UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

João Gabriel Viana Hirasawa

APLICAÇÃO DE MÉTODOS DE REDUÇÃO DE DIMENSIONALIDADE NÃO LINEARES EM CLASSIFICADORES PARAMÉTRICOS E NÃO PARAMÉTRICOS

São Carlos, SP

2023

UNIVERSIDADE FEDERAL DE SÃO CARLOS
CENTRO DE CIÊNCIAS EXATAS E DE TECNOLOGIA
DEPARTAMENTO DE COMPUTAÇÃO

João Gabriel Viana Hirasawa

APLICAÇÃO DE MÉTODOS DE REDUÇÃO DE DIMENSIONALIDADE NÃO LINEARES EM CLASSIFICADORES PARAMÉTRICOS E NÃO PARAMÉTRICOS

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia de Computação da Universidade Federal de São Carlos, como requisito parcial para a obtenção do título de Bacharel em Engenharia de Computação.

Orientador: Prof. Dr. Alexandre Luís Magalhães Levada

São Carlos, SP
2023

Dedico este trabalho ao meu gatinho, Tico, que me ensinou o que é o amor.
E o meu amor por ele é infinito.

Agradecimentos

Agradeço à minha mãe, Rosângela, e ao meu pai, Paulo, pelo apoio em todas as etapas de minha vida. Aos meus irmãos, Ana e André, que provavelmente são os que mais entendem, de fato, a odisseia que foram estes últimos anos.

Também agradeço às outras pessoas da minha família, em especial à minha *obaa-chan*, vovó Masako, e minha prima Ana Cláudia, que foram e sempre serão muito amadas (*in memoriam*).

Agradeço a todas as pessoas que conheci e de quem me aproximei nos últimos anos, dentro e fora da faculdade. Graças a todes, a vida tem mais cor, e a graduação se tornou a melhor época da minha vida. São tantos nomes que eu desejaria citar, mas isso resultaria num texto do tamanho de outra monografia.

Ao meu orientador, Prof. Dr. Alexandre Levada, por sua paciência e generosidade em me guiar neste caminho de pedras. E aos outros professores que, pela magia da didática e do amor ao ensino, ajudaram-me a moldar o felicíssimo profissional que sou hoje.

E agradeço ao Tico, meu gatinho e irmãozinho felino, que me ajuda a encontrar as forças para enfrentar qualquer desafio.

Por último, mas não menos importante, agradeço a todas as pessoas não citadas aqui mas que também geraram algum impacto positivo na minha vida.

"– Aqui, tudo o que temos são pequenos fragmentos de tempo em que algo realmente faz sentido.

– Então, eu irei apreciar esses pequenos fragmentos de tempo."

(Tudo em Todo Lugar ao Mesmo Tempo, 2022)

Resumo

Muitos dos dados coletados e utilizados nas aplicações de aprendizado de máquina estão estruturados em conjuntos de alta dimensionalidade. Imagens, documentos de texto e dados de sensores são alguns exemplos de dados coletados o tempo todo, e cujo número de atributos pode ultrapassar facilmente a quantidade de amostras no conjunto. Como consequência, a maldição da dimensionalidade torna pertinente o estudo de meios para mitigar seus efeitos em modelos que utilizam esses conjuntos de dados de alta dimensionalidade. Uma solução para lidar com isso são os métodos de redução de dimensionalidade, que buscam gerar representações com um número mais palpável de dimensões, minimizando a perda de informação. Dessa forma, o uso de tais métodos dentro do aprendizado de máquina se torna um campo com potencial, à medida que simplificam a estrutura dos dados que alimentam os modelos. Este trabalho teve como objetivo avaliar o uso de diferentes métodos de redução de dimensionalidade não lineares junto a modelos paramétricos e não paramétricos em tarefas de classificação. Foram utilizados o UMAP e o PaCMAP em conjuntos de dados com alta dimensionalidade, disponíveis na plataforma do OpenML, e foi avaliado o desempenho de classificação dos modelos Quadratic Discriminant Analysis (QDA), Gaussian Naive Bayes, k-NN e XGBoost. Os resultados obtidos mostram uma melhora de desempenho para os modelos paramétricos, principalmente com o uso da implementação supervisionada do UMAP. Apesar de não terem sido tão efetivos num modelo mais robusto e pesado, o XGBoost, o uso dos métodos representou uma melhora no tempo de execução do modelo, que indica uma oportunidade de aplicação e estudo nessas situações.

Palavras-chave: Redução de dimensionalidade. Métodos não lineares de redução de dimensionalidade. Aprendizado de máquina supervisionado. Tarefas de classificação. Modelos paramétricos e não paramétricos. Maldição da dimensionalidade. UMAP. PaCMAP.

Abstract

Much of the data collected and used in machine learning applications are structured in high-dimensional datasets. Images, text documents, and sensor data are some examples of data collected all the time, in which the number of attributes can easily surpass the quantity of samples in the dataset. Consequently, the curse of dimensionality turns pertinent the study of means to mitigate its effects on models using high-dimensional datasets. A solution to deal with this is through dimensionality reduction methods, which aim to construct representations with a more manageable dimensionality, while minimizing information loss. Thus, the use of such methods in machine learning becomes a field with potential, as they simplify the structure of the data used by the models. The purpose of this work is to evaluate the use of different non-linear dimensionality reduction methods alongside parametric and non-parametric models in classification tasks. UMAP and PaCMAP were applied on high-dimensional datasets available on OpenML, and the classification performance of Quadratic Discriminant Analysis (QDA), Gaussian Naive Bayes, k-NN, and XGBoost was measured. The results demonstrate a performance improvement for parametric models, specially with the use of supervised UMAP. Although not as effective in XGBoost, a more robust and heavier model, the use of the methods showed an improvement in the model's execution time, indicating an opportunity for the application and further study of these situations.

Keywords: Dimensionality reduction. Nonlinear dimensionality reduction techniques. Supervised machine learning. Classification tasks. Parametric and nonparametric models. Curse of dimensionality. UMAP. PaCMAP.

Lista de ilustrações

Lista de tabelas

Sumário

1 Introdução

Os avanços em aprendizado de máquina têm trazido, a cada ano, novos modelos e aplicações para o grande número de dados sendo coletado no mundo. No entanto, desses avanços também surgem desafios significativos, como a interpretação dos resultados obtidos, o custo computacional e a crescente complexidade dos conjuntos de dados. Em particular, o aumento na dimensionalidade dos dados é um problema crucial, resultante não apenas do aumento no número de observações, mas também do incremento no número de atributos.

A alta dimensionalidade de um conjunto de dados pode prejudicar o desempenho dos modelos, tornando-os mais complexos e dependentes dos dados de treino. Além disso, um grande número de atributos pode indicar a presença de atributos redundantes ou insignificantes (Anowar; Sadaoui; Selim, 2021). O fenômeno relacionado é chamado de maldição da dimensionalidade, e seus efeitos ficam evidentes na maioria das tarefas de otimização e modelos probabilísticos (Binois; Wycoff, 2022; Köppen, 2000).

Nesse contexto, surgem métodos de redução de dimensionalidade, que reduzem os conjuntos de dados a um número de dimensões que seja adequado à situação em que eles serão aplicados. Esses métodos são muito utilizados em visualizações de conjuntos de alta dimensionalidade, como representações de texto em vetores (*embeddings*), sequências de RNA ou imagens, podendo ajudar a encontrar agrupamentos úteis nos dados (Smilkov et al., 2016; Becht et al., 2019; Brown et al., 2023).

Uma consequência da redução de dimensionalidade é a simplificação da estrutura dos dados, mantendo as informações que sejam mais relevantes, podendo trazer diversos ganhos computacionais sem perda de assertividade (Wang; Carreira-Perpinan, 2014). Alguns ganhos observados na literatura são a redução do espaço necessário de armazenamento e do tempo de execução, além de ser uma solução para a interpretabilidade de parâmetros dos modelos (Wang et al., 2021; Himeur et al., 2023).

A crescente quantidade de aplicações com dados de alta dimensionalidade torna a redução de dimensionalidade uma estratégia utilizada em muitas áreas. Por exemplo, seu uso pode ser observado na automação predial (Himeur et al., 2023), bioinformática (Belkina et al., 2019; Becht et al., 2019; Bagger; Kinalis; Rapin, 2019; Platzer, 2013), segurança da informação (Salo; Nassif; Essex, 2019), no sensoriamento remoto (Du et al., 2021), reconhecimento facial (Belhumeur; Hespanha; Kriegman, 1997) e desenvolvimento de soluções de aprendizado de máquina interpretáveis (Rudin et al., 2022).

Muitos algoritmos de redução de dimensionalidade não lineares têm sido propostos nos últimos anos, dentre os quais alguns notáveis são t-Stochastic Neighbor Embedding

(t-SNE), Uniform Manifold Approximation and Projection (UMAP), Pairwise Controlled Manifold Approximation (PaCMAP), TriMap, dentre outros (Maaten; Hinton, 2008; McInnes et al., 2018; Wang et al., 2021; Amid; Warmuth, 2019). Sendo métodos de extração de atributos (em contraste com seleção de atributos), esses algoritmos possuem a limitação de causar uma perda de significado do conjunto de dados original (Jia et al., 2022).

A redução de dimensionalidade tem sido amplamente estudada, com trabalhos como os de Wang et al. (2021) e McInnes et al. (2018), que exploram o uso dos métodos na criação de representações visuais. No entanto, o uso desses métodos com redução para mais de 2 ou 3 dimensões ainda é pouco explorado na literatura. Os estudos sobre métodos de redução de dimensionalidade se concentram na comparação direta entre eles, priorizando a avaliação de métricas específicas, como preservação de estrutura local ou global. O uso de modelos e avaliação de suas acurácias, como do k-NN e do SVM no trabalho de Wang et al. (2021), serve apenas a esse fim, sem ser pensada a aplicação dos modelos em si.

Nesse contexto, surge a oportunidade de oferecer uma nova perspectiva na avaliação desses métodos. Este trabalho visa explorar o uso da redução de dimensionalidade além da visualização, investigando sua eficácia em tarefas de aprendizado de máquina com uma metodologia sistemática de testes.

1.1 Objetivos

1.1.1 Objetivo geral

O objetivo deste trabalho é avaliar o uso de métodos de redução de dimensionalidade junto a classificadores em conjuntos de dados de alta dimensionalidade. Sendo assim, a hipótese estudada é que a aplicação da redução de dimensionalidade antes da aplicação do modelo melhore o desempenho final da classificação, com base nas métricas de avaliação definidas.

1.1.2 Objetivos específicos

Os objetivos específicos podem ser descritos como:

- Realizar uma revisão dos problemas advindos da maldição da dimensionalidade e os métodos de redução considerados estado da arte;
- Avaliar soluções para melhorar o desempenho de classificação e de tempo de execução no uso de modelos em conjuntos de dados com alta dimensionalidade; e

- Aplicar testes estatísticos para comparar os resultados obtidos da aplicação de métodos de redução de dimensionalidade não lineares em problemas de classificação.

1.2 Organização do trabalho

Este trabalho está organizado em cinco capítulos e um apêndice. No presente Capítulo, 1, foi apresentada uma contextualização para a proposta do trabalho, além de seus objetivos e justificativa. O Capítulo 2 traz uma revisão da fundamentação teórica utilizada no trabalho. No Capítulo 3 é descrita a metodologia utilizada para realização dos experimentos, cujos detalhes de realização e resultados estão resumidos no Capítulo 4 e detalhados no Apêndice A. Por fim, o Capítulo 5 traz as conclusões obtidas com o trabalho e propostas de trabalhos futuros.

2 Fundamentação Teórica

Este capítulo traz uma revisão dos principais conceitos utilizados neste trabalho, sendo esses o aprendizado de máquina e a redução de dimensionalidade, além dos modelos e algoritmos utilizados.

2.1 Aprendizado de máquina

O aprendizado de máquina (AM) é uma área de pesquisa fortemente relacionada com os campos de inteligência artificial, estatística e matemática. De um modo geral, o AM pode ser definido como o processo de "induzir uma função ou hipótese capaz de resolver um problema a partir de dados que representam instâncias do problema a ser resolvido" (Faceli et al., 2011, p. 3). Para Russell et al. (2022, p. 669), o AM ocorre quando "um computador observa os dados, constrói um modelo com base neles e usa esse modelo tanto como hipótese sobre o mundo quanto como um software capaz de resolver problemas". Apesar de algumas diferenças de escrita e abstração das duas definições, ambas enfatizam o uso de dados para modelar e solucionar algum problema como um aspecto essencial do AM.

A área de AM evoluiu consideravelmente ao longo das últimas décadas, proporcionando não apenas escalabilidade diante da crescente quantidade de informações coletadas a cada dia, como também permitindo testar e validar hipóteses rapidamente, tomando um papel essencial no desenvolvimento de muitos softwares. Essa evolução de técnicas e de seu papel mostra a capacidade do AM de oferecer novas soluções eficazes para uma variedade de desafios em diferentes domínios.

2.1.1 Aprendizado supervisionado

O aprendizado supervisionado é uma subárea do aprendizado de máquina, na qual se utiliza de rótulos dos dados para construir os modelos. O treino de um modelo com aprendizado supervisionado pode ser visto como tentar induzir uma função que se assemelha à função original de um conjunto de observações de entrada e saída. Por meio de um algoritmo de otimização, essa função aprendida é ajustada com base na diferença das saídas originais e as saídas produzidas. Com esse aprendizado por exemplo, espera-se que as saídas do modelo aprendido se aproximem das saídas originais e que esse modelo possa generalizar para novas entradas (Hastie; Tibshirani; Friedman, 2009, p. 29).

Ainda, podemos separar os problemas de aprendizado supervisionado como problemas de regressão ou classificação. Na regressão, tem-se um modelo cuja saída é um valor

real, enquanto a classificação gera uma saída dentro de um conjunto discreto, ou classes.

2.1.2 Modelos paramétricos

Os modelos paramétricos são um tipo de modelo nos quais é utilizada alguma premissa sobre a distribuição estatística dos dados. Com isso, objetivo de um modelo paramétrico é estimar um conjunto fixo de parâmetros relacionados a essa distribuição, a partir dos exemplos. Isso implica em uma restrição do aprendizado sobre os conjuntos de dados (Russell et al., 2022, p. 704).

2.1.2.1 Quadratic Discriminant Analysis

O Quadratic Discriminant Analysis (QDA) é uma generalização do Linear Discriminant Analysis (LDA), um método de análise discriminante baseado em combinações lineares de atributos. O objetivo dessa classe de métodos é aprender uma representação do espaço que separe as classes o máximo possível com a maior distribuição espacial possível (Ghojogh et al., 2023, p. 181-182).

Ao aplicar a análise discriminante, a distribuição dos dados é assumida como gaussiana ou normal, e utiliza-se como base o Teorema de Bayes. No QDA, as classes ficam separadas por fronteiras quadráticas e essa representação é utilizada para gerar o resultado da classificação.

2.1.2.2 Naive Bayes

O Naive Bayes é uma classe de modelos baseados na teoria bayesiana, que se apoiam no uso de probabilidades condicionais para determinar a saída do modelo. O termo *naive*, ou ingênuo em português, se refere à pressuposição do modelo de que os atributos do conjunto de dados são condicionalmente independentes um do outro, mesmo quando não for o caso (Russell et al., 2022, p. 420).

A regra principal utilizada no Naive Bayes é que a probabilidade condicional de determinada classe y dado um conjunto de atributos $x_1, \cdots, x_n$ é diretamente proporcional à regra de probabilidade conjunta da Equação 2.1. Essa probabilidade é utilizada para decidir qual é a classe, dada uma entrada.

$$P(y \mid x_1, \cdots, x_n) \propto P(y) \prod^{n} P(x_i \mid y) \tag{2.1}$$

Ainda, é possível assumir a forma da distribuição de probabilidade de $P(x_i \mid y)$. Quando se assume que a distribuição é normal, tem-se o Gaussian Naive Bayes, que pode ser utilizado com atributos de valores contínuos. Outros exemplos de distribuições utilizadas são a multinomial, de Bernoulli, ou mesmo sendo possível aproximar a distribuição com otimização não paramétrica (Hastie; Tibshirani; Friedman, 2009, p. 210).

2.1.3 Modelos não paramétricos

Os modelos não paramétricos não são definidos ou restritos por um conjunto fixo de parâmetros, sendo aprendidos sem assumir a distribuição de probabilidade do problema (Russell et al., 2022, p. 704).

2.1.3.1 *k*-Nearest Neighbors

O modelo de k-Nearest Neighbors (k-NN ou k-vizinhos mais próximos) (Fix; Hodges, 1989) consiste em utilizar uma métrica de distância para conectar uma amostra desconhecida a seus k vizinhos mais próximos. As classes desses vizinhos são utilizadas para determinar a classe da amostra.

Para o cálculo da proximidade no k-NN, diversas métricas de distância podem ser empregadas, incluindo a distância Euclidiana, de Manhattan, de cosseno, haversine, dentre outras. O valor de k ótimo pode variar de problema para problema e é um parâmetro que pode ser otimizado para melhorar o desempenho do modelo (Faceli et al., 2011, p. 62).

2.1.3.2 XGBoost

O XGBoost (Chen; Guestrin, 2016), sigla para *eXtreme Gradient Boosting*, é um método de *boosting* de árvores de decisão, que otimiza uma combinação de múltiplos modelos fracos para construir um modelo robusto, muito similar a um outro método de combinação de modelos, Random Forests (Ho, 1995).

O bom desempenho do XGBoost, além de sua escalabilidade, podendo ser implementado em ambientes distribuídos como Hadoop ou Spark, o torna muito utilizado em competições de ciência de dados e soluções empresariais (Chen; Guestrin, 2016).

2.1.4 Maldição da dimensionalidade

Um fenômeno muito observado em aplicações de AM é o desempenho de um modelo melhorar conforme aumenta-se a dimensionalidade do conjunto, até atingir um pico, e então piorar (Jia et al., 2022). A geometria em espaços de alta dimensionalidade se comporta diferente de espaços com poucas dimensões (Köppen, 2000), o que faz surgir problemas de difícil otimização.

A quantidade de dados necessários para treinar um modelo cresce exponencialmente com a dimensionalidade do conjunto: para manter o mesmo desempenho, se 10 amostras são suficientes para um modelo de 1 dimensão, seriam necessárias 100 para 2 dimensões e 1000 para 3 dimensões (Verleysen; François, 2005). Isso se mostra um obstáculo para conjuntos de dados com alta dimensionalidade e pouca quantidade de amostras, como são muitos conjuntos de imagens, por exemplo.

A redução de dimensionalidade, portanto, pode ter um papel em mitigar esses efeitos e tornar possível a modelagem mesmo em cenários mais limitados. Outras estratégias também têm sido pesquisadas, como, mais recentemente, o uso de aprendizado profundo se mostrou eficaz para tratar a maldição da dimensionalidade em algumas áreas (Blechschmidt; Ernst, 2021; Poggio et al., 2017).

2.2 Redução de dimensionalidade

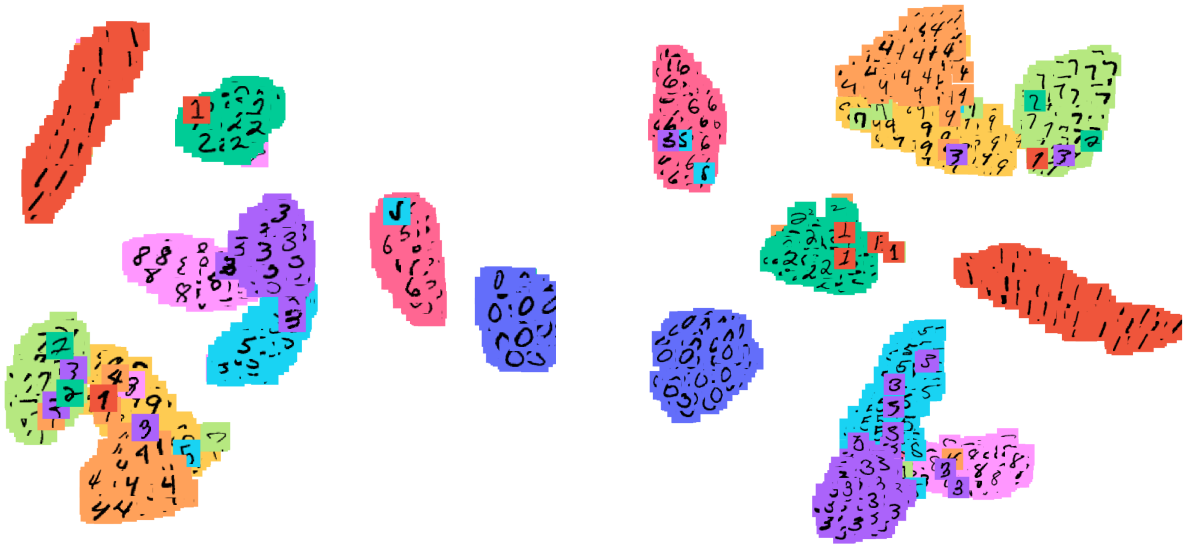
A redução de dimensionalidade, muitas vezes descrita como aprendizado de variedades (*manifold learning*), é o processo de reduzir o número de dimensões de um conjunto de dados, mapeando o espaço original para um espaço menor. O intuito é que essa representação resultante mantenha o quanto for possível da estrutura original dos dados.

O grande aumento da dimensionalidade dos dados, como mencionado anteriormente, fez com que a redução de dimensionalidade se tornasse uma parte muito importante da área de dados, tendo aplicações em diversos campos, incluindo visão computacional, bioinformática, visualização de dados, dentre outras.

A Figura 1 mostra o uso de dois métodos de redução de dimensionalidade no conjunto de dados MNIST, cuja dimensionalidade original é de 784, representando os *pixels* de imagens de tamanho 28×28 de dígitos escritos à mão. Ao reduzir para 2 dimensões, é possível visualizar a distribuição dos dígitos e os agrupamentos formados pelos métodos ao reconhecerem padrões de escrita entre os números.

Figura 1 – Visualização de dígitos do conjunto de dados MNIST, reduzidos a 2 dimensões com métodos de redução de dimensionalidade não lineares.

(a) Redução com o UMAP. **(b)** Redução com o PaCMAP.



Fonte: Elaborado pelo autor.

2.2.1 Dimensionalidade intrínseca

Os conjuntos de dados com alta dimensionalidade podem possuir uma dimensionalidade intrínseca muito menor do que a que estão estruturados os dados. A hipótese de variedades indica que as observações estão dispostas ou próximas de uma variedade de dimensionalidade d, que está embutida em um espaço de dimensionalidade D, sendo $d < D$.

Um exemplo de variedades de baixa dimensionalidade encontradas em espaços de alta dimensionalidade são dados de imagens de objetos 3D, com iluminação e ângulos variados (Belhumeur; Hespanha; Kriegman, 1997; Fefferman; Mitter; Narayanan, 2016). Ainda, os dados podem estar dispostos não apenas em uma variedade, como em uma união disjunta de múltiplas variedades, e essa disposição pode variar entre as distintas classes do conjunto de classificação (Brown et al., 2023).

O aprendizado de variedades tem como objetivo encontrar essa dimensionalidade intrínseca dos conjuntos de dados. A partir dela, os dados podem ser descritos por um conjunto menor de variáveis, reduzindo efetivamente a dimensionalidade. Sendo assim, redução de dimensionalidade é benéfica por remover a informação redundante dos dados (Lotlikar; Kothari, 2000; Jia et al., 2022; Himeur et al., 2023).

2.2.2 Classificação de métodos de redução de dimensionalidade

2.2.2.1 Linear e não linear

Os métodos de redução de dimensionalidade lineares aplicam apenas transformações lineares nos dados. Sendo assim, assume-se que a variedade intrínseca é, ao menos, aproximadamente linear. Alguns algoritmos lineares populares são Principal Component Analysis (PCA), Singular Value Decomposition (SVD) e Non-Negative Matrix Factorization (NMF) (Pearson, 1901; Golub; Reinsch, 1970; Lee; Seung, 1999).

Por outro lado, os métodos não lineares conseguem encontrar estruturas não-lineares nos dados, ou seja, podem lidar com dados mais complexos, presentes em muitas aplicações reais (Ayesha; Hanif; Talib, 2020). Como métodos não lineares, temos, dentre outros, Laplacian Eigenmaps, Kernel PCA, t-SNE, UMAP e PaCMAP (Belkin; Niyogi, 2001; Schölkopf; Smola; Müller, 1997; Maaten; Hinton, 2008; McInnes et al., 2018; Wang et al., 2021).

2.2.2.2 Supervisionado e não supervisionado

A ideia principal dos métodos supervisionados é utilizar as rotulações dos dados para reduzir a variância para amostras da mesma classe e aumentá-la entre diferentes classes. Ainda, técnicas de redução de dimensionalidade supervisionadas podem ser mais adequadas para uso em problemas de classificação ou regressão (Hajderanj; Weheliye;

Chen, 2019). O LDA, apresentado anteriormente, é um dos primeiros métodos de redução de dimensionalidade supervisionados e serviu como base para métodos futuros (Ghojogh et al., 2023, p. 181-182).

Os métodos não supervisionados não utilizam informações de classe para gerar a representação. Existem diversas formas de realizar essa redução, sendo uma das principais estratégias a de tentar aproximar as relações dos pontos no espaço original para o espaço de representação. Dessa forma, pontos similares se encontrarão próximos e os dissimilares distantes. O Locally Linear Embedding (LLE) segue essencialmente essa lógica, dentre muitos outros métodos (Saul; Roweis, 2003; Ghojogh et al., 2023, p. 207-208).

Muitos métodos originalmente não supervisionados também possuem versões supervisionadas, como o PCA (Barshan et al., 2011), UMAP (Sainburg; McInnes; Gentner, 2021), e mesmo o LLE (Ridder et al., 2003).

2.2.2.3 Estrutura local e global

Os métodos de estrutura local tendem a preservar vizinhanças, ou seja, relações de distância entre pontos vizinhos, resultando em agrupamentos mais precisos. A preservação da estrutura local se mostra mais efetiva para tarefas como identificação de agrupamentos ou de conjuntos (Xia et al., 2022). Alguns métodos desta categoria são Laplacian Eigenmaps, Locally Linear Embedding (LLE), t-SNE, UMAP e PaCMAP (Belkin; Niyogi, 2001; Roweis; Saul, 2000; Maaten; Hinton, 2008; McInnes et al., 2018; Wang et al., 2021).

Em contrapartida, os métodos de estrutura global tendem a preservar as relações de distância entre todos os pontos, o que resulta em uma distanciação mais adequada entre grupos (Wang et al., 2021). Como exemplos de métodos de estrutura global, tem-se Multi-Dimensional Scaling (MDS), Kernel PCA e ISOMAP (Bronstein; Bronstein; Kimmel, 2006; Schölkopf; Smola; Müller, 1997; Tenenbaum; Silva; Langford, 2000). O PaCMAP também pode ser considerado um método de estrutura global, já que foi pensado para manter um balanço entre preservação da estrutura local e global (Wang et al., 2021).

2.3 Algoritmos de redução de dimensionalidade

Esta seção visa detalhar o funcionamento dos algoritmos de redução de dimensionalidade utilizados neste trabalho.

2.3.1 Notação

Considere um conjunto de dados $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, com $\mathbf{X} \in \mathbf{R}^{N \times P}$, contendo N observações de P dimensões. Então, um algoritmo de redução de dimensionalidade tem

o objetivo de mapear $\mathbf{X}$ a uma representação $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_N\}$, com $\mathbf{Y} \in \mathbf{R}^{N \times Q}$, em um número menor de dimensões Q.

Idealmente, o valor de Q equivale à dimensionalidade intrínseca do conjunto. Geralmente, essa dimensionalidade não é conhecida, mas podem ser inferidos valores que possam ser razoáveis para a redução de dimensionalidade (Maaten; Postma; Herik, 2007).

2.3.2 UMAP

O algoritmo UMAP, sigla para *Uniform Manifold Approximation and Projection*, foi proposto por McInnes et al. (2018), como um algoritmo de redução de dimensionalidade não linear. Sua origem vem de resultados obtidos na aplicação de conceitos de topologia e teoria de categorias.

O UMAP possui duas partes de funcionamento, as quais estão descritas abaixo, com base no trabalho de McInnes et al. (2018).

Parte I. Construção do grafo de vizinhanças. Para cada amostra $\mathbf{x}_i$ são calculados os seus k vizinhos mais próximos $\{\mathbf{x}_{i_1}, \cdots, \mathbf{x}_{i_k}\}$, com base em uma métrica de similaridade d.

Em cada ponto $\mathbf{x}_i$, são calculados dois valores, ρ_i e σ_i. Primeiramente, o valor de ρ_i é definido com base na função de distância, como pode ser visto na Equação 2.2.

$$\rho_i = \min \Big\{ d(\mathbf{x}_i, \mathbf{x}_{i_j}) \mid 1 \leq j \leq k, d(\mathbf{x}_i,\ \mathbf{x}_{i_j}) > 0 \Big\} \tag{2.2}$$

A seguir, o valor de σ_i é calculado resolvendo a Equação 2.3.

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(\mathbf{x}_i, \mathbf{x}_{i_j}) - \rho_i)}{\sigma_i} \right) = \log_2(k) \tag{2.3}$$

Então, é construído um grafo direcionado ponderado, conectando como arestas cada amostra $\mathbf{x}_i$ a seus k vizinhos. O peso das arestas direcionadas, então, segue a Equação 2.4.

$$w((\mathbf{x}_i, \mathbf{x}_{i_j})) = \exp\left(\frac{-\max(0, d(\mathbf{x}_i, \mathbf{x}_{i_j}) - \rho_i)}{\sigma_i} \right) \tag{2.4}$$

Finalmente, da matriz de adjacência A desse grafo ponderado direcionado, é calculada uma matriz simétrica de adjacência B, de acordo com a Equação 2.5.

$$B = A + A^{\intercal} - A \circ A^{\intercal} \tag{2.5}$$

Assim, pode-se definir o grafo construído como o grafo ponderado não direcionado G, com matriz de adjacência B, cujos vértices são os pontos em $\mathbf{X}$.

Parte II. Otimização da estrutura do grafo. O UMAP utiliza um algoritmo de forças direcionadas para otimização do grafo. Neste tipo de algoritmo, os vértices são atraídos ou repelidos entre si, de acordo com as forças de atração e repulsão definidas, que são aplicadas nas arestas.

Para isso, é criado um grafo H da representação a ser construída $\mathbf{Y}$, semelhante a G, utilizando inicialização espectral ou aleatória. Os pesos das arestas do grafo H então são ajustados iterativamente pelas forças que incidem neles.

A força de atração entre dois vértices $\mathbf{y}_i$ e $\mathbf{y}_j$ no grafo H é determinada pela Equação 2.6, enquanto a força de repulsão é determinada pela Equação 2.7.

$$\frac{-2ab||\mathbf{y}_i - \mathbf{y}_j||_2^{2(b-1)}}{1 + ||\mathbf{y}_i - \mathbf{y}_j||_2^2} w((\mathbf{x}_i, \mathbf{x}_j))(\mathbf{y}_i - \mathbf{y}_j) \tag{2.6}$$

$$\frac{2b}{(\epsilon + ||\mathbf{y}_i - \mathbf{y}_j||_2^2)(1 + a||\mathbf{y}_i - \mathbf{y}_j||_2^{2b})} (1 - w((\mathbf{x}_i, \mathbf{x}_j)))(\mathbf{y}_i - \mathbf{y}_j) \tag{2.7}$$

Como pode ser observado, o peso de cada aresta em H (que representa $\mathbf{Y}$) depende também dos pontos em G (que representa $\mathbf{X}$). Isso é devido ao interesse de se aproximar H de G, passando a topologia capturada do grafo de alta dimensionalidade para o grafo de baixa dimensionalidade (McInnes et al., 2018).

O Algoritmo 1 descreve o funcionamento em alto nível do UMAP.

Algoritmo 1: UMAP simplificado.

Entrada: O conjunto de dados $\mathbf{X}$ e a dimensão desejada Q.
Saída: A representação $\mathbf{Y}$.
início
  $G \leftarrow$ Grafo de vizinhanças de $\mathbf{X}$.
  $H \leftarrow$ Inicialização espectral da representação Q-dimensional a partir de G.
  repita
    Atualize H com aplicação das forças e otimização por meio de gradiente descendente estocástico.
  até *atingir o número de épocas definido*;
  $\mathbf{Y} \leftarrow$ Representação de baixa dimensionalidade de $\mathbf{X}$, presente em H.
fim

O UMAP ainda possui alguns hiperparâmetros que podem ser otimizados para cada aplicação, dentre eles: o número de dimensões `Q`, às quais $\mathbf{X}$ será reduzido; o número de vizinhos `k`, para construção do grafo inicial; a distância mínima `min_dist` entre vizinhos no grafo da representação H; e o número de épocas de otimização `n_epochs`.

O número de vizinhos `k` altera a forma como as características das variedades são capturadas, representando um *tradeoff* entre capturar estruturas mais ou menos detalhadas. A distância mínima `min_dist`, por sua vez, determina a proximidade (compactação) entre os pontos na representação, podendo ser considerado um parâmetro essencialmente estético (McInnes et al., 2018). Apesar dos hiperparâmetros modificarem o comportamento e o resultado da representação final, não é possível escolher entre preservar estrutura local ou global no UMAP ajustando-os (Wang et al., 2021).

2.3.3 PaCMAP

Proposto por Wang et al. (2021), o PaCMAP, ou *Pairwise Controlled Manifold Approximation Projection*, também é um algoritmo de redução de dimensionalidade não linear, projetado pensando na preservação tanto de estrutura local quanto global. O algoritmo foi desenvolvido com conhecimento obtido a partir de observações empíricas do funcionamento de outros métodos de redução de dimensionalidade.

O PaCMAP, como definido por Wang et al. (2021), possui três fases de otimização, que ocorrem após a construção e inicialização dos grafos iniciais.

Parte I. Construção do grafo de vizinhanças e inicialização do grafo de representação. O PaCMAP realiza três tipos de pareamento nos vértices, classificando-os em pares vizinhos, pares meio próximos (*mid-near*) e pares distantes.

Os pares vizinhos são os k vizinhos mais próximos, selecionados de acordo com a função de distância da Equação 2.8. Considere σ_i a distância euclidiana média de um ponto $\mathbf{x}_i$ de seu quarto ao sexto vizinhos.

$$d_{ij}^2 = \frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma_i \sigma_j} \tag{2.8}$$

Os pares meio próximos são construídos pareando o segundo vizinho mais próximo de cada ponto $\mathbf{x}_i$ ao próprio ponto, a partir de amostras aleatórias de pontos de $\mathbf{X}$.

Por fim, os pares distantes são criados a partir de pontos mais distantes do que os vizinhos e pares meio próximos de cada $\mathbf{x}_i$.

Parte II. Otimização do grafo de representação. Diferentemente do UMAP, o PaCMAP funciona otimizando uma função de custo, com o objetivo de aproximar

as estruturas locais e globais da representação de baixa dimensionalidade das encontradas no conjunto original.

Primeiramente, é criada a representação $\mathbf{Y}$, com inicialização com PCA ou aleatória. Com isso, define-se a distância $\tilde{d}_{ij}$ entre dois pontos da representação com a Equação 2.9.

$$\tilde{d}_{ij} = ||\mathbf{y}_i - \mathbf{y}_j||^2 + 1 \tag{2.9}$$

Cada tipo de pareamento recebe uma função de custo, definidas em 2.10, sendo Loss_{NB} para os pares vizinhos, Loss_{MN} para os pares meio próximos e Loss_{FP} para os pares distantes.

$$\text{Loss}_{\text{NB}} = \frac{\tilde{d}_{ij}}{10 + \tilde{d}_{ij}} \qquad \text{Loss}_{\text{MN}} = \frac{\tilde{d}_{ik}}{10000 + \tilde{d}_{ik}} \qquad \text{Loss}_{\text{FP}} = \frac{1}{1 + \tilde{d}_{il}} \tag{2.10}$$

Finalmente, a função de custo a ser otimizada é uma combinação de todas as funções de custo aplicadas nos pares, com pesos associados a cada uma. Essa função está disposta na Equação 2.11.

$$\text{Loss} = w_{\text{NB}} \sum_{\text{vizinhos}} \text{Loss}_{\text{NB}} + w_{\text{MN}} \sum_{\text{meio próximos}} \text{Loss}_{\text{MN}} + w_{\text{FP}} \sum_{\text{distantes}} \text{Loss}_{\text{FP}} \tag{2.11}$$

As três fases de otimização do PaCMAP são delimitadas pelo número da iteração, sendo definidos como padrão $\tau_1 = 1$, $\tau_2 = 101$ e $\tau_3 = 201$. Em cada fase, são atribuídos valores diferentes para os pesos w_{NB}, w_{MN} e w_{FP}.

Fase 1. $[\tau_1, \tau_2)$ **Maior peso de pares meio próximos.** É aplicado um alto peso às distâncias dos pares meio próximos, que vai decrescendo até que seja atingida a segunda fase.

O peso dos pares meios próximos segue a função da Equação 2.12.

$$w_{\text{MN}}(t) = 1000 \cdot \left(1 - \frac{t-1}{\tau_2 - 1}\right) + 3 \cdot \frac{t-1}{\tau_2 - 1} \tag{2.12}$$

Os outros pesos são definidos como $w_{\text{NB}} = 2$ e $w_{\text{FP}} = 1$.

Fase 2. $[\tau_2, \tau_3)$ **Pesos equivalentes de pares vizinhos e pares meio próximos.** A partir dessa fase, os pesos são constantes, nesse momento definidos como $w_{\text{NB}} = 3$, $w_{\text{MN}} = 3$ e $w_{\text{FP}} = 1$.

Fase 3. $[\tau_3, n_{\text{iterações}}]$ **Peso nulo para os pares meio próximos.** Nessa última fase, os pares meio próximos são ignorados e a função de custo utiliza apenas os pares vizinhos e pares distantes, sendo $w_{\text{NB}} = 1$, $w_{\text{MN}} = 0$ e $w_{\text{FP}} = 1$.

O funcionamento em alto nível do PaCMAP é descrito no Algoritmo 2.

Algoritmo 2: PaCMAP simplificado.

Entrada: O conjunto de dados $\mathbf{X}$ e a dimensão desejada Q.

Saída: A representação $\mathbf{Y}$.

início

Gere os conjuntos de pares vizinhos, meio próximos e distantes para os pontos de $\mathbf{X}$.

$\mathbf{Y} \leftarrow$ Inicialização da representação Q-dimensional a partir de $\mathbf{X}$.

repita

Ajuste os pesos w_{NB}, w_{MN} e w_{FP} com base no número da iteração.

Atualize $\mathbf{Y}$ com base na otimização da função de custo.

até *atingir o número de iterações definido*;

fim

Alguns hiperparâmetros também podem ser modificados, o principal sendo o número de vizinhos `k`. Esse parâmetro permite escolher uma representação mais assertiva em relação à estrutura local (com um `k` pequeno) ou uma representação com agrupamentos mais compactos e com menos detalhe de estrutura local (com um `k` grande). Além disso, a quantidade de pares meio próximos e distantes a serem gerados também é ajustável, o que também impacta na conservação de estruturas locais e globais.

3 Metodologia

Este capítulo tem como objetivo descrever a metodologia de experimentação utilizada neste trabalho, assim como os conjuntos de dados e algoritmos utilizados.

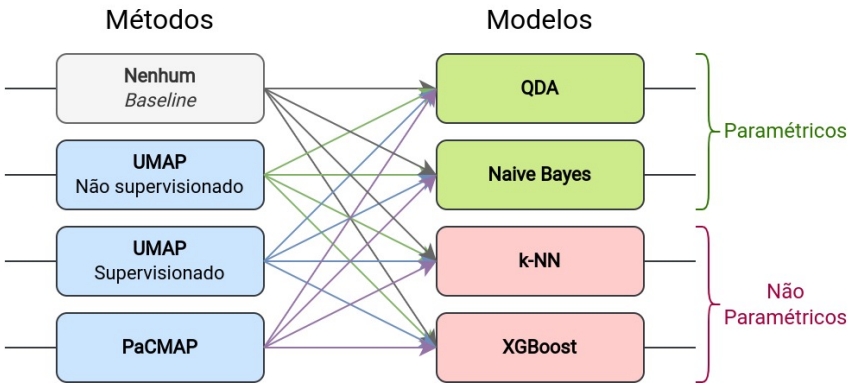
3.1 Algoritmos utilizados

O principal objeto de análise está na aplicação de estratégias de redução de dimensionalidade visando melhorar o desempenho dos modelos de classificação. Sendo assim, foram avaliados três métodos de redução com quatro modelos diferentes, mais um *baseline*, que representa o uso apenas dos modelos, sem redução prévia.

Os métodos de redução utilizados foram o UMAP não supervisionado, UMAP supervisionado e o PaCMAP. Os modelos utilizados foram o QDA e o Naive Bayes, para os paramétricos, e o k-NN e o XGBoost, para os não paramétricos.

A Figura 2 ilustra as combinações de métodos e modelos experimentadas.

Figura 2 – Diagrama representando as combinações de métodos e modelos.



Fonte: Elaborado pelo autor.

Os métodos de redução de dimensionalidade foram testados variando o número de componentes finais, com os valores `[10, 20, 30, 50, 80, 100]`. Para os outros parâmetros, foram utilizados os valores padrão das implementações de cada método e modelo. Os principais valores estão listados nas Tabelas 1 e 2.

Tabela 1 – Parâmetros padrão para cada método de redução de dimensionalidade.

Método	**Parâmetros**
UMAP	`n_neighbors=15, init='random', min_dist=0.1`
PaCMAP	`n_neighbors=10, init='random', MN_ratio=0.5, FP_ratio=2`

Tabela 2 – Parâmetros padrão para cada modelo.

Modelo	**Parâmetros**
QDA	`reg_param=0.0, tol=1.0e-4`
Naive Bayes	`var_smoothing=1e-9`
k-NN	`n_neighbors=5, weights='uniform', metric='minkowski', p=2`
XGBoost	`max_depth=None, n_estimators=100`

3.2 Validação cruzada

Para realizar a avaliação, foi utilizada a validação cruzada estratificada K-fold com $K = 5$. Neste tipo de validação, o conjunto de dados é inicialmente dividido em K subconjuntos (*folds*). Para cada *fold* k, o modelo é treinado utilizando os outros $K-1$ *folds*, e então avalia-se o desempenho do modelo em k. Ao final das K iterações, é calculada a média das métricas obtidas em cada *fold*.

O diferencial da validação estratificada é que tenta-se manter a proporção de classes por *fold*. Se o conjunto original possui 20% de amostras da classe A e 80% da classe B, idealmente cada *fold* também possuiria 20% de A e 80% de B. Sendo assim, é possível mitigar algum *overfitting* gerado por desbalanceamento de dados.

3.3 Métricas de avaliação

Em um problema de classificação com duas classes, estas podem ser definidas em classe positiva e classe negativa. Então, da predição, pode-se contar quantas amostras positivas e negativas foram classificadas corretamente (TP, ou *True Positives*, e TN, ou *True Negatives*) e quantas foram classificadas incorretamente (FP, ou *False Positives*, e FN, ou *False Negatives*).

A partir desses valores, é possível derivar duas métricas, a precisão e a revocação. A precisão (Equação 3.1) é a medida da proporção, entre todas as amostras que foram classificadas como positivas, das que são realmente positivas. A revocação (Equação 3.2) mede quanto das amostras realmente positivas foram classificadas como positivas (Faceli et al., 2011, p. 164-165).

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{3.1}$$

$$\text{Revocação} = \frac{TP}{TP + FN} \tag{3.2}$$

A principal métrica utilizada neste trabalho é o F_1 *score* (Equação 3.3), que se dá pela média harmônica da precisão e da revocação, possibilitando combinar as duas métricas em uma só.

$$F_1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}} \tag{3.3}$$

Nos conjuntos onde são observadas mais de duas classes, a estratégia utilizada para medir a performance com o F_1 *score* foi a *macro-average*, na qual os *scores* são calculados para cada classe e é realizada a média aritmética entre esses valores.

Além do F_1 *score*, também foram avaliados os tempos de execução para cada método e modelo, de acordo com a Equação 3.4. Desta forma, é possível avaliar também se a redução de dimensionalidade gera ou não um ganho de velocidade de treino dos modelos.

$$\text{Tempo de execução} = \text{Tempo}_{\text{Redução}} + \text{Tempo}_{\text{Modelo}} \tag{3.4}$$

3.4 Conjuntos de dados

Os conjuntos de dados foram selecionados com o critério de serem conjuntos que possam ocasionar problemas de dimensionalidade para os modelos, ou seja, nos quais o número de amostras não seja suficientemente grande para a quantidade de atributos dos dados. Para tal, foram selecionados 11 conjuntos para classificação, disponíveis para download no portal do OpenML[1], com número variado de atributos, amostras[2] e classes. A descrição dos conjuntos está na Tabela 3.

Tabela 3 – Relação de número de atributos, de amostras e de classes para cada conjunto de dados utilizado.

#	**Nome**	**Atributos**	**Amostras**	**Classes**
1	Speech	400	3686	2
2	har	561	2574	6
3	cnae-9	856	1080	9
4	micro-mass	1300	360	10
5	CIFAR_10	3072	6000	10
6	eating	6373	945	7
7	amazon-commerce-reviews	10000	1500	50
8	AP_Ovary_Uterus	10935	322	2
9	AP_Endometrium_Breast	10935	405	2
10	AP_Breast_Kidney	10935	604	2
11	OVA_Uterus	10935	1545	2

[1] Disponível em: https://openml.org/.

[2] Os conjuntos har e CIFAR_10 estão reduzidos a 25% e 10% de seus tamanhos originais, respectivamente, por meio de amostragem estratificada.

3.5 Pré-processamento

Foram realizadas duas etapas de pré-processamento nos conjuntos de dados, descritas a seguir:

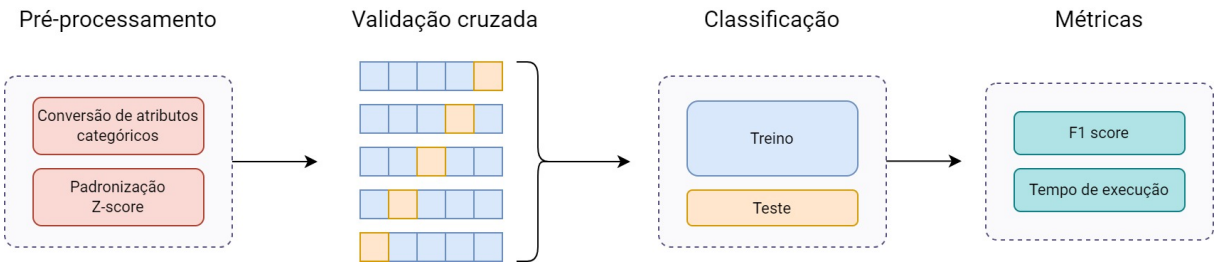
1. **Conversão de atributos categóricos em atributos binários**, gerando n atributos novos com valor 0 ou 1, para cada atributo categórico presente no conjunto com n valores distintos. O valor da amostra será 1 para o atributo correspondente ao valor original, e 0 para todos os outros novos atributos.

2. **Padronização dos valores de atributos com Z-score**, na qual cada valor é redimensionado para $x' = \frac{x - \mu}{\sigma}, \forall x \in \mathbf{X}$, fazendo-se uso da média e do desvio padrão do conjunto.

3.6 Arquitetura

Com base nas definições anteriores, a Figura 3 mostra um diagrama da arquitetura utilizada neste trabalho. No diagrama, estão dispostos os passos detalhados anteriormente neste capítulo, sendo que os classificadores treinados na etapa de Classificação são as combinações da Figura 2.

Figura 3 – Diagrama da arquitetura de experimentação adotada.



Fonte: Elaborado pelo autor.

3.7 Teste estatístico

O teste utilizado para avaliar o desempenho dos métodos neste trabalho é o proposto por Friedman (1937), um teste não paramétrico para avaliar diferenças em múltiplas amostragens entre vários grupos. Os grupos testados são cada um dos métodos, e as amostragens são seus resultados nos diferentes conjuntos de dados.

A hipótese a ser testada, então, pode ser definida como (Sprent; Smeeton, 2001, p. 219-222):

- **Hipótese $\mathbf{H_0}$**: não há diferença de distribuição dos resultados entre os métodos aplicados.
- **Hipótese $\mathbf{H_1}$**: pelo menos um dos métodos aplicados possui uma distribuição diferente de resultados.

No qual testa-se a hipótese $\mathbf{H_0}$ contra a hipótese $\mathbf{H_1}$.

Se a hipótese $\mathbf{H_0}$ for rejeitada, pode-se realizar um teste *post-hoc* para identificar quais métodos obtiveram resultados significantemente diferentes. O método utilizado neste trabalho é o proposto por Nemenyi (1963).

3.8 Ambiente computacional

Os experimentos foram executados em um computador com processador Intel Core i7-11800H @ 4.60GHz e 24GB de memória RAM. Para os métodos de redução e modelos foram utilizadas as seguintes versões das bibliotecas do Python `3.11`[3]: `umap-learn 0.5.4`, `pacmap 0.7.0`, `scikit-learn 1.3.1` e `xgboost 2.0.0`. Os testes estatísticos foram realizados com `numpy 1.25.2` e `scikit-posthocs 0.8.0`.

[3] Todas as bibliotecas estão disponíveis para download pelo Python Package Index (PyPI). Disponível em: https://pypi.org/.

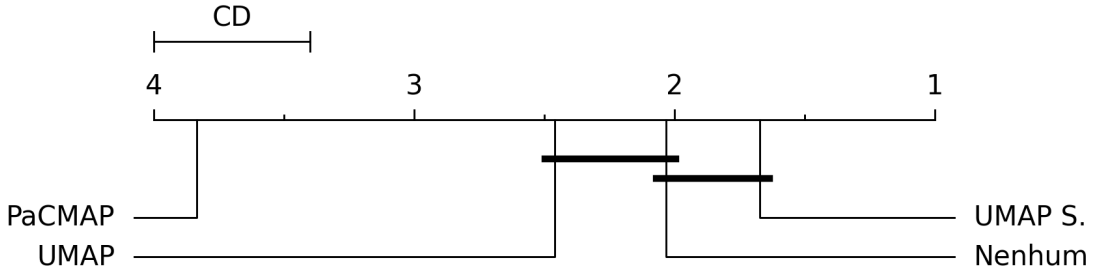
4 Experimentos e resultados

Este capítulo apresenta os resultados obtidos com os experimentos realizados seguindo a metodologia descrita no Capítulo 3. Para mais detalhes, os resultados completos obtidos podem ser visualizados no Apêndice A.

Assim como apresentado nas Figuras 2 e 3, cada combinação de método e modelo foi testada nos conjuntos de dados selecionados e foram coletadas as métricas definidas. As métricas finais para cada combinação foram as médias dos *folds*, tanto para o F1 score quanto para o tempo de execução.

O teste de Friedman foi aplicado nesses resultados e resultou num p-valor de $3,317 \cdot 10^{-21}$. Com um nível de significância $\alpha = 0,05$, pode-se rejeitar a hipótese nula $\mathbf{H_0}$ de que não há diferença de distribuição dos resultados. Portanto, foi realizado o teste *post-hoc* de Nemenyi para identificar quais grupos possuem diferença significante estatisticamente. A Figura 4 traz o diagrama de diferença crítica gerado com o teste.

Figura 4 – Diagrama de diferença crítica dos métodos aplicados.



Fonte: Elaborado pelo autor.

Do diagrama, pode-se observar que o UMAP supervisionado obteve o maior *rank* médio, porém sem diferença estatística da aplicação de nenhum método. O UMAP não supervisionado também não obteve diferença estatística da aplicação de nenhum método, mas obteve da versão supervisionada. O PaCMAP, por sua vez, teve o pior *rank* médio entre os métodos, com diferença estatística significante de todos os outros.

As seções 4.1 e 4.2 trazem análises do desempenho de F1 score e do tempo de execução total para cada método avaliado.

4.1 Avaliação de desempenho de classificação com F1 score

Os resultados para os modelos estão dispostos em quatro tabelas, cujas colunas representam os métodos utilizados. Os valores apresentados são os F1 scores médios de todos os números de dimensões testados (10, 20, 30, 50, 80, 100), junto com o desvio padrão.

O UMAP supervisionado foi o método que mais apresentou ganhos para o QDA (Tabela 4), trazendo aumentos significativos de desempenho nos conjuntos em que ele foi o melhor. O PaCMAP obteve o melhor desempenho no conjunto de menor dimensionalidade, porém piorou o resultado em sete dos onze conjuntos. Em dois conjuntos, o desempenho com redução de dimensionalidade foi pior do que o *baseline*. O QDA também foi o modelo que mais sofreu flutuações de desempenho com a variação do número de componentes, que piorou conforme esse número aumentava.

Tabela 4 – F1 score médio para cada método utilizado com o QDA.

#	**Nenhum**	**UMAP**	**UMAP S.**	**PaCMAP**
1	0,4958	0,4983 ± 0,0063	0,4958 ± 0,0000	**0,4990** ± 0,0080
2	0,3875	0,5324 ± 0,1822	**0,8730** ± 0,0677	0,4017 ± 0,0376
3	0,1111	0,3967 ± 0,2933	**0,6314** ± 0,0801	0,0567 ± 0,0087
4	0,0819	0,3691 ± 0,1094	**0,6679** ± 0,1291	0,0705 ± 0,0282
5	**0,1048**	0,0777 ± 0,0293	0,0333 ± 0,0253	0,0420 ± 0,0054
6	**0,1532**	0,1132 ± 0,0308	0,1248 ± 0,0313	0,1187 ± 0,0099
7	0,0181	**0,0313** ± 0,0307	0,0233 ± 0,0093	0,0062 ± 0,0022
8	0,5020	0,4882 ± 0,1238	**0,7355** ± 0,0058	0,4317 ± 0,0243
9	0,4101	0,5188 ± 0,1252	**0,7832** ± 0,1903	0,4452 ± 0,0307
10	0,5786	0,6694 ± 0,2131	**0,9422** ± 0,0297	0,3485 ± 0,0284
11	0,2836	0,4905 ± 0,0184	**0,6420** ± 0,1274	0,4791 ± 0,0000

Para o Naive Bayes (Tabela 5), o UMAP supervisionado continua sendo o método que melhor desempenhou, e o único a trazer melhoras sobre o *baseline*. Nesses conjuntos, o aumento de desempenho também foi significativo. No entanto, nos conjuntos em que o *baseline* foi melhor, houve uma piora alta causada pelo UMAP supervisionado. O uso do PaCMAP trouxe um desempenho um pouco melhor do que com o QDA, porém ainda ficou atrás do UMAP não supervisionado.

Tabela 5 – F1 score médio para cada método utilizado com o Naive Bayes.

#	Nenhum	UMAP	UMAP S.	PaCMAP
1	**0,6157**	0,5518 ± 0,0121	0,5606 ± 0,0172	0,4511 ± 0,0338
2	0,6597	0,8422 ± 0,0030	**0,9157** ± 0,0016	0,8341 ± 0,0025
3	**0,8470**	0,7669 ± 0,0058	0,8242 ± 0,0164	0,6819 ± 0,0102
4	**0,8822**	0,7082 ± 0,0145	0,7653 ± 0,0293	0,0431 ± 0,0072
5	**0,2734**	0,2008 ± 0,0029	0,2313 ± 0,0113	0,2003 ± 0,0051
6	0,0383	0,1354 ± 0,0040	**0,1548** ± 0,0127	0,1093 ± 0,0261
7	**0,5514**	0,0546 ± 0,0032	0,0934 ± 0,0112	0,0405 ± 0,0032
8	0,4859	0,6305 ± 0,0319	**0,7423** ± 0,0081	0,4812 ± 0,0189
9	0,7757	0,8582 ± 0,0063	**0,8950** ± 0,0031	0,4594 ± 0,0070
10	0,9244	0,9564 ± 0,0009	**0,9575** ± 0,0000	0,3744 ± 0,0364
11	0,6755	0,5986 ± 0,0162	**0,7335** ± 0,0023	0,4886 ± 0,0176

Com o *k*-NN (Tabela 6), o *baseline* passa a obter os melhores resultados para a maioria dos conjuntos, havendo apenas um caso em que o uso do UMAP melhorou o desempenho consideravelmente, subindo a métrica de 0,0239 para 0,1734 no conjunto 7. Novamente, o PaCMAP levou ao pior resultado em quase todos os conjuntos de dados.

Tabela 6 – F1 score médio para cada método utilizado com o k-NN.

#	Nenhum	UMAP	UMAP S.	PaCMAP
1	0,5912	**0,6056** ± 0,0130	0,5051 ± 0,0164	0,5082 ± 0,0082
2	0,9267	0,8960 ± 0,0047	**0,9277** ± 0,0003	0,8855 ± 0,0016
3	0,8319	0,7886 ± 0,0081	**0,8415** ± 0,0110	0,7140 ± 0,0084
4	**0,8214**	0,7670 ± 0,0159	0,8154 ± 0,0036	0,0620 ± 0,0211
5	**0,2465**	0,2160 ± 0,0049	0,2297 ± 0,0095	0,2120 ± 0,0038
6	**0,3689**	0,1662 ± 0,0131	0,1776 ± 0,0092	0,0940 ± 0,0201
7	0,0239	0,1339 ± 0,0051	**0,1734** ± 0,0085	0,0933 ± 0,0059
8	**0,7618**	0,6909 ± 0,0239	0,7370 ± 0,0029	0,5041 ± 0,0266
9	**0,9383**	0,8732 ± 0,0100	0,8979 ± 0,0025	0,4641 ± 0,0073
10	**0,9729**	0,9547 ± 0,0009	0,9575 ± 0,0000	0,4602 ± 0,0396
11	**0,7800**	0,6717 ± 0,0115	0,7324 ± 0,0029	0,4866 ± 0,0188

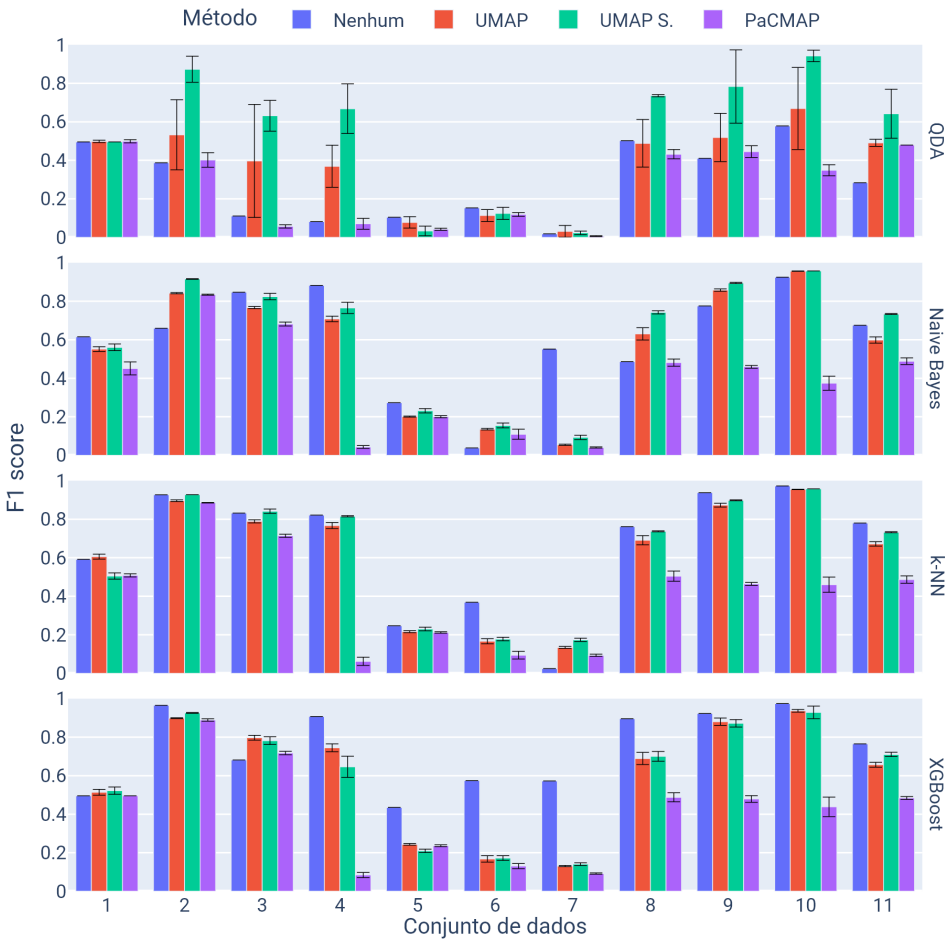
O XGBoost (Tabela 7), assim como o *k*-NN, não foi muito beneficiado pelo uso da redução, tendo apenas duas situações em que os métodos aumentaram a métrica, sendo uma delas um aumento substancial, cerca de 0,1158 a mais com o UMAP no conjunto 3.

Tabela 7 – F1 score médio para cada método utilizado com o XGBoost.

#	Nenhum	UMAP	UMAP S.	PaCMAP
1	0,4958	0,5138 ± 0,0153	**0,5223** ± 0,0191	0,4955 ± 0,0001
2	**0,9655**	0,8988 ± 0,0018	0,9260 ± 0,0019	0,8890 ± 0,0061
3	0,6813	**0,7971** ± 0,0127	0,7820 ± 0,0199	0,7175 ± 0,0087
4	**0,9078**	0,7448 ± 0,0206	0,6459 ± 0,0551	0,0841 ± 0,0136
5	**0,4352**	0,2433 ± 0,0043	0,2098 ± 0,0091	0,2364 ± 0,0048
6	**0,5749**	0,1678 ± 0,0169	0,1724 ± 0,0123	0,1306 ± 0,0133
7	**0,5728**	0,1306 ± 0,0032	0,1408 ± 0,0073	0,0919 ± 0,0033
8	**0,8961**	0,6893 ± 0,0316	0,7003 ± 0,0258	0,4878 ± 0,0233
9	**0,9235**	0,8806 ± 0,0189	0,8721 ± 0,0194	0,4791 ± 0,0176
10	**0,9746**	0,9367 ± 0,0069	0,9287 ± 0,0329	0,4383 ± 0,0508
11	**0,7655**	0,6569 ± 0,0126	0,7104 ± 0,0110	0,4842 ± 0,0076

A Figura 5 traz a média de F1 score nos experimentos para cada modelo, com os métodos representados por cores, permitindo visualizar as observações apontadas neste capítulo. Foram adicionadas barras de erro para representar a variação entre os resultados com diferentes números de dimensões.

Figura 5 – F1 score médio dos modelos nos conjuntos de dados.

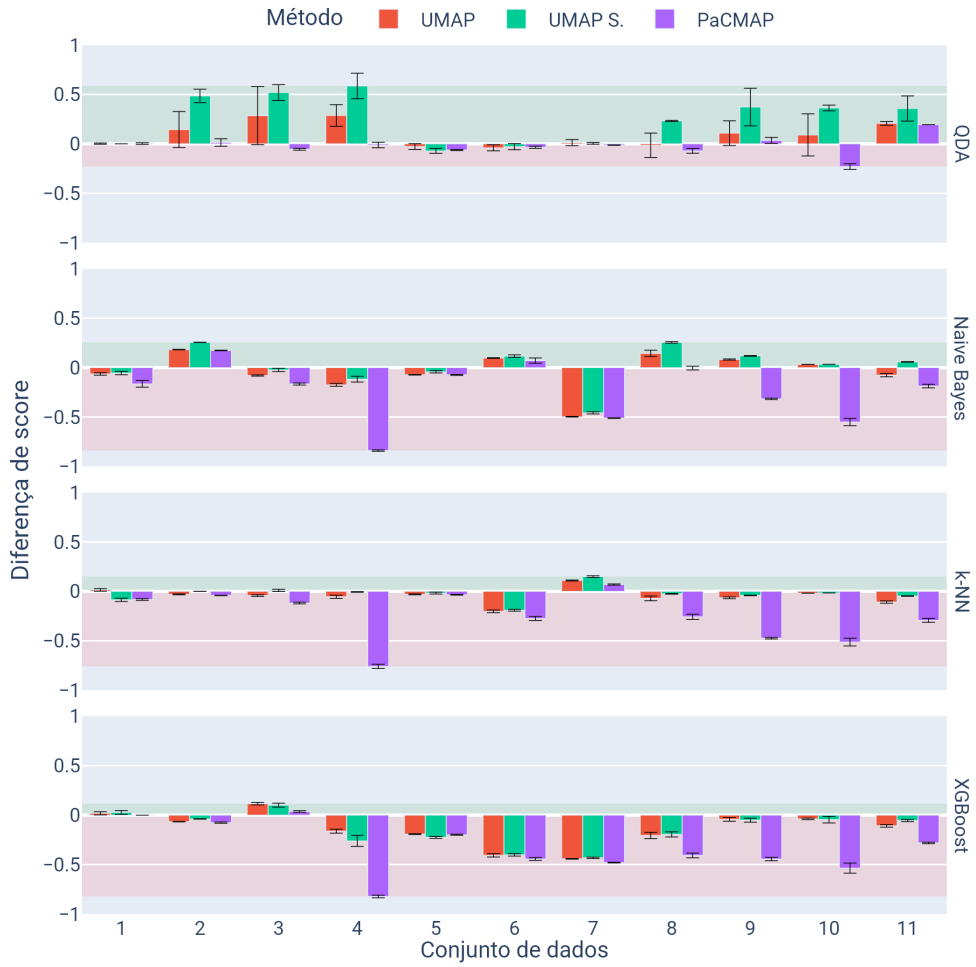


Fonte: Elaborado pelo autor.

Uma característica específica do QDA foi a grande variação dos resultados, gerando margens maiores de erro. Além disso, é possível notar a consistência do XGBoost em desempenhar melhor do que os métodos testados entre todos os conjuntos de dados, exceto nos conjuntos 1 e 3. Outro resultado melhor evidenciado pelo gráfico é a queda de desempenho causada pelo PaCMAP, como nos conjuntos 4 e 10.

Na Figura 6, está a diferença média de F1 score de cada método contra o *baseline*. Nela, é possível observar a melhora de desempenho para vários dos conjuntos de dados no QDA e no Naive Bayes, além da piora causada no desempenho do k-NN e do XGBoost.

Figura 6 – Diferença média de F1 score dos modelos nos conjuntos de dados.



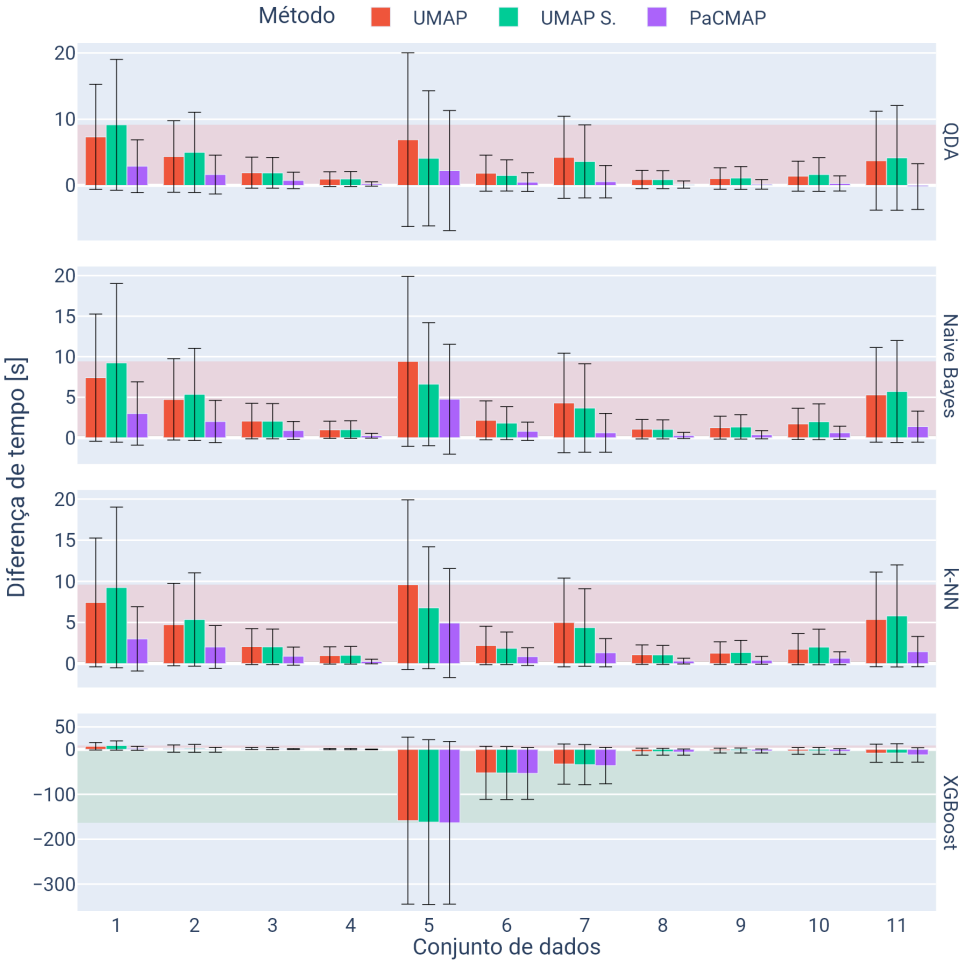
Fonte: Elaborado pelo autor.

4.2 Avaliação do tempo de execução total

Por fim, a Figura 7 mostra a diferença de tempo de execução média de cada método contra o *baseline*. Aqui, o tempo de execução total significa o tempo de execução do método de redução de dimensionalidade mais o tempo de execução para treino e inferência do modelo. Como se pode notar, a redução diminuiu drasticamente o tempo

de execução para o XGBoost em alguns conjuntos, enquanto representou um *overhead* adicional para os outros modelos.

Figura 7 – Diferença média de tempo de execução total nos conjuntos de dados.



Fonte: Elaborado pelo autor.

5 Conclusão

A redução de dimensionalidade é um campo bastante relevante para as aplicações de aprendizado de máquina. O uso de dados com alta dimensionalidade é cada vez mais comum e pode aparecer até em aplicações mais simples, trazendo junto as possíveis consequências da maldição da dimensionalidade. Muitas vezes, os dados aparecem dispostos em espaços com mais dimensões do que eles realmente necessitam, e essas estruturas não triviais prejudicam modelos que partem de premissas mais simples sobre a distribuição dos dados. Logo, isso torna oportuna a redução de dimensionalidade e a busca da dimensionalidade intrínseca desses dados.

As limitações impostas pelo uso de modelos paramétricos podem ser, até certo ponto, contornadas com a aplicação da redução de dimensionalidade. Nos experimentos, o QDA, que desempenhava muito pior do que os outros modelos com os dados em sua dimensionalidade original, foi o mais beneficiado pela aplicação dos métodos, e com o uso do UMAP supervisionado chegou a ter um desempenho comparável. O Naive Bayes também apresentou melhora com a redução. Essa equiparada traz à tona a possibilidade de trocar modelos pesados, como o XGBoost, por esses modelos compactos e definidos por números fixos de parâmetros.

Dos métodos de redução de dimensionalidade, o PaCMAP se demonstrou não ser um método tão adequado *out-of-the-box* para uso com classificadores, apesar de trazer bons resultados com visualizações na literatura. Com os resultados obtidos, mesmo em dimensionalidades mais baixas, seu desempenho ficou aquém dos outros métodos. O UMAP supervisionado, por sua vez, representa uma melhora significativa em relação à sua versão não supervisionada, tanto em score quanto em tempo de execução, o tornando uma boa escolha para a redução quando os rótulos de classe estão disponíveis.

Entretanto, foram poucos os casos em que o melhor desempenho veio da aplicação da redução de dimensionalidade. Na maioria das vezes, o XGBoost sem nenhum método foi o melhor, e a redução não trouxe um aumento suficiente para que os outros modelos o superassem. Como ponto negativo do XGBoost, o seu tempo de execução foi consideravelmente alto para conjuntos com muitos dados. Os ganhos em performance obtidos com a redução indicam um possível caminho de melhoria nessa direção, podendo haver um método que diminua o tempo de execução sem perda do desempenho de classificação.

Para trabalhos futuros, a seguir encontram-se propostas algumas questões não abordadas neste trabalho, mas que são possivelmente relevantes:

- **Avaliação com otimização de hiperparâmetros.** Os métodos e modelos utili-

zados neste trabalho foram executados com os valores padrão fornecidos por cada implementação. No entanto, esses valores podem estar longe do ideal, e um processo de busca por valores ótimos de hiperparâmetros em toda a arquitetura pode alavancar os resultados obtidos.

- **Realizar uma estimativa da dimensionalidade intrínseca para cada conjunto de dados.** Os métodos utilizados foram testados com redução a 10, 20, 30, 50, 80 e 100 componentes em todos os conjuntos. Apesar de ser uma avaliação relativamente abrangente nesse contexto, idealmente, o valor de dimensionalidade intrínseca poderia ser estimado para cada conjunto a fim de se aproximar de uma redução ótima.

- **Foco em conjuntos de dados com aplicações específicas.** Este trabalho teve foco em avaliar os métodos para classificação em conjuntos de dados, em geral, numéricos e de 2 ou mais classes. Porém, problemas mais especializados, como classificação de texto, imagens ou séries temporais, podem apresentar comportamentos diferentes e pertinentes.

- **Análise técnica e de complexidade.** Existem diferentes implementações e possíveis melhorias para os métodos de redução de dimensionalidade utilizados neste trabalho. Um estudo mais aprofundado nos detalhes de implementação e gargalos técnicos pode oferecer novas perspectivas sobre os métodos.

- **Exploração de técnicas de redução de dimensionalidade baseadas em redes neurais.** O aprendizado profundo é um campo muito relevante para dados de alta dimensionalidade, com muitas técnicas sendo exploradas como alternativa aos métodos lineares e não lineares. Os *autoencoders*, por exemplo, têm sido muito utilizados para redução de dimensionalidade, e poderiam contribuir com o desempenho de modelos.

Referências

AMID, E.; WARMUTH, M. K. *TriMap: Large-scale Dimensionality Reduction Using Triplets*. arXiv, 2019. Disponível em: https://arxiv.org/abs/1910.00204. Citado na página 12.

ANOWAR, F.; SADAOUI, S.; SELIM, B. Conceptual and empirical comparison of dimensionality reduction algorithms (PCA, KPCA, LDA, MDS, SVD, LLE, ISOMAP, LE, ICA, t-SNE). *Computer Science Review*, v. 40, p. 100378, maio 2021. ISSN 15740137. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S1574013721000186. Citado na página 11.

AYESHA, S.; HANIF, M. K.; TALIB, R. Overview and comparative study of dimensionality reduction techniques for high dimensional data. *Information Fusion*, v. 59, p. 44–58, jul. 2020. ISSN 15662535. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S156625351930377X. Citado na página 18.

BAGGER, F. O.; KINALIS, S.; RAPIN, N. BloodSpot: a database of healthy and malignant haematopoiesis updated with purified and single cell mRNA sequencing profiles. *Nucleic Acids Research*, v. 47, n. D1, p. D881–D885, jan. 2019. ISSN 0305-1048, 1362-4962. Disponível em: https://academic.oup.com/nar/article/47/D1/D881/5160974. Citado na página 11.

BARSHAN, E. et al. Supervised principal component analysis: Visualization, classification and regression on subspaces and submanifolds. *Pattern Recognition*, v. 44, n. 7, p. 1357–1371, jul. 2011. ISSN 00313203. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S0031320310005819. Citado na página 19.

BECHT, E. et al. Dimensionality reduction for visualizing single-cell data using UMAP. *Nature Biotechnology*, v. 37, n. 1, p. 38–44, jan. 2019. ISSN 1087-0156, 1546-1696. Disponível em: https://www.nature.com/articles/nbt.4314. Citado na página 11.

BELHUMEUR, P. N.; HESPANHA, J. P.; KRIEGMAN, D. J. Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, v. 19, n. 7, 1997. Citado 2 vezes nas páginas 11 e 18.

BELKIN, M.; NIYOGI, P. Laplacian Eigenmaps and Spectral Techniques for Embedding and Clustering. In: DIETTERICH, T.; BECKER, S.; GHAHRAMANI, Z. (Ed.). *Advances in Neural Information Processing Systems*. MIT Press, 2001. v. 14. Disponível em: https://proceedings.neurips.cc/paper_files/paper/2001/file/f106b7f99d2cb30c3db1c3cc0fde9ccb-Paper.pdf. Citado 2 vezes nas páginas 18 e 19.

BELKINA, A. C. et al. Automated optimized parameters for T-distributed stochastic neighbor embedding improve visualization and analysis of large datasets. *Nature Communications*, v. 10, n. 1, p. 5415, nov. 2019. ISSN 2041-1723. Disponível em: https://www.nature.com/articles/s41467-019-13055-y. Citado na página 11.

BINOIS, M.; WYCOFF, N. A Survey on High-dimensional Gaussian Process Modeling with Application to Bayesian Optimization. *ACM Transactions on Evolutionary Learning and Optimization*, v. 2, n. 2, p. 1–26, jun. 2022. ISSN 2688-299X, 2688-3007. Disponível em: https://dl.acm.org/doi/10.1145/3545611. Citado na página 11.

BLECHSCHMIDT, J.; ERNST, O. G. Three ways to solve partial differential equations with neural networks—A review. *GAMM-Mitteilungen*, v. 44, n. 2, p. e202100006, 2021. Publisher: Wiley Online Library. Citado na página 17.

BRONSTEIN, A. M.; BRONSTEIN, M. M.; KIMMEL, R. Generalized multidimensional scaling: a framework for isometry-invariant partial surface matching. *Proceedings of the National Academy of Sciences of the United States of America*, v. 103, n. 5, p. 1168–1172, jan. 2006. ISSN 0027-8424 1091-6490. Place: United States. Citado na página 19.

BROWN, B. C. A. et al. Verifying the Union of Manifolds Hypothesis for Image Data. 2023. Citado 2 vezes nas páginas 11 e 18.

CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco California USA: ACM, 2016. p. 785–794. ISBN 978-1-4503-4232-2. Disponível em: https://dl.acm.org/doi/10.1145/2939672.2939785. Citado na página 16.

DU, X. et al. Multisource Remote Sensing Data Classification With Graph Fusion Network. *IEEE Transactions on Geoscience and Remote Sensing*, v. 59, n. 12, p. 10062–10072, dez. 2021. ISSN 0196-2892, 1558-0644. Disponível em: https://ieeexplore.ieee.org/document/9325097/. Citado na página 11.

FACELI, K. et al. *Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina*. 1. ed. Rio de Janeiro: LTC, 2011. ISBN 978-85-216-1880-5. Citado 3 vezes nas páginas 14, 16 e 26.

FEFFERMAN, C.; MITTER, S.; NARAYANAN, H. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, v. 29, n. 4, p. 983–1049, fev. 2016. ISSN 0894-0347, 1088-6834. Disponível em: https://www.ams.org/jams/2016-29-04/S0894-0347-2016-00852-4/. Citado na página 18.

FIX, E.; HODGES, J. L. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, v. 57, n. 3, p. 238–247, 1989. Publisher: JSTOR. Citado na página 16.

FRIEDMAN, M. The Use of Ranks to Avoid the Assumption of Normality Implicit in the Analysis of Variance. *Journal of the American Statistical Association*, v. 32, n. 200, p. 675–701, 1937. ISSN 01621459. Publisher: [American Statistical Association, Taylor & Francis, Ltd.]. Disponível em: http://www.jstor.org/stable/2279372. Citado na página 28.

GHOJOGH, B. et al. *Elements of Dimensionality Reduction and Manifold Learning*. Cham: Springer International Publishing, 2023. ISBN 978-3-031-10601-9 978-3-031-10602-6. Disponível em: https://link.springer.com/10.1007/978-3-031-10602-6. Citado 2 vezes nas páginas 15 e 19.

GOLUB, G. H.; REINSCH, C. Singular value decomposition and least squares solutions. *Numerische Mathematik*, v. 14, n. 5, p. 403–420, abr. 1970. ISSN 0945-3245. Disponível em: https://doi.org/10.1007/BF02163027. Citado na página 18.

HAJDERANJ, L.; WEHELIYE, I.; CHEN, D. A New Supervised t-SNE with Dissimilarity Measure for Effective Data Visualization and Classification. In: *Proceedings of the 2019 8th International Conference on Software and Information Engineering*. Cairo Egypt: ACM, 2019. p. 232–236. ISBN 978-1-4503-6105-7. Disponível em: https://dl.acm.org/doi/10.1145/3328833.3328853. Citado na página 19.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. 2. ed. New York: Springer-Verlag, 2009. Disponível em: https://hastie.su.domains/ElemStatLearn/. Citado 2 vezes nas páginas 14 e 15.

HIMEUR, Y. et al. AI-big data analytics for building automation and management systems: a survey, actual challenges and future perspectives. *Artificial Intelligence Review*, v. 56, n. 6, p. 4929–5021, jun. 2023. ISSN 0269-2821, 1573-7462. Disponível em: https://link.springer.com/10.1007/s10462-022-10286-2. Citado 2 vezes nas páginas 11 e 18.

HO, T. K. Random decision forests. In: *Proceedings of 3rd International Conference on Document Analysis and Recognition*. [S.l.: s.n.], 1995. v. 1, p. 278–282 vol.1. Journal Abbreviation: Proceedings of 3rd International Conference on Document Analysis and Recognition. Citado na página 16.

JIA, W. et al. Feature dimensionality reduction: a review. *Complex & Intelligent Systems*, v. 8, n. 3, p. 2663–2693, jun. 2022. ISSN 2198-6053. Disponível em: https://doi.org/10.1007/s40747-021-00637-x. Citado 3 vezes nas páginas 12, 16 e 18.

KÖPPEN, M. The curse of dimensionality. In: *5th Online World Conference on Soft Computing in Industrial Applications (WSC5)*. [S.l.: s.n.], 2000. Citado 2 vezes nas páginas 11 e 16.

LEE, D. D.; SEUNG, H. S. Learning the parts of objects by non-negative matrix factorization. *Nature*, v. 401, n. 6755, p. 788–791, out. 1999. ISSN 1476-4687. Disponível em: https://doi.org/10.1038/44565. Citado na página 18.

LOTLIKAR, R.; KOTHARI, R. Adaptive linear dimensionality reduction for classification. *Pattern Recognition*, 2000. Citado na página 18.

MAATEN, L. v. d.; HINTON, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, v. 9, n. 86, p. 2579–2605, 2008. Disponível em: http://jmlr.org/papers/v9/vandermaaten08a.html. Citado 3 vezes nas páginas 12, 18 e 19.

MAATEN, L. van der; POSTMA, E.; HERIK, H. Dimensionality Reduction: A Comparative Review. *Journal of Machine Learning Research - JMLR*, v. 10, jan. 2007. Citado na página 20.

MCINNES, L. et al. UMAP: Uniform Manifold Approximation and Projection. *Journal of Open Source Software*, v. 3, n. 29, p. 861, set. 2018. ISSN 2475-9066. Disponível em: http://joss.theoj.org/papers/10.21105/joss.00861. Citado 6 vezes nas páginas 12, 18, 19, 20, 21 e 22.

NEMENYI, P. B. *Distribution-Free Multiple Comparisons*. Tese (Ph.D.) — Princeton University, 1963. Citado na página 29.

PEARSON, K. LIII. On lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, v. 2, n. 11, p. 559–572, nov. 1901. ISSN 1941-5982. Publisher: Taylor & Francis. Disponível em: https://doi.org/10.1080/14786440109462720. Citado na página 18.

PLATZER, A. Visualization of SNPs with t-SNE. *PLoS ONE*, v. 8, n. 2, p. e56883, fev. 2013. ISSN 1932-6203. Disponível em: https://dx.plos.org/10.1371/journal.pone.0056883. Citado na página 11.

POGGIO, T. et al. Why and when can deep-but not shallow-networks avoid the curse of dimensionality: a review. *International Journal of Automation and Computing*, v. 14, n. 5, p. 503–519, 2017. Publisher: Springer. Citado na página 17.

RIDDER, D. D. et al. Supervised locally linear embedding. In: *International Conference on Artificial Neural Networks*. [S.l.]: Springer, 2003. p. 333–341. Citado na página 19.

ROWEIS, S. T.; SAUL, L. K. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, v. 290, n. 5500, p. 2323–2326, dez. 2000. Publisher: American Association for the Advancement of Science. Disponível em: https://doi.org/10.1126/science.290.5500.2323. Citado na página 19.

RUDIN, C. et al. Interpretable machine learning: Fundamental principles and 10 grand challenges. *Statistics Surveys*, v. 16, n. none, p. 1 – 85, 2022. Publisher: Amer. Statist. Assoc., the Bernoulli Soc., the Inst. Math. Statist., and the Statist. Soc. Canada. Disponível em: https://doi.org/10.1214/21-SS133. Citado na página 11.

RUSSELL, S. J. et al. *Artificial Intelligence: A Modern Approach*. Fourth edition, global edition. Harlow: Pearson, 2022. (Pearson series in artificial intelligence). ISBN 978-1-292-40113-3. Citado 3 vezes nas páginas 14, 15 e 16.

SAINBURG, T.; MCINNES, L.; GENTNER, T. Q. Parametric UMAP Embeddings for Representation and Semisupervised Learning. *Neural Computation*, p. 1–27, ago. 2021. ISSN 0899-7667, 1530-888X. Disponível em: https://direct.mit.edu/neco/article/doi/10.1162/neco_a_01434/107068/Parametric-UMAP-Embeddings-for-Representation-and. Citado na página 19.

SALO, F.; NASSIF, A. B.; ESSEX, A. Dimensionality reduction with IG-PCA and ensemble classifier for network intrusion detection. *Computer Networks*, v. 148, p. 164–175, jan. 2019. ISSN 13891286. Disponível em: https://linkinghub.elsevier.com/retrieve/pii/S1389128618303037. Citado na página 11.

SAUL, L. K.; ROWEIS, S. T. Think Globally, Fit Locally: Unsupervised Learning of Low Dimensional Manifolds. *Journal of Machine Learning Research*, v. 4, p. 119–155, 2003. Citado na página 19.

SCHÖLKOPF, B.; SMOLA, A.; MÜLLER, K.-R. Kernel principal component analysis. In: GERSTNER, W. et al. (Ed.). *Artificial Neural Networks — ICANN'97*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997. p. 583–588. ISBN 978-3-540-69620-9. Citado 2 vezes nas páginas 18 e 19.

SMILKOV, D. et al. *Embedding Projector: Interactive Visualization and Interpretation of Embeddings*. arXiv, 2016. ArXiv:1611.05469 [cs, stat]. Disponível em: http://arxiv.org/abs/1611.05469. Citado na página 11.

SPRENT, P.; SMEETON, N. C. *Applied Nonparametric Statistical Methods*. 3. ed. Washington, D.C.: Chapman & Hall/CRC, 2001. ISBN 978-0-429-12166-1. Citado na página 28.

TENENBAUM, J. B.; SILVA, V. d.; LANGFORD, J. C. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, v. 290, n. 5500, p. 2319–2323, dez. 2000. Publisher: American Association for the Advancement of Science. Disponível em: https://doi.org/10.1126/science.290.5500.2319. Citado na página 19.

VERLEYSEN, M.; FRANÇOIS, D. The Curse of Dimensionality in Data Mining and Time Series Prediction. In: CABESTANY, J.; PRIETO, A.; SANDOVAL, F. (Ed.). *Computational Intelligence and Bioinspired Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005. p. 758–770. ISBN 978-3-540-32106-4. Citado na página 16.

WANG, W.; CARREIRA-PERPINAN, M. The Role of Dimensionality Reduction in Classification. *Proceedings of the AAAI Conference on Artificial Intelligence*, v. 28, n. 1, jun. 2014. ISSN 2374-3468, 2159-5399. Disponível em: https://ojs.aaai.org/index.php/AAAI/article/view/8975. Citado na página 11.

WANG, Y. et al. Understanding How Dimension Reduction Tools Work: An Empirical Approach to Deciphering t-SNE, UMAP, TriMap, and PaCMAP for Data Visualization. *Journal of Machine Learning Research*, v. 22, p. 1–73, 2021. Disponível em: http://jmlr.org/papers/v22/20-1061.html. Citado 5 vezes nas páginas 11, 12, 18, 19 e 22.

XIA, J. et al. Revisiting Dimensionality Reduction Techniques for Visual Cluster Analysis: An Empirical Study. *IEEE Transactions on Visualization and Computer Graphics*, v. 28, n. 1, p. 529–539, jan. 2022. ISSN 1077-2626, 1941-0506, 2160-9306. Disponível em: https://ieeexplore.ieee.org/document/9552226/. Citado na página 19.

Apêndices

APÊNDICE A – Resultados completos

Este apêndice traz todos os resultados de F1 score obtidos na avaliação dos métodos, além de visualizações separadas por número de componentes utilizados na redução.

A.1 Resultados da redução a 10 componentes

Tabela 8 – F1 score de classificação do QDA para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,4958	0,4953	0,4958	**0,5153**
har	0,3875	0,7921	**0,9109**	0,3580
cnae-9	0,1111	**0,7524**	0,6480	0,0658
micro-mass	0,0819	0,5078	**0,7828**	0,0427
CIFAR_10	0,1048	**0,1329**	0,0844	0,0397
eating	0,1532	0,1619	**0,1647**	0,1155
amazon-commerce-reviews	0,0181	**0,0939**	0,0414	0,0035
AP_Ovary_Uterus	0,5020	0,6672	**0,7343**	0,4366
AP_Endometrium_Breast	0,4101	0,7424	**0,9063**	0,4586
AP_Breast_Kidney	0,5786	0,9508	**0,9575**	0,3055
OVA_Uterus	0,2836	0,5214	**0,7466**	0,4791

Tabela 9 – F1 score de classificação do Naive Bayes para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,6157**	0,5310	0,5304	0,5153
har	0,6597	0,8394	**0,9172**	0,8368
cnae-9	**0,8470**	0,7652	0,7912	0,6673
micro-mass	**0,8822**	0,7280	0,7799	0,0491
CIFAR_10	**0,2734**	0,2030	0,2302	0,1925
eating	0,0383	**0,1371**	0,1367	0,1241
amazon-commerce-reviews	**0,5514**	0,0576	0,0717	0,0387
AP_Ovary_Uterus	0,4859	0,5831	**0,7347**	0,4665
AP_Endometrium_Breast	0,7757	0,8632	**0,8936**	0,4497
AP_Breast_Kidney	0,9244	0,9558	**0,9575**	0,4280
OVA_Uterus	0,6755	0,6234	**0,7374**	0,5226

Tabela 10 – F1 score de classificação do k-NN para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,5912	**0,5935**	0,4958	0,4950
har	0,9267	0,8908	**0,9277**	0,8857
cnae-9	**0,8319**	0,7796	0,8255	0,7081
micro-mass	**0,8214**	0,7800	0,8148	0,0423
CIFAR_10	**0,2465**	0,2155	0,2168	0,2081
eating	**0,3689**	0,1419	0,1695	0,1022
amazon-commerce-reviews	0,0239	0,1289	**0,1587**	0,0901
AP_Ovary_Uterus	**0,7618**	0,6765	0,7347	0,5165
AP_Endometrium_Breast	**0,9383**	0,8676	0,8961	0,4731
AP_Breast_Kidney	**0,9729**	0,9541	0,9575	0,4419
OVA_Uterus	**0,7800**	0,6822	0,7298	0,5241

Tabela 11 – F1 score de classificação do XGBoost para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,4958	0,4956	**0,5103**	0,4954
har	**0,9655**	0,8974	0,9244	0,8828
cnae-9	0,6813	**0,7787**	0,7674	0,7092
micro-mass	**0,9078**	0,7229	0,6922	0,0714
CIFAR_10	**0,4352**	0,2445	0,1972	0,2288
eating	**0,5749**	0,1538	0,1529	0,1391
amazon-commerce-reviews	**0,5728**	0,1291	0,1320	0,0887
AP_Ovary_Uterus	**0,8961**	0,6488	0,6886	0,5094
AP_Endometrium_Breast	**0,9235**	0,8812	0,8622	0,4732
AP_Breast_Kidney	**0,9746**	0,9459	0,9409	0,5388
OVA_Uterus	**0,7655**	0,6480	0,6968	0,4956

A.2 Resultados da redução a 20 componentes

Tabela 12 – F1 score de classificação do QDA para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,4958	**0,5111**	0,4958	0,4958
har	0,3875	0,6908	**0,9145**	0,3904
cnae-9	0,1111	0,6601	**0,6816**	0,0562
micro-mass	0,0819	0,4043	**0,7492**	0,0519
CIFAR_10	**0,1048**	0,0865	0,0281	0,0370
eating	**0,1532**	0,1313	0,1445	0,1040
amazon-commerce-reviews	0,0181	**0,0211**	0,0161	0,0057
AP_Ovary_Uterus	0,5020	0,4691	**0,7294**	0,4583
AP_Endometrium_Breast	0,4101	0,4593	**0,8936**	0,4586
AP_Breast_Kidney	0,5786	0,8822	**0,9558**	0,3747
OVA_Uterus	0,2836	0,5053	**0,7154**	0,4791

Tabela 13 – F1 score de classificação do Naive Bayes para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,6157**	0,5592	0,5669	0,4607
har	0,6597	0,8426	**0,9142**	0,8307
cnae-9	**0,8470**	0,7610	0,8284	0,6724
micro-mass	**0,8822**	0,7055	0,7632	0,0496
CIFAR_10	**0,2734**	0,2042	0,2379	0,2023
eating	0,0383	0,1299	**0,1457**	0,1382
amazon-commerce-reviews	**0,5514**	0,0578	0,0915	0,0378
AP_Ovary_Uterus	0,4859	0,6479	**0,7538**	0,4945
AP_Endometrium_Breast	0,7757	0,8595	**0,9006**	0,4696
AP_Breast_Kidney	0,9244	**0,9575**	**0,9575**	0,4110
OVA_Uterus	0,6755	0,6103	**0,7336**	0,4930

Tabela 14 – F1 score de classificação do k-NN para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,5912	**0,6040**	0,4958	0,5094
har	0,9267	0,8960	**0,9274**	0,8838
cnae-9	**0,8319**	0,7860	0,8304	0,7069
micro-mass	**0,8214**	0,7441	0,8115	0,0702
CIFAR_10	**0,2465**	0,2140	0,2231	0,2106
eating	**0,3689**	0,1727	0,1753	0,1304
amazon-commerce-reviews	0,0239	0,1313	**0,1758**	0,0857
AP_Ovary_Uterus	**0,7618**	0,6663	0,7409	0,5326
AP_Endometrium_Breast	**0,9383**	0,8619	0,9006	0,4586
AP_Breast_Kidney	**0,9729**	0,9541	0,9575	0,5286
OVA_Uterus	**0,7800**	0,6522	0,7291	0,4730

Tabela 15 – F1 score de classificação do XGBoost para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,4958	**0,5267**	0,5188	0,4957
har	**0,9655**	0,9008	0,9241	0,8957
cnae-9	0,6813	**0,7951**	0,7737	0,7150
micro-mass	**0,9078**	0,7322	0,5660	0,0991
CIFAR_10	**0,4352**	0,2439	0,2073	0,2348
eating	**0,5749**	0,1846	0,1897	0,1420
amazon-commerce-reviews	**0,5728**	0,1334	0,1397	0,0950
AP_Ovary_Uterus	**0,8961**	0,6806	0,6696	0,5004
AP_Endometrium_Breast	**0,9235**	0,8496	0,8895	0,4519
AP_Breast_Kidney	**0,9746**	0,9326	0,9348	0,4141
OVA_Uterus	**0,7655**	0,6544	0,7021	0,4832

A.3 Resultados da redução a 30 componentes

Tabela 16 – F1 score de classificação do QDA para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,4958**	**0,4958**	**0,4958**	**0,4958**
har	0,3875	0,5567	**0,9100**	0,4443
cnae-9	0,1111	0,5480	**0,7067**	0,0529
micro-mass	0,0819	0,2949	**0,4958**	0,0838
CIFAR_10	**0,1048**	0,0715	0,0230	0,0435
eating	**0,1532**	0,1145	0,1440	0,1119
amazon-commerce-reviews	0,0181	0,0199	**0,0247**	0,0062
AP_Ovary_Uterus	0,5020	0,4271	**0,7433**	0,4326
AP_Endometrium_Breast	0,4101	0,4753	**0,8936**	0,4593
AP_Breast_Kidney	0,5786	0,7136	**0,9575**	0,3402
OVA_Uterus	0,2836	0,4791	**0,7356**	0,4791

Tabela 17 – F1 score de classificação do Naive Bayes para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,6157**	0,5493	0,5669	0,4425
har	0,6597	0,8480	**0,9152**	0,8322
cnae-9	**0,8470**	0,7691	0,8318	0,6806
micro-mass	**0,8822**	0,6932	0,7928	0,0371
CIFAR_10	**0,2734**	0,1962	0,2394	0,1975
eating	0,0383	0,1337	**0,1590**	0,1353
amazon-commerce-reviews	**0,5514**	0,0569	0,0968	0,0381
AP_Ovary_Uterus	0,4859	0,6406	**0,7409**	0,4635
AP_Endometrium_Breast	0,7757	0,8541	**0,8936**	0,4593
AP_Breast_Kidney	0,9244	0,9558	**0,9575**	0,3587
OVA_Uterus	0,6755	0,5940	**0,7349**	0,4787

Tabela 18 – F1 score de classificação do k-NN para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,5912	**0,6186**	0,4958	0,5078
har	0,9267	0,8998	**0,9278**	0,8846
cnae-9	0,8319	0,7797	**0,8494**	0,7056
micro-mass	**0,8214**	0,7745	0,8202	0,0879
CIFAR_10	**0,2465**	0,2131	0,2296	0,2080
eating	**0,3689**	0,1677	0,1860	0,0922
amazon-commerce-reviews	0,0239	0,1313	**0,1826**	0,0915
AP_Ovary_Uterus	**0,7618**	0,7187	0,7378	0,4709
AP_Endometrium_Breast	**0,9383**	0,8797	0,8961	0,4710
AP_Breast_Kidney	**0,9729**	0,9558	0,9575	0,4878
OVA_Uterus	**0,7800**	0,6747	0,7356	0,4805

Tabela 19 – F1 score de classificação do XGBoost para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,4958	0,5267	**0,5499**	0,4953
har	**0,9655**	0,8991	0,9268	0,8908
cnae-9	0,6813	**0,7872**	0,7537	0,7130
micro-mass	**0,9078**	0,7570	0,6655	0,0739
CIFAR_10	**0,4352**	0,2428	0,2250	0,2356
eating	**0,5749**	0,1843	0,1768	0,1136
amazon-commerce-reviews	**0,5728**	0,1295	0,1321	0,0922
AP_Ovary_Uterus	**0,8961**	0,7076	0,7183	0,4561
AP_Endometrium_Breast	**0,9235**	0,8907	0,8879	0,4978
AP_Breast_Kidney	**0,9746**	0,9409	0,9544	0,4420
OVA_Uterus	**0,7655**	0,6817	0,7102	0,4914

A.4 Resultados da redução a 50 componentes

Tabela 20 – F1 score de classificação do QDA para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,4958**	**0,4958**	**0,4958**	**0,4958**
har	0,3875	0,4503	**0,9188**	0,3738
cnae-9	0,1111	0,2223	**0,6450**	0,0436
micro-mass	0,0819	0,2695	**0,6437**	0,1034
CIFAR_10	**0,1048**	0,0627	0,0203	0,0360
eating	**0,1532**	0,1050	0,0980	0,1300
amazon-commerce-reviews	0,0181	**0,0185**	0,0179	0,0089
AP_Ovary_Uterus	0,5020	0,3892	**0,7350**	0,3994
AP_Endometrium_Breast	0,4101	**0,4593**	**0,4593**	**0,4593**
AP_Breast_Kidney	0,5786	0,5057	**0,8829**	0,3329
OVA_Uterus	0,2836	0,4791	**0,6965**	0,4791

Tabela 21 – F1 score de classificação do Naive Bayes para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,6157**	0,5667	0,5515	0,4321
har	0,6597	0,8411	**0,9138**	0,8362
cnae-9	**0,8470**	0,7775	0,8280	0,6917
micro-mass	**0,8822**	0,7063	0,7252	0,0366
CIFAR_10	**0,2734**	0,2007	0,2427	0,2021
eating	0,0383	0,1400	**0,1667**	0,0928
amazon-commerce-reviews	**0,5514**	0,0523	0,1026	0,0463
AP_Ovary_Uterus	0,4859	0,6506	**0,7399**	0,5004
AP_Endometrium_Breast	0,7757	0,8645	**0,8936**	0,4593
AP_Breast_Kidney	0,9244	0,9558	**0,9575**	0,3634
OVA_Uterus	0,6755	0,5947	**0,7323**	0,4789

Tabela 22 – F1 score de classificação do k-NN para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,5912**	0,5875	0,4958	0,5084
har	0,9267	0,8927	**0,9282**	0,8856
cnae-9	0,8319	0,7990	**0,8460**	0,7175
micro-mass	**0,8214**	0,7693	0,8151	0,0474
CIFAR_10	**0,2465**	0,2106	0,2301	0,2170
eating	**0,3689**	0,1810	0,1869	0,0829
amazon-commerce-reviews	0,0239	0,1344	**0,1789**	0,0921
AP_Ovary_Uterus	**0,7618**	0,7021	0,7347	0,4964
AP_Endometrium_Breast	**0,9383**	0,8867	0,9006	0,4586
AP_Breast_Kidney	**0,9729**	0,9558	0,9575	0,4310
OVA_Uterus	**0,7800**	0,6654	0,7356	0,4856

Tabela 23 – F1 score de classificação do XGBoost para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,4958	**0,5267**	0,4951	0,4955
har	**0,9655**	0,8989	0,9250	0,8814
cnae-9	0,6813	**0,8121**	0,7964	0,7132
micro-mass	**0,9078**	0,7669	0,6597	0,0920
CIFAR_10	**0,4352**	0,2352	0,2069	0,2368
eating	**0,5749**	0,1597	0,1770	0,1427
amazon-commerce-reviews	**0,5728**	0,1328	0,1458	0,0929
AP_Ovary_Uterus	**0,8961**	0,7200	0,7248	0,4855
AP_Endometrium_Breast	**0,9235**	0,8998	0,8434	0,4882
AP_Breast_Kidney	**0,9746**	0,9291	0,9542	0,4171
OVA_Uterus	**0,7655**	0,6495	0,7198	0,4787

A.5 Resultados da redução a 80 componentes

Tabela 24 – F1 score de classificação do QDA para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,4958**	**0,4958**	**0,4958**	**0,4958**
har	0,3875	0,3753	**0,7525**	0,4504
cnae-9	0,1111	0,0565	**0,6285**	0,0549
CIFAR_10	**0,1048**	0,0540	0,0187	0,0461
eating	**0,1532**	0,0921	0,1138	0,1260
amazon-commerce-reviews	0,0181	0,0161	**0,0193**	0,0043
AP_Endometrium_Breast	0,4101	0,4578	**0,7634**	0,3903
AP_Breast_Kidney	0,5786	0,4390	**0,9575**	0,3550
OVA_Uterus	0,2836	**0,4791**	**0,4791**	**0,4791**

Tabela 25 – F1 score de classificação do Naive Bayes para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,6157**	0,5494	0,5790	0,4278
har	0,6597	0,8415	**0,9160**	0,8330
cnae-9	**0,8470**	0,7636	0,8303	0,6906
CIFAR_10	**0,2734**	0,1991	0,2261	0,2078
eating	0,0383	0,1392	**0,1514**	0,0837
amazon-commerce-reviews	**0,5514**	0,0501	0,0991	0,0404
AP_Endometrium_Breast	0,7757	0,8496	**0,8936**	0,4593
AP_Breast_Kidney	0,9244	0,9558	**0,9575**	0,3453
OVA_Uterus	0,6755	0,5766	**0,7309**	0,4791

Tabela 26 – F1 score de classificação do k-NN para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,5912	**0,6178**	0,5113	0,5076
har	0,9267	0,8934	**0,9278**	0,8848
cnae-9	0,8319	0,7934	**0,8443**	0,7262
CIFAR_10	**0,2465**	0,2246	0,2340	0,2158
eating	**0,3689**	0,1686	0,1648	0,0780
amazon-commerce-reviews	0,0239	0,1343	**0,1757**	0,1017
AP_Endometrium_Breast	**0,9383**	0,8698	0,8961	0,4593
AP_Breast_Kidney	**0,9729**	0,9541	0,9575	0,4320
OVA_Uterus	**0,7800**	0,6729	0,7311	0,4777

Tabela 27 – F1 score de classificação do XGBoost para cada conjunto de dados e método aplicado.

Conjunto	**Nenhum**	**UMAP**	**UMAP S.**	**PaCMAP**
Speech	0,4958	0,4958	**0,5253**	0,4956
har	**0,9655**	0,8961	0,9263	0,8955
cnae-9	0,6813	**0,8083**	0,7967	0,7216
CIFAR_10	**0,4352**	0,2472	0,2121	0,2391
eating	**0,5749**	0,1786	0,1716	0,1311
amazon-commerce-reviews	**0,5728**	0,1253	0,1471	0,0953
AP_Endometrium_Breast	**0,9235**	0,8815	0,8774	0,4843
AP_Breast_Kidney	**0,9746**	0,9413	0,8665	0,4125
OVA_Uterus	**0,7655**	0,6506	0,7071	0,4777

A.6 Resultados da redução a 100 componentes

Tabela 28 – F1 score de classificação do QDA para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,4958**	**0,4958**	**0,4958**	0,4953
har	0,3875	0,3290	**0,8311**	0,3935
cnae-9	0,1111	0,1411	**0,4784**	0,0668
CIFAR_10	**0,1048**	0,0587	0,0250	0,0496
eating	**0,1532**	0,0741	0,0835	0,1249
amazon-commerce-reviews	0,0181	0,0185	**0,0205**	0,0085
AP_Breast_Kidney	0,5786	0,5250	**0,9416**	0,3824
OVA_Uterus	0,2836	**0,4791**	**0,4791**	**0,4791**

Tabela 29 – F1 score de classificação do Naive Bayes para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	**0,6157**	0,5549	0,5689	0,4285
har	0,6597	0,8406	**0,9177**	0,8360
cnae-9	**0,8470**	0,7647	0,8358	0,6885
CIFAR_10	**0,2734**	0,2016	0,2119	0,1997
eating	0,0383	0,1325	**0,1696**	0,0818
amazon-commerce-reviews	**0,5514**	0,0533	0,0987	0,0418
AP_Breast_Kidney	0,9244	**0,9575**	**0,9575**	0,3397
OVA_Uterus	0,6755	0,5926	**0,7322**	0,4791

Tabela 30 – F1 score de classificação do k-NN para cada conjunto de dados e método aplicado.

Conjunto	Nenhum	UMAP	UMAP S.	PaCMAP
Speech	0,5912	**0,6123**	0,5360	0,5208
har	0,9267	0,9031	**0,9274**	0,8884
cnae-9	0,8319	0,7941	**0,8533**	0,7197
CIFAR_10	**0,2465**	0,2184	0,2447	0,2126
eating	**0,3689**	0,1655	0,1830	0,0783
amazon-commerce-reviews	0,0239	0,1436	**0,1689**	0,0986
AP_Breast_Kidney	**0,9729**	0,9541	0,9575	0,4401
OVA_Uterus	**0,7800**	0,6825	0,7330	0,4784

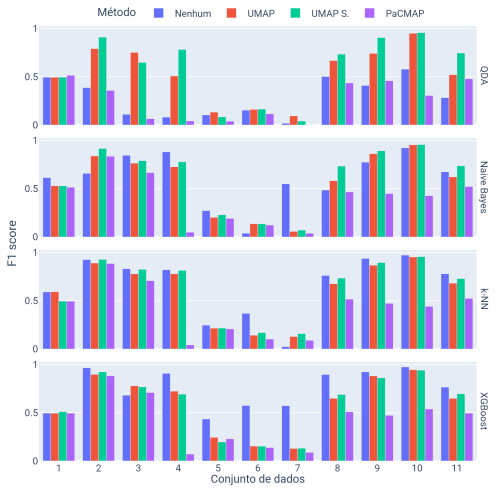
Tabela 31 – F1 score de classificação do XGBoost para cada conjunto de dados e método aplicado.

Conjunto	**Nenhum**	**UMAP**	**UMAP S.**	**PaCMAP**
Speech	0,4958	0,5113	**0,5345**	0,4957
har	**0,9655**	0,9007	0,9292	0,8875
cnae-9	0,6813	0,8010	**0,8038**	0,7332
CIFAR_10	**0,4352**	0,2464	0,2103	0,2433
eating	**0,5749**	0,1457	0,1664	0,1151
amazon-commerce-reviews	**0,5728**	0,1335	0,1479	0,0872
AP_Breast_Kidney	**0,9746**	0,9304	0,9216	0,4051
OVA_Uterus	**0,7655**	0,6569	0,7263	0,4786

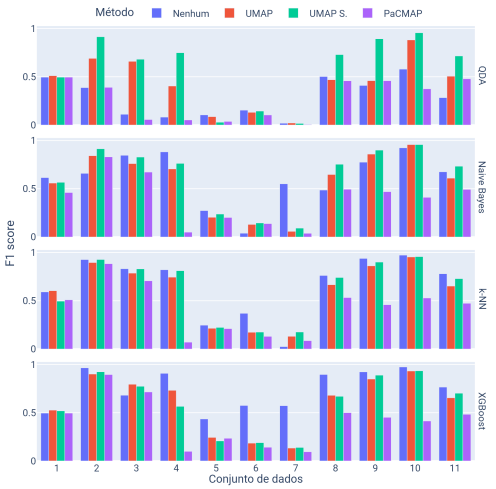
A.7 Visualização do F1 score para cada método

Figura 8 – F1 score para cada modelo, conjunto de dados e método.

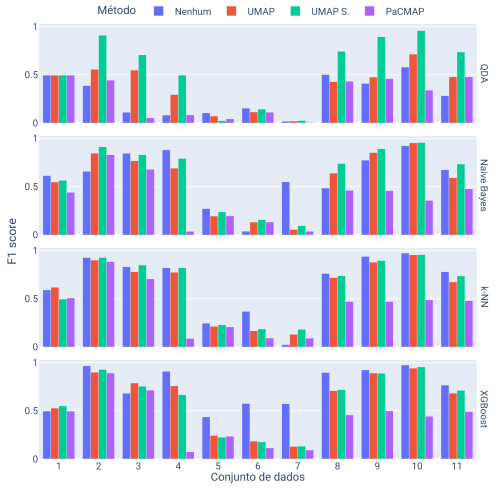
(a) 10 componentes



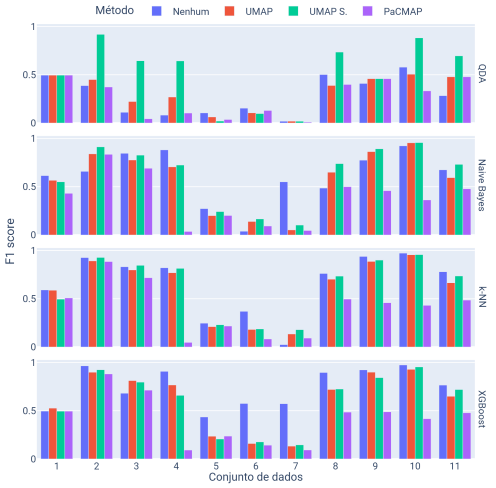
(b) 20 componentes



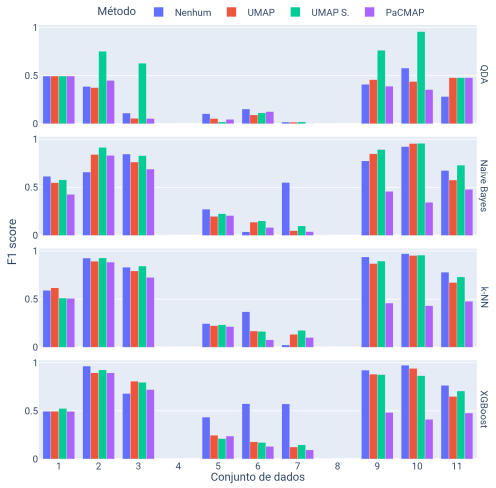
(c) 30 componentes



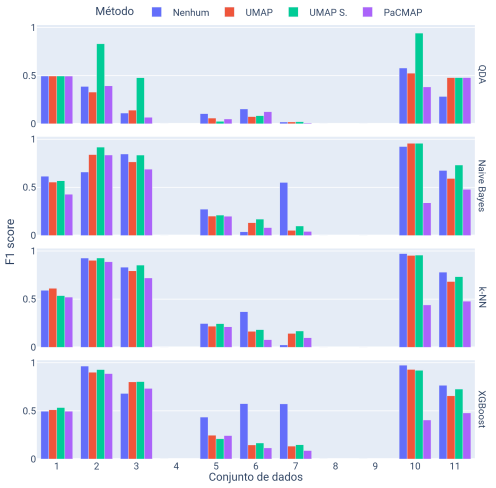
(d) 50 componentes



(e) 80 componentes



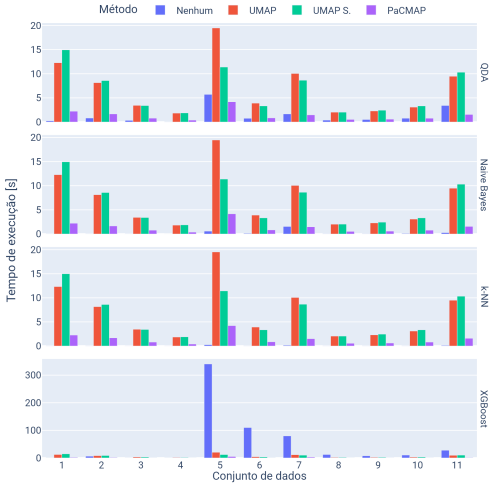
(f) 100 componentes



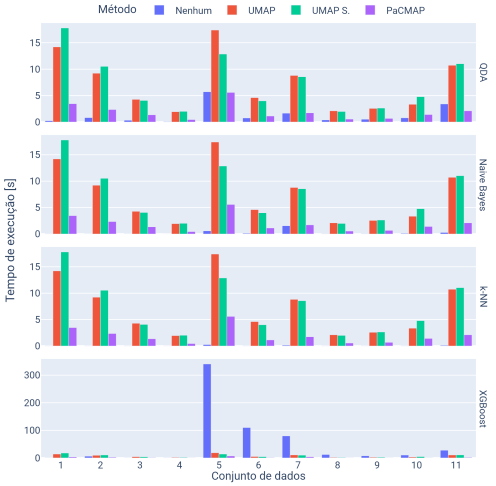
A.8 Visualização do tempo de execução para cada método

Figura 9 – Tempo de execução para cada modelo, conjunto de dados e método.

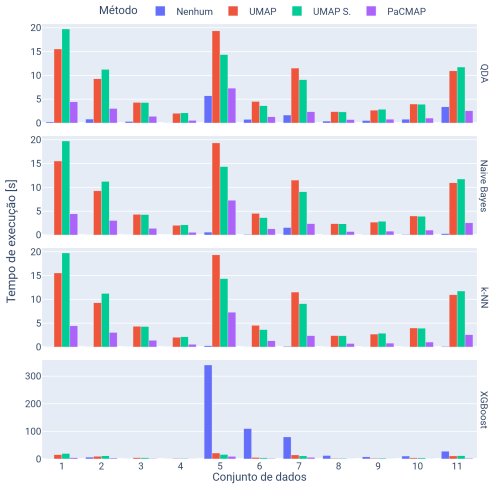
(a) 10 componentes



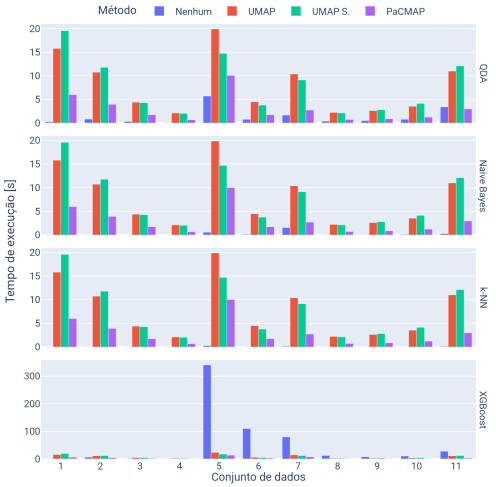
(b) 20 componentes



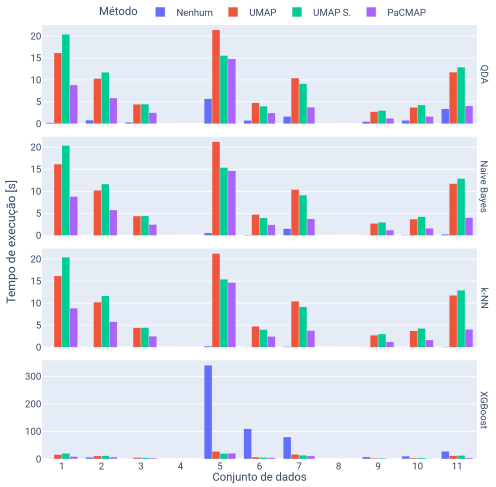
(c) 30 componentes



(d) 50 componentes



(e) 80 componentes



(f) 100 componentes