UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE ENGENHARIA DE COMPUTAÇÃO

AFONSO MARTINI SPEZIA

Estudo de Estratégias de Validação Cruzada Baseadas em Clusters

Monografia apresentada como requisito parcial para a obtenção do grau de Bacharel em Engenharia da Computação

Orientador: Prof.ª Dr.ª Mariana Recamonde-Mendoza

Porto Alegre
2024

AGRADECIMENTOS

Em primeiro lugar, gostaria de expressar minha profunda gratidão à minha família. Aos meus pais, Angela e Mario, e à minha irmã, Amanda, por seu amor incondicional, apoio e por sempre acreditarem em mim. Agradeço também à minha orientadora, Professora Mariana, com quem tive o privilégio de iniciar minha jornada acadêmica no curso e agora a honra de concluir esta etapa final. Gostaria de dedicar um agradecimento especial ao meu namorado, Guilherme, por sua paciência e apoio no final dessa jornada. Agradeço também aos meus colegas de curso e amigos, que estiveram ao meu lado nesta jornada, prestando apoio nos momentos de dificuldade e celebrando as conquistas junto comigo. Por fim, gostaria de agradecer a todos os professores e funcionários da UFRGS, que contribuíram direta ou indiretamente para a realização deste trabalho

RESUMO

A Validação Cruzada desempenha um papel fundamental no Aprendizado de Máquina, permitindo uma avaliação robusta do desempenho dos modelos e evitando a superestimação desse desempenho em dados de treinamento e validação. No entanto, uma de suas desvantagens é a possibilidade de criar subconjuntos de dados (folds) que não representam adequadamente a diversidade do conjunto original, o que pode levar a estimativas de desempenho enviesadas. O objetivo deste trabalho é aprofundar a pesquisa em estratégias de validação cruzada baseadas em clusters, analisando o desempenho de diferentes algoritmos de agrupamento por meio de uma comparação experimental. Além disso, é proposta uma nova técnica de validação cruzada que combina Mini Batch K-Means com estratificação por classe. Experimentos foram conduzidos em 20 conjuntos de dados (balanceados e desbalanceados) utilizando quatro algoritmos de aprendizado supervisionado, comparando as estratégias de validação cruzada em termos de viés, variância e custo computacional. A técnica que utiliza Mini Batch K-Means com estratificação por classe superou outras em termos de viés e variância em datasets balanceados, mas não reduziu significativamente o custo computacional. Em datasets desbalanceados, a validação cruzada estratificada tradicional foi consistentemente superior, apresentando menor viés, variância e custo computacional, tornando-se uma escolha segura para avaliação de desempenho em cenários com desbalanceamento de classes. Na comparação entre diferentes algoritmos de agrupamento, não houve um algoritmo que se destacou consistentemente como superior. De forma geral, este trabalho contribui para o aprimoramento das estratégias de avaliação de modelos preditivos, oferecendo um melhor entendimento sobre o potencial das técnicas de divisão de dados baseadas em clusters e a eficácia de estratégias bem estabelecidas, como a validação cruzada estratificada. Além disso, aponta perspectivas para aumentar a robustez e a confiabilidade na avaliação de modelos de AM, especialmente em conjuntos de dados com características de agrupamento.

Palavras-chave: Validação cruzada. aprendizado de máquina. algoritmos de agrupamento. avaliação de modelos.

Study of Cluster-Based Cross-Validation Strategies

ABSTRACT

Cross-Validation plays a fundamental role in Machine Learning, allowing for a robust evaluation of model performance and preventing the overestimation of this performance in training and validation data. However, one of its drawbacks is the potential to create data subsets (folds) that do not adequately represent the diversity of the original dataset, which can lead to biased performance estimates. The objective of this work is to deepen the research on cluster-based cross-validation strategies by analyzing the performance of different clustering algorithms through an experimental comparison. Additionally, a new cross-validation technique that combines Mini Batch K-Means with class stratification is proposed. Experiments were conducted on 20 datasets (both balanced and imbalanced) using four supervised learning algorithms, comparing cross-validation strategies in terms of bias, variance, and computational cost. The technique that uses Mini Batch K-Means with class stratification outperformed others in terms of bias and variance in balanced datasets, but did not significantly reduce computational cost. In imbalanced datasets, traditional stratified cross-validation consistently performed better, showing lower bias, variance, and computational cost, making it a safe choice for performance evaluation in scenarios with class imbalance. In the comparison of different clustering algorithms, no single algorithm consistently stood out as superior. Overall, this work contributes to the enhancement of predictive model evaluation strategies, providing a better understanding of the potential of cluster-based data splitting techniques and the effectiveness of well-established strategies like stratified cross-validation. Moreover, it highlights perspectives for increasing the robustness and reliability of model evaluations, especially in datasets with clustering characteristics.

Keywords: cross-validation, machine learning, clustering algorithms, model evaluation.

LISTA DE ABREVIATURAS E SIGLAS

ACBCV	*Agglomerative Cluster-Based Cross-Validation*
AD	*Árvore de Decisão*
AM	Aprendizado de Máquina
KCBCV	*K-Means Cluster-Based Cross-Validation*
KCBCV Mini	*Mini Batch K-Means Cluster-Based Cross-Validation*
CV	*Cross-Validation*
DBSCAN	*Density-Based Spatial Clustering of Applications with Noise*
DBSCANBCV	*DBSCAN-Based Cross-Validation*
PMLB	*Penn Machine Learning Benchmarks*
RBF	*Radial Basis Function*
RF	*Random Forest*
RL	*Regressão Logística*
SCBCV	*Stratified Cluster-Based Cross-Validation*
SCBCV Mini	*Mini Batch Stratified Cluster-Based Cross-Validation*
SCV	*Stratified Cross-Validation*
SVM	*Support Vector Machines*

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

O campo de aprendizado de máquina (AM) tem ganhado exponencial relevância nos últimos anos devido a suas aplicações em diversas áreas de estudos, entre elas a da saúde, empresarial, industrial e militar (AGGARWAL et al., 2022). Os modelos criados capacitam os sistemas a aprenderem a partir de extensos conjuntos de dados, possibilitando a detecção de padrões complexos e a tomada de decisões baseadas neles. Contudo, a confiabilidade dos resultados desses modelos depende da qualidade das métricas utilizadas para avaliá-los. A obtenção de métricas precisas é fundamental para a compreensão do real desempenho desses modelos, dessa forma destaca-se a necessidade de se ter ferramentas robustas de avaliação.

A validação cruzada se destaca nesse contexto como uma técnica para a avaliação e seleção de modelos preditivos. Sendo o método de avaliação mais popular utilizado atualmente, ele funciona dividindo o conjunto de dados em partes (denominadas *folds*) que são utilizadas para treinar e testar os modelos de forma iterativa. Isso ajuda a garantir que eles sejam robustos e com bom poder de generalização para diferentes dados, evitando problemas como a superestimação do desempenho.

Nos últimos anos, diversos trabalhos vêm propondo maneiras mais precisas e eficientes de se avaliar os modelos criados. Por meio da estratificação por classe e da divisão baseada em clusters, por exemplo, podemos obter avaliações mais precisas. No entanto, algumas destas soluções, como divisões dos dados baseadas em clusters, podem ser computacionalmente custosas em grandes conjuntos de dados (IKOTUN et al., 2023).

Em um trabalho anterior, Fontanari, Fróes e Recamonde-Mendoza (2022) propuseram o uso de Mini-Batches K-Means para diminuir o processamento necessário no processo de clusterização. Entretanto, os autores mostraram que ainda que estratégias mais elaboradas de validação cruzada possam apresentar potenciais ganhos na avaliação de desempenho dos modelos, a validação cruzada estratificada é preferível com um pequeno número de *folds* ou com conjuntos de dados desbalanceados.

Nesse sentido, este trabalho visa propor uma implementação de K-Fold e Mini Batch K-Fold estratificado por classe para uma divisão baseada em clusters, visando capturar possíveis subgrupos intraclasse que outras técnicas não capturariam, em uma tentativa de obter estimativas com viés, variância e custo computacional equilibrados. Adicionalmente, objetiva-se realizar um estudo comparativo entre métodos de separação de dados já consolidados e o método proposto, bem como entre o uso de K-Means e

dos algoritmos de agrupamento DBSCAN (SANDER et al., 1998) e *Agglomerative Hierarchical Clustering* (DIAMANTIDIS; KARLIS; GIAKOUMAKIS, 2000). Durante os experimentos executados, diferentes métodos de separação são analisados objetivando avaliar se algum se destaca em termos de viés, variância e custo computacional.

O presente trabalho está estruturado em 6 capítulos. No Capítulo 2, será apresentada a fundamentação teórica necessária para a compreensão dos conceitos de aprendizado de máquina, validação cruzada e algoritmos utilizados neste estudo. No Capítulo 3, serão discutidos os trabalhos relacionados que abordam técnicas de validação cruzada e seus impactos na avaliação de modelos preditivos, comparando-os com a proposta deste trabalho. No Capítulo 4, será detalhada a metodologia adotada, incluindo a implementação do Mini-Batch K-Fold estratificado e os algoritmos de agrupamento utilizados. No Capítulo 5, serão apresentados os resultados dos experimentos e a análise comparativa entre os métodos estudados, discutindo-se o viés, a variância e o custo computacional de cada abordagem. Por fim, no Capítulo 6, serão expostas as conclusões a partir dos resultados obtidos e as possíveis direções para trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo serão abordados os principais conceitos relacionados ao desenvolvimento deste trabalho, incluindo uma breve revisão sobre aprendizado de máquina supervisionado e não-supervisionado, e alguns algoritmos selecionados para cada tipo de aprendizado. As técnicas de validação utilizadas para avaliação dos modelos preditivos gerados com aprendizado de máquina são o tópico principal abordado no presente trabalho e também serão discutidas, focando especificamente na validação cruzada e suas variações.

2.1 Aprendizado de Máquina

O Aprendizado de Máquina (AM) é um termo que se refere a uma ampla gama de algoritmos utilizados para realizar previsões e detecção de padrões de forma automatizada em conjuntos de dados. Os avanços mais recentes na área alcançaram um nível de compreensão semântica e extração de informações semelhante ou até mesmo melhores que o humano. Seu crescimento exponencial nas últimas décadas deve-se ao aumento do volume de dados, ao rápido crescimento da capacidade computacional e aos avanços no design de algoritmos (NICHOLS; CHAN; BAKER, 2019).

De forma geral, o AM descreve a capacidade de sistemas de aprenderem e fazerem inferências a partir de conjuntos de dados de problemas específicos, automatizando o processo de construção de modelos analíticos e resolvendo tarefas associadas a eles (BISHOP, 2006). Dentre os principais tipos de aprendizado, destacamos o aprendizado supervisionado e o aprendizado não-supervisionado, revisados nas seções a seguir.

2.1.1 Aprendizado Supervisionado

O aprendizado supervisionado é uma técnica de AM onde um modelo é treinado com base em um conjunto de dados rotulados. Isso significa que cada dado no conjunto de treinamento é composto por uma entrada e pela saída desejada, ou rótulo. O objetivo dos algoritmos de aprendizado supervisionado é aprender a mapear entradas para saídas corretas, de forma a permitir que o modelo faça previsões precisas em dados não vistos anteriormente.

Esse tipo de aprendizado tem sido amplamente estudado e aplicado em diversas áreas, mostrando-se eficaz na solução de problemas complexos de classificação e regressão. A revisão de aprendizado supervisionado de Mohri, Rostamizadeh e Talwalkar (2012) destaca a ampla aplicação e eficácia dessas técnicas em diversos domínios, incluindo reconhecimento de fala, visão computacional, bioinformática e detecção de fraudes.

No contexto deste trabalho, técnicas de aprendizado supervisionado, como a Validação Cruzada Estratificada por Classe, são empregadas para avaliar o desempenho de modelos preditivos em dados rotulados. Além disso, algoritmos de aprendizado supervisionado são utilizados para treinar os modelos, na tentativa de estabelecer uma relação entre os dados de entrada e os rótulos de saída, neste trabalho referidos como classes.

2.1.1.1 Regressão Logística

Regressão Logística (RL) é um algoritmo de aprendizado de máquina amplamente utilizado para problemas de classificação, especialmente quando o objetivo é prever a probabilidade de ocorrência de uma classe binária. Conforme discute Menard (2002), o RL é robusto frente a situações em que as relações entre variáveis independentes e a variável dependente não são perfeitamente lineares, o que amplia sua aplicabilidade em dados do mundo real, onde essas relações são frequentemente mais complexas (MENARD, 2002).

Ele funciona modelando a relação entre uma ou mais variáveis independentes e uma variável dependente binária. A variável dependente binária pode assumir dois valores possíveis, geralmente codificados como 0 e 1, representando, por exemplo, uma classificação negativa ou positiva. Primeiramente, é criado uma combinação linear dessa variáveis e, em seguida, é aplicada a função sigmoide para transformar essa combinação em uma probabilidade entre 0 e 1. Por fim, essa probabilidade é usada para fazer uma classificação, geralmente comparando-a a um limiar, como 0,5. Se a probabilidade for maior que o limiar, o modelo prevê uma classe; se for menor, prevê a outra.

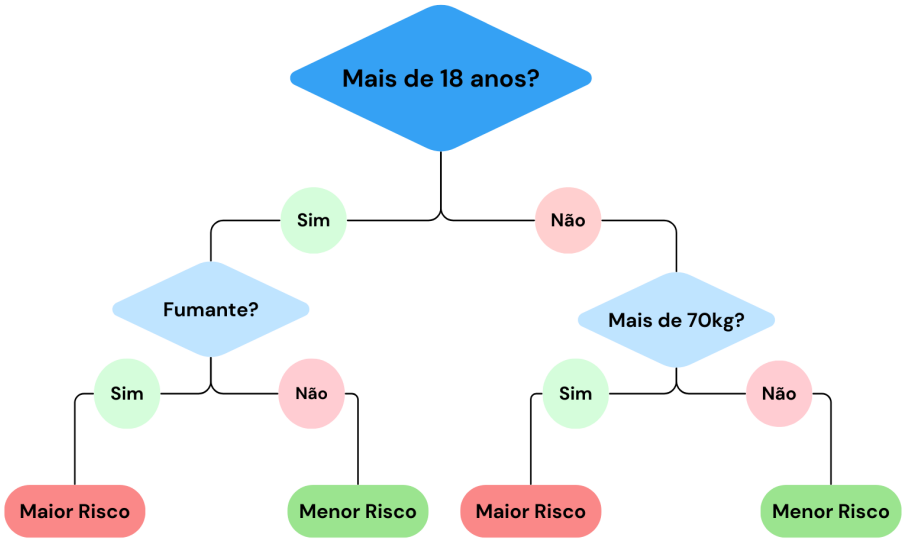
2.1.1.2 Árvore de Decisão

Árvore de Decisão (AD) é outro algoritmo de aprendizado de máquina utilizado para problemas de classificação e regressão. O algoritmo funciona construindo um modelo em forma de árvore a partir de um conjunto de dados, onde cada nó interno representa uma condição em uma variável de entrada, cada ramo representa o resultado dessa condi-

ção, e cada folha final representa a previsão de uma classe, no caso de uso em problemas de classificação.

De acordo com Breiman et al. (1984), o AD pode ser facilmente interpretado e visualizado, o que facilita a compreensão dos padrões e decisões tomadas pelo modelo. A Figura 2.1 ilustra o processo de decisão no algoritmo, onde cada nó faz uma pergunta específica, e o caminho seguido ao longo dos ramos leva à classificação final do problema.

Figura 2.1: Exemplo de uma Árvore de DEcisão classificando a chance de ataque cardíaco com base em critérios como idade, peso e hábito de fumar. Note que as informações apresentadas aqui têm o propósito exclusivo de ilustrar o funcionamento do algoritmo e não refletem dados reais ou baseados em evidências científicas.



Fonte: O Autor

2.1.1.3 Support Vector Machines

Support Vector Machines (SVM) são algoritmos de aprendizado de máquina utilizados principalmente para problemas de classificação. Eles funcionam encontrando o hiperplano que maximiza a separação entre as classes. Quando as classes não são linearmente separáveis, o SVM utiliza o "kernel trick" para mapear os dados para um espaço de maior dimensionalidade, onde a separação se torna possível. De acordo com Cortes e Vapnik (1995), essa abordagem permite ao SVM lidar eficazmente com dados complexos e de alta dimensionalidade (CORTES; VAPNIK, 1995).

Além disso, o SVM é conhecido por sua robustez e capacidade de evitar o sobreajuste, especialmente em casos onde as classes estão claramente separadas. Ele também pode ser adaptado a diferentes tipos de problemas através da escolha de funções de kernel, como a *Radial Basis Function* (RBF) e a polinomial. Essa flexibilidade torna o SVM uma escolha popular em diversas aplicações, desde bioinformática até processamento de texto.

2.1.1.4 Random Forest

O *Random Forest* (RF) é um algoritmo de aprendizado de máquina utilizado tanto para classificação quanto para regressão. Ele funciona criando um conjunto de Árvores de Decisão independentes a partir de diferentes subconjuntos do conjunto de dados original, e combina suas previsões para melhorar a precisão e reduzir o risco de sobreajuste. Breiman (2001) introduziu o conceito de RF, destacando sua capacidade de lidar com grandes conjuntos de dados com grandes números de características, o que torna o algoritmo altamente eficaz em diversos contextos.

2.1.2 Aprendizado Não Supervisionado

Ao contrário do aprendizado supervisionado, o aprendizado não supervisionado trabalha com dados não rotulados. O objetivo é descobrir padrões ou estruturas ocultas nos dados sem a orientação de rótulos pré-definidos. Uma tarefa comum de aprendizado não supervisionado é o agrupamento de dados (*clustering*), através de algoritmos como K-Means (MACQUEEN, 1967) e DBSCAN (ESTER et al., 1996). Este tipo de aprendizado é amplamente utilizado em tarefas como segmentação de mercado, compressão de dados e redução de dimensionalidade.

No presente trabalho, os algoritmos de agrupamento são utilizado para criar clusters a partir dos dados, que são então empregados na validação cruzada visando melhorar a representatividade dos folds. São explorados três desses algoritmos, que são: K-Means (e a sua variação Mini Batch K-Means), DBSCAN e Agglomerative Clustering. Cada um deles possui características distintas e aplicações específicas, o que os torna adequados para diferentes tipos de dados e problemas. A avaliação feita nesse trabalho favorece algoritmos que funcionem bem de modo geral, sem se ater às particularidades de cada um. Dessa forma, é esperado que o desempenho de alguns algoritmos se sobressaia a de

outros diante dos conjuntos de dados escolhidos, proporcionando uma visão abrangente e comparativa da eficácia de cada método.

2.1.2.1 K-Means

O K-Means é um dos algoritmos de agrupamento mais simples e amplamente utilizados. Ele divide um conjunto de dados em *K* clusters, onde cada ponto pertence ao cluster com o centróide mais próximo. O algoritmo segue os seguintes passos:

1. **Definição inicial de centróides:** Define o centro inicial dos K clusters, sendo K um parâmetro fornecido pelo usuário. A definição inicial pode ser tanto aleatória quanto pré-definida pelo uso de outros algoritmos.
2. **Cálculo da distância e atribuição de clusters:** Calcula-se a distância de cada ponto a cada centróide e, em seguida, cada elemento do conjunto de dados é atribuído ao cluster cujo centróide está mais próximo.
3. **Recálculo dos centróides:** Recalcula o centro dos clusters baseando-se nos elementos do próprio cluster.

Os passos 2, 3 e 4 são, então, repetidos até que não haja mais mudanças significativas de elementos e clusters entre as iterações.

Embora o K-Means seja eficiente para conjuntos de dados de tamanho moderado, sua eficiência diminui significativamente à medida que o tamanho do conjunto de dados aumenta. Isso ocorre porque o algoritmo precisa calcular a distância de cada ponto para todos os centróides em cada iteração, resultando em um alto custo computacional quando utilizado em grandes volumes de dados. Além disso, o K-Means pode ser sensível à inicialização dos centróides, exigindo múltiplas execuções para garantir um bom resultado, o que pode aumentar ainda mais o tempo de processamento.

2.1.2.2 Mini Batch K-Means

O Mini Batch K-Means é uma variação do algoritmo K-Means projetada para melhorar a eficiência computacional ao lidar com grandes conjuntos de dados. Em vez de usar o conjunto de dados completo em cada iteração, o Mini Batch K-Means processa pequenos subconjuntos aleatórios dos dados, chamados de mini-batches. Em cada iteração, um mini-batch é selecionado aleatoriamente do conjunto total de dados, e cada ponto do mini-batch é atribuído ao centro de cluster mais próximo. Os centros dos clusters são

então atualizados com base nos pontos do mini-batch, e esse processo é repetido com novos mini-batches até que os centros dos clusters se estabilizem ou um número máximo de iterações seja alcançado.

As principais vantagens do Mini Batch K-Means incluem maior velocidade e escalabilidade. Ao processar apenas uma pequena fração dos dados em cada iteração, o tempo de processamento é reduzido significativamente em comparação ao K-Means tradicional. Embora o Mini Batch K-Means possa ser ligeiramente menos preciso, ele geralmente converge mais rapidamente, alcançando um balanço entre precisão e eficiência (SCULLEY, 2010).

2.1.2.3 DBSCAN

O DBSCAN (*Density-Based Spatial Clustering of Applications with Noise*) é um algoritmo de agrupamento baseado em densidade que pode detectar clusters com formatos complexos e identificar pontos de ruido no conjunto de dados. Ele recebe como parâmetros os valores ε, o raio de busca ao redor de cada ponto, e *min_samples*, o número mínimo de pontos necessários para formar um cluster, e funciona da seguinte maneira:

- **Identificação de vizinhos:** Para cada ponto, identifica os pontos vizinhos dentro de um raio ε;
- **Classificação dos pontos:** Classifica os pontos como núcleo (com pelo menos *min_samples* vizinhos), borda (não núcleo, mas vizinho de um ponto núcleo) ou ruído (não núcleo e sem vizinhos suficientes);
- **Formação dos clusters:** Cria os clusters conectando pontos núcleo e seus vizinhos.

DBSCAN é eficaz para formar cluster com dados dispostos de formas irregulares e robusto contra ruídos, mas a escolha adequada dos parâmetros ε e *min_samples* é fundamental para a boa performance do algoritmo. Um estudo realizado por Schubert et al. (2017) demonstrou que a eficácia do DBSCAN depende fortemente da escolha desses parâmetros, especialmente em datasets com densidades variáveis, onde a sensibilidade do ε pode resultar na detecção de clusters falsos ou na falha em identificar clusters reais. Eles também propuseram melhorias no algoritmo que adaptam o ε dinamicamente, resultando em uma melhor detecção de clusters em situações onde a densidade não é uniforme.

2.1.2.4 Agglomerative Clustering

Agglomerative Clustering é um algoritmo hierárquico que constrói uma árvore de clusters a partir de uma abordagem de baixo para cima. Este método é conhecido por sua simplicidade e eficácia na formação de clusters hierárquicos, que podem ser visualizados como dendrogramas. O processo é descrito por:

- **Inicialização:** Cada ponto é considerado um cluster individual.
- **Combinação de clusters:** Iterativamente combina os dois clusters mais próximos até que todos os pontos estejam em um único cluster ou até que um critério de parada seja atingido.

Existem diferentes métodos para definir a proximidade entre clusters, como ligação simples, ligação completa e ligação média. O algoritmo é flexível, permitindo a análise em diferentes níveis de granularidade, mas pode ser computacionalmente intensivo para grandes conjuntos de dados (MURTAGH; CONTRERAS, 2014).

2.2 Validação Cruzada

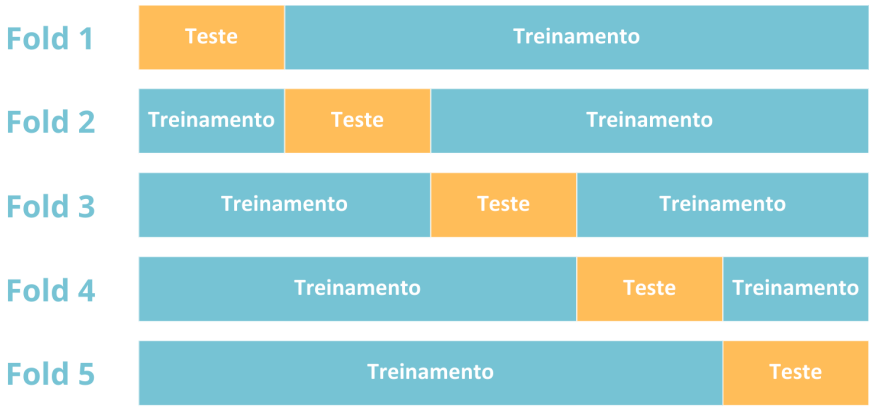
A validação cruzada é uma técnica utilizada para avaliar o desempenho de um modelo preditivo de forma robusta e reduzir o risco de overfitting, que ocorre quando o modelo se ajusta excessivamente aos dados de treinamento, levando a uma superestimação de seu desempenho e, consequentemente, à sua ineficácia em dados de teste (ou seja, dados ainda não vistos) (MALEKI et al., 2020). A técnica consiste em treinar vários modelos de AM em subconjuntos de dados de entrada disponíveis, avaliando-os no subconjunto complementar de dados. Existem diversas variações desta técnica de acordo com a forma que os subconjuntos de dados são criados. Algumas destas variações serão discutidas a seguir.

2.2.1 Validação Cruzada K-Fold

Dentre as técnicas de validação cruzada, a mais comumente usada é a validação cruzada K-fold, na qual o conjunto de dados é dividido em K partes, chamadas de folds. Então, o modelo é treinado em K-1 partes e testado na parte restante, e esse processo é repetido K vezes, com um fold diferente sendo utilizado para a validação em cada

iteração. Por fim, o desempenho do modelo é a média aritmética do desempenho de todas as iterações. A Figura 2.2 ilustra os processos de treino e teste com validação cruzada K-Fold, utilizando 5 folds.

Figura 2.2: Validação Cruzada K-Fold utilizando 5 folds.



Fonte: O Autor

No entanto, devido à aleatoriedade natural desse método na separação dos dados, os folds criados podem não representar de maneira adequada o conjunto de dados como um todo. Dessa forma, um fold pode conter uma quantidade de dados desproporcional de uma determinada classe em comparação ao conjunto de dados completo, enquanto em outros folds essas classes podem até mesmo estarem ausentes. Essa variação na composição dos folds tem o potencial de gerar forte instabilidade nos resultados da validação cruzada K-Fold, especialmente em conjuntos de dados desbalanceados, afetando a confiabilidade das métricas de desempenho do modelo. Portanto, é fundamental considerar estratégias adicionais para amenizar os efeitos dessa variabilidade e garantir uma representação mais consistente do conjunto de dados durante o processo de avaliação do modelo.

2.2.2 Validação Cruzada Estratificada por Classe

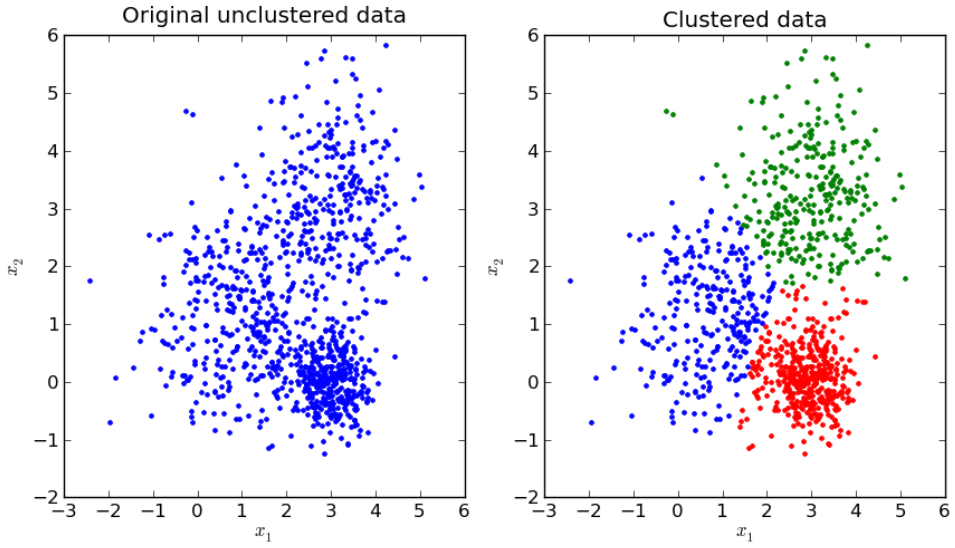
A Validação Cruzada Estratificada por Classe (*Stratified Cross-Validation*, SCV) é uma técnica especialmente útil quando se lida com conjuntos de dados desbalanceados, nos quais as classes têm proporções de instâncias muito distintas. Nessa abordagem, é

garantido que cada fold preserve a proporção original de instâncias de cada classe. Isso é crucial para evitar vieses na avaliação do modelo, pois permite que todas as classes sejam representadas tanto no conjunto de treinamento quanto no de teste em cada iteração do processo de validação.

2.2.3 Validação Cruzada baseada em Clusters

A separação em clusters, grupos de dados semelhantes, é uma técnica central no aprendizado não supervisionado, no qual não se tem o conhecimento de a qual classe cada dado pertence. Ao agrupar dados em clusters, os algoritmos de agrupamento ajudam a identificar e separar diferentes subgrupos dentro do conjunto de dados. Isso facilita a análise e interpretação dos dados, revelando padrões que podem não ser imediatamente aparentes e permitindo uma melhor compreensão das relações existentes entre as variáveis dos dados. A Figura 2.3 ilustra um exemplo de como um algoritmo de clustering, como o K-Means, pode segmentar um conjunto de dados em múltiplos clusters, agrupando elementos com características semelhantes.

Figura 2.3: Exemplo de aplicação de algoritmo de clustering em um conjunto de dados.



Fonte: Adaptado de (SCIENCE, 2021).

No contexto da validação cruzada, a divisão dos dados baseada em clusters tem sido proposta a fim de criar folds mais representativos do conjunto de dados original (DIAMANTIDIS; KARLIS; GIAKOUMAKIS, 2000). Os clusters são formados utilizando

algoritmos de agrupamento, como o K-Means, criando grupos distintos de dados semelhantes. Dessa forma, os dados de um mesmo cluster são atribuídos a folds diferentes na validação cruzada, na tentativa de proporcionar uma representação mais fiel do conjunto original em cada subconjunto, mesmo que o dataset seja desbalanceado. Alguns métodos de validação cruzada baseada em clusters serão discutidos em mais detalhes no Capítulo 3.

3 REVISÃO BIBLIOGRÁFICA

A validação cruzada e os algoritmos de agrupamento têm sido amplamente estudados na literatura, dada sua importância no desenvolvimento e avaliação de modelos de aprendizado de máquina. Diversos métodos foram propostos para melhorar a eficiência e a precisão desses processos, especialmente em contextos de grandes volumes de dados e de dados desbalanceados.

Neste capítulo, serão explorados trabalhos que, assim como este, visam aprimorar as técnicas de validação cruzada e agrupamento. A revisão abrange estudos que fazem uso tanto do aprendizado supervisionado quanto do não supervisionado e introduzem novos métodos de validação cruzada, como a validação baseada em clusters, e adaptações que melhoram a escalabilidade e a precisão, como o Mini Batch K-Means. Além disso, serão discutidos os avanços e as comparações entre diferentes algoritmos de agrupamento, destacando suas aplicações práticas e contribuições para a melhoria da análise de dados em aprendizado de máquina.

3.1 *Distribution-balanced Stratified Cross-Validation* (DBSCV)

Zeng e Martinez (2000) propuseram a técnica *Distribution-balanced Stratified Cross-Validation* (DBSCV) para abordar as limitações da validação cruzada tradicional em conjuntos de dados desbalanceados, visando garantir que cada fold preserve a distribuição de classes do conjunto de dados original. A técnica busca gerar folds que sejam representativos do conjunto de dados completo, atribuindo instâncias vizinhas a folds diferentes.

O processo começa com a seleção aleatória de uma instância, que é então atribuída a um fold. Em seguida, o algoritmo segue para a instância mais próxima da mesma classe e a atribui ao próximo fold, e esse procedimento é repetido até que todas as instâncias dessa classe tenham sido distribuídas pelos folds. O mesmo método é aplicado às demais classes, garantindo que cada fold contenha aproximadamente o mesmo número de instâncias por classe.

3.2 *Distribution Optimally Stratified Cross-Validation* (DOBSCV)

Moreno-Torres, Saez e Herrera (2012) propuseram a técnica *Distribution Optimally Balanced Stratified Cross-Validation* (DOBSCV), uma melhoria da *Distribution-Balanced Stratified Cross-Validation* (DBSCV) introduzida por Zeng e Martinez (2000). O objetivo da DOBSCV é abordar ainda mais as questões de enviesamento e variação na validação cruzada, assegurando que os folds gerados sejam não apenas balanceados em termos de classe, mas também otimizados para reduzir a interferência entre as distribuições de treino e teste. A DOBSCV busca garantir que cada fold seja uma representação mais fiel do conjunto de dados original, mitigando os efeitos do *dataset shift*, uma condição em que a distribuição dos dados de treino e teste difere, afetando negativamente a validação dos modelos.

A técnica de DOBSCV funciona semelhantemente à de DBSCV, iniciando com uma instância aleatória do conjunto de dados, mas em vez de seguir para a mais próxima da mesma classe, o DOBSCV encontra os (k-1) vizinhos, sendo k o número de folds, mais próximos da instância atual pertencentes à mesma classe e atribui cada um deles a um fold diferente. Esse processo é repetido independentemente para cada classe, de forma semelhante ao DBSCV, até que todas as instâncias tenham sido atribuídas a um fold.

3.3 *Unsupervised stratification of cross-validation for accuracy estimation*

Diamantidis, Karlis e Giakoumakis (2000) introduziram uma abordagem inovadora para a validação cruzada, visando resolver o problema da subrepresentatividade de classes em folds gerados aleatoriamente. Em seu artigo "Unsupervised stratification of cross-validation for accuracy estimation,"os autores propuseram a utilização de técnicas de agrupamento para melhorar a representatividade dos folds na validação cruzada. A principal ideia é dividir o conjunto de dados original em clusters utilizando algoritmos de agrupamento, como o algoritmo hierárquico Agglomerative Clustering e o K-Means, explorados no artigo. A formação desses clusters permite que os dados sejam organizados de maneira que pontos semelhantes sejam agrupados, facilitando a distribuição equilibrada entre os folds.

A técnica proposta começa com a aplicação de um algoritmo de agrupamento, como o K-Means ou o Agglomerative Clustering, para dividir o conjunto de dados em m clusters. Uma vez formados os clusters, os elementos de cada cluster são ordenados

pelas suas distâncias ao centro do cluster. Em seguida, esses elementos são distribuídos entre os k diferentes folds da validação cruzada, assegurando que instâncias adjacentes sejam atribuídas a folds diferentes. Esse processo garante que cada fold contenha uma representação mais diversificada e balanceada do conjunto de dados original, minimizando a possibilidade de que certos folds sejam dominados por instâncias de classes específicas. Dessa forma, a abordagem reduz o viés e proporciona uma avaliação mais precisa do desempenho dos modelos de aprendizado de máquina.

3.4 *Cross-validation Strategies for Balanced and Imbalanced Datasets*

Fontanari, Fróes e Recamonde-Mendoza (2022) propuseram uma adaptação do algoritmo de validação cruzada baseada em clusters utilizando o Mini-Batch K-Means, com o objetivo de reduzir o custo computacional em comparação ao K-Means tradicional. Originalmente, o algoritmo K-Means pode ser computacionalmente intensivo, especialmente em cenários com grandes conjuntos de dados, devido à necessidade de recalcular os centróides de todos os dados em cada iteração. Para resolver esse problema, o artigo explora o uso do Mini-Batch K-Means, que processa apenas uma pequena fração dos dados em cada iteração, reduzindo significativamente o tempo de execução e a carga computacional.

Os autores implementaram e compararam diversas estratégias de particionamento de dados, incluindo a validação cruzada estratificada balanceada por distribuição (DBSCV), a validação cruzada estratificada balanceada otimamente por distribuição (DOBSCV) e a validação cruzada baseada em clusters (CBDSCV). O principal destaque do estudo foi a adaptação do CBDSCV utilizando o Mini-Batch K-Means, que permitiu reduzir significativamente o tempo de execução, mantendo um desempenho similar ao método tradicional de CBDSCV. A abordagem de Mini-Batch K-Means seleciona apenas uma amostra dos dados em cada iteração, o que diminui o custo computacional sem comprometer significativamente a qualidade dos resultados obtidos.

Os experimentos conduzidos pelos autores avaliaram a eficácia das diferentes estratégias em conjuntos de dados de variados tamanhos e níveis de desbalanceamento de classes. Os resultados indicaram que as estratégias mais elaboradas de validação cruzada mostraram ganhos potenciais em cenários com um pequeno número de folds, mas a validação cruzada estratificada foi preferível em cenários com 10 folds ou em conjuntos de dados desbalanceados. A adaptação do CBDSCV com Mini-Batch K-Means se mostrou eficaz na redução do custo computacional, tornando-se uma alternativa viável para

situações onde a eficiência é crucial.

3.5 Resumo

Neste capítulo, foram revisadas diversas técnicas de validação cruzada, com foco em métodos que buscam melhorar a representatividade e a eficiência em cenários de dados desbalanceados. A *Distribution-balanced Stratified Cross-Validation* (DBSCV) e a sua otimização pela *Distribution Optimally Balanced Stratified Cross-Validation* (DOBSCV) mostraram-se eficazes em abordar as limitações da validação cruzada tradicional, assegurando que os folds preservem a distribuição original das classes. Além disso, técnicas que incorporam métodos de agrupamento, como as propostas por Diamantidis, Karlis e Giakoumakis (2000), demonstraram o potencial de melhorar a distribuição dos dados entre os folds, reduzindo o viés e proporcionando avaliações mais precisas. No entanto, essas abordagens também introduzem novos desafios, como a complexidade computacional e a escalabilidade, especialmente quando aplicadas a grandes volumes de dados.

Apesar dos avanços, ainda existem lacunas importantes na literatura, particularmente no que diz respeito ao equilíbrio entre a representatividade dos dados e a eficiência computacional. O trabalho de Fontanari, Fróes e Recamonde-Mendoza (2022) com o Mini-Batch K-Means oferece uma solução promissora para reduzir o custo computacional, mas novas adaptações e combinações de técnicas são necessárias para otimizar ainda mais esses processos. O presente trabalho focará em explorar e desenvolver novas estratégias que integrem métodos de agrupamento mais avançados com validação cruzada e estratificação por classe, buscando não apenas melhorar o desempenho e a robustez das validações cruzadas, mas também garantir que essas técnicas sejam escaláveis e aplicáveis a diferentes domínios.

4 METODOLOGIA

O principal objetivo deste trabalho é avaliar diversas técnicas de validação cruzada, com ênfase na comparação entre métodos tradicionais e aqueles baseados em algoritmos de agrupamento. A proposta deste trabalho inclui a implementação de uma nova técnica que combina K-Means e Mini Batch K-Means com estratificação por classe, buscando um equilíbrio entre desempenho e eficiência computacional. Neste capítulo, será abordada a metodologia utilizada nos experimentos realizados. A seguir, são detalhados os métodos de validação cruzada empregados nos experimentos, denominados conforme sua abordagem específica.

- **SCBCV** (*Stratified Cluster-Based Cross-Validation*): Técnica de validação cruzada proposta neste trabalho, que combina a estratificação por classe com o algoritmo de agrupamento K-Means;
- **SCBCV Mini**: Versão do SCBCV que utiliza Mini Batch K-Means;
- **KCBCV** (*K-Means Cluster-Based Cross-Validation*): Técnica de validação cruzada que utiliza o algoritmo K-Means na formação de clusters, sem a etapa de estratificação;
- **KCBCV Mini**: Versão do KCBCV que utiliza Mini Batch K-Means;
- **ACBCV** (*Agglomerative Cluster-Based Cross-Validation*): Técnica que utiliza o algoritmo de agrupamento Agglomerative Clustering na formação de clusters;
- **DBSCANBCV** (*DBSCAN-Based Cross-Validation*): Técnica que utiliza o algortimo de agrupamento DBSCAN na formação de clusters;
- **SCV** (*Stratified Cross-Validation*): Técnica de validação cruzada k-fold estratificada por classe, já consolidada na literatura.

Por fim, os experimentos propostos neste trabalho foram separados em 3 distintos conjuntos que são descritos abaixo:

- **1º Conjunto:** Comparação entre SCBCV utilizando como parâmetros o número de cluster variável definido para cada dataset e os valores fixos de 2, 3, 4 e 5 clusters;
- **2º Conjunto:** Comparação entre SCBCV, SCBCV Mini e SCV;
- **3º Conjunto:** Comparação entre os diferentes tipos de validação cruzada baseada em clusters abordados no trabalho – SCBCV, SCBCV Mini, KCBCV, KCBCV Mini, DBSCANBCV e ACBCV.

4.1 Datasets

A escolha dos datasets para este trabalho foi feita com base em suas características diversas e abrangentes dos dados disponíveis no PMLB (Penn Machine Learning Benchmarks) (OLSON et al., 2017). O PMLB é uma coleção extensa de datasets padronizados para avaliação e comparação de algoritmos de aprendizado de máquina. Os datasets selecionados variam em termos de número de classes, número de instâncias, número de atributos e desbalanceamento entre classes, permitindo realizar uma sólida análise das técnicas de validação cruzada dispostas anteriormente e verificar como elas se comportam com diferentes tipos de conjuntos de dados.

A Tabela 4.1 mostra a relação de datasets utilizados nos experimentos, divididos entre balanceados e desbalanceados. Essa divisão foi feita para que pudesse ser analisada a eficácia das técnicas de validação cruzada em cenários distintos de distribuição de classes, utilizando as métricas corretas para avaliar cada tipo. O desbalanceamento dos conjuntos de dados do PMLB é medido calculando o somatório da distância quadrada da proporção de instâncias de cada classe em relação ao equilíbrio perfeito no dataset, conforme a Equação 4.1, onde n_i é o número de instâncias da classe i, K é o número total de classes e N é o tamanho do dataset.

$$I = K \sum_{i=1}^{K} \left(\frac{n_i}{N} - \frac{1}{K} \right)^2 \tag{4.1}$$

Quanto maior o valor do índice I, maior é o desbalanceamento entre as classes, e I se aproxima de 1 quando quase todas as instâncias pertencem à mesma classe. Neste trabalho, datasets com um índice de desbalanceamento superior a $0,20$ foram classificados como desbalanceados (FONTANARI; FRÓES; RECAMONDE-MENDOZA, 2022). Exemplificando, o dataset *haberman* utilizado no experimento possui um índice de desbalanceamento de $0,22$ com duas classes, onde $26,5\%$ dos dados pertencem a uma classe e $73,5\%$ à outra, caracterizando um desbalanceamento considerável. Outro dataset utilizado, o *dis*, apresenta um índice de desbalanceamento de $0,94$ também com duas classes, onde $98,5\%$ das instâncias pertencem a uma classe e apenas $1,5\%$ à outra, exemplificando um desbalanceamento extremo e significativo.

Tabela 4.1: Lista de datasets utilizados no experimento, dividida entre datasets com distribuições balanceadas e desbalanceadas e em ordem crescente do número de instâncias.

Dataset	Instâncias	Atributos	Classes	Desbalanceamento	Clusters
Balanceados					
cloud	108	7	4	0,01	4
iris	150	4	3	0,00	4
analcatdata_germangss	400	5	4	0,00	4
movement_libras	360	90	15	0,00	5
sonar	208	60	2	0,00	4
vowel	990	13	11	0,00	4
contraceptive	1473	9	3	0,03	5
splice	3188	60	3	0,08	2
waveform_21	5000	21	3	0,00	4
optdigits	5620	64	10	0,00	5
Desbalanceados					
analcatdata_cyyoung9302	92	10	2	0,34	4
appendicitis	106	7	2	0,36	6
backache	180	32	2	0,52	5
new_thyroid	215	5	3	0,30	6
haberman	306	3	2	0,22	5
wine_quality_red	1599	11	6	0,23	5
allrep	3772	29	4	0,91	7
dis	3772	29	2	0,94	7
churn	5000	20	2	0,51	4
ann_thyroid	7200	21	3	0,79	7

4.2 Algoritmos de Classificação e Otimização de Hiperparâmetros

Para os experimentos, foram selecionado algoritmos de aprendizado supervisionado que apresentam diferentes níveis de viés e variância. Especificamente, foram utilizados os algoritmos já apresentados no Capítulo 2: Regressão Logística (RL), Árvore de Decisão (AD), Support Vector Machines (SVM) e Random Forest (RF).

O RL tende a ter um viés relativamente alto e uma variância baixa, pois assume uma relação linear entre as variáveis, o que pode simplificar demais a complexidade dos

dados (HASTIE; TIBSHIRANI; FRIEDMAN, 2009). Em contraste, o AD apresenta baixo viés, já que se ajusta fortemente aos dados de treinamento, mas sofre de alta variância, o que pode levar ao sobreajuste, especialmente quando as árvores são profundas (JAMES et al., 2013). O SVM equilibra viés e variância de acordo com a escolha do kernel; com um kernel linear, o viés tende a ser maior e a variância menor, enquanto kernels mais complexos, como o RBF, podem reduzir o viés, mas aumentar a variância (SCHöLKOPF; SMOLA, 2002). Neste trabalho, foi optado por utilizar o kernel padrão RBF, visto que funções de decisões lineares já serão representadas pelo RL. Por fim, o RF consegue reduzir a variância sem aumentar significativamente o viés ao combinar múltiplas AD (BREIMAN, 2001).

Antes dos experimentos, foi realizado um ajuste dos hiperparâmetros de cada algoritmo de aprendizado para cada conjunto de dados. Seguindo o que já foi estabelecido em trabalhos como o de Hsu, Chang e Lin (2003), foi utilizado Grid Search com 5 folds para determinar os melhores hiperparâmetros de cada algoritmo em cada um dos datasets. A métrica de precisão equilibrada, que é mais apropriada para lidar com conjuntos de dados balanceados e desbalanceados, foi utilizada, e os hiperparâmetros com a maior pontuação foram selecionados para o experimento.

Os hiperparâmetros otimizados foram os presentes na Tabela 4.2. Para o RL e o SVM, foi ajustado o hiperparâmetro C, que controla a regularização do modelo, com valores menores impondo maior regularização. No SVM, também foi ajustado o $gamma$, que determina a influência de cada ponto na função de decisão. Para o AD e o RF, ajustou-se o max_depth, que limita a profundidade das árvores, controlando a complexidade do modelo e prevenindo o sobreajuste. Os valores testados foram escolhidos para cobrir uma ampla gama de cenários, garantindo que tanto modelos simples quanto complexos fossem avaliados e permitindo a seleção dos hiperparâmetros mais adequados para diferentes características dos datasets.

Tabela 4.2: Hiperparâmetros otimizados e valores testados para cada algoritmo.

Algoritmo	**Hiperparâmetro**	**Valores Testados**
Regressão Logística (RL)	C	[0.003, 0.03, 0.3, 3.0, 30.0]
Support Vector Machines (SVM)	C	[0.3, 3.0, 30.0, 300.0]
	gamma	[0.00003, 0.0003, 0.003, 0.03, 0.3]
Random Forest (RF)	max_depth	[1, 5, 10, 15, 50]
Árvore de Decisão (AD)	max_depth	[1, 5, 10, 15, 50]

Esses hiperparâmetros, então, foram mantidos fixos durante todo o processo experimental, visando assegurar que qualquer variação no desempenho fosse atribuída exclusivamente à técnica de validação cruzada empregada, e não a ajustes subsequentes deles. Por sua natureza, o RF contém flutuações intrínsecas devido à aleatoriedade na seleção de amostras e na construção das árvores, o que pode impactar levemente os resultados desse algoritmo. Esse procedimento foi essencial para reduzir a influência externa dos hiperparâmetros sobre as medições, permitindo uma avaliação mais precisa das características de viés e variância de cada técnica de validação cruzada.

4.3 Algoritmos de Clustering e Definição dos Hiperparâmetros

Os algoritmos de clustering utilizados na validação cruzada nos experimentos requerem hiperparâmetros fornecidos como entrada. Enquanto o K-Means e o Agglomerative Clustering necessitam do número de clusters como entrada, o DBSCAN necessita dos parâmetros ε e $min\_samples$.

Dessa forma, anterior aos experimentos foram estimados os hiperparâmetros dos algoritmos de clustering, baseando-se em cálculos. Seguindo a mesma estratégia de Diamantidis, Karlis e Giakoumakis (2000), o número de clusters dos algoritmos K-Means e Agglomerative Clustering foram calculados com base na aplicação repetida de clustering hierárquico em pequenas amostras dos conjuntos de dados e na utilização de um limite na similaridade entre clusters sendo unidos para determinar o número de clusters. O número de clusters resultante para cada conjunto de dados é mostrado na Tabela 4.1.

Para o algoritmo DBSCAN, foi seguido o que sugere o trabalho de Sander et al. (1998). O parâmetro $min\_samples$, que define a quantidade mínima de dados vizinhos para ser considerado um cluster, foi definido como $2 *$ número de colunas do dataset. Já o ε, que define o raio entre os dados para serem considerados vizinhos, foi calculado através de um gráfico de distância entre os k-ésimos vizinho mais próximo de cada ponto em ordem decrescente, sendo *k = min_samples* - 1. O ponto em que há uma mudança significativa na inclinação do gráfico é onde se encontra o valor apropriado para o ε.

Nos algoritmos SCBCV Mini e KCBCV Mini, que utilizam o Mini Batch K-Means, o hiperparâmetro $batch\_size$ foi utilizado com o valor padrão de 1024. O valor do *batch size* determina o número de amostras que serão processadas em cada iteração do algoritmo, influenciando diretamente a eficiência computacional e a precisão do agrupamento. Um *batch size* maior permite processar mais dados por vez, o que pode acelerar a

convergência do algoritmo, mas também exige mais memória e pode levar a uma menor precisão em datasets menores.

4.3.1 Validação Cruzada K-Means Estratificada por Classe

A técnica de validação cruzada proposta neste trabalho, a validação cruzada utilizando K-Means com estratificação por classe (SCBCV), foi implementada com base em ambas as técnicas bases: validação cruzada K-Means e validação cruzada estratificada por classe. Seu pseudocódigo está presente no Algoritmo 1.

Em seus parâmetros de entrada, o SCBCV utiliza o conjunto de dados X, contendo as características a serem analisadas, e y, que representa os rótulos das classes correspondentes. O parâmetro k_splits define o número de divisões para a validação cruzada, enquanto $k\_clusters$ especifica a quantidade de clusters que o algoritmo K-Means deve formar. O parâmetro rng garante a reprodutibilidade dos experimentos por meio de um gerador de números aleatórios, e $minibatch\_kmeans$ indica se a versão otimizada Minibatch K-Means será empregada para lidar eficientemente com grandes volumes de dados.

O algoritmo SCBCV inicia organizando os dados de entrada X e as classes correspondentes y, ordenando e dividindo-os por classe (linha 2). Em seguida, nas linhas 3 a 7, o algoritmo verifica se deve utilizar o Mini Batch K-Means ou o K-Means tradicional, dependendo do parâmetro $minibatch\_kmeans$ fornecido, para agrupar os dados em $k\_clusters$. Após essa definição, o algoritmo procede iterando sobre cada classe (linha 9), aplicando o K-Means para transformar as instâncias da classe em vetores de distância aos centróides (linha 10) e obtendo os índices dos clusters para cada dado (linha 11). Nas linhas 13 a 16, são iterados todos os elementos da classe separando-os em clusters. Ele faz isso criando um $elemento$ para cada dado que terá o índice desse dado associado à sua distância ao centróide de seu cluster (linha 14). Esse elemento, então, é adicionado à lista de elementos de seu cluster, na linha 15, de acordo com o índice definido anteriormente. Cada cluster resultante é então ordenado pela distância dos elementos ao centróide (linha 18), e esses elementos ordenados são adicionados à lista de índices geral $index\_list$ (linha 19). Por fim, na linha 22, os elementos em $index\_list$ são distribuídos circularmente entre os k_splits para formar os diferentes folds de validação cruzada, retornando esses folds como saída do algoritmo na linha 23. Dessa forma, esse procedimento assegura que cada fold mantenha uma estrutura equilibrada de dados, com dados da mesma classe

Algoritmo 1 SCBCV

procedure SCBCV(X, y, k_splits, $k\_clusters$, rng, $minibatch\_kmeans$)
    $X, y \leftarrow$ ordena e divide X, y por classe
    if $minibatch\_kmeans$ **then**
        $kmeans \leftarrow$ **MiniBatchKMeans(**$k\_clusters$**,** rng**)**
    else
        $kmeans \leftarrow$ **KMeans(**$k\_clusters$**,** rng**)**
    end if
    $index\_list \leftarrow []$
    for classe em X **do**
        $class\_dist \leftarrow$ executa $kmeans$ e retorna matriz de distâncias de cada ponto a cada centróide
        $clusters\_index \leftarrow$ obtém índices dos clusters de cada elemento em $class\_dist$
        $clusters \leftarrow$ listas vazias para cada um dos $k\_clusters$
        for index em $class\_dist$ **do**
            $elemento \leftarrow$ (index, distância ao centróide)
            $clusters[clusters\_index[index]] \leftarrow elemento$
        end for
        for cluster em $clusters$ **do**
            ordena cluster por distância ao centróide
            $index\_list \leftarrow index\_list + cluster$ ordenado
        end for
    end for
    $folds \leftarrow$ distribui $index\_list$ entre k_splits $folds$ de forma circular
    return $folds$
end procedure

pertencentes ao mesmo cluster sendo distribuídos de acordo com a distância ao centróide, ou seja, de acordo com sua semelhança.

4.4 Métricas de Avaliação

Para avaliar o desempenho das técnicas de validação cruzada no experimento, é essencial utilizar métricas que forneçam uma compreensão completa da qualidade das estimativas produzidas. As métricas escolhidas neste trabalho foram o viés e a variância, seguindo outros trabalhos de cunho semelhante como o de Kohavi (1995) e o de Cawley e Talbot (2010). O viés mede a diferença entre a estimativa esperada do desempenho do modelo e o verdadeiro desempenho, permitindo avaliar o quanto a técnica de validação pode subestimar, quando seu valor é negativo, ou superestimar, quando o valor é positivo, a capacidade preditiva do modelo. Já a variância quantifica a sensibilidade da técnica de validação às variações nos dados de treinamento, indicando a estabilidade das estimativas obtidas. Ao combinar essas duas métricas, é possível ter uma visão equilibrada do desempenho das técnicas de validação cruzada testadas.

Para datasets balanceados, foi utilizada a acurácia como métrica de desempenho, que avalia o número de predições corretas sobre o número total de predições feitas, conforme a Equação 4.2. Esta métrica é confiável para conjuntos de dados em que as classes possuem quantidades similares de dados, entretanto, em casos desbalanceados, ela tende a não refletir adequadamente a capacidade do modelo de identificar exemplos da classe menos representada. Portanto, em datasets desbalanceados, foi utilizada a métrica de desempenho F1, que se dá conforme a Equação 4.3, preferível nesse caso por ser a média harmônica entre precisão e recall (ou sensibilidade).

$$\text{Acurácia} = \frac{VP + VN}{VP + VN + FP + FN} \tag{4.2}$$

$$\text{F1} = 2 \times \frac{\text{Precisão} \times \textit{Recall}}{\text{Precisão} + \textit{Recall}} \tag{4.3}$$

Na Equação 4.2, VP e VN se referem a Verdadeiros Positivos e Verdadeiros Negativos, que são os número de exemplos positivos que foram corretamente classificados como positivos pelo modelo, e de exemplos negativos classificados como negativos, respectivamente, enquanto FP e FN se referem a Falsos Positivos e Falsos Negativos, sendo esses os números de exemplos negativos classificados como positivos pelo modelo, e po-

sitivos classificados como negativos, respectivamente. Já na Equação 4.3, Precisão é a proporção de predições corretas para a classe positiva em relação a todas as predições feitas para essa classe, enquanto *Recall* é a proporção de instâncias da classe positiva que foram corretamente identificadas em relação a todas as instâncias que realmente pertencem à classe.

Como este trabalho utiliza datasets reais, é inviável obter o desempenho de teste verdadeiro. No entanto, é possível calcular as estimativas para ele usando um alto número de repetições de holdout estratificado, como feito por Budka e Gabrys (2013). Foram feitas 100 repetições dessa estratégia, com 90% do conjunto de dados sendo utilizado para treinamento, por fim obtendo a média entre elas. O conjunto de treinamento grande foi escolhido a fim de reduzir o viés que poderia ser causado caso fosse usado um conjunto pequeno, enquanto a variância tende a ser atenuada pelo alto número de repetições do holdout.

A estimativa esperada de cada técnica de validação cruzada foi calculada para cada dataset ao fazer uma amostragem de 90% do dataset 20 vezes, e aplicando a técnica de validação cruzada para obter as estimativas de desempenho verdadeiro. O valor médio das 20 estimativas foi utilizado como a estimativa esperada do desempenho a ser estimado pelo método de validação cruzada. Sendo assim, considerando CV_i a estimativa de desempenho ao executar a validação cruzada K-fold em um determinado conjunto de dados com uma das técnicas de validação cruzada, então a estimativa esperada da validação cruzada é conforme a Equação 4.4.

$$CV = \frac{1}{20} \sum_{i=1}^{20} CV_i \tag{4.4}$$

O viés é, então, calculado usando $b_{CV} = CV - \hat{P}$, onde $\hat{P}$ é a estimativa do desempenho verdadeiro que foi computada usando holdout estratificado repetido 100 vezes, conforme descrito acima.

Com relação à variância, seu cálculo foi feito com base na Equação 4.5. Neste trabalho, para fins de legibilidade, será utilizado o desvio padrão (s) ao invés da variância, visto que ambos são diretamente relacionados. Por fim, foram avaliados o viés e a variância das diferentes estratégias de validação cruzada em 20 diferentes conjuntos de dados e quatro algoritmos de aprendizado. Para cada estratégia de validação cruzada K-fold,

foram feitos experimentos com 2 e 10 folds.

$$s^2_{CV} = \frac{1}{20-1}\sum_{i=1}^{20}(CV_i - CV)^2 \tag{4.5}$$

Por fim, o custo computacional das técnicas de validação cruzada utilizadas nos experimentos foi calculado utilizando a função *perf_counter()* da biblioteca Python *time*. O tempo foi registrado antes e depois da execução de cada técnica de validação cruzada, permitindo uma medição precisa do tempo de execução e, consequentemente, do custo computacional associado a cada técnica testada.

4.5 Análise Estatística

Neste trabalho, foi utilizado o teste de Friedman para avaliar a significância estatística das diferenças de desempenho entre as diferentes técnicas de validação cruzada aplicadas nos experimentos. O teste de Friedman é um teste não-paramétrico, utilizado quando se deseja comparar três ou mais tratamentos ou algoritmos em condições semelhantes e os dados são medidas repetidas em blocos, como ocorre nos experimentos de validação cruzada (FRIEDMAN, 1940).

Para cada conjunto de experimentos, o teste de Friedman foi aplicado separadamente para as métricas de viés e desvio padrão. O teste avalia a hipótese nula de que todas as técnicas de validação cruzada têm o mesmo desempenho. Se o valor *p* resultante do teste for menor que um nível de significância predefinido (geralmente 0,05), a hipótese nula é rejeitada, indicando que pelo menos uma das técnicas tem desempenho significativamente diferente das outras.

4.6 Implementação Prática

A implementação dos algoritmos e experimentos foi desenvolvida utilizando a linguagem Python com a biblioteca Scikit-Learn, que oferece uma ampla gama de ferramentas para aprendizado de máquina e análise de dados. A configuração do ambiente incluiu a utilização de outras bibliotecas auxiliares, como NumPy para manipulação de arrays, Pandas para a gestão e manipulação dos dados, Scipy para efeturas os testes de Friedman e Matplotlib para a visualização dos resultados dos gráficos produzidos pela

biblioteca Kneed para a determinação dos parâmetros ε.

Para a realização dos experimentos de validação cruzada, foram empregados os algoritmos de clustering K-Means, Mini Batch K-Means, DBSCAN e Agglomerative Clustering, todos disponíveis na biblioteca scikit-learn. A escolha dos hiperparâmetros para esses algoritmos seguiu os métodos descritos anteriormente, e os valores foram ajustados conforme as características específicas de cada dataset utilizado. O código foi estruturado de forma modular, permitindo a replicação e modificação dos experimentos de maneira flexível, além de garantir a reprodutibilidade dos resultados. Todos os experimentos foram executados em uma máquina com múltiplos núcleos, utilizando paralelização para otimizar o tempo de execução, especialmente nas etapas que envolvem a aplicação repetitiva dos algoritmos de clustering e validação cruzada.

O framework utilizado para os experimentos foi baseado no trabalho de Fontanari, Fróes e Recamonde-Mendoza (2022), com alterações nas estruturas de paralelização, definição de parâmetros para os clusters, definição de datasets e experimentos feitos. O código deste trabalho encontra-se na íntegra em <https://github.com/amspezia/K-Fold-Partitioning-Methods>.

5 RESULTADOS E DISCUSSÃO

Os experimentos envolveram a aplicação das técnicas de validação cruzada a cada um dos 20 datasets e 4 algoritmos de aprendizado, resultando em um total de 80 amostras de viés e variância para cada técnica. O valor de k no método K-fold foi definido como 2 e 10, representando, respectivamente, um cenário com menor quantidade de folds, mais suscetível à variabilidade, e outro com uma quantidade maior de folds, que equilibra de forma mais eficaz viés e variância. Abaixo são descritos cada um dos conjuntos de experimentos definidos no Capítulo 4 e os resultados obtidos.

Os experimentos foram executados em um notebook equipado com processador 13th Gen Intel(R) Core(TM) i7-1355U 1.70 GHz, com 10 núcleos, e 16 GB de RAM. Para otimizar o desempenho, as execuções de cada validação cruzada foram paralelizadas entre todos os núcleos do processador, permitindo uma utilização eficiente dos recursos computacionais disponíveis. Os resultados foram então agregados para a análise final.

5.1 Conjunto 1

O principal objetivo com o experimento deste conjunto foi estabelecer como o número de clusters utilizado como parâmetro na técnica de validação cruzada SCBCV interfere nos resultados produzidos por ela. O número de clusters utilizado nesse algoritmo se difere das outras aplicações pois ele será executado intra-classe, e não no conjunto todo, combinando a estratificação por classe com o algoritmo de agrupamento K-Means.

Conforme exposto no capítulo anterior, os números de clusters utilizados nos algoritmos K-Means e Agglomerative Clustering foram calculados para cada dataset antes dos experimentos, baseando-se no trabalho de Diamantidis, Karlis e Giakoumakis (2000). Esse número, então, foi utilizado também na técnica SCBCV e comparado com a utilização dos valores fixos de 2, 3, 4 e 5 clusters. Essas execuções com diferentes números de clusters serão chamadas de SCBCV, para o valor variável de acordo com o dataset, e SCBCV2, SCBCV3, SCBCV4 e SCBCV5 para os valores fixos.

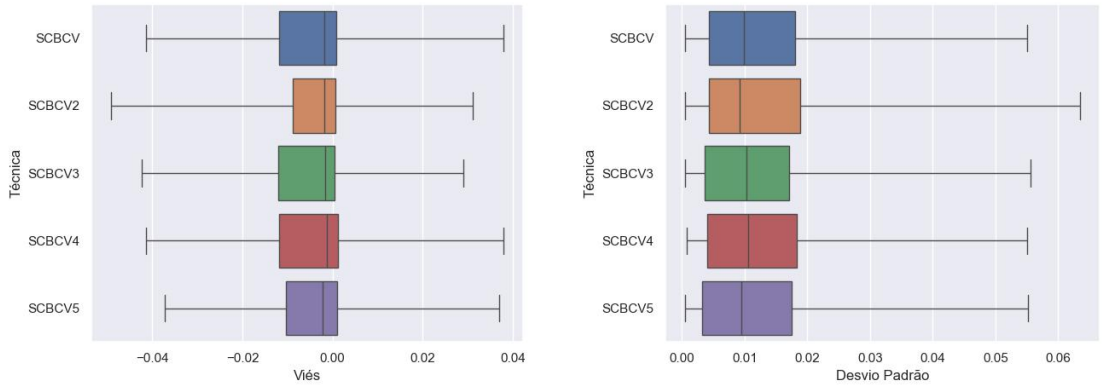
Os valores p dos testes de Friedman para o Conjunto 1, expostos na Tabela 5.1, indicam que, para os datasets e métricas testados, não houve diferenças estatisticamente significativas no desempenho do SCBCV quando diferentes números de clusters foram utilizados. Tanto para o viés quanto para o desvio padrão. Os resultados não mostraram variações significativas, com todos os valores p acima do limiar convencional de 0,05.

Tabela 5.1: Tabela de valores *p* de viés e desvio padrão dos testes de Friedman realizados no Conjunto 1 de experimentos.

Métrica	**Balanceamento**	**Folds**	**valor *p***	
			viés	**desvio padrão**
Acurácia	Balanceado	2	0.29901	0.89176
		10	0.20231	0.44294
F1	Desbalanceado	2	0.82845	0.92485
		10	0.86324	0.46102

A Figura 5.1 mostra o viés e o desvio padrão médio de cada técnica utilizando K-fold com k=10 em datasets balanceados. Nesses conjuntos, o SCBCV2 se destaca pelo viés, com uma média mais próxima de zero. Entretanto, apresenta o maior desvio padrão entre as técnicas analisadas. Em contrapartida, o SCBCV3 exibe um viés ligeiramente maior, porém com um desvio padrão inferior. No geral, o desempenho das técnicas é bastante semelhante quando considerados tanto o viés quanto o desvio padrão.

Figura 5.1: Viés (a) e Desvio Padrão (b) médio das técnicas SCBCV 10-fold no Conjunto 1 de experimentos, considerando datasets balanceados.

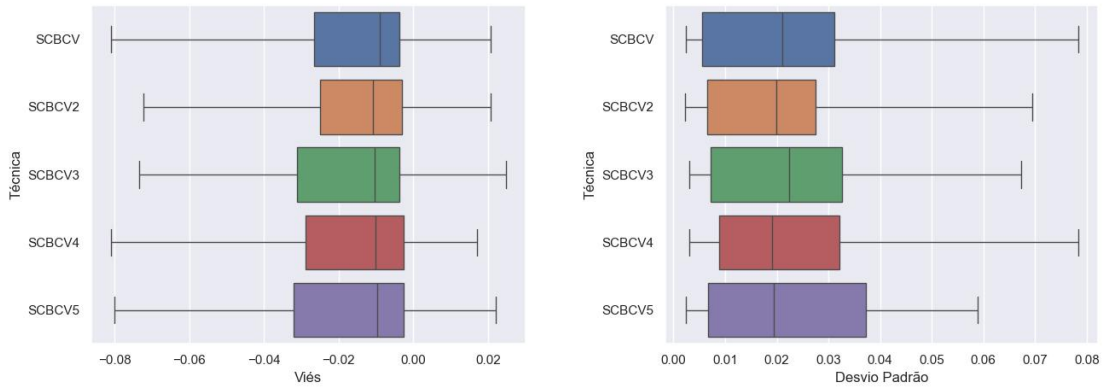


(a) (b)

Na Figura 5.2, o mesmo experimento 10-fold é aplicado a conjuntos desbalanceados. O SCBCV se destaca com a mediana de viés mais próxima de zero, seguido pelo SCBCV2 que apresenta a média de valores de desvio padrão mais próxima de zero, enquanto as demais técnicas apresentam valores mais elevados nessas métricas. Contudo, nos experimentos utilizando 2 folds, como mostrado nas Figuras 5.3 e 5.4, o SCBCV2 não mantém o mesmo desempenho, sendo o SCBCV a técnica mais estável em comparação com as outras.

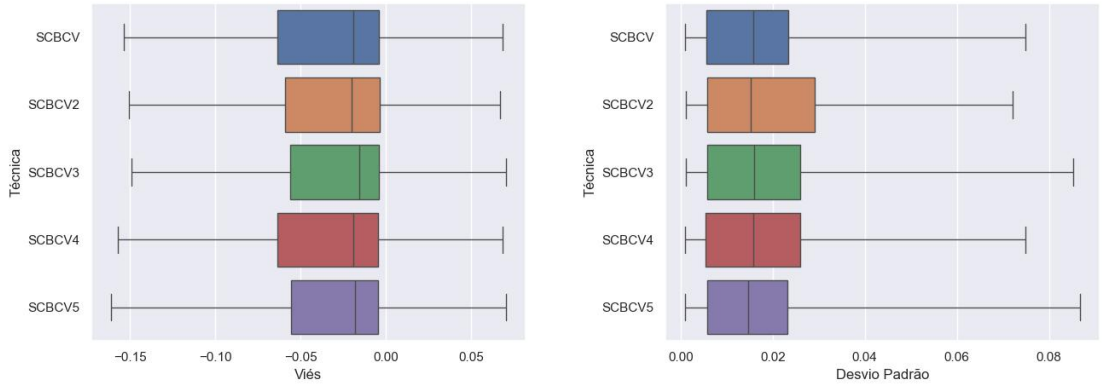
Com relação ao custo computacional, todas as execuções da técnica tiveram tem-

Figura 5.2: Viés (a) e Desvio Padrão (b) médio das técnicas SCBCV 10-fold no Conjunto 1 de experimentos, considerando datasets desbalanceados.



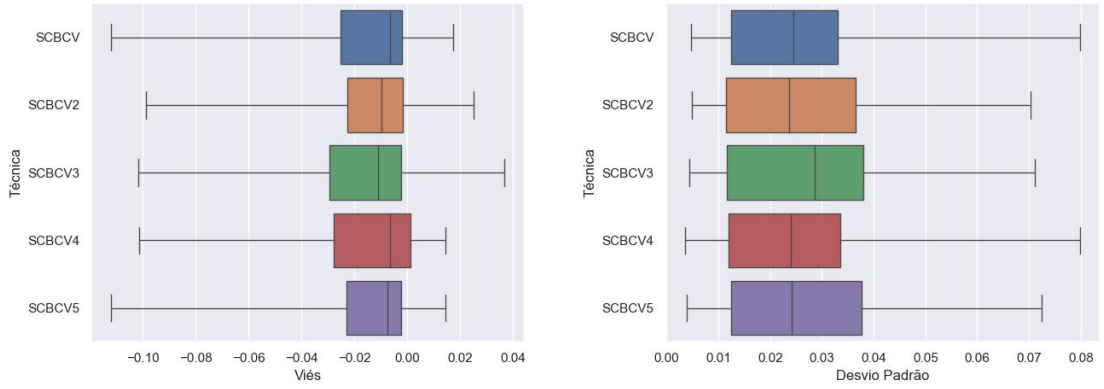
(a) (b)

Figura 5.3: Viés (a) e Desvio Padrão (b) médio das técnicas SCBCV 2-fold no Conjunto 1 de experimentos, considerando datasets balanceados.



(a) (b)

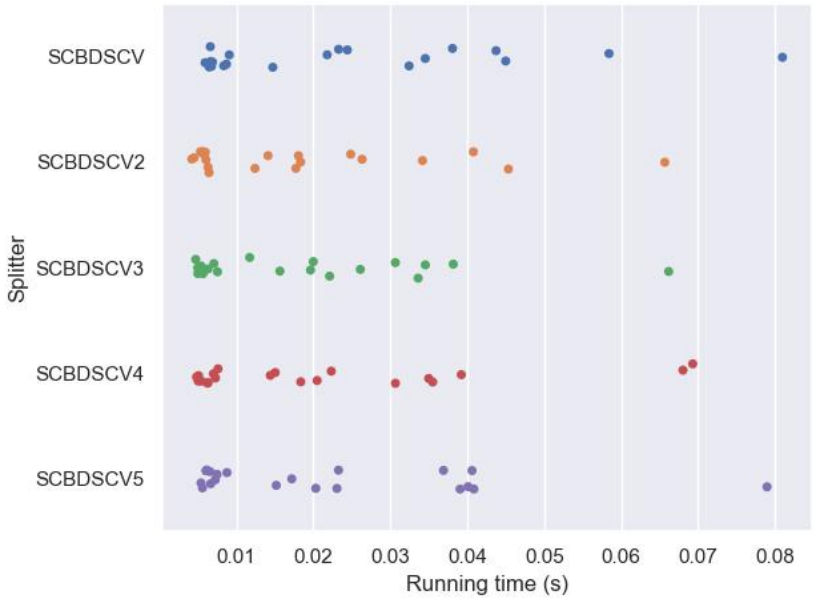
Figura 5.4: Viés (a) e Desvio Padrão (b) médio das técnicas SCBCV 2-fold no Conjunto 1 de experimentos, considerando datasets desbalanceados.



(a) (b)

pos de execução muito próximos. Quanto maior o número de clusters, maior o tempo médio de execução, como pode ser observado na Figura 5.5. A SCBCV com numero variável de clusters teve uma média de tempo próxima das demais, sendo levemente mais devagar, mas não a ponto de ser realmente considerada uma desvantagem significativa. Em resumo, o impacto do número de clusters no tempo de execução da técnica SCBCV é perceptível, mas muito pequeno, o que reforça a viabilidade do uso dessa técnica, mesmo em cenários onde o custo computacional é uma preocupação.

Figura 5.5: Tempo de execução geral das técnicas do Conjunto 1 para 20 datasets selecionados, utilizando 10 folds.



No geral, corroborando os testes de Friedman, os gráficos das figuras sugerem que a técnica SCBCV é relativamente insensível ao número de clusters utilizado, seja ele determinado pelo método de Diamantidis, Karlis e Giakoumakis (2000) ou fixado em 2, 3, 4 ou 5 clusters. Embora existam variações, elas são mínimas, com uma técnica ocasionalmente apresentando vantagem sobre a outra.

Essa insensibilidade indica que o algoritmo pode ser utilizado de forma flexível, sem a necessidade rigorosa de otimizar o número de clusters para cada dataset. A robustez observada em relação ao número de clusters reflete a estabilidade do algoritmo. Como a técnica SCBCV com o número de clusters variável pré-definido de acordo com o dataset demonstrou-se a mais estável nas diversas aplicações, ela foi selecionada para os experimentos subsequentes.

5.2 Conjunto 2

Este conjunto de experimentos visa comparar a técnica SCBCV, utilizando K-Means tradicional, com a técnica SCBCV Mini, utilizando Mini Batch K-Means, e a já consolidada Validação Cruzada Estratificada, aqui referida como SCV. O experimento nesse conjunto foi realizado da mesma forma como o anterior, utilizando cada um dos 20 datasets da Tabela 4.1 com cada um dos 4 algoritmos de aprendizado apresentados.

A análise de Friedman realizada sobre os dados obtidos no conjunto 2 de experimentos, presente na Tabela 5.2, sugere que, para a métrica de acurácia em conjuntos balanceados, os valores de viés e desvio padrão dos testes com 2 e 10 folds não indicam diferenças estatisticamente significativas, visto que todos os valores de p são maiores que 0,05. Isso sugere uma relativa estabilidade no desempenho das técnicas de validação cruzada para esses conjuntos, independente do número de folds.

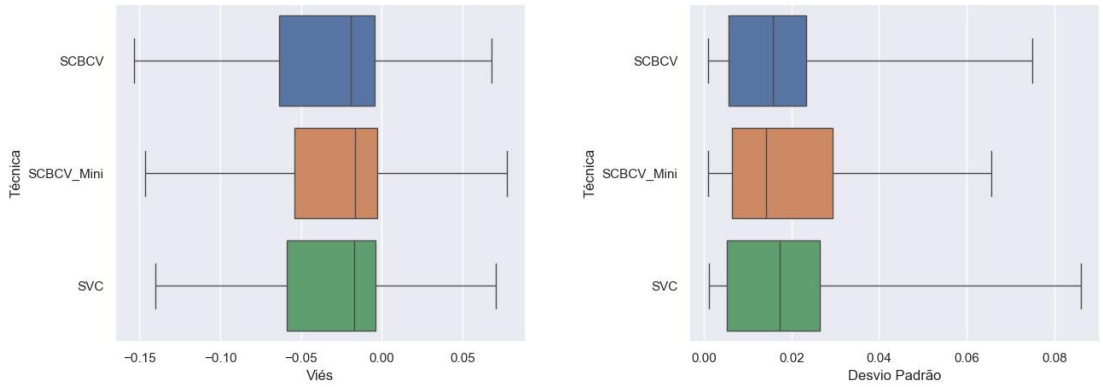
Tabela 5.2: Tabela de valores p de viés e desvio padrão dos testes de Friedman realizados no Conjunto 2 do experimento.

Métrica	**Tipo de Dataset**	**Folds**	**valor *p***	
			viés	**desvio padrão**
Acurácia	Balanceado	2	0.09778	0.07261
		10	0.90484	0.74082
F1	Desbalanceado	2	0.20190	0.01357
		10	<0.00001	0.01488

Para a métrica F1 em datasets desbalanceados, os valores de p indicam uma diferença mais pronunciada entre as execuções com 2 e 10 folds. Com valores de p próximo de zero para o viés com 10 folds e inferiores a 0,02 para o desvio padrão tanto com 2 quanto com 10 folds, o teste de Friedman revela uma diferença estatisticamente significativa nesses conjuntos de dados. Dessa forma, esse resultado sugere que tanto a estimativa do viés quanto a do desvio padrão são influenciadas de acordo com a técnica de validação usada.

Na Figura 5.7 é possível verificar a semelhança do desempenho das técnicas tanto em termos de viés quanto de desvio padrão para datasets balanceados utilizando-se uma validação cruzada com 2 folds. Nesse cenário, o SCBCV Mini se destaca por ter a mediana mais próxima de zero para o viés e o mennor valor de mediana no desvio padrão. O SCV, em contrapartida, enquanto possui a menor dispersão de valores de viés, possui a maior dispersão de valores de desvio padrão.

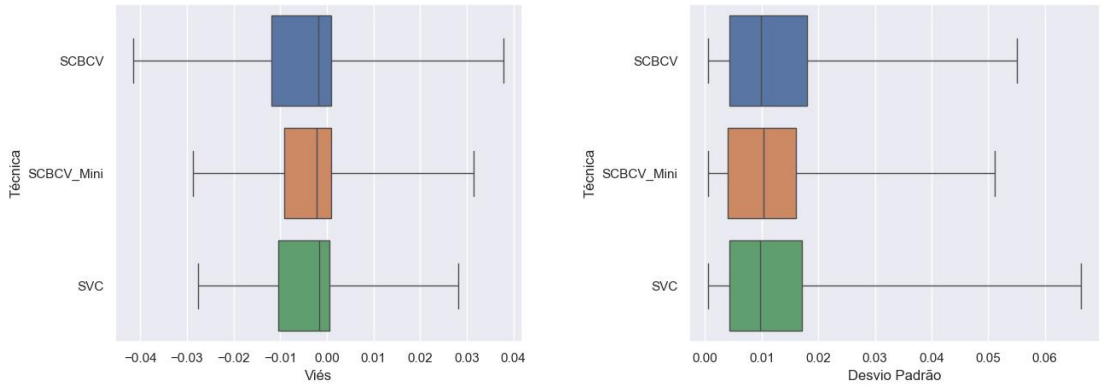
Figura 5.6: Viés (a) e Desvio Padrão (b) médio das técnicas SCBCV, SCBCV Mini e SCV 2-fold no Conjunto 2 de experimentos, considerando datasets balanceados.



(a) (b)

Em relação a datasets balanceados utilizando 10 folds, na Figura 5.7, o SCV possui uma mediana de viés e desvio padrão ligeiramente menor que o SCBCV Mini, porém a maior concentração de valores em suas métricas é mais distante de zero. O SCBCV demonstra a maior dispersão de valores de viés, se demonstrando menos competitivo quando comparado com sua versão utilizando Mini Batch.

Figura 5.7: Viés (a) e Desvio Padrão (b) médio das técnicas SCBCV, SCBCV Mini e SCV 10-fold no Conjunto 2 de experimentos, considerando datasets balanceados.

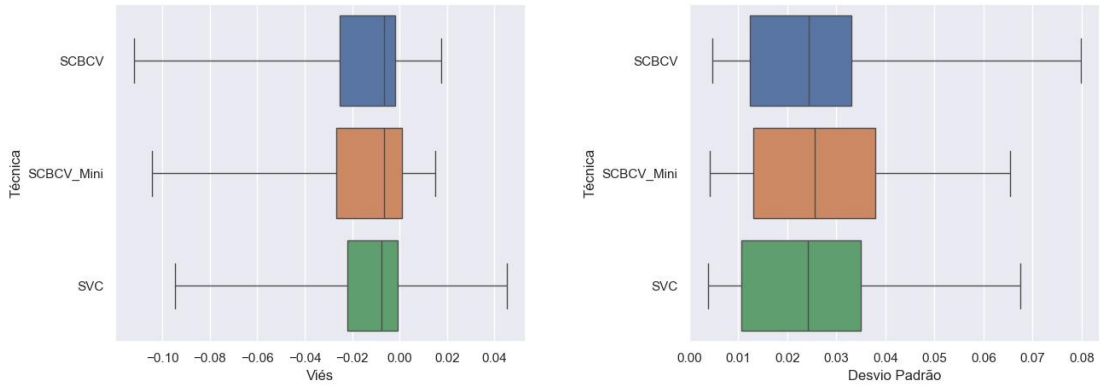


(a) (b)

Em datasets desbalanceados, a diferença entre as técnicas se torna mais evidente. Na Figura 5.8, o SCV se destaca levemente possuindo valores de desvio padrão mais próximos de zero quando utilizado com 2 folds, mas também valores de viés mais positivos, sinalizando que ele tende a superestimar o desempenho dos modelos mais que os métodos baseados em clusters. No entanto, o destaque da técnica perante as outras se intensifica fortemente ao se utilizar 10 folds, conforme é possível observar na Figura 5.9. As medianas tanto do viés quanto do desvio padrão do SCV são visivelmente mais próximas de

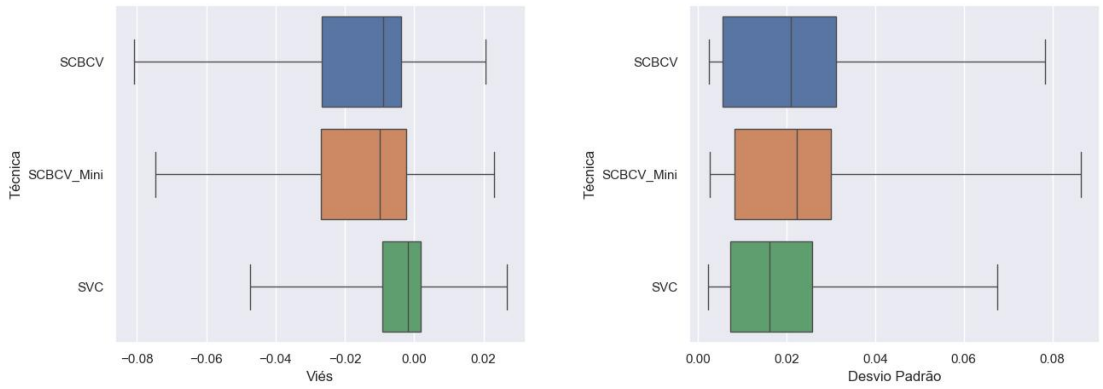
zero em comparação com as técnicas SCBCV, além de exibirem uma faixa de variação consideravelmente menor.

Figura 5.8: Viés (a) e Desvio Padrão (b) médio das técnicas SCBCV, SCBCV Mini e SCV 2-fold no Conjunto 2 de experimentos, considerando datasets desbalanceados.



(a) (b)

Figura 5.9: Viés (a) e Desvio Padrão (b) médio das técnicas SCBCV, SCBCV Mini e SCV 10-fold no Conjunto 2 dos experimentos, considerando datasets desbalanceados.



(a) (b)

A Tabela 5.3 destaca quantas vezes cada técnica de validação obteve o melhor desempenho para cada tipo de dataset e métrica. Os resultados confirmam que a técnica SCV se sobressai em datasets desbalanceados, alinhando-se com a análise preliminar dos testes de Friedman e com as observações dos gráficos. Já em datasets balanceados, embora segundo os testes de Friedman as diferenças entre as técnicas sejam estatisticamente insignificantes, os gráficos revelam uma ligeira vantagem da SCBCV Mini em relação às demais, o que é corroborado pelos dados da Tabela 5.3.

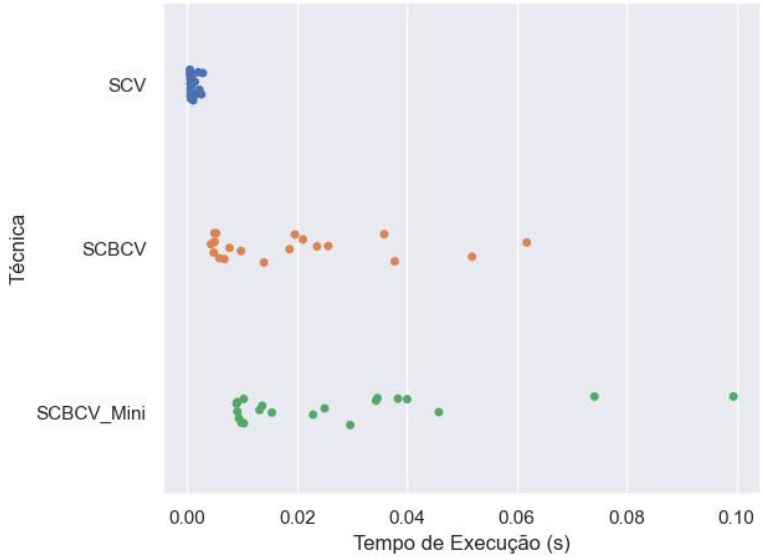
Por fim, a Figura 5.10 apresenta o tempo de execução de cada técnica nos diferentes datasets. O SCV se destaca como a técnica mais rápida, confirmando os resultados de estudos anteriores que já compararam o SCV à validação cruzada baseada em clusters

Tabela 5.3: Número de vezes em que cada técnica de validação testada no Conjunto 2 teve o melhor resultado em termos de viés e desvio padrão. A técnica vencedora de cada linha está destacada.

Métrica	Tipo de Dataset	Folds	Medida	SCBCV	SCBCV Mini	SCV
Acurácia	Balanceado	2	Viés	13	**16**	11
			Desvio Padrão	16	**19**	5
		10	Viés	9	**16**	15
			Desvio Padrão	**15**	14	11
F1	Desbalanceado	2	Viés	**18**	11	11
			Desvio Padrão	14	6	**20**
		10	Viés	5	5	**30**
			Desvio Padrão	15	6	**19**

(FONTANARI; FRÓES; RECAMONDE-MENDOZA, 2022). No entanto, de forma contrária ao esperado antes da execução do experimento, a técnica SCBCV utilizando Mini Batch K-Means mostrou-se mais lenta que a SCBCV com K-Means tradicional. Esse fato sugere que o uso de Mini Batch K-Means, embora teoricamente mais eficiente em termos de processamento para grandes volumes de dados, pode não oferecer as mesmas vantagens de desempenho em todos os cenários.

Figura 5.10: Tempo de execução geral das técnicas do Conjunto 2 nos 20 datasets utilizando 10 folds.



Os resultados obtidos nesse conjunto de experimentos sugerem que, enquanto o SCV é a escolha mais robusta para cenários desbalanceados, o SCBCV Mini pode ser considerado uma alternativa eficaz em datasets balanceados. A escolha da técnica de validação deve, portanto, ser orientada pelo tipo de dataset e pela métrica de desempenho mais relevante para o modelo em questão.

5.3 Conjunto 3

Este conjunto de experimentos tem como objetivo comparar diferentes técnicas de validação cruzada baseadas em clusters, especificamente SCBCV, SCBCV Mini, KCBCV, KCBCV Mini, ACBCV e DBSCANBCV. A escolha dessas técnicas permite avaliar não apenas a eficácia do K-Means tradicional e do Mini Batch K-Means, como explorado nos conjuntos anteriores, mas também expandir a análise para outros algoritmos de agrupamento, como Agglomerative Clustering (ACBCV) e DBSCAN (DBSCANBCV). Essa comparação visa avaliar o impacto da escolha do algoritmo de agrupamento sobre a estimativa de desempenho, e determinar se algum método oferece uma evidente melhor relação entre viés, variância e tempo de execução, em cenários com diferentes tipos de distribuição e complexidade dos dados.

A Tabela 5.4 mostra os valores *p* dos testes de Friedman para os dados de viés e desvio padrão obtidos no Conjunto 3. Ela evidencia que, na maior parte das métricas utilizadas, não existe diferença estatisticamente significativa entre as técnicas. No entanto, em datasets desbalanceados com 10 folds, o valor *p* 0,00422 do viés indica que a escolha da técnica tende a impactar nos resultados para esta métrica.

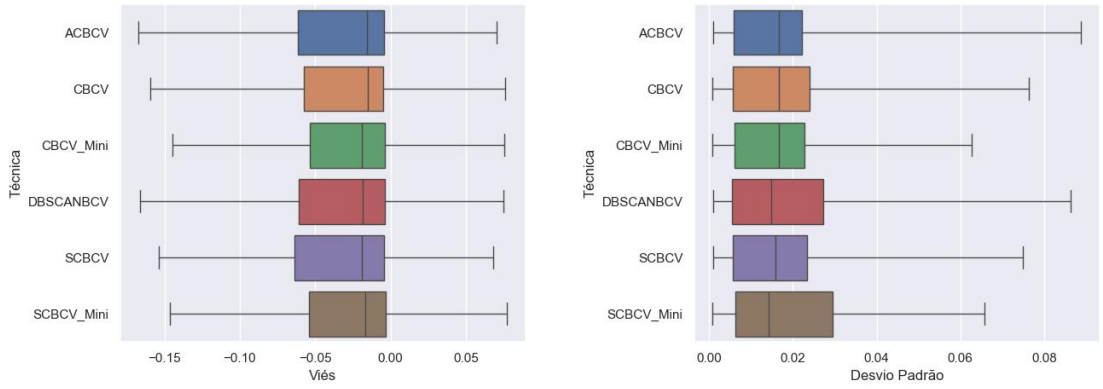
Tabela 5.4: Tabela de valores *p* de viés e desvio padrão dos testes de Friedman realizados no Conjunto 3 de experimentos.

Métrica	**Tipo de Dataset**	**Folds**	**valor *p***	
			viés	**desvio padrão**
Acurácia	Balanceado	2	0.24267	0.88889
		10	0.31952	0.09580
F1	Desbalanceado	2	0.15311	0.16582
		10	0.00422	0.13075

Na Figura 5.11 é possível visualizar que as diferentes técnicas de validação cruzada baseadas em clusters possuem viés e desvio padrão muito semelhantes em datasets balanceados utilizando 2 folds. Enquanto a KCBCV e a ACBCV possuem os viéses com medianas mais próximas de zero, a DBSCANBCV e a SCBCV Mini possuem o desvio padrão com menores medianas.

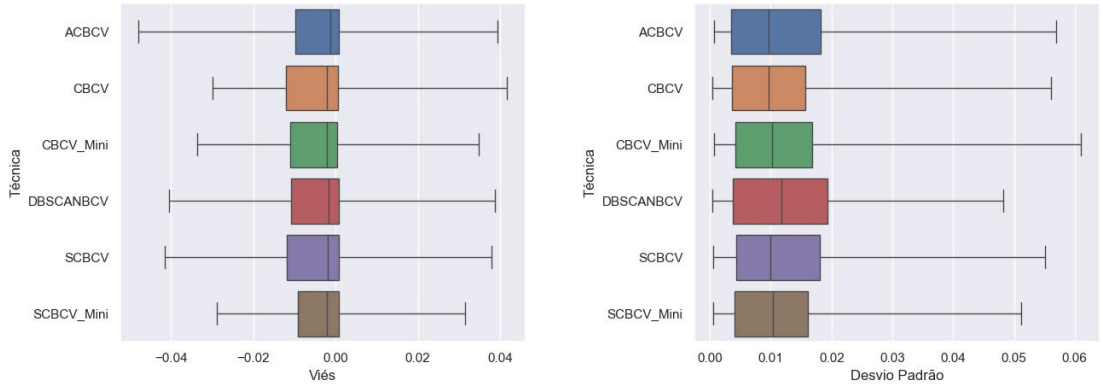
Quando são utilizados 10 folds em datasets balanceados, na Figura 5.12, o SCBCV Mini se destaca em ambas as métricas por possuir a maior parte de seus dados em intervalos mais próximos de zero em comparação às demais técnicas, porém as diferenças novamente são mínimas.

Figura 5.11: Viés (a) e Desvio Padrão (b) médio das técnicas de validação cruzada baseadas em clusters 2-fold no Conjunto 3 de experimentos, considerando datasets balanceados.



(a) (b)

Figura 5.12: Viés (a) e Desvio Padrão (b) médio das técnicas de validação cruzada baseadas em clusters 10-fold no Conjunto 3 de experimentos, considerando datasets balanceados.

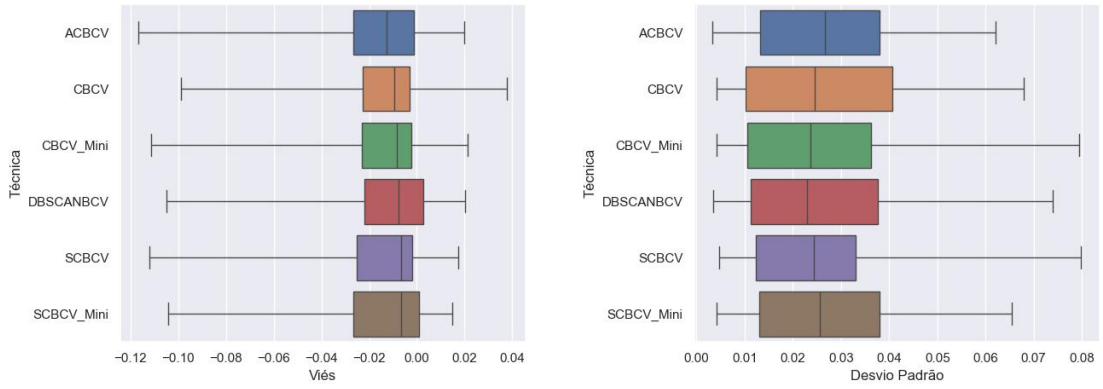


(a) (b)

Em datasets desbalanceados com 2 folds, na Figura 5.13, mais uma vez os valores das métricas são muito semelhantes. O destaque nesse caso é da técnica DBSCANBCV, que possui um viés com mais valores próximo de zero e o desvio padrão com a menor mediana. Por fim, no caso dos datasets desbalanceados utilizando 10 folds, a técnica DBSCANBCV aparenta se sobressair às demais com valores de viés e desvio padrão mais aproximados de zero. As técnicas SCBCV, no entanto, demonstram pior desempenho nessas condições. Porém, mais uma vez, é necessário considerar que as diferenças permanecem sutis entre as técnicas.

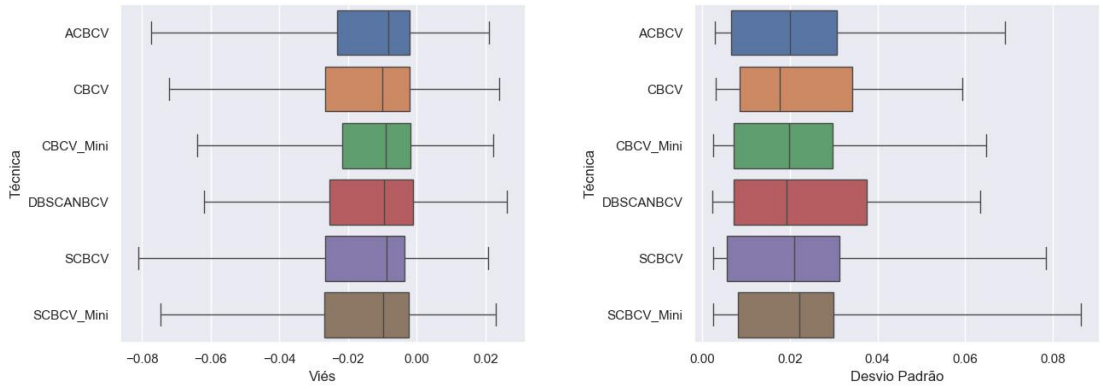
Por meio da Tabela 5.5, que mostra o número de vezes que cada técnica teve o melhor resultado em comparação às outras, é possível aprofundar a análise dos resultados. O método SCBCV não se destacou em nenhum dos cenários testados, enquanto o

Figura 5.13: Viés (a) e Desvio Padrão (b) médio das técnicas de validação cruzada baseadas em clusters 2-fold no Conjunto 3 de experimentos, considerando datasets desbalanceados.



(a) (b)

Figura 5.14: Viés (a) e Desvio Padrão (b) médio das técnicas de validação cruzada baseadas em clusters 10-fold no Conjunto 3 de experimentos, considerando datasets desbalanceados.



(a) (b)

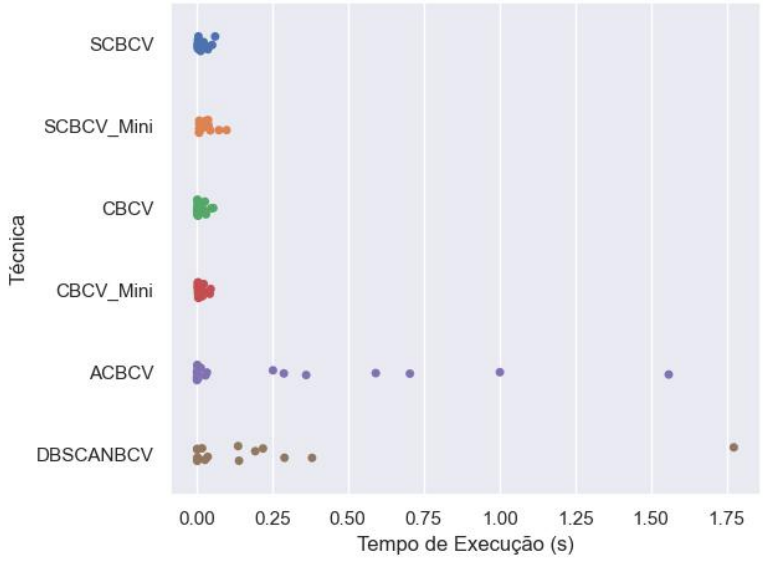
ACBCV obteve os melhores resultados em conjuntos balanceados. Nos desbalanceados, o KCBCV se sobressaiu quando utilizados 2 folds. Apesar de não determinísticos, os números presentes na tabela agregam para o melhor entendimento do desempenho dos algoritmos nos diferentes cenários de balanceamento e de quantidades de folds.

Em termos de custo computacional, a Figura 5.15 mostra o tempo de execução de cada técnica deste conjunto de experimentos nos 20 datasets. Os menores tempos são, respectivamente, das técnicas KCBCV, seguidos das técnicas SCBCV, e por fim a DBSCANBCV e ACBCV. Conforme o que se esperava, as técnicas que utilizam K-Means, algoritmo de agrupamento relativamente simples, foram mais rápidas que as demais envolvendo algoritmos mais complexos, como o DBSCAN e o Agglomerative Clustering.

Tabela 5.5: Número de vezes em que cada técnica de validação testada no Conjunto 3 teve o melhor resultado em termos de viés e desvio padrão. A técnica vencedora de cada linha está destacada. Foram abreviados Desbalanceados para Desb., Balanceados para Bal., Desvio Padrão para D.P., e foi removido o termo BCV (*Based Cross-Validation*), para melhor visualização dos dados.

				AC.	**KC.**	**KC. Mini**	**DBSCAN.**	**SC.**	**SC. Mini**
Acurácia	Bal.	2	Viés	9	4	6	3	7	**11**
		2	D. P.	**9**	3	**9**	**9**	5	5
		10	Viés	**12**	6	3	4	6	9
		10	D. P.	6	**11**	5	2	9	7
F1	Desb.	2	Viés	5	**11**	6	6	5	7
		2	D. P.	6	**10**	7	5	9	3
		10	Viés	5	5	**11**	8	8	3
		10	D. P.	9	7	5	**10**	7	2

Figura 5.15: Tempo de execução geral das técnicas do Conjunto 3 nos 20 datasets utilizando 10 folds.



6 CONCLUSÃO

Os experimentos conduzidos neste trabalho permitiram uma avaliação detalhada de diferentes técnicas de validação cruzada, considerando viés, variância e custo computacional. Embora nenhuma técnica se destaque fortemente em todos os cenários analisados, alguns padrões importantes emergem.

Em datasets balanceados, a técnica proposta por este trabalho, a validação cruzada estratificada utilizando Mini Batch K-Means (SCBCV Mini), apresentou desempenho superior em termos de viés e variância, mantendo valores mais próximos de zero em comparação com a validação cruzada estratificada (SCV) e com outras técnicas de validação utilizando algoritmos de agrupamento. Isso indica que ela é efetivamente eficaz em cenários onde a distribuição dos dados é equilibrada. No entanto, o ganho em desempenho não foi acompanhado por uma redução significativa no custo computacional, e em alguns casos, o SCBCV Mini foi até mais lento quando comparado com sua versão que utiliza o K-Means tradicional (SCBCV), contrariando expectativas iniciais.

Em contraste, em datasets desbalanceados, a SCV foi consistentemente superior. A técnica apresentou viés e variância mais baixos, especialmente em cenários com 10 folds, mostrando-se mais robusta e estável ao lidar com distribuições de classes desiguais. Entre as técnicas que envolvem algoritmos de agrupamento, o KCBCV se destacou levemente, mas seu desempenho não alcançou o nível do SCV. Além de ser uma técnica consolidada, o SCV também se sobressaiu pelo menor custo computacional, tornando-se a opção preferencial em cenários onde o desbalanceamento de classes é um fator crítico na avaliação do modelo.

Embora algumas das técnicas baseadas em clusters, como o ACBCV e o KCBCV, tenham demonstrado potencial em alguns contextos, ocasionalmente uma técnica apresentando vantagem sobre a outra, seus benefícios foram sutis. Isso sugere que o uso de algoritmos de agrupamento na validação cruzada deve ser cuidadosamente adaptado ao formato e às particularidades do conjunto de dados em análise, pois no cenário deste trabalho, em que se avaliou a capacidade de generalização das técnicas, não houve um algoritmo que se destacou de forma consistente como superior. Dessa forma, a seleção e configuração dos algoritmos devem ser feitas com atenção às características específicas de cada dataset para maximizar os benefícios da validação cruzada baseada em clusters.

Em resumo, os resultados indicam que, enquanto nenhuma técnica de validação cruzada baseada em clusters se destaca fortemente em todos os aspectos, o SCV per-

manece como a escolha mais equilibrada e eficiente em situações de desbalanceamento de classes. A técnica proposta por este trabalho, SCBCV Mini, destaca-se em relação ao SCV e aos outros algoritmos de agrupamento em cenários com conjuntos de dados balanceados, porém com um custo computacional levemente elevado. Esse custo computacional diferente do esperado pode ser justificado pelo tamanho do $batch\_size$ utilizado, um tamanho fixo de 1024 em todos os datasets, incluindo nos com menos instâncias, sendo essa uma limitação que pode ter influenciado nos resultados do algoritmo.

Para trabalhos futuros, sugere-se a investigação de métodos mais eficientes de otimização do número de clusters no algoritmo Agglomerative Clustering, considerando diferentes tipos de *linkage* e a possibilidade de determinar o melhor corte no dendrograma. Explorar a aplicação de técnicas de otimização mais avançadas, como *Hyperopt* ou *Optuna*, pode trazer ganhos significativos ao testar uma vasta gama de hiperparâmetros. Esses métodos podem complementar a validação cruzada com abordagens mais inteligentes de busca e ajuste fino dos parâmetros, garantindo uma avaliação mais robusta e precisa dos modelos.

Além disso, um estudo mais aprofundado sobre a complexidade computacional de cada etapa do algoritmo, particularmente no uso do Mini Batch K-Means, seria de grande importância. Isso incluiria a análise da relação entre o tamanho do batch e o desempenho do algoritmo em datasets de diferentes tamanhos, especialmente em cenários de Big Data. Também é recomendável realizar testes com datasets maiores que desafiem as limitações atuais das metodologias, fornecendo uma base mais sólida para o uso de técnicas robustas em situações de alta demanda computacional. A inclusão de métricas de avaliação de clusters, como o score de silhueta, e uma análise mais detalhada da distribuição entre os conjuntos de treino e validação, comparando as versões clusterizadas e não clusterizadas, pode oferecer percepções adicionais sobre a eficácia das estratégias de validação cruzada propostas.

REFERÊNCIAS

AGGARWAL, K. et al. Has the future started? the current growth of artificial intelligence, machine learning, and deep learning. **Iraqi Journal For Computer Science and Mathematics**, v. 3, n. 1, p. 115–123, Jan. 2022. Available from Internet: <https://journal.esj.edu.iq/index.php/IJCM/article/view/100>.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. 1st. ed. New York: Springer, 2006. ISBN 9780387310732.

BREIMAN, L. Random forests. **Machine learning**, Springer, v. 45, n. 1, p. 5–32, 2001.

BREIMAN, L. et al. **Classification and regression trees**. [S.l.]: CRC press, 1984.

BUDKA, M.; GABRYS, B. Density-preserving sampling: Robust and efficient alternative to cross-validation for error estimation. **IEEE Transactions on Neural Networks and Learning Systems**, IEEE, v. 24, n. 1, p. 22–34, 2013.

CAWLEY, G. C.; TALBOT, N. L. C. On over-fitting in model selection and subsequent selection bias in performance evaluation. **Journal of Machine Learning Research**, JMLR, v. 11, p. 2079–2107, 2010.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, n. 3, p. 273–297, 1995.

DIAMANTIDIS, N.; KARLIS, D.; GIAKOUMAKIS, E. Unsupervised stratification of cross-validation for accuracy estimation. **Artificial Intelligence**, v. 116, n. 1, p. 1–16, 2000. ISSN 0004-3702. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0004370299000946>.

ESTER, M. et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In: **Proceedings of the Second International Conference on Knowledge Discovery and Data Mining**. [S.l.: s.n.], 1996. p. 226–231.

FONTANARI, T.; FRÓES, T. C.; RECAMONDE-MENDOZA, M. Cross-validation strategies for balanced and imbalanced datasets. In: XAVIER-JUNIOR, J. C.; RIOS, R. A. (Ed.). **Intelligent Systems**. Cham: Springer International Publishing, 2022. p. 626–640.

FRIEDMAN, M. A comparison of alternative tests of significance for the problem of m rankings. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 11, n. 1, p. 86–92, 1940.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. [S.l.]: Springer Science & Business Media, 2009.

HSU, C.-W.; CHANG, C.-C.; LIN, C.-J. **A practical guide to support vector classification**. [S.l.], 2003. <https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>.

IKOTUN, A. M. et al. K-means clustering algorithms: A comprehensive review, variants analysis, and advances in the era of big data. **Information Sciences**, v. 622, p. 178–210, 2023. ISSN 0020-0255. Available from Internet: <https://www.sciencedirect.com/science/article/pii/S0020025522014633>.

JAMES, G. et al. **An Introduction to Statistical Learning: with Applications in R**. [S.l.]: Springer, 2013.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MORGAN KAUFMANN PUBLISHERS INC. **Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI)**. [S.l.], 1995. v. 2, p. 1137–1145.

MACQUEEN, J. B. Some methods for classification and analysis of multivariate observations. In: **Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1**. [S.l.: s.n.], 1967. p. 281–297.

MALEKI, F. et al. Machine learning algorithm validation: from essentials to advanced applications and implications for regulatory certification and deployment. **Neuroimaging Clinics**, Elsevier, v. 30, n. 4, p. 433–445, 2020.

MENARD, S. **Applied logistic regression analysis**. [S.l.]: Sage, 2002.

MOHRI, M.; ROSTAMIZADEH, A.; TALWALKAR, A. **Foundations of Machine Learning**. Cambridge, MA: MIT Press, 2012. ISBN 978-0262018258.

MORENO-TORRES, J. G.; SAEZ, J. A.; HERRERA, F. Study on the impact of partition-induced dataset shift on k-fold cross-validation. **IEEE Transactions on Neural Networks and Learning Systems**, v. 23, n. 8, p. 1304–1312, 2012.

MURTAGH, F.; CONTRERAS, P. Algorithms for hierarchical clustering: an overview. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 2, p. 86–97, 2014.

NICHOLS, J. A.; CHAN, H. W. H.; BAKER, M. A. B. Machine learning: applications of artificial intelligence to imaging and diagnosis. **Biophys Rev.**, v. 11, n. 1, p. 111–118, Feb 2019.

OLSON, R. S. et al. Pmlb: A large benchmark suite for machine learning evaluation and comparison. **BioData Mining**, v. 10, p. 36, 2017. Available from Internet: <https://doi.org/10.1186/s13040-017-0154-4>.

SANDER, J. et al. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. **Data Mining and Knowledge Discovery**, Kluwer Academic Publishers, v. 2, n. 2, p. 169–194, 1998.

SCHUBERT, E. et al. Dbscan revisited, revisited: Why and how you should (still) use dbscan. **ACM Transactions on Database Systems (TODS)**, v. 42, n. 3, p. 1–21, 2017.

SCHöLKOPF, B.; SMOLA, A. J. **Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond**. [S.l.]: MIT Press, 2002.

SCIENCE, T. D. **K-Means Data Clustering**. 2021. <https://towardsdatascience.com/k-means-data-clustering-bce3335d2203>. Acesso em: 28 ago. 2024.

SCULLEY, D. Web-scale k-means clustering. In: **Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. [s.n.], 2010. p. 595–603. Available from Internet: <https://dl.acm.org/doi/10.1145/1835804.1835877>.

ZENG, X.; MARTINEZ, T. R. Distribution-balanced stratified cross-validation for accuracy estimation. **Journal of Experimental & Theoretical Artificial Intelligence**, Taylor Francis, v. 12, n. 1, p. 1–12, 2000. Available from Internet: <https://doi.org/10.1080/095281300146272>.