Meta-learning for screening data with label noise

Matheus Bueno Figueiredo Lima[1], **Ana Carolina Lorena**[1]

[1]Instituto Tecnológico de Aeronáutica (ITA)
São José dos Campos, SP – Brasil

{figueiredombfl,aclorena}@ita.br

Abstract. *Supervised learning performance depends on the quality of the training data, and data-centric practice prioritizes reviewing suspect instances in triage workflows. Label noise impairs generalization, yet inspecting entire datasets manually is costly, motivating automated data-centric strategies for their detection. This work examines whether instance hardness measures can be used as input to a binary meta-classifier to detect potential label noise, and contrasts data-based with training-dynamics-based hardness descriptors. Experimentally, the meta-models using data-based hardness descriptors were able to identify approximately twice as many noisy instances as a random inspection under the same budget.*

Resumo. *O desempenho de modelos de aprendizado supervisionado depende da qualidade dos dados de treinamento, e práticas centradas nos dados priorizam a revisão de instâncias suspeitas nos fluxos de triagem. Erros e ruído de rotulagem comprometem a generalização, enquanto a inspeção manual de todo o conjunto é onerosa, o que motiva o uso de estratégias centradas em dados para a sua detecção. Este trabalho avalia se medidas de dificuldade de instância podem ser usadas como entrada para um meta-classificador binário para detectar potenciais ruídos nos dados, e contrasta descritores de dificuldade baseados nos dados com descritores baseados na dinâmica de treino. Experimentalmente, os meta-modelos com descritores de dificuldade baseados em dados foram capazes de identificar aproximadamente o dobro de instâncias ruidosas em comparação com uma inspeção aleatória, com o mesmo orçamento.*

1. Introdução

Segundo [Polyzotis et al. 2017], o desempenho de sistemas de Aprendizado de Máquina (AM) em produção depende não apenas do modelo utilizado, mas também da qualidade, da governança e da evolução dos dados empregados no treinamento desses sistemas. Erros de rotulagem, inconsistências e vieses podem comprometer a capacidade de generalização mesmo quando algoritmos sofisticados são utilizados [Frénay and Verleysen 2014]. Nesse contexto, cresce o interesse por abordagens que colocam os dados no centro do processo de desenvolvimento (*Data-Centric AI*), promovendo ciclos contínuos de curadoria e refinamento da rotulagem em paralelo à melhoria dos modelos [Zha et al. 2023].

Uma das linhas de pesquisa relacionadas a esse paradigma explora o conceito de **dificuldade de instâncias** (*instance hardness*), associado a padrões que podem indicar ambiguidades na fronteira de decisão ou, em alguns casos, problemas de qualidade dos rótulos [Smith et al. 2014]. Ferramentas como o PyHard[1] disponibilizam

[1]https://pypi.org/project/pyhard/.

medidas clássicas de dificuldade baseadas em metacaracterísticas extraídas dos dados [Paiva et al. 2022, Lorena et al. 2024], enquanto o H-CAT (*Hardness Characterization Analysis Toolkit*[2]) utiliza informações provenientes da dinâmica de treinamento de modelos iterativos, frequentemente redes neurais [Seedat et al. 2024].

Esta pesquisa investiga a viabilidade de utilizar vetores de medidas de dificuldade como entrada para meta-classificadores binários capazes de distinguir instâncias sem ruído de rótulo de instâncias com ruído de rótulo simulado, contribuindo para processos de triagem e curadoria de dados em ambientes *Data-Centric AI*. A hipótese central é que as medidas de *instance hardness* preservam informação suficiente para discriminar instâncias potencialmente problemáticas. Para atingir esse objetivo, foi desenvolvido um *pipeline* de meta-aprendizado para conjuntos de dados tabulares, utilizando como entrada medidas de dificuldade por instância e, como saída, um indicador de qualidade das observações. Em seguida, classificadores binários foram treinados e avaliados, utilizando exclusivamente essas medidas como atributos para prever a presença de problemas de qualidade nos dados. Experimentalmente, as medidas extraídas pelo PyHard foram capazes de identificar instâncias de perturbação de rótulo com maior eficácia.

Este artigo está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação e literatura correlata do artigo. A Seção 3 apresenta o desenvolvimento da proposta, cujos resultados experimentais são apresentados e discutidos na Seção 4. A Seção 5 conclui o artigo.

2. Fundamentação Teórica

Esta seção apresenta a fundamentação teórica da bibliografia correlata ao trabalho.

2.1. *Data-Centric AI*

A inteligência artificial centrada nos dados (*Data-Centric AI*) representa uma mudança de foco em relação à abordagem tradicional centrada no modelo (*Model-Centric AI*). Nesta última, o conjunto de treinamento é geralmente mantido fixo enquanto se exploram diferentes arquiteturas, funções de perda, técnicas de regularização e configurações de hiperparâmetros. Em contraste, a *Data-Centric AI* direciona os esforços para a melhoria contínua dos dados, envolvendo atividades como o aperfeiçoamento da rotulagem, a correção de exemplos ambíguos ou ruidosos e a atualização das bases de dados, frequentemente mantendo o modelo ou uma família restrita de modelos relativamente estável [Zha et al. 2023]. O princípio central dessa abordagem é que, quando o principal fator limitante do desempenho está na qualidade dos dados e dos rótulos, melhorias sistemáticas e iterativas no conjunto de treinamento podem produzir ganhos comparáveis ou até superiores aos obtidos apenas pela substituição ou pelo ajuste de algoritmos.

Na prática, a *Data-Centric AI* adota um processo cíclico de desenvolvimento: o modelo é treinado, seus erros são analisados, instâncias com rótulos suspeitos ou inconsistentes são identificadas e corrigidas, e o ciclo é repetido à medida que novos dados são coletados e incorporados ao sistema [Zha et al. 2023]. Nesse contexto, a curadoria dos dados passa a ter importância equivalente à modelagem, tornando essencial a adoção de mecanismos de triagem que permitam priorizar as instâncias mais relevantes para inspeção quando a revisão manual completa se torna inviável.

[2] `https://github.com/seedatnabeel/H-CAT`.

Zha et al. [Zha et al. 2025] organizam a área em três grandes objetivos: o desenvolvimento dos dados de treinamento, o desenvolvimento dos dados utilizados durante a inferência e a manutenção da qualidade dos dados ao longo de todo o ciclo de vida do sistema. Os autores destacam que a engenharia e a garantia da qualidade dos dados de treinamento, especialmente em cenários com ruído de rótulo, constituem fatores centrais para o desempenho dos modelos de AM.

Complementarmente, Jain et al. [Jain et al. 2024] demonstram que a priorização de instâncias difíceis em um contexto de meta-aprendizado pode ampliar as margens de decisão e melhorar a capacidade de generalização dos modelos. Esses resultados reforçam a importância da caracterização da dificuldade das instâncias não apenas para a identificação de possíveis problemas de qualidade, mas também para apoiar estratégias de curadoria de dados.

2.2. Rótulos incorretos

Em aplicações reais de aprendizado supervisionado, os rótulos associados às instâncias nem sempre correspondem à classe correta dessas instâncias. Esse tipo de erro pode surgir por diversas razões, incluindo falhas nos protocolos de coleta ou de anotação, ambiguidades clínicas ou operacionais, divergências entre especialistas, desatualização dos dados ou até mesmo modificações acidentais durante o processamento das informações.

Frénay e Verleysen [Frénay and Verleysen 2014] apresentam uma ampla revisão sobre o problema do ruído de rótulo (*label noise*) e propõem uma taxonomia baseada no mecanismo de geração dos erros. Os autores distinguem três cenários principais: NCAR (*Noisy Completely at Random*), em que a ocorrência do erro é independente tanto dos atributos quanto da classe verdadeira; NAR (*Noisy at Random*), em que a probabilidade de erro depende da classe; e NNAR (*Noisy Not at Random*), em que o erro também está relacionado às características da própria instância. Cada um desses cenários representa diferentes graus de complexidade na identificação e correção de rótulos incorretos.

Além de sistematizar os mecanismos de geração de ruído, os autores discutem seus efeitos nos modelos de aprendizado, destacando a degradação da capacidade de generalização, a distorção das estatísticas descritivas e o comprometimento dos processos de tomada de decisão baseados em dados. Como resposta a esses problemas, a literatura tem explorado estratégias como algoritmos robustos ao ruído, técnicas de limpeza de dados e abordagens que modelam explicitamente o processo de geração de erros [Frénay and Verleysen 2014].

A avaliação dessas estratégias frequentemente utiliza experimentos controlados, nos quais uma parcela dos rótulos do conjunto de treinamento é artificialmente modificada, mantendo-se o registro das instâncias alteradas. Essa abordagem permite analisar o comportamento dos métodos de detecção e de correção de ruído sem a necessidade de uma inspeção manual extensa. Os mecanismos de perturbação mais comuns incluem [Seedat et al. 2024]: (i) a perturbação *uniforme* (*uniform mislabeling*), na qual o novo rótulo é sorteado aleatoriamente entre as demais classes possíveis; (ii) a perturbação *assimétrica* (*asymmetric mislabeling*), que segue probabilidades de confusão específicas entre pares de classes; (iii) a perturbação *condicionada aos atributos* (*instance-specific mislabeling*), em que a probabilidade de erro depende da posição da instância no espaço de atributos, caracterizando um cenário NNAR; e (iv) a perturbação *adjacente* (*adjacent*

mislabeling), que restringe a troca a classes vizinhas em uma ordenação predefinida. Embora tais protocolos não reproduzam integralmente os erros observados em aplicações reais, tornam o fenômeno controlável e auditável.

2.3. *Instance Hardness*

O conceito de dificuldade de classificação ao nível da instância (*instance hardness*) foi introduzido por Smith et al. [Smith et al. 2014] como uma forma de compreender por que determinados exemplos são sistematicamente mais difíceis de classificar do que outros. Embora o desempenho dos algoritmos seja normalmente reportado por meio de métricas agregadas sobre todo o conjunto de dados, os erros tendem a concentrar-se em regiões específicas do espaço de atributos, como áreas próximas às fronteiras de decisão, vizinhanças heterogêneas ou zonas de sobreposição entre classes.

Os autores definem a dificuldade de uma instância em função da taxa de erro observada sobre ela por um conjunto diversificado de classificadores. Quanto maior a frequência de erros cometidos por diferentes algoritmos, maior a sua dificuldade intrínseca. Para explicar esse comportamento, foram propostas diversas medidas associadas à estrutura local dos dados. Os resultados experimentais indicam que a combinação dessas medidas fornece uma caracterização mais robusta da dificuldade do que qualquer indicador isolado, o que motiva o uso de vetores de *hardness* como entrada para modelos de meta-aprendizado [Smith et al. 2014, Lorena et al. 2024]. Paiva et al. [Paiva et al. 2022] disponibilizaram essas medidas na biblioteca PyHard [Lorena et al. 2024]. A Tabela 1 apresenta um resumo das medidas usadas neste trabalho.

Categoria	**Medida**	**Acrônimo**
Vizinhança	*k-Disagreeing Neighbors*	kDN
	Ratio of intra-extra class distances at instance level	$N2I$
	Local set cardinality at instance-level	$LSCI$
	Local set radius	LSR
	Harmfulness	H
Verossimilhança	*Class Likelihood*	CL
	Class Likelihood Difference	CLD
Baseadas em árvores	*Disjunct Class Percentage*	DCP
	Tree Depth (pruned)	TD_P
	Tree Depth (unpruned)	TD_U

Tabela 1. Sumário das medidas de dificuldade de instâncias usadas no trabalho, descritas em [Lorena et al. 2024].

Enquanto as medidas do PyHard são calculadas diretamente a partir da distribuição dos dados e de suas relações geométricas ou probabilísticas, abordagens mais recentes passaram a explorar informações provenientes do próprio processo de treinamento dos modelos. Nessa direção, Seedat et al. [Seedat et al. 2024] propuseram o H-CAT (*Hardness Characterization Analysis Toolkit*), um arcabouço voltado à caracterização sistemática de diferentes formas de dificuldade presentes nos dados, incluindo ruído de rótulo, exemplos atípicos e observações fora da distribuição (*out-of-distribution*). O H-CAT organiza e compara métodos de caracterização baseados em diferentes princípios, como a confiança preditiva, a dinâmica de treinamento, as propriedades

dos gradientes e as estatísticas dos dados. A Tabela 2 resume os métodos do H-CAT utilizados neste trabalho.

Categoria	Método	Acrônimo	Referência
Confiança	*Conf. agreement*	$ConfAgr$	[Jiang and Nachum 2020]
	Area Under the Margin	AUM	[Pleiss et al. 2020]
Dinâmica de treino	*Data IQ*	DIQ	[Seedat et al. 2022]
	Forgetting events	FE	[Toneva et al. 2019]
Gradiente	*Gradient Norm*	$GraNd$	[Paul et al. 2021]
	Error L2-Norm	$EL2N$	[Paul et al. 2021]
Estatísticas	*Prototypicality*	$Proto$	[Sorscher et al. 2022]

Tabela 2. Métodos de caracterização de *hardness* do H-CAT usados neste trabalho, conforme [Seedat et al. 2024].

3. Desenvolvimento

Esta seção descreve o *pipeline* experimental utilizado na pesquisa. Inicialmente, são apresentados os conjuntos de dados e o procedimento de inserção controlada de ruído nos rótulos. Em seguida, detalha-se a extração das medidas de *instance hardness* por meio das ferramentas PyHard e H-CAT, bem como a construção dos meta-conjuntos empregados no treinamento dos modelos. Por fim, são descritos os meta-classificadores binários utilizados e os procedimentos adotados para sua avaliação.

3.1. Conjuntos de dados e protocolo de perturbação

O estudo foi conduzido com dois conjuntos de dados tabulares, com características distintas, a fim de avaliar a generalidade da abordagem proposta. O primeiro, *Diabetes 130-US Hospitals*, contém 101.766 registros de internações hospitalares descritos por atributos numéricos e categóricos, tendo como variável-alvo uma classificação em três categorias relacionadas à readmissão do paciente. O segundo, *Forest Covertype*, é composto originalmente por 581.012 observações cartográficas referentes a sete tipos de cobertura florestal, representadas exclusivamente por atributos numéricos. Devido às limitações de memória e de tempo de processamento, utilizou-se uma amostra estratificada de 120.000 observações do Covertype (aproximadamente 20% do conjunto original), preservando-se a distribuição das sete classes.

A geração de instâncias com problemas de qualidade seguiu o protocolo de perturbação de rótulos amplamente empregado na literatura sobre ruído supervisionado [Frénay and Verleysen 2014, Seedat et al. 2024]. Inicialmente, cada conjunto de dados original é mantido inalterado, formando o *Bloco A*, cujas instâncias recebem o rótulo *problema_qualidade* = `não`. Em seguida, são geradas três versões perturbadas do conjunto, denominadas *Bloco B*, nas quais parte das instâncias tem sua classe modificada por meio de perturbação uniforme (*uniform mislabeling*). Nesse procedimento, o novo rótulo é selecionado aleatoriamente, com probabilidade uniforme, dentre todas as classes distintas da classe verdadeira. Foram consideradas três taxas de perturbação, $p \in \{0{,}1,\ 0{,}2,\ 0{,}3\}$, produzindo três conjuntos perturbados independentes. Apenas as instâncias efetivamente modificadas são incorporadas aos respectivos blocos e recebem o rótulo *problema_qualidade* = `sim`. O conjunto final utilizado nas etapas subsequentes corresponde à união do conjunto original com os três blocos perturbados, isto é, Bloco A $\cup$ Bloco B10% $\cup$ Bloco B20% $\cup$ Bloco $\mathrm{B}_{30\%}$.

Como resultado, obtêm-se metaconjuntos com 162.826 instâncias para o *Diabetes 130-US Hospitals* e 192.000 instâncias para o *Forest Covertype*. A variável *problema_qualidade* constitui o *ground truth* binário utilizado na avaliação dos métodos de detecção de instâncias potencialmente problemáticas. Essa união faz que um mesmo registro-base (*encounter_id* no *Diabetes*, *instance_id* no *Covertype*) gere múltiplas meta-instâncias no pool experimental: cerca de 30% dos registros em ambas as bases aparecem mais de uma vez, com versões "limpa" e "ruidosa" coexistindo no mesmo conjunto, ao contrário do que ocorre na triagem operacional. O meta-classificador recebe apenas o vetor de medidas de *hardness*, e não os atributos brutos das bases originais; assim, não se trata de *leakage* clássico de vazamento de atributos entre treino e teste. Ainda assim, versões derivadas de um mesmo registro podem compartilhar informação estrutural via meta-atributos: a perturbação de rótulo frequentemente altera medidas que dependem de vizinhança e estrutura local, mas, em parte dos registros repetidos, o vetor de *hardness* coincide entre versões enquanto *problema_qualidade* difere.

3.2. Extração de medidas de dificuldade

A implementação original do PyHard [Paiva et al. 2022] calcula as medidas baseadas em distância a partir da matriz completa de distâncias de Gower entre todas as N instâncias do conjunto de dados. Como essa matriz possui dimensão $N \times N$, o consumo de memória cresce quadraticamente com o número de instâncias, resultando em complexidade espacial de $O(N^2)$. Para contornar essa limitação, adotou-se uma estratégia de processamento em lotes. Nesse procedimento, o conjunto de dados é particionado em blocos de instâncias e, para cada bloco, calcula-se apenas a matriz retangular de distâncias entre as instâncias do lote e o conjunto completo. Dessa forma, em vez de materializar uma matriz de dimensão $N \times N$, utiliza-se uma matriz de dimensão $(\text{lote} \times N)$. Essa estratégia foi utilizada para as medidas baseadas em vizinhança, uma vez que seu cálculo depende apenas das distâncias entre cada instância e seus vizinhos mais relevantes. Em contrapartida, as medidas baseadas em árvores de decisão e em verossimilhança não exigem a construção da matriz completa de distâncias e, portanto, podem ser calculadas diretamente pela implementação original da ferramenta.

O H-CAT [Seedat et al. 2024] caracteriza a dificuldade das instâncias com base na dinâmica de treinamento de um modelo preditivo. Neste trabalho, foi utilizada a configuração padrão disponibilizada pelos autores, baseada em um *Multi-Layer Perceptron* (MLP) para dados tabulares. O protocolo de treinamento seguiu a implementação de referência do H-CAT, preservando a arquitetura da rede, os hiperparâmetros e o procedimento de otimização originalmente propostos, sem modificações no laço de treinamento.

3.3. Meta-classificador

A hipótese central desta pesquisa é que medidas de *instance hardness* podem ser utilizadas como atributos de um classificador supervisionado capaz de identificar instâncias com problemas de qualidade. Em vez de selecionar previamente um único indicador de dificuldade, o modelo aprende automaticamente relações, pesos e interações entre os diferentes descritores extraídos por cada ferramenta.

A construção dos meta-conjuntos segue essa premissa. Cada instância dos conjuntos originais é representada por um vetor de medidas de *hardness*, acompanhado da

variável *problema_qualidade*, que indica se a instância teve ou não o rótulo alterado durante o processo de perturbação. Assim, as medidas de dificuldade constituem os atributos preditivos, enquanto a coluna *problema_qualidade* é utilizada como variável-alvo em um problema de classificação binária. Como resultado, são gerados dois meta-conjuntos para cada base de dados: um contendo exclusivamente os descritores extraídos pelo PyHard e outro contendo apenas os descritores produzidos pelo H-CAT. Essa separação permite avaliar, de forma independente, a capacidade preditiva de medidas baseadas na estrutura dos dados e de medidas derivadas da dinâmica de treinamento dos modelos.

Para avaliar a hipótese proposta, foram considerados quatro algoritmos de classificação supervisionada para gerar os meta-modelos: *Logistic Regression*, *Random Forest*, *Gradient Boosting* e *XGBoost*. O conjunto inclui tanto abordagens lineares quanto métodos de *ensemble* baseados em *bagging* e *boosting*, reduzindo a influência de um único viés indutivo sobre os resultados observados. Como a classe positiva (*problema_qualidade* = `sim`) é minoritária, foram empregadas estratégias de balanceamento durante o treinamento. Nos modelos *Logistic Regression* e *Random Forest*, utilizou-se o parâmetro *class_weight* = *balanced*, enquanto no *XGBoost* adotou-se o parâmetro *scale_pos_weight*, ajustado de acordo com a razão entre as classes no conjunto de treinamento.

O desempenho dos modelos foi estimado por meio de validação cruzada estratificada em 5 *folds*. As partições foram definidas no nível das meta-instâncias, sem agrupamento por registro-base. Nessas divisões, a versão do Bloco A e as versões perturbadas dos Blocos B de um mesmo registro podem cair em *folds* distintos de treino e teste. Isso não equivale a repetir em treino e teste a mesma meta-instância com o mesmo vetor de *hardness* e a mesma saída (*problema_qualidade*). As medidas de dificuldade foram calculadas sobre o conjunto completo antes da partição. Os resultados *in-domain* reportados refletem esse desenho de avaliação.

Em cada execução foram calculadas as métricas de acurácia, precisão, *recall* e F1 para a classe positiva, além da área sob a curva ROC (AUC-ROC) e da área sob a curva Precisão-Revocação (AUC-PRC). As métricas foram posteriormente agregadas pela média aritmética dos *folds*. As métricas AUC-ROC e AUC-PRC foram adotadas como principais critérios de avaliação. A primeira quantifica a capacidade geral de separação entre as classes, independentemente de sua prevalência, enquanto a segunda é particularmente adequada para cenários desbalanceados, fornecendo uma visão mais sensível do desempenho em relação à classe minoritária. Essa característica é especialmente relevante em aplicações de triagem de qualidade, nas quais deixar de identificar instâncias potencialmente problemáticas tende a ser mais prejudicial do que sinalizar algumas observações adicionais para inspeção.

Além da validação cruzada, realizou-se um experimento de transferência de domínio bidirecional para avaliar a capacidade de generalização dos modelos. Nesse cenário, o classificador treinado no meta-conjunto de um conjunto de dados é aplicado diretamente ao meta-conjunto do outro, sem qualquer etapa de adaptação ou ajuste, nos dois sentidos: *Diabetes 130-US Hospitals* $\rightarrow$ *Forest Covertype* e *Forest Covertype* $\rightarrow$ *Diabetes 130-US Hospitals*. O procedimento é executado separadamente para os descritores produzidos pelo PyHard e pelo H-CAT, uma vez que cada ferramenta gera conjuntos distintos de atributos. A comparação entre os resultados obtidos nesses regimes e os observados

na validação cruzada permite avaliar até que ponto os padrões aprendidos refletem características específicas do conjunto de treinamento ou propriedades mais gerais associadas à dificuldade das instâncias.

4. Resultados dos meta-classificadores

Esta seção apresenta os resultados obtidos pelos meta-classificadores nos regimes experimentais considerados: (i) avaliação *in-domain*, realizada por meio de validação cruzada estratificada para cada combinação de conjunto de dados e fonte de descritores; e (ii) transferência de domínio bidirecional, na qual os modelos treinados no meta-conjunto de um conjunto são avaliados diretamente no meta-conjunto do outro.

Nos experimentos *in-domain*, os resultados correspondem à média da validação cruzada; na transferência, correspondem ao desempenho no conjunto de teste. A Tabela 3 apresenta os resultados da avaliação *in-domain*. Neste cenário, os meta-classificadores construídos a partir dos descritores do PyHard apresentam desempenho consistentemente elevado nos dois conjuntos de dados. Os valores de AUC-ROC variam de 0,745 a 0,861, enquanto a AUC-PRC varia de 0,500 a 0,703. O melhor desempenho é obtido pelo *Gradient Boosting*, que atinge AUC-ROC de 0,861 no conjunto *Diabetes* e de 0,834 no *Covertype*. Em ambos os casos, os resultados indicam uma capacidade substancial de distinguir instâncias perturbadas de não perturbadas utilizando apenas os vetores de medidas de dificuldade. Em contraste, os descritores extraídos pelo H-CAT apresentam desempenho próximo ao do acaso em praticamente todas as configurações avaliadas. Os maiores valores de AUC-ROC observados não ultrapassam 0,536, enquanto as métricas AUC-PRC permanecem próximas da prevalência da classe positiva. Esses resultados sugerem que, para os conjuntos tabulares considerados, os descritores baseados na dinâmica de treinamento do H-CAT capturam pouco sinal discriminativo associado à presença de ruído.

Conjunto	Ferramenta	Modelo	Acurácia	Precisão	*Recall*	F1	AUC-ROC	AUC-PRC
Diabetes	PyHard	*Gradient Boosting*	**0,812**	**0,714**	**0,834**	**0,769**	**0,861**	**0,703**
Diabetes	PyHard	XGBoost	0,803	0,694	0,850	0,764	0,846	0,662
Diabetes	PyHard	*Logistic Regression*	0,812	0,713	0,834	0,769	0,834	0,687
Diabetes	PyHard	*Random Forest*	0,731	0,623	0,714	0,666	0,745	0,500
Covertype	PyHard	*Gradient Boosting*	**0,812**	**0,714**	0,833	**0,769**	**0,834**	0,678
Covertype	PyHard	*Logistic Regression*	0,796	0,685	**0,847**	0,757	0,829	**0,683**
Covertype	PyHard	XGBoost	0,809	0,708	0,835	0,766	0,818	0,639
Covertype	PyHard	*Random Forest*	0,735	0,625	0,732	0,674	0,705	0,480
Diabetes	H-CAT	*Random Forest*	0,603	0,443	0,229	0,302	0,536	0,403
Diabetes	H-CAT	XGBoost	0,507	0,378	0,486	0,425	0,507	0,378
Diabetes	H-CAT	*Gradient Boosting*	0,625	0,280	0,001	0,001	0,500	0,375
Diabetes	H-CAT	*Logistic Regression*	0,507	0,373	0,463	0,412	0,497	0,373
Covertype	H-CAT	*Random Forest*	0,600	0,438	0,237	0,308	0,536	0,401
Covertype	H-CAT	XGBoost	0,507	0,381	0,504	0,434	0,510	0,379
Covertype	H-CAT	*Gradient Boosting*	0,625	0,325	0,001	0,002	0,499	0,374
Covertype	H-CAT	*Logistic Regression*	0,496	0,376	0,518	0,434	0,499	0,374

Tabela 3. Desempenho *in-domain* dos classificadores meta por conjunto e ferramenta.

Outro aspecto relevante é a consistência dos resultados obtidos com o PyHard nos dois domínios analisados. Apesar das diferenças substanciais entre os conjuntos *Diabetes* e *Covertype*, os níveis de desempenho permanecem semelhantes, indicando que os padrões associados à dificuldade das instâncias apresentam comportamento relativamente estável entre diferentes aplicações. Para complementar essa observação, foram realiza-

dos experimentos de transferência de domínio. Os resultados são apresentados na Tabela 4. Com descritores do PyHard, o desempenho permanece substancialmente elevado em ambos os sentidos. No sentido *Diabetes* $\rightarrow$ *Covertype*, o *Gradient Boosting* alcança AUC-ROC de 0,834 e AUC-PRC de 0,686, valores praticamente idênticos aos observados *in-domain* no Covertype. No sentido inverso *Covertype* $\rightarrow$ *Diabetes*, o mesmo modelo atinge AUC-ROC de 0,859 e AUC-PRC de 0,701, próximos dos 0,861 e 0,703 obtidos *in-domain* no Diabetes. Com descritores do H-CAT, o desempenho permanece em torno do acaso em ambas as direções (AUC-ROC $\approx 0{,}50$, AUC-PRC $\approx 0{,}375$), reforçando as evidências observadas na validação cruzada.

Direção	Ferramenta	Melhor modelo	AUC-ROC	AUC-PRC
Diabetes $\rightarrow$ Covertype	PyHard	*Gradient Boosting*	**0,834**	**0,686**
Covertype $\rightarrow$ Diabetes	PyHard	*Gradient Boosting*	**0,859**	**0,701**
Diabetes $\rightarrow$ Covertype	H-CAT	*Gradient Boosting*	0,500	0,376
Covertype $\rightarrow$ Diabetes	H-CAT	*Gradient Boosting*	0,502	0,376

Tabela 4. Transferência de domínio bidirecional (melhor modelo por direção e fonte de descritores).

A proximidade entre os resultados obtidos nos cenários *in-domain* e de transferência, em ambos os sentidos, sugere que os meta-classificadores treinados sobre descritores do PyHard aprendem padrões relacionados à própria estrutura das medidas de dificuldade, e não apenas características específicas do domínio clínico ou florestal. Em contrapartida, os descritores do H-CAT permanecem próximos ao desempenho aleatório em ambos os regimes de transferência.

A Figura 1 apresenta os resultados em termos de *lift*@k, métrica particularmente relevante em cenários de triagem, para $k \in \{5\%, 10\%, 20\%, 30\%\}$, nos quatro cenários avaliados: dois *in-domain* e duas transferências bidirecionais. Os modelos construídos a partir dos descritores do PyHard mantêm valores próximos de 1,9 em todas as faixas de k e em todos os cenários, inclusive na transferência de domínios. Em termos práticos, isso significa que os modelos conseguem identificar aproximadamente o dobro de instâncias perturbadas em comparação com uma estratégia de seleção aleatória, com o mesmo orçamento de revisão. Por outro lado, os classificadores baseados nos descritores do H-CAT apresentam valores de *lift* próximos de 1,0 em todos os cenários, sem oferecer vantagem operacional significativa em relação à inspeção aleatória.

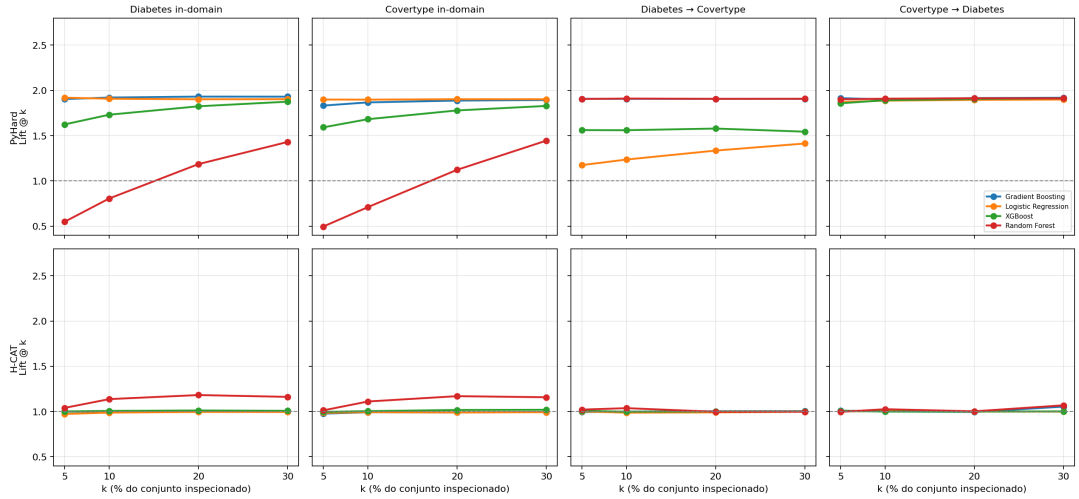


Figura 1. *Lift@*k dos meta-classificadores para os quatro modelos avaliados, nos quatro cenários: Diabetes *in-domain*, Covertype *in-domain* e transferências bidirecionais (Diabetes $\rightarrow$ Covertype e Covertype $\rightarrow$ Diabetes). Painel superior: trilha PyHard; painel inferior: trilha H-CAT. A linha horizontal pontilhada indica o desempenho da seleção aleatória (1,0).

A Tabela 5 consolida, por cenário e ferramenta, o melhor desempenho em AUC-ROC e *lift*@10%. Sob a perspectiva de aplicações em *Data-Centric AI*, os resultados obtidos fornecem evidências favoráveis à utilização de meta-classificadores baseados em medidas de *instance hardness* para apoiar processos de curadoria de dados. Considerando um orçamento de inspeção correspondente a apenas 10% das instâncias, os modelos construídos com descritores do PyHard recuperam aproximadamente 19% das observações perturbadas, praticamente dobrando a eficiência esperada de uma seleção aleatória, e mantêm ganho semelhante sob transferência bidirecional entre os dois conjuntos.

Cenário	Ferramenta	AUC-ROC	*Lift* @ 10%
Diabetes *in-domain*	PyHard	**0,861**	**1,92**
Diabetes *in-domain*	H-CAT	0,536	1,14
Covertype *in-domain*	PyHard	**0,834**	**1,90**
Covertype *in-domain*	H-CAT	0,535	1,11
Transferência Diab. $\rightarrow$ Cov.	PyHard	**0,834**	**1,91**
Transferência Diab. $\rightarrow$ Cov.	H-CAT	0,500	1,04
Transferência Cov. $\rightarrow$ Diab.	PyHard	**0,859**	**1,91**
Transferência Cov. $\rightarrow$ Diab.	H-CAT	0,502	1,03

Tabela 5. Melhor classificador meta por (*cenário*, *ferramenta*), em AUC-ROC e *lift*@10%.

Além do ganho em eficiência, os resultados demonstram robustez à mudança de domínio nos dois sentidos de transferência e estabilidade para diferentes valores de k, características desejáveis em sistemas de apoio à revisão de dados. Uma vez calculadas as medidas de dificuldade, o custo de inferência do meta-classificador é reduzido, o que favorece sua utilização como etapa inicial de filtragem em fluxos de curadoria, direcionando a atenção dos especialistas para as instâncias com maior probabilidade de apresentar problemas de qualidade.

5. Conclusão

Este trabalho investigou a viabilidade de usar medidas de *instance hardness* como entrada para meta-classificadores destinados à identificação de instâncias com ruído de rótulo em cenários de *Data-Centric AI*. Para isso, foi desenvolvido um *pipeline* capaz de gerar meta-conjuntos a partir de descritores extraídos pelas ferramentas PyHard e H-CAT, treinar classificadores supervisionados e avaliar sua capacidade de distinguir instâncias originais de instâncias submetidas a perturbações controladas de rótulo.

Os resultados obtidos demonstram que os descritores fornecidos pelo PyHard contêm informação discriminativa suficiente para apoiar processos de triagem de qualidade em dados tabulares. Os melhores modelos alcançaram valores de AUC-ROC de até 0,861 e *lift*@10% próximo de 1,9, indicando que é possível identificar aproximadamente o dobro de instâncias perturbadas em comparação com uma estratégia de seleção aleatória sob o mesmo orçamento de inspeção. Além disso, o desempenho permaneceu estável no experimento de transferência bidirecional entre os conjuntos de dados, sugerindo que os padrões aprendidos estão associados às próprias medidas de dificuldade, e não apenas às características específicas de cada domínio.

Em contraste, os descritores extraídos pelo H-CAT apresentaram desempenho próximo ao do acaso nos conjuntos tabulares avaliados, tanto *in-domain* quanto nas duas direções de transferência (AUC-ROC $\approx 0{,}50$, *lift*@10% $\approx 1{,}0$). Embora essa observação não invalide a utilidade da ferramenta em outros contextos, ela indica que medidas baseadas na dinâmica de treinamento, ao menos na configuração utilizada neste estudo, fornecem menos informação para a identificação de ruído de rótulo em dados tabulares do que as medidas estruturais derivadas diretamente da distribuição dos dados.

As conclusões deste estudo devem ser interpretadas à luz de algumas limitações. Os experimentos foram conduzidos com apenas dois conjuntos de dados tabulares e utilizaram exclusivamente perturbações sintéticas de rótulo geradas por meio de ruído uniforme. As métricas *in-domain* (AUC-ROC e *lift*@k) refletem o desenho experimental discutido anteriormente e podem diferir do desempenho em triagem operacional, em que cada observação aparece uma única vez. Como trabalhos futuros, destacam-se três direções principais. A primeira consiste em avaliar a abordagem em mais conjuntos de dados e em outras modalidades de dados. A segunda envolve a investigação de mecanismos de ruído mais complexos, como perturbações assimétricas ou dependentes dos atributos, permitindo analisar a robustez dos meta-classificadores sob diferentes estruturas de erro. Inclui-se também a reavaliação *in-domain* com validação cruzada agrupada por registro-base (`GroupKFold`), para quantificar o efeito desse desenho sobre AUC-ROC e *lift*@k. Por fim, pretende-se consolidar o *pipeline* desenvolvido em um *toolkit* reprodutível que integre a geração de meta-conjuntos, o treinamento dos classificadores e as rotinas de triagem, facilitando a aplicação da abordagem e a incorporação de novos descritores de dificuldade em cenários de *Data-Centric AI*.

Agradecimentos

O presente trabalho foi realizado com apoio da Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Brasil. Processo n. 2021/06870-3. O trabalho também contou com o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processo 310065/2023-2.

Referências

Frénay, B. and Verleysen, M. (2014). Classification in the presence of label noise: A survey. *IEEE Trans. on Neural Networks and Learning Systems*, 25(5):845–869.

Jain, N., Suggala, A. S., and Shenoy, P. (2024). Improving generalization via meta-learning on hard samples. *https://arxiv.org/abs/2403.12236*.

Jiang, H. and Nachum, O. (2020). Identifying and correcting label bias in machine learning. In *Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 702–712.

Lorena, A. C., Paiva, P. Y., and Prudêncio, R. B. (2024). Trusting my predictions: on the value of instance-level analysis. *ACM Computing Surveys*, 56(7):1–28.

Paiva, P. Y. A., Moreno, C. C., Smith-Miles, K., Valeriano, M. G., and Lorena, A. C. (2022). Relating instance hardness to classification performance in a dataset: a visual approach. *Machine Learning*, 111(8):3085–3123.

Paul, M., Ganguli, S., and Dziugaite, G. K. (2021). Deep learning on a data diet: Finding important examples early in training. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34, pages 20596–20607.

Pleiss, G., Zhang, T., Elenberg, E. R., and Weinberger, K. Q. (2020). Identifying mislabeled data using the Area Under the Margin ranking. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, pages 17044–17056.

Polyzotis, N., Roy, S., Whang, S. E., and Zinkevich, M. (2017). Data management challenges in production machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data (SIGMOD)*, pages 1723–1726.

Seedat, N., Crabbé, J., Bica, I., and van der Schaar, M. (2022). Data-IQ: Characterizing subgroups with heterogeneous outcomes in tabular data. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.

Seedat, N., Imrie, F., and van der Schaar, M. (2024). Dissecting sample hardness: A fine-grained analysis of hardness characterization methods for data-centric ai. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Smith, M. R., Martinez, T., and Giraud-Carrier, C. (2014). An instance level analysis of data complexity. *Machine Learning*, 95(2):225–256.

Sorscher, B., Geirhos, R., Shekhar, S., Ganguli, S., and Morcos, A. S. (2022). Beyond neural scaling laws: Beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.

Toneva, M., Sordoni, A., Combes, R. T. d., Trischler, A., Bengio, Y., and Gordon, G. J. (2019). An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations (ICLR)*.

Zha, D., Bhat, Z. P., Lai, K.-H., Yang, F., and Hu, X. (2023). Data-centric ai: Perspectives and challenges. In *Proceedings of the 2023 SIAM international conference on data mining (SDM)*, pages 945–948. SIAM.

Zha, D., Bhat, Z. P., Lai, K.-H., Yang, F., Jiang, Z., Zhong, S., and Hu, X. (2025). Data-centric artificial intelligence: A survey. *ACM Computing Surveys*, 57(5). 42 pages.