Universidade Federal do ABC
Centro de Engenharia, Modelagem e Ciências Sociais Aplicadas
Trabalho de Graduação em Engenharia de Informação

Algoritmo de Aprendizado Não Supervisionado Aplicado ao Mercado Acionário Brasileiro: Estudo Prático do Desenvolvimento da Plataforma Web "Atomus Invest"

Gabriel de Oliveira Souza
Victor Gabriel Ferreira dos Santos

Santo André

Agosto de 2024

GABRIEL DE OLIVEIRA SOUZA
VICTOR GABRIEL FERREIRA DOS SANTOS

ALGORITMO DE APRENDIZADO NÃO SUPERVISIONADO APLICADO AO MERCADO ACIONÁRIO BRASILEIRO: ESTUDO PRÁTICO DO DESENVOLVIMENTO DA PLATAFORMA WEB "ATOMUS INVEST"

Trabalho de Graduação apresentado ao curso de graduação em Engenharia de Informação da Universidade Federal do ABC, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Informação.

Orientador: Ricardo Suyama

Universidade Federal do ABC

Santo André

Agosto de 2024

GABRIEL DE OLIVEIRA SOUZA
VICTOR GABRIEL FERREIRA DOS SANTOS

ALGORITMO DE APRENDIZADO NÃO SUPERVISIONADO APLICADO AO MERCADO ACIONÁRIO BRASILEIRO: ESTUDO PRÁTICO DO DESENVOLVIMENTO DA PLATAFORMA WEB "ATOMUS INVEST"

Trabalho de Graduação apresentado ao curso de graduação em Engenharia de Informação da Universidade Federal do ABC, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Informação.

Santo André, 13 de setembro de 2024

Ricardo Suyama
Orientador

Luneque Del Rio de Souza e Silva Junior
Convidado 1

Kenji Nose Filho
Convidado 2

Santo André
Agosto de 2024

"A curto prazo o mercado é uma máquina de votação,
mas, a longo prazo, o mercado é uma balança."
(Benjamin Graham)

Resumo

Este trabalho tem como objetivo principal explorar o desenvolvimento de uma plataforma de análise de investimentos que visa auxiliar os investidores no processo de tomada de decisão, através da união das técnicas de modelagem de dados e estatística aos modelos convencionais de análise fundamentalista, aplicadas no contexto do mercado acionário brasileiro. Por meio de estatística e modelagem de dados, este trabalho visa simplificar a análise de investimentos em renda variável, promovendo a democratização de informações relativas aos indicadores fundamentalistas das ações brasileiras, com aplicação de métodos estatísticos para análise automatizada dos balanços das empresas, seguindo a filosofia de investimento de grandes investidores, por meio de algoritmos de aprendizado não supervisionado (clusterização) e valoração de empresas (*valuation*). Tendo em vista o contexto de crescimento e democratização do mercado acionário brasileiro, foi identificada a oportunidade de integrar tecnologia aos métodos de análise fundamentalista tradicionais, para criação de um website (Atomus Invest) que auxilie o investidor de longo prazo na construção de estratégias de investimento sólidas. Para tal, este trabalho se propõe a obter grupos de ações com características fundamentalistas similares, através de diferentes modelos para clusterização de dados, combinados com métodos de redução de dimensionalidade, avaliando a performance técnica destes métodos, sendo posteriormente submetidos ao processo experimental de composição de portfólios, utilizando os clusters como instrumentos para diminuição dos vieses inerentes ao processo de seleção de ativos.

Palavras-chaves: Clusterização, IBOV, Valuation, Análise Fundamentalista.

Abstract

The main objective of this work is to explore the development of an investment analysis platform that aims to help investors in the decision-making process by combining data modeling and statistical techniques with conventional fundamental analysis models, applied in the context of the Brazilian stock market. Through statistics and data modeling, this work aims to simplify the analysis of investments in variable income, promoting the democratization of information related to the fundamental indicators of Brazilian stocks, with the application of statistical methods for automated analysis of company balance sheets, following the investment philosophy of great investors, through unsupervised learning algorithms (clustering) and valuation of companies (valuation). Considering the growth and democratization of the Brazilian stock market, an opportunity was identified to integrate technology with traditional fundamental analysis methods to create a website (Atomus Invest) that helps long-term investors to build solid investment strategies. To this end, this work proposes to obtain groups of stocks with similar fundamental characteristics, through different data clustering models, combined with dimensionality reduction methods, evaluating the technical performance of these methods and being subsequently submitted to the experimental process of portfolio composition, using clusters as instruments for reducing biases in the stock picking process.

Key-words: Clustering, IBOV, Valuation, Fundamental Analysis

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Nos últimos 5 anos, com o aumento expressivo do ingresso de pessoas físicas na bolsa de valores brasileira (B3), atingindo a marca de 3.2 milhões de investidores, no primeiro semestre de 2021 (1), foi observada a problemática de um número expressivo de investidores iniciantes e pouco qualificados que ao entrarem no universo dos investimentos, são impactados por uma enorme quantidade de informações, dentre as quais, cabe ao iniciante, aprender a interpretar e discernir entre os sinais e ruídos que são apresentados.

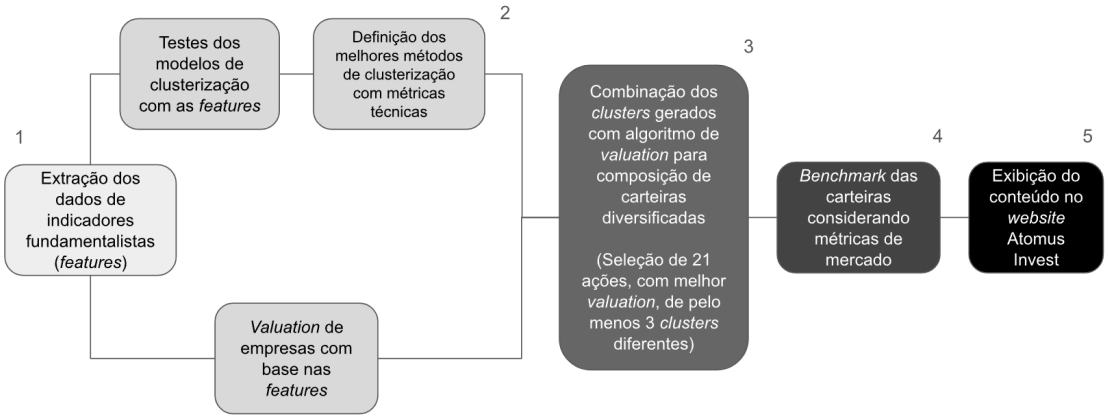
Dentro da história recente do mercado financeiro, se destacam grandes investidores, como Warren Buffett, Peter Lynch e Benjamin Graham, que muitas vezes podem parecer um pouco distantes da realidade dos pequenos investidores pessoa física, contudo, além da aura mística que envolve a história destes grandes Investidores, observamos que suas filosofias de investimento que se provaram no mercado, comumente, se baseiam no investimento de longo-prazo, balizado pela análise dos fundamentos contábeis e mercadológicos das empresas listadas no mercado acionário.

Dentro do contexto previamente introduzido, o presente estudo se propõe a promover a integração entre a tecnologia e as concepções filosóficas dos grandes investidores, objetivando o desenvolvimento final de uma plataforma de investimentos (Atomus Invest), com a proposta de entregar informações de alta qualidade ao investidor, possibilitando a diversificação dos investimentos, de forma acessível e democrática.

Através da difusão de estratégias historicamente performáticas de investimento (2), de forma integrada nos algoritmos da plataforma, espera-se fomentar um processo de tomada de decisão consciente e de alto nível, mesmo frente ao crescente número de investidores iniciantes presentes no mercado acionário brasileiro.

Desta forma, durante a implementação do projeto foram utilizados 4 diferentes algoritmos de clusterização de dados, com objetivo de segmentar as ações listadas na bolsa brasileira de acordo com *features* de análise fundamentalista, possibilitando encontrar grupos de ações que possuem características de *valuation* similares. Para tal, foram extraídos os dados dos múltiplos fundamentalistas referentes ao balanço das empresas, divulgados no segundo trimestre de 2024, que serviram como entrada para o teste dos diferentes modelos para clusterização de dados. Após a implementação dos modelos, todos foram submetidos à análise técnica para avaliação da qualidade e definição dos clusters resultantes, e posteriormente, ao processo experimental de composição de portfólios, utilizando os clusters como instrumentos para diminuição dos vieses e riscos, intrínsecos ao processo de avaliação financeira na composição de portfólios, seguindo as etapas descritas no fluxograma da Figura 1.

Figura 1 – Fluxograma de Desenvolvimento do Projeto



Fonte: Própria (2024)

2 Fundamentação Teórica

Com o objetivo de obter grupos de ações que possuem características fundamentalistas similares e experimentar o processo de composição de portfólio, unindo as técnicas de *valuation* com os grupos resultantes, foram utilizados 2 pilares teóricos que fundamentam todas as etapas do projeto, descritas na Figura 1. O primeiro pilar consiste no processo de clusterização, incluindo os 4 diferentes algoritmos explorados durante o estudo; já o segundo, consiste no processo de avaliação de empresas (*valuation*), descritos de forma mais detalhada nas Seções 2.1 e 2.3, respectivamente.

2.1 Clusterização (Aprendizado de Máquina Não-Supervisionado)

A Clusterização é um tipo de aprendizado de máquina não supervisionado, muito utilizado em análise estatística para classificar dados pertencentes a uma amostra em grupos que compartilham características comuns. Na Plataforma da Atomus, as ações são agrupadas com base em indicadores fundamentalistas previamente selecionados.

Os gráficos gerados a partir do algoritmo de clusterização K-Means fornecem ideias sobre a afinidade de uma empresa em relação a outra, baseada em seus indicadores fundamentalistas. Sendo assim, quanto maior a proximidade de um grupo de empresas, maior será a similaridade dos seus fundamentos contábeis.

O K-Means é uma ferramenta pertencente à família dos algoritmos de Aprendizado de Máquina Não-Supervisionado, esta que, por sua vez, é constituída por uma classe de operações que implicam a capacidade de uma interface de máquina de compreender, elaborar e premeditar um conjunto de regras e relações dentro de um conjunto de dados arbitrário, sem que exista a necessidade de conhecer uma resposta pré-computada por parte do algoritmo.

Em particular, a técnica utilizada pelo K-Means parte do princípio de que seja possível segregar partições de dados em torno de regiões geométricas centrais (o que denotamos por centróides), de modo que obtenhamos n observações particionadas em k grupos, de modo que cada observação pertença ao grupo mais próximo da média, permitindo, dessa forma, a divisão espacial do conjunto de dados em um formato de Diagrama de Voronoi (3).

Diante de uma perspectiva computacional, o K-Means possui uma complexidade da ordem np-difícil, o que implica a existência de um problema NP-completo (3). Dessa forma, é natural pensarmos que, para conjuntos de dados exponencialmente crescentes, devam existir algoritmos heurísticos tão ou mais eficientes que o K-Means, sendo estes co-

mumente empregados em tarefas que exigem conversão rápida para o que denotamos por "local optimum". Como uma aproximação, podemos pensar no algoritmo de maximização da expectativa para misturas de distribuições gaussianas, através de uma abordagem de refinamento iterativo e sob a mesma abordagem de obtenção de centróides como regiões de convergência, porém, é sempre válido ressaltar que a clusterização por K-Means tende a encontrar clusters de extensão espacial comparáveis, enquanto o mecanismo de maximização da expectativa permite a existência de comparações assimétricas.

Além disso, em termos práticos, para a constituição das regiões de convergência, o K-Means se apropria de elementos da geometria elementar, como a própria nomenclatura "centróide" implica. O termo é oriundo da geometria euclidiana, a conjuntura a qual este conceito espacial de clusterização está implicitamente baseado, vez que a norma do relacionamento entre as features do conjunto de dados (i.e, os elementos do conjunto) é inteiramente baseada nas distâncias euclidianas de pares entre estes pontos geométricos, dado que a soma dos desvios quadrados do centróide é igual à soma das distâncias euclidianas quadradas de pares divididas pelo número de pontos.

2.2 Comparação das Técnicas de Clusterização

A bibliografia de algoritmos para clusterização, envolve diferentes métodos para execução da clusterização, tais como K-Means, DBSCAN, *Affinity Propagation*, *Hierarchical Clustering*, Análise de Correspondência e FAMD. Dentro do contexto de testes e desenvolvimento da plataforma Atomus Invest, foram exploradas outras três técnicas de clusterização, além do K-Means que apresentaram diferentes comportamentos para análise dos indicadores fundamentalistas das empresas, sendo as técnicas de DBSCAN, *Hierarchical Clustering* e *Affinity Propagation*.

2.2.1 DBSCAN

O DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) é um algoritmo popular de clusterização baseado na diferença de densidade para uma representação dos dados em um espaço euclidiano, sendo comumente utilizado para tarefas de aprendizado de máquina não supervisionadas. Ele foi introduzido por Martin Ester, Hans-Peter Kriegel, Jörg Sander e Xiaowei Xu, em 1996 (4). O DBSCAN é particularmente útil para agrupar pontos de dados, em datasets com presença de ruído e clusters de formato irregular. O recurso mais interessante do DBSCAN é sua maior tolerância a *outliers*. Ele também não exige que o número de clusters seja informado de antemão, ao contrário do K-Means, em que é preciso especificar o número de centróides.

O DBSCAN é um algoritmo que define os clusters como regiões contínuas de alta densidade e seu funcionamento exige que todos os clusters sejam suficientemente densos

e bem separados por regiões de baixa densidade.

Outros métodos de clusterização, como K-Means, PAM e o *Hierarchical Clustering* funcionam para encontrar clusters esféricos ou convexos, sendo adequados para clusters compactos e podem ser mais afetados pela presença de ruídos e *outliers* nos dados. O K-Means e o *Hierarchical Clustering* falham na criação de clusters com formas arbitrárias (pouco definidos), não sendo capazes de formar clusters com base em densidades variáveis, diferentemente do DBSCAN.

Nesse algoritmo, temos 3 tipos de pontos de dados: - *Core Point*, descrito como um ponto central, caso exista mais de *MinPts* (quantidade mínima de pontos) dentro do raio epsilon. - *Border Point*: Um ponto que tem menos do que *MinPts* dentro do raio epsilon, mas está na vizinhança de um ponto central. - Ruído ou *outlier*: Um ponto que não é um ponto central ou um ponto de borda.

Para avaliar a performance deste algoritmo de clusterização, podemos usar métricas como o *Silhouette Score* e o *Rand Score*. A pontuação de *Silhouette* pode assumir valores no intervalo de -1 a 1. Uma pontuação próxima de 1 denota o melhor, o que significa que o ponto de dados i é muito compacto dentro do cluster ao qual pertence e distante dos outros *clusters*. O pior valor é -1. Valores próximos a 0 indicam clusters sobrepostos. O *Rand Score*, pode assumir valores no intervalo de 0 a 1. Mais de 0,9 denota excelente definição dos *clusters*, e acima de 0,8 denota uma boa definição, já um score inferior à 0,5 indica *clusters* pouco definidos.

O algoritmo de DBSCAN, possui certas vantagens e desvantagens, de acordo com os casos específicos de uso para aplicação do algoritmo, conforme descritas na Tabela 1(4).

Tabela 1 – Comparativo de Aplicabilidade do DBSCAN

Vantagens	Desvantagens
Pode separar clusters de alta densidade e de baixa densidade em um conjunto de dados.	Pode ser confuso quando há pontos de borda que pertencem a dois clusters.
Lida bem com outliers dentro do conjunto de dados.	Não funciona bem para clusters sem densidade variável.
Não requer um número pré-definido de clusters.	Sensível à escolha dos parâmetros (epsilon e minPoints).
Pode identificar clusters de formas arbitrárias.	Não é determinístico, ou seja, pode formar diferentes clusters em diferentes tentativas.

2.2.2 Affinity Propagation

O algoritmo de *Affinity Propagation* utiliza uma abordagem baseada em grafos, desta forma não requer especificação prévia do número de clusters, sendo baseado no conceito de propagação de afinidade, onde os pares dos pontos de dados são comparados e trocam mensagens para determinar seus pertencimentos aos clusters. O método consiste em duas etapas principais: cálculo da matriz de similaridade e propagação de afinidade.

1. O cálculo da matriz de similaridade é realizado entre todos os pares de pontos de dados. A similaridade entre dois pontos é medida usando uma função de similaridade, como a distância euclidiana ou a similaridade de cosseno. Esta matriz reflete o grau de afinidade entre todos os pontos de dados de um dado conjunto.
2. Na propagação de afinidade, os pontos de dados trocam mensagens entre si para determinar seus pertencimentos aos clusters. Cada ponto de dados envia mensagens de "responsabilidade para todos os outros pontos, indicando o quão adequado seria constituir um centro de cluster (centróide). Ao mesmo tempo, cada ponto recebe mensagens de "disponibilidade" de outros pontos, indicando quão adequado é para se juntar ao cluster de outro ponto como centro. Essas mensagens são atualizadas iterativamente até que os pertencimentos aos clusters convirjam.
    1. As mensagens de disponibilidade, denotadas por $a(i, k)$, representam a disponibilidade relativa de cada ponto k em servir como exemplar para o ponto i.
    2. As mensagens de responsabilidade, denotadas por $r(i, k)$, representam a responsabilidade relativa de cada ponto i de dados em escolher outro ponto de dados k como seu exemplar

Com isso, o número de clusters é determinado pelos próprios dados, com cada cluster representado por um ponto de dados específico (exemplar) e outros pontos atribuídos a ele com base em suas mensagens de disponibilidade e responsabilidade.

O algoritmo de *Affinity Propagation*, possui certas vantagens e desvantagens, de acordo com os casos específicos de uso para aplicação do algoritmo, conforme descritas na Tabela 2 (5).

Tabela 2 – Comparativo de Aplicabilidade do AP

Vantagens	Desvantagens
Autodeterminação do Número de Clusters	Sensibilidade aos Parâmetros de inicialização

Tabela 2 – Comparativo de Aplicabilidade do AP

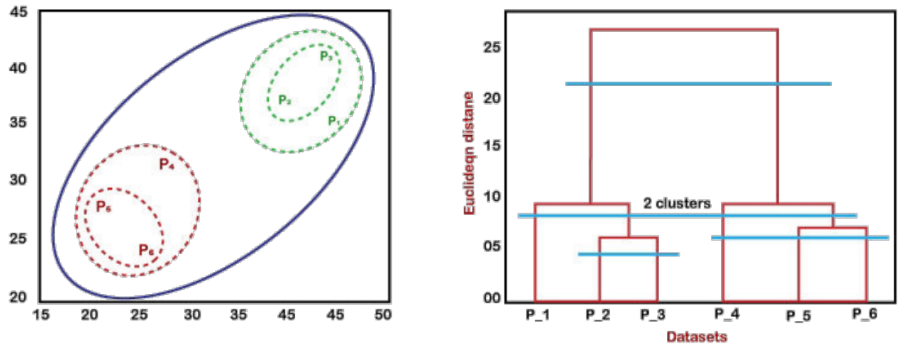
Robustez a Dados de Alta Dimensão	Computacionalmente exigente, especialmente para conjuntos de dados grandes.
Capacidade de Lidar com Diferentes Métricas de Similaridade	Alta Sensibilidade a *Outliers*

2.2.3 Hierarchical Clustering

O *Hierarchical Clustering* é um algoritmo que constrói uma estrutura de agrupamento em forma de uma árvore hierárquica, chamada dendrograma, a qual permite uma compreensão visual da relação entre os clusters, sendo um algoritmo útil para análises exploratórias. O método possui pontos positivos tanto visuais, como citado acima, quanto na flexibilidade e robustez, dado que não requer a especificação prévia do número de clusters e é um algoritmo menos sensível a *outliers* em comparação com outros, como o K-Means, se tornando adequado para análises onde a presença de *outliers* é comum. Contudo o método, apesar de robusto, pode ser computacionalmente intensivo, o cálculo das distâncias entre todos os pares de pontos de dados e a construção do dendrograma podem se tornar um problema em conjuntos de dados muito grandes.

O dendrograma que é o destaque visual deste método é uma estrutura em forma de árvore que é utilizada para armazenar cada passo como uma memória que o algoritmo de agrupamento hierárquico efetua. No gráfico do dendrograma, o eixo Y apresenta as distâncias euclidianas entre os pontos de dados e o eixo x apresenta todos os pontos de dados do conjunto de dados em causa. O funcionamento pode ser exemplificado na Figura 2:

Figura 2 – Funcionamento do Dendograma e Agrupamento



Fonte: Própria (2024)

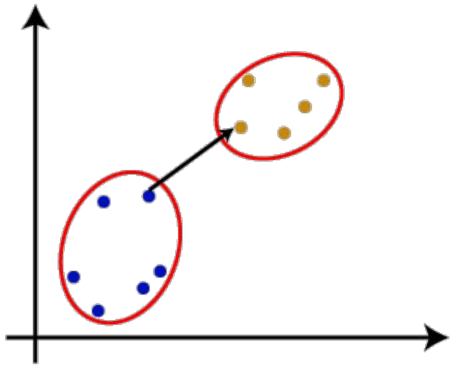
No diagrama acima, a figura da esquerda mostra como os agrupamentos são criados no agrupamento aglomerativo e a figura da direita mostra o dendrograma correspondente.

1. Em primeiro lugar, os pontos de dados P2 e P3 combinam-se e formam um cluster, sendo criado um dendrograma que liga P2 e P3 com uma forma retangular. A altura é decidida de acordo com a distância euclidiana entre os pontos de dados.
2. Na etapa seguinte, P5 e P6 formam um cluster, e o dendrograma correspondente é criado. É superior ao anterior, uma vez que a distância euclidiana entre P5 e P6 é um pouco superior à de P2 e P3.
3. Mais uma vez, são criados dois novos dendrogramas que combinam P1, P2 e P3 num dendrograma e P4, P5 e P6 noutro dendrograma.
4. Por fim, é criado o dendrograma final que combina todos os pontos de dados. Para escolher o número de clusters, selecione a linha vertical mais longa de modo a que nenhuma linha horizontal passe por ela.

Os agrupamentos realizados em P2, P3, . . . , P5, são decididos a partir dos cálculos da distância mais próxima entre os agrupamentos. Existem várias formas de calcular a distância entre dois clusters, e estas formas decidem a regra de agrupamento. Estas medidas são designadas por métodos de ligação. Alguns dos métodos de ligação mais populares são:

1. Ligação simples: É a distância mais curta entre os pontos mais próximos dos clusters, conforme exemplificado na Figura 3.

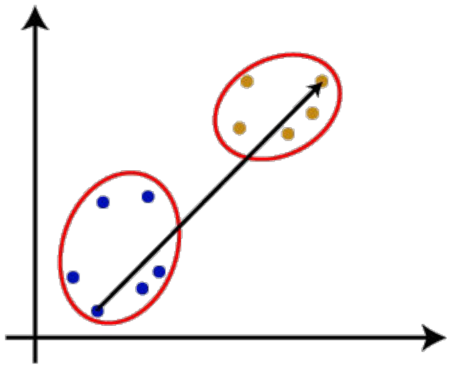
Figura 3 – Exemplo de Ligação Simples (HC)



Fonte: Própria (2024)

2. Ligação completa: É a maior distância entre os dois pontos de dois clusters diferentes. É um dos métodos de ligação mais populares, uma vez que forma clusters mais apertados do que a ligação simples, conforme exemplificado na Figura 4.

Figura 4 – Exemplo de Ligação Completa (HC)



Fonte: Própria (2024)

3. Ligação de centroides: É o método de ligação em que é calculada a distância entre os centroides dos clusters.

O algoritmo de *Hierarchical Clustering*, possui certas vantagens e desvantagens, de acordo com os casos específicos de uso para aplicação do algoritmo, conforme descritas na Tabela 3 (6).

Tabela 3 – Comparativo de Aplicabilidade do Hierarchical Clustering

Vantagens	Desvantagens
Capacidade de lidar com clusters não convexos e clusters de diferentes tamanhos e densidades.	Necessidade de um critério para parar o processo de agrupamento e determinar o número final de agrupamentos.
Capacidade de tratar dados em falta e dados ruidosos.	Custo computacional pode ser elevado, especialmente em grandes conjuntos de dados.
Capacidade de revelar a estrutura hierárquica dos dados, que pode ser útil para compreender as relações entre os clusters	Resultados podem ser sensíveis às condições iniciais, ao critério de ligação e à métrica de distância utilizada

2.2.4 K-Means

O K-Means é um algoritmo que segue uma abordagem iterativa para definir os melhores clusters, possuindo como duas de suas principais características, a simplicidade de implementação e a eficiência computacional.

Para definição dos centróides o método seleciona aleatoriamente k pontos de dados como centróides iniciais, onde k é o número especificado de clusters, cada ponto de dados

é atribuído ao cluster representado pelo centróide mais próximo, com base em alguma medida de distância, comumente a distância euclidiana. Os centróides de cada cluster são recalculados como a média de todos os pontos atribuídos a esse cluster, redefinindo a posição dos centróides para o centro dos seus respectivos clusters, buscando reduzir a distância euclidiana de cada cluster para cada ponto. Por fim, a operação é repetida iterativamente até que os centróides não mudam significativamente de uma iteração para outra ou até que um critério de convergência seja alcançado.

O algoritmo K-Means, possui certas vantagens e desvantagens, de acordo com os casos específicos de uso para aplicação do algoritmo, conforme descritas na Tabela 4 (3).

Tabela 4 – Comparativo de Aplicabilidade do K-Means

Vantagens	Desvantagens
Eficiência computacional, sendo capaz de lidar com grandes conjuntos de dados e alto dimensionamento	Sensibilidade à Inicialização, como a necessidade de definir previamente o número de clusters
Simplicidade e facilidade de implementação	Clusters de formas irregulares, pois assume que os clusters são esféricos e isotrópicos
Fácil compreensão dos resultados	Alta Sensibilidade a *Outliers*

2.3 Avaliação de Empresas (*Valuation*)

A técnica de *valuation*, ou avaliação de empresas, pode ser aplicada para determinar o valor de um ativo qualquer, seja ele real, como um imóvel, ou financeiro, como uma ação (Núcleo das análises desenvolvidas com a Atomus), ou até mesmo o valor de um título de dívida, como uma debênture; sendo possível ainda, determinar o valor global de uma empresa. Deste modo, o *valuation* procura relacionar o valor do ativo, ao nível de incerteza e expectativa de aumento futuro no fluxo de caixa que o investimento poderá proporcionar.

O grande objetivo do investidor de longo prazo é possuir a capacidade de encontrar empresas sólidas, com a melhor relação de preço-retorno. Para isso, os investidores recorrem a métodos de *Valuation* (Avaliação de Empresas) (7).

Dentro do conceito de *Valuation*, existem múltiplas ramificações para se obter a estimativa de retorno sobre risco de uma empresa, a plataforma Atomus explora a Análise Fundamentalista como um método de *Valuation*, proporcionando uma visão de longo prazo para o investidor, a partir das perspectivas atuais e históricas de cada empresa.

A análise fundamentalista consiste na avaliação de uma empresa, de acordo com sua situação financeira e mercadológica, usando como insumos dados econômicos, indica-

dores do mercado financeiro e balanços dos resultados operacionais das empresas. A fim de facilitar a identificação das perspectivas e oportunidades de mercado, são comumente utilizados alguns indicadores padronizados. Alguns dos principais indicadores para análise utilizados na Atomus, são: P/L[1], P/VP [2], PSR[3], EV/EBIT[4], Dív.Bruta/Patrim [5], ROE[6], Div.Yield[7].

O algoritmo de *valuation* desenvolvido para a Atomus se baseou principalmente em encontrar valores ótimos para cada uma das métricas, combinando métodos estatísticos a abordagens conceituais de grandes investidores, sendo as principais duas grandes obras literárias de finanças utilizadas como referência: "O Investidor Inteligente" - Benjamin Graham e "Ações comuns, Lucros Extraordinários" - Philip Fisher.

Para exemplificação da abordagem conceitual pode-se utilizar o autor Benjamin Graham o qual descreve como o P/L de uma empresa pode ser definido sobrevalorizado ou subvalorizado, ao se comparar o P/L histórico contra o P/L atual, se a relação for 15% superior à média histórica o ativo em questão está sobrevalorizado e se estiver 15% abaixo estará subvalorizado.

O mesmo raciocínio de Graham, pode ser aplicado aos demais *KPIs* (*Key Performance Indicator*), porém com resultados ótimos diferentes, os conceitos abaixo foram utilizados para direcionamento do algoritmo:

1. P/L: Um P/L mais baixo indica que a ação está potencialmente subvalorizada em relação aos seus ganhos. Um P/L mais alto pode indicar que a ação está cara em relação aos seus lucros.
2. P/VP: Um P/VP abaixo de 1 pode indicar que a ação está subvalorizada em relação ao seu valor patrimonial. Um P/VP acima de 1 pode indicar que a ação está sobrevalorizada.
3. PSR: Um PSR mais baixo pode indicar que a ação está subvalorizada em relação à sua receita. Um PSR mais alto pode indicar que a ação está sobrevalorizada em relação à sua receita.
4. EV/EBIT: Um EV/EBIT mais baixo pode indicar que a empresa está subvalorizada em relação ao seu lucro operacional. Um EV/EBIT mais alto pode indicar que a empresa está cara em relação ao seu lucro operacional.
5. Dívida Bruta/Patrimônio: Um índice mais baixo geralmente é considerado mais saudável, pois indica menos alavancagem financeira e menor risco de insolvência.

[1] P/L = Preço atual da Ação / Lucro por Ação referente aos últimos 12 meses
[2] P/VP = Preço atual da Ação / Patrimônio líquido por Ação nos últimos 12 meses
[3] PSR = Preço da Ação / Receita líquida por Ação nos últimos 12 meses
[4] EV/EBIT = (Valor de Mercado + Dívida Líquida) / (Lucro Líquido + Resultado Financeiro + Impostos)
[5] Dív.Bruta/ Patrim = dívida bruta/patrimônio líquido do período
[6] ROE = Lucro líquido no período / Patrimônio líquido no período
[7] Dividend Yield = Proventos pagos por Ação nos últimos 12 meses / Preço por Ação

6. ROE: Essa métrica avalia a capacidade da empresa de gerar lucro em relação ao seu patrimônio líquido. Um ROE mais alto é geralmente preferível, pois indica que a empresa está gerando mais lucro com seu capital próprio.
7. *Dividend Yield*: Um *Dividend Yield* mais alto pode indicar que a empresa está pagando dividendos generosos em relação ao seu preço de mercado.

Cada ativo foi comparado dentro do seu respectivo segmento de atuação, buscando definir valores ótimos a partir da distribuição de cada *KPI*.

A combinação análise fundamentalista pautada nos principais *KPIs* de saúde financeira e as análises estatísticas fornecem informações sobre o momento e valor relativo de uma empresa, auxiliando na tomada de decisão de investidores.

3 Implementação

Durante a implementação do projeto, foram realizadas as etapas 1 e 2, descritas no fluxograma da Figura 1, com objetivo de segmentar as ações listadas na bolsa brasileira de acordo com as 7 *features* de análise fundamentalista, descritas no Capítulo 2. Para tal, foram extraídos os dados dos múltiplos fundamentalistas referentes ao balanço das empresas, divulgados no segundo trimestre de 2024, que serviram como entrada para o teste de 4 diferentes modelos para clusterização de dados, combinados com os métodos de redução de dimensionalidade PCA e t-SNE. Após a implementação dos modelos, todos foram submetidos à análise técnica para avaliação da qualidade e definição dos clusters resultantes, utilizando as métricas descritas na Seção 3.3.

3.1 Implementação das Funções Utilizadas

3.1.1 Fonte e Formatação dos Dados

Os dados utilizados para o experimento foram obtidos da fonte Fundamentus (8), uma plataforma gratuita que consolida os resultados dos balanços trimestrais das empresas de capital aberto, listadas na bolsa brasileira, calculando uma coletânea de múltiplos e indicadores financeiros, úteis para a prática de *valuation* e avaliação fundamentalista das empresas. A forma de extração desses dados foi feita por meio de bibliotecas de *web scraping* para a linguagem de programação Python, como *requests* e *BeautifulSoup* (9), através de um *script* proprietário para obtenção e formatação dos dados. A disposição dos dados extraídos, pode ser visualizada na Figura 5.

Figura 5 – Exemplo da disposição dos dados obtidos via Fundamentus

	Papel	nome_setor	P/L	P/VP	PSR	Div.Yield	P/Ativo	P/Cap.Giro	EV/EBIT	EV/EBITDA	Liq. Corr.	ROIC	ROE	Cresc. Rec.5a
18	FRIO3	Máquinas e Equipamentos	-232.79	-4.57	0.962	0.0000	1.057	-3.23	36.96	22.27	0.68	0.0711	0.0196	0.0928
22	ENAT3	Petróleo, Gás e Biocombustíveis	-161.62	1.89	5.283	0.0054	0.902	7.30	63.41	10.44	1.88	0.0220	-0.0117	0.1363
24	APER3	Previdência e Seguros	-133.06	1.86	2.738	0.0072	1.063	-16.33	21.78	14.62	0.74	0.0529	-0.0140	0.3989
25	SYNE3	Exploração de Imóveis	-131.94	0.87	3.120	0.0000	0.331	3.93	11.15	8.31	5.28	0.0484	-0.0066	-0.0228
27	MRVE3	Construção Civil	-125.91	0.56	0.505	0.0000	0.150	0.62	40.17	27.17	2.07	0.0116	-0.0044	0.0417
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
956	JALL3	Alimentos Processados	102.15	1.07	1.325	0.0587	0.330	1.19	48.13	6.29	3.88	0.0194	0.0105	0.0980
957	PETZ3	Comércio	103.54	0.94	0.533	0.0047	0.424	2.74	9.54	3.39	1.81	0.0564	0.0090	0.3522
968	HBSA3	Transporte	176.71	2.25	1.616	0.0000	0.498	5.39	17.42	9.10	1.78	0.0719	0.0127	0.1767
982	CEAB3	Comércio	1452.12	1.13	0.505	0.0000	0.360	2.73	6.76	2.89	1.39	0.0847	0.0008	0.0936
985	PGMN3	Comércio e Distribuição	6906.22	0.57	0.135	0.1060	0.168	1.21	7.17	2.99	1.39	0.0571	0.0001	0.1509

317 rows × 14 columns

Fonte: Própria (2024)

3.1.2 Scikit-Learn

O Scikit-Learn (10) é um dos pacotes mais famosos para modelagem de dados em Python. Ele conta com inúmeras classes e funções úteis na preparação, treinamento e

avaliação de modelos. Este pacote foi utilizado diversas vezes durante o projeto para implementação das técnicas de clusterização introduzidas no Capítulo 2, além do tratamento e normalização dos dados, através das técnicas de escalonamento e redução de dimensionalidade, bem como cálculo das métricas de avaliação dos algoritmos de clusterização, introduzidas adiante.

3.2 Métodos para Redução de Dimensionalidade

Durante a fase de pré-processamento dos dados para implementação dos algoritmos de clusterização, foram utilizadas duas técnicas distintas para a redução de dimensionalidade das features utilizadas para clusterizar as ações analisadas. Desta forma, o espaço de *N-features*, foi reduzido por meio de componentes bidimensionais (2D), possibilitando a visualização e plot no plano bidimensional, bem como eliminando os efeitos de multicolinearidade e sobreajuste, inerentes à correlação do espaço multidimensional das variáveis independentes utilizadas (11).

3.2.1 PCA

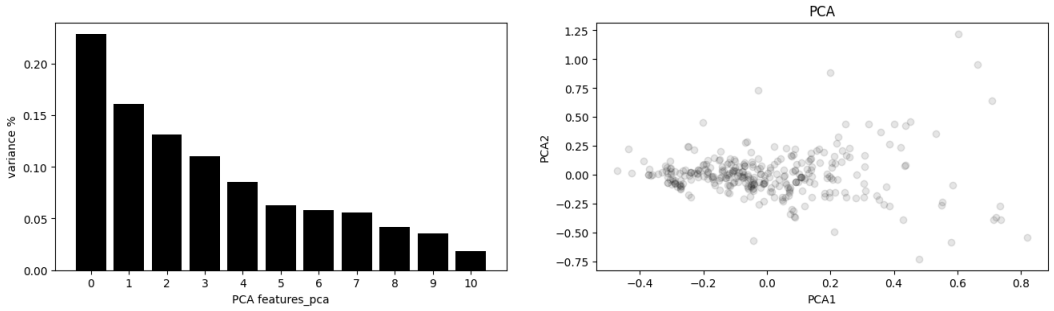
A análise de componentes principais (PCA) é uma técnica de redução de dimensionalidade linear que transforma variáveis potencialmente correlacionadas em um conjunto menor de variáveis chamadas de componentes principais. A PCA reduz o número de dimensões e, ao mesmo tempo, retém o máximo de informações do conjunto de dados original (11).

A PCA é um algoritmo de aprendizado não supervisionado que é comumente usado por cientistas de dados para o pré-processamento de dados. Ele transforma um grande conjunto de dados em um espaço de menor dimensão, extraindo os recursos mais informativos e preservando as informações mais relevantes do conjunto de dados inicial. Esse pré-processamento de dados reduz a complexidade do modelo, pois a adição de cada novo recurso afeta negativamente o desempenho do modelo, o que também é comumente chamado de “maldição da dimensionalidade”.

Ao projetar dados de alta dimensão em um espaço de recursos menor, a PCA também minimiza, ou elimina completamente, problemas comuns como multicolinearidade e sobreajuste. Essa simplificação dos dados também tem implicações para as tarefas de visualização de dados e seu uso adicional com outros algoritmos de aprendizado de máquina. Os dados em um espaço de menor dimensão melhoram a interpretabilidade das visualizações de dados e o desempenho geral do modelo.

A Figura 6 ilustra um exemplo de redução bidimensional, utilizando o método PCA aplicado ao *dataset* extraído, bem como a variância explicada das componentes resultantes.

Figura 6 – Exemplo do Método de PCA e Variância Explicada das Componentes



Fonte: Própria (2024)

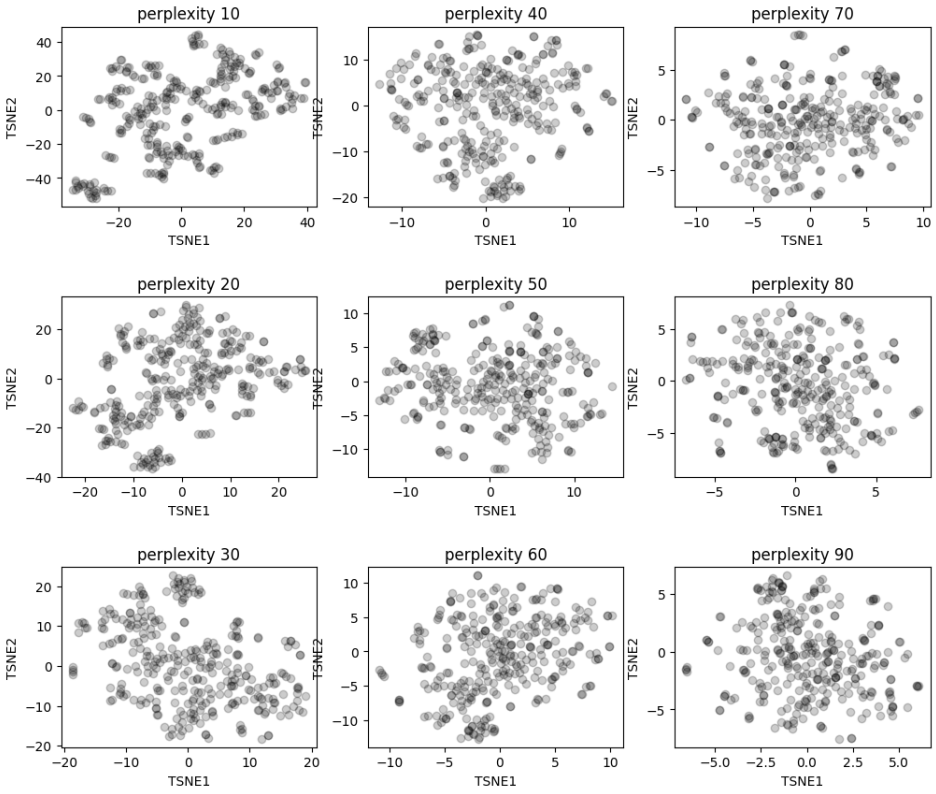
3.2.2 t-SNE

O *T-Stochastic Neighbor Embedding* (t-SNE) é uma técnica de redução de dimensionalidade, que preserva as semelhanças entre os pares de pontos de dados, utilizando uma distribuição t para calcular as semelhanças entre os pares de dados. Comparativamente com seu antecessor, o *Stochastic Neighbor Embedding* (SNE), que usa distribuições gaussianas para modelar semelhanças, o t-SNE se permite uma modelagem mais precisa de pequenas distâncias entre pares, preservando assim as estruturas locais de forma eficaz (12).

O algoritmo t-SNE, começa definindo semelhanças de pares entre pontos de dados no espaço de alta dimensão usando um *kernel* gaussiano. Em seguida, ele constrói uma distribuição de probabilidade semelhante no espaço de dimensão inferior, ajustando iterativamente as posições dos pontos de dados para minimizar a incompatibilidade entre essas distribuições.

Esse processo envolve a minimização da divergência de Kullback-Leibler entre as probabilidades conjuntas dos pontos de dados nos espaços original e de dimensão inferior. O t-SNE, tem como principal parâmetro de implementação, a perplexidade efetiva, que determina o número de vizinhos mais próximos considerados durante o processo de incorporação, conforme ilustrado na Figura 7.

Figura 7 – Exemplo de variação do parâmetro de Perplexidade: t-SNE



Fonte: Própria (2024)

3.3 Métricas de Avaliação dos algoritmos

3.3.1 Silhouette Score

O *Silhouette Score* é uma métrica que avalia a separação e coesão entre os clusters, fornecendo uma avaliação sobre a qualidade de definição dos clusters, considerando sobreposição e distância intra-cluster. Seus valores podem variar entre -1 (agrupamento ruim) a +1 (agrupamento perfeito). Uma pontuação próxima a 1 sugere clusters bem separados. O *Silhouette Score* é definido por:

$$S(i) = \frac{(b(i) - a(i))}{\max(a(i), b(i))} \tag{3.1}$$

Onde:

- $a(i)$ É a distância média do ponto de dados (i) para os outros pontos do mesmo cluster
- $b(i)$ É a menor distância média do ponto de dados (i) para os outros pontos de um cluster diferente

3.3.2 Davies-Bouldin Index

O Índice Davies-Bouldin avalia a compactação e a separação dos clusters de um conjunto de dados. Os clusters mais bem definidos são indicados por um Índice de Davies-Bouldin mais baixo, que é determinado pela comparação da proporção média de similaridade/dissimilaridade de cada cluster com a de seu vizinho mais similar. Os clusters com as menores distâncias intra-cluster e as maiores distâncias inter-cluster fornecem um índice mais baixo, sendo particularmente útil para auxiliar na determinação do número ideal de clusters. O Índice Davies-Bouldin é definido por:

$$DB = \left(\frac{1}{n}\right) \sum \max(R_{ij}) \tag{3.2}$$

Onde:

- n é o número de clusters.
- R_{ij} é uma medida de dissimilaridade entre o cluster (i) e o cluster mais semelhante a i.

3.3.3 Calinski-Harabasz Index

O Índice Calinski-Harabasz é uma métrica usada para avaliar a qualidade dos clusters em um conjunto de dados, utilizando a razão entre a variância dentro do cluster e a variância entre clusters. Valores mais altos indicam clusters compactos e bem separados. O Índice Calinski-Harabasz é definido por:

$$CH = \left(\frac{B}{W}\right) \left(\frac{N-K}{K-1}\right) \tag{3.3}$$

Onde:

- B é a soma das distâncias quadráticas entre clusters
- W é a soma das distâncias quadráticas dentro dos clusters
- N é o número total de pontos de dados.
- K é o número de clusters.

4 Resultados e Discussão

Após a implementação dos modelo, todos os 4 modelos de clusterização foram avaliados, seguindo a etapa 2, descrita no fluxograma da Figura 1, visando selecionar as melhores combinações de algoritmos para clusterização e métodos para redução de dimensionalidade implementadas ao *dataset*. Todos os resultados obtidos foram analisados tecnicamente, utilizando as métricas de coesão e separação dos clusters, durante a Seção 4.1.

Definidos os métodos tecnicamente mais adequados ao *dataset* utilizado, foi realizado um experimento através da combinação entre os dados dos clusters resultantes para cada algoritmo, com o processo de *valuation*, visando avaliar a performance de portfólios teóricos implementados a partir da seleção das 21 ações, distribuídas dentro de ao menos 3 diferentes clusters para cada algoritmo, visando criar carteiras com bons fundamentos e devidamente diversificadas, diminuindo vieses e riscos intrínsecos ao processo de *valuation* na composição de portfólios de investimento, conforme descrito na etapa 3 do fluxograma da Figura 1.

Por fim, após os experimentos, todas as carteiras resultantes foram avaliadas seguindo métricas de risco-retorno dos portfólios, seguindo a etapa 4 do fluxograma da Figura 1, visando definir empiricamente, qual dos algoritmos seria capaz de obter a melhor performance no processo de composição de carteiras. Após este processo experimental, todo o estudo foi disponibilizado no *website* "Atomus Invest" (13), com objetivo de democratizar as discussões e resultados obtidos durante o estudo para os investidores, seguindo a etapa 5 do fluxograma da Figura 1.

4.1 Algoritmos de Clusterização

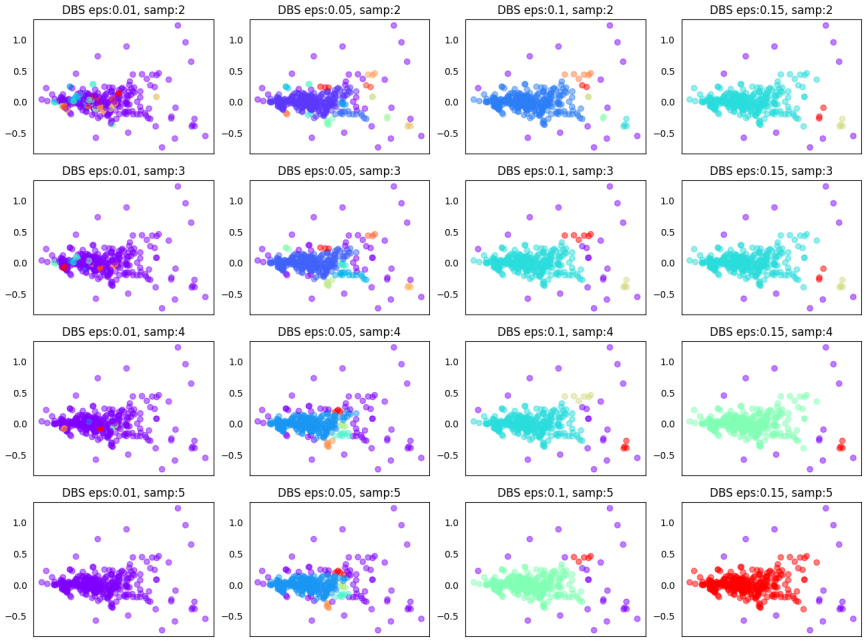
Durante a etapa de clusterização, foram experimentados 4 tipos diferentes de algoritmos para este fim, sendo: DBScan, *Hierarchical Clustering*, *Affinity Propagation* e K-Means, utilizando dois métodos distintos para redução de dimensionalidade em cada algoritmo: PCA e t-SNE. Em todas as aplicações, as 7 *features* utilizadas, descritas no Capítulo 2, foram reduzidas para 2 dimensões, possibilitando uma visualização bidimensional dos dados.

Para o PCA, foram selecionadas por padrão, as duas melhores componentes (PC), tomando por critério de seleção a variância explicada. Para o método de t-SNE, foram adotadas as duas componentes obtidas a partir de uma perplexidade efetiva de 20, possibilitando maior distribuição espacial dos clusters.

4.1.1 DBScan - PCA e t-SNE

O algoritmo do DBScan (14) aplicado com a redução de dimensionalidade PCA para reduzir as dimensões das features para o espaço bidimensional, foi implementado através de uma matriz de sensibilidade, variando os parâmetros *epsilon* com os valores [0.01, 0.05, 0.1, 0.15] e o parâmetro *Min Points* com os valores de [2,3,4,5]. Os resultados obtidos podem ser vistos na Figura 8.

Figura 8 – Matriz de Sensibilidade dos parâmetros - DBScan (PCA)



Fonte: Própria (2024)

Analisando os resultados das métricas de avaliação do algoritmo, na Tabela 5, foi possível concluir que o melhor resultado foi obtido com 2 clusters e parâmetros epsilon = 0.15 e Min Points = 5, contudo a concentração de mais 90% da amostra em um único cluster, evidenciando uma falta de acurácia do modelo, durante os testes de sensibilidade.

Tabela 5 – Avaliação Algoritmo DBScan (PCA)

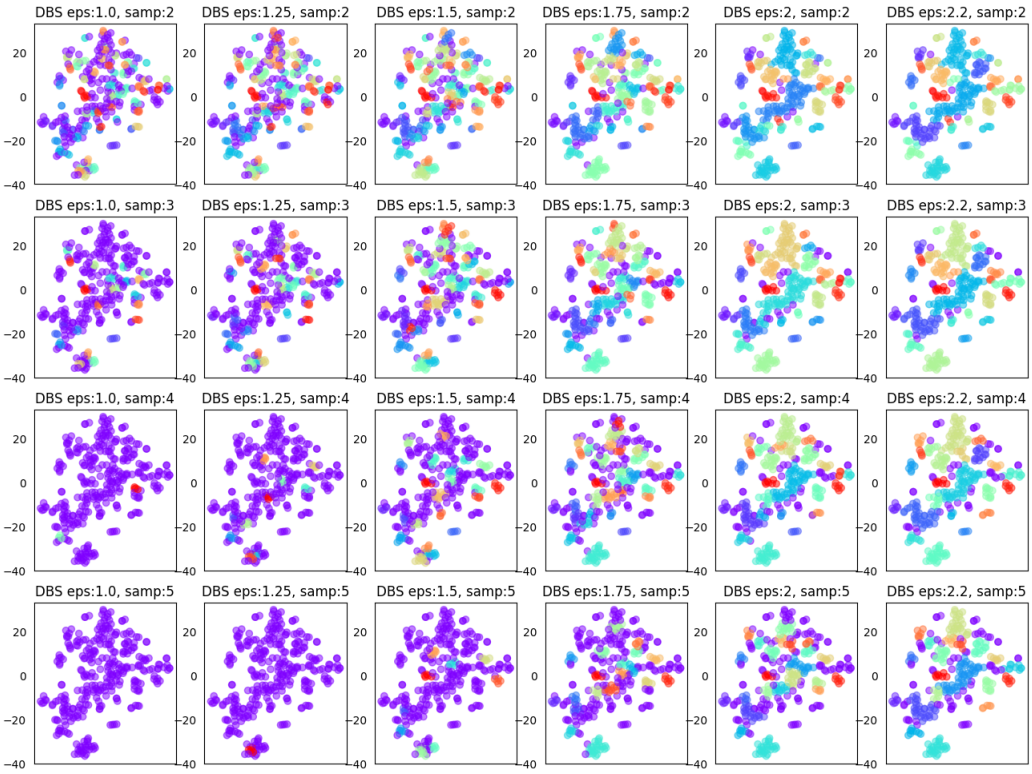
no_clusters	silhouette_score	db_index	ch_index	eps	samp	no_noise
2	0,60	1,64	53,19	0,15	5	18
3	0,57	1,79	33,58	0,15	4	14
3	0,49	1,56	48,12	0,10	5	26
4	0,48	1,97	24,53	0,15	2	11
4	0,48	1,97	24,53	0,15	3	11
4	0,47	1,67	40,11	0,10	3	21
4	0,47	1,67	40,11	0,10	4	21
7	0,36	3,73	21,21	0,10	2	15

Tabela 5 – Avaliação Algoritmo DBScan (PCA)

6	0,07	2,92	20,31	0,05	5	70
6	0,06	3,03	18,46	0,05	4	63
9	0,05	2,13	18,53	0,05	3	43
15	-0,16	1,87	13,60	0,05	2	31
42	-0,26	1,78	1,38	0,01	2	215
14	-0,44	1,90	1,54	0,01	3	271
7	-0,51	2,44	1,01	0,01	4	293

O algoritmo do DBScan aplicado com a redução de dimensionalidade t-SNE para reduzir as dimensões das *features* para o espaço bidimensional, foi implementado através de uma matriz de sensibilidade, variando os parâmetros epsilon com os valores [1, 1.25, 1.5, 1.75, 2, 2.2] e o parâmetro *Min Points* (samp) com os valores de [2,3,4,5]. Os resultados obtidos podem ser vistos na Figura 9.

Figura 9 – Matriz de Sensibilidade dos parâmetros - DBScan (t-SNE)



Fonte: Própria (2024)

Analisando os resultados das métricas de avaliação do algoritmo, na Tabela 6, foi possível concluir que apesar do *Silhouette Score* inferior, relativamente ao método com PCA, o algoritmo aplicado com t-SNE, teve melhores valores de *Davies-Bouldin Index* e Calinski-Harabaz, além de conseguir encontrar uma maior quantidade de clusters. Os

melhores resultados, foram obtidos com o parâmetro *Min Points* (samp) assumindo o valor 2.

Tabela 6 – Avaliação Algoritmo DBScan (t-SNE)

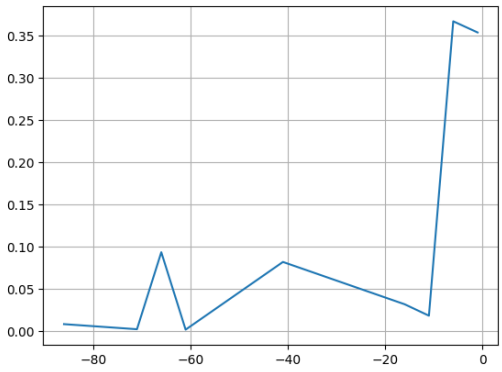
no_clusters	silhouette_score	db_index	ch_index	eps	samp	no_noise
55	0,369	1,466	63,814	1,75	2	25
30	0,328	1,167	27,120	1,5	2	46
73	0,274	1,606	72,612	2	2	41
41	0,269	1,210	128,335	2,2	2	7
39	0,267	1,526	37,346	1,75	4	9
35	0,265	0,996	161,529	2,2	3	3
28	0,247	1,608	103,195	2,2	3	17
2	0,243	0,533	15,978	1,25	5	312
31	0,242	1,419	77,875	2	3	7
27	0,224	1,619	45,870	2	4	58
21	0,189	1,697	50,384	2,5	5	40
82	0,177	1,213	8,543	1,25	2	92
46	0,118	1,381	18,242	1,5	3	100
33	0,096	1,548	18,347	1,75	4	111
24	0,048	1,734	20,520	2	5	4
71	-0,017	1,187	4,884	1	2	147
20	-0,156	1,832	13,732	1,75	5	185
36	-0,188	1,453	7,090	1,25	4	184
23	-0,224	1,568	11,293	1,5	4	198
3	-0,288	1,136	5,282	1	4	309
24	-0,392	1,593	4,138	2	3	241
10	-0,422	1,562	10,171	1,5	5	267
11	-0,503	1,807	6,333	1,25	4	275

4.1.2 Affinity Propagation - PCA e t-SNE

O algoritmo do *Affinity Propagation* (5) foi aplicado com as reduções de dimensionalidade PCA e t-SNE, reduzindo as dimensões das *features* para o espaço bidimensional. Durante a implementação, foram considerados diferentes valores para o parâmetro "*preference*" em cada algoritmo de redução de dimensionalidade.

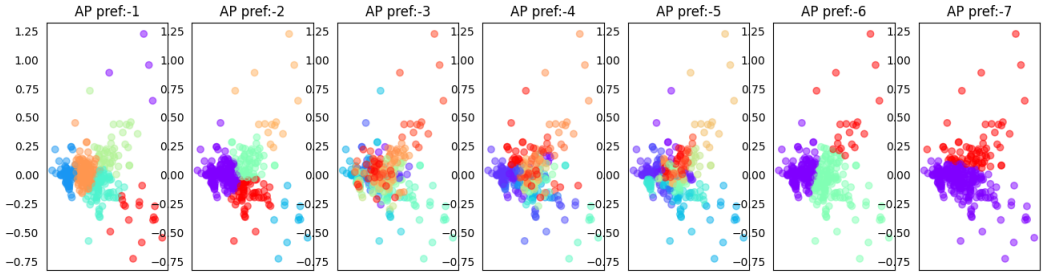
Durante a determinação do parâmetro ideal de "*preference*" com a redução de dimensionalidade PCA, foi considerada a relação do plot do score de *silhouette* em função do parâmetro *preference*, demonstrado na Figura 10, além da avaliação visual do *output* dos clusters resultantes, observados na Figura 11.

Figura 10 – Silhouette Score x Parâmetro Preference: AP (PCA)



Fonte: Própria (2024)

Figura 11 – Output AP (PCA): Sensibilidade com variação de preference



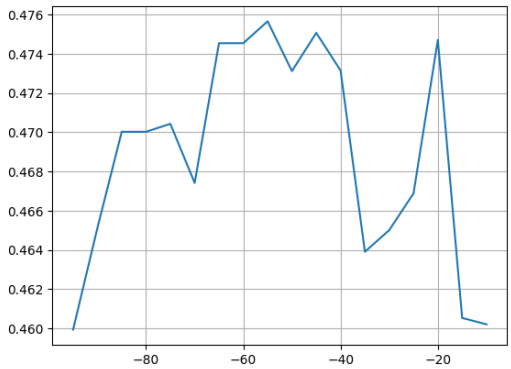
Fonte: Própria (2024)

Tabela 7 – Avaliação Algoritmo AP (PCA)

no_clusters	silhouette_score	db_index	ch_index	pref
5	0,395	0,814	215,60	-2
3	0,367	0,890	178,51	-6
6	0,356	0,809	227,13	-1
2	0,333	1,277	92,87	-7
75	0,132	0,833	29,14	-5
129	0,071	0,575	17,69	-4
236	-0,001	0,502	3,88	-3

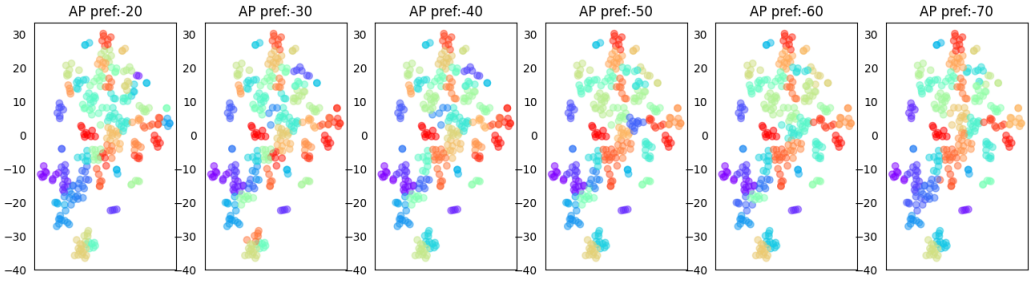
Similarmente, durante a determinação dos parâmetro ideal de "*preference*" com a redução de dimensionalidade t-SNE, foi considerada a relação do plot do score de *silhouette* em função do parâmetro *preference*, demonstrado na Figura 12, além da avaliação visual do *output* dos clusters resultantes, observados na Figura 13.

Figura 12 – Silhouette Score x Parâmetro Preference: AP (t-SNE)



Fonte: Própria (2024)

Figura 13 – Output AP (t-SNE): Sensibilidade com variação de preference



Fonte: Própria (2024)

Tabela 8 – Avaliação Algoritmo AP (t-SNE)

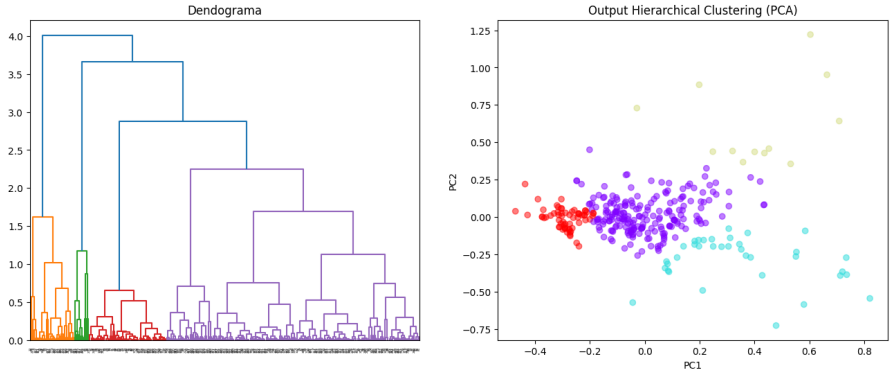
no_clusters	silhouette_score	db_index	ch_index	pref
60	0,47470	0,5861	764,21	-20
39	0,47454	0,6295	660,00	-60
45	0,47315	0,6212	692,01	-40
40	0,47311	0,6320	666,57	-50
36	0,46742	0,6369	630,08	-70
51	0,46501	0,6360	729,72	-30

O resultado do algoritmo em ambas as redução não tiveram *overfit*, sendo que a redução por t-SNE que obteve um maior *Calinski-Harabaz Index*, resultou em 60 clusters, o qual apesar da melhora em todas as métricas de performance, torna-se de difícil compreensão para uma análise generalista por parte de um investidor. A redução por PCA apresentou resultados inferiores em comparação a redução por t-SNE em todos os parâmetros analisados: *Silhouette Score*, *Davies-Bouldin Index* e *Calinski-Harabaz Index*, porém mantendo a análise com apenas 5 clusters, sendo mais indicada para o caso de uso da plataforma.

4.1.3 Hierarchical Clustering - PCA e t-SNE

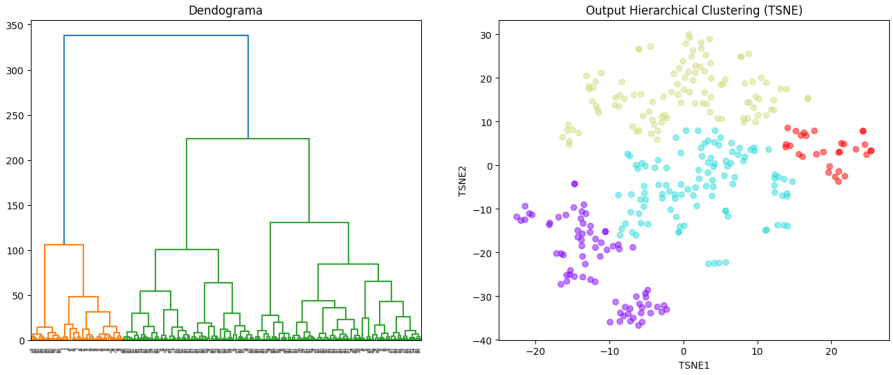
O algoritmo *Hierarchical Clustering* (6) foi aplicado com as reduções de dimensionalidade PCA e t-SNE, reduzindo as dimensões das features para o espaço bidimensional, resultando na construção dos dendrogramas que podem ser vistos nas figuras 14 e 15:

Figura 14 – Dendograma e Output: Hierarchical Clustering (PCA)



Fonte: Própria (2024)

Figura 15 – Dendograma e Output: Hierarchical Clustering (t-SNE)



Fonte: Própria (2024)

Tabela 9 – Avaliação Algoritmo Hierarchical Clustering

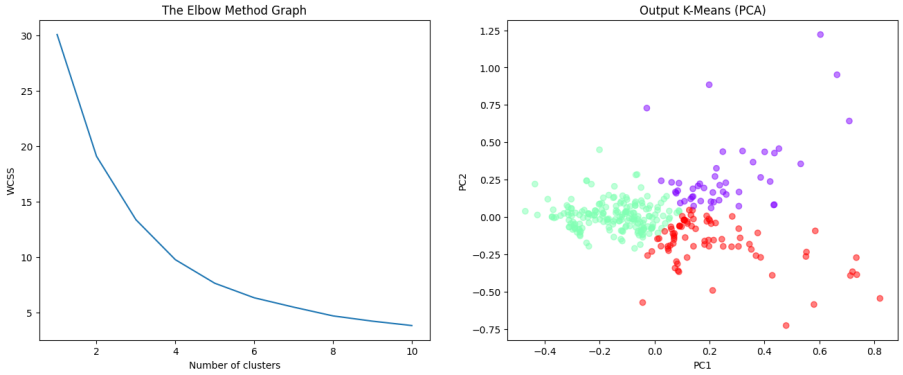
no_clusters	silhouette_score	db_index	ch_index	method	metric	linkage
4	0,320	0,805	176,39	PCA	euclidean	ward
4	0,367	0,811	314,62	t-SNE	euclidean	ward

O resultado do algoritmo em ambos os tipos de redução conseguiu criar clusters bem distribuídos e sem *overfit*, sendo 4 o número de clusters ideal definido pelo método. A redução por PCA apresentou resultados inferiores em comparação a redução por t-SNE em todos os parâmetros analisados: *Silhouette Score*, *Davies-Bouldin Index* e *Calinski-Harabaz Index*.

4.1.4 K-Means - PCA e t-SNE

O algoritmo K-Means (15) foi aplicado com as reduções de dimensionalidade PCA e t-SNE, reduzindo as dimensões das features para o espaço bidimensional, resultando na segmentação das Figuras 16 e 17. Em ambos os métodos, foi utilizada a análise de *Elbow* para determinar o número ideal de clusters, que considera a soma cumulativa do quadrado das distâncias intra-clusters, resultando na definição de 3 clusters como parâmetro do algoritmo de K-Means.

Figura 16 – Gráfico de Elbow e Output K-Means (PCA), n = 3 clusters

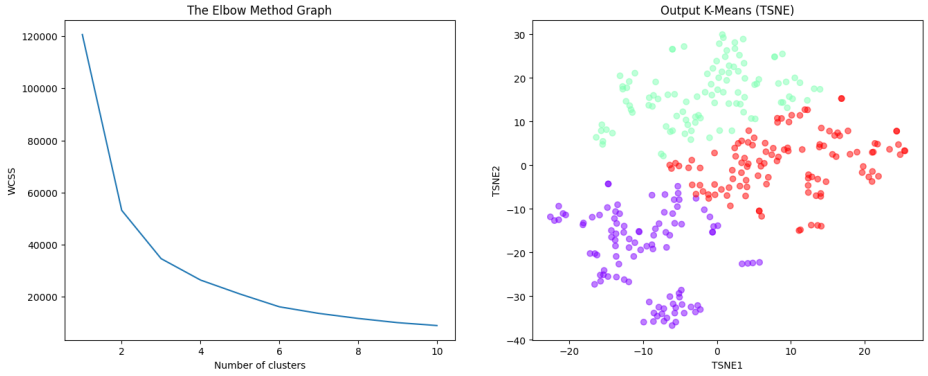


Fonte: Própria (2024)

Tabela 10 – Avaliação Algoritmo K-Means (PCA)

no_clusters	silhouette_score	db_index	ch_index	method
3	0,4346	0,9224	196,458	KMeans - PCA

Figura 17 – Gráfico de Elbow e Output K-Means (t-SNE), n = 3 clusters



Fonte: Própria (2024)

Tabela 11 – Avaliação Algoritmo K-Means (t-SNE)

no_clusters	silhouette_score	db_index	ch_index	method

Tabela 11 – Avaliação Algoritmo K-Means (t-SNE)

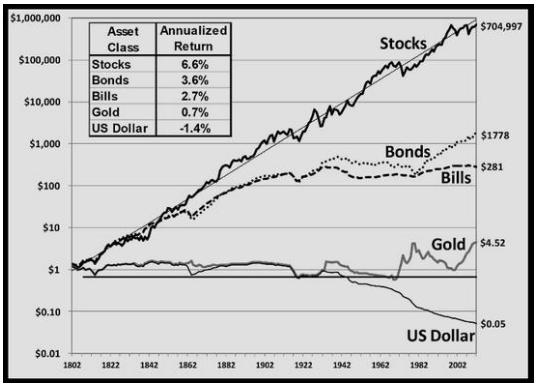
3	0,4267	0,8823	390,985	KMeans - t-SNE

O resultado do algoritmo em ambos os tipos de redução conseguiu criar clusters bem distribuídos e sem *overfit*. A redução por PCA apresentou resultados inferiores em comparação a redução por t-SNE em dois parâmetros analisados: *Davies-Bouldin Index* e *Calinski-Harabaz Index*, porém com resultado superior no *Silhouette Score*.

4.2 Experimentos e Benchmark

Após a implementação dos algoritmos de clusterização descritos na Seção 4.1, foram conduzidos experimentos, a partir dos 4 algoritmos, visando otimizar a construção de um portfólio teórico, com foco no investidor de longo prazo, baseado em estratégias de investimentos objetivando resultados no médio-longo prazo.

Figura 18 – Retorno de diferente classes de ativos ao longo do tempo em US dólares



Fonte: (SIEGEL, 2014)

A figura 18, extraída do livro *Stocks for the Long Run* (2) corrobora com a estratégia de investimento de longo prazo, baseada na avaliação fudamentalista, dentro do mercado acionário, demonstrando que o mercado através das décadas, acompanha os lucros das empresas, garantindo um retorno superior à outras classes de ativos no longo prazo. O gráfico demonstra um retorno de 6.6% a.a para ações, enquanto o *US Dollar* teve uma queda de 1,4%, corroborando que ações retém mais valor do que moedas fiduciárias e até mesmo reservas de valor, como o ouro.

Visando diluir o risco entre os ativos que compõem os portfólios teóricos, a técnica de clusterização, foi utilizada em conjunto com o algoritmo de avaliação fundamentalista, visando escolher os ativos com os melhores fundamentos disponíveis, de forma descorrelacionada, ao selecionar as ativos com melhores avaliações de cada um dos Clusters.

Para isto, foi realizado um benchmark teórico, utilizando as 4 técnicas de clusterização com melhor avaliação nos parâmetros de *Silhouette Score*, *Davies-Bouldin Index*

e *Calinski-Harabasz Index*, sendo as técnicas de DBScan - t-SNE, *Hierarchical Clustering* - t-SNE, *Affinity Propagation* - PCA e K-Means - PCA.

Partindo destas técnicas, foram avaliadas a performance teórica de um portfólio das melhores 21 ações dos clusters, de acordo com a avaliação fundamentalista, os portfólios serão avaliados seguindo as métricas de retorno total, nas janelas de 6 anos, 4 anos, 12 meses, 6 meses e YTD *(Year to Date)*, Volatilidade dos portfólios, correlação entre os ativos escolhidos e VaR (*Value at Risk*).

4.2.1 Benchmark Carteira - Métricas de Avaliação do Experimento

1. **Rentabilidade**

   A rentabilidade pode ser definida como o percentual de remuneração obtido a partir da quantia que você investiu, podemos resumir que é o valor que será obtido no retorno de uma aplicação. No mercado acionário a rentabilidade se dá a partir da valorização do ativo investido seja por meio do preço ou distribuição de dividendos.

$$\text{Rentabilidade (\%)} = \frac{\text{Preço Atual Ativo}}{\text{Preço de Compra Ativo}} - 1 \tag{4.1}$$

   Foram utilizadas 5 janelas temporais para analisar o retorno do portfólio, 6 anos, 4 anos, 12 meses, 6 meses e *YTD*, para assegurar consistência do portfólio ao longo do tempo.

   A rentabilidade de investimento é um indicador essencial para avaliar o desempenho e a viabilidade de opções de investimento. Compreender como calculá-la e os fatores que a influenciam ajuda investidores a tomar decisões informadas e a otimizar seus portfólios.

2. **Volatilidade**

   A volatilidade de investimento é uma medida estatística que indica a intensidade das variações no preço de um ativo ao longo do tempo. Em termos mais simples, ela reflete o grau de risco ou incerteza sobre as mudanças no valor do investimento. Altas volatilidades significam grandes flutuações nos preços, enquanto baixas volatilidades indicam estabilidade. O cálculo da volatilidade é usualmente dado pelo desvio padrão dos retornos do investimento. O desvio padrão mede a dispersão dos retornos em torno da média.

   Desta forma, a volatilidade é uma medida crucial para entender e avaliar o risco associado a um ativo, oferecendo insights sobre a estabilidade do investimento e ajuda os investidores a equilibrar seu portfólio de acordo com o de risco do investidor.

   A Volatilidade é obtida em 3 etapas, cálculo retorno diário, média do retorno diário e desvio padrão. O retorno diário é dado por:

$$\text{Retorno Diário } (\%)_{Rd} = \frac{\text{Preço dia}}{\text{Preço dia anterior}} - 1 \tag{4.2}$$

A média dos retornos é dada pela somatória dos retornos dividido pelo número de dias/retorno (n)

$$\mu = \frac{1}{n}\sum_{i=1}^{n} Rd \tag{4.3}$$

A volatilidade é o desvio padrão dos retornos diários sendo calculada como:

$$\sigma = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(Rd-\mu)^2} \tag{4.4}$$

Alguns fatores impactam impactaram diretamente a volatilidade, como:

- Natureza do Ativo: Diferentes ativos têm diferentes níveis de volatilidade. Ações, por exemplo, tendem a ser mais voláteis do que títulos de dívida (como *bonds*).
- Horizonte Temporal: A volatilidade pode variar com o horizonte temporal considerado. Volatilidades de curto prazo podem ser muito diferentes das de longo prazo.
- Condições de Mercado: Eventos econômicos, políticos e sociais podem impactar a volatilidade. Mercados em crises ou sob incertezas tendem a ser mais voláteis.
- Setor Econômico: Alguns setores são intrinsecamente mais voláteis que outros. Tecnologias e biotecnologias, por exemplo, podem apresentar maiores flutuações em comparação com utilidades públicas.
- Liquidez: Ativos com alta liquidez (facilidade de compra e venda) tendem a ter menor volatilidade, enquanto ativos menos líquidos podem experimentar grandes variações de preço.

3. **VaR (*Value at Risk*)**

   É utilizado para quantificar o risco de perda de um investimento ou portfólio de investimentos. Ele estima a perda potencial máxima em um determinado horizonte de tempo com um nível de confiança específico. O VaR é amplamente utilizado em finanças para gerenciar e controlar o risco de mercado.

   O VaR pode ser calculado de várias maneiras, incluindo métodos paramétricos, históricos e de simulação de Monte Carlo. Para o método paramétrico (assumindo uma distribuição normal dos retornos), são necessários alguns dados e definições:

   i) Estimativa da Volatilidade e Retorno Médio: σ e μ calculados no item 2.
   ii) Determinação do intervalo de confiança: $(1-\alpha)$ entre 95% a 99%.
   iii) Cálculo do valor crítico: utilize a distribuição normal padrão para encontrar o valor crítico (Z_a) correspondente ao nível de confiança.

Com isso o VaR em um horizonte de tempo(T) e aporte inicial (V_0) é obtido por:

$$VaR = -\left(\mu T + Z_\alpha \sigma T\right) V_0 \tag{4.5}$$

O *Value at Risk* (VaR) é uma ferramenta essencial para medir e gerenciar o risco de mercado de investimentos e portfólios, assim como a volatilidade gerenciar o VaR e os fatores que o influenciam permite uma melhor gestão do risco.

4. **Correlação Média dos Pares de Ativos**

   O cálculo da correlação média entre pares de ativos de uma carteira de ações é uma medida importante para avaliar o comportamento conjunto dos ativos em relação ao seu movimento de preços. A correlação entre dois ativos mede o grau de dependência linear entre eles, sendo expressa por um valor entre -1 e 1.

   A correlação de Pearson entre dois ativos X_i e X_j é dada pela fórmula:

$$p_{ij} = \frac{\text{Cov}(X_i, X_j)}{\sigma i \cdot \sigma j} \tag{4.6}$$

   Onde:

   - p_{ij} é o coeficiente de correlação entre os ativos X_i e X_j.
   - $Cov(X_i, X_j)$ é a covariância entre os retornos dos ativos X_i e X_j.
   - $\sigma_i \sigma_j$ são os desvios-padrão dos retornos dos ativos X_i e X_j.

   A correlação média é obtida somando todas as correlações p_{ij}, para $i \neq j$ e dividindo pelo número de pares:

$$p_{\text{média}} = \frac{2}{N \cdot (N-1)} \sum p_{ij} \tag{4.7}$$

4.2.2 Benchmark Carteira - Avaliação de Resultados do Experimento

Para avaliação de performance, foram adotados 4 algoritmos: O algoritmo do DBSCAN (16) foi aplicado com a redução de dimensionalidade t-SNE e parâmetros epsilon = 1.75 e Min Points = 2, resultando em 55 clusters. O algoritmo *Hierarchical Clustering* foi aplicado com a redução de dimensionalidade t-SNE, resultando em 4 clusters. O Algoritmo Affinity Propagation, com a redução de dimensionalidade PCA, foi aplicado com parâmetro preference = -2, resultando em 5 clusters. O Algoritmo K-Means, foi aplicado com a redução de com a redução de dimensionalidade PCA, resultando em 3 clusters. O comparativo das métricas de avaliação técnica dos algoritmos está detalhado na Tabela 12.

Tabela 12 – Comparativo Métricas Avaliação Algoritmos

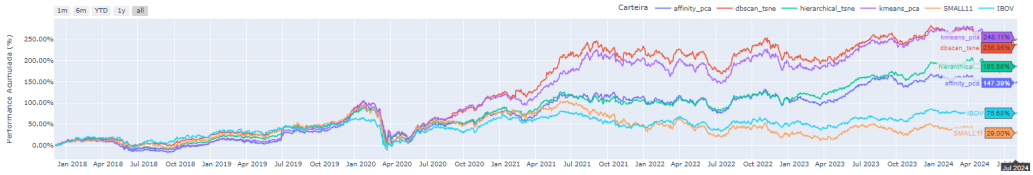
Algoritmo	Clusters	Silhouette Score	Davies-Bouldin	Calinski-Harabasz
DBScan + PCA	2	0,600	1,640	53,190
DBScan + TSNE	55	0,369	1,466	63,814
AP + PCA	5	0,395	0,814	215,600
AP + TSNE	60	0,475	0,586	764,210
HC + PCA	4	0,320	0,805	176,390
HC + TSNE	4	0,367	0,811	314,620
K-Means + PCA	3	0,435	0,922	196,458
K-Means + TSNE	3	0,427	0,882	390,985

Utilizando estas configurações, foram selecionadas as 21 ações com maior avaliação fundamentalista de cada cluster, para cada um dos algoritmos, seguindo uma limitação máxima de 7 ativos por cluster. Os resultados obtidos nas métricas de avaliação podem ser vistos nas Tabelas 13, 14, 15, 16.

1. **Resultados Rentabilidade**

A Figura 19 ilustra um comparativo de rentabilidade das carteiras resultantes dos métodos de clusterização e *valuation*, implementados com os dados de indicadores fundamentalistas do Segundo Trimestre de 2024, comparando o retorno acumulado de 2018 para cada um dos portfólios, com outros índices de mercado, como o IBOV e SMALL11.

Figura 19 – Gráfico Comparação Rentabilidade 2018-TD: Experimento Benchmark



Fonte: Própria (2024)

Tabela 13 – Benchmark Comparativo Rentabilidade

Algoritmo	2018-TD	2020-TD	6m	12m	YTD
kmeans_pca	248.5%	123.4%	-5.2%	0.4%	-6.7%
dbscan_tsne	238.3%	117.1%	-8.1%	-5.9%	-11.5%
hierarchical_tsne	185.9%	83.4%	-1.9%	7.9%	-3.6%
affinity_pca	147.7%	67.1%	-4.3%	-4.0%	-7.6%

Analisando os dados de rentabilidade das quatro carteiras geradas pelos diferentes algoritmos de clusterização, observa-se que o método K-Means com PCA se destaca, principalmente em horizontes de longo prazo (2018-TD e 2020-TD), apresentando as maiores

rentabilidades acumuladas de 248,5% e 123,4%, respectivamente. Embora esse método tenha um desempenho negativo no curto prazo (6m, YTD), sua performance superior em períodos mais longos sugere que ele é mais eficaz para estratégias de investimento que priorizam retornos de longo prazo.

Por outro lado, o algoritmo *hierarchical clustering* com t-SNE apresentou maior equilíbrio entre os períodos analisados, sendo o único a obter uma rentabilidade positiva no horizonte de 12 meses (+7,9%), e uma menor queda no YTD (-3,6%). Destacando-se com maior estabilidade nos períodos curtos em comparação com os demais algoritmos, que sofreram perdas mais expressivas.

2. **Resultados Volatilidade Diária**

Tabela 14 – Benchmark Comparativo Vol. Diária

Algoritmo	2018-TD	2020-TD	6m	12m	YTD
affinity_pca	1.429%	1.530%	0.796%	0.816%	0.785%
dbscan_tsne	1.363%	1.486%	0.825%	0.789%	0.787%
kmeans_pca	1.351%	1.443%	0.836%	0.801%	0.803%
hierarchical_tsne	1.306%	1.415%	0.781%	0.738%	0.738%

Ao avaliar a volatilidade diária das diferentes carteiras, observa-se que o método hierarchical clustering com t-SNE apresenta a menor volatilidade em todos os horizontes de tempo analisados. Especificamente, este algoritmo obteve uma volatilidade de 1,306% no período de 2018-TD e 1,415% em 2020-TD, além de manter a volatilidade mais baixa nos períodos de curto prazo, como nos últimos 6 meses (0,781%), 12 meses (0,738%), e no acumulado do ano (YTD) (0,738%). Esses resultados indicam que o hierarchical com t-SNE é o método mais eficaz na redução da volatilidade, tornando-se uma escolha preferencial para investidores que buscam minimizar o risco em diferentes horizontes temporais.

Em comparação, o método affinity propagation com PCA apresentou as maiores volatilidades em quase todos os períodos, exceto no acumulado do ano (YTD), onde o K-Means com PCA registrou volatilidade ligeiramente superior (0,803% contra 0,785%). Esses dados sugerem que, embora o affinity propagation com PCA possa ser menos eficiente na gestão de risco, ele ainda se mantém competitivo em termos de estabilidade no curto prazo. No entanto, o *hierarchical* com t-SNE se destacou como a estratégia mais robusta em termos de controle de volatilidade ao longo do tempo.

3. **Resultados Correlação Média dos Pares de Ativos**

Tabela 15 – Benchmark Comparativo Correlação Média

Algoritmo	2018-TD	2020-TD	6m	12m	YTD
affinity_pca	21.00%	21.00%	17.83%	19.86%	18.03%
dbscan_tsne	20.24%	20.24%	17.85%	18.17%	16.87%
hierarchical_tsne	28.40%	30.17%	17.36%	18.20%	16.71%
kmeans_pca	20.23%	20.23%	17.97%	18.85%	17.42%

Ao analisar a correlação média dos pares de ativos, o método *dbscan* com t-SNE apresentou as correlações médias mais baixas em quase todos os horizontes de tempo, com destaque para os períodos mais recentes, como os últimos 12 meses (18,17%) e o acumulado do ano (YTD) (16,87%). O método K Means com PCA também apresentou um desempenho de correlação mais baixa, especialmente nos horizontes de longo prazo, superando o método DBScan com t-SNE. Isso sugere que os métodos *dbscan* com t-SNE e K Means com PCA, foram mais eficazes na construção de carteiras diversificadas, onde a menor correlação entre ativos pode reduzir o risco total da carteira. Dessa forma, este método pode ser preferido por investidores que buscam maior diversificação e resiliência em períodos de alta volatilidade.

Em contraste, o método *hierarchical clustering* com t-SNE destaca-se por apresentar uma correlação mais alta nos horizontes de longo prazo, especialmente nos períodos de 2018-TD (28,40%) e 2020-TD (30,17%). Embora a alta correlação possa indicar uma maior dependência entre os ativos, o que pode aumentar o risco em momentos de crise, essa característica pode ser benéfica em mercados ascendentes, onde ativos correlacionados tendem a se mover juntos, amplificando os retornos.

4. **Resultados VaR - Value at Risk**

Tabela 16 – Benchmark Comparativo VaR

Algoritmo	2018-TD	2020-TD	6m	12m	YTD
affinity_pca	-2.74%	-2.74%	-2.04%	-2.18%	-2.05%
dbscan_tsne	-2.69%	-2.69%	-2.11%	-2.17%	-2.08%
hierarchical_tsne	-3.42%	-3.53%	-1.86%	-1.88%	-1.82%
kmeans_pca	-2.72%	-2.72%	-2.07%	-2.14%	-2.02%

Ao avaliar o *Value at Risk* (VaR 99%) das carteiras geradas pelos diferentes algoritmos de clusterização, o método *hierarchical clustering* com t-SNE destaca-se por apresentar os menores valores de VaR nos horizontes de curto prazo, especificamente nos últimos 6 meses (-1,86%), 12 meses (-1,88%), e no acumulado do ano (YTD) (-1,82%).

Esses resultados indicam que, embora o *hierarchical* com t-SNE tenha um maior potencial de perda em períodos mais longos (como evidenciado pelos valores de 2018-TD e 2020-TD).

Por outro lado, os métodos *affinity propagation* com PCA e *K-Means* com PCA apresentaram valores de VaR muito próximos e mais baixos nos horizontes de longo prazo, com destaque para o *affinity* com PCA, que obteve o menor VaR em 2020-TD (-2,74%), sugerindo uma performance consistente em termos de controle de risco ao longo do tempo. No entanto, em termos de proteção contra perdas no curto prazo, o *hierarchical clustering* com t-SNE se mostrou superior.

4.3 Aplicação Prática - Plataforma

Após a definição do método de clusterização e desenvolvimento do algoritmo a Atomus Invest foi transformada em plataforma de Investimentos aberta ao público, tendo como base dois pilares fundamentais: A técnica de clusterização de dados e o algoritmo proprietário de avaliação fundamentalista das empresas, implementados por meio da linguagem de programação Python.

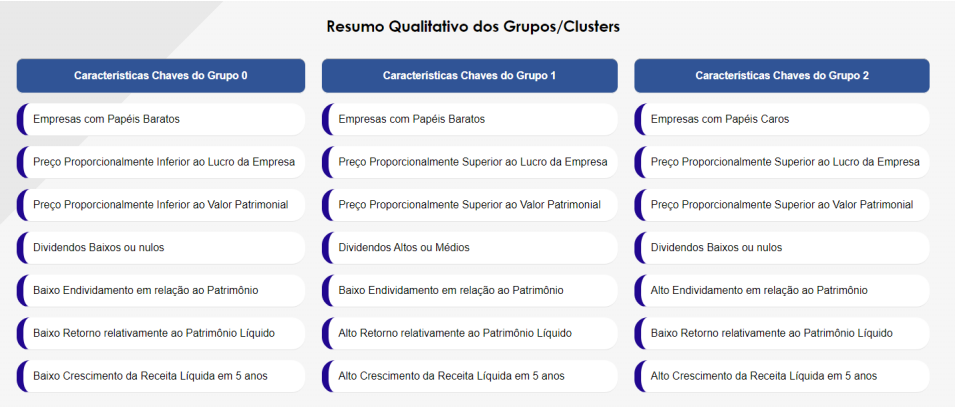
Dentro do pilar de clusterização, o processo de aprendizado de máquina não supervisionado (*Unsupervised Learning*) é aplicado por meio do algoritmo de K-Means, visando classificar as ações em 3 grandes grupos que compartilham características comuns, com base em seus indicadores fundamentalistas.

Na plataforma Atomus, a clusterização é aplicada de forma ampla, para todas as ações listadas na Bolsa brasileira, na página referente à "Análise Geral" (13), conforme ilustrado na Figura 20, bem como de forma segmentada para cada um dos 30 setores de indústria, dado que o contexto de atuação de cada empresa, muitas vezes, torna as métricas do setor mais ou menos normalizadas, possuindo valores ótimos distintos de acordo com cada dinâmica mercadológica.

Figura 20 – Gráfico de Clusterização de Análise Geral das Ações Listadas na B3

Fonte: Própria (2024)

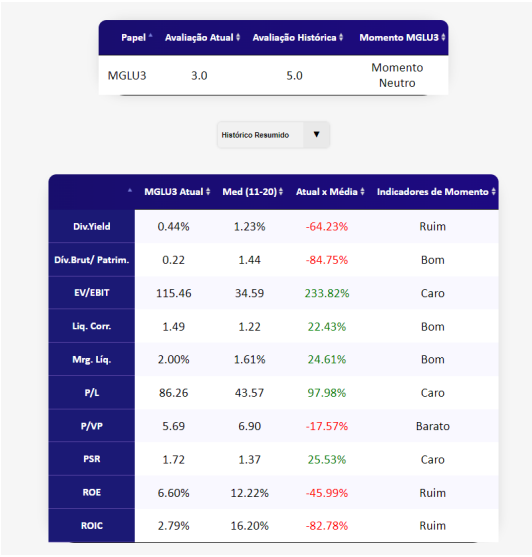
Figura 21 – Tabela de Caracterização dos Clusters da Página de Análise Geral



Fonte: Própria (2024)

Dentro do pilar de avaliação fundamentalista, o algoritmo consiste na interpretação em uma escala de 0 a 10 que resume a saúde dos indicadores da empresa. Estas avaliações são baseadas tanto no momento atual, quanto nos dados históricos de cada empresa e relação frente o segmento/mercado que estão inseridos, conforme ilustrado na Figura 22.

Figura 22 – Valuation Fundamentalista da Ação Magazine Luiza - MGLU3



Papel	Avaliação Atual	Avaliação Histórica	Momento MGLU3
MGLU3	3.0	5.0	Momento Neutro

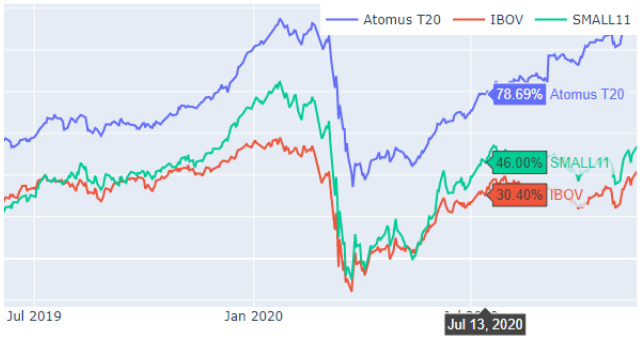
Histórico Resumido

	MGLU3 Atual	Med (11-20)	Atual x Média	Indicadores de Momento
Div.Yield	0.44%	1.23%	-64.23%	Ruim
Div.Brut/ Patrim.	0.22	1.44	-84.75%	Bom
EV/EBIT	115.46	34.59	233.82%	Caro
Liq. Corr.	1.49	1.22	22.43%	Bom
Mrg. Liq.	2.00%	1.61%	24.61%	Bom
P/L	86.26	43.57	97.98%	Caro
P/VP	5.69	6.90	-17.57%	Barato
PSR	1.72	1.37	25.53%	Caro
ROE	6.60%	12.22%	-45.99%	Ruim
ROIC	2.79%	16.20%	-82.78%	Ruim

Fonte: Própria (2024)

Através da união da análise fundamentalista clássica ao *Unsupervised Learning*, a plataforma Atomus Invest otimiza a capacidade de analisar ações em larga escala, baseando em seus indicadores, constituindo assim, uma ferramenta de análise que além de prover dados ao investidor, provê análises prontas.

Figura 23 – Rentabilidade da carteira sugerida pelo algoritmo versus Benchmark



Fonte: Própria (2024)

Comparando a performance da Atomus com benchmarks de mercado como IBOV11 e SMALL11, a plataforma superou os resultados destes ativos durante 2018-2021, com rentabilidade aproximadamente 2 vezes maior, como pode ser visto na Figura 23.

Sendo assim, através da plataforma Atomus Invest é esperado democratizar as estratégias de investimento exploradas neste trabalho, se consolidando como uma das principais ferramentas gratuitas de fornecimento e análise de dados do mercado de capitais. Para isto, é necessário aumentar sua base de usuários, gerando leads e incentivadores da plataforma, assim, atraindo investidores interessados em *Private Equity*. Posteriormente, todo capital será investido na aquisição de recursos técnicos necessários para expansão das operações da plataforma.

5 Conclusões e Trabalhos Futuros

5.1 Conclusões

Através do presente estudo e projeto desenvolvido, foi possível analisar como diferentes tipos de algoritmos de clusterização se comportam utilizando as métricas que avaliam separação, compactação, coesão e quantidade de clusters em cada modelo de forma técnica, além da avaliação de negócio das métricas tradicionais de risco do mercado acionário brasileiro, através de simulações e experimentos, em diferentes períodos para verificar a relação risco-retorno dos algoritmos.

Pode-se concluir que apesar das diferentes naturezas e métodos utilizados para o treinamento dos modelos, as métricas dispostas foram similares entre os algoritmos e períodos analisados. Desta forma, os indicadores técnicos utilizados no projeto, bem como os algoritmos de clusterização não devem ser aplicados de forma isolada, necessitando de uma camada de validação através dos métodos de *valuation* que possibilitam analisar o mercado acionário brasileiro de forma completa.

Os resultados obtidos nos benchmarks, demonstraram que o retorno acumulado do índice (IBOV), nos cinco anos analisados, foi inferior a todos os algoritmos, mesmo com a presença de desvios pontuais para períodos específicos.

5.2 Trabalhos Futuros

Para dar continuidade ao desenvolvimento do projeto, seria necessário explorar a otimização dos parâmetros dos modelos aplicados e as métricas selecionadas para avaliação técnica dos resultados.

Tendo em vista que a saída dos algoritmos são grupos de ativos construídos a partir de modelos não supervisionados, os quais não possuem como objetivo otimizar as métricas de retorno e risco, pode-se adotar o treinamento de algoritmos supervisionados para auxiliar na construção de cada cluster, contudo mantendo a metodologia inicial para mitigar possíveis vieses.

Outro aprofundamento possível, seria a inclusão de outros indicadores técnicos para avaliação dos algoritmos. Neste experimento, foram utilizados três indicadores dentre um vasto catálogo de métricas para avaliar a coesão de cada cluster. Da mesma forma, dentro das métricas de risco e retorno, existem testes e métricas adicionais que podem ser exploradas no *benchmark* dos algoritmos.

A inclusão de novas metodologias de redução dimensional e clusterização também

podem ser exploradas, métodos como *LDA (Linear Discriminant Analysis)*, *Autoencoders*, entre outros, no campo da clusterização temos outros como *Gaussian Mixture Models (GMM)* e *Spectral Clustering* que são exemplos de modelos com potencial para serem experimentados.

Referências

1 B3. **Total de investidor pessoa física cresce 43% no primeiro semestre, mostra estudo da B3**. 2021. B3. Disponível em: <https://www.b3.com.br/pt_br/noticias/porcentagem-de-investidores-pessoa-fisica-cresce-na-b3.htm>. Acesso em: 23 nov. 2021.

2 SIEGEL, J. **Stocks for the Long Run 5/E: The Definitive Guide to Financial Market Returns & Long-Term Investment Strategies**. 5th revised ed.. ed. New York: McGraw-Hill, 2014.

3 HARTIGAN, J. A. **Clustering Algorithms**. [S.l.]: John Wiley & Sons, Inc., 1975.

4 ESTER HANS-PETER KRIEGEL, J. S. M. **A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise**. 1996. Disponível em: <https://www.dbs.ifi.lmu.de/Publikationen/Papers/KDD-96.final.frame.pdf>. Acesso em: 23 nov. 2021.

5 YENIGüN, O. **The Mechanics of Affinity Propagation Clustering**. 2024. Plain English. Disponível em: <https://python.plainenglish.io/the-mechanics-of-affinity-propagation-clustering-eb199cc7a7c2>. Acesso em: 05 abr. 2024.

6 GULCAN, M. **Hierarchical Clustering with Python: Basic Concepts and Application**. 2024. Medium. Disponível em: <https://medium.com/@muratgulcan/hierarchical-clustering-with-python-basic-concepts-and-application-cd5f5dc95b1f>. Acesso em: 05 abr. 2024.

7 GRAHAM, B. **O Investidor Inteligente**. 1ª ed.. ed. New York: Harper Collins, 2016.

8 FUNDAMENTUS. **Fundamentus**. 2024. Fundamentus. Disponível em: <https://www.fundamentus.com.br/>. Acesso em: 05 abr. 2024.

9 DOCS, R. the. **Beautiful Soup Documentation**. 2024. Read the Docs. Disponível em: <https://beautiful-soup-4.readthedocs.io/en/latest/>. Acesso em: 05 abr. 2024.

10 AL., F. P. et. Scikit-learn: Machine learning in python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

11 KAVLAKOGLU, E. **Reducing Dimensionality with Principal Component Analysis**. 2024. IBM Developer. Disponível em: <https://developer.ibm.com/tutorials/awb-reducing-dimensionality-with-principal-component-analysis/>. Acesso em: 05 abr. 2024.

12 ERDEM, K. **t-SNE Clearly Explained**. 2024. Towards Data Science. Disponível em: <https://towardsdatascience.com/t-sne-clearly-explained-d84c537f53a>. Acesso em: 05 abr. 2024.

13 INVEST, A. **Atomus Invest Geral**. 2021. Atomus Invest. Disponível em: <https://atomusinvest.com.br/geral>. Acesso em: 23 nov. 2021.

14 FOO, D. **High Dimension Clustering w/ t-SNE & DBSCAN**. 2024. Towards Data Science. Disponível em: <https://towardsdatascience.com/high-dimension-clustering-w-t-sne-dbscan-dcec77e6a39b>. Acesso em: 05 abr. 2024.

15 B, S. **K-means Clustering Using Elbow Method**. 2024. Medium. Disponível em: <https://imswapnilb.medium.com/k-means-clustering-using-elbow-method-208b23c78150>. Acesso em: 05 abr. 2024.

16 DASGUPTA, S. **Understanding the Epsilon Parameter of DBSCAN Clustering Algorithm**. 2024. Medium. Disponível em: <https://medium.com/@saurabh.dasgupta1/understanding-the-epsilon-parameter-of-dbscan-clustering-algorithm-fe85669e0cae>. Acesso em: 05 abr. 2024.

17 INVEST, A. **Históricos MGLU3**. 2021. Atomus Invest. Disponível em: <https://atomusinvest.com.br/Historicos/MGLU3>. Acesso em: 23 nov. 2021.