Trabalho de Conclusão de Curso

Meta-Aprendizagem para seleção de técnicas de redução de dimensionalidade em problemas de Big Data

Daniel José da Silva
djs@ic.ufal.br

Orientador:

Bruno Almeida Pimentel

Maceió, de 2024

Daniel José da Silva

Meta-Aprendizagem para seleção de técnicas de redução de dimensionalidade em problemas de Big Data

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação do Instituto de Computação da Universidade Federal de Alagoas.

Orientador:

Bruno Almeida Pimentel

Maceió, de 2024

Catalogação na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Girlaine da Silva Santos – CRB-4 – 1127

S586m Silva, Daniel José da.
Meta-Aprendizagem para seleção de técnicas de redução de dimensionalidade em problemas de Big Dat / Daniel José da Silva. – 2025.
54 f.: il.

Orientador: Bruno Almeida Pimentel.
Monografia (Trabalho de Conclusão de Curso em Computação) - Universidade Federal de Alagoas, Instituto de Computação. Maceió, 2025.

Bibliografia: f. 49- 54.

1. Aprendizagem de máquina. 2. Meta aprendizagem. 3. Algoritmos. 4. Inteligência artificial. 5. Big data . I. Título.

CDU: 004.8

Agradecimentos

Primeiramente, agradeço a Deus por me permitir chegar até aqui, aos meus pais Claudevan e Rosenilda, aos meus avós maternos Cícera e José Antônio, ao meu irmão Tiago, à minha prima Talia, por todo o apoio ao longo dessa jornada e por acreditarem em mim, mesmo quando minhas conquistas pareciam difíceis. Este diploma é para vocês, e sou imensamente feliz por levar o primeiro diploma para nossa família. Agradeço também à minha esposa Gerllayne pela paciência e apoio durante este final de graduação, e ao meu orientador e professor Bruno Almeida Pimentel pelo privilégio de trabalharmos juntos ao longo desses anos. Agradeço aos meus colegas de curso, em especial a Priscila, Wallace, Gabriel e Ramon, por estarem comigo desde o início e torcerem pelas minhas conquistas. E aos meus amigos de vida, Ranilze, Alex, Jakeline e Felipe, pelo apoio constante.

"A única forma de chegar ao impossível é acreditar que é possível."
- Alice no País das Maravilhas

Resumo

Universidades e indústrias produzem uma enorme quantidade de dados, muitas vezes caracterizados por alta dimensionalidade, o que pode afetar negativamente o desempenho de algoritmos de Aprendizagem de Máquina. A redução de dimensionalidade se torna uma solução fundamental para simplificar esses dados sem perder informações importantes, permitindo uma análise mais eficiente. No entanto, a escolha manual do algoritmo de redução de dimensionalidade mais adequado para cada conjunto de dados é um processo complexo e demorado. Com o objetivo de automatizar essa seleção, este trabalho propõe o desenvolvimento de um meta-aprendiz que possa prever qual algoritmo de redução de dimensionalidade será mais eficiente para um determinado conjunto de dados. Este trabalho aborda a seleção automatizada de algoritmos de redução de dimensionalidade em cenários de Big Data, utilizando meta-aprendizagem para aprender padrões entre os conjuntos de dados e os algoritmos que produzem os melhores resultados. A proposta é testar diferentes técnicas de redução de dimensionalidade e, por meio da construção de rankings comparativos, verificar se o modelo de recomendação de algoritmos consegue prever corretamente o algoritmo mais adequado.

Palavras-chave: Meta aprendizagem, Algoritmos, Dados. Redução, Aprendizagem de máquina

Abstract

Universities and industries generate vast amounts of data, often characterized by high dimensionality, which can negatively impact the performance of Machine Learning algorithms. Dimensionality reduction becomes a crucial solution to simplify these data without losing important information, allowing for more efficient analysis. However, manually selecting the most suitable dimensionality reduction algorithm for each dataset is a complex and time-consuming process. To automate this selection, this study proposes the development of a meta-learner capable of predicting which dimensionality reduction algorithm will be most efficient for a given dataset. This work focuses on the automated selection of dimensionality reduction algorithms in Big Data scenarios, utilizing meta-learning to identify patterns between datasets and the algorithms that yield the best results. The approach involves testing different dimensionality reduction techniques and, through the construction of comparative rankings, verifying whether the algorithm recommendation model can accurately predict the most suitable algorithm.

Palavras-chave: Meta-learning, Algorithms, Data, Reduction, Machine Learning.

Conteúdo

1

Introdução

A inteligência artificial (IA) (McCarthy, 2007) é um termo amplamente reconhecido no campo da Ciência da Computação, tendo ganhado notoriedade desde os anos 50. De acordo com Alan Turing, o funcionamento de uma máquina tão inteligente que poderia se passar por um humano sem ser detectada como uma máquina, conhecido como o teste de Turing (Turing, 2009).

Ao longo dos tempos, vários cientistas buscam aprimorar algoritmos que possam realizar tarefas mais rápidas/eficientes do que os humanos, para isso, a máquina precisa aprender e entender padrões por meio de algum tipo de aprendizagem que, por sua vez, demanda uma grande quantidade de dados para que os algoritmos entendam e conheçam padrões, pois tais algoritmos são alimentados por esses dados.

Observa-se, com o avanço da internet e de seu acesso facilitado, que os algoritmos de IA estão sendo cada vez mais procurados e utilizados em aplicações do dia a dia, como, por exemplo, redes sociais, sistemas de banco, sistemas hospitalares e entre outras aplicações para diversos problemas (Fetzer and Fetzer, 1990). A IA tornou-se uma grande facilitadora para a humanidade e, com isso, tornou-se um grande campo da ciência que engloba subáreas como aprendizagem de máquina, aprendizagem profundo e visão computacional, entre outras.

A Aprendizagem de Máquina (Zhou, 2021) é um campo que se utiliza de técnicas computacionais para realizar tarefas como classificação, agrupamento e previsão a partir de dados (Mitchell, 1997). Os algoritmos de Aprendizagem de Máquina são usualmente empregados em soluções para problemas reais, seja usando regressão, classificação ou outras técnicas, existem diversos algoritmos de Aprendizagem de Máquina conhecidos, como: Árvore de Decisão, Floresta Aleatória, Máquina de Vetores de Suporte, Redes Neurais, K-Means. Essa variedade se origina principalmente pela especificidade que cada algoritmo trás para solucionar problemas. (Muhammad and Yan, 2015).

Com a grande quantidade de dados disponíveis, muitos pesquisadores recorrem a algoritmos na expectativa de que estes aprendam (Soofi and Awan, 2017a), através do conjunto de suposições embutidas, as características derivadas dos dados. Nesse sentido, a busca por um modelo

eficaz que se adeque aos dados e aprenda com eles(Novaković et al., 2017), aumentando seu desempenho ao longo do tempo, reveste-se de fundamental importância tanto para pesquisas quanto para a indústria.

A disponibilidade de dados tem impulsionado significativamente as pesquisas em Aprendizagem de Máquina e ciência de dados (Van Der Aalst and van der Aalst, 2016). No entanto, o aumento simultâneo no volume de dados frequentemente desafia a capacidade dos modelos de aprendizagem de máquina em extrair informações relevantes de maneira eficaz. Para abordar essa complexidade, técnicas de Redução de Dimensionalidade (Van Der Maaten et al., 2009a) emergem como ferramentas cruciais para lidar com esse dilema, tornando os dados mais gerenciáveis e facilitando a compreensão de dados de alta dimensão (Jia et al., 2022).

A capacidade de aprender com experiências anteriores por meio de dados e gerar conhecimento desempenha um papel crucial na melhoria do processo de seleção de modelos eficientes para tomada de decisão. O Meta-Aprendizagem (Vanschoren, 2019) possibilita que sistemas de Aprendizagem de Máquina adquiram conhecimento sobre o desempenho de diferentes algoritmos e técnicas em diversos conjuntos de dados. Esse conhecimento prévio capacita o sistema a selecionar automaticamente as abordagens mais adequadas para novos conjuntos de dados, simplificando o processo de análise e interpretação de grandes volumes de informações (Farrell, 1983).

Meta-Aprendizagem (Vanschoren, 2019), ou aprender a aprender, é a ciência de aprender sistematicamente observando o desempenho de diferentes abordagens de aprendizagem de máquina em uma ampla gama de tarefas de aprendizagem. A partir dessa experiência, ou metadados, o sistema aprende a lidar com novas tarefas de forma muito mais rápida do que seria possível sem essa base de conhecimento. Isso não apenas acelera e melhora dramaticamente o design de pipelines de aprendizagem de máquina ou arquiteturas neurais, mas também permite substituir algoritmos projetados manualmente por novas abordagens aprendidas de maneira baseada em dados (Vilalta and Drissi, 2002).

Neste campo fascinante e em constante evolução, o Meta-Aprendizagem apresenta uma visão inovadora para a construção de sistemas de aprendizagem de máquina mais eficientes e adaptáveis (Scheinker, 2021). Ele oferece uma metodologia robusta para a seleção de modelos, otimizando o processo de tomada de decisão e potencializando a capacidade de análise e interpretação de dados em diversas aplicações.

Este estudo apresenta uma investigação aprofundada sobre a seleção de algoritmos de redução de dimensionalidade, empregando o Meta-Aprendizagem (Pavel Brazdil, 2009). como uma abordagem automatizada para aprimorar a prática da ciência de dados. O objetivo é construir um sistema de Meta-Aprendizagem que ofereça a melhor estratégia para selecionar esses algoritmos (Giraud-Carrier et al., 2004). A proposta involves a combinaço de diversos algoritmos de redução de dimensionalidade, empregando técnicas de classificação tanto antes quanto depois da aplicação dos algoritmos de redução. Isso possibilita uma avaliação comparativa dos resultados antes e depois da redução, visando identificar melhorias no desempenho dos algoritmos.

1.1 Motivação

A redução de dimensionalidade (Van Der Maaten et al., 2009b) é crucial no contexto de Big Data, onde lidar com grandes conjuntos de dados complexos pode ser desafiador. Essa técnica visa simplificar a representação dos dados sem perder informações essenciais, facilitando a análise e o processamento computacional. Este estudo aborda a seleção automatizada de técnicas de redução de dimensionalidade, visando resolver desafios que podem acelerar a pesquisa científica e industrial. A complexidade dos dados de alta dimensionalidade impõe desafios aos algoritmos de aprendizagem de máquina.

Investigar a aplicação do Meta-Aprendizagem para selecionar as técnicas de redução mais adequadas oferece oportunidades para contribuir com soluções práticas e eficazes em problemas reais. Este trabalho contribui para o avanço do conhecimento em um campo em rápido desenvolvimento. A automação da seleção de técnicas de redução de dimensionalidade tem implicações práticas em diversos setores, como análises financeiras e diagnósticos médicos. Realizar um estudo sobre esse tema permite o aprimoramento de habilidades em áreas como análise de dados, aprendizagem de máquina, programação e interpretação de resultados.

1.2 Objetivo Geral

Este trabalho tem como objetivo criar um meta-aprendiz que busca prever qual algoritmo de redução de dimensionalidade é eficiente para determinado conjunto de dados. Para avaliar sua eficácia, serão criados dois rankings: um sem algoritmos de redução e outro com eles, a fim de verificar se o meta-aprendiz consegue prever de forma correta. Apesar da importância de reduzir a dimensão dos dados, há poucos estudos sobre a automatização da escolha desses algoritmos. Portanto, este trabalho propõe uma automação da análise dos algoritmos de redução de dimensionalidade.

1.3 Objetivo Específico

Desenvolver um meta-aprendiz que selecione automaticamente os algoritmos de redução de dimensionalidade mais apropriados para diferentes conjuntos de dados. Implementar e testar uma variedade de algoritmos de redução de dimensionalidade e técnicas de classificação para avaliar a eficácia do meta-aprendiz. A avaliação será feita através da comparação de rankings de desempenho dos algoritmos de redução de dimensionalidade antes e depois da sua aplicação, medindo o impacto das reduções.

A automatização do processo de seleção de algoritmos de redução de dimensionalidade visa simplificar a análise de dados em cenários de Big Data. Isso contribuirá para o avanço do conhecimento em meta-aprendizagem e redução de dimensionalidade, fornecendo uma base

para futuros estudos e aplicações.

1.4 Organização do documento

Após esse capítulo de instrodução desse trabalho de conclusão de curso, o mesmo foi organizadoda seguinte forma:

Capítulo 2 explica a fundamentação teórica, como o conceito de aprendizagem de máquina, aprendizagem supervisionada, a partir de técnicas usadas para descoberta do conhecimento. Será explicado o funcionamente de alguns algoritmos de redução de dimensionalidade. Além disso, as meta features usandas seram mecionada, por fim, os algoritmos usados para classificação e métricas de avaliação de algoritmos.

Capítulo 3 descreve a metodologia usada, serão explicados os motivos das escolhas da bases de dados, metafeatures e como foi feita a seleção das componentes dos algoritmos de redução de dimensionalidade. Como também, motivo de reduzir para ter uma melhor desempenho.

Capítulo 4 descreve os resultados e discussões, os dados coletados durante a pesquisa são apresentados de forma organizada, acompanhados de análises e interpretações. São discutidas as descobertas em relação aos objetivos da pesquisa, destacando-se as conclusões principais e como estas contribuem para a compreensão do tema em questão.

Capítulo 5 são retomados os principais pontos abordados na pesquisa e ressaltadas as contribuições para o conhecimento na área de estudo. Além disso, são discutidas possíveis limitações do estudo e sugestões para pesquisas futuras.

2

Fundamentação Teórica

Este capítulo tem como objetivo apresentar alguns dos conceitos essenciais para a compreensão deste trabalho, abordando tópicos fundamentais da Inteligência Artificial, Aprendizagem de Máquina e aspectos críticos da análise de dados. Serão exploradas diferentes abordagens de aprendizagem, desde métodos supervisionados até técnicas avançadas, como a meta-aprendizagem e a descoberta de conhecimento em grandes volumes de dados.

2.1 Inteligência Artificial

A Inteligência Artificial (IA) (McCarthy, 2007) é um campo de estudo da Ciência da Computação voltado para o desenvolvimento de sistemas e máquinas capazes de executar tarefas que, tradicionalmente, requerem inteligência humana. Entre essas capacidades, destacam-se a aprendizagem, o raciocínio, a tomada de decisões, o reconhecimento de padrões, o processamento de linguagem natural e a resolução de problemas complexos. Diferentemente da simulação da inteligência humana, a IA pode empregar métodos computacionais que não são necessariamente observáveis em seres humanos, possibilitando que as máquinas realizem atividades de forma eficaz e, em muitos casos, em larga escala.

2.2 Aprendizagem de Máquina

A Aprendizagem de Máquina (AM) (Zhou, 2021) é a área de estudo que possibilita a melhoria do desempenho de sistemas por meio da aprendizagem com a experiência, utilizando métodos computacionais. Dessa forma, os sistemas adquirem habilidades de aprendizagem sem a necessidade de uma programação explícita para cada tarefa, conforme postulado por A. L. Samuel (Samuel, 2000). Este domínio de investigação pertence ao campo da Inteligência Artificial (IA), que, por sua vez, faz parte do espectro mais amplo da Ciência da Computação (Mitchell and

Mitchell, 1997). Conforme ilustrado na figura acima, a Aprendizagem de Máquina (AM) representa uma vertente mais específica da IA, focada em capacitar máquinas a aprenderem a partir de dados disponíveis.

Em vez de serem programados explicitamente para realizar tarefas específicas, os sistemas de AM são treinados utilizando grandes conjuntos de dados. Nesses conjuntos, padrões e relações são identificados pelos algoritmos desenvolvidos (Heil et al., 2021).

Dessa forma, o conhecimento é adquirido por meio de representações matemáticas (Ragone et al., 2022) e é avaliado por modelos criados por esses algoritmos, permitindo que os computadores aprendam a partir dos dados e façam previsões ou tomem decisões sem a necessidade de programação explícita para cada tarefa específica.

A Aprendizagem de Máquina (AM) (Zhou, 2021) visa resolver uma variedade de problemas, que podem ser abordados usando algoritmos de classificação, clusterização, regressão, entre outros (Sarker, 2021). Estes problemas representam desafios distintos, cada um com suas características e métodos específicos de resolução.

Por meio de técnicas e algoritmos especializados, a AM busca não apenas identificar padrões e tendências nos dados, mas também fornecer soluções eficazes para uma ampla gama de aplicações práticas em diversos domínios.

A classificação (Sarker, 2021) é uma tarefa na qual o objetivo é atribuir categorias ou rótulos a instâncias de dados com base em características observadas. A resolução de problemas de classificação na AM requer a aplicação de uma variedade de algoritmos disponíveis. A escolha do algoritmo adequado depende da natureza específica do problema em questão, demandando uma análise cuidadosa das características dos dados e dos requisitos da aplicação.

Os algoritmos de AM são geralmente categorizados em três paradigmas principais: Aprendizagem Supervisionada, Aprendizagem não Supervisionada e Aprendizagem por Reforço (Li, 2017). A seguir, cada um desses paradigmas será detalhado em termos de princípios fundamentais, métodos de aplicação e exemplos de algoritmos relevantes.

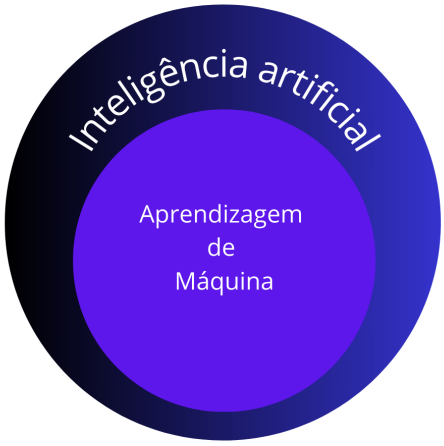

Figura 2.1: Relação entre Inteligência Artificial (IA) e Aprendizagem de Máquina (AM). Figura produzida pelo autor.

2.2.1 Aprendizagem supervisionada

Existem diversas estratégias de aprendizagem que podem ser utilizadas para desenvolver algoritmos de Aprendizagem de Máquina (AM). Atualmente, a mais estudada baseia-se no conceito de indução, segundo o qual é possível obter conclusões a partir de fatos ou observações particulares (de Souza, 2011). Sendo assim, a indução forma-se a partir de inferências lógicas, permitindo obter resultados genéricos sobre um conjunto específico de exemplos. A aprendizagem supervisionada é uma facilitadora para resolver problemas de classificação e regressão.

A aprendizagem indutiva pode ser dividida em aprendizagem supervisionada e não supervisionada, que será abordada na subseção seguinte. Na aprendizagem supervisionada, é fornecido ao algoritmo de aprendizagem um conjunto de exemplos de treinamento para os quais o rótulo da classe associada é conhecido.

A aprendizagem supervisionada (Cunningham et al., 2008) é essencial para extrair padrões e realizar previsões a partir de dados rotulados. Cada exemplo no conjunto de dados possui um rótulo conhecido, permitindo que o algoritmo aprenda a mapear os padrões nos dados de entrada para os rótulos correspondentes.

K-Vizinhos Mais Próximos

O K-Vizinhos Mais Próximos (*K-Nearest Neighbors* - KNN) (Abu Alfeilat et al., 2019) é um algoritmo de aprendizagem supervisionada utilizado principalmente para problemas de classificação e regressão. Ele pertence à classe de métodos de aprendizagem baseados em instâncias, onde o modelo não é explicitamente treinado, mas armazena todos os exemplos de treinamento. A classificação de novas instâncias é determinada pela maioria das classes dos

K exemplos de treinamento mais próximos de cada ponto: um ponto de consulta é atribuído à classe de dados que tem mais representantes dentro dos vizinhos mais próximos do ponto, medida através de uma função de distância, como a distância Euclidiana (Danielsson, 1980).

Fórmula da distância euclidiana entre duas instâncias $\mathbf{P} = (p_1, \ldots, p_2, \ldots, p_n)$ e $\mathbf{Q} = (q_1, \ldots, q_2, \ldots, 1_n)$ é definida como:

$$\sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \ldots + (p_n - q_n)^2} = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{2.1}$$

$\mathbf{p_i}$ e $\mathbf{q_i}$ para i = 1, ... , n são os n atributos que descrevem as instâncias $\mathbf{p_i}$ e $\mathbf{q_i}$, respectivamente.

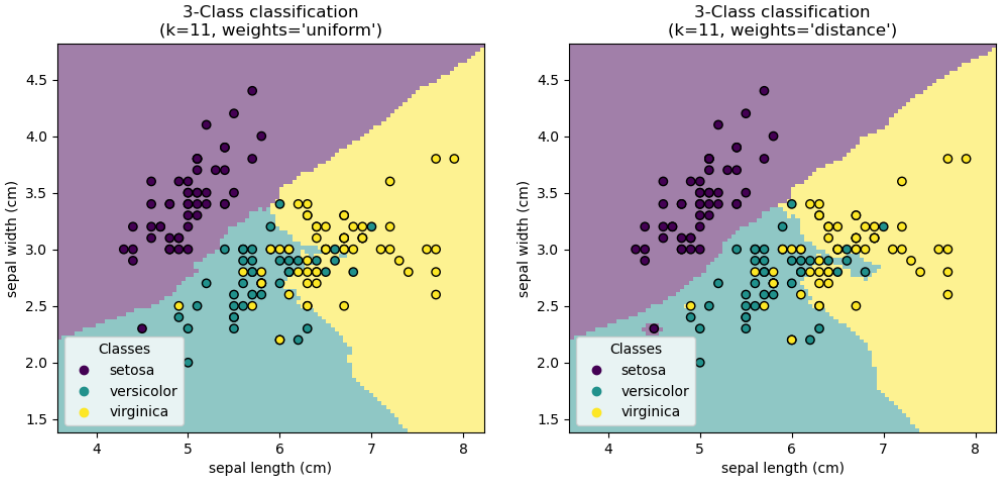

Figura 2.2: Classificação usando KNN. Figura do site scikit-learn

Floresta Aleatória (Random Forest)

O algoritmo Random Forest (Biau and Scornet, 2016), respresentado pelo campo da aprendizagem supervisionada, conhecido pela sua capacidade de lidar com uma veriedade de problemas de classificação e regressão. Como também, tem habilidade para mitigar problemas de overfitting (Ying, 2019), ao mesmo tempo que é capaz de oferecer alta precisão preditiva.

Em sua essência, uma Random Forest é composta por um conjunto de árvores de decisão individuais. Cada árvore é construída de forma independente, utilizando uma amostra aleatória dos dados de treinamento e um subconjunto aleatório de características. Esse processo de amostragem aleatória introduz diversidade entre as árvores, garantindo que cada uma contribua com diferentes perspectivas para o modelo final. Durante a fase de previsão, as previsões de todas as árvores na floresta são combinadas, geralmente por meio de uma média (para problemas de regressão) ou votação majoritária (para problemas de classificação).

A principal vantagem é sua capacidade de lidar com conjuntos de dados complexos e de grande escala, capturando eficientemente relações não lineares e interações entre variáveis.

2.2.2 Aprendizagem não supervisionada

Na aprendizagem não supervisionada (Barlow, 1989), ao contrário da aprendizagem supervisionada onde as informações possuem rótulos e a saída desejada é conhecida, os dados estão desprovidos de rótulos e não há uma saída esperada. Nesse contexto, os algoritmos, durante o processo de treinamento, não recebem resultados pré-determinados, sendo responsáveis por descobrir padrões e relações intrínsecas entre os dados de maneira autônoma (Chinnamgari, 2019).

O objetivo da aprendizagem não supervisionada é identificar regularidades nos dados com o intuito de agrupá-los ou organizá-los com base nas similaridades que apresentam entre si (Hastie et al., 2009). Dessa forma, o processo de aprendizagem busca explorar e revelar estruturas subjacentes nos dados, promovendo a descoberta de grupos ou a organização dos mesmos em função de suas características comuns.

a. Representação esquemática de um modelo de aprendizagem não supervisionada:

Em aprendizagem de máquina não supervisionada (Chinnamgari, 2019), os algoritmos são aplicados a dados não rotulados para descobrir padrões ou agrupamentos intrínsecos nos dados. O objetivo é identificar estruturas ocultas sem a necessidade de pré-rotulação. Um exemplo comum é o algoritmo de clustering, como o K-Means, onde dados são agrupados em clusters baseados em similaridades intrínsecas.

b. Representação esquemática de um modelo de aprendizagem supervisionada:

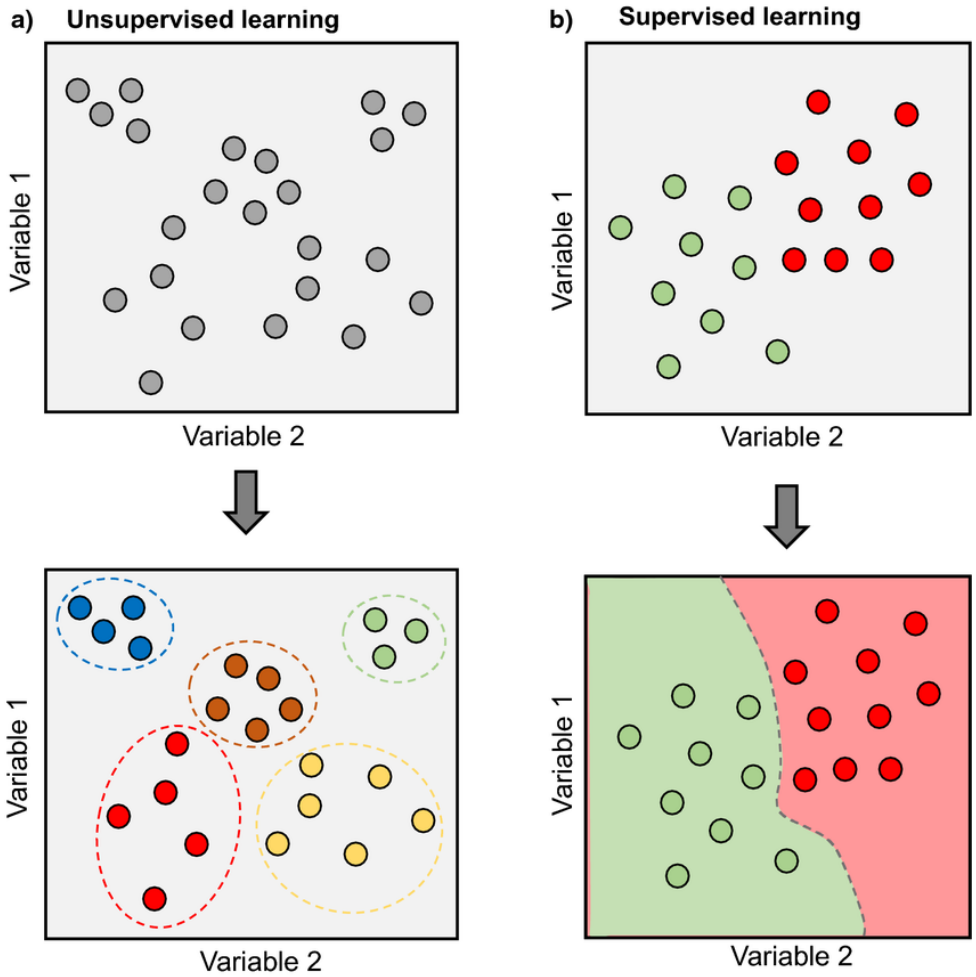

Figura 2.3: Aprendizagem de máquina supervisionada e não supervisionada. Scientific Figure on ResearchGate

Na aprendizagem de máquina supervisionada (Cunningham et al., 2008) o algoritmo é treinado usando um conjunto de dados rotulados, onde cada entrada de dados está associada a uma saída desejada. O objetivo é aprender uma função de mapeamento a partir dos dados de entrada para as saídas.

2.3 Processo de descoberta de conhecimento

Nas subseções anteriores, Aprendizagem de Máquina (AM) foi um dos pontos principais. Usar algoritmos de AM para extrair conhecimento a partir de dados e produzir bons resultados é um trabalho demorado, principalmente devido aos grandes volumes de dados existentes. Seria inviável que humanos analisassem esses dados manualmente e extrair informações significativas. Portanto, antes mesmo de aplicar algoritmos de AM, é crucial seguir conceitos bem consolidados na literatura, como os do processo de Descoberta de Conhecimento(KDD) (Frawley et al., 1992). O KDD envolve 5 etapas essenciais para realizar a extração de conhecimento,

demonstrado na figura abaixo (Mariscal et al., 2010).

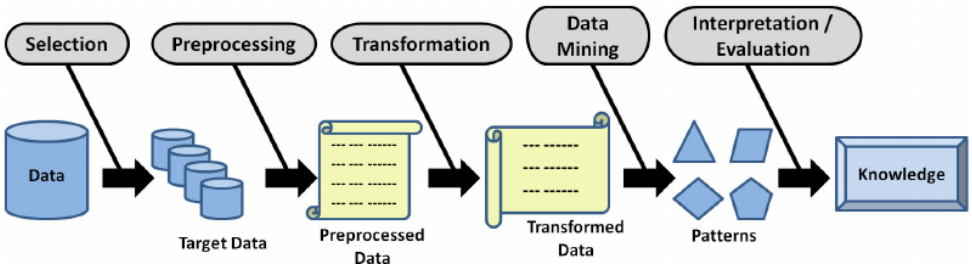

Figura 2.4: O processo de descoberta de conhecimento (KDD)

Ao escolher a base de dados(Mariscal et al., 2010), devem ser considerados vários critérios, como a relevância das variáveis para a tarefa de mineração e a disponibilidade dos dados. Em seguida, realiza-se o pré-processamento desses dados, que consiste em limpeza, remoção de valores ausentes, correção de erros e outras técnicas para transformar dados brutos em um formato adequado para análise. Esse pré-processamento pode ser feito em colunas, linhas ou em várias partes dos dados. Tal passo é responsável por enriquecer a base de dados e, além disso, deve garantir a qualidade e a consistência dos dados.

A transformação dos dados envolve modificar os dados para que se ajustem melhor aos algoritmos. Isso inclui identificar atributos que sejam realmente úteis para o modelo, levando em consideração o problema abordado. O formato atual dos dados pode dificultar o processamento pelo modelo, especialmente se contiver textos. Nesta etapa, os dados são adequados de forma que a máquina possa processá-los sem perda de informação. A transformação busca melhorar a eficiência e a eficácia dos métodos de mineração de dados que serão aplicados. Algumas técnicas responsáveis pela transformação dos dados incluem agregação, normalização e construção de novos atributos a partir dos dados originais.

A fase de mineração dos dados é a aplicação de métodos e algoritmos para extrair padrões e conhecimentos dos dados. Nessa etapa, os dados já estão pré-processados e transformados, e são divididos em conjuntos de treino e teste para o treinamento e validação do modelo. Esta é a etapa central do processo, onde técnicas como classificação, regressão e outros métodos de mineração são aplicados para descobrir padrões e relações nos dados.

Por fim, a interpretação e a avaliação dos dados são de fundamental importância para a obtenção de resultados válidos. Os resultados obtidos na etapa de mineração de dados devem ser avaliados quanto à qualidade dos padrões descobertos e à determinação de sua utilidade e relevância (Frawley et al., 1992). Isso é feito por meio de métricas de avaliação, como validação cruzada e acurácia do modelo. Após obter os resultados dos testes, com a acurácia do modelo, realiza-se a interpretação do conhecimento descoberto. Para isso, é importante ter conhecimento do problema.

2.4 Big Data

Big Data (Sagiroglu and Sinanc, 2013) refere-se ao grande volume de dados, sejam eles estruturados ou não, que empresas e indivíduos geram diariamente. Esses dados não abrangem apenas o volume, mas também a variedade, a velocidade, a veracidade e o valor com que são gerados e processados (Taurion, 2013).

As fontes desses dados são diversas, incluindo transações financeiras e mídias sociais. A análise de Big Data (Tsai et al., 2015) permite a descoberta de padrões e correlações ocultas, proporcionando insights valiosos que podem melhorar a tomada de decisões. Isso pode levar à otimização de operações, ao aumento da lucratividade e ao impulso da inovação.

A importância do Big Data reside na capacidade de transformar dados brutos em informações acionáveis (Taurion, 2013). Com o advento de tecnologias avançadas, as organizações podem agora analisar grandes volumes de dados, melhorar a eficiência operacional e oferecer experiências personalizadas aos clientes.

Uma das características do conceito de Big Data são os "Vs"(Younas, 2019) : volume, que se refere à quantidade massiva de dados gerados diariamente; variedade, que se refere aos diferentes tipos de dados disponíveis, como imagens e textos; velocidade, que se refere à rapidez com que os dados são gerados e precisam ser processados, facilitando a tomada de decisão; e valor, que se refere à utilidade dos dados para gerar insights valiosos.

No entanto, trabalhar com Big Data apresenta desafios significativos (Fasel and Meier, 2014), incluindo o armazenamento, a gestão e a proteção desses dados, bem como a necessidade de ferramentas e técnicas avançadas para uma análise eficiente.

2.5 Redução de dimensionalidade

Redução de dimensionalidade (Sorzano et al., 2014) é uma técnica utilizada em AM e estatística para simplificar conjuntos de dados, diminuindo o número de variáveis sob consideração. É útil para melhorar a performance dos algoritmos, reduzir o tempo de computação e ajudar na visualização dos dados.Algumas das técnicas mais comuns incluem:

Principal Component Analysis-PCA (Análise de Componentes Principais)

É uma técnica estatística usada para simplificar a complexidade de conjuntos de dados de alta dimensionalidade, transformando-os em um novo conjunto de variáveis não correlacionadas, conhecidas como componentes principais (Kurita, 2019).

De modo mais matemático utiliza uma transformação ortogonal (ortogonalização de vetores) para converter um conjunto de observações de variáveis possivelmente correlacionadas num conjunto de valores de variáveis linearmente não correlacionadas chamadas de componentes principais (Abdi and Williams, 2010).. Esse processo pode ser entendido como a construção de

uma nova base ortogonal no espaço vetorial das variáveis originais. Uma das formas de realizar essa ortogonalização é através do processo de Gram-Schmidt (Björck, 1994), que transforma vetores correlacionados em vetores mutuamente ortogonais, removendo as projeções lineares de cada vetor nas direções já ortogonalizadas.

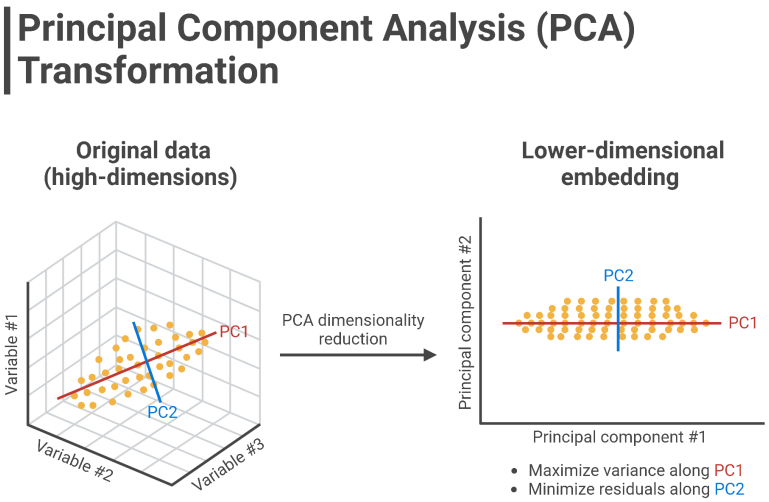

Figura 2.5: Principal Component Analysis (PCA) . Figura Mina Nashed

Matematicamente, a PCA é definida da seguinte maneira:

Dados de entrada: Considere uma matriz de dados X com n observações e p variáveis.

Centragem: Subtrai-se a média de cada variável para centralizar os dados na origem:

$$\tilde{X} = X - \bar{X} \tag{2.2}$$

onde $\bar{X}$ é o vetor de médias das variáveis.

Cálculo da matriz de covariância: A matriz de covariância é calculada como:

$$C = \frac{1}{n-1}\tilde{X}^T\tilde{X} \tag{2.3}$$

Autovalores e autovetores: Os autovalores λ e autovetores v da matriz de covariância C são determinados pela equação característica:

$$Cv = \lambda v \tag{2.4}$$

Ordenação: Os autovetores são ordenados em função de seus autovalores, do maior para o menor.

Projeção: Os dados são projetados nos componentes principais:

$$Z = \tilde{X}v_k \tag{2.5}$$

onde v_k são os autovetores correspondentes aos k maiores autovalores

Independent Component Analysis - ICA (Análise de Componentes Independentes)

A Análise de Componentes Independentes (ICA) é uma técnica estatística utilizada para separar um conjunto de sinais misturados em suas fontes independentes (Stone, 2004). Diferentemente da Análise de Componentes Principais (PCA), que se concentra em componentes não correlacionados, a ICA busca identificar componentes que são estatisticamente independentes entre si, permitindo a recuperação de informações de fontes que podem estar sobrepostas ou misturadas.

Matematicamente, a ICA é definida da seguinte maneira:

Considere um conjunto de sinais misturados $\mathbf{X}$, onde $\mathbf{X} \in \mathbb{R}^{n \times m}$ é uma matriz que contém n observações de m sinais misturados. O modelo de mistura linear pode ser expresso como:

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S} \tag{2.6}$$

onde:

- $\mathbf{X}$ é a matriz de sinais observados.
- $\mathbf{A}$ é a matriz de mistura (ou matriz de coeficientes) que relaciona os sinais independentes $\mathbf{S}$ aos sinais observados $\mathbf{X}$.
- $\mathbf{S} \in \mathbb{R}^{n \times m}$ é uma matriz cujas colunas representam os m sinais independentes que desejamos recuperar.

O objetivo da ICA é encontrar uma matriz $\mathbf{W}$ tal que, ao aplicar essa matriz aos sinais observados, obtemos uma estimativa dos sinais independentes:

$$\mathbf{S} \approx \mathbf{W} \cdot \mathbf{X} \tag{2.7}$$

onde $\mathbf{S}$ são os sinais recuperados.

Os componentes recuperados $\mathbf{S}$ são considerados independentes se a seguinte condição for satisfeita: para quaisquer duas variáveis s_i e s_j de $\mathbf{S}$, a seguinte relação de independência é verdadeira:

$$P(s_i, s_j) = P(s_i) \cdot P(s_j) \tag{2.8}$$

onde P denota a função de densidade de probabilidade.

Para encontrar a matriz $\mathbf{W}$, várias abordagens podem ser empregadas, como a maximização da não-gaussianidade dos componentes recuperados. A não-gaussianidade pode ser medida

usando o conceito de kurtose ou a entropia. Um método comum é o uso do algoritmo FastICA, que aplica a seguinte fórmula:

$$\mathbf{S} = g(\mathbf{W} \cdot \mathbf{X}) \tag{2.9}$$

onde $g(\cdot)$ é uma função de ativação não linear que ajuda a separar os sinais.

***T-Distribuição Stochastic Neighbor Embedding - t-SNE* (Incrustação Estocástica de Vizinhos Próximos com Distribuição)**

A T-Distribuição Stochastic Neighbor Embedding (t-SNE) é uma técnica de redução de dimensionalidade não linear que visa preservar as relações locais entre pontos de dados de alta dimensão em um espaço de menor dimensão, geralmente 2D ou 3D. O t-SNE é particularmente eficaz para a visualização de dados de alta dimensão, pois mantém a estrutura dos dados e facilita a identificação de padrões, clusters e agrupamentos que poderiam ser ocultos em representações de alta dimensão. (Belkina et al., 2019)

Matematicamente, o t-SNE é definida da seguinte maneira:

O t-SNE transforma a representação de dados de alta dimensão $\mathbf{X} \in \mathbb{R}^{n \times m}$ (onde n é o número de amostras e m é o número de dimensões) em uma nova representação $\mathbf{Y} \in \mathbb{R}^{n \times d}$, onde d é a nova dimensão (tipicamente 2 ou 3). O algoritmo t-SNE opera em duas etapas principais: a conversão de distâncias em probabilidades e a minimização de divergência.

Etapa 1: Cálculo das Probabilidades

Para cada ponto de dados x_i, a t-SNE calcula a similaridade de cada par de pontos x_i e x_j em alta dimensão usando uma distribuição Gaussiana, onde a probabilidade de x_j ser semelhante a x_i é dada por:

$$p_{j|i} = \frac{\exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|x_i - x_k\|^2}{2\sigma_i^2}\right)} \tag{2.10}$$

onde σ_i é a largura da distribuição Gaussiana para o ponto x_i, que pode ser determinada por uma abordagem baseada em vizinhos mais próximos.

A probabilidade simétrica p_{ij} entre os pontos x_i e x_j é então calculada como:

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n} \tag{2.11}$$

Etapa 2: Minimização da Divergência

Em seguida, o t-SNE modela a distribuição de similaridade nos dados de baixa dimensão y_i e y_j usando uma distribuição t de Student com 1 grau de liberdade (equivalente a uma distribuição Cauchy):

$$q_{ij} = \frac{(1 + \|y_i - y_j\|^2)^{-1}}{\sum_{k \neq l}(1 + \|y_i - y_k\|^2)^{-1}} \tag{2.12}$$

O objetivo do t-SNE é minimizar a divergência de Kullback-Leibler entre as distribuições de probabilidade de alta e baixa dimensão:

$$\mathrm{KL}(P\|Q) = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) \tag{2.13}$$

onde P e Q representam as distribuições de probabilidade nos espaços de alta e baixa dimensão, respectivamente.

Latent Dirichlet Allocation - LDA (Atribuição de Dirichlet Latente)

O Latent Dirichlet Allocation (LDA) é um modelo generativo probabilístico amplamente utilizado para a modelagem de tópicos em coleções de documentos. Ele assume que cada documento é uma mistura de tópicos, onde cada tópico é representado como uma distribuição sobre palavras. O LDA é especialmente eficaz para descobrir a estrutura latente em grandes conjuntos de dados textuais, permitindo a extração de temas subjacentes que caracterizam os documentos (Jelodar et al., 2019).

Matematicamente, o LDA é definida da seguinte maneira:
Considere um conjunto de documentos D contendo N palavras. O LDA é definido por três componentes principais: tópicos, palavras e documentos.

Componentes do Modelo

Tópicos: Cada tópico k é representado como uma distribuição sobre um vocabulário V:

$$\phi_k \sim \mathrm{Dirichlet}(\beta) \tag{2.14}$$

onde β é um vetor de parâmetros que controla a distribuição de palavras em cada tópico.

Documentos: Cada documento d é representado como uma mistura de tópicos. A distribuição de tópicos em um documento d é dada por:

$$\theta_d \sim \mathrm{Dirichlet}(\alpha) \tag{2.15}$$

onde α é um vetor de parâmetros que controla a distribuição de tópicos em cada documento.

Palavras: Para cada palavra $w_{d,n}$ no documento d, o processo de geração é descrito da seguinte forma:

1. Escolha um tópico $z_{d,n}$ de acordo com a distribuição de tópicos θ_d. 2. Escolha uma palavra $w_{d,n}$ a partir da distribuição de palavras do tópico $z_{d,n}$:

$$w_{d,n} \sim \mathrm{Multinomial}(\phi_{z_{d,n}}) \tag{2.16}$$

O objetivo do LDA é inferir as distribuições θ_d e ϕ_k a partir dos dados observados (as palavras nos documentos). Isso é feito usando o método de inferência, que pode ser realizado por técnicas como Variational Inference ou Gibbs Sampling.

A função de verossimilhança do modelo é dada por:

$$p(w,z,\theta,\phi|\alpha,\beta) = \prod_{d=1}^{D} \left(\prod_{n=1}^{N_d} p(w_{d,n}|z_{d,n},\phi) \cdot p(z_{d,n}|\theta_d) \right) \cdot p(\theta_d|\alpha) \cdot p(\phi_k|\beta) \tag{2.17}$$

onde:

- w são as palavras observadas, - z são os tópicos latentes, - θ é a distribuição de tópicos para cada documento, - ϕ é a distribuição de palavras para cada tópico.

Técnica	Tipo	Considera Classes	Linearidade	Principal Aplicação
PCA	Linear	Não	Linear	Redução de dimensionalidade e visualização
ICA	Linear	Não	Não-linear	Separação de sinais misturados
t-SNE	Não Linear	Não	Não-linear	Visualização de dados de alta dimensão
LDA	Linear	Sim	Linear	Classificação supervisionada

Tabela 2.1: Comparação das Técnicas de Redução de Dimensionalidade

O objetivo é reduz o número de variáveis (dimensões) em um conjunto de dados enquanto retém a maior parte da informação relevante. Facilita a visualização de dados em um espaço de menor dimensão, geralmente em 2D ou 3D. Com isso, melhora a eficiência computacional ao reduzir a carga de processamento e armazenamento. Elimina ou minimiza o ruído nos dados, focando nas variáveis que mais contribuem para a variabilidade dos dados.

2.6 Meta-aprendizagem

O Meta-aprendizagem (do inglês, Meta-Learning, abreviado por MtL) é um campo de estudo dentro da aprendizagem de máquina (AM) que se concentra em como otimizar o desempenho de algoritmos de aprendizagem (Pavel Brazdil, 2009). O objetivo é reduzir o custo computacional e economia de tempo, permitindo que modelos desenvolvam suas próprias estratégias de aprendizagem com base na análise passada. Para atingir esse aprimoramento, a Meta-aprendizagem utiliza o conceito de meta-dados, que consiste em uma coleção de meta-features e a performance de algoritmos avaliados.

Além disso, a Meta-aprendizagem, ou "aprender a aprender"(Vanschoren, 2019), é uma área de crescente interesse na inteligência artificial (IA), especialmente com o avanço das redes neurais profundas. O motivo é que redes neurais profundas são geralmente inicializadas com pesos aleatórios e possuem vieses indutivos muito fracos, o que gera a necessidade de aprender ou projetar vieses indutivos para melhorar a aprendizagem. Isso levou a um crescente interesse em abordagens de Meta-aprendizagem, que focam em como construir um modelo que aprende a aprender, em vez de apenas aprender, além do aprimoramento de sistemas de Aprendizagem de Máquina. Este conceito, que tem raízes nas ciências cognitivas e na psicologia (Wang, 2021), refere-se à capacidade de um sistema de melhorar seu desempenho em novas tarefas com base em experiências anteriores.

Diferentemente das abordagens tradicionais de Aprendizagem de Máquina, que são projetadas para resolver uma tarefa específica, a Meta-aprendizagem permite que um sistema utilize seu histórico de aprendizagem para se adaptar mais rapidamente a novos desafios, acelerando o processo de aprendizagem ao longo do tempo (Vanschoren, 2018).

Um dos princípios centrais da Meta-aprendizagem é a capacidade de adquirir viéses indutivos ou conhecimentos que facilitam a aprendizagem futura. Isso é alcançado por meio da construção de mecanismos que identificam padrões em processos de aprendizagem anteriores,

otimizando as decisões e a seleção de modelos mais adequados para diferentes tipos de problemas.

2.6.1 Arquitetura do Meta-aprendiz

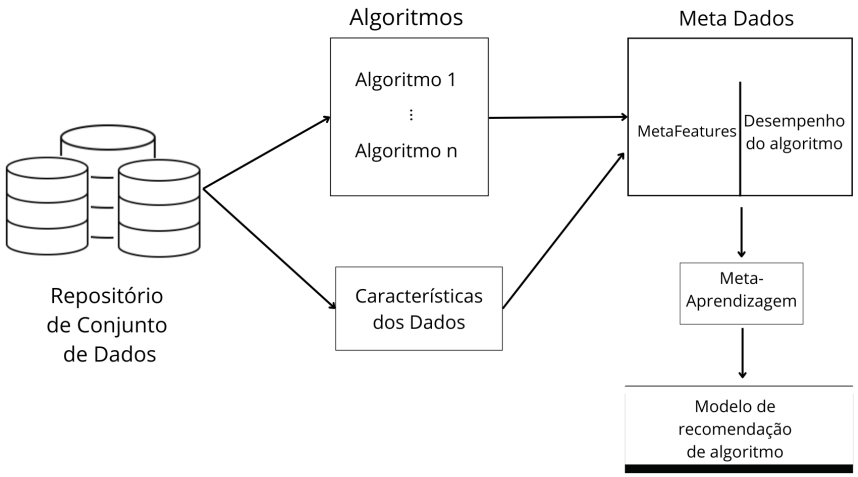

Figura 2.6: Arquitetura do meta-aprendiz, adaptada de (Pavel Brazdil, 2009)

A arquitetura de Meta-aprendizagem visa otimizar a seleção de algoritmos em tarefas de mineração de dados, utilizando informações extraídas de metadados. Esses meta-dados são compostos por características dos conjuntos de dados (meta-features) e pelo desempenho de diferentes algoritmos em problemas anteriores. A partir disso, o sistema é capaz de recomendar algoritmos mais adequados para novos datasets, reduzindo o tempo e os recursos gastos em experimentação, sem que haja perda significativa da qualidade dos resultados obtidos.

Esse processo utiliza Aprendizagem de Máquina para mapear as características dos dados ao desempenho relativo dos algoritmos, priorizando a recomendação de algoritmos que tendem a apresentar melhor performance. A arquitetura facilita a tomada de decisões ao guiar o usuário na escolha de algoritmos, evitando a necessidade de testar múltiplas opções manualmente. Isso torna a Meta-aprendizagem uma ferramenta eficaz para aplicações em que há uma grande variedade de algoritmos disponíveis e recursos computacionais limitados.

2.6.2 Meta-features

Meta-features são características extraídas de conjuntos de dados ou modelos de aprendizagem que fornecem informações valiosas sobre as propriedades e a estrutura dos dados (Rivolli et al., 2022). Elas são utilizadas em Meta-aprendizagem (do inglês, Meta-learning, abreviado por MtL) para permitir que modelos aprendam a escolher o algoritmo ou a abordagem mais adequada para um determinado conjunto de dados.

As meta-features ajudam a entender melhor as características dos dados, facilitando a seleção de algoritmos, ajustes de hiperparâmetros e a previsão de desempenho. Elas podem ser divididas em várias categorias, cada uma com suas particularidades:

Categoria	Exemplo	Descrição
Meta-feature Simples	Número de instâncias	Contagem total de observações no dataset.
	Número de features	Número total de variáveis explicativas.
	Número de classes	Quantidade de classes de saída em problemas de classificação.
Meta-feature Statistical	Média e desvio padrão dos atributos	Estatísticas descritivas para os atributos numéricos do dataset.
	Curtose e assimetria	Medidas de distribuição dos dados, indicando simetria e picos.
	Correlação entre atributos	Mede a relação linear entre os atributos, útil para identificar redundâncias.
Meta-feature Info-Theory	Entropia da classe	Medida da incerteza ou impureza da variável de classe.
	Entropia dos atributos	Avalia a incerteza ou dispersão dos atributos individuais.
	Informação mútua entre atributos e a classe	Quantifica a dependência entre atributos e a classe de saída.

Tabela 2.2: Exemplos de Meta-features

2.6.3 Meta-data

No contexto do Meta-aprendizagem, a meta-data desempenha um papel crucial na eficácia dos Meta-aprendizes (Castiello et al., 2005). Ela consiste na combinação de duas principais fontes de informação: as meta-features, que descrevem as características e propriedades dos conjuntos de dados utilizados, e o ranking de desempenho dos modelos testados. Essa junção fornece uma visão abrangente do problema que se busca resolver, permitindo que os meta-aprendizes façam previsões mais precisas e informadas.

A qualidade da meta-data é vital; ela deve refletir de maneira fiel as nuances do problema real que se pretende solucionar. Caso contrário, o meta-aprendiz pode falhar em generalizar e apresentar um desempenho insatisfatório em situações práticas. Além disso, o uso de bases de dados sintéticas para gerar meta-data é desencorajado, pois essas bases frequentemente não capturam adequadamente as complexidades dos dados reais, resultando em meta-aprendizes que não são adequados para serem aplicados em problemas do mundo real. Assim, garantir a relevância e a qualidade da meta-data (Vanschoren, 2018) é fundamental para o sucesso dos meta-aprendizes.

2.6.4 Meta-Aprendiz

O Meta-aprendiz é um sistema ou algoritmo dentro do campo da meta-aprendizagem que se caracteriza pela capacidade de "aprender a aprender". Utilizando o conhecimento adquirido a partir de tarefas anteriores, o meta-aprendiz otimiza seu desempenho em novas tarefas de Aprendizagem de Máquina. Ao analisar Meta-dados, como Meta-features e o histórico de desempenho de algoritmos em diferentes problemas, o sistema ajusta automaticamente a seleção de algoritmos e estratégias, visando melhorar a eficiência do processo de aprendizagem.

Esse mecanismo permite ao Meta-aprendiz identificar padrões recorrentes em experiências anteriores, possibilitando uma adaptação rápida e eficaz a novos desafios, sem a necessidade de extensa experimentação manual. Assim, o uso de Meta-aprendizes resulta em uma economia significativa de tempo e recursos computacionais, tornando-os ferramentas valiosas para contextos onde há uma grande diversidade de algoritmos e conjuntos de dados a serem explorados.

2.7 Métricas de avaliação

Em Aprendizagem de Máquina, as métricas de avaliação (Japkowicz, 2013) são fundamentais para medir a performance dos modelos. Elas são essenciais para determinar a eficácia de um modelo em realizar previsões precisas. Essas métricas variam conforme o tipo de problema e são geralmente classificadas em dois grandes grupos: métricas para classificação e métricas para regressão. As métricas para classificação (Wang et al., 2020) avaliam o desempenho do algoritmo na identificação correta das classes em um conjunto de dados. E as métricas para regressão (Tatachar, 2021) avaliam o quão próximas as previsões do modelo estão dos valores reais. Neste trabalho, apenas as métricas de classificação serão abordadas devido à sua aplicabilidade no processo do experimento.

2.7.1 Metricas de Classificação

Existem diversas métricas (Japkowicz, 2013) que podem ser utilizadas para avaliar classifiçação do modelo, cada uma com suas próprias características e aplicabilidades. Abaixo estão algumas das métricas mais comuns:

Acurácia (Accuracy)

A acurácia é a proporção de previsões corretas em relação ao total de previsões realizadas. É calculada como:

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.18}$$

onde:

- TP (True Positive) são os verdadeiros positivos.
- TN (True Negative) são os verdadeiros negativos.
- FP (False Positive) são os falsos positivos.
- FN (False Negative) são os falsos negativos.

Precisão (Precision)

A precisão mede a proporção de verdadeiros positivos entre as previsões positivas realizadas pelo modelo. É calculada como:

$$\text{Precisão} = \frac{TP}{TP+FP} \tag{2.19}$$

Revocação (Recall) ou Sensibilidade (Sensitivity)

A revocação, ou sensibilidade, mede a proporção de verdadeiros positivos que foram corretamente identificados pelo modelo em relação ao total de positivos reais. É calculada como:

$$\text{Recall} = \frac{TP}{TP+FN} \tag{2.20}$$

F1-Score

O F1-Score é a média harmônica entre precisão e revocação. Ele oferece um equilíbrio entre essas duas métricas, especialmente útil quando há um desequilíbrio entre as classes. É calculado como:

$$\text{F1} = 2 \times \frac{\text{Precisão} \times \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{2.21}$$

AUC-ROC (Area Under the Receiver Operating Characteristic Curve)

A AUC-ROC é uma métrica que avalia a capacidade do modelo em distinguir entre as classes. A curva ROC é um gráfico que mostra a taxa de verdadeiros positivos (sensibilidade) contra a taxa de falsos positivos em diferentes limiares de classificação. A área sob essa curva (AUC) indica a capacidade do modelo de separar as classes. Um modelo perfeito tem AUC igual a 1, enquanto um modelo aleatório tem AUC igual a 0.5.

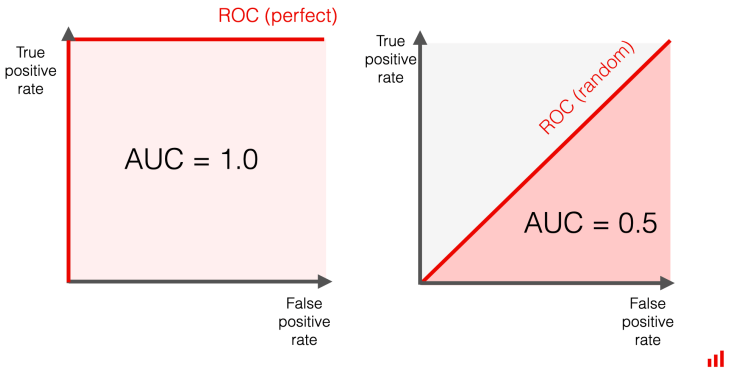

Figura 2.7: ROC AUC. Figura evidentlyai

Matriz de Confusão

A matriz de confusão é uma tabela que resume o desempenho do modelo, mostrando a quantidade de previsões corretas e incorretas por classe. Ela é especialmente útil para identificar erros específicos e ajustar o modelo adequadamente.

	Predito Positivo	**Predito Negativo**
Verdadeiro Positivo (TP)	50	10
Verdadeiro Negativo (TN)	5	35

Tabela 2.3: Matriz de Confusão

Cada célula da matriz representa a contagem de ocorrências para cada combinação de classe verdadeira e classe prevista.

3

Metodologia

O presente trabalho de conclusão de curso tem como objetivo explorar a aplicação de técnicas de seleção de algoritmos para aprimorar o processo de recomendação de algoritmos de aprendizagem de máquina, com ênfase na automação da escolha de algoritmos de redução de dimensionalidade. Neste contexto, será desenvolvido um sistema de recomendação de algoritmos de redução de dimensionalidade utilizando meta-aprendizagem. Ao final, um algoritmo de aprendizagem de máquina será aplicado para induzir um modelo meta-aprendiz capaz de predizer o ranking de algoritmos. Esse meta-aprendiz terá como função prever quais algoritmos de redução são mais adequados para diferentes conjuntos de dados, automatizando essa seleção. A eficácia da abordagem será avaliada por meio da criação de dois rankings de desempenho: um sem a aplicação das metafeatures e outro com elas, comparando os resultados para verificar se o meta-aprendiz faz previsões corretas.

3.1 Base de dados

Os dados utilizados neste trabalho foram obtidos a partir da plataforma OpenML, acessada via sua API. O OpenML é uma plataforma que disponibiliza diversos conjuntos de dados para a pesquisa em ciência de dados e aprendizagem de máquina. Por meio da API, foi possível selecionar e verificar os conjuntos de dados mais adequados para os objetivos do estudo.

O conjunto de dados obtido da OpenML consiste em 5.760 datasets, conforme ilustrado na figura abaixo.

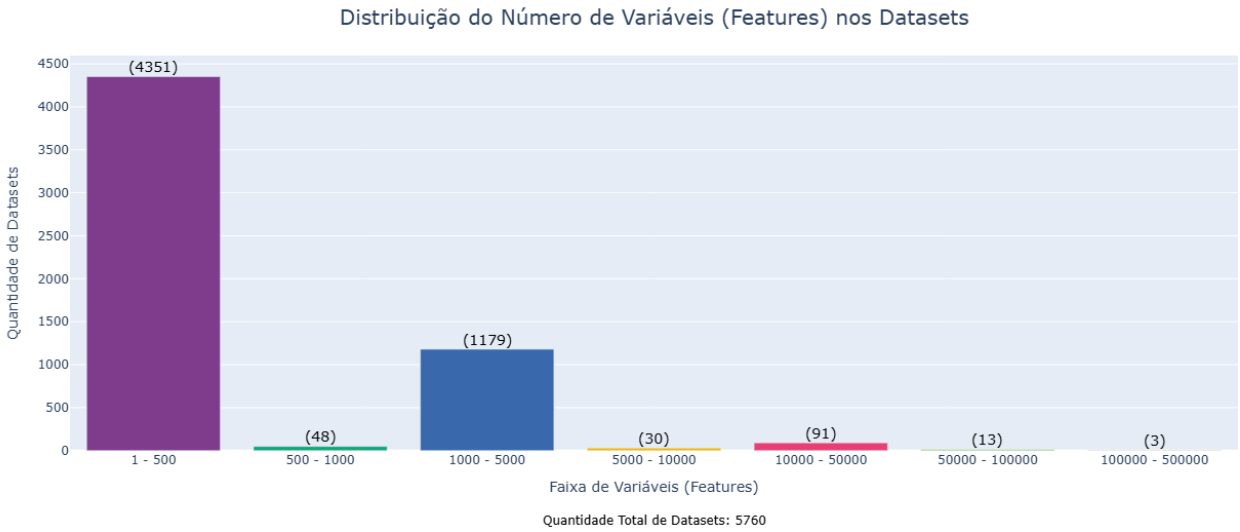

Figura 3.1: Distribuição do Número de Variáveis (NumberOfFeatures) nos Datasets. Figura produzida pelo autor

Após essa análise, foram selecionados 66 conjuntos. Para chegar a esse resultado, realizamos uma filtragem para incluir os conjuntos de dados que contêm entre 1.000 e 30.0000 variáveis (features) e que não possuem valores faltantes. Optamos por conjuntos de dados com um elevado número de variáveis, uma vez que este estudo tem como foco a análise de problemas relacionados a Big data, onde a alta dimensionalidade é um fator relevante. Ademais, evitamos o uso do formato Sparse_ARFF, devido à predominância de valores nulos. Além disso, observamos a distribuição do número de variáveis nos conjuntos de dados.

ID	Nome	ID	Nome	ID	Nome
1039	hiva_agnostic	1127	AP_Breast_Omentum	1149	AP_Ovary_Kidney
1084	BurkittLymphoma	1128	OVA_Breast	1150	AP_Breast_Lung
1104	leukemia	1129	AP_Uterus_Kidney	1151	AP_Endometrium_Omentum
1107	tumors_C	1130	OVA_Lung	1152	AP_Prostate_Ovary
1122	AP_Breast_Prostate	1131	AP_Prostate_Uterus	1153	AP_Colon_Ovary
1123	AP_Endometrium_Breast	1132	AP_Omentum_Lung	1154	AP_Endometrium_Lung
1124	AP_Omentum_Uterus	1133	AP_Endometrium_Colon	1155	AP_Prostate_Lung
1125	AP_Omentum_Prostate	1134	OVA_Kidney	1156	AP_Omentum_Ovary
1126	AP_Colon_Lung	1135	AP_Colon_Prostate	1157	AP_Endometrium_Kidney
1136	AP_Lung_Uterus	1138	OVA_Uterus	1158	AP_Breast_Kidney
1137	AP_Colon_Kidney	1139	OVA_Omentum	1159	AP_Endometrium_Ovary
1140	AP_Ovary_Lung	1141	AP_Endometrium_Prostate	1160	AP_Colon_Uterus
1142	OVA_Endometrium	1143	AP_Colon_Omentum	1161	OVA_Colon
1144	AP_Prostate_Kidney	1145	AP_Breast_Colon	1162	AP_Ovary_Uterus
1146	OVA_Prostate	1147	AP_Omentum_Kidney	1163	AP_Lung_Kidney
1148	AP_Breast_Uterus	1154	AP_Endometrium_Lung	1164	AP_Endometrium_Uterus
1165	AP_Breast_Ovary	1166	OVA_Ovary	1233	eating
1457	amazon-commerce-reviews	1458	arcene	1514	micro-mass
1515	micro-mass	4134	Bioresponse	40926	CIFAR_10_small
41084	UMIST_Faces_Cropped	41103	STL-10	41157	arcene
41159	guillermo	41161	riccardo	41163	dilbert
41165	robert	42140	SVHN_small	42766	kits-subset
42809	kits				

Tabela 3.1: Tabela de IDs e Nomes dos datasets

did	NumberOfNumericFeatures	NumberOfInstances	NumberOfClasses
1039	1617	4229	2
1084	22283	220	3
1104	7129	72	2
1107	7129	60	2
1122	10935	413	2
1123	10935	405	2
1124	10935	201	2
1125	10935	146	2
1126	10935	412	2
1127	10935	421	2
1128	10935	1545	2
1129	10935	384	2
1130	10935	1545	2
1131	10935	193	2
1132	10935	203	2
1133	10935	347	2
1134	10935	1545	2
1135	10935	355	2
1136	10935	250	2
1137	10935	546	2
1138	10935	1545	2
1139	10935	1545	2
1140	10936	324	2
1141	10935	130	2
1142	10935	1545	2
1143	10935	363	2
1144	10935	329	2
1145	10935	630	2
1146	10936	1545	2
1147	10935	337	2
1148	10936	468	2
1149	10935	458	2
1150	10935	470	2
1151	10935	138	2
1152	10935	267	2
1153	10935	484	2
1154	10935	187	2
1155	10935	195	2
1156	10935	275	2
1157	10935	321	2
1158	10935	604	2
1159	10935	259	2
1160	10935	410	2
1161	10935	1545	2
1162	10935	322	2
1163	10935	386	2
1164	10935	185	2
1165	10935	542	2
1166	10935	1545	2
1233	6373	945	7
1457	10000	1500	50
1458	10000	200	2
1514	1300	360	10
1515	1300	571	20
4134	1776	3751	2
40926	3072	20000	10
41084	10304	575	20
41103	27648	13000	10
41157	10000	100	2
41159	4296	20000	2
41161	4296	20000	2
41163	2000	10000	5
41165	7200	10000	10
42140	3072	9927	10
42766	27648	100	2
42809	27648	1000	2

Tabela 3.2: Tabela de IDs, número de atributos numéricos, instâncias e classes dos datasets

3.2 Pré-Processamento

Neste trabalho, a abordagem de pré-processamento dos dados foi mantida de forma reduzida, uma vez que o foco principal não foi o ajuste minucioso dos dados, mas sim a análise de modelos de aprendizagem de máquina em conjuntos de dados de grande escala. A justificativa para essa escolha reside na natureza dos conjuntos de dados utilizados, que possuem mais de 1000 variáveis. O tratamento exaustivo de tais dados, considerando a grande quantidade de variáveis e a diversidade dos conjuntos, seria extremamente custoso em termos computacionais, além de envolver um alto risco de perda de informações importantes.

Embora fosse possível que um pré-processamento mais cuidadoso pudesse resultar em um desempenho aprimorado dos modelos (García et al., 2016) a opção por uma abordagem mais simplificada visou priorizar a viabilidade computacional. O custo de tempo e os recursos necessários para realizar ajustes detalhados seriam excessivos, especialmente considerando o volume de dados. Além disso, como o objetivo deste estudo é lidar com grandes volumes de dados, característicos de cenários de Big Data, procurou-se minimizar intervenções que pudessem modificar excessivamente a natureza dos dados originais. Isso ajuda a preservar as características originais dos conjuntos de dados, o que é desejável quando se busca obter resultados que reflitam de maneira fiel a complexidade dos dados, sem introduzir alterações que possam afetar sua representatividade.

No entanto, foi necessária a normalização (Ali et al., 2014) de alguns conjuntos de dados, uma vez que certos modelos de aprendizagem de máquina requerem dados em escalas semelhantes para o seu bom desempenho. Isso é particularmente importante em modelos baseados em distâncias, como KNN, que podem ser sensíveis a valores desproporcionais em diferentes escalas. Para tal, foi aplicada a normalização utilizando o StandardScaler (Raju et al., 2020), que realiza a transformação dos dados para que tenham média zero e desvio padrão unitário.

Essa normalização é crucial, especialmente em datasets com variáveis cujos valores estão em diferentes ordens de magnitude, pois ajuda a evitar que características com maior escala dominem o processo de aprendizagem, promovendo uma convergência mais eficiente e uma performance mais robusta dos modelos de Aprendizagem de Máquina.

3.3 Meta-Features

O presente trabalho tem como objetivo explorar o uso de meta-features (Pinto et al., 2016) no contexto de Meta-Aprendizagem, buscando desenvolver um sistema de recomendação de algoritmos de aprendizagem de máquina que possa auxiliar na seleção e parametrização de modelos mais adequados a diferentes tipos de conjuntos de dados. A Meta-Aprendizagem é o estudo que visa usar o conhecimento prévio (meta-conhecimento) para otimizar processos de aprendizagem, e as meta-features desempenham um papel central ao capturar características dos dados que são relevantes para entender seu comportamento em relação a diferentes algoritmos.

As meta-features, ou metacaracterísticas, são características descritivas extraídas dos dados e usadas para compreender melhor as propriedades de um conjunto de dados e seu relacionamento com algoritmos de aprendizagem. Essas características são essenciais para identificar padrões e peculiaridades dos dados, facilitando a criação de modelos que podem prever qual algoritmo terá o melhor desempenho para determinado problema.

Nesse trabalho foi utlizados três categorias principais de meta-features:

Categoria	**Descrição**	**Detalhes**
General	Informações básicas sobre o dataset	Inclui o número de instâncias, atributos e classes. Ajuda a definir um perfil geral do conjunto de dados.
Statistical	Medidas estatísticas	Contém métricas como média, variância e assimetria, que descrevem as propriedades numéricas da distribuição dos dados e auxiliam na compreensão da complexidade do dataset.
Info-Theory	Medidas de teoria da informação	Agrupa métricas úteis para atributos discretos (categóricos), como a entropia, avaliando o relacionamento dos atributos com as classes.

Tabela 3.3: Características do Dataset por Categoria

Essas três categorias fornecem uma descrição robusta do conjunto de dados, o que contribui para um entendimento mais aprofundado de suas peculiaridades e desafios.

O trabalho também utiliza meta-dados, que consistem na combinação das meta-features com informações sobre o desempenho de diferentes algoritmos. Esses meta-dados ajudam a construir um sistema de recomendação (Rivolli et al., 1808) que não apenas avalia as características dos dados, mas também indica quais algoritmos são mais adequados para diferentes tipos de problemas.

Grupo	Nome	Descrição
General	attr_to_inst	Calcular a razão entre o número de atributos e o número de instâncias.
General	cat_to_num	Calcular a razão entre o número de feições categóricas e numéricas.
General	freq_class	Calcular a frequência relativa de cada classe distinta.
General	inst_to_attr	Calcular a razão entre o número de instâncias e atributos.
General	nr_attr	Calcular o número total de atributos.
General	nr_bin	Calcular o número de atributos binários.
General	nr_cat	Calcular o número de atributos categóricos.
General	nr_class	Calcular o número de classes distintas.
General	nr_inst	Calcular o número de instâncias no conjunto de dados.
General	nr_num	Calcular o número de recursos numéricos.
Statistical	can_cor	Calcular correlações canônicas dos dados.
Statistical	cor	Calcular o valor absoluto da correlação de pares de colunas de conjuntos de dados distintos.
Statistical	cov	Calcular o valor absoluto da covariância de pares de atributos de conjuntos de dados distintos.
Statistical	eigenvalues	Calcular os autovalores da matriz de covariância do conjunto de dados.
Statistical	g_mean	Calcular a média geométrica de cada atributo.
Statistical	gravity	Calcular a distância entre o centro de massa das classes minoritária e majoritária.
Statistical	h_mean	Calcular a média harmônica de cada atributo.
Statistical	iq_range	Calcular o intervalo interquartílico (IQR) de cada atributo.
Statistical	kurtosis	Calcular a curtose de cada atributo.
Statistical	lh_trace	Calcular o rastreamento de Lawley-Hotelling.
Statistical	mad	Calcular o Desvio Absoluto Mediano (MAD) ajustado por um fator.
Statistical	max	Calcular o valor máximo de cada atributo.
Statistical	mean	Calcular o valor médio de cada atributo.
Statistical	median	Calcular o valor mediano de cada atributo.
Statistical	min	Calcular o valor mínimo de cada atributo.
Statistical	nr_cor_attr	Calcular o número de pares de atributos altamente correlacionados distintos.
Statistical	nr_disc	Calcular o número de correlação canônica entre cada atributo e classe.
Statistical	nr_norm	Calcular o número de atributos normalmente distribuídos com base em um método.
Statistical	nr_outliers	Calcular o número de atributos com pelo menos um valor discrepante.
Statistical	p_trace	Calcular o traço de Pillai.
Statistical	range	Calcular o intervalo (máx - min) de cada atributo.
Statistical	roy_root	Calcular a maior raiz de Roy.
Statistical	sd	Calcular o desvio padrão de cada atributo.
Statistical	sd_ratio	Calcular um teste estatístico para homogeneidade de covariâncias.
Statistical	skewness	Calcular a assimetria para cada atributo.
Statistical	sparsity	Calcular a métrica de esparsidade (possivelmente normalizada) para cada atributo.
Statistical	t_mean	Calcular a média truncada de cada atributo.
Statistical	var	Calcular a variância de cada atributo.
Statistical	w_lambda	Calcular o valor Lambda de Wilks'.
Info-theory	attr_conc	Calcular o coeficiente de concentração de cada par de atributos distintos.
Info-theory	attr_ent	Calcular a entropia de Shannon para cada atributo alvo.
Info-theory	class_conc	Calcular o coeficiente de concentração entre cada atributo e classe.
Info-theory	class_ent	Calcular a entropia de Shannon do atributo alvo.
Info-theory	eq_num_attr	Calcular o número de atributos equivalentes para uma tarefa preditiva.
Info-theory	joint_ent	Calcular a entropia conjunta entre cada atributo e a classe.
Info-theory	mut_inf	Calcular as informações mútua entre cada atributo e destino.
Info-theory	ns_ratio	Calcular o ruído dos atributos.

Tabela 3.4: Meta-feature utilizadas

3.4 Métodos de Redução de dimensionalidade

No presente trabalho, exploramos diversas técnicas de redução de dimensionalidade (Sorzano et al., 2014), com o objetivo de desenvolver um Meta-Aprendiz capaz de selecionar automaticamente o algoritmo mais adequado para diferentes conjuntos de dados. A escolha criteriosa de técnicas de redução de dimensionalidade é fundamental para otimizar a performance dos algoritmos de aprendizagem de máquina, reduzir o tempo de processamento e facilitar a visualização de dados de alta dimensão. Em contextos de Big Data, a redução dimensional não só aprimora a eficiência computacional, como também possibilita o tratamento de dados de grande escala.

Para maximizar a diversidade de opções de redução dimensional, selecionamos 10 algoritmos que cobrem diferentes abordagens e características. A Tabela 3.5 apresenta esses algoritmos e uma breve justificativa para a escolha de cada um, considerando a diversidade de conjuntos de dados e a adequação de cada técnica a diferentes tipos de estrutura e complexidade dos dados.

Algoritmo	**Descrição**
Principal Component Analysis (PCA)	Reduz a dimensionalidade ao identificar as direções principais (componentes principais) da variabilidade nos dados. É eficaz para dados lineares.
Kernel PCA (kPCA)	Expande o PCA convencional para dados não lineares ao aplicar um kernel. Ideal para dados com estruturas complexas.
Latent Dirichlet Allocation (LDA)	Reduz a dimensionalidade em dados textuais, útil para identificar tópicos latentes. É amplamente utilizado em NLP.
T-Distributed Stochastic Neighbor Embedding (t-SNE)	Especialmente eficaz para visualização de dados de alta dimensão ao preservar relações de similaridade.
Locally Linear Embedding (LLE)	Preserva a estrutura local dos dados, indicado para dados com múltiplas subestruturas complexas.
Truncated Singular Value Decomposition (SVD)	Decompõe a matriz de dados para reduzir a dimensionalidade, sendo muito eficiente para dados textuais (como em LSA).
Incremental PCA	Variante do PCA para grandes volumes de dados que não cabem na memória, ideal para aplicações de Big Data.
Random Trees Embedding	Usa árvores de decisão para criar embeddings, adequado para transformar dados complexos para modelos lineares.
SelectKBest	Seleciona as melhores features com base em testes estatísticos, útil para problemas em que apenas algumas variáveis são significativas.
Spectral Embedding	Explora as propriedades espectrais dos dados, sendo adequado para dados que se distribuem em um grafo.

Tabela 3.5: Algoritmos de Redução de Dimensionalidade Selecionados

3.5 Métricas

Neste trabalho, abordamos duas métricas importantes para avaliação de modelos de classificação: o F-score (ou F1-score) e o coeficiente de correlação de Spearman. Ambas são essenciais para avaliar o desempenho dos modelos em diferentes contextos no presente trabalho.

O F-score, também conhecido como F1-score (Yacouby and Axman, 2020), é uma medida de precisão que leva em consideração tanto a precisão quanto a revocação (ou recall) de um modelo de classificação. O mesmo foi usado para medir o desequilíbrio nas classes, pois oferecia uma visão mais equilibrada do desempenho do modelo. Uma explicação mais detalhada encontra-se no Capítulo 2.

O coeficiente de correlação de Spearman (Restrepo and González, 2007) é uma medida não-paramétrica que avalia a relação monotônica entre duas variáveis, sem pressupor uma distribuição específica dos dados. Ele é particularmente útil quando as variáveis não seguem uma distribuição normal ou quando a relação entre elas é monotônica, mas não necessariamente linear.

O coeficiente de Spearman é calculado a partir das classificações das variáveis, e sua fórmula é dada por:

$$\rho = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{3.1}$$

Onde:

- d_i é a diferença entre os postos de cada observação.
- n é o número de observações.

O valor de ρ varia entre -1 e 1, onde:

- **1** indica uma correlação positiva perfeita,
- **-1** indica uma correlação negativa perfeita,
- **0** indica nenhuma correlação monotônica.

No contexto deste trabalho, o coeficiente de Spearman foi utilizado para calcular a correlação entre os rankings de dois conjuntos de dados, visando medir o grau de associação entre as classificações de diferentes variáveis.

3.6 Algoritmos bases usados

3.6.1 KNN sem redução de dimensionalidade

Neste trabalho, utilizamos o algoritmo de aprendizage de máquina KNN (K-Nearest Neighbors ou K-Vizinhos Mais Próximos). Optamos por esse método devido à sua simplicidade e por se adequar bem a problemas de classificação, como o que estamos abordando. O KNN classifica novos pontos com base na proximidade em relação a pontos já conhecidos no conjunto de dados. Ao definir um valor de $K = 3$, por exemplo, o algoritmo analisa os três pontos mais próximos do ponto que se deseja classificar. A proximidade entre os pontos é medida, em geral, pela distância Euclidiana, conforme apresentada a seguir:

$$d(A,B) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + \cdots + (a_n - b_n)^2} \tag{3.2}$$

$A = (a_1, a_2, \ldots, a_n)$ e $B = (b_1, b_2, \ldots, b_n)$ representam o ponto de teste e um ponto do conjunto de treinamento, respectivamente, sendo n o número de características (features) de cada ponto.

O objetivo inicial foi realizar a classificação de 66 datasets sem utilizar algoritmos de redução de dimensionalidade, para que fosse possível comparar os resultados com e sem essa técnica posteriormente. Para classificar cada dataset, utilizamos valores de k variando entre 1 e 31, de modo a observar o impacto dessa variação nos resultados obtidos. Após a seleção de k, dividimos o conjunto de dados em 70% para treino e 30% para teste, utilizando a função ‘train_test_split()‘, que permite avaliar de forma confiável o desempenho do modelo em dados não vistos. Por fim, utilizamos a métrica F1-score para medir o desempenho dos algoritmos, considerando a média "micro", que leva em conta o balanceamento entre as classes.

3.6.2 KNN com redução de dimensionalidade

Após realizar a classificação sem a aplicação de técnicas de redução de dimensionalidade, foi essencial incorporar essa etapa para avaliar o impacto no desempenho do algoritmo. Para essa tarefa, mantivemos a mesma estrutura do KNN, com valores de k variando entre 1 e 31, conforme discutido na subseção anterior. No entanto, antes de proceder à classificação, aplicamos diferentes algoritmos de redução de dimensionalidade.

A redução de dimensionalidade é uma técnica que visa transformar um conjunto de dados de alta dimensionalidade em uma representação mais compacta, preservando o máximo possível da variância original dos dados.

Para a escolha do número ideal de componentes a serem mantidos em cada algoritmo de redução, adotamos diversas abordagens, como Kaiser’s stopping rule (que retém apenas os componentes com autovalores maiores que 1), A priori criterion (que define previamente o número de componentes com base no conhecimento sobre o problema), entre outras técnicas discutidas

na literatura, como sugerido por Brown (2009) (Brown, 2009).

Matematicamente, a escolha do número de componentes em métodos como Análise de Componentes Principais (PCA) é definida pela decomposição da matriz de covariância dos dados. Dados os autovalores $\lambda_1, \lambda_2, \ldots, \lambda_n$, associados aos componentes principais, o número m de componentes selecionados maximiza a seguinte relação:

$$\frac{\sum_{i=1}^{m} \lambda_i}{\sum_{i=1}^{n} \lambda_i} \tag{3.3}$$

onde m é o número de componentes que retém uma quantidade significativa da variância dos dados.

Assim, após a redução dimensionalidade, os dados foram divididos novamente em 70% para treino e 30% para teste, mantendo a consistência metodológica do experimento anterior. A métrica F1-score foi utilizada para avaliar o desempenho dos modelos reduzidos, utilizando a média "micro"para contabilizar o balanceamento entre as classes. Dessa forma, foi possível comparar o desempenho do KNN em cenários com e sem redução de dimensionalidade.

Após a classificação com redução de dimensionalidade, foi criado um dataset contendo os resultados de classificação para cada algoritmo de redução. Em seguida, foi gerado um ranking para avaliar o desempenho relativo entre esses algoritmos. O ranking foi calculado da seguinte forma: para cada um dos 66 datasets, os valores de desempenho dos algoritmos foram ordenados, e atribuímos uma pontuação de 10 ao melhor desempenho, 9 ao segundo, e assim sucessivamente, até o pior desempenho, que recebeu pontuação 1. O resultado foi um DataFrame contendo os rankings dos algoritmos: IncrementalPCA, KPCA, LDA, LLE, PCA, RTreeE, SelectBest, SpectralEmbedding, TruncatedSVD, e t-SNE.

Index	IncrementalPCA	KPCA	LDA	LLE	PCA	RTreeE	SelectBest	SpectralEmbedding	TruncatedSVD	t-SNE
0	4	3	6	2	5	8	1	10	9	8
1	4	2	5	8	4	6	2	9	7	10
2	4	7	2	5	4	7	2	10	10	10
3	10	10	10	1	10	6	3	4	3	6
4	5	5	5	8	5	7	5	7	10	9
5	4	4	6	10	4	7	4	6	8	9
6	4	5	4	10	4	1	6	10	8	7
7	6	6	3	7	6	3	3	8	10	9
8	5	5	2	10	3	8	2	8	6	9
9	5	5	2	8	5	6	2	9	7	10

Tabela 3.6: Ranking dos algoritmos de redução de dimensionalidade para os datasets.

Em seguida, o ranking foi combinado com as meta-features (características dos datasets) para analisar a relação entre as propriedades dos datasets e o desempenho dos algoritmos de redução. As meta-features foram usadas para descrever as características que descrevem a estrutura dos datasets. Nesse caso, as meta-features serviram como as variáveis explicativas ($\mathbf{X}$), enquanto o desempenho do algoritmo foi a variável alvo ($\mathbf{y}$).

Para realizar a classificação, utilizamos o algoritmo **RandomForestClassifier**, um método baseado em um conjunto de árvores de decisão. O Random Forest constrói várias árvores de decisão durante o treinamento e, para realizar a classificação, utiliza a média dos votos de todas

as árvores. Matematicamente, o modelo pode ser representado como uma coleção de árvores $T_1, T_2, \ldots, T_B$, onde B é o número de árvores. Cada árvore faz uma previsão, e a classe final $\hat{y}$ é determinada pela maioria dos votos:

$$\hat{y} = \text{argmax} \sum_{b=1}^{B} I(T_b(x) = y) \tag{3.4}$$

Para validar o modelo, utilizamos a técnica de **Leave-One-Out Cross-Validation (LOO)**. Nesta abordagem, o modelo é treinado em $n-1$ exemplos e testado em 1 exemplo, sendo n o número total de exemplos. Esse processo é repetido n vezes, garantindo que cada dado seja utilizado tanto para treino quanto para teste. As variáveis X consistiram nas meta-features, e y representou o desempenho do algoritmo.

Index	IncrementalPCA	KPCA	LDA	LLE	PCA	RTreeE	SelectBest	SpectralEmbedding	TruncatedSVD	t-SNE
0	0.957447	0.958235	0.947991	0.963751	0.956659	0.946414	0.964539	0.921198	0.944050	0.946414
1	0.757576	0.803030	0.651515	0.333333	0.757576	0.636364	0.803030	0.318182	0.590909	0.166667
2	0.818182	0.681818	0.954545	0.727273	0.818182	0.681818	0.954545	0.590909	0.590909	0.590909
3	0.388889	0.388889	0.388889	0.666667	0.388889	0.444444	0.611111	0.500000	0.611111	0.444444
4	1.000000	1.000000	1.000000	0.975806	1.000000	0.991935	1.000000	0.991935	0.709677	0.798387
5	0.950820	0.950820	0.942623	0.483607	0.950820	0.909836	0.950820	0.942623	0.877049	0.819672
6	0.885246	0.868852	0.885246	0.180328	0.885246	0.901639	0.852459	0.180328	0.672131	0.786885
7	0.954545	0.954545	0.977273	0.840909	0.954545	0.977273	0.977273	0.090909	0.022727	0.045455
8	0.854839	0.854839	0.895161	0.508065	0.887097	0.838710	0.895161	0.838710	0.846774	0.524194
9	0.905512	0.905512	0.929134	0.779528	0.905512	0.850394	0.929134	0.755906	0.818898	0.551181

Tabela 3.7: Junção dos datasets com resultados classificação

Por fim, é gerado um ranking com base nas predições realizadas a partir da combinação das meta-features e do desempenho dos algoritmos. Esse ranking, que reflete a classificação obtida a partir das meta-features associadas aos algoritmos de redução de dimensionalidade, é denominado "meta-base"neste trabalho.

Index	IncrementalPCA	KPCA	LDA	LLE	PCA	RTreeE	SelectBest	SpectralEmbedding	TruncatedSVD	t-SNE
0	3	2	6	10	5	4	1	7	8	9
1	9	9	5	10	9	7	2	2	9	10
2	10	10	10	1	10	6	2	4	3	6
3	4	7	2	5	4	7	2	10	10	10
4	5	5	2	10	4	7	2	9	8	9

Tabela 3.8: Ranking Meta Base

3.7 Meta-aprendiz

O objetivo deste trabalho é criar um modelo de recomendação de algoritmos capaz de otimizar tanto o tempo quanto o custo computacional no processamento de dados, além de desenvolver um modelo que possa gerar suas próprias estratégias de aprendizagem por meio da utilização de meta-dados. Para isso, construímos um repositório com 66 conjuntos de dados, cada um contendo entre 1.000 e 30.000 variáveis (features). Esses dados foram escolhidos de forma diversa para garantir uma boa representatividade de diferentes domínios e níveis de complexidade.

Para a avaliação dos dados, selecionamos 10 algoritmos de redução de dimensionalidade, incluindo técnicas como Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), Latent Dirichlet Allocation (LDA) e Random Trees Embedding. A

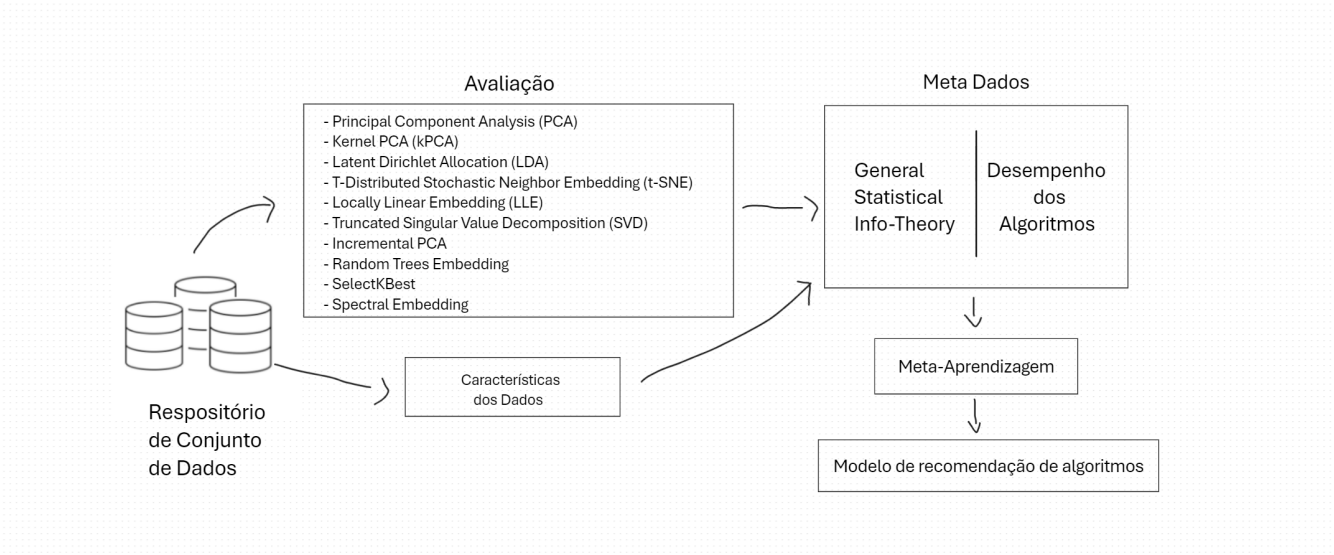

Figura 3.2: Arquitetura do Meta-Aprendiz

escolha dos algoritmos não seguiu um critério específico, pois nosso foco foi explorar diferentes abordagens que pudessem capturar a variabilidade dos conjuntos de dados de maneira ampla.

As Características dos Dados, neste contexto, referem-se a descritores ou atributos extraídos dos conjuntos de dados que servem para resumir suas propriedades, como o número de variáveis, a dimensionalidade e a complexidade estrutural dos dados. Essas características, conhecidas como meta-features, são essenciais para o processo de meta-aprendizagem. As meta-features selecionadas foram divididas em três categorias principais: General, que envolve medidas estatísticas simples; Statistical, que inclui métricas mais avançadas, como variância e correlação; e Info-Theory, que avalia a quantidade de informação contida nos dados, como entropia e redundância.

Distribuições específicas dos dados também foram levadas em consideração para garantir que a análise capturasse corretamente as nuances dos diferentes tipos de dados. Essas características foram fundamentais para alimentar o modelo de recomendação de algoritmos, que utiliza a meta-aprendizagem para encontrar a melhor estratégia de redução de dimensionalidade para cada conjunto de dados com base em seu desempenho anterior.

Na figura 3.2, vemos que os dados passam por uma fase de avaliação onde as características são extraídas e utilizadas para formar os meta-dados. Esses meta-dados, que incluem tanto as descrições dos dados (como distribuição e complexidade) quanto o desempenho dos algoritmos aplicados a esses dados, são então alimentados no modelo de meta-aprendizagem. Dessa forma, possível criar um modelo capaz de aprender a partir dos meta-dados e, com base nas características de novos conjuntos de dados, recomendar os algoritmos mais adequados de forma automatizada, otimizando, assim, o tempo e o custo computacional.

4

Resultados e Discussões

Nesta seção, apresentamos os resultados obtidos a partir da análise dos conjuntos de dados e a criação de uma meta base de dados, que serviu como base para o desenvolvimento de um modelo de recomendação de algoritmos de redução de dimensionalidade.

4.1 Classificação usando KNN

4.1.1 Sem aplicação de algoritmos de redução de dimensionalidade

Ao realizar a classificação das bases de dados utilizando o algoritmo KNN Abu Alfeilat et al. (2019), com os parâmetros e métricas descritos no capítulo de metodologia, observamos desempenhos variados. Algumas bases de dados, como a **AP_Breast_Prostate**, atingiram um desempenho perfeito (1.0), enquanto outras apresentaram resultados bem abaixo, com desempenho inferior a 0.20, como por exemplo, **eating**. A maioria das bases, no entanto, obteve desempenhos entre 0.7 e 0.95.

É importante ressaltar que, até este ponto, não foram aplicados tratamentos ou pré- processamentos nos 66 conjuntos de dados analisados. Os resultados refletem o desempenho bruto, sem qualquer alteração nos dados originais. A partir da análise visual da imagem abaixo, podemos concluir que, de maneira geral, os resultados foram positivos, com uma média geral de desempenho de 0.79.

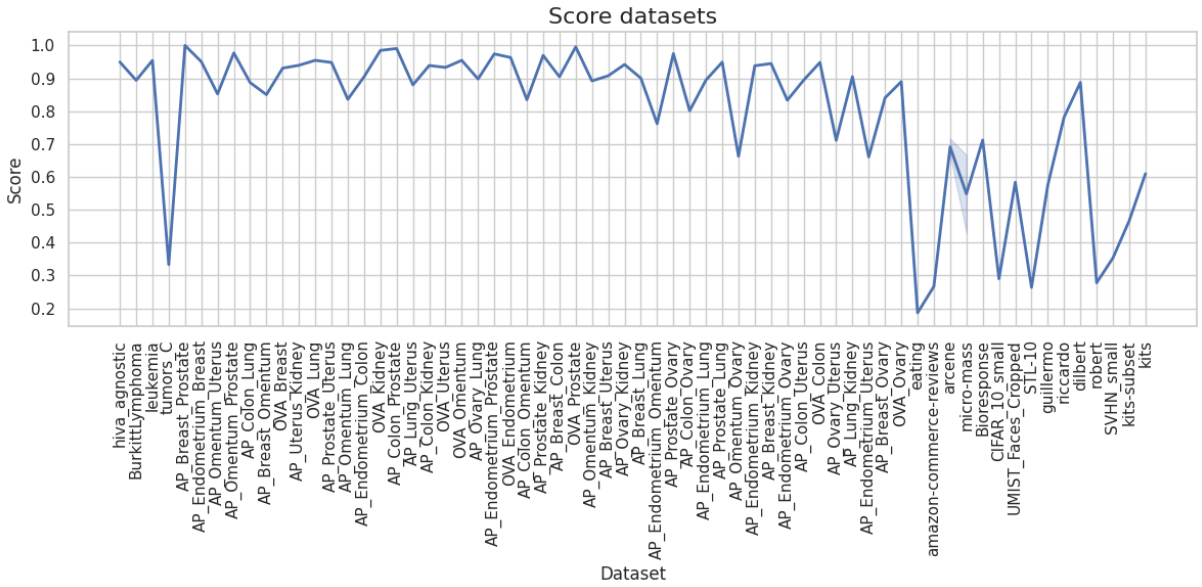

Figura 4.1: Desempenho sem redução de dimensionalidade

Na figura abaixo (Boxplot), observa-se a concentração dos dados e a presença de possíveis outliers. A mediana, juntamente com os quartis (Q1 e Q3), indica que a maioria dos valores de Score está concentrada entre 0,7 e 0,9. Os whiskers se estendem para representar a dispersão dos dados dentro de 1,5 vezes o intervalo interquartil, revelando que há poucos outliers, evidenciando, portanto, uma baixa incidência de valores atípicos no conjunto de dados.

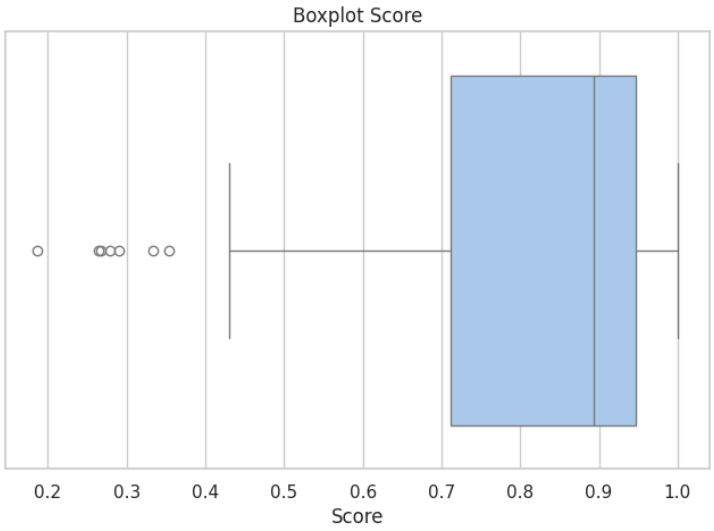

Figura 4.2: Boxplot desempenho dos datasets

4.1.2 Com Aplicação de algoritmos de redução de dimensionalidade

Para realizar a classificação dos conjuntos de dados, foi aplicada a redução de dimensionalidade em cada um deles. Nesta etapa, o método K-Nearest Neighbors (KNN) (Abu Alfeilat

et al. (2019))) foi utilizado com os parâmetros e métricas previamente descritos no capítulo de Metodologia. No entanto, foi necessária a normalização dos dados, uma vez que alguns algoritmos de redução de dimensionalidade requerem que os dados estejam normalizados para um desempenho eficaz. Para a normalização, foi utilizado o StandardScaler, que transforma as características para terem média igual a 0 e desvio padrão igual a 1.

A classificação foi realizada com 10 diferentes algoritmos de redução de dimensionalidade, e as imagens a seguir de alguns algoritmos ilustram o desempenho de cada abordagem, permitindo uma comparação entre os resultados com e sem a aplicação da redução de dimensionalidade.

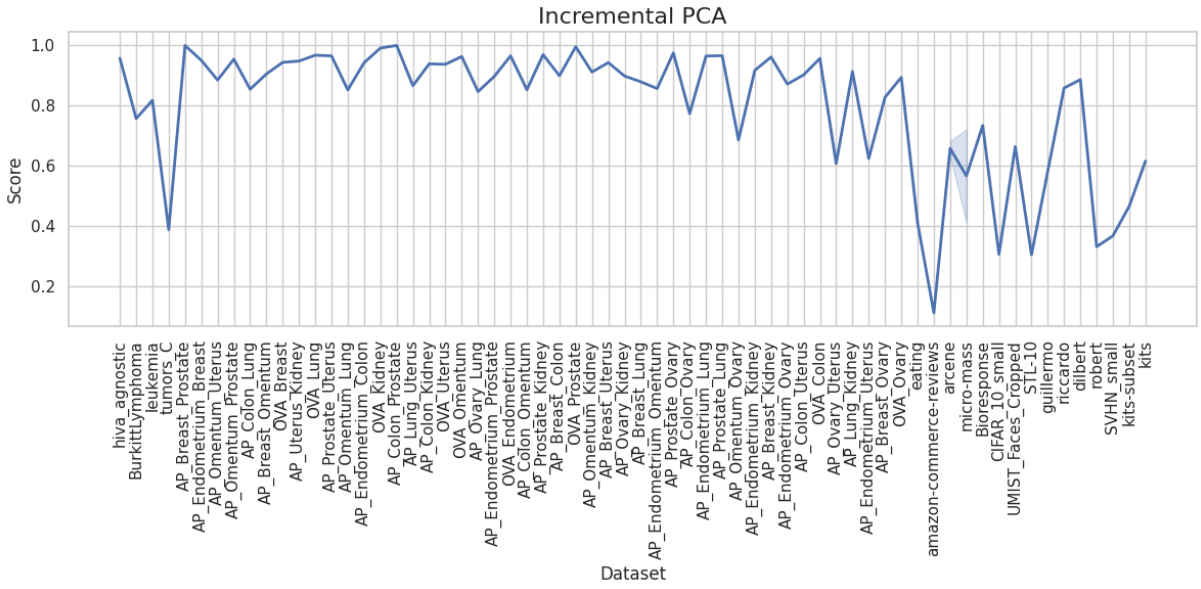

Figura 4.3: Desempenho dos datasets

Na Figura 4.3, observa-se que, ao aplicar o algoritmo Incremental PCA, os valores de desempenho não apresentam mudanças significativas em comparação com os resultados obtidos sem a redução de dimensionalidade. A principal diferença está na variação de desempenho entre diferentes conjuntos de dados: alguns mantêm um desempenho mais baixo, enquanto outros apresentam uma melhora mais notável.

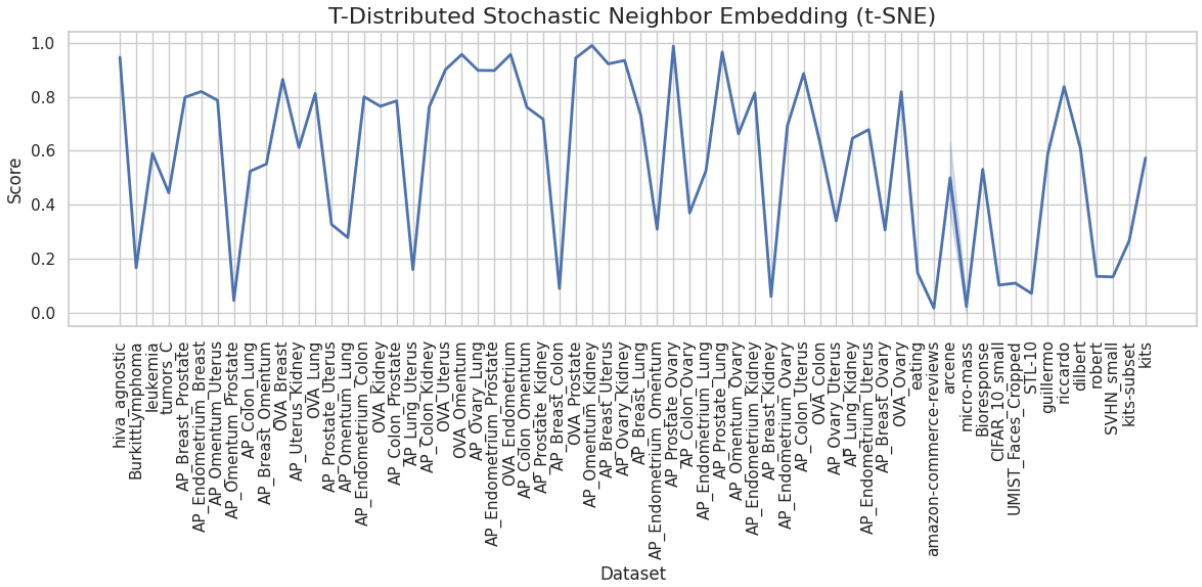

Figura 4.4: Desemenho dos datasets

Observando a Figura 4.4, percebe-se que o algoritmo T-Distributed Stochastic Neighbor Embedding (t-SNE) apresenta um desempenho consideravelmente diferente dos gráficos anteriores. Os valores estão mais dispersos e não demonstram resultados tão satisfatórios em termos de desempenho, comparados aos demais algoritmos.

Talvez essa diferença de desempenho entre os dois algoritmos usados na classificação com redução de dimensionalidade esteja no fato de que um, o t-SNE, é focado em preservar relações de proximidade local, o que é útil para visualizações detalhadas, mas pode distorcer a estrutura global dos dados. Já o Incremental PCA preserva a variância global dos dados de forma linear, sendo mais eficiente e adequado para análise quantitativa e conjuntos de dados grandes.

Foram citados apenas dois dos dez algoritmos utilizados neste trabalho; os demais apresentaram desempenhos próximos aos desses dois mencionados. Assim, observa-se que o uso de técnicas de redução de dimensionalidade frequentemente proporciona melhorias no desempenho. No nosso cenário, essas melhorias foram observadas em alguns conjuntos de dados, enquanto em outros, não ocorreram Reddy et al. (2020).

A figura abaixo apresenta boxplots que comparam o desempenho de 10 algoritmos de redução de dimensionalidade. A distribuição dos resultados varia entre os algoritmos, com alguns, como Incremental PCA e SelectBest, mostrando desempenho mais consistente e concentrado em valores altos, enquanto outros, como t-SNE, exibem maior dispersão, indicando maior variabilidade nos resultados.

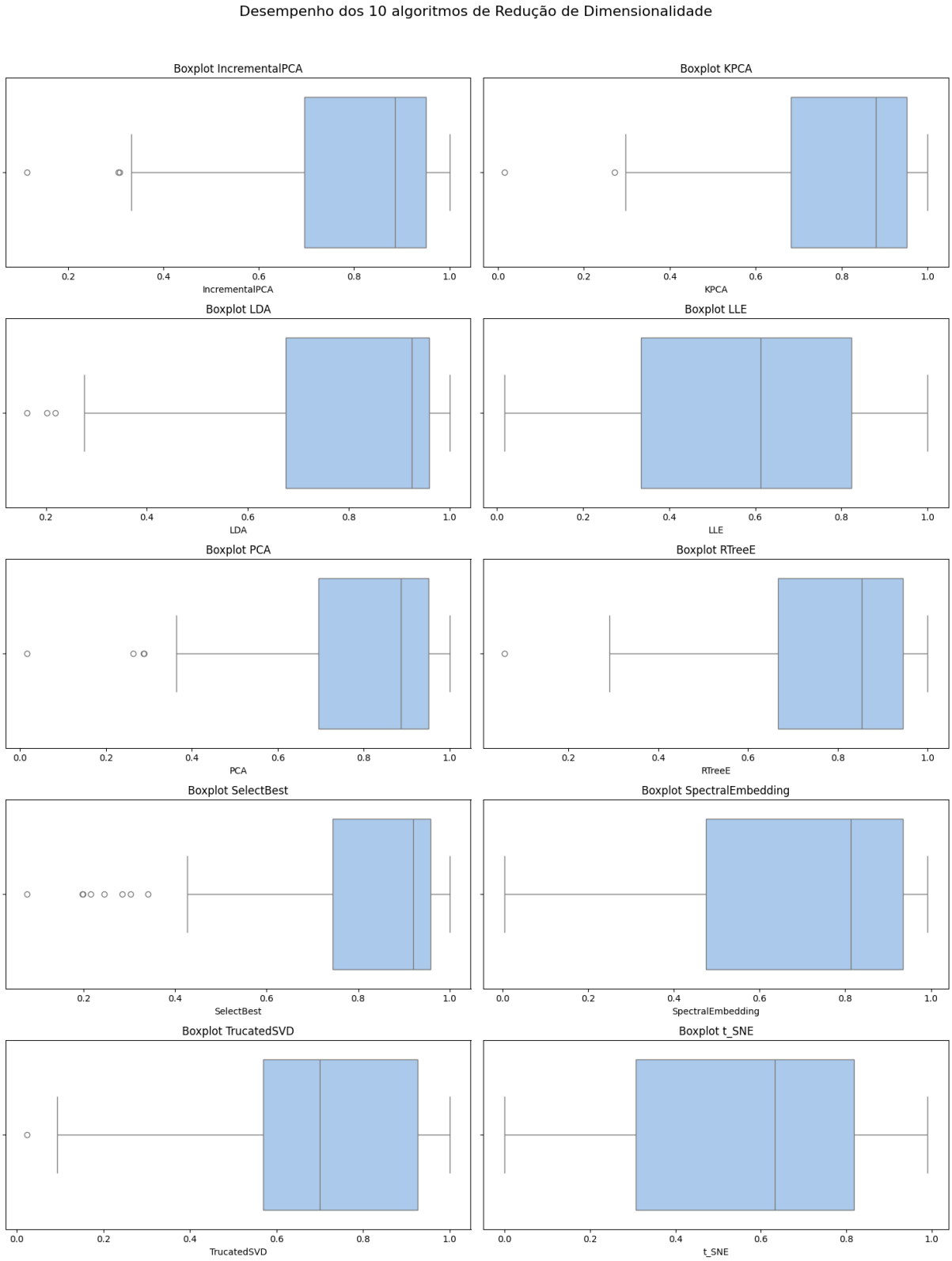

Figura 4.5: Ranking dos algoritmos de redução de dimensionalidade para os datasets

4.2 Comparação os resultados do modelo de recomendação de algoritmo

4.2.1 Meta-features

Nesta seção, apresentaremos os resultados utilizando a metafeature general na metabase, a fim de criar o meta-aprendizagem e possibilitar a construção do sistema de recomendação de algoritmos. A figura abaixo mostra o ranking dos algoritmos de redução de dimensionalidade, o qual será comparado com o ranking gerado pelo modelo de recomendação, para avaliar seu desempenho. Observa-se que muitos dos rankings são semelhantes, o que pode ser explicado pelo fato de que o desempenho dos algoritmos é igual ou muito próximo entre si.

index	IncrementalPCA	KPCA	LDA	LLE	PCA	RTreeE	SelectBest	SpectralEmbedding	TrucatedSVD	t_SNE
0	4	3	6	2	5	8	1	10	9	8
1	4	2	5	8	4	6	2	9	7	10
2	4	7	2	5	4	7	2	10	10	10
3	10	10	10	1	10	6	3	4	3	6
4	5	5	5	8	5	7	5	7	10	9
5	4	4	6	10	4	7	4	6	8	9
6	4	5	4	10	4	1	6	10	8	7
7	6	6	3	7	6	3	3	8	10	9
8	5	5	2	10	3	8	2	8	6	9
9	5	5	2	8	5	6	2	9	7	10
10	3	5	3	9	4	6	1	10	8	7
11	6	2	6	10	2	6	8	6	7	9
12	2	4	4	8	5	6	1	8	9	10
13	7	1	7	10	7	7	7	8	7	9
14	4	4	5	7	4	7	1	10	8	9
15	6	6	2	10	8	6	2	3	8	9
16	1	3	5	10	3	7	5	6	8	9
17	8	8	8	8	8	8	8	9	8	10
18	4	6	2	10	4	6	1	9	7	8
19	4	5	2	10	4	7	2	8	7	9
20	3	3	8	6	3	7	4	9	6	10
21	1	8	10	7	2	7	9	7	7	7
22	7	8	1	4	5	9	2	7	10	3
23	9	9	2	10	9	4	2	9	4	9
24	5	8	9	5	8	8	1	5	5	10

Figura 4.6: Ranking dos algoritmos de redução de dimensionalidade para os datasets

A criação do ranking foi realizada utilizando a metafeature geral, e os algoritmos de redução de dimensionalidade foram combinados para formar nossa metabase de dados. Para predizer os valores do ranking, utilizamos o algoritmo Random Forest, com o objetivo de identificar qual algoritmo de redução de dimensionalidade seria mais indicado para cada contexto. Os resultados apresentados na figura abaixo mostram que o modelo de recomendação de algoritmos, na maioria das vezes, acerta a indicação do algoritmo correto. No entanto, também observamos que, em várias situações, o modelo comete erros. Esse desempenho abaixo do esperado pode ser atribuído à necessidade de um ajuste mais fino nos parâmetros do nosso modelo de recomendação, o que possibilitaria predições mais precisas e eficientes.

index	ML_IncrementalPCA	ML_KPCA	ML_LDA	ML_LLE	ML_PCA	ML_RTreeE	ML_SelectBest	ML_SpectralEmbedding	ML_TrucatedSVD	ML_t_SNE
0	3	2	6	10	5	4	1	7	8	9
1	9	9	5	10	9	7	2	2	9	9
2	10	10	10	1	10	6	3	4	3	6
3	4	7	2	5	4	7	2	10	10	10
4	5	5	2	10	4	7	2	9	8	9
5	5	5	2	8	5	7	5	7	10	9
6	4	4	5	7	4	7	1	10	8	9
7	7	5	3	3	7	8	4	9	9	10
8	5	7	2	10	4	1	7	3	10	8
9	5	5	5	8	5	7	5	7	10	9
10	4	6	3	7	6	3	3	8	9	10
11	6	6	6	10	6	8	1	7	6	9
12	3	3	8	6	3	7	4	9	6	10
13	6	6	1	10	6	9	2	9	9	6
14	4	5	4	10	4	1	6	10	8	7
15	8	8	8	8	8	8	8	9	8	10
16	4	6	3	7	6	3	3	8	9	10
17	6	8	2	10	8	6	1	3	3	9
18	4	4	3	10	4	3	3	8	7	9
19	4	7	1	8	5	3	2	8	9	10
20	2	4	4	8	5	6	1	8	9	10
21	6	6	6	8	6	6	6	9	8	10
22	6	6	2	10	4	1	3	8	7	10
23	7	5	3	3	7	3	4	1	9	10
24	6	6	6	8	6	6	6	9	8	10

Figura 4.7: Ranking dos algoritmos usando Meta Dados e Metafeature General

Nos resultados obtidos com a comparação dos rankings de algoritmos de redução de dimensionalidade, utilizando a correlação de Spearman, foi possível observar diferentes desempenhos ao considerar as médias e medianas dos valores. A correlação média dos rankings originais em relação à média das iterações do processo de Leave-One-Out mostrou coeficientes variando principalmente entre 0.5 e 0.9, com alguns casos de correlações negativas, como -0.204, sugerindo que os rankings médios se aproximam, em sua maioria, dos rankings previstos, mas ainda apresentam divergências notáveis.

Ao analisar o impacto das medianas, os coeficientes de correlação de Spearman demonstraram uma tendência levemente superior em relação às médias, com alguns valores ultrapassando 0.9 e indicando uma melhor adequação do modelo de recomendação. No entanto, também foram observadas correlações negativas mais acentuadas, como -0.43, sugerindo que, em certos contextos, as medianas podem levar a discrepâncias maiores em relação aos rankings originais. Esses resultados indicam a necessidade de ajustes mais precisos nos parâmetros do modelo para reduzir as variações nos rankings e aprimorar a precisão das predições.

index	SRC_Algoritmo	SRC_Media	SRC_Mediana
0	0.4363636363636364	0.5516829836829835	0.5696969696969697
1	0.04242424242424475	0.863351627362381	0.9030303030303031
2	-0.6000000000000001	0.7260735162273624	0.7818181818181819
3	-0.6000000000000001	-0.20457737134660237	-0.4303030303030304
4	0.8121212121212121	0.9169374215528062	0.9212121212121213
5	0.8242424242424242	0.907848664156356 5	0.9212121212121213
6	0.5393939393939393	0.7429255872332795	0.7333333333333334
7	0.7090909090909091	0.856913752913752 9	0.8545454545454545
8	0.2666666666666667	0.8491504393042855	0.8848484848484849
9	0.8	0.9112569481800251	0.9393939393939394
10	0.7757575757575758	0.8449847588309127	0.9090909090909091
11	0.4727272727272728	0.7003220369374215	0.6545454545454545
12	0.7212121212121212	0.8495291375291376	0.9272727272727272
13	0.3575757575757575 5	0.683848664156356 4	0.6545454545454545
14	0.5393939393939393	0.8512332795409718	0.9030303030303031
15	0.2424242424242243	0.7361090191859423	0.7272727272727273
16	0.6	0.7902628653397883	0.8303030303030303
17	0.03636363636363636	0.5602036937421553	0.503030303030303
18	0.8787878787878788	0.8627835753989 6	0.9151515151515152
19	0.8121212121212121	0.8974344629729245	0.9454545454545454
20	0.7212121212121212	0.7510675990675991	0.7454545454545455
21	0.4787878787878876	0.3979315043930 4303	0.3333333333333337
22	-0.012121212121212 2	0.4149729245113 8623	0.3757575757575757 3
23	-0.024242424242424176	0.5353989600143447	0.4909090909090096
24	0.509090909090909	0.5342628653397885	0.4727272727272728

Figura 4.8: Tabela de Correlação de Spearman-General

A Figura 4.9 apresenta os resultados da comparação entre os rankings de algoritmos de

redução de dimensionalidade, utilizando a correlação de Spearman aplicada à meta-base com a metafeature statistical. Nota-se que os resultados são bastante semelhantes aos obtidos com metafeature General. Já a Figura 4.10 mostra os coeficientes de correlação de Spearman, onde se observam tanto correlações positivas quanto negativas

index	ML_IncrementalPCA	ML_KPCA	ML_LDA	ML_LLE	ML_PCA	ML_RTreeE	ML_SelectBest	ML_SpectralEmbedding	ML_TrucatedSVD	ML_t_SNE
0	3	2	6	10	5	4	1	7	8	9
1	4	1	6	10	3	5	7	9	8	8
2	10	10	10	1	10	6	3	4	3	6
3	4	7	2	5	4	7	2	10	10	10
4	4	4	6	10	4	8	4	7	8	9
5	5	5	3	10	5	6	2	7	9	9
6	7	6	3	10	4	8	1	6	9	9
7	7	1	2	10	7	7	8	8	7	9
8	5	7	2	10	4	1	1	3	10	9
9	4	4	3	10	4	8	6	7	9	9
10	4	6	3	7	6	3	3	8	9	10
11	8	9	2	10	8	8	5	4	2	9
12	5	8	9	5	8	8	1	5	5	10
13	8	6	3	10	6	6	2	9	10	9
14	4	6	2	10	4	9	1	9	10	8
15	7	8	2	10	4	8	7	7	3	10
16	6	6	6	8	6	6	6	9	8	10
17	7	6	2	10	4	7	2	8	10	9
18	4	3	5	10	4	7	1	10	8	9
19	5	8	2	6	6	8	2	8	9	10
20	2	5	5	6	3	8	1	10	8	9
21	3	3	8	6	3	7	4	9	6	10
22	6	6	2	10	5	1	1	10	9	10
23	8	6	3	10	8	7	3	8	9	9
24	2	5	5	6	3	8	1	10	8	9

Figura 4.9: Ranking dos algoritmos usando Meta Dados e Metafeature Statistical

index	SRC_Algoritmo	SRC_Media	SRC_Mediana
0	0.4363636363636364	0.5516829836829835	0.5696969696969697
1	0.7696969696969697	0.8633516227362381	0.9030303030303031
2	-0.6000000000000001	0.7260735162273624	0.7818181818181819
3	-0.6000000000000001	-0.20457737134660237	-0.4303030303030304
4	0.9151515151515152	0.9169374215528062	0.9212121212121213
5	0.8848484848484849	0.9078466641563565	0.9212121212121213
6	0.35757575757575755	0.7429255872332795	0.7333333333333334
7	0.4727272727272728	0.8569137529137529	0.8545454545454545
8	0.4181818181818182	0.8491504393042855	0.8848484848484849
9	0.7757575757575758	0.9112569481800251	0.9393939393939394
10	0.7757575757575758	0.8449847588309127	0.9090909090909091
11	0.10909090909090913	0.7003220369374215	0.6545454545454545
12	0.4121212121212121	0.8495291375291376	0.9272727272727272
13	0.5212121212121212	0.6838486641563564	0.6545454545454545
14	0.806060606060606	0.8512332795409718	0.9030303030303031
15	0.4424242424242424	0.7361090191859423	0.7272727272727273
16	0.6363636363636364	0.7902628653397883	0.8303030303030303
17	0.36969696969696975	0.5602036937421553	0.503030303030303
18	0.8666666666666667	0.86278357539896	0.9151515151515152
19	0.7818181818181819	0.8974344629729245	0.9454545454545454
20	0.8181818181818181	0.7510675990675991	0.7454545454545455
21	0.5515151515151515	0.3979315043930430 3	0.3333333333333337
22	-0.0060606060606061	0.41497292451138623	0.37575757575757573
23	0.7090909090909091	0.5353989600143447	0.49090909090909096
24	0.4242424242424242	0.5342628653397885	0.472727272727728

Figura 4.10: Correlação de Spearman-Statistical

Além disso, os resultados obtidos utilizando a metafeature Info-Theory (figura 4.11) para a criação do modelo de recomendação de algoritmos apresentam desempenhos similares aos modelos que utilizaram as metafeatures General e Statistical. Ao comparar os rankings das Figuras 4.6 e 4.11, nota-se que a precisão da recomendação dos algoritmos não é tão elevada. Observando a correlação de Spearman, os valores obtidos são bastante próximos dos encontrados para as outras metafeatures, reforçando a consistência dos resultados.

index	ML_IncrementalPCA	ML_KPCA	ML_LDA	ML_LLE	ML_PCA	ML_RTreeE	ML_SelectBest	ML_SpectralEmbedding	ML_TrucatedSVD	ML_t_SNE
0	2	6	6	10	5	3	6	10	8	9
1	3	6	2	10	4	6	1	9	7	8
2	9	9	5	5	9	6	3	9	9	9
3	9	6	2	10	3	7	2	10	9	9
4	6	7	2	10	8	6	7	3	10	9
5	2	5	2	10	5	6	2	7	9	9
6	4	4	2	10	4	6	1	7	8	9
7	9	9	2	10	9	4	2	9	4	9
8	4	7	2	10	4	7	1	7	10	9
9	3	6	2	10	4	6	1	6	8	10
10	4	6	5	8	6	7	1	9	9	10
11	9	9	6	10	6	8	1	4	6	6
12	3	3	8	6	3	7	1	9	6	10
13	6	6	3	10	6	9	2	9	10	9
14	4	5	2	10	4	6	2	10	9	10
15	7	5	2	10	4	7	1	7	10	9
16	6	5	3	8	6	3	2	10	9	10
17	8	8	8	10	8	6	8	8	7	9
18	7	5	2	10	4	1	2	7	10	9
19	3	9	3	6	3	6	6	9	9	6
20	2	4	4	8	5	6	1	8	9	10
21	5	8	9	5	8	8	1	5	5	10
22	4	5	2	10	4	1	2	7	8	9
23	6	6	3	10	6	3	3	9	10	9
24	1	8	10	7	2	7	9	7	7	10

Figura 4.11: Ranking dos algoritmos usando Meta Dados e Metafeature Info Theory

index	SRC_Algoritmo	SRC_Media	SRC_Mediana
0	0.21818181818181814	0.5516829836829835	0.5696969696969697
1	0.7878787878787878	0.8633516227362381	0.9030303030303031
2	0.5878787878787879	0.7260735162273624	0.7818181818181819
3	-0.7818181818181817	-0.20457737134660237	-0.4303030303030304
4	0.7090909090909091	0.9169374215528062	0.9212121212121213
5	0.8242424242424242	0.9078486641563565	0.9212121212121213
6	0.5878787878787879	0.7429255872332795	0.7333333333333334
7	0.5393939393939393	0.8569137529137529	0.8545454545454545
8	0.8484848484848485	0.8491504393042855	0.8848484848484849
9	0.8727272727272728	0.9112569481800251	0.9393939393939394
10	0.8606060606060606	0.8449847588309127	0.9090909090909091
11	0.1454545454545455	0.7003220369374215	0.6545454545454545
12	0.7757575757575758	0.8495291375291376	0.9272727272727272
13	0.503030303030303	0.6838486641563564	0.6545454545454545
14	0.8606060606060606	0.8512332795409718	0.9030303030303031
15	0.7575757575757576	0.7361090191859423	0.7272727272727273
16	0.4606060606060606	0.7902628653397883	0.8303030303030303
17	0.9333333333333333	0.5602036937421553	0.503030303030303
18	0.696969696969697	0.862783575398 96	0.9151515151515152
19	0.6	0.8974344629729245	0.9454545454545454
20	0.7212121212121212	0.7510675990675991	0.7454545454545455
21	0.1575757575757576	0.3979315043930 4303	0.3333333333333337
22	0.030303030303030276	0.4149729245113 8623	0.3757575757575573
23	0.6	0.5353989600143447	0.4909090909090096
24	0.21212121212121215	0.5342628653397885	0.4727272727272728

Figura 4.12: Tabela de Correlação de Spearman-Info Theory

4.3 Conclusão

Como observado nos resultados, o modelo de recomendação de algoritmos, baseado nos rankings gerados a partir das metafeatures General, Statistical e Info-Theory, demonstrou eficácia ao identificar o algoritmo mais adequado para um determinado conjunto de dados, em comparação com o ranking dos datasets na Figura 4.6. A análise da correlação de Spearman revelou desempenhos variados ao considerar as médias e medianas dos valores. No entanto, algumas discrepâncias importantes foram observadas, como uma correlação negativa significativa (-0,204), assim como correlações mais altas, como 0,5 e 0,9, que indicam um bom desempenho ao utilizar a média. Ainda assim, melhorias no desempenho do modelo de recomendação são necessárias para reduzir a variação das predições, visando atingir correlações superiores a 0,6 de forma mais consistente.

5

Conclusão

Este trabalho investigou a aplicação de técnicas de meta-aprendizagem para automatizar a seleção de algoritmos de redução de dimensionalidade em cenários de Big Data. Através de uma combinação de metafeatures e modelos de recomendação, foi possível identificar algoritmos mais adequados para conjuntos de dados específicos, contribuindo significativamente para o aprimoramento do desempenho em tarefas de aprendizagem de máquina. Os resultados obtidos demonstraram a eficácia do método proposto, embora também tenham revelado pontos que exigem refinamentos adicionais.

O presente trabalho faz uma importante contribuição ao campo de aprendizagem de máquina, especialmente no contexto de problemas de alta dimensionalidade, ao propor uma abordagem de recomendação de algoritmos de redução de dimensionalidade baseada em meta-aprendizagem. Foi criado um modelo que, por meio do uso de metafeatures extraídas dos dados, automatiza a seleção de algoritmos de redução de dimensionalidade, otimizando tanto o tempo quanto o custo computacional de experimentos com grandes volumes de dados. Além disso, explorou-se o uso de metafeatures em três categorias principais General, Statistical e Info-Theory mostrando que essas características são valiosas para descrever e comparar o desempenho de algoritmos em diferentes contextos. A análise dos rankings gerados a partir dessas metafeatures destacou a relevância de cada categoria no processo de recomendação.

Os resultados demonstraram que o modelo de recomendação foi eficaz na maioria dos casos, conforme evidenciado pelas correlações de Spearman entre rankings originais e previstos. Essa avaliação reforçou a validade do método proposto, indicando que ele pode ser aplicado a diferentes domínios e conjuntos de dados. Ao aplicar algoritmos de redução de dimensionalidade, foi possível observar melhorias significativas no desempenho de algoritmos de aprendizagem de máquina em diversos conjuntos de dados, especialmente em termos de preservação da estrutura dos dados e da eficiência computacional. Essas contribuições colocam o presente trabalho como um avanço importante em sistemas automatizados e otimizados para a escolha de algorit-

mos em cenários com dados de alta dimensionalidade, oferecendo uma ferramenta promissora para pesquisadores que trabalham com grandes volumes de dados.

Embora o modelo de recomendação de algoritmos proposto tenha se mostrado promissor, ainda existem oportunidades para expandir e aprimorar este trabalho em futuros estudos. Uma área de melhoria está na otimização dos parâmetros dos modelos de recomendação. Um ajuste mais preciso desses parâmetros pode levar a uma maior precisão nas predições, reduzindo a variação observada entre rankings previstos e reais. Futuros trabalhos podem explorar outras metafeatures que captem aspectos mais profundos dos dados, como a complexidade estrutural, e essa ampliação poderia melhorar a qualidade das recomendações em domínios mais específicos.

Em conclusão, o presente trabalho representa um avanço significativo no uso da técnica de meta-aprendizagem para a recomendação de algoritmos de redução de dimensionalidade. A abordagem proposta demonstrou ser eficaz na automação e otimização da escolha de algoritmos para diferentes conjuntos de dados, com resultados promissores que evidenciam o potencial dessa metodologia. Embora ainda existam desafios a serem enfrentados, os resultados indicam que a aplicação de metafeatures e a recomendação automatizada de algoritmos podem se tornar ferramentas valiosas para problema de Big Data, possibilitando análises mais eficientes e precisas. O trabalho também oferece uma base sólida para futuras pesquisas, incluindo o desenvolvimento de sistemas mais robustos e aplicáveis a cenários reais.

Referências bibliográficas

Abdi, H. and Williams, L. J. (2010). Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Abu Alfeilat, H. A., Hassanat, A. B., Lasassmeh, O., Tarawneh, A. S., Alhasanat, M. B., Eyal Salman, H. S., and Prasath, V. S. (2019). Effects of distance measure choice on k-nearest neighbor classifier performance: a review. volume 7, pages 221–248.

Ali, P. J. M., Faraj, R. H., Koya, E., Ali, P. J. M., and Faraj, R. H. (2014). Data normalization and standardization: a technical report. *Mach Learn Tech Rep*, 1(1):1–6.

Barlow, H. B. (1989). Unsupervised learning. volume 1, pages 295–311. MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info ....

Belkina, A. C., Ciccolella, C. O., Anno, R., Halpert, R., Spidlen, J., and Snyder-Cappione, J. E. (2019). Automated optimized parameters for t-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature communications*, 10(1):5415.

Biau, G. and Scornet, E. (2016). A random forest guided tour. volume 25, pages 197–227. Springer.

BILALLI, Besim; ABELLÓ GAMAZO, A. A. B. T. (2017). On the predictive power of meta-features in openml. international journal of applied mathematics and computer science. 27(4):697–712.

Björck, Å. (1994). Numerics of gram-schmidt orthogonalization. *Linear Algebra and Its Applications*, 197:297–316.

Brereton, R. G. and Lloyd, G. R. (2010). Support vector machines for classification and regression. volume 135, pages 230–267. Royal Society of Chemistry.

BROWN, J. (2003). Choosing the right number of components or factors in pca and efa. 13(2).

Brown, J. (2009). Choosing the right number of components or factors in pca and efa. *JALT Testing & Evaluation SIG Newsletter*, 13(2).

Castiello, C., Castellano, G., and Fanelli, A. M. (2005). Meta-data: Characterization of input features for meta-learning. In *International conference on modeling decisions for artificial intelligence*, pages 457–468. Springer.

CHICCO, Davide; JURMAN, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. 21(1):1–13.

Chinnamgari, S. K. (2019). *R Machine Learning Projects: Implement supervised, unsupervised, and reinforcement learning techniques using R 3.5*. Packt Publishing Ltd.

Cunningham, P., Cord, M., and Delany, S. J. (2008). Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer.

Danielsson, P.-E. (1980). Euclidean distance mapping. volume 14, pages 227–248. Elsevier.

de Souza, B. F. (2011). Meta-aprendizagem aplicada à classificação de dados de expressão gênica.

Farrell, E. J. (1983). Color display and interactive interpretation of three-dimensional data. *IBM Journal of Research and Development*, 27(4):356–366.

Fasel, D. and Meier, A. (2014). *Big data*. Springer.

Fetzer, J. H. and Fetzer, J. H. (1990). What is artificial intelligence? Springer.

Flennerhag, S., Schroecker, Y., Zahavy, T., van Hasselt, H., Silver, D., and Singh, S. (2021). Bootstrapped meta-learning. *arXiv preprint arXiv:2109.04504*.

Frawley, W. J., Piatetsky-Shapiro, G., and Matheus, C. J. (1992). Knowledge discovery in databases: An overview. *AI magazine*, 13(3):57–57.

García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J. M., and Herrera, F. (2016). Big data preprocessing: methods and prospects. *Big data analytics*, 1:1–22.

Giraud-Carrier, C., Vilalta, R., and Brazdil, P. (2004). Introduction to the special issue on meta-learning. *Machine learning*, 54:187–193.

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., and Friedman, J. (2009). Unsupervised learning. pages 485–585. Springer.

Heil, B. J., Hoffman, M. M., Markowetz, F., Lee, S.-I., Greene, C. S., and Hicks, S. C. (2021). Reproducibility standards for machine learning in the life sciences. *Nature Methods*, 18(10):1132–1135.

Jakkula, V. (2006). Tutorial on support vector machine (svm). volume 37, page 3.

Japkowicz, N. (2013). Assessment metrics for imbalanced learning. *Imbalanced learning: Foundations, algorithms, and applications*, pages 187–206.

Jelodar, H., Wang, Y., Yuan, C., Feng, X., Jiang, X., Li, Y., and Zhao, L. (2019). Latent dirichlet allocation (lda) and topic modeling: models, applications, a survey. *Multimedia tools and applications*, 78:15169–15211.

Jia, W., Sun, M., Lian, J., and Hou, S. (2022). Feature dimensionality reduction: a review. volume 8, pages 2663–2693. Springer.

Kaelbling, L. P., Littman, M. L., and Moore, A. W. (1996). Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285.

Kurita, T. (2019). Principal component analysis (pca). *Computer vision: a reference guide*, pages 1–4.

Li, Y. (2017). Deep reinforcement learning: An overview. *arXiv preprint arXiv:1701.07274*.

Lorena, A. C. and De Carvalho, A. C. (2007). Uma introdução às support vector machines. volume 14, pages 43–67.

Mariscal, G., Marban, O., and Fernandez, C. (2010). A survey of data mining and knowledge discovery process models and methodologies. *The Knowledge Engineering Review*, 25(2):137–166.

McCarthy, J. (2007). What is artificial intelligence. Stanford University.

Meyer, D. and Wien, F. (2001). Support vector machines. volume 1, pages 23–26. Citeseer.

Mitchell, T. (1997). Machine learning. 1.

Mitchell, T. M. and Mitchell, T. M. (1997). *Machine learning*, volume 1. McGraw-hill New York.

Morales, E. F. and Escalante, H. J. (2022). A brief introduction to supervised, unsupervised, and reinforcement learning. In *Biosignal processing and classification using computational learning and intelligence*, pages 111–129. Elsevier.

Muhammad, I. and Yan, Z. (2015). Supervised machine learning approaches: A survey. ictact journal on soft computing. 5(3).

Novaković, J. D., Veljović, A., Ilić, S. S., Papić, Ž., and Tomović, M. (2017). Evaluation of classification models in machine learning. volume 7, page 39. "Aurel Vlaicu"University of Arad Department of Mathematics and Computer ....

Pavel Brazdil, Christophe Giraud G Carrier, C. S. R. V. (2009). Metalearning - applications to data mining. 1.

Pinto, F., Soares, C., and Mendes-Moreira, J. (2016). Towards automatic generation of metafeatures. In *Advances in Knowledge Discovery and Data Mining: 20th Pacific-Asia Conference, PAKDD 2016, Auckland, New Zealand, April 19-22, 2016, Proceedings, Part I 20*, pages 215–226. Springer.

POLAT, Kemal; GÜNEŞ, S. (2009). A new feature selection method on classification of medical datasets: Kernel f-score feature selection. expert systems with applications. 36(4):10367–10373.

PRUDÊNCIO, Ricardo BC; DE SOUTO, M. C. L. T. B. (2011). Selecting machine learning algorithms using the ranking meta-learning approach. meta-learning in computational intelligence. pages 225–243.

Qiang, W. and Zhongli, Z. (2011). Reinforcement learning model, algorithms and its application. In *2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)*, pages 1143–1146. IEEE.

Ragone, M., Braccia, P., Nguyen, Q. T., Schatzki, L., Coles, P. J., Sauvage, F., Larocca, M., and Cerezo, M. (2022). Representation theory for geometric quantum machine learning. *arXiv preprint arXiv:2210.07980*.

Raju, V. G., Lakshmi, K. P., Jain, V. M., Kalidindi, A., and Padma, V. (2020). Study the influence of normalization/transformation process on the accuracy of supervised classification. In *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pages 729–735. IEEE.

Reddy, G. T., Reddy, M. P. K., Lakshmanna, K., Kaluri, R., Rajput, D. S., Srivastava, G., and Baker, T. (2020). Analysis of dimensionality reduction techniques on big data. *Ieee Access*, 8:54776–54788.

Restrepo, L. F. and González, J. (2007). De pearson a spearman. *Revista Colombiana de Ciencias Pecuarias*, 20(2):183–192.

Rivolli, A., Garcia, L., Soares, C., Vanschoren, J., and de Carvalho, A. (1808). Characterizing classification datasets: a study of meta-features for metalearning. in arxiv.

Rivolli, A., Garcia, L. P., Soares, C., Vanschoren, J., and de Carvalho, A. C. (2022). Meta-features for meta-learning. *Knowledge-Based Systems*, 240:108101.

Sagiroglu, S. and Sinanc, D. (2013). Big data: A review. In *2013 international conference on collaboration technologies and systems (CTS)*, pages 42–47. IEEE.

Samuel, A. L. (2000). "some studies in machine learning using the game of checkers,"in ibm journal of research and development. 44:206–226.

Sarker, I. H. (2021). Machine learning: Algorithms, real-world applications and research directions. *SN computer science*, 2(3):160.

Scheinker, A. (2021). Adaptive machine learning for time-varying systems: low dimensional latent space tuning. *Journal of Instrumentation*, 16(10):P10008.

Semolini, R. et al. (2002). Support vector machines, inferência transdutiva e o problema de classificação.

Soman, K., Loganathan, R., and Ajay, V. (2009). Machine learning with svm and other kernel methods. PHI Learning Pvt. Ltd.

Soofi, A. A. and Awan, A. (2017a). Classification techniques in machine learning: applications and issues. *Journal of Basic & Applied Sciences*, 13:459–465.

Soofi, A. A. and Awan, A. (2017b). Classification techniques in machine learning: applications and issues. *J. Basic Appl. Sci*, 13(1):459–465.

Sorzano, C. O. S., Vargas, J., and Montano, A. P. (2014). A survey of dimensionality reduction techniques. *arXiv preprint arXiv:1403.2877*.

Stone, J. (2004). *Independent Component Analysis: A tutorial Introduction*. The MIT Press Cambridge.

Sutton, R. S. (1992). Introduction: The challenge of reinforcement learning. In *Reinforcement learning*, pages 1–3. Springer.

Tatachar, A. V. (2021). Comparative assessment of regression models based on model evaluation metrics. *International Journal of Innovative Technology and Exploring Engineering*, 8(9):853–860.

Taurion, C. (2013). *Big data*. Brasport.

Tsai, C.-W., Lai, C.-F., Chao, H.-C., and Vasilakos, A. V. (2015). Big data analytics: a survey. *Journal of Big data*, 2:1–32.

Turing, A. M. (2009). Computing machinery and intelligence. in parsing the turing test, pages. page 23–65.

Van Der Aalst, W. and van der Aalst, W. (2016). Data science in action.

Van Der Maaten, L., Postma, E., Van den Herik, J., et al. (2009a). Dimensionality reduction: a comparative. volume 10.

Van Der Maaten, L., Postma, E., Van den Herik, J., et al. (2009b). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71).

Vanschoren, J. (2018). Meta-learning: A survey. *arXiv preprint arXiv:1810.03548*.

Vanschoren, J. (2019). Meta-learning. *Automated machine learning: methods, systems, challenges*, pages 35–61.

VANSCHOREN, J. (2019). Meta-learning. automated machine learning: methods, systems, challenges. pages 35–61.

Vilalta, R. and Drissi, Y. (2002). A perspective view and survey of meta-learning. *Artificial intelligence review*, 18:77–95.

Wang, D., Hu, L., Du, H., Liu, Y., Huang, J., Xu, Y., and Liu, J. (2020). Classification, experimental assessment, modeling methods and evaluation metrics of trombe walls. *Renewable and Sustainable Energy Reviews*, 124:109772.

Wang, J. X. (2021). Meta-learning in natural and artificial intelligence. *Current Opinion in Behavioral Sciences*, 38:90–95.

Yacouby, R. and Axman, D. (2020). Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. In *Proceedings of the first workshop on evaluation and comparison of NLP systems*, pages 79–91.

Ying, X. (2019). An overview of overfitting and its solutions. In *Journal of physics: Conference series*, volume 1168, page 022022.

Younas, M. (2019). Research challenges of big data. *Service Oriented Computing and Applications*, 13:105–107.

ZAR, J. H. (1972). Significance testing of the spearman rank correlation coefficient. journal of the american statistical association. 67(339):578–580.

ZAR, J. H. (2005). Spearman rank correlation. encyclopedia of biostatistics. 7.

ZBRAZDIL, Pavel B.; SOARES, C. D. C. J. P. (2003). Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. machine learning. 50:251–277.

ZHANG, T. (2003). Leave-one-out bounds for kernel methods. neural computation. 13(6):1397–1437.

Zhou, Z.-H. (2021). Machine learning.