'], label='Probability of Default, Question D')

plt.legend( );

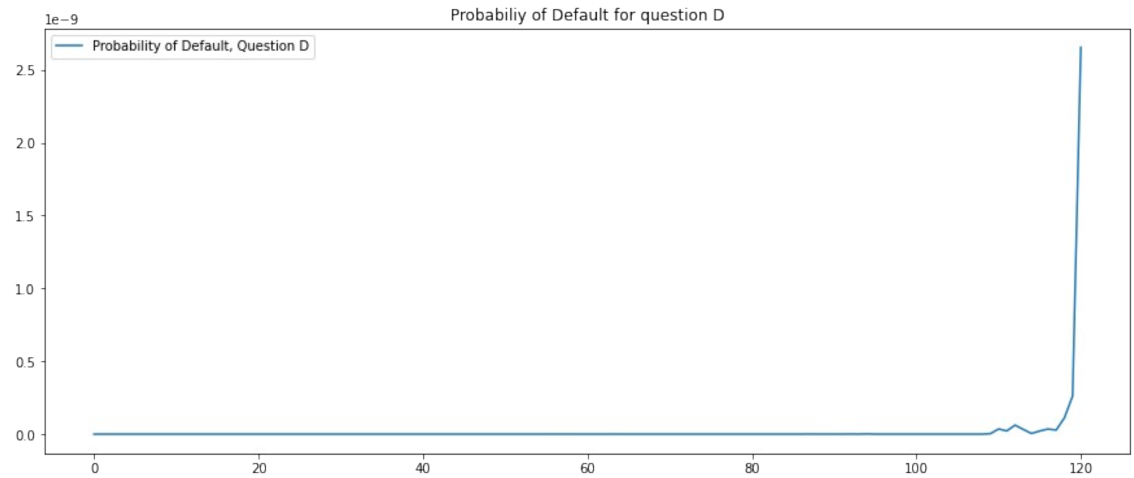

Figura 35: *Data Frame* questão D (*Question* D no código fonte). Fonte: Os autores, 2022

A seguir, no *Jupyter lab*, executamos o código que segue (resultados apresentados na Figura 36):

plt.figure(figsize=(15, 6))
plt.title('Probabiliy of Default for question E')
plt.plot(df_E['Probability of Default'], label='Probability of Default, Question E')
plt.legend( );

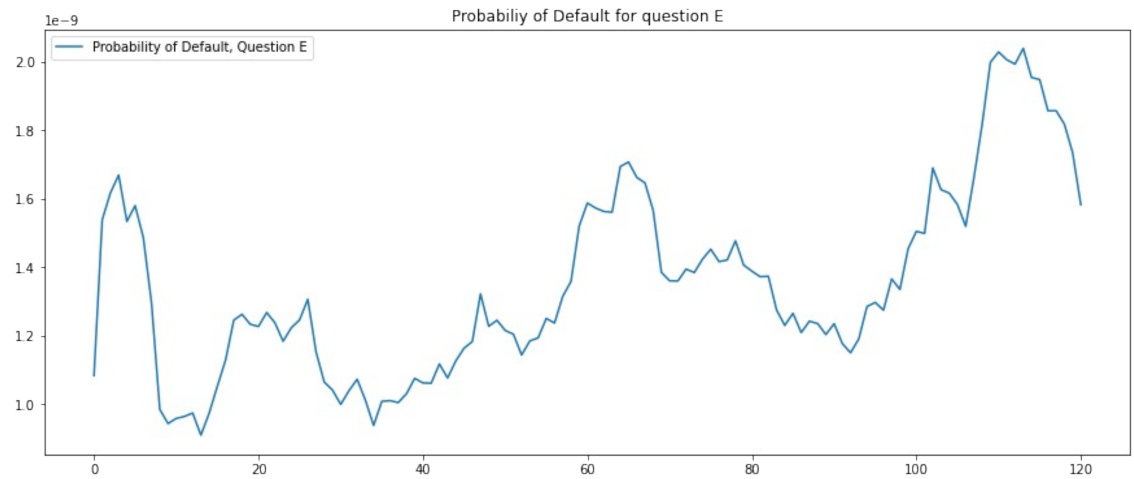

Figura 36: *Data Frame* questão E (*Question* E). Fonte: Os autores, 2022

Ainda seguindo na execução dos comandos para mostrar a *Probability of Default* temos o código seguinte, cujos resultados são apresentados na Figura 37:

plt.figure(figsize=(15, 6))

```
plt.title('Probabiliy of Default for question F')
plt.plot(df_F['Probability of Default'], label='Probability of Default, Question F')
plt.legend( );
```

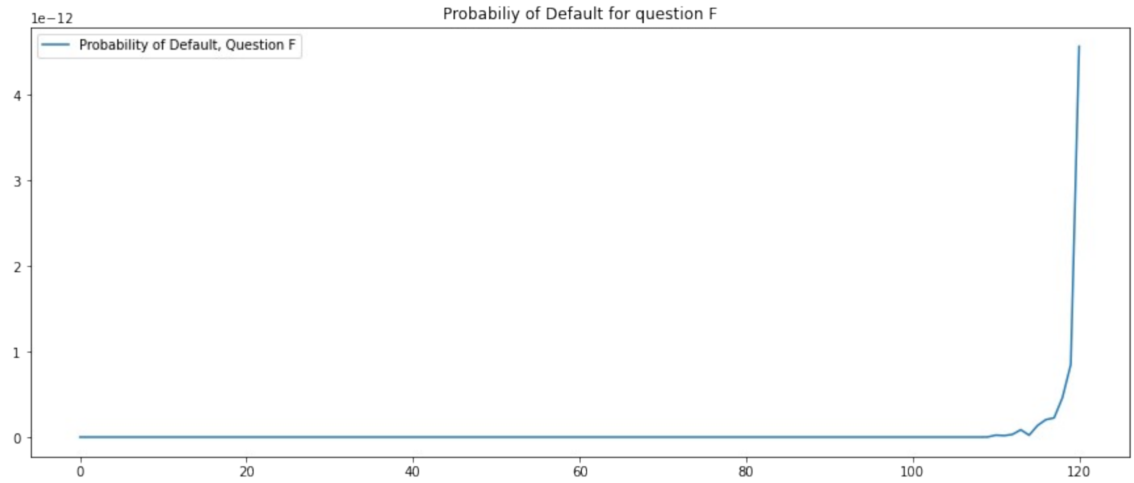

Figura 37: *Probability of Defaul*t da questão F (*Question* F). Fonte: Os autores, 2022

Etapa 7 - Análise dos Resultados: no começo foram utilizadas as primeiras células do *Jupyterlab* para carregar os dados. Uma vez que os dados estejam carregados e prontos para análise, o resultado mostra que os dados estão limpos, completos e organizados (após remover qualquer dado modificado que contenha inconstâncias, esteja incorreto, incompleto, duplicado, formatado incorretamente ou irrelevante). A ideia principal por trás dessa operação é comparar a valorização que os ativos tiveram ao longo do tempo analisado, no período de 2015 a 2022. Ao criar a carteira foram propostos pesos iguais para todos os ativos. Após feito isso foi calculado o peso de cada ação pelo seu retorno. Após estimar o produto das ações para comparar o desempenho em relação a outra carteira é necessário atribuir novos pesos para cada ação para obter retornos para efeito de comparação e estudo, no gráfico todas as linhas iniciam no eixo vertical, como mostra a Figura 38.

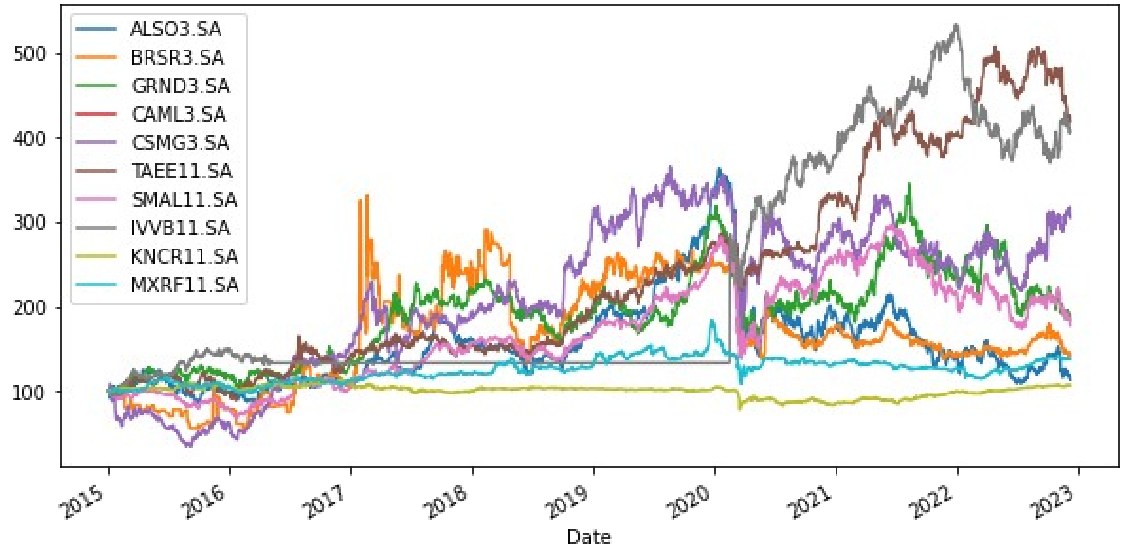

Figura 38: Desempenho das ações. Fonte: Os autores, 2022

Para obter as cotações e informações dos ativos utilizou-se a biblioteca *Yahoo Finance a yfinance*, junto com o *Pandas*. Para plotar os gráficos foi utilizado o *matplotlib*.

Um dos autores do trabalho possui conhecimento de mercado financeiro para selecionar os ativos para montagem do portfólio de investimentos e aplicou métodos utilizados em teorias de estudos econômicos, diversificação em diferentes classes de ativos, geografias diferentes e ativos denominados em moeda forte como o Dólar por exemplo, muito utilizado para proteção em tempos incertos no mercado financeiro.

5 Considerações Finais

Acreditamos que os objetivos propostos foram alcançados, já que foi realizada a implementação das teorias propostas e estudadas (Fronteira Eficiente de Harry Markowitz, Critério de Kelly de John Larry Kelly e KMV-Merton de Robert Coxx Merton) em *Python*, mostrando a teoria aliada à prática.

Entre as dificuldades destacamos a de encontrar o Critério de Kelly (QUANT PLATFORM, 2022), pois apenas fazendo muitas pesquisas para encontrar, principalmente o KMV-Merton pois são temas ainda desconhecidos pela maioria do meio Acadêmico.

Ao desenvolver a carteira digital proposta destacamos o nível avançado de conhecimento em Análise de dados e *Data Science* desenvolvidos, além da capacidade que a linguagem *Python* possui para auxiliar no desenvolvimento destas áreas.

Como trabalhos futuros seria interessante conseguir implementar um *dashboard,* com a biblioteca *Plotly Dash* (PLOTY.COM, 2022) que será interativo e que execute essas teorias e análises de otimização de portfólios com poucos comandos para executar o código, pois seria mais viável para os investidores realizarem seus estudos e análises.

Referências

B3.COM. **ETFs Listados**. Disponível em: https://www.b3.com.br/pt_br/produtos-e-servicos/negociacao/renda-variavel/etf/renda-variavel/etfs-listados/. Acesso em: 10 dez. 2022.

BLACK, F.; SCHOLES, M. The pricing of options and corporate liabilities. **The Journal of Political Economy,** Vol. 81, Issue 3, pp. 637-654, 1973.

BLACKROCK.COM.BR. **iShares BM&FBOVESPA Small Cap Fundo de Índice.** Disponível em: https://www.blackrock.com/br/products/251752/ishares-bmfbovespa-small-cap-fundo-de-ndice-fund. Acesso em: 17 dez. 2022.

COLMAN, DE PIETRO, WIENANDTS). Max **Montagem de uma carteira de investimentos utilizando a teoria de Markowitz**. 2013. Instituto de Ensino e Pesquisa (INSPER) – Sao Paulo, SP – Brazil.

CSHG Logística Fundo de Investimento CREDIT SUISSE Heding Griffo Corretora de Valores S.A. **FII Hglg11** - Funds Explorer. Disponível em: https://www.fundsexplorer.com.br. Acesso em 23 maio. 2022.

ELTON, E. et al. **Modern Portfolio Theory and Investment Analysis.** 9th. ed. [S.l.: s.n.], 2014.

EDWARD, T. O. **Understanding the Kelly Criterion**. Wilmatt Magazine, 8, 2008.

FAMA, E. F.; FRENCH, K.R., Common risk factors in the returns on stocks and bonds. Journal of Financial Economics, v. 33, n. 1, p. 3-56, 1993.

FIGUEIREDO, Rodrigo. **Risco de crédito e o retorno Acionário: Análise Empírica da Correlação para o Mercado Brasileiro**. 2018. 43f. Dissertação de Mestrado Dissertação apresentada no Programa de Pós-Graduação em Economia como parte dos requisitos necessários à obtenção de grau de Mestre em Finanças e Economia. FGV.

FUNDSEXPLORER.COM.BR. **Fundo Investimento Imobiliário FII Xpml11, XP MALLS - Funds Explorer.** Disponível em: https://www.fundsexplorer.com.br. Acesso em: 23 mai. 2022.

FONSECA, M. **Credit_Risk_Walmart.** Disponível em: https://github.com/MacarocoFonseca/Credit_Risk_Walmart. Acesso em: 08 dez. 2022.

HILPISCH, Y. Python for Finance. Sebastopol, California O'Reilly, 2018.

INFOMONEY. **Carteira de investimentos para iniciantes**: tudo o que você precisa saber para montar a sua. São Paulo, 10 out. 2019. Disponível em: Carteira de investimentos para iniciantes; como montar | InfoMoney. Acesso em: 30 mai. 2022.

JUPYTER.ORG. **Jupyter.** Disponível em: https://jupyter.org/. Acesso em: 26 dez. 2022.

KELLY, J. New Interpretation of Information Rate. Bell Siystem Technical Jornal. 1956.

KINEA. **Renda Imobiliária Fundo Investimento Imobiliário FII Knri11 - Funds Explorer.** Disponível em: https://www.fundsexplorer.com.br. Acesso em 23 maio. 2022.

LOPES, A. **Modelo de Opções Financeiras e Risco de Insolvência Aplicado ao Mercado Portugues**. 2013. ISCTE Business School, Departamento de Finanças – Lisboa, PT – Portugal.

MANGRAM, M. **A simplified perspective of the markowitz portfolio theory**. Global Journal of Business Research, 2013.

MARKOWITZ, H. Portfolio Selection. **Journal of Finance,** 7(1), 77-91, 1952.

MEIRELLES, L. **Aplicação de estratégias quantitativas com ativos do mercado financeiro Brasileiro**. **2021.** Projeto Final apresentado ao curso Engenharia de Produção da Universidade Federal Fluminense, como requisito parcial para aquisição do Grau de Bacharel em Engenharia de Produção – UFF. Escola de engenharia Departamento de Engenharia de Produção, Graduação em Engenharia de produção.

MERTON, R.C. On the pricing of corporate debt: The risk structure of interest rates. **The Journal of Finance**, Vol. 29, No. 2, pp. 449-470, 1974.

NUMPY.ORG. **NumPy.** Disponível em: https://numpy.org/. Acesso em: 26 dez. 2022.

PANDAS.PYDATA.ORG. **Pandas.** Disponível em: https://pandas.pydata.org/. Acesso em: 26 dez. 2022.

PLOTLY.COM. **Dash Enterprise.** Disponível em: https://plotly.com/dash/. Acesso em: 28 dez. 2022.

PYPI.ORG. **Jupyter.** Disponível em: https://pypi.org/project/yfinance/. Acesso em: 23 dez. 2022.

POUNDSTONE, W. Fortune´s formula. 2005. Disponível em: Fortune's Formula : William Poundstone: Free Download, Borrow, and Streaming: Internet Archive.

RANGAJARAN, K. S. **The Merton/KMV Approach to Pricing Credit Risk**. Working Paper, Stern School of Business, New York University, 2001.

QUANTNET. **Eu sou incapaz de validar critério de Kelly com python.** Disponível em: https://quantnet.com/threads/i-am-unable-to-validate-kelly-criterion-with-python.37733/ **I am unable to Validate Kelly Criterion with Python | QuantNet Community.** Comunidade QuantNet. Acesso em 27 julho. 2 022.

QUANTPLATFORM**. The Python Quants** Disponivel em: https://base.pqp.io/base/fo/get_search_html?st=kelly. Acesso em 16 novembro 2022.

UDEMY 2022. **Python para Finanças**: Investimentos & Análise de Dados. Udemy.

VELLEDA, I. **Quais são as ações que mais pagam dividendos?** Disponível em: https://forbes.com.br/forbes-money/2022/06/quais-sao-as-acoes-que-mais-pagam-dividendos/. Acesso em: 13 dez. 2022.

WAINBERG, R. **Melhores *Small* Caps para 2023.** Disponível em: https://www.suno.com.br/artigos/melhores-small-caps-para-2022/. Acesso em: 07 dez. 2022.